UNIVERSITÉ DE NEUCHÂTEL
INSTITUT DE MICROTECHNIQUE

Robust Template Matching in Automated Audience Measurement

Dequn Sun

THÈSE PRÉSENTÉE À LA FACULTÉ DES SCIENCES POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

Février 2001

IMPRIMATUR POUR LA THESE

Robust Template Matching in Automated Audience Measurement

de M. Dequn Sun

UNIVERSITE DE NEUCHATEL

FACULTE DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel sur le rapport des membres du jury,

MM. F. Pellandini (directeur de thèse),
N. de Rooij, P. Balsiger, M. Bichsel (ETH Zürich) et
J. Feitknecht (Liechti Ag, Kriegstetten)

autorise l'impression de la présente thèse.

Neuchâtel, le 23 février 2001

Le doyen:

J.-P. Derendinger

To Anne Gottraux

Acknowledgements

First of all, I would like to thank Prof. F. Pellandini, Prof. N. de Rooij, Dr. P. Balsiger, Dr. M. Bichsel and Dr. J. Feitknecht, who form the examination committee, for their interests, efforts and contributions during the co-examination of this thesis.

Many people have helped me to realize this PhD work and provided advice and encouragement. I would like to express my gratitude to the following persons:

- Professor Fausto Pellandini, my supervisor, for his clear vision and sensible way of approaching problems, and for his great art of managing human resources and his grand humanity towards all his collaborators. All these have formed the fertile soil on which a healthy and pleasant working ambiance has been cultivated.
- Dr. Peter Balsiger, the project manager, for putting his trust in me, and for his encouragement supporting me to carry out this dissertation.
- Dr. Martin Bichsel, the initiator of the present work, for his kind help providing clear guidelines to the understanding of the material.
- Dr. Michael Ansorge for his very valuable review and comments on this thesis.
- Dr. Birgit Marxer for English correction of this dissertation.
- Mrs. Anna Lieti-Staffelbach & Mr. Laurent Staffelbach for correction of the Résumé in French.

- Mr. Christophe Calame for the direct contribution to the work described in this report, and for the friendship built during the days we have worked together.
- All the collaborators of the laboratory of Electronics and Signal Processing of the institute for the friendly working ambience. In particular Alain Dufaux, Louisa Grisoni, Sara Grassi, Catherine Marcelli, Manuel Gomez, and Eric Meurville, who are related directly to the achievement of this work.
- My darling wife Meng Hua for her love.

Finally, I would like extend my special thanks to Switzerland who allowed me to realise one of my dreams.

Abstract

This PhD thesis presents a new, acoustic pattern matching approach to the media audience-rating automation problem.

The constructed pattern matching system identifies media audiences by comparing wristwatch-recorded sound with studio-recorded one using a robust, Bayesian template matching technique. High-ratio (285) data reduction is achieved on the watch, allowing the audience data storage for one-week non-stop functioning, while the system is invariant to various disturbances such as harmonic distortion, echo, time shift, frequency dependent sound power damping, white noise, colored and structured noise. The resulting classification error is less than 4%.

The key points conducting to the industrially proved success of the approach are: its simple, efficient feature extraction and data compression scheme; its robust, Laplacian conditional distance metric; and its low-power featured implementation.

Comparing to the competing “code insertion” approach, the main advantages that this present approach can provide are the quasi full passive audience measurement and the adaptability to media technical environment.

Résumé

Cette thèse de doctorat présente une nouvelle approche de l'automatisation des mesures d'audience, via des techniques de *mise en correspondance* robuste des signaux acoustiques.

Ce système de *mise en correspondance* identifie l'audience de différents médias (TV, radio, etc.) en comparant le son enregistré par une montre spéciale avec celui enregistré par le studio. Ce système est rendu robuste grâce à la technique de *mise en correspondance* dérivée de la théorie de Bayes. Il atteint un taux élevé (285) de réduction des données dans la montre, ce qui permet le stockage d'une semaine ininterrompue de données d'audience. En même temps, le système est insensible aux différentes perturbations ambiantes inhérentes à ce type d'enregistrement : distorsion harmonique, écho, imprécision de temps, amortissement de son selon fréquence, bruit blanc, bruit coloré et structuré. L'erreur de classification du système est inférieure à 4%.

Les éléments clé conduisant à la réussite de cette approche, prouvée industriellement, sont : une procédure simple et efficace d'extraction de caractéristiques, une méthode simple et judicieuse de compression de données, une mesure robuste de distance conditionnelle laplacian, et une implantation à faible consommation.

Comparés à l'approche concurrentielle « par insertion de code » les principaux avantages offerts par cette approche sont la mesure quasi totalement passive et l'adaptabilité à l'environnement technique des médias.

Contents

Chapter 1
Introduction

This thesis addresses the acoustic signal recognition (detection) problem in the context of an automated TV/radio audience measurement system, as well as all related digital signal processing problems.

In this introductory chapter, we will explain the main motivation, give an overview of the proposed approach and a reader's guide to the text structure accompanied by looking through the main line of the algorithm. Finally, an alternative state-of-the-art approach is discussed and commented.

1.1. Motivation

Media audience-rating information is regularly required by TV and Radio stations to help them create the types of programs their audiences enjoy, and by advertisers that want to reach people who might be interested in their products.

The traditional methods for collecting media audience-rating information such as interview, self-completion diary and aided recall questionnaire etc. require heavy cooperation and compliance from the participant and furthermore, the results could neither be reliable nor precise in time.

With today's available digital signal processing facilities, high speed computation techniques and notably the advance of low-power consumption electronic devices, an acoustic pattern recognition based

approach has become one of the possibilities to realise a completely automated "electronic diary".

Comparing to the "code insertion" approach (see Section 1.4), the main advantages that this present approach can provide are:

1. Quasi full passive audience measurement: Practically no co-operation, no compliance would be required from neither the participant part nor the TV/radio station part;
2. Adaptability to media technical environment: This system is totally independent from the evolution of the techniques and industrial standards involved in the audio media domain, which include the medium type, data compression scheme and all other related signal processing techniques.

Pattern recognition is a vast domain. Innumerable contributions have been made during last 4 decades in various disciplines [Duda73]. However, each application area has its unique characteristics that mould and shape the corresponding approach. No literature could be found in an application similar to the pattern recognition based audience measurement problem. This situation explains the motivation of the present work.

1.2. Overview of the proposed approach

The principle of the proposed audience measurement system is illustrated in Figure 1-1.

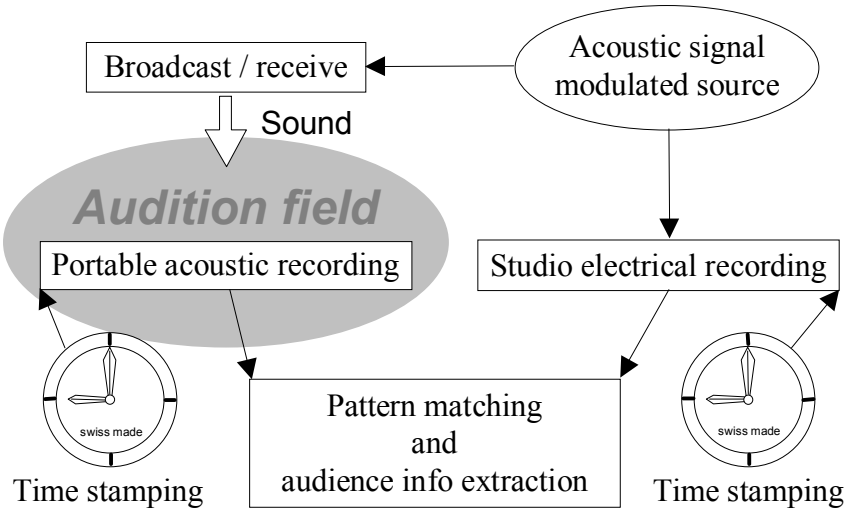


Figure 1-1: Automated audience measurement based on pattern matching

As it can be seen, the input front-end of the system is composed of a studio recording unit and a portable audience recording unit. At one hand, the studio unit records reference audio signals within some given time slots. At the other hand, carried by a selected participant, the audience unit performs the audience recording within the same time slots but in a real-life audition environment. A pattern matching algorithm is then run between these two-path recorded signals to produce a match value indicating whether or not the participant has listened to the program performed by that studio in a particular time slot.

Obviously, the time stamping at each recording side is of limited precision. In order to absorb the time stamping error, the studio recording time slot must be longer than and cover the audience one. And consequently, the pattern matching process should run in a search scanning mechanism.

We are therefore confronted to the problem of massive, multi-type (music, speech, and all kinds of environmental sounds), but with known reference (template), acoustic signal recognition (or detection) in a heavily noisy environment.

In this approach we identify three crucial requirements:

1. User-friendly portable audience recording unit: small sized (wristwatch), low power consumption (one to two weeks non-stop functioning);
2. High reliable matching result: robust against various noises;
3. Reasonable matching process complexity: fast enough to deal with a huge amount of data to be matched.

1.3. Overview of the algorithm and thesis structure

In order to fulfil the above requirements, the recorded data must be pre-processed before entering into the matching process so that the amount of data is reduced and the matching process becomes more efficient (quality and speed). It is the so-called *feature extraction* procedure that accomplishes this task. At this level, the algorithm has been designed to exploit the sub-band time domain structure of the signal. Its principle operations are illustrated in Figure 1-2. For sake of illustration, both studio recorded data and audience recorded data paths have been placed into the same data flowchart. At left is the studio data path, at right the audience data path. Common operations are placed in the middle. The part

before “Data storage and transfer” is implemented in recording units and the part after it is implemented in matching program.

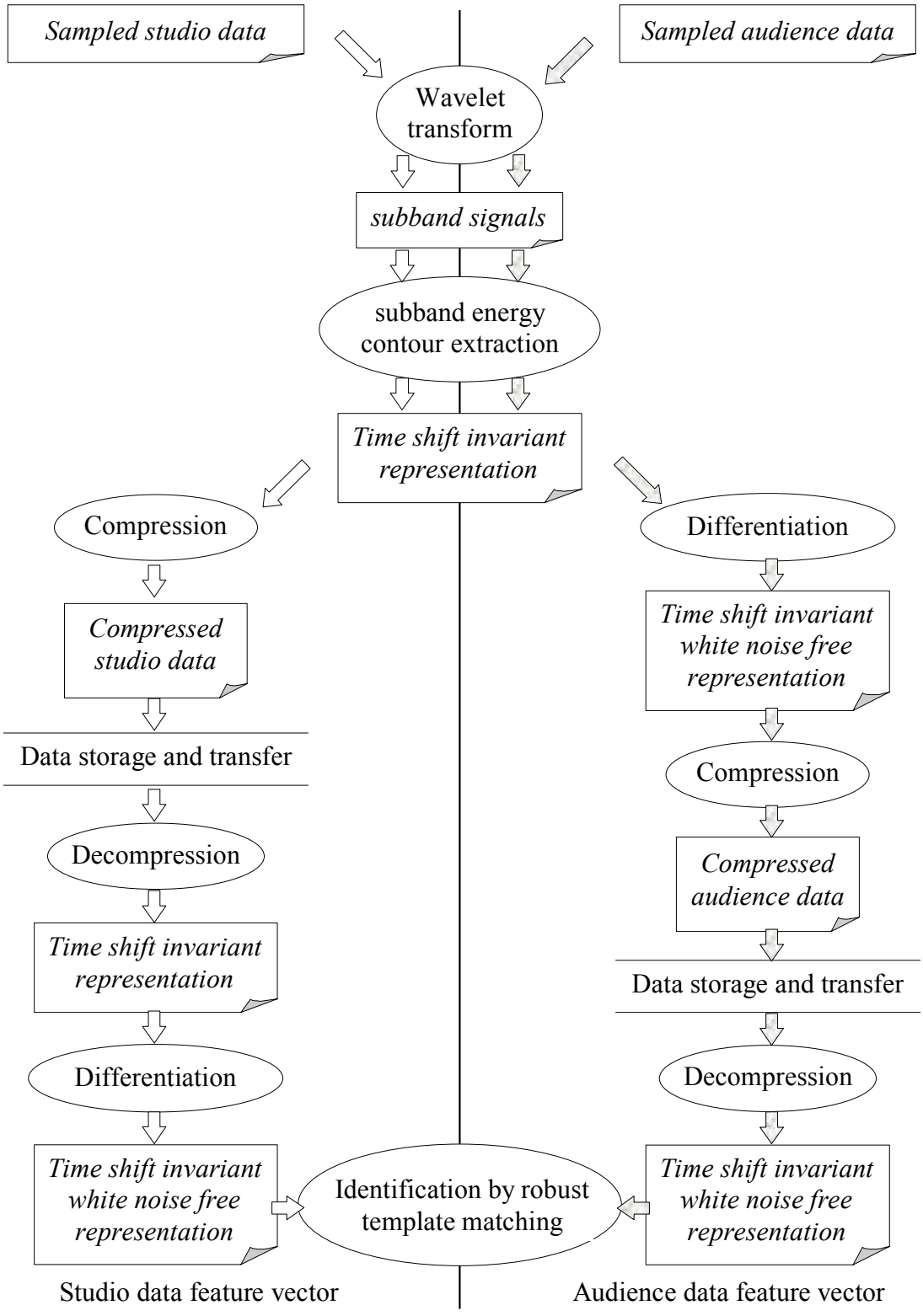


Figure 1-2: Feature extraction part of the algorithm.

As can be seen in Figure 1-2, the feature extraction consists mainly of wavelet transform, subband energy contour extraction and derivation to obtain a time shift invariant, white noise free representation. The resulting data forms the so-called *feature vector*. The conception of feature extraction and all related signal processing tools will be fully explained in Chapter 3.

The data compression, subject of Chapter 4, is incorporated in the algorithm to further facilitate data storage and transfer. Their design influences significantly the overall pattern matching quality. Notice that, in Figure 1-2, the data compression on the studio data path is applied earlier than on the audience data path. And the corresponding compression schemes are accordingly different. This situation is a result of the matching quality requirement and natural choices of compression schemes as will be explained in Chapter 3 and Chapter 4.

After the feature extraction process the resulting feature vectors are fed forwards into the pattern matching process (Figure 1-3).

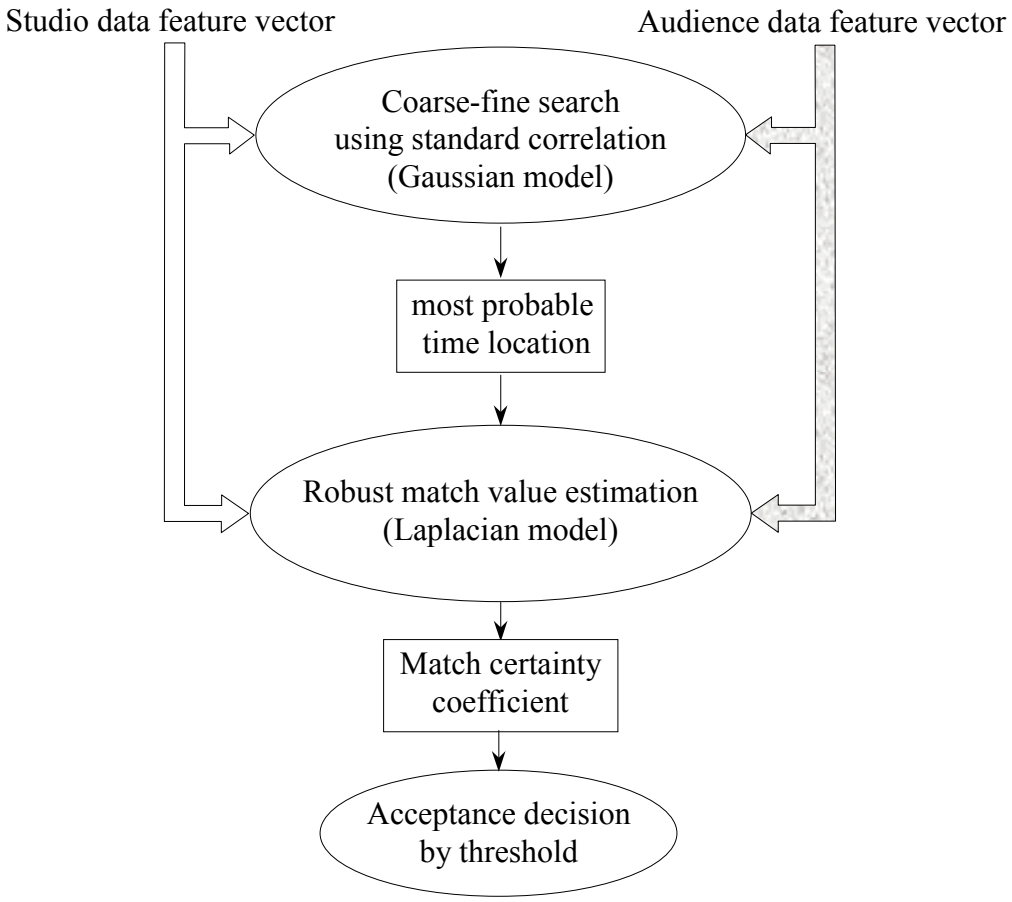


Figure 1-3: Pattern matching part of the algorithm

In order to efficiently obtain a reliable matching result, we have adopted the strategy called *robust template matching*. As illustrated in Figure 1-3, the robust template matching is a coarse-fine searching process involving two statistical models (Gaussian and Laplacian). This is a compromise of speed and robustness. We use the fast Gaussian model to find the time location of the audience signal with respect to studio signal and use the robust Laplacian model to compute the similarity measure, *match certainty coefficient* (or simple *match value*), at the found time location. Chapter 5 describes in detail all related issues.

The so-produced match value indicates the similarity level between the audience data and studio data. We decide the acceptance of "having listened" if the similarity is above a given threshold. Chapter 6 will discuss the acceptance threshold elaboration together with the system performance evaluation, since these two tasks can be accomplished by solving a common problem. Several other important simulation results will be also presented in Chapter 6.

Chapter 7 focuses on some important implementation issues leading to correct and efficient functioning of the system.

Finally, Chapter 8 concludes the work and gives final remarks.

The dedicated pattern matching system is built on a statistical basis. In order to give a clear guideline at very beginning, Chapter 2 will provide a general review of fundamental concept of pattern recognition and outline the principle axes of the material that will be developed in this thesis.

The other detailed theoretical reviews of various applied tools will be introduced just before they are employed to keep the development of different subjects natural and ease to be understood.

1.4. Alternative approach [Arbi]

Since 1992, the American firm, the Arbitron company, has developed an automated audience measurement system using a different approach. This approach consists in dynamically inserting a code onto the audio source signal, over a range of multiple frequencies. The code is rendered inaudible to the human ear through the use of the well-known psycho-acoustic masking phenomenon. That is, it takes advantage of the human ear's inability to discern a slightly weaker frequency component of a sound signal that is immediately adjacent to a stronger frequency component. A portable electronic device tuned to listen at a specific fre-

quency is able to detect the weaker sound that the human ear cannot perceive.

While this approach represents an advance of fast measuring audiences, it is not fully passive. It still requires cooperation from the survey participants and from the TV/radio stations. People must be convinced to keep a unusual portable meters (pager sized) with them at all times, and the station identification code must be embedded onto the sound sources.

The technical requirements for the audio encoding scheme of this approach are severe. Notably, the codes had to survive the most hostile environments, which include digital-to-analogue and analogue-to-digital conversions, steady state time compression and expansion, as well as all of the data reduction techniques (such as MPEG) being used or contemplated by the industry. Furthermore, the coding scheme had to conform to all industry standards for playing back and had to be capable of encoding monophonic, stereophonic and multi-channel signals. All these requirements make this code insertion approach non-flexible.

So far, their portable meter is quite power consuming. Daily recharge is necessary.

No detection performance is reported.

Chapter 2
Definitions and theoretical guideline

This chapter gives theoretical concepts of pattern recognition and defines technical terms used in this thesis. We will briefly review the well-known Bayes decision theory and show how the proposed approach, the robust template matching, is guided by this theory.

2.1. Introduction to pattern recognition

The objective of pattern recognition is to classify real world entities into a number of categories[1] or *classes* by machine. Depending on the application, these entities can be represented by images in the case of visual scene, by acoustical waveforms in the case of sound signal or by any other type of measurements. In almost all applications these measurements must be further processed into reduced number of discernible quantities, called *features*, so as to facilitate the classification task. In this discipline *pattern* is used as a generic term that refers to the set of features (feature vector) representing real world entities.

More precisely, a typical pattern recognition procedure can be illustrated as in Figure 2-1. It is constituted mainly from four distinct but highly related operating stages as represented by the ovals in this figure: the sensing, the feature extraction, the classification and the system evaluation. At the sensing stages the real world entities are captured and converted into digital data set by a sensor to obtain their measured repre-

[1] It is worthwhile clarifying that the term "category" or "class" is a human mental creation meant to reflect some structural nature that exists in a massive collection of real world entities.

sentation. This is usually followed by a feature extraction operation to reduce the amount of data and make them more suitable for an efficient classification. Then, the classification operation takes place in order to attribute the real world entities the membership of one or another class according to the generated feature vector or some transformation of them, that is, to stick a class (category) label onto the real world entities. The system evaluation stage provides the tools allowing the examination of the classification performance.

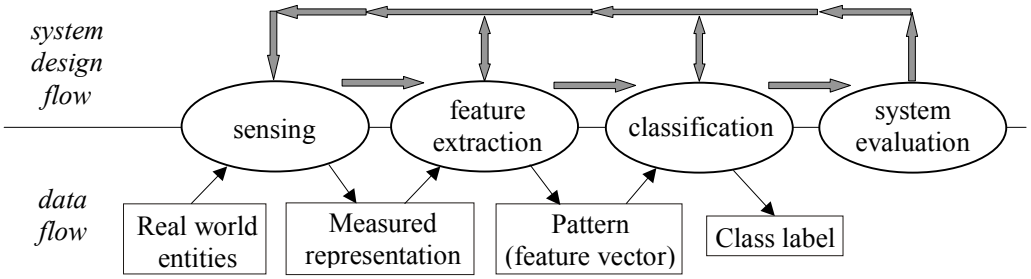


Figure 2-1: Pattern recognition system

When designing such a pattern recognition system we must answer the following questions:

- How are the features generated and what is the appropriate size of a feature vector? This is an application dependent problem.
- How does one design a classifier or an algorithm that does the classification job optimally and efficiently?
- How can the classification error be adequately measured and evaluated.

The sensor is the front-end of the system. Its design, however, is not considered in the pattern recognition system design. We need only to select the appropriate sensing device with correct parameter set.

The upper-half part of Figure 2-1 shows such a system design flow. As stressed by the feedback arrows, all design tasks are highly interdependent.

In this discipline people usually distinguishes two types of classification systems: *supervised systems* and *unsupervised systems.*

Supervised classification

A supervised classification system is based on known classes. For example, in a medical image when the classifier tries to decide whether a

special region is a benign tumour or a cancer, the classes "benign tumour" and "cancer" exist *a priori*.

These *a priori* classes are often represented by a database, called *training database*, in which numerous example patterns, called *training patterns*, are collected to represent the various classes. A training database is the machine "experience". A supervised classification system design is guided by, and exploits, the *a priori* knowledge about the different classes, contained in training database.

Let us consider a generic example for a 2-class (class A and class B), 2-variate (generic feature vector $\boldsymbol{x} = [x_j]_{j=1}^{2}$) case to show the basic aspects of the problem. In the training data base we have two training pattern sets representing the two classes: $\mathscr{A} = \{\boldsymbol{a}_k\}_{k=1}^{K_A}$ for class A and $\mathscr{B} = \{\boldsymbol{b}_k\}_{k=1}^{K_B}$ for class B, where $\boldsymbol{a}_k = [a_{kj}]_{j=1}^{2}$ and $\boldsymbol{b}_k = [b_{kj}]_{j=1}^{2}$ are elements of the sets, K_A and K_B their sizes.

To be more concrete, we can associate this example with the above tumour identification example: class A stands for "benign tumour", class B stands for "cancer", x_1 stands for, e.g., the mean intensity value of the image region of interest and x_2 stands for, e.g., the intensity variance of this region.

When the plan (x_1, x_2) (*feature vector space*) is filled with training pattern sets, a situation similar to that shown in Figure 2-2 (a) may result. Our concern now is to class an incoming pattern $\boldsymbol{f} = [f_j]_{j=1}^{2}$, called *test pattern*, in the *best-fit* or *most likely* class among two possibilities. This is the task of the classifier. Designing a classifier consists of separating the feature vector space into two subspaces R_A and R_B so that if $\boldsymbol{f}$ falls into R_A it is considered belonging to class A, otherwise it is considered belonging to class B.

In Figure 2-2 (a) we have placed two linear classifiers. With the help of the training database, it becomes evident that the classifier II being better than classifier I in terms of classification error probability. With the help of the training database one can also assess the efficiency of the selected feature: a well-selected feature enhances the class separability and therefore facilitates the classifier design.

In the so-called Bayesian classification frame, the exploitation of the class *a priori* knowledge consists in estimating the probabilistic properties of class as will be explained later on in some more detail.

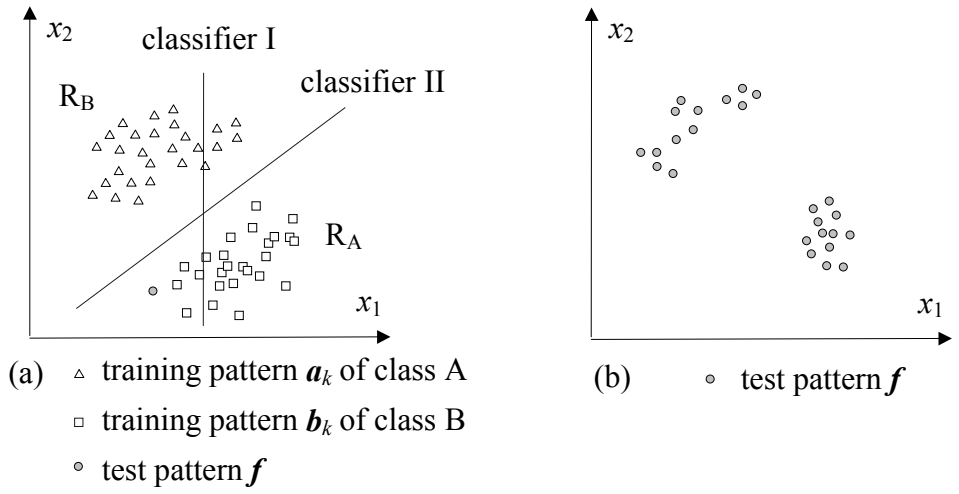


Figure 2-2: (a) supervised classification; (b) unsupervesed classification (clustering).

Clearly, the dimension of the feature vector and the number of classes are not limited to 2. When they become larger the design complexity increases accordingly.

Unsupervised classification (clustering)

An unsupervised classification system addresses the problems where the training database is not available. Its task is basically to form the classes. An example situation is illustrated in Figure 2-2 (b) where a set of test patterns $\mathscr{F} = \{\boldsymbol{f}_k\}$ is given with the goal to find the underlying similarity structures in $\mathscr{F}$ and to group the vectors accordingly.

In such a system, the feature selection plays an even more important role than it does in a supervised system. It will not only influence the class separability but also change the class forming process [Theo99].

In general, an unsupervised classification is much more complex than a supervised one.

2.2. Template matching and Bayes classification rule

Template matching, the technique employed in the present work, is another type of classification task where we assume a set of reference patterns (templates) being available and we wish to decide which one of these reference patterns best matches a test pattern. Furthermore, such a

template matching problem is often associated with a searching scan process because of the feature vector size difference between test pattern and template.

Two different dimensionality appear in a template matching problem: the feature vector dimensionality (feature vector size) and searching scan dimensionality (number of scan direction). In this thesis, we will use the suffix "-variate" to denote the former and the suffix "-dimensional" to denote the latter.

Template matching problem arises primarily in scene analyse where we wish to decide existence and find the location of a smaller image, representing a particular object, within a larger image, representing a scene. However, the basic ideas of template matching have been extended into enlarged scope of applications, such as robot vision, motion estimation for video coding, speech recognition and now automated audience measurement. A statistical interpretation of template matching should in general exist that can provide a valuable guideline and deeper insight of the problem.

2.2.1. Template matching

Let us consider a one-dimensional template matching problem. Given a reference pattern (template) set of M patterns $\mathscr{R} = \{\boldsymbol{r}_i\}_{i=1}^{M}$ where each template is a N-feature vector $\boldsymbol{r}_i = [r_{ij}]_{j=1}^{N}$, and an incoming K-feature test pattern $\boldsymbol{f} = [f_j]_{j=1}^{K}$ with $K \geq N$ (or $N \geq K$), the template matching task consists in comparing each $\boldsymbol{r}_i$ with $\boldsymbol{f}$, by means of searching scan in the range of $K - N$ (or $N - K$) (Figure 2-3), to determine which template at which scan position matches the test pattern best in the sense of minimising an appropriate *matching error* (distance) measure $d(i,t)$.

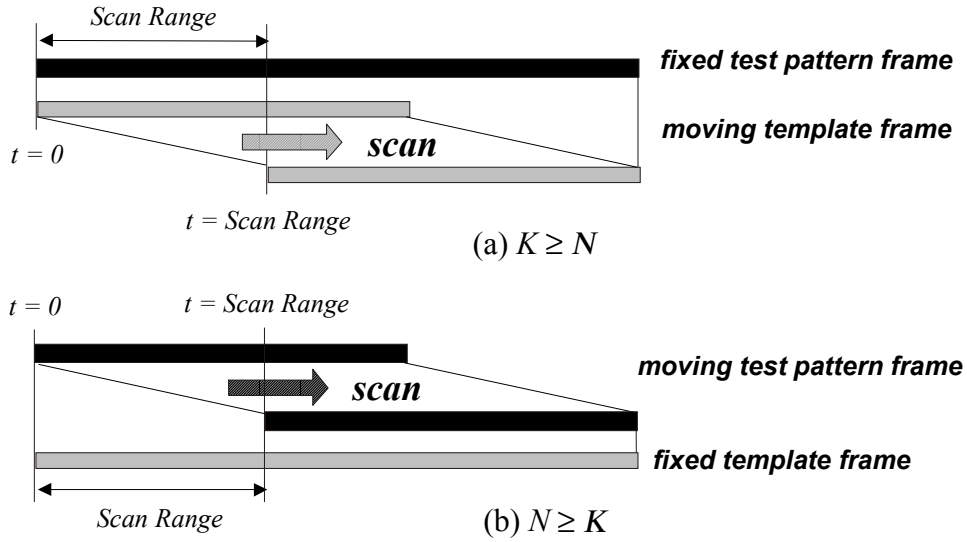


Figure 2-3: Searching scan: (a) find template in test pattern; (b) find test pattern in template.

In the present application the pattern vector index j of $\boldsymbol{r}_i$ and $\boldsymbol{f}$, as well as the scan position t have the same physical meaning: the time.

Let us focus our intention now on the metric that provides the appropriate distance measure $d(i,t)$. For this purpose, we assume $K = N$ ($t \equiv 0$ then), that is, we ignore for the moment the searching scan aspect. Generally, $\boldsymbol{f}$ is a more or less noisy image potentially containing $\boldsymbol{r}_i$. This situation can be modeled as

$$\boldsymbol{f} = \boldsymbol{r}_i + \boldsymbol{v} \tag{2.1}$$

where $\boldsymbol{v}$ represents the noise carrying all random characteristics of $\boldsymbol{f}$ and the term $\boldsymbol{r}_i$ should be considered as deterministic.

Whether the distance function d is adequate or not depends on the distribution of $\boldsymbol{v}$. To illustrate the relation (2.1), let us consider a large collection of test patterns potentially covering the set $\mathscr{R}$. When placing these test patterns and templates into the feature vector space, with $K = N = 2$, $t \equiv 0$, we can expect to observe a situation similar to Figure 2-4 where the test patterns are clustered around their templates due to (2.1).

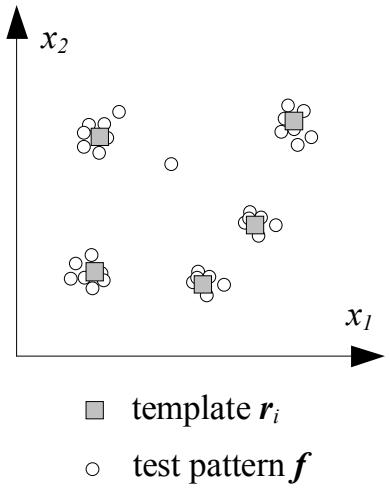


Figure 2-4: Template matching viewed as "labeled clustering"

It can be readily seen that if the term $\boldsymbol{v}$ in (2.1) has zero mean then taking expected value will lead to $E\{\boldsymbol{f}\} = \boldsymbol{r}_i$. In other words, a template matching task can be viewed as a simplified clustering problem with known class mean[2], the template, provided that the noise $\boldsymbol{v}$ has zero mean.

How can *d* be determined when the distribution of $\boldsymbol{v}$ is available? This problem is solved by the well-known Bayes decision theory [Theo99] [Duda73].

2.2.2. Fitting template matching to Bayes decision frame

Due to the random variation of pattern caused by all kind of noises, a pattern recognition system is often built upon probabilistic arguments.

Given a classification task of M classes $\{c_i\}_{i=1}^{M}$, and an incoming test pattern $\boldsymbol{f} = [f_j]_{j=1}^{K}$, the *a posteriori probabilities* $P(c_i|\boldsymbol{f})$ represents the probability that $\boldsymbol{f}$ belongs to class c_i. The Bayes classification rule can be stated as: $\boldsymbol{f}$ belongs to a particular c_n if

$$P(c_n|\boldsymbol{f}) > P(c_i|\boldsymbol{f}) \quad \text{for all} \quad i \neq n \tag{2.2}$$

[Theo99]. This is a natural decision. It offers a precise formulation of what is meant by "most likely belong to …" when we classify the in-

[2] Thus an unsupervised task converges towards a supervised one.

coming test pattern. The decision criterion (2.2) is also called maximum *a posteriori* probability (MAP) criterion in the general decision theory.

The *a posteriori* probability can be computed according to the well known Bayes conditional probability relation

$$P(c_i|\boldsymbol{f}) = \frac{p(\boldsymbol{f}|c_i)P(c_i)}{p(\boldsymbol{f})} \tag{2.3}$$

where

- $p(\boldsymbol{f}|c_i)$ is the likelihood function of c_i when viewed as function of c_i, or the conditional joint PDF (Probability Density Function) of $\boldsymbol{f}$ for class c_i when viewed as function of $\boldsymbol{f}$;
- $P(c_i)$ is the *a priori* probability of the class c_i representing the probability of occurrence of class c_i therefore satisfying $\sum_{i=1}^{M} P(c_i) = 1$;
- $p(\boldsymbol{f}) = \sum_{i=1}^{M} p(\boldsymbol{f}|c_i)P(c_i)$ is the total PDF of $\boldsymbol{f}$.

Let us examine more closely the relation (2.3). The total PDF $p(\boldsymbol{f})$ does not contribute to the decision because it is the same for all classes. The *a priori* probability $P(c_i)$ plays a role of weighting the likelihood function $p(\boldsymbol{f}|c_i)$ of each class according to their occurrence frequency. Quite often, we have to assume that the *a priori* probabilities are equal for all classes because of lack of knowledge on them. In this case $P(c_i)$ can also be eliminated from decision formulation and the condition (2.2) can be equivalently stated as maximum likelihood condition

$$p(\boldsymbol{f}|c_n) > p(\boldsymbol{f}|c_i) \quad \text{for all } i \neq n \tag{2.4}$$

In many cases it is more practical to use a logarithmic version of (2.4) for exponential type PDF like Gaussian and Laplacian.

$$ln(p(\boldsymbol{f}|c_n)) > ln(p(\boldsymbol{f}|c_i)) \quad \text{for all } i \neq n \tag{2.5}$$

The equivalence of (2.4) and (2.5) is guaranteed by the monotonicity of logarithmic function.

The relations (2.2), (2.4) or (2.5) can be considered as the basis of the pattern recognition theory although many alternative approaches leave this Bayesian frame due to difficulty of PDF estimation. It can be

shown that the Bayesian classification is optimal in the sense of minimising the classification error probability. The other approaches are, in general, suboptimal comparing with the Bayesian one [Theo99].

The estimation of the joint conditional density is difficult, notably when the dimension of $\boldsymbol{f}$ is large. Fortunately, in cases where the individual features in $\boldsymbol{f}$ are mutually independent, (2.5) can be greatly simplified, resulting in maximisation of

$$\sum_{j=1}^{K} ln\left(p\left(f_j|c_i\right)\right) \tag{2.6}$$

where $p\left(f_j|c_i\right)$, called marginal PDF is the conditional PDF of the individual feature. This is so because in this case we have [Papo91]

$$p(\boldsymbol{f}|c_i) \overset{\text{feature independence}}{=} \prod_{j=1}^{K} p\left(f_j|c_i\right). \tag{2.7}$$

Therefore, if feature independence is valid, the joint density estimation becomes a marginal density estimation. And if the result of this estimation well fits certain simple PDF model, like Gaussian or Laplacian, we can state Bayes classification rule using some efficient and simple distance measures. To be more convincing, let us apply the Gaussian PDF

$$p(f_j|c_i) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\frac{-\left(f_j-\eta_{i,j}\right)^2}{2\sigma^2} \tag{2.8}$$

and the Laplacian PDF [Sayo96]

$$p(f_j|c_i) = \frac{1}{\sqrt{2\sigma^2}} exp\frac{-\sqrt{2}\left|f_j-\eta_{i,j}\right|}{\sigma} \tag{2.9}$$

to $p\left(f_j|c_i\right)$ in (2.6) assuming same variance σ^2 for all classes. This will lead to minimisation of squared *Euclidean norm*

$$d_{euc}^2 = \sum_{j=1}^{K} \left(f_j - \eta_{i,j} \right)^2 \tag{2.10}$$

and *Manhattan norm*

$$d_{man} = \sum_{j=1}^{K} \left| f_j - \eta_{i,j} \right| \tag{2.11}$$

respectively. It becomes evident now that in order to fit template matching into the Bayes decision frame it suffices to replace the class mean η in (2.10) and (2.11) by the template *r*.

2.2.3. Robust estimation

The Bayesian classification approach we used here is essentially an error minimization problem. From this point of view it shares the same mathematical framework with the regression analysis.

Using appropriate distance measure according to the error distribution is the basic idea of the *robust estimation* issued from regression analysis. Assigning the term robust to the approach, presented in this thesis, stresses the use of Laplacian PDF model that fits best the noise distribution (see Section 3.6 for empirical proof).

Detailed description about robust estimation can be found in [Drap98] [Sebe77] [Sebe89].

2.2.4. Practical formulation of the problem

In practice a template matching problem is often viewed from a slightly different angle. Instead of comparing the test pattern with all possible template patterns one should rather decide if the test pattern contains a specific template pattern. This is even a must in an audience measurement problem since it is impossible to obtain all possible templates: the set $\mathscr{R}$ is an opening set with unbounded number of templates M.

In this case we must compute the matching with one template, producing a distance value (or similarity value). We must then make the "accept" or "reject" decision according to the comparison of the computed

distance (or similarity) with a threshold value. This threshold value in turn should be determined using a training database. Thus, we decompose the original template matching problem into two classification problems (Figure 2-5):

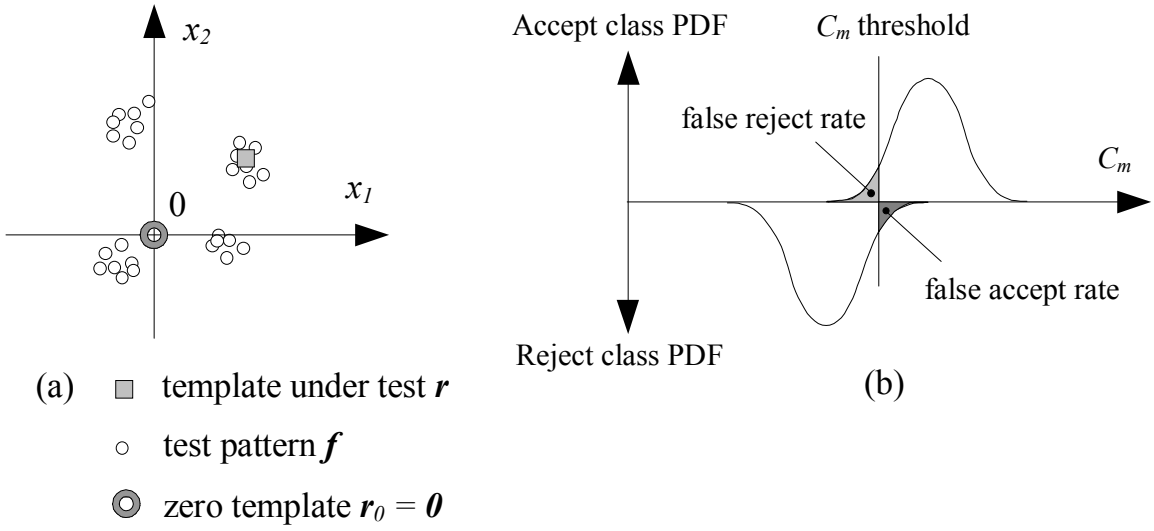


Figure 2-5: Practical view of template matching: (a) "labelled clustering" step, (b) similarity threshold determination step

Problem 1. A 1-dimensional multi-variate, 2-class template matching ("labeled clustering") problem (Figure 2-5 (a)). One of the two classes stands for the sound we wish to decide its presence in the incoming test pattern, let us call it the *class under test* and its representative the *template under test*. The other class stands for a mixture of other possible sounds (noise natured class), let us call it *zero class* and its representative *zero template* (with all features at the origin). The feature vector is the same as the original one. This problem will be discussed in detail in Chapter 5.

Problem 2. A 1-variate, 2-class general supervised classification problem (Figure 2-5 (b)). The unique feature is the distance (or similarity C_m) produced in Problem 1. One of the two classes is the class "accept", the other "reject". A "training pattern" in the training database is formed by a match result between a training-test-pattern and a training-template. This problem will be resolved in Chapter 6.

These two problems are closely related. For problem 2, problem 1 is a continuation of feature extraction (or the computation of test statistic);

for problem 1, problem 2 is a procedure producing the template matching threshold value.

This practical vision of template matching does not change what we have discussed previously about the distance measure.

It is worthwhile pointing out that such a template matching task is essentially a supervised classification task.

2.3. Noise

Noise is a delicate term in our application and its meaning must therefore be clarified.

In the context of audience measurement applications any structured sound[3] has to be considered under two aspects. A structured sound is considered to be a signal when it should be recognised. However, if a structured sound S_1 is mixed with another structured sound S_2 while S_1 should be recognised, the sound S_2 is considered as a noise, called structured and coloured noise. This type of noise may include the sound directly generated from a source or from some parasite phenomena such as frequency domain aliasing and imaging, harmonic distortion, etc. While a structured noise is harmful to the classification task, its elimination from the signal is very difficult once it is added to the signal. The system robustness against this kind of noise has to be covered by the adequate similarity measure as will be explained in Chapter 5.

There exist other types of acoustic disturbances that do not possess any remarkable structure, for example, white noise, quantization error or other signal distortion of this kind. Their cancellation is relatively easy. To do so, different techniques are used as will be explained in Chapter 3 and Chapter 5.

2.4. Summary

In this chapter we have explained the basic pattern recognition frame and the importance of its three main constituents: feature extraction, classification and performance evaluation.

---

[3] By structured sound we mean an acoustic signal whose short time average energy varies with time and along frequency axis. In other words, a structured sound possesses clustered energy concentration distributed along time and frequency axes. See Figure 3-1 for examples.

Using well-known Bayes classification rule, we have demonstrated that the usual distance measures, Euclidean norm and Manhattan norm, have their probabilistic backgrounds. Their optimal use depends on the noise distribution characteristic, which constitutes the basic meaning of the term “robust”.

We have equally given the practical definition of the problem as two distinct and tightly related tasks: a 1-dimensional, multi-variate, 2-class template matching and an 1-variate, 2-class supervised classification.

All the above points constitute the main axes in the sequel development of this thesis.

As a particular consideration we have also clarified the term “noise” in order to help the reader to always retain a clear notion of it.

Chapter 3
Feature extraction

This chapter will show the conception of feature generation and selection versus application constraints. It will give a detailed description of involved DSP tools, discuss how the white type noise, echo and small time shift disturbances can be cancelled or attenuated at this stage. Finally, it will give the noise distribution estimate using the non-parametric method applied to generated features.

3.1. Introduction

The basic reason of performing feature extraction in the context of classification is to remove information redundancies that usually exist in the measured samples due to the general purpose sensing devices. This information redundancy removal is often achieved by exploiting the intrinsic structures that exist in measured data and results in a few of discernible quantities after important data reduction. The direct benefice is reduced computational complexity. A secondary benefice of having a limited number of features is that for a given number of training patterns the classification error estimation can be improved when the feature vector dimension decreases [Theo99]. The third benefice, not directly related to the pattern matching system performance, but important for the audience measurement application, is facilitated data storage and transfer.

Another important reason for feature extraction is the removal of correlation between the neighbour samples of the measured signal. This correlation, allowing signals to be distinguished from white noise, is is-

sued from intrinsic morphological consistencies (in an image) or physical conditions (sound wave) of signal source. This correlation gives rise to higher energy levels at low frequency area and decayed energy level when frequency increases. In a classification system a completely decorrelated feature set is generally desired to make statistic modelling and inference easy therefore leading to a simplified matching process.

Let us now make some observations on the acoustical signal examples whose spectrograms are portrayed in Figure 3-1.

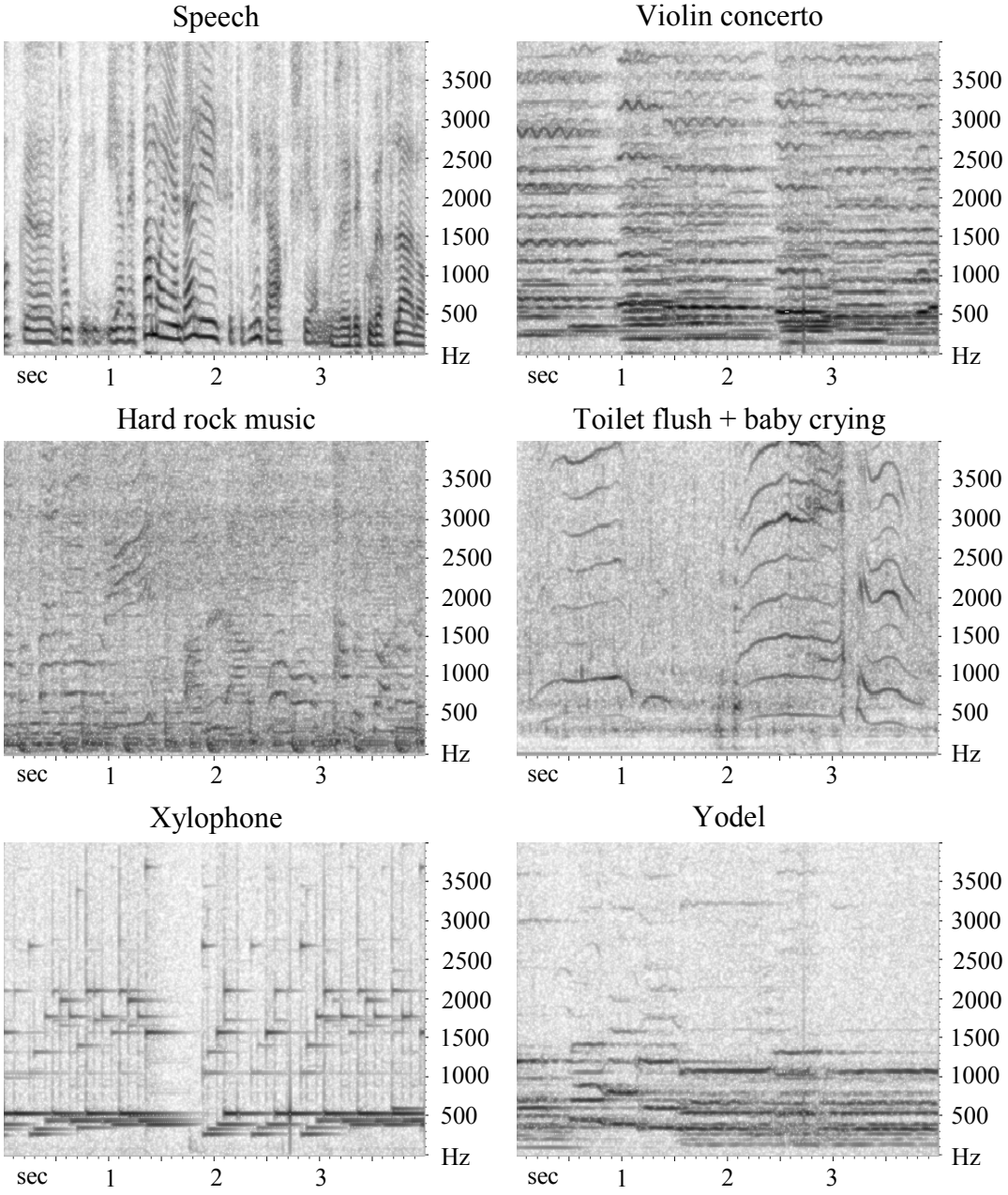


Figure 3-1: Spectrogram of acoustic signal. High energy locates at darker region

In Figure 3-1 we immediately remark that the sound energy appears in a clustered form (darker region) for all examples and different sound types possess different forms of energy clusters spreading randomly along the frequency axe and the time axe in different manners. In the violin concerto example we see long retained notes with beautiful vibrato; in the baby crying example we see the powerful high frequency formant just like a great lyric singer; the nail-like energy clusters in the xylophone example; etc. In a word, a sound signal possesses a sub-band time structure that makes itself distinct from other sound signals. This observation has inspired the basic idea of the proposed feature extraction scheme as illustrated in Figure 3-2. Notice that the random character of the energy cluster distribution will be exploited to decorrelate the signal.

Further analysis of Figure 3-1 tells us that the sound energy is mainly distributed inside the lower frequency range due to sample correlation as we have mentioned above. The most informative area is ranged commonly below 1500 Hz and above 200 Hz. This motivates our choice of 3 kHz sampling frequency at the sensing stage.

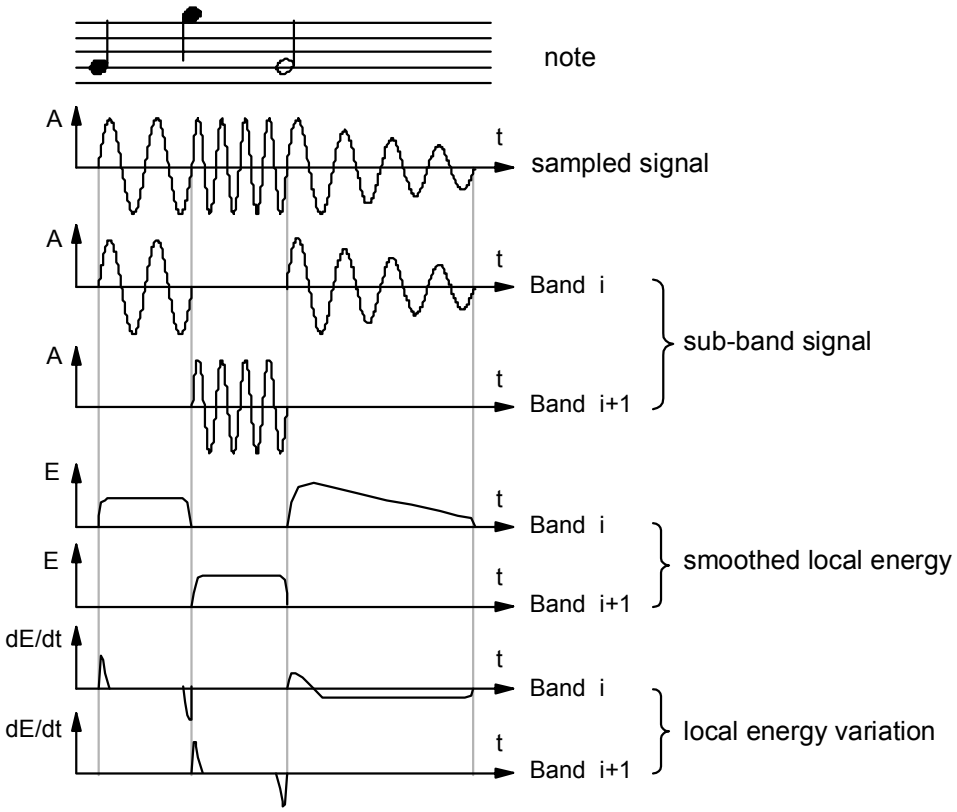


Figure 3-2: Basic idea in feature generation (synthetic signal)

As can be seen in Figure 3-2 after the signal sensing stage the algorithm splits the sampled signal into sub-band signals by wavelet transform. After appropriate band selection the algorithm extracts each remaining sub-band signal into smoothed energy[1] and takes its time derivative to achieve small time shift invariance, echo cancellation and white noise removal. As a result, a highly decorrelated, compact representation (feature vector) is obtained. The following section will give detailed description of all these procedures.

3.2. Sub-band decomposition using wavelet transform

As explained in the previous section the proposed algorithm has been designed to exploit the sub-band time domain structure of the signal. This requires the signal to be transformed into time-frequency representation. Two widely used tools can accomplish this task: the Short Time Fourier Transform (STFT) and the Discrete Time Wavelet Transform (DTWT). Compared to STFT, the most important advantages of DTWT are, in order of relevance to our application, its implementation simplicity, the facility of realising non-uniform bandwidth decomposition and the possibility of realising perfect reconstruction systems.

3.2.1. DTWT and filter bank

In practice the most convenient way to implement discrete time wavelet transforms is using a dyadic tree structured filter bank as illustrated in Figure 3-3, an example for the case of 2-level 4 equal bandwidth channels (uniform) analysis bank.

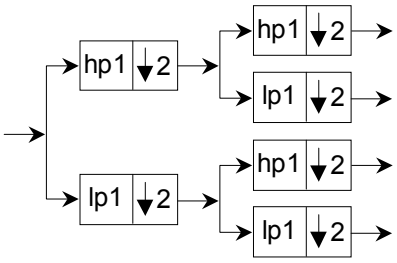


Figure 3-3: Two level dyadic tree structure of a uniform filter bank

At each level an input signal is split into two output channel signals by

[1] Strictly speaking, it is the power.

high-pass and low-pass 2-fold decimations respectively. Using wavelet language we say that the input signal space is separated into two subspaces at the output. The high-pass subspace is spanned by a basis composed of wavelets and the low-pass subspace is spanned by a basis composed of scaling functions. The elements of the bases in the subspace are two times finer in frequency and thus two times rougher in time than those of the basis that spans the original signal space. The final channel signals are merely the expansion coefficients of these bases. The equivalence between DTWT and such a filter bank can be more explicitly understood as: for a given dyadic tree structure and appropriate decimation and interpolation filter design there is always a corresponding wavelet-scaling-function set that can be constructed using more or less complex procedures.

If the filters $lp1$, $hp1$ and their corresponding synthesis filters satisfy the so-called *paraunitary* conditions, leading to a perfect signal reconstruction, the filter bank then corresponds to an orthonormal wavelet set whose construction is probably the easiest. An orthonormal wavelet system performs so-called lossless transform so that Parseval's theorem applies, that is, the sum of the channel signal energy, represented by the sum of squares of the expansion coefficients, equals exactly the original signal energy. In this case the filters $lp1$ and $hp1$ are *power complementary*, that is, their power spectra sum 1 everywhere along frequency axe [Vaid93] [Flie94]. Unfortunately, a paraunitary system has a complex design procedure and an important run time complexity. This class of filter banks is mentioned here in the hope that it can give a qualitative signal decomposition performance reference. Any non alias-free practical filter bank will tend to increase the channel signal energy and thus enlarge class variance range. However, comparing to other coloured and structured disturbances (environmental), this aliasing effect does not constitute any major concern for us. As explained in Chapter 5, the matching algorithm is designed to be robust against coloured and structured noise.

3.2.2. Decimation filter design

Primarily constrained by design and implementation simplicity consideration, our choice for the low-pass decimation filter $lp1$ is the widely used linear phase FIR *half-band filter*.

The main characteristics of such a filter, for an 11-tap case, are illustrated in Figure 3-4 in its zero-phase non-causal prototype form to fa-

cilitate certain important symmetry observations. The imaginary part of its frequency response is null in this form.

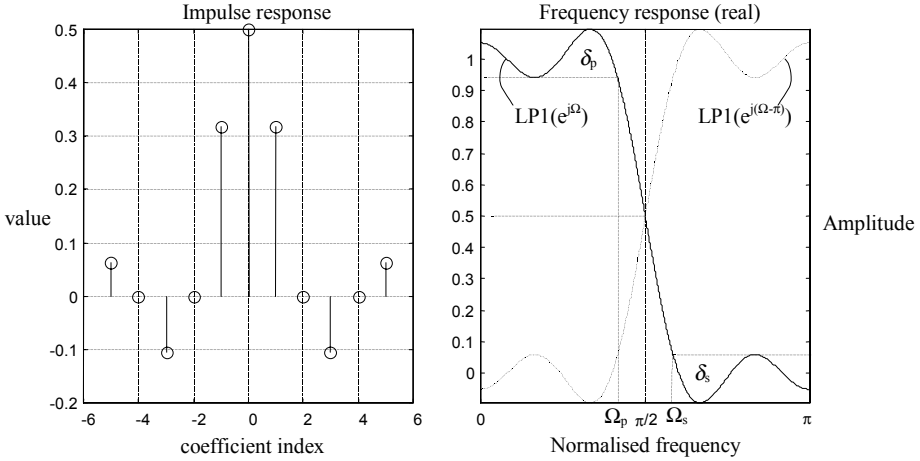


Figure 3-4: Half-band low-pass filter (Zero-phase)

The frequency response $LP1$ of such a half-band low-pass filter $lp1$ has shift symmetry in the frequency domain about the point $\Omega = \pi / 2$, that is

$$LP1\left(e^{j\frac{\pi}{2}} \right) = 0.5 \quad \text{and} \quad LP1\left(e^{j\Omega}\right) + LP1\left(e^{j(\Omega-\pi)}\right) = 1 \tag{3.1}$$

where Ω is the normalised frequency. Its pass-band and stop-band cut-off frequencies, Ω_p and Ω_s, are symmetrically positioned about the shift symmetry point $\Omega = \pi / 2$

$$\frac{\pi}{2} - \Omega_p = \Omega_s - \frac{\pi}{2} \tag{3.2}$$

and corresponding ripples, δ_p and δ_s in the pass-band and stop-band, being equal

$$\delta_p = \delta_s \tag{3.3}$$

In time domain, such filter $lp1$ generally has an odd number of coefficients $N = 4i - 1$ with i a positive integer. Its impulse response is symmetric about the central coefficient. Further more, due to (3.1), the even indexed coefficients of $lp1$ are zero except the central one being 0.5 [Flie94] [Vaid93].

$$lp1(n) = \begin{cases} 0.5 & \text{if } n = 0 \\ 0 & \text{if } n = \pm 2, \pm 4, \cdots \end{cases} \tag{3.4}$$

If the property (3.1) is exploited to define the high-pass pair $hp1$ of $lp1$ then the frequency response $HP1$ of $hp1$ is given by

$$HP1\left(e^{j\Omega}\right) = LP1\left(e^{j(\Omega-\pi)}\right) \text{ and therefore } LP1\left(e^{j\Omega}\right) + HP1\left(e^{j\Omega}\right) = 1 \tag{3.5}$$

that is, $lp1$ and $hp1$ form a complementary pair (but not power complementary). From (3.5) the time domain relation between $hp1$ and $lp1$ is given by the inverse Fourier transform of (3.5)

$$hp1(n) = d(n) - lp1(n) \tag{3.6}$$

where $d(n)$ is the unit impulse function.

Note that the above discussions about $lp1$ and $hp1$ assume the zero-phase condition. However, the causal version of $lp1$ and $hp1$ can be obtained from their zero-phase version after a time-shift of $(N-1)/2$ to the right.

The relation (3.4) and (3.6) constitute the most remarkable attraction of the half-band filter. The former allows halving the computational complexity while the latter replaces $hp1$ filtering by subtraction of the $lp1$ filtered signal from the original one.

Among the number of available design methods the simplest is windowed SINC function. With this method the filter quality is mainly influenced by the type of window and the number of coefficients. Among the number of available windows the simplest one is rectangular (also called boxcar). The two most relevant characteristics of boxcar window are the highest (–13 dB) side-lobe and the narrowest main-lobe. Due to these characteristics, a boxcar-windowed SINC function has poorest –20 dB ripple level, corresponding to a signal 10 times higher than eventual

aliasing, but sharpest transition band. Further adjustment of transition bandwidth should be done by adjusting filter length. Clearly, we have to trade-off between the computational complexity and the sharpness of transition band. A 19-tap seems to be reasonable. The Figure 3-5 shows the retained solution.

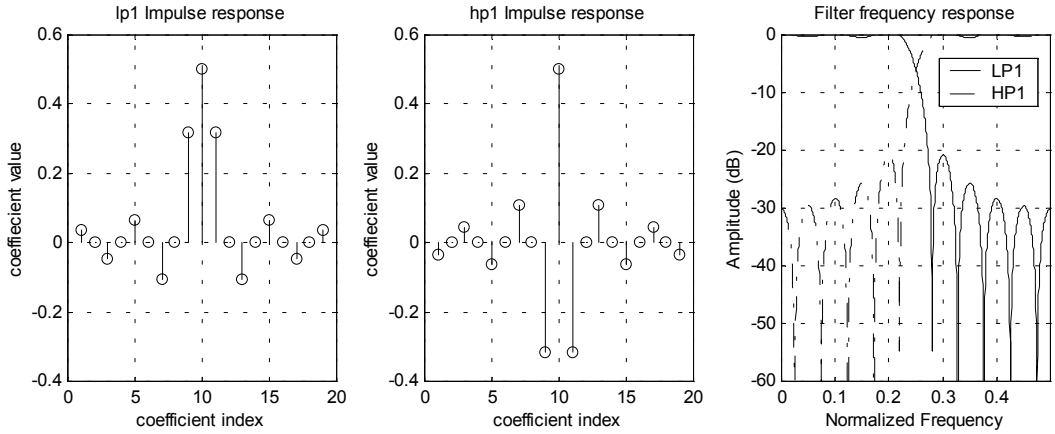


Figure 3-5: Decimation filters: 19-tap boxcar-windowed SINC half-band filter pair (causal version)

The coefficients of this half-band filter pair are

i	*lp*1 (i)	*hp*1 (i)
1 or 19	0.03536776513153	-0.03536776513153
2 or 18	0	0
3 or 17	-0.04547284088340	0.04547284088340
4 or 16	0	0
5 or 15	0.06366197723676	-0.06366197723676
6 or 14	0	0
7 or 13	-0.10610329539460	0.10610329539460
8 or 12	0	0
9 or 11	0.31830988618379	-0.31830988618379
10	0.50000000000000	0.50000000000000

It is noteworthy that the linear-phase property of *lp*1 and *hp*1 is not crucial condition for the present application since the algorithm is designed to be invariant to time shifts. However, we don't need to remove a general desirable property that cost nothing to obtain.

When using a 19-tap boxcar-windowed SINC half-band filter the decimation process potentially gives an amplitude amplification of the samples by a factor of 1.6378 (the sum of the absolute value of the coefficients) at each level of sub-band decomposition. In a fixed-point DSP implementation this factor should be treated carefully to avoid overflow. An overflow affects the algorithm much more strongly than an aliasing does. Possible remedies are downscaling and/or saturating the samples at the adequate places.

The determination of the frequency resolution in this subband decomposition is again a question of trade-off between quality and complexity. Experiments have shown that 8-division of the region below 1500 Hz is a good compromise. This results in a 3-level uniform dyadic tree bank with sub-band bandwidth equal to 187.5 Hz. As we have seen from Figure 3-1, the range below 200 Hz is not very informative. This allows us to reject the two lower bands to save more computation power. The remaining 6 bands, covering a range from 375 Hz to 1500 Hz, is judged sufficient.

Before concluding this section, it is worthwhile to mention a particular characteristic of tree structured filter banks due to the well known phenomenon of high-pass image inversion as shown in Figure 3-6. In this figure, the axis ω is non-normalised frequency; the vertical arrows represent sampling pulses; 0 index corresponds to a base band and the other index numbers correspond to sampling images. The “magic” happens when the high-pass filtered signal is down sampled, because at the next decimation stage it is the shifted sampling image 1 and –1 that are treated as base band.

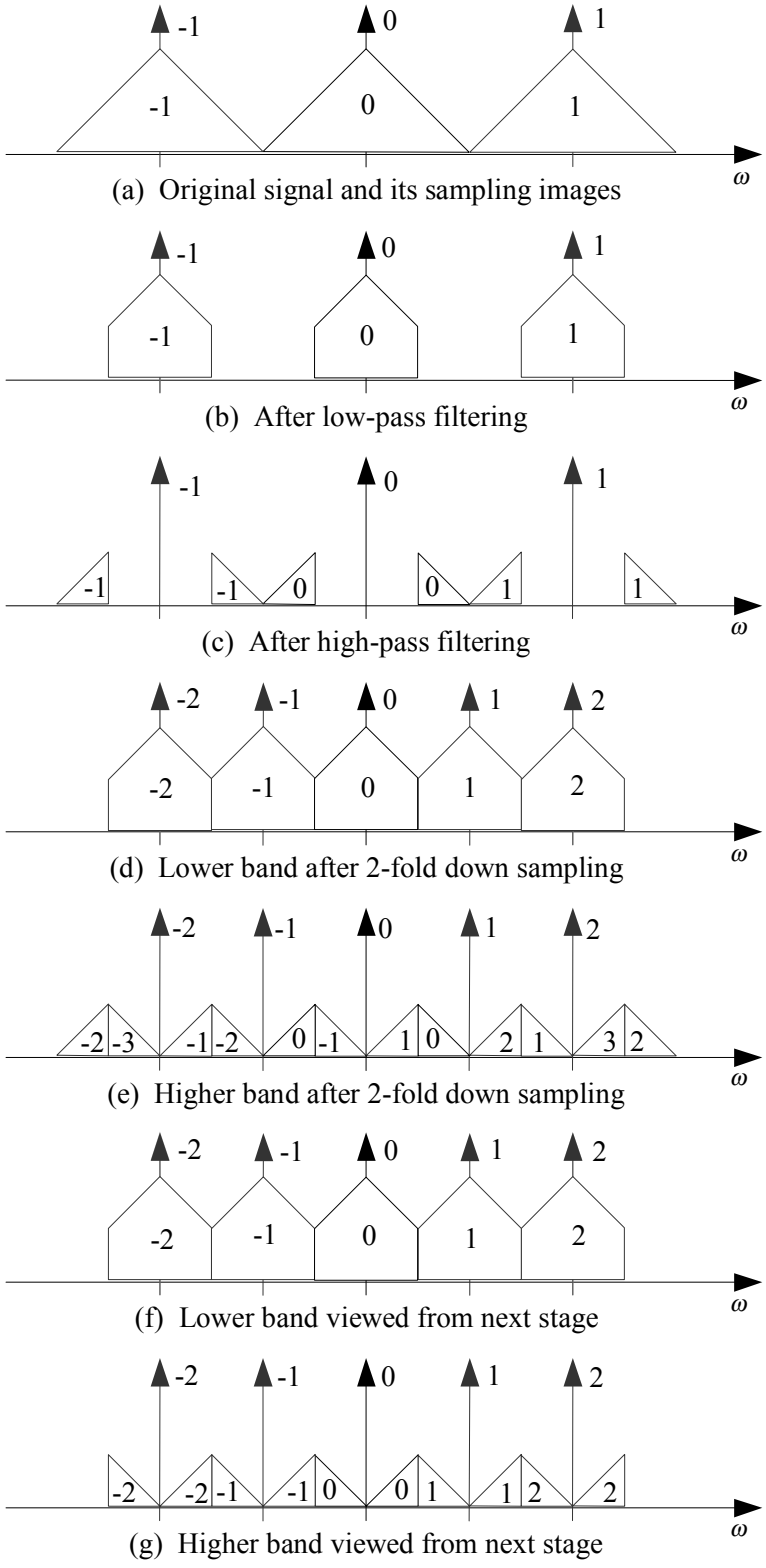


Figure 3-6: Decimation process viewed in the frequency domain

This phenomenon causes the output data to be arranged in a quite unusual manner as shown in Figure 3-7. It is noteworthy that the delay of

filter *lp*1 is ignored in this filter bank structure illustration. Other multi-rate techniques can further reduce the computational complexity and will be developed in Chapter 7.

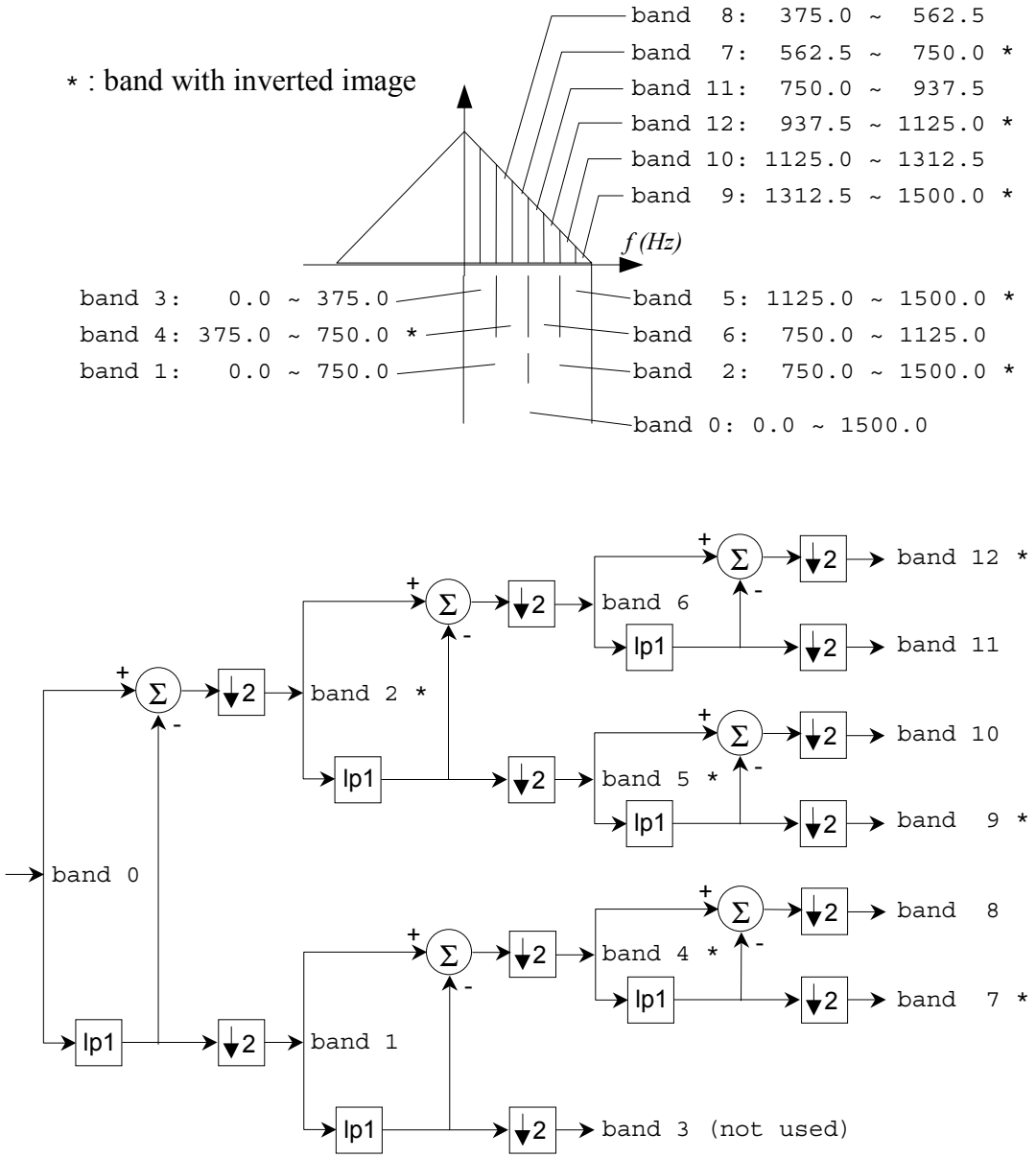


Figure 3-7: Retained sub-band decomposition scheme

The sub-band signal, also called channel signal, has a time resolution of

$$T_{ch} = \frac{M_{sp}}{F_s} = \frac{2^l}{3000\,Hz} \overset{l=3}{=} 2.666\,ms \tag{3.7}$$

where $l = 3$ is the number of the dyadic tree level, $M_{sp} = 2^l = 8$ is the band splitting factor, $F_s = 3000\,Hz$ is the input sampling frequency. The parameter T_{ch} corresponds to the finest searching resolution for the matching process and determines the finest resolvable time structure in the sound signal.

Of course, the sub-band decomposition in itself does not give any data reduction (except the rejection of the two lower sub-bands and filtering side effect[2] loss). The data reduction is mainly achieved in the local energy extraction as explained in the next section.

3.3. Local energy extraction

As outlined in Section 3.1 the basic idea of the proposed feature extraction scheme is to exploit the sub-band (local) clustered energy structure. Our interest is focused on the contour of these energy clusters. To obtain the sub-band energy contour it suffices to compute the local signal energy, then pass it through a low-pass (smoothing) filter and down sample correspondingly. The result will be a more compact signal representation. In this section we will explain how this operation is realised and see that some more profound considerations can be taken into account at the same time.

3.3.1. White noise, echo attenuation and small time shift invariance realisation

In real life environment a lot of sound types have more or less white noise characteristics, such as the noise in a moving car, the noise of a coffee grinder, the background noise produced by the radio receiver due to electronic circuit, wind, … etc. It is a great advantage for a system to be robust against this kind of noises. In this subsection it will be shown that it is possible to accomplish this task at the feature generation stage.

[2] The general relation between the length of the input sequence L_{in} and that of the output sequence L_{out} of a FIR decimator is: $L_{out} = \text{ceil}((L_{in} - w + 1)/M)$, where w is the filter length and M is the decimation factor.

Our intuition tells us that the loudness of a white type noise is constant in time or at least in a short time period. Now, we will make some more formal reasoning to confirm this intuition.

Considering a reference signal $\boldsymbol{x}(t)$ damaged by an additional noise $\boldsymbol{n}(t)$, and resulting in a noisy signal $\boldsymbol{y}(t)$, the following model applies

$$\boldsymbol{y}(t) = \boldsymbol{x}(t) + \boldsymbol{n}(t) \tag{3.8}$$

where we should treat each term as a stochastic process[3], t is time. Bold face character stresses the fact that these quantities are randomly valued. The energy relation of these three processes can be obtained by squaring either side of (3.8) as shown in (3.9).

$$\boldsymbol{y}^2(t) = \boldsymbol{x}^2(t) + \boldsymbol{n}^2(t) + 2 \cdot \boldsymbol{x}(t) \cdot \boldsymbol{n}(t) \tag{3.9}$$

Now, for any given instant t (note that each term becomes a random variable now) we take the expected value and we obtain

$$E\{\boldsymbol{y}^2(t)\} = E\{\boldsymbol{x}^2(t)\} + E\{\boldsymbol{n}^2(t)\} + E\{2\boldsymbol{x}(t)\boldsymbol{n}(t)\} \tag{3.10}$$

where we have applied the linearity of expected value operation. Assuming uncorrelatedness between reference signal and noise, the last term on right-hand side of (3.10), representing their cross-correlation, vanishes. Thus we get

$$E\{\boldsymbol{y}^2(t)\} = E\{\boldsymbol{x}^2(t)\} + E\{\boldsymbol{n}^2(t)\} \tag{3.11}$$

The second term on right-hand side of (3.11) represents the average power of the noise. It can be shown [Papo91] that if the noise is white and is wide sense stationary (WSS) process its average power is independent of time t. Let us denote this constant value by q then (3.11) becomes

[3] A stochastic (or random) process can be defined as a time function whose value at each particular instant is a random variable. It can equivalently defined as an experiment space (ensemble), of which every outcome (element) is a time function. More rigorous definition can be found in different literatures, for example, [Papo91] [McDo95].

$$E\{\boldsymbol{y}^2(t)\} = E\{\boldsymbol{x}^2(t)\} + q \tag{3.12}$$

It is clear that if we take the derivative of (3.12) with respect to time the white noise term will completely disappear. However, the expected value operation $E\{\cdot\}$ requires an important number of realisations of the process. This is not practical. The solution is to replace it by time average assuming ergodicity[4] (covariance-ergodicity to be precise) of the processes $\boldsymbol{x}(t)$ and $\boldsymbol{n}(t)$, that is, we can approximate (3.11) and (3.12) with

$$\frac{1}{T_{av}} \int_{t-T_{av}/2}^{t+T_{av}/2} \boldsymbol{y}^2(r)dr = \frac{1}{T_{av}} \int_{t-T_{av}/2}^{t+T_{av}/2} \boldsymbol{x}^2(r)dr + \frac{1}{T_{av}} \int_{t-T_{av}/2}^{t+T_{av}/2} \boldsymbol{n}^2(r)dr \tag{3.13}$$

$$\frac{1}{T_{av}} \int_{t-T_{av}/2}^{t+T_{av}/2} \boldsymbol{y}^2(r)dr = \frac{1}{T_{av}} \int_{t-T_{av}/2}^{t+T_{av}/2} \boldsymbol{x}^2(r)dr + q \tag{3.14}$$

where the time average is run on a time interval that is T_{av} long and symmetrical to the time instant t.

Although the above discussion is given in an analogue form, it is straightforward to restate this idea in digital form and the conclusion will be the same [Papo91]. Now we should respond to the question: how should we choose the time duration T_{av} on which we calculate the time average of signal energy. We should compromise approximation precision (needing large T_{av} since ergodicity is an asymptotic property) and time structure exploitation of the signal (needing T_{av} not to be too large since time average is equivalent to low-pass filtering). Too large T_{av} tends to destroy the finer time structures of the signal. Other criterion for the determination of T_{av} can now be taken into account.

In the real recording environment, sound signals are often mixed with their echo leading to an important deformation of their time domain structure. The signals in Figure 3-8 show echo examples with 200 and 400 ms decay time[5]. As can be seen in Figure 3-8, echoes of long decay

[4] A stochastic process is ergodic if time averages equal ensemble averages [Mix95]. More rigorous definition and description of ergodicity can be found in [Papo91].

[5] Decay time, called also time constant, is the duration for echo intensity decaying to 1/e = 0.36.

time tend to enlarge energy clusters in time and in the worst case tend to destroy the signal time structure.

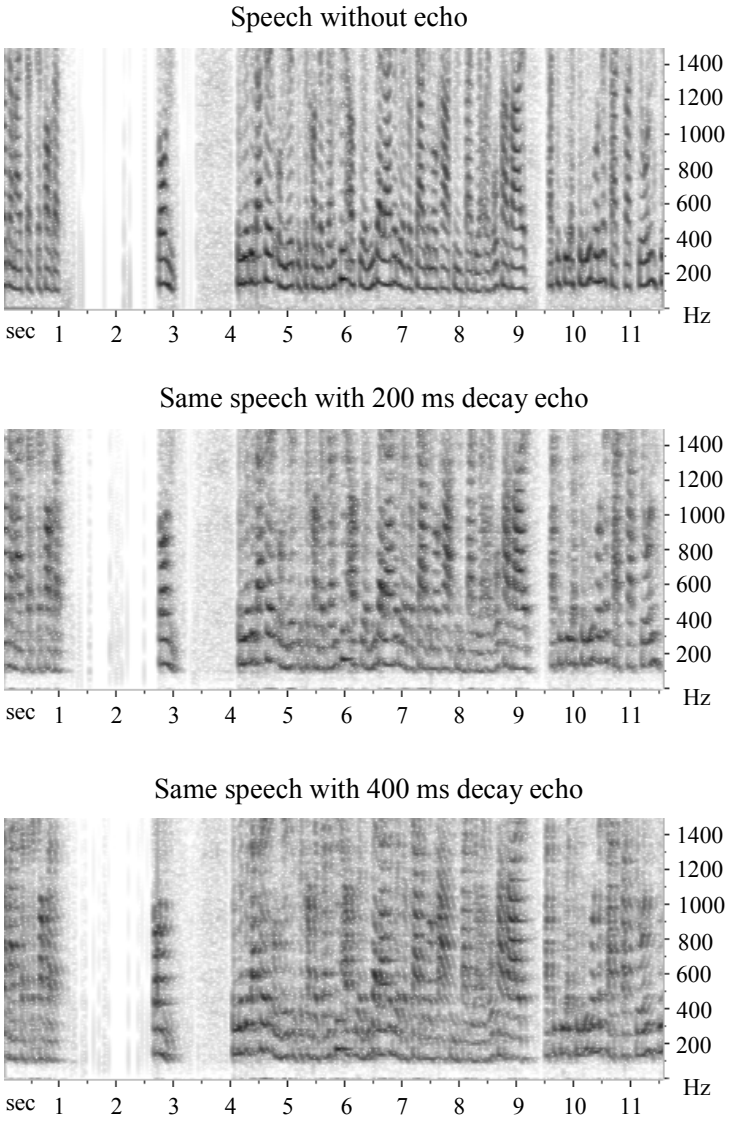


Figure 3-8: Influence of echo

In most of the cases, inside a room or a car for example, the maximum echo decay time is about 100 ms. In order to make the system less sensible to this phenomenon the time resolution should be rougher than 100 ms.

The final choice of T_{av} is 128 ms roughly compromising the above considerations on white noise and echo cancellation and finer time structure exploitation.

As a desirable consequence this time averaging tends to make the resulting data invariant to any random small time shift, ranging within T_{av}, eventually caused by sound delay, imprecise recording time stamp, etc.

3.3.2. Algorithm for energy extraction

Based on the above considerations the following algorithm is applied to each channel signal (Figure 3-9)

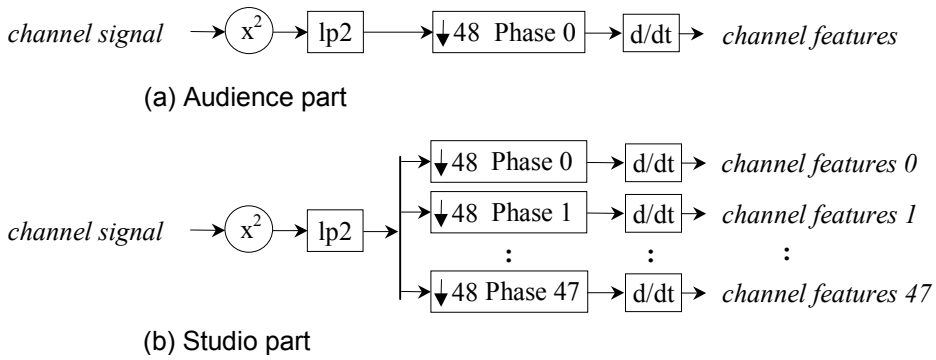


Figure 3-9: Energy extraction algorithm applied on each channel

The channel signal is first squared to convert the amplitude into energy, then low pass filtered by smoothing filter $lp2$ to get energy contours. From here the algorithm for the audience part differs slightly from the algorithm for the studio part. However, the basic subsequent operations are down sampling by a factor of 48 to cancel the echo, then derive with respect to the time to achieve the white noise cancellation. The channel signal energy decimation ($lp2$ followed by down sample by 48) is nothing else than the channel signal energy time averaging (see (3.13) and (3.14)) but stated in DSP language. After this procedure the following time relation is verified

$$T_{av} = T_{ch} \cdot M_{sm} = 2.66 \cdot 48 = 128\, ms \tag{3.15}$$

where $M_{sm} = 48$ is the $lp2$ decimation factor.

As a response to eventual general concern of privacy, **this procedure is not invertible**.

Let us now examine the difference between studio part down sampling and that of audience part. In the studio part this operation is performed for each decimation phase[6], in other words there is no down sampling. Then the subsequent derivative operation is performed for each phase. In the audience part these operations are run only on decimation phase 0 so that the data reduction actually takes place (factor 48). Thus, the matching scan process can run in T_{ch} resolution so that the small time shift invariance can be further enhanced, resulting in a higher matching quality.

We have to mention here that an experimental modification has been made to the algorithm. After the down sampling by 48 the signal can be optionally turned back into the amplitude scale by a square root operation. This will give non-negligible improvement of the matching performance. Chapter 6 will show simulation results about this improvement.

Because of this subsequent square root operation, the coefficients of the energy smoothing filter $lp2$ should be non-negative valued. A possible solution is the use of window functions. The characteristics of three (among others) such windows are shown in Figure 3-10.

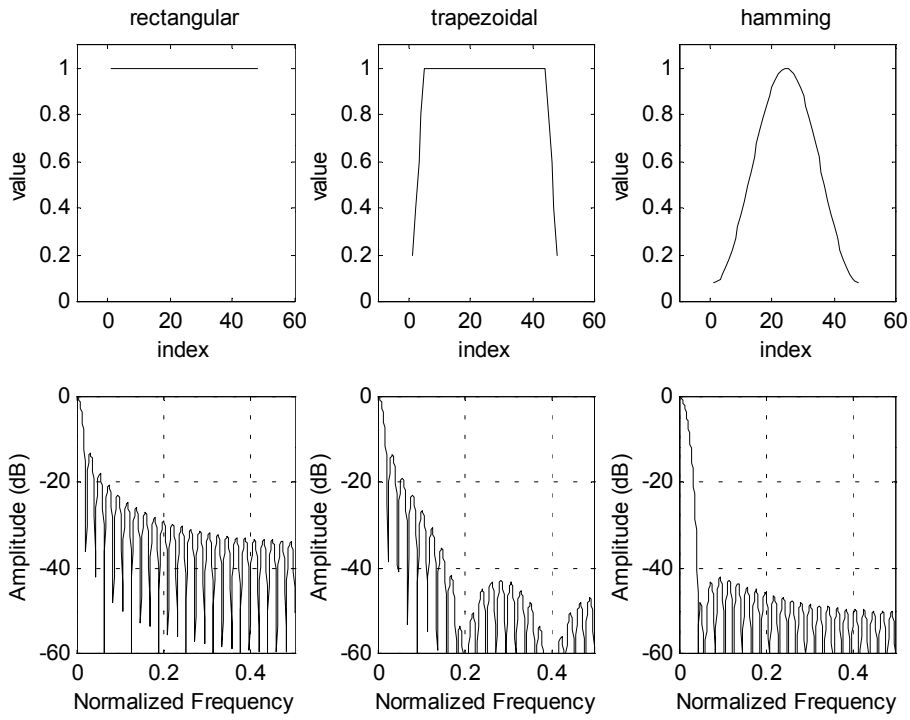


Figure 3-10: some 48-tap window functions

[6] Recall that decimation is a time-varying process [Croc83].

The advantage of using a rectangular window is its computational simplicity: no multiplication is necessary. And because of its very narrow main lobe we can use the M_{sm}-tap window and down sample at M_{sm} without introducing harmful aliasing.

The trapezoidal window has almost the same near pass-band characteristics as those of the rectangular window but higher far stop band attenuation. Thus, M_{sm}-tap window can also be used with M_{sm} down sampling. However, multiplications are necessary.

The hamming window has much better side lobe characteristics but with a wider main lobe. Therefore, the window size should be enlarged to about $2M_{sm}$-tap for down sampling at M_{sm}.

Since the data is presented now in channel signals, the frequency dependent signal power damping, which eventually occurred during signal transmission, can now be easily corrected by a band wise scale retrieval operation in the robust matching process (Chapter 5).

3.4. Feature vector length (recording duration)

The feature extraction design has two tasks: the feature generation and the decision of feature vector length. In the previous section we have discussed in detail how, in the present classification system, the features are generated.

For the determination of the feature vector length our major concerns are the following. First, we should compromise the matching process quality and the computational complexity. Second, we should think about the reliability of the classification error estimation that depends on the ratio of the training database size over the feature vector length [Theo99]. The higher this ratio is, the more reliable the classification error estimation will be.

If the feature generation is performed as explained in the previous sections, the feature vector length is directly proportional to the recording duration.

In Chapter 6 we will re-discuss this issue and give the experimental determination of the recording duration.

3.5. Feature statistics estimation

In Chapter 2, we have pointed out that the classification quality depends on the estimation of the distribution of noise term $\boldsymbol{\nu}$ in model (2.3).

After white noise cancellation and echo attenuation, the extracted features contain essentially structured sounds. This means that $\boldsymbol{\nu}$ is essentially constituted from structured noises. Due to the double nature of structured sounds (signal and noise) there is no more need to distinguish a structured noise from a structured signal in the estimation of the distribution of $\boldsymbol{\nu}$.

In order to obtain the *a priori* knowledge of this statistical property, some empirical density estimation has been made using samples from the training database. Figure 3-11 shows the plots of some feature vector examples for different types of sound. The aspect of these plots is very similar to white noise showing high uncorrelatedness between individual features.

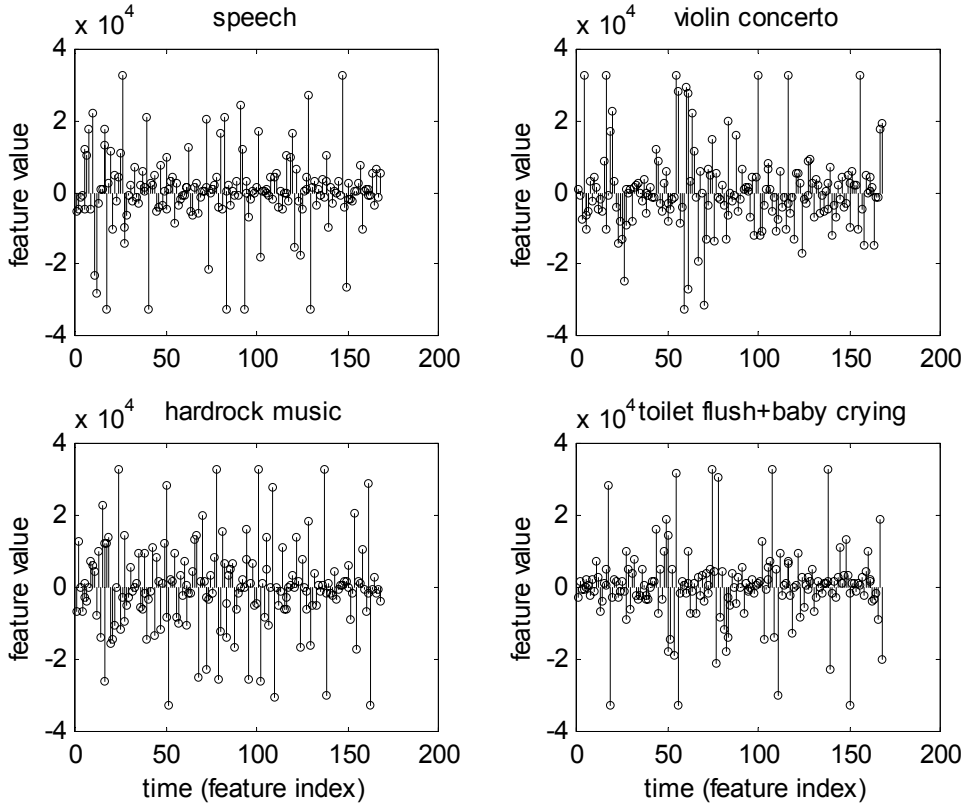


Figure 3-11: Feature vector examples

The corresponding histograms of these feature vectors are given in Figure 3-12, assuming all features of a feature vector are independently and identically distributed. In this empirical estimation the number of bin for the histograms is 21. Higher numbers of bin will show strong local fluctuation due to the Poisson phenomenon, since the number of features taken into the computation is limited.

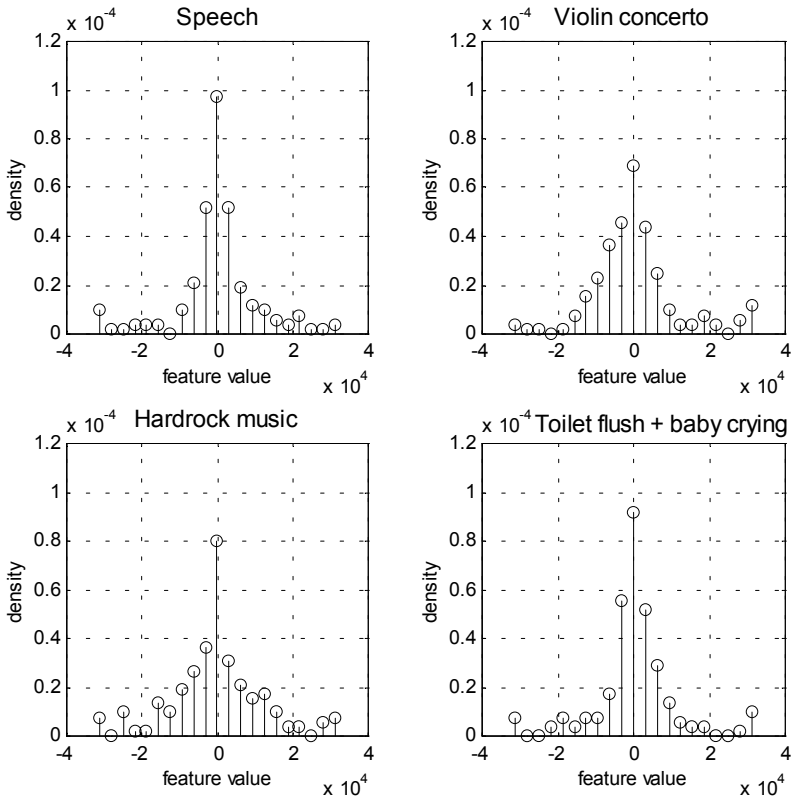


Figure 3-12: Empirical density (likelihood) function estimation of features.

It is noteworthy that the zero mean in the above density estimations confirms the term "zero template" defined in Section 2.2.4.

The main purpose of performing this density estimation is to find a best-fit statistical model that can be used to construct a simple and efficient similarity measure. From the above observation it becomes clear that the most sensible candidate for this statistical model is Laplacian.

A Laplacian PDF can be expressed as [Sayo96]

$$f(x) = \frac{1}{\sqrt{2\sigma^2}} exp \frac{-\sqrt{2}|x-\eta|}{\sigma} \tag{3.16}$$

with η mean and σ the standard deviation. Figure 3-13 gives Laplacian PDF curves with 0 mean and different variances.

Comparing Figure 3-12 and Figure 3-13 we can conclude that the feature vectors, representing variation of energy contour of structured sound, consistently converge towards a Laplacian like distribution. The signal types alter only the variance of the distribution. This consistency is a very valuable property that can be beneficial not only for building the robust distance measure (Chapter 5) but also for subsequent data compression in portable audience recording units (Chapter 4).

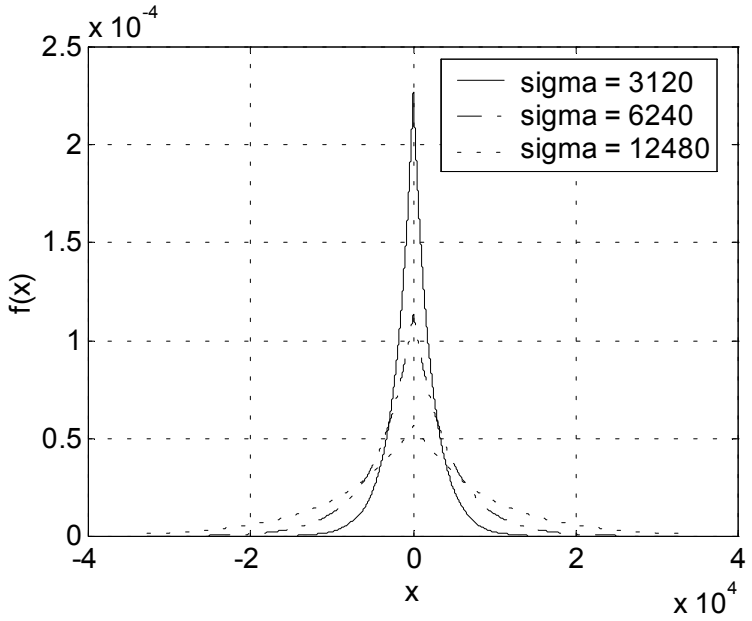


Figure 3-13: Laplacian PDF curve

3.6. Summary

In this chapter we have shown that the basic idea of the feature generation strategy is to extract the variation of the time average of local energy in the most informative frequency range (375 to 1500 Hz).

In order to achieve this objective, the algorithm performs the following operations:

- decompose the interesting frequency range of the incoming signal into 6 sub-bands using a 3-level dyadic tree filter bank (wavelet transform);
- extract each channel signal into smoothed energy using square operation and strong decimation;
- run the time derivative over the smoothed energy.

We have demonstrated that the white type noise and echo can be attenuated in the feature generation stage. And by special energy decimation, that is, keeping 1 decimation phase for audience data and keeping all decimation phase for studio data, the small time shift invariance can be obtained.

The data reduction factor realised at this feature extraction stage is more than 70 including strong decimation, rejection of irrelevant subbands and some minor filtering side effect loss.

The empirical noise distribution estimation has shown that remaining noises become highly decorrelated and Laplacian distributed.

Chapter 4
Data compression

The data compression described in this chapter is not part of the classification system. However, incorporating it into the overall audience measurement system is a necessity and it influences the classification quality by quantisation and added distortion.

After a brief theoretical review, we will show and justify the selected data compression scheme, the gamma companded scalar quantization for portable audience recording units, and the decimation-interpolation for studio recording units.

4.1. Data compression in portable audience recording units

After the feature vector extraction the data is considerably reduced. Even so, the portable audience recording unit, very limited in size, should gain further data reduction. This is accomplished by data compression.

4.1.1. Review for data compression and choice of method

When we speak of data compression or equivalently of source coding we refer to paired operations: compression (encoding) and reconstruction (decoding). The goal is to obtain in-between a compact data representation that makes data storage or data transmission effective. Such a compact representation is often obtained by exploiting some structures or properties existing in the data.

The coding method is multiple. Among all these data compression schemes we usually distinguish two major categories: lossless compression and lossy compression.

Lossless method compresses the data without loss of information and the original data can be reconstructed exactly from the compressed data. In such a compression scheme the major concern is to obtain a highest *compression ratio* or equivalently lowest *rate* (average bit per sample) with a reasonable complexity. Lempel-Ziv, Huffman, Arithmetic are the most popular coding methods in this category. They in general provide higher rates.

Lossy method involves a certain information loss when compressing the data, and the original data can therefore not be reconstructed exactly from the compressed data. In such a compression scheme, in addition to reducing rate, we must handle and minimise the information loss measured by *distortion* (difference between original and reconstructed data). Scalar quantization, vector quantization, differential encoding, predictive encoding are the most popular ones. In general, a lossy method is used for obtaining lower rates.

The effectiveness of a compression scheme depends on the characteristics of the data. A vector quantization scheme is most effective if the output of the source shows a high degree of clustering. A differential encoding scheme is most effective when the sample-to-sample difference is small. If the source output is truly random, it is best to use scalar quantization or lattice vector quantization. Thus, if a source exhibits certain well-defined characteristic, we can choose a compression scheme most suited for that characteristic.

Our choice is the scalar quantization because of the highly random source data (Figure 3-11) and the simplicity of the method. This choice will be further justified in the following subsection.

4.1.2. Gamma companded scalar quantization

Scalar quantization is a simple process, suitable for power limited implementation. However, the design of the quantizer has a significant impact on the amount of compression obtained and loss incurred in a lossy compression scheme. Let us have a close look at the related issues.

Quantization is the process of representing a large set of values with a much smaller set. It is executed by a quantizer.

The encoder part of the quantizer divides the range of values that an information source generates into a certain number of intervals, and represents each interval by a distinct codeword. Knowing the code only tells us the interval to which the source sample value belongs. It does not tell us which of the many values in the interval is the actual source sample value. This is a non-retrievable information loss.

For every codeword generated by the encoder, the decoder part of the quantizer generates a reconstruction value, the representative of the corresponding source interval. How this representative should be placed in the interval depends on the source probabilistic distribution.

Supposing that we have an input modelled by a random variable X with a probability density function $f_X(x)$ and we wish to quantise the source in question using a quantizer with M intervals, then the endpoints of the intervals, known as *decision boundaries*, form a $(M+1)$-element set $\{b_i\}_{i=0}^{M}$ and the representatives of the intervals, known as *reconstruction levels*, form a M-element set $\{y_i\}_{i=1}^{M}$. The information loss is often measured either by the *mean squared quantization error (msqe)* or its relative version *signal to noise ratio (SNR)*

$$msqe = \sum_{i=1}^{M} \int_{b_{i-1}}^{b_i} (x - y_i)^2 f_X(x) dx \tag{4.1}$$

$$SNR = \frac{\sigma_x^2}{msqe} \tag{4.2}$$

where σ_x^2 is the input signal variance. *SNR* is the ratio of the average energy of the input signal and the average energy of the coding error. Although these measures are given in continuous source form, it is straightforward to restate it for a discrete source case and the same properties follow.

In general, for different intervals the code lengths can be different and they form an M-element set $\{l_i\}_{i=1}^{M}$. The coding rate can then be written as

$$R = \sum_{i=1}^{M} l_i \int_{b_{i-1}}^{b_i} f_X(x) dx \tag{4.3}$$

Now, the general quantizer design task statement can be made as:

Given a distortion constraint

$$msqe \le C \tag{4.4}$$

find the decision boundaries, reconstruction levels and binary codes that minimise the rate given by (4.3), while satisfying (4.4).

Or, given a rate constraint

$$R \le C \tag{4.5}$$

find the decision boundaries, reconstruction levels and binary codes that minimise the distortion given by (4.1), while satisfying (4.5).

This is a complex task. It is often simplified in practice using fixed-length code words. In this case, the rate is simply the number of bit used to encode each output, and the quantizer design problem is simplified to finding decision boundaries and reconstruction levels that minimise distortion (4.1).

Often in practice the estimation of PDF is to be avoided. In this case we can use the sample variance as estimate of *msqe* and *SNR* [Papo91], that is, in N-sample case we have

$$msqe \approx \frac{1}{N-1}\sum_{j=1}^{N}\left(x_j - z_j\right)^2 \tag{4.6}$$

$$SNR = \frac{\sum_{j=1}^{N} x_j^2}{\sum_{j=1}^{N}\left(x_j - z_j\right)^2} \tag{4.7}$$

where z is the output of the quantizer. Notice that (4.1) remains valuable for theoretical development.

Uniform quantizer

There are many quantization techniques available with different characteristics. *Uniform quantizer* is the simplest design consisting of uniformly placed decision boundaries and reconstruction levels within the input range. A 3-bit most commonly used uniform quantizer, *midrise uniform quantizer*, for a uniformly distributed input, is illustrated in Figure 4-1 (a).

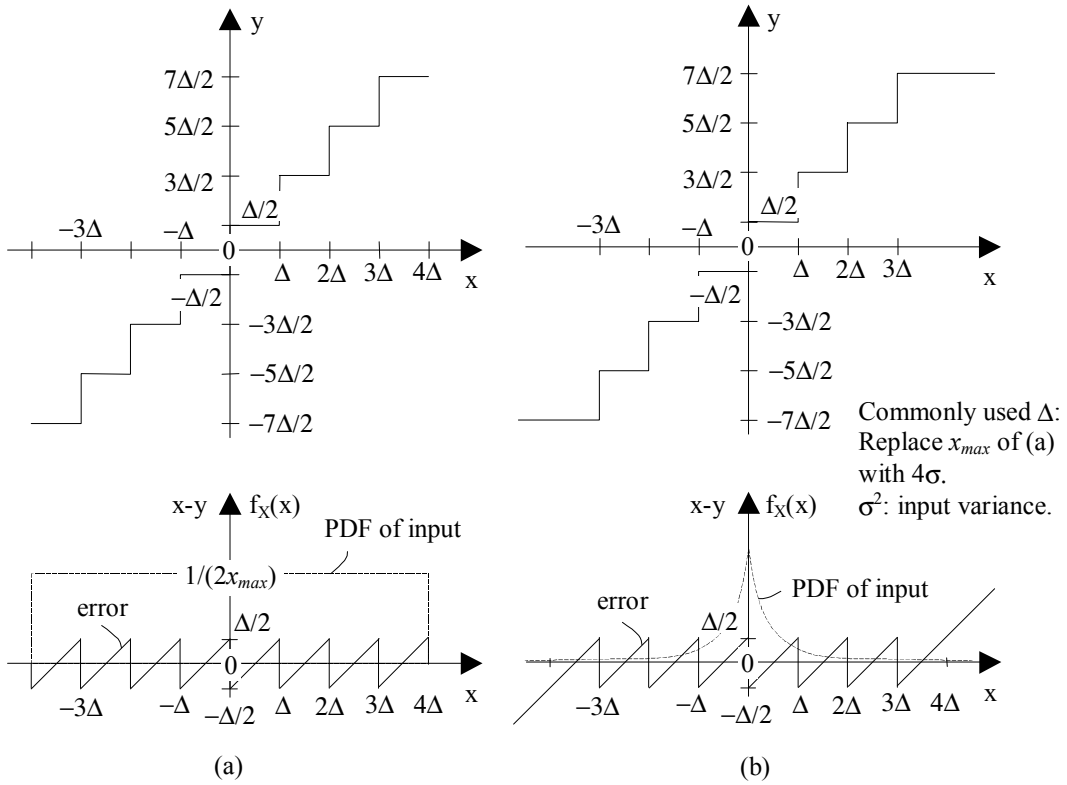


Figure 4-1: Input-output map of a midrise uniform quantizer. (a) for uniformly distributed input; (b) for non-uniformly distributed input.

In Figure 4-1(a), $\Delta = 2x_{max}/2^n = x_{max}/4$ is the interval size assuming input x is symmetrically distributed in $[-x_{max}, x_{max}]$ and $n = 3$ is the number of bits of the quantizer.

A uniform quantizer with uniformly distributed input can give a very simple relation between the SNR and number of bits n: $SNR = 6.02 \cdot n$ and higher quality of compression [Sayo96]. However, in case of non-uniformly distributed input, a uniform quantizer gives poor quality and if in addition the distribution is non-bounded, it is even not

directly practicable. A commonly used solution, called *4σ loading* approach, is illustrated in the Figure 4-1 (b). It is basically a uniform quantizer taking into account the input data distribution to set inner interval size. They provide improvement in quality but are highly sensitive to the mismatch of the PDF model. Many further improved solutions are possible. We will mention only a few of them that help to build our proposed solution.

PDF-optimised non-uniform quantizer (Lloyd-Max)

In order to decrease the average distortion, we can try to do finer quantization in the regions of high probability and rougher quantization in the low probability regions. This is the basic idea of the *non-uniform quantizer* family.

A direct approach for designing a best non-uniform quantizer is to place the decision boundaries and reconstruction levels, using the available probability model, that minimise (4.1). This type of quantizer is called *PDF optimised non-uniform quantizer*. To do so, it suffices to set to zero the derivative of (4.1) with respect to y_i and b_i. And as solutions for y_i and b_i, we get

$$y_i = \frac{\int_{b_{i-1}}^{b_i} x\, f_X(x)dx}{\int_{b_{i-1}}^{b_i} f_X(x)dx} \tag{4.8}$$

$$b_i = \frac{y_{i+1} + y_i}{2} \tag{4.9}$$

that is, a reconstruction level y_i is the centroid of the probability mass in the interval it represents and a decision boundary b_i is simply the midpoint of the two neighbouring reconstruction levels. The resolution of (4.8) (4.9) is not trivial. It should be done iteratively as Lloyd-Max algorithm does [Sayo96].

As can be expected, the PDF-optimised non-uniform quantizer improves significantly the quantization quality compared to the *4σ loading* uniform quantizer, notably in cases where there is higher concentration of data near origin. For example, in Laplacian distribution case more than 1 dB SNR improvement can be obtained for 3-bit quantizers. However, the PDF-optimised non-uniform quantizer is still sensible to the mis-

match of PDF model. The input-output map and input-error map of such a 3-bit quantizer for Laplacian input is plotted in Figure 4-2.

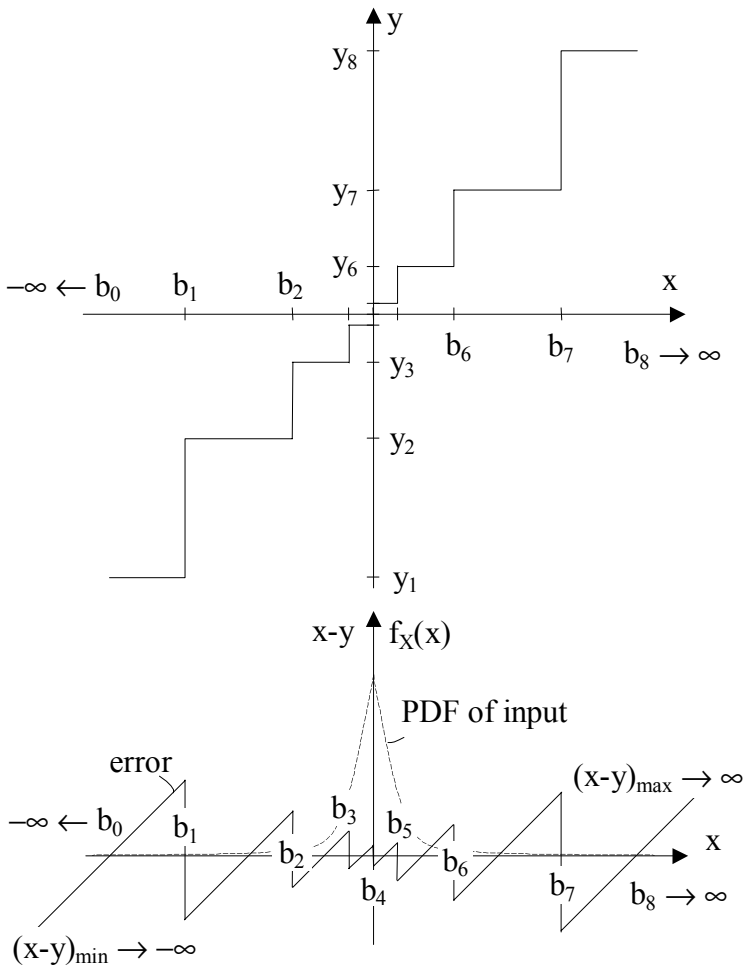


Figure 4-2: Non-uniform quantizer example: Laplacian PDF-optimised non-uniform quantizer by Lloyd-Max algorithm

An important drawback of the above methods dealing with the non-uniformly distributed input is the need for the exact PDF knowledge. It is highly unpractical for the application requiring lower power consumption.

As we can see in the Figure 4-2 a non-uniform quantizer, in the case where the input PDF is peaked at origin, can be viewed as a quantizer generated from a uniform quantizer with compressed inner intervals and expanded outer intervals. As a result, the input samples fall into more outer intervals.

Another possibility for finer quantization in the regions of high probability and rougher quantization in the low probability regions exists if we can bound the input by some value x_{max} (similar to normalising the input). This is the case of companding quantization.

Companding quantizer

We keep using a uniform quantizer. But before encoding, the input is pre-processed by a function called *compressor* to stretch the data outwards towards x_{max}, and after decoding, the output is post-processed by an inverse function called *expander* to release the data inwards towards the origin. This is the basic idea of so-called *companded quantization*[1]. The idea is shown in Figure 4-3. The question now is to find the compressor-expander pair that reflects the input data distribution so as to minimise the quantization distortion. Notice that the input data boundaries, x_{max} and $-x_{max}$, should not be changed during the companding operation.

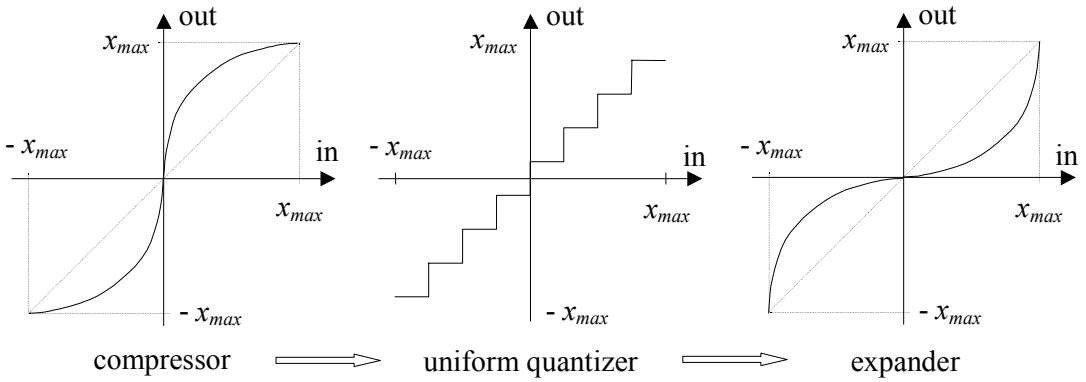


Figure 4-3: Companded quantization

According to the famous result, known as the *Bennett integral* [Benn48], in the case of high-rate companding quantizer (number of level M large, small interval size), the msqe can be expressed as

$$msqe = \frac{x_{max}^2}{3M^2} \int_{-x_{max}}^{x_{max}} \frac{f_X(x)}{(c'(x))^2} dx \tag{4.10}$$

[1] The term compand comes from com(press)+(ex)pand.

where $c(x)$ is the compressor function. If it takes the logarithmic form so that

$$c'(x) = \frac{x_{max}}{a|x|} \tag{4.11}$$

where a is a constant making $c(x)$ scalable, (4.10) can be evaluate as

$$msqe = \frac{a^2}{3M^2}\sigma_x^2 \tag{4.12}$$

That is, the SNR of such a quantizer is invariant to PDF mismatch! Such a compressor takes a logarithmic form as

$$c(x) = x_{max} + b \cdot ln\left(\frac{|x|}{x_{max}}\right) \tag{4.13}$$

where $b = x_{max} / a$ is a constant playing the role of scale. Unfortunately (4.13) cannot be directly applied in practice because it becomes unbounded when input tends towards zero. Despite of the fact that a logarithmic compressor is derived from some idealised assumption and has some difficulty in the region very near to the origin, it gives a valuable guideline for the search of a practical compressor.

Let us have a look at a widely used *μ-law* compander, applied to the telephone systems in North America and Japan. It belongs to the logarithmic compander family and will be used here as a comparison reference for a quick assessment of our quantizer. A *μ-law* compander is given by [Sayo96]

$$c(x) = x_{max}\frac{ln\left(1+\mu\frac{|x|}{x_{max}}\right)}{ln(1+\mu)}sgn(x) \tag{4.14}$$

$$c^{-1}(x) = \frac{x_{max}}{\mu}\left[(1+\mu)^{\frac{|x|}{x_{max}}} - 1\right]sgn(x) \tag{4.15}$$

where (4.14) is its compressor and (4.15) its expander. The μ-law companding function is plotted in Figure 4-4 for normalised input, that is, $x_{max} = 1$ but this is not necessary.

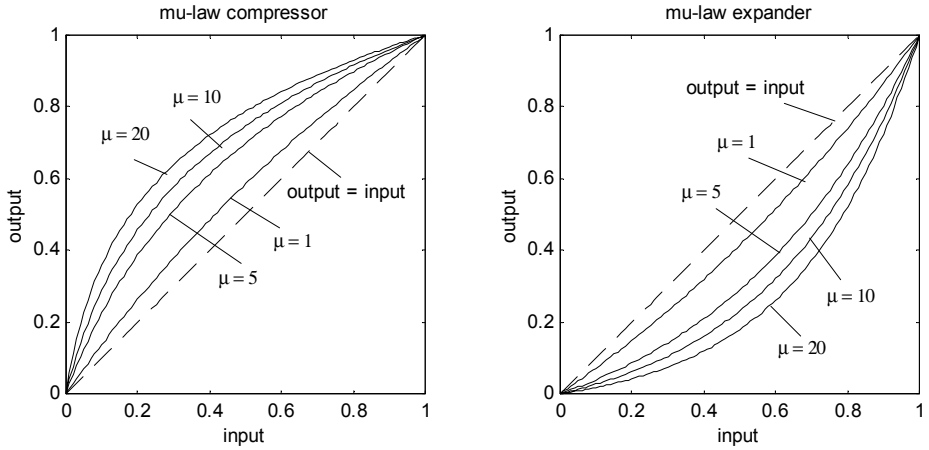


Figure 4-4: μ-law companding function

A much simpler compander can be proposed if we relax our pursuit of an idealised invariance on PDF mismatch, as with the gamma companding quantizer.

Gamma companding quantizer

Gamma function is often used in image intensity reproduction to compensate non-linear phenomena [Poyn96]. It is a power function

$$c(x) = \begin{cases} x^{\gamma} & x \geq 0 \\ -(-x)^{\gamma} & x < 0 \end{cases} \tag{4.16}$$

where γ is the function parameter. The inverse function of (4.16) takes exactly the same form except for the value of the parameter being the inverse to the one in (4.16). In order to emphasise the fact of it being a paired function we write down the inverse of (4.16) as follows:

$$c^{-1}(x) = \begin{cases} x^{\frac{1}{\gamma}} & x \geq 0 \\ -(-x)^{\frac{1}{\gamma}} & x < 0 \end{cases} \tag{4.17}$$

A graphical representation of the gamma function with different parameter values is given in Figure 4-5. Notice that the compressor and the expander are merged together since they are issued from the same function but different parameter values. Notice also that the input data normalisation is a necessity when using gamma companders. This is because a compander should not change the source data boundary x_{max} and the only value a power function does not change is $x_{max} = 1$.

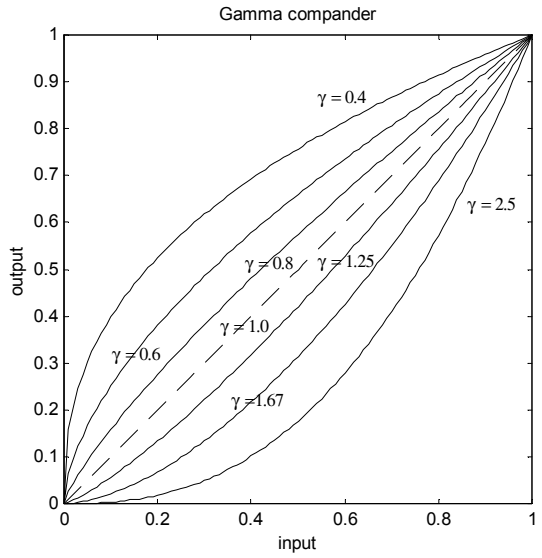


Figure 4-5: Gamma function

To find an optimal gamma compressor-expander pair it suffices to run the quantizer in a distortion minimisation process tuning the single parameter γ. This process is run on the available training database and the average result in case of 4-bit quantizer is illustrated in Figure 4-6.

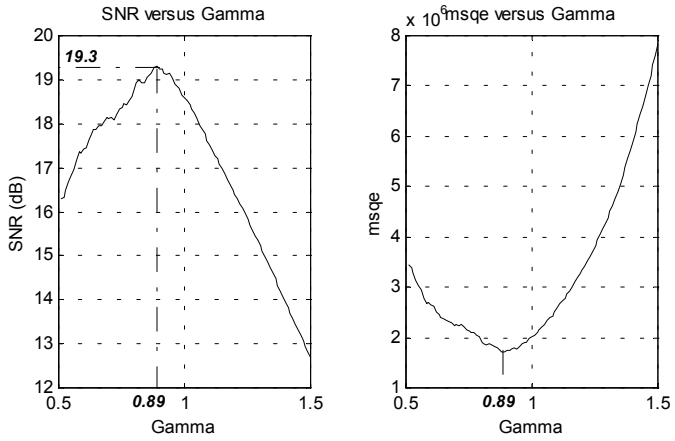


Figure 4-6: Distortion versus Gamma using 4-bit gamma companding quantizer

To make a valid comparison with the μ-law companding quantization, the same process is executed using a 4-bit μ-law companding quantizer and the result is presented in Figure 4-7.

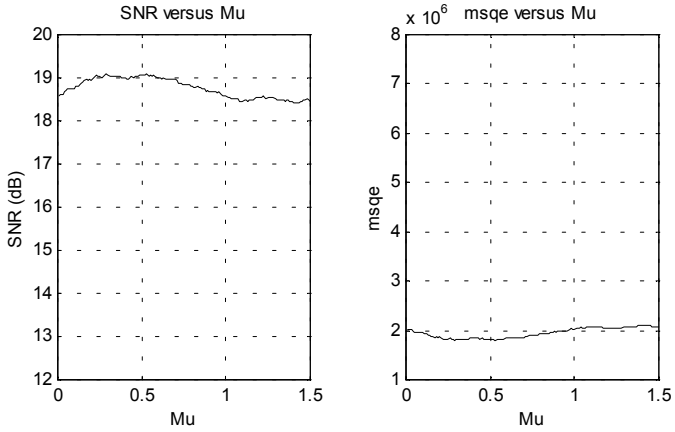


Figure 4-7: Distortion versus μ using 4-bit μ-law companding quantizer

As can be observed, the quantization quality of these two quantizers is similar at the optimal level. The input signal was in 16-bit representation for both μ-law and γ companded quantizers in the simulation.

It is worthwhile to recall that an intuitive thinking has led us from PDF-optimised non-uniform quantization to the companded quantization. Now, we can do the same but in the inverted sense. In other words, instead of applying compressor on the input data we can obtain exactly the same code set by applying the expander on the decision boundaries and this can be done off line during the design phase. However, on the decoding side we have no similar alternative.

The entropy computation of coded data, resulting in average entropy of 3.3, indicates that a minor additional compression is possible (by entropy coding or Lempel-Ziv coding for example). But further decrease of the rate leads to varying code length and a more complex code-decode scheme.

The simulation results (see Chapter 6) have shown that this quantization can significantly affect the matching quality in both the positive and the negative sense. This is because when it adds the quantization noise to the signal it suppresses simultaneously the small variation of the signal most often due to structured noise. Even so, the added quantization noise is undesirable and should be somehow eliminated. See Chapter 5 for the detailed discussion of the quantization noise compensation during the matching process.

4.2. **Data compression in studio units**

As pointed out in Chapter 3, after *lp*2 filtering the bandwidth of channel signals becomes very narrow allowing the audience data to be down sampled by an important factor (48). We will show that the studio data can also take advantage of this filtering to reduce the amount of data so as to facilitate the data transfer, although it has to retain all decimation phases to achieve the small time shift invariance.

The adopted technique is decimation / interpolation compression as shown in Figure 4-8. This is a natural choice.

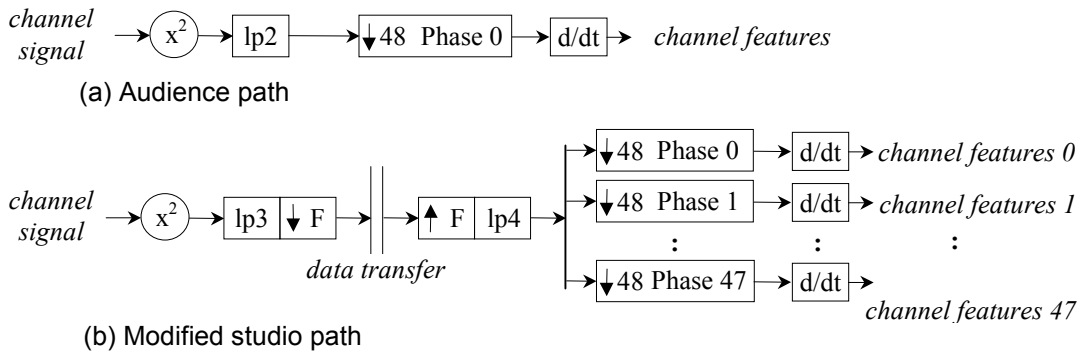


Figure 4-8: Studio data compression

As we can see in Figure 4-8b, after the channel signal is transformed into energy it is low-pass filtered by *lp*3, a window function having

equivalent frequency response as that of $lp2$. Then, the signal is down sampled by a loose factor $F < 48$. At this point the data transfer can take place with a reduced data volume. Between the data transfer and the phase-by-phase down sampling, the data is interpolated by a factor of F using $lp4$ as interpolation filter. $lp4$ is also a window function having equivalent frequency response to that of $lp2$.

Various combinations of $lp2$, $lp3$, $lp4$ and F have been tested. The simulation results show several interesting combinations of them that lead to a dramatic reduction of studio data without significant degradation of the system performance. See Chapter 6 for simulation results.

Note that the studio data compression can also follow the same compression scheme as that used in audience data compression but applied to each decimation phase. However, this method can only achieve compression by a factor of 4. As it will be shown in Chapter 6, the method decimation-interpolation can achieve a much more important compression rate.

4.3. Summary

In this chapter we have discussed in detail a 4-bit gamma companded quantization compression scheme for further data reduction in portable audience recording units, with optimised gamma value of about 0.9. This compression, in case of 16-bit arithmetic, will give further data reduction by a factor of 4, resulting in a total data reduction by factor of 280.

We have also described the possibility of compressing studio data using a decimation-interpolation scheme.

Chapter 5
Robust template matching

This chapter will give a detailed elaboration of a new distance metric, a generalised Manhattan norm, based on which the robust template matching is constructed. It will show the achievement of an efficient searching strategy using a coarse-fine method involving successively Gaussian and Laplacian statistical models.

5.1. Robust template matching using Laplacian statistic models

As pointed out in Chapter 2, if the distribution of the noise contained in the signal is available we can build an adequate distance measure that optimises the classification quality in Bayes' sense. What makes this possible is that in most cases the sample probabilistic properties of the features, representing the structured noise, consistently converge towards a Laplacian like distribution as pointed out in Chapter 3. Assigning the term robust to the algorithm stresses the use of Laplacian PDF models that best fit the noise distributions. This situation suggests the use of the Manhattan norm as a distance measure.

It is worthwhile also to recall that in this classification system the template matching operation addresses a 1-dimensional, multi-variate, 2-class problem as pointed out in Chapter 2.

5.1.1. Generalisation of the Manhattan norm: new metric

The Manhattan norm (2.13) is not yet optimal since the class variances are not correctly introduced. Let us now elaborate a new conditional distance measure that generalises the Manhattan norm.

Let us apply the Bayes classification rule to our problem. Given two classes c and c_0, where c (class under test) stands for the sound under test, c_0 (zero class) stands for a mixture of other possible sounds (noise), and a test feature vector $\boldsymbol{f}$, we class $\boldsymbol{f}$ into c if the *a posteriori* probabilities verify

$$P(c|\boldsymbol{f}) > P(c_0|\boldsymbol{f}) \tag{5.1}$$

Using L, the logarithmic of the ratio of these two probabilities, the Bayes classification rule can be equivalently stated as: we class $\boldsymbol{f}$ into c if

$$L = \ln\left(\frac{P(c_0|\boldsymbol{f})}{P(c|\boldsymbol{f})}\right) \tag{5.2}$$

is less than zero. Applying Bayes conditional probability relation

$$P(c_i|\boldsymbol{f}) = \frac{p(\boldsymbol{f}|c_i)P(c_i)}{p(\boldsymbol{f})} \tag{5.3}$$

where $i = 0, \text{nil}$, L becomes

$$L = \ln\left(\frac{p(\boldsymbol{f}|c_0)P(c_0)}{p(\boldsymbol{f}|c)P(c)}\right) = \ln\left(\frac{p(\boldsymbol{f}|c_0)}{p(\boldsymbol{f}|c)}\right) + \ln\left(\frac{P(c_0)}{P(c)}\right) \tag{5.4}$$

This formula can be interpreted as: we class $\boldsymbol{f}$ into c if the logarithmic of likelihood ratio

$$l = \ln\left(\frac{p(\boldsymbol{f}|c_0)}{p(\boldsymbol{f}|c)}\right) \overset{\text{feature independence}}{=} \sum_{j=1}^{N} ln\left(\frac{p(f_j|c_0)}{p(f_j|c)}\right) = \sum_{j=1}^{N} l_j \tag{5.5}$$

is less than a certain threshold value l_{thr} when the *a priori* probability

ratio can be practically treated as constant. In the context of audience measurement application it is difficult to get the information about class a priori probability. This means that the threshold value l_{thr} should be determined using some learning mechanism, from a training database for example. Clearly, this is a suboptimal solution.

In relation (5.5), we find again the assumption of feature independence that simplifies a complex joint relation into a simple sum of marginal relation (see (2.8)). Applying Laplacian models to marginal PDF $p(f_j|c)$ and $p(f_j|c_0)$, assuming all features for one class have the same variance, we get

$$p(f_j|c) = \frac{1}{\sqrt{2\sigma^2}} exp \frac{-\sqrt{2}|f_j - r_j|}{\sigma} \tag{5.6}$$

$$p(f_j|c_0) = \frac{1}{\sqrt{2\sigma_0^2}} exp \frac{-\sqrt{2}|f_j - r_{0,j}|}{\sigma_0} \tag{5.7}$$

where we have replaced the class mean by template. Substituting them for $p(f_j|c)$ and $p(f_j|c_0)$ in (5.5) and knowing $r_{0,j} = 0$ (zero template), the likelihood ratio l_j for j^{th} individual feature becomes

$$l_j = \ln\left(\frac{\sigma}{\sigma_0}\right) + \sqrt{2}\left(\frac{|f_j - r_j|}{\sigma} - \frac{|f_j|}{\sigma_0}\right) \tag{5.8}$$

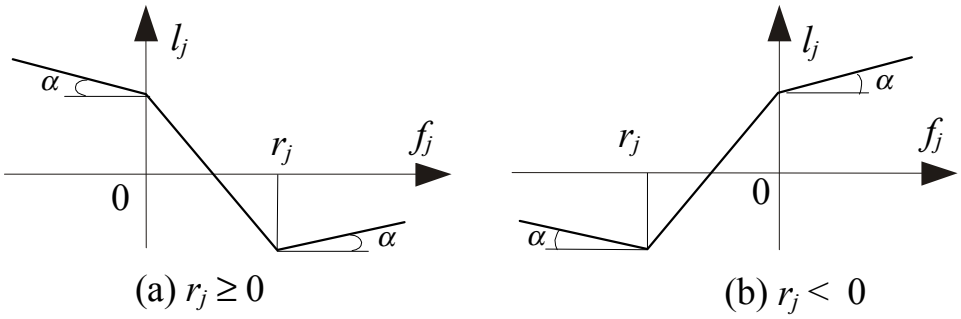
The expression (5.8) shows that the capability of distinguishing c from c_0 depends on both class means and class variances. Even in the case where the two class-means coincide completely, we can still distinguish them as long as their variances are sufficiently different.

Using likelihood ratio (5.5) as decision criterion can lead to minimum classification error, but as shown in (5.8) it requires the knowledge of class variances. The direct estimation of these variances can bring about a difficult situation. In the following paragraphs a new but equivalent to likelihood ratio measurement will be built and it will lead to an implicit but simplified and optimised estimation of class variances.

Let us begin with a closer look at (5.8). If $r_j \geq 0$ (5.8) can be re-written as:

$$l_j = \begin{cases} \ln\left(\frac{\sigma}{\sigma_0}\right) + \sqrt{2}\frac{r_j}{\sigma} - \sqrt{2}\left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right) f_j & f_j < 0 \\ \ln\left(\frac{\sigma}{\sigma_0}\right) + \sqrt{2}\frac{r_j}{\sigma} - \sqrt{2}\left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right) f_j & 0 \leq f_j \leq r_j \\ \ln\left(\frac{\sigma}{\sigma_0}\right) - \sqrt{2}\frac{r_j}{\sigma} + \sqrt{2}\left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right) f_j & r_j < f_j \end{cases} \quad (5.9)$$

The Figure 5-1 (a) gives a more intuitive presentation of (5.9) with $\frac{1}{\sigma} - \frac{1}{\sigma_0} = tg(\alpha)$.



(a) $r_j \geq 0$ (b) $r_j < 0$

Figure 5-1: The logarithmic likelihood ratio of j^{th} feature

Similarly, if $r_j < 0$ (5.8) can take the form:

$$l_j = \begin{cases} \ln\left(\frac{\sigma}{\sigma_0}\right) - \sqrt{2}\frac{r_j}{\sigma} + \sqrt{2}\left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right) f_j & 0 < f_j \\ \ln\left(\frac{\sigma}{\sigma_0}\right) - \sqrt{2}\frac{r_j}{\sigma} + \sqrt{2}\left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right) f_j & r_j \leq f_j \leq 0 \\ \ln\left(\frac{\sigma}{\sigma_0}\right) + \sqrt{2}\frac{r_j}{\sigma} - \sqrt{2}\left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right) f_j & f_j < r_j \end{cases} \quad (5.10)$$

And its graphical representation is shown in Figure 5-1 (b).

Recall that the likelihood ratio criterion (5.5) requires an associate threshold value l_{thr} below which the test pattern is considered belonging to c. The change of origin or the scaling operation of Figure 5-1 will not modify this property. Now we consider another discrimination measure, let us call it D. Like in the case of likelihood ratio l, its j^{th} summand D_j for the j^{th} individual feature is defined as follows (change of origin):

$$D_j = l_j - \ln\left(\frac{\sigma}{\sigma_0}\right) + \sqrt{2}\frac{|r_j|}{\sigma_0} = \sqrt{2}\left(\frac{|f_j - r_j|}{\sigma} - \frac{|f_j|}{\sigma_0} + \frac{|r_j|}{\sigma_0}\right) \tag{5.11}$$

Like in the development of (5.8), according to $r_j \geq 0$ or $r_j < 0$ (5.11) can be respectively rewritten as:

$$D_j = \begin{cases} \sqrt{2}\left[-\left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right)r_j + \left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right)f_j\right] & r_j < f_j \\ \sqrt{2}\left[\left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right)r_j - \left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right)f_j\right] & 0 \leq f_j \leq r_j \\ \sqrt{2}\left[\left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right)r_j - \left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right)f_j\right] & f_j < 0 \end{cases} \tag{5.12}$$

$$D_j = \begin{cases} \sqrt{2}\left[\left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right)r_j - \left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right)f_j\right] & f_j < r_j \\ \sqrt{2}\left[-\left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right)r_j + \left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right)f_j\right] & r_j \leq f_j \leq 0 \\ \sqrt{2}\left[-\left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right)r_j + \left(\frac{1}{\sigma} - \frac{1}{\sigma_0}\right)f_j\right] & 0 < f_j \end{cases} \tag{5.13}$$

Further scaling D_j into

$$d_j = \frac{D_j}{\sqrt{2}\left(\frac{1}{\sigma} + \frac{1}{\sigma_0}\right)} \tag{5.14}$$

so that (5.12) and (5.13) become:

$$\text{if } r_j \geq 0 \qquad d_j = \begin{cases} -(r_j - f_j)\cdot\beta & r_j < f_j \\ r_j - f_j & 0 \leq f_j \leq r_j \\ r_j - \beta \cdot f_j & f_j < 0 \end{cases} \tag{5.15}$$

$$\text{if } r_j < 0 \qquad d_j = \begin{cases} (r_j - f_j)\cdot\beta & f_j < r_j \\ -(r_j - f_j) & r_j \leq f_j \leq 0 \\ -(r_j - \beta \cdot f_j) & 0 < f_j \end{cases} \tag{5.16}$$

where

$$\beta = \frac{\frac{1}{\sigma} - \frac{1}{\sigma_0}}{\frac{1}{\sigma} + \frac{1}{\sigma_0}} = \frac{\sigma_0 - \sigma}{\sigma_0 + \sigma} \tag{5.17}$$

is an unique parameter containing useful information about class variances. Let us call it *weighting factor* since it places a role of tuning parameter of down-weight large error in robust estimation [Drap98]. It is positive valued because c, representing one type of sound to be recognised, should have smaller variance than that of c_0, representing the sum of all other sounds. It can be estimated using the training database as will be explained in Chapter 6.

The expression (5.15) and (5.16) can be combined into a single, more compact and tidy form using the absolute value of d_j:

$$Delta_j = |d_j| = \begin{cases} |r_j - f_j|\cdot\beta & \text{sign}(r_j) = \text{sign}(f_j) \ \&\ |f_j| > |r_j| \\ |r_j - f_j| & \text{sign}(r_j) = \text{sign}(f_j) \ \&\ |f_j| \leq |r_j| \\ |r_j - \beta \cdot f_j| & \text{sign}(r_j) \neq \text{sign}(f_j) \end{cases} \tag{5.18}$$

It is interesting to notice that the role of test pattern $\boldsymbol{f}$ and template $\boldsymbol{r}$ can be inter-changed without incurring any classification quality difference. That is, $Delta_j$ can be also defined as:

$$Delta_j = \begin{cases} |f_j - r_j| \cdot \beta & \text{sign}(f_j) = \text{sign}(r_j) \ \& \ |r_j| > |f_j| \\ |f_j - r_j| & \text{sign}(f_j) = \text{sign}(r_j) \ \& \ |r_j| \le |f_j| \\ |f_j - \beta \cdot r_j| & \text{sign}(f_j) \neq \text{sign}(r_j) \end{cases} \tag{5.19}$$

This is because $\boldsymbol{f} = \boldsymbol{r} + \boldsymbol{v}$ and $\boldsymbol{r} = \boldsymbol{f} + \boldsymbol{v}$ have exactly the same statistic significance. We are free to choose between (5.18) and (5.19).

We complete the building procedure of the new robust distance by replacing r_j with $s \cdot r_j$ where s is obtained from a scale retrieval process as will be explained in Section 5.1.3. Finally, the total distance measure in this new metric becomes

$$Delta = \sum_{j=1}^{N} Delta_j \tag{5.20}$$

with $Delta_j$ defined as

$$Delta_j = \begin{cases} |f_j - s \cdot r_j| \cdot \beta & \text{sign}(f_j) = \text{sign}(s \cdot r_j) \ \& \ |s \cdot r_j| > |f_j| \\ |f_j - s \cdot r_j| & \text{sign}(f_j) = \text{sign}(s \cdot r_j) \ \& \ |s \cdot r_j| \le |f_j| \\ |f_j - \beta \cdot s \cdot r_j| & \text{sign}(f_j) \neq \text{sign}(s \cdot r_j) \end{cases} \tag{5.21}$$

(5.21) is illustrated in Figure 5-2.

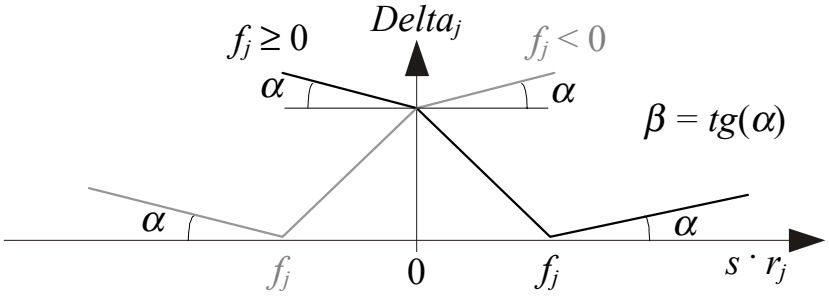


Figure 5-2: Optimal conditional distance measure

If we consider that class c has much smaller variance compared with that of the class c_0 we have $\sigma << \sigma_0$ and therefore $\beta \approx 1$ (quasi-ideal noiseless audience data). This leads to $Delta_j \approx |f_j - s \cdot t_j|$ and

Manhattan norm results in all cases. In other words, Manhattan norm is, as a special case of *Delta*, more robust for noiseless situations. Another extreme case is $\sigma \approx \sigma_0$ and therefore $\beta \approx 0$, corresponding to heavily noisy audience data. The real world is halfway in-between these two extremes.

5.1.2. Quantization noise cancellation

The data compression in portable audience recording units introduces important quantization noise. The statistical nature of quantization noise is quite different from that of environmental structured sound noise. It is often modelled as white and uniformly or Gaussian distributed [Vaid93] so as being unable to be covered by robust estimation. We have to treat it separately. A simple solution is to define an uncertain interval for each expected decoded value of audience data and ignore the matching error ($Delta = 0$) in this interval.

In our 4-bit gamma companding quantization case all possible expected reconstruction values and their corresponding uncertainty intervals are listed in Table 5-1.

Table 5-1: Reconstructed audience data set and uncertainty interval

f_{min} :	$-7^{\frac{1}{\gamma}}$	$-6^{\frac{1}{\gamma}}$	$\cdots$	-1	0	0	1	$\cdots$	$6^{\frac{1}{\gamma}}$	$7^{\frac{1}{\gamma}}$
f_{exp} :	$-7.5^{\frac{1}{\gamma}}$	$-6.5^{\frac{1}{\gamma}}$	$\cdots$	$-1.5^{\frac{1}{\gamma}}$	$-0.5^{\frac{1}{\gamma}}$	$0.5^{\frac{1}{\gamma}}$	$1.5^{\frac{1}{\gamma}}$	$\cdots$	$6.5^{\frac{1}{\gamma}}$	$7.5^{\frac{1}{\gamma}}$
f_{max} :	$-8^{\frac{1}{\gamma}}$	$-7^{\frac{1}{\gamma}}$	$\cdots$	$-2^{\frac{1}{\gamma}}$	-1	1	$2^{\frac{1}{\gamma}}$	$\cdots$	$7^{\frac{1}{\gamma}}$	$8^{\frac{1}{\gamma}}$

In Table 5-1, f_{exp} denotes the expected value set of the uncertainty intervals, f_{min} denotes the corresponding lower bound value set, and f_{max} denotes the upper bound value set. Notice that there is a "symmetry" between f_{min} and f_{max}: For each uncertainty interval it is the border nearer to origin that is called lower border with the other being called upper border. Each column in Table 5-1 represents such an interval with the expected value in the middle.

Applying this "insensitive interval" solution to the robust distance measure (5.21), we get

$$Delta_j = \begin{cases} \beta \cdot \left| f_{j,max} - s \cdot r_j \right| & \text{sign}(f_{j,exp}) = \text{sign}(r_j) \,\&\, \left| s \cdot r_j \right| > \left| f_{j,max} \right| \\ 0 & \text{sign}(f_{j,exp}) = \text{sign}(r_j) \,\&\, \left| f_{j,min} \right| < \left| s \cdot r_j \right| \le \left| f_{j,max} \right| \\ \left| f_{j,min} - s \cdot r_j \right| & \text{sign}(f_{j,exp}) = \text{sign}(r_j) \,\&\, \left| s \cdot r_j \right| \le \left| f_{j,min} \right| \\ \left| f_{j,min} - \beta \cdot s \cdot r_j \right| & \text{sign}(f_{j,exp}) \neq \text{sign}(r_j) \end{cases} \quad (5.22)$$

The metric (5.22) can be graphically illustrated as in Figure 5-3.

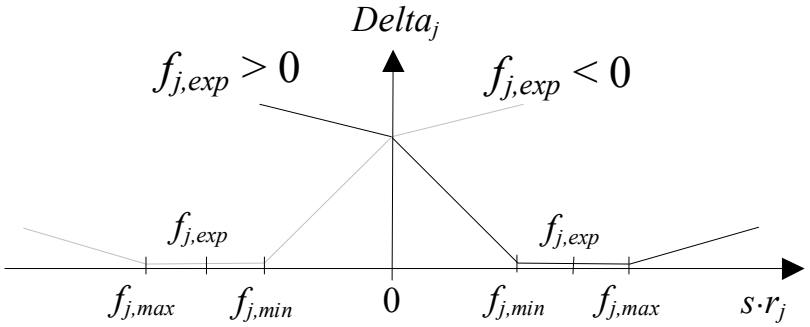


Figure 5-3: Modified optimal conditional distance measure using "insensitive interval" solution to compensate the quantization error.

Notice, when $f_{min} = f_{max} = f_{exp}$, all will turn back to the result given in Section 5.1.1.

5.1.3. Scale retrieval and matching error minimisation

In general, the audience data is not on a common scale (loudness) with the studio data. This means that a scale retrieval process has to be carried out in order to make valid matches. This explains the reason for introducing the parameter s in (5.21).

The scale retrieval is done within a band-wise error minimisation process

$$ErrMin_k = \min_s \sum_{j=1}^{N_l} Delta_{k,j} \quad (5.23)$$

where $Delta_{k,j}$ is the robust distance measure as defined in (5.22), N_l is the number of features within one subband, $k \in [1 \ldots N_b]$ is the band index, and N_b is the number of bands.

Doing this scale retrieval band-wisely has two advantages: 1) it can make the system invariant against frequency dependent sound power damping; 2) it can speed up the process when compared to running over all bands.

The most sensible candidate for estimating s can be deduced as follows. Consider the pattern $\boldsymbol{f}$ being an intact image of reference $\boldsymbol{r}$, that is, $\boldsymbol{f}$ is purely a scaled version of $\boldsymbol{r}$. In this case we can expect to obtain $\boldsymbol{f} - s \cdot \boldsymbol{r} = \boldsymbol{0}$. This gives $s = f_j / r_j$ for any $j \in [1 \cdots N_l]$. In noisy cases if $\boldsymbol{f}$ contains damaged $\boldsymbol{r}$ we expect to be able to find some scale $s_j = f_j / r_j$ for certain $j \in [1 \cdots N_l]$ that minimises the distance. Based on this consideration, this scale retrieval operation is run in a loop that scans all N_l possible s_j and selects the one that minimises the band-wise distance (error). Finally, it is these minimised band-wise distances that are summed together to form the overall distance measure

$$ErrMin = \sum_{k=1}^{N_b} ErrMin_k \tag{5.24}$$

This scale retrieval is a heavy task since to obtain one minimised match value $ErrMin_k$ we have to compute N_l time $\sum_{j=1}^{N_l} Delta_j$. In Chapter 7 we will discuss the possibility of lightening this process.

5.1.4. Normalised matching certainty coefficient

The robust distance metric developed in the above subsections does not take into account the signal energy level. Clearly, the same *ErrMin* is much more significant for a low energy signal than for a high energy one. Furthermore, *ErrMin* is numerically unbounded. To solve these problems we need a normalised distance measure $\widetilde{D}$.

Besides, it is natural to prefer a "positive" measure of the matching goodness to a "negative" one like $\widetilde{D}$. For this reason, we define the *matching certainty coefficient* or, to be short, *match value* as

$$C_{Lap} = 1 - \widetilde{D} \tag{5.25}$$

and impose its range to be $[-1,1]$. Consequently, the range of $\widetilde{D}$ results in $[0,2]$. The subscript "Lap" of C_{Lap} indicates that the statistic basis is Laplacian.

Assuming that in the worst case we have $s \cdot r_j = -f_j$ and $\beta = 1$, from (5.21), (5.23) and (5.24) we can estimate the range of *ErrMin* as being

$$\left[0 \quad 2 \cdot \sum_{j=1}^{N} \left| f_j \right| \right]$$

where N is the total number of features. In the case we use (5.22) instead of (5.21), that is, we wish to cancel the quantization noise, f_j should be replaced by $f_{j,min}$ since it is most often used in *ErrMin* computation. In the rare case where *ErrMin* exceeds this range we can saturate it at the border. Now, we can write down the normalised matching certainty coefficient for robust template matching as

$$C_{Lap} = 1 - \widetilde{D} = 1 - \frac{ErrMin}{\sum_{j=1}^{N} \left| f_{j,min} \right|} \tag{5.26}$$

or

$$C_{Lap} = 1 - \widetilde{D} = 1 - \frac{ErrMin}{\sum_{j=1}^{N} \left| f_j \right|} \tag{5.27}$$

according to whether or not we want to cancel quantization noise.

The robust certainty coefficient (match value) elaborated here can be viewed as a result of the continuation of the feature extraction based on a sample Laplacian statistic. It tends to give a *sufficient statistic*[1] for

[1] Refer to [McDo95] [Kay93] [Kay98] for detailed description on sufficient statistic.

class labelling to form a much simpler one-variate two-class problem. The study of the latter, the subject of the next chapter, will give us the decision rules so that we can label the test pattern "belonging" or "not belonging" to the template under test with reasonable labelling error.

5.2. Hierarchical searching and the use of Gaussian models

5.2.1. Situation

In the audience measurement system, at both studio and audience ends, the recording processes are time-stamped by calibrated clocks with reasonable precision. However, the compensation of long-period cumulated time stamp error often requires expensive equipment. The solution to this problem is enlarging recording margins on the studio side where the data storage problem is less critical. Correspondingly, the searching scan should be done by moving a small test pattern (audience data) within a large template (studio data) as shown in Figure 5-4, where the length K of the test pattern $\boldsymbol{f} = [f_j]_{j=1}^{K}$ is smaller than the length N of the template under test $\boldsymbol{r} = [r_j]_{j=1}^{N}$.

Unfortunately, performing directly the finest searching scan using robust Laplacian similitude metric C_{Lap} is computationally expensive. We want to find a practical searching strategy using a simpler distance or matching certainty measure to locate the test pattern in the template. As long as this location is consistent with that found using robust metric, we can then compute the value of the robust metric at the found location to give better matching quality. This is the basic reason of using Gaussian template matching.

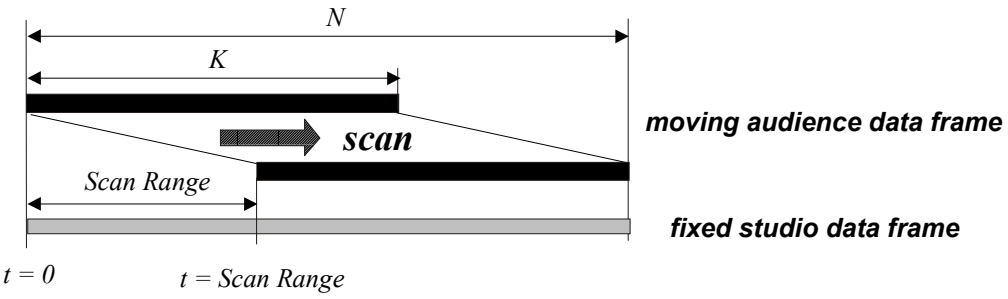


Figure 5-4: Searching scan of a test pattern (audience data) in a template (studio data)

5.2.2. Template matching using Gaussian models

As shown in Chapter 2, when applying Gaussian models to the likelihood functions, the Bayesian classification rule results in minimisation of the squared Euclidean norm

$$d_{euc}^2(t)=\sum_{j=t}^{t+N}\left(f(j-t)-r(j)\right)^2 \tag{5.28}$$

where $t\in[0 \quad \cdots \quad (N-K)]$ is the scan parameter (time). Notice that the test pattern remains unchanged in the scan process. Our problem now is to find the time location $\hat{t}$ within the scan range at which (5.28) is minimised.

The direct use of (5.28) will lead us to almost the same degree of complexity as in the case where the Laplacian model would be applied, because of the scale retrieval operation. In order to avoid this, let us construct a computationally more attractive measure [Theo99].

In fact, (5.28) is equivalent to

$$d_{euc}^2(t)=\sum_{j=t}^{t+N-1} f^2(j-t)+\sum_{j=t}^{t+N-1} r^2(j)-2\sum_{j=t}^{t+N-1} f(j-t)\cdot r(j) \tag{5.29}$$

For time-location searching purpose, when we move the test pattern within the template, the first summand is constant hence contributes nothing to minimisation of d_{euc}^2. The second summand is the template energy with which d_{euc}^2 varies increasingly. The third summand is the cross-correlation between the template and the test pattern and d_{euc}^2 varies decreasingly with it. Based on these observations the following alternative of similitude measurement (matching certainty coefficient or match value) can be defined as:

$$C_{Gau}(t)=\frac{\left(\sum_{j=t}^{t+N-1} f(j-t)\cdot r(j)\right)^2}{\sum_{j=t}^{t+N-1} f^2(j-t)\sum_{j=t}^{t+N-1} r^2(j)} \tag{5.30}$$

where the subscript "Gau" indicates the statistic basis is Gaussian. The corresponding dissimilitude is related to $C_{Gau}(t)$ as

$$d(t) = 1 - C_{Gau}(t) \tag{5.31}$$

$C_{Gau}(t)$ is nothing else than the *correlation coefficient* [Bend93]. This widely used similitude measurement is dominated by the cross-correlation between the test pattern $\boldsymbol{f}$ and the template $\boldsymbol{r}$. The influence of the template energy variation is taken into account in its denominator. The test pattern energy in the denominator of $C_{Gau}(t)$ is just for normalisation purpose.

According to Cauchy-Schwarz inequality [Papo91]

$$\left(\sum_{j=t}^{t+N-1} f(j) \cdot r(j-t) \right)^2 \le \sum_{j=t}^{t+N-1} f^2(j) \sum_{j=t}^{t+N-1} r^2(j-t) \tag{5.32}$$

$C_{Gau}(t)$ takes its value in the interval $[0,1]$. It reaches its maximum 1 when, in the moving observation window $[t,\ \ t+N-1]$, $\boldsymbol{f}$ is a scaled version of $\boldsymbol{r}$, and takes its minimum value 0 when they are completely uncorrelated.

The most remarkable advantage of using $C_{Gau}(t)$ as similitude measure is that the problem of scale retrieval, as it arose in the case of Laplacian statistic, is self-solved in the formulation. This can be easily verified by substitution of $\boldsymbol{r}$ by $s \cdot \boldsymbol{r}$, where s is the scale that should be otherwise estimated. That is why Gaussian models are computationally attractive.

For implementation convenience $C_{Gau}(t)$ defined in (5.30) takes a slightly different form as given in (5.33),

$$C_{Gau}(t) = \frac{\sum_{k=1}^{N_b} \left\{ \left(\sum_{j=t}^{t+N_l-1} f(k, j-t) \cdot r(k,j) \right)^2 \Big/ \sum_{j=t}^{t+N_l-1} r^2(k,j) \right\}}{\sum_{i=t}^{t+N_b \cdot N_l - 1} f^2(i-t)} \tag{5.33}$$

that is, the sums are first divided into band wise sums of length N_l, then

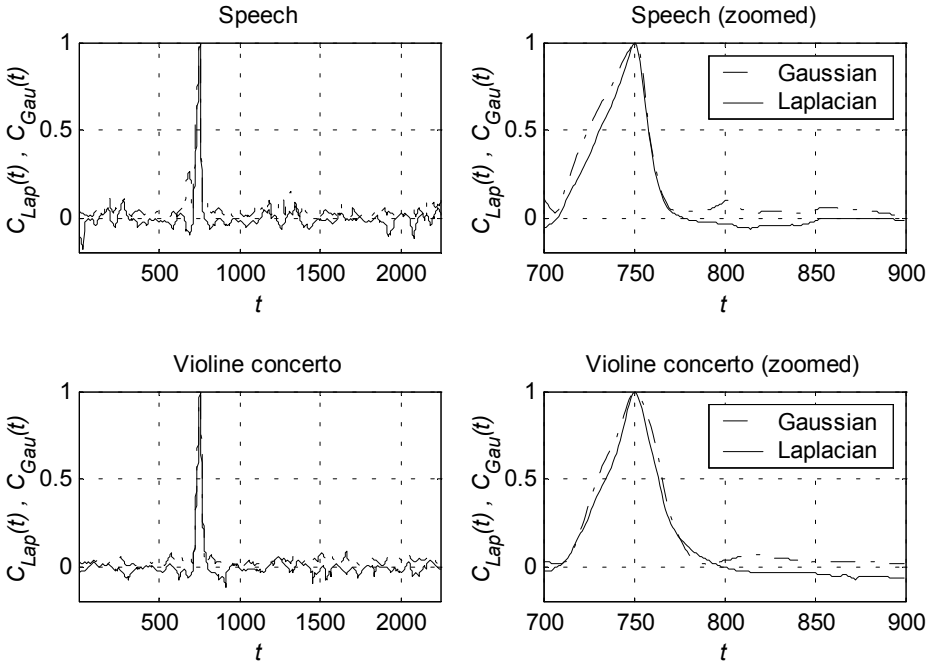
sum all N_b bands together, except for the term of f^2. Note that (5.30) and (5.33) have the common range $[0,1]$ and reach the borders with the same conditions. However, (5.33) is more suitable for a mixed-metric searching scheme as explained later in this section.

It has been shown, by simulation, that the Gaussian match computation is at least 5 time faster than the Laplacian one.

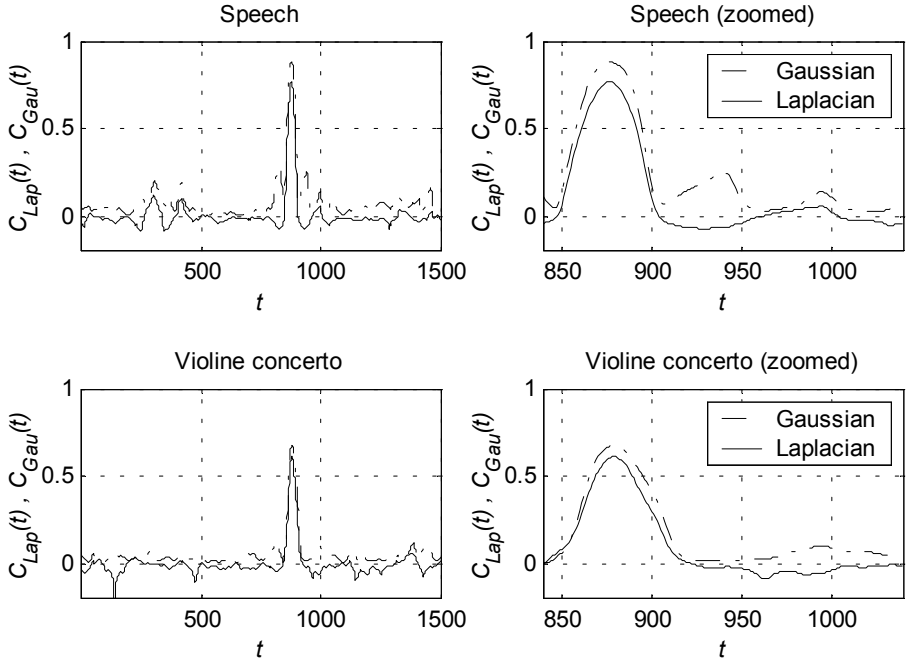
5.2.3. Time location consistency

The Gaussian match value $C_{Gau}(t)$ is expressed as a function of scan time in (5.33), let us call it Gaussian *matching function*. Similarly, we can introduce the scan time parameter t into the formulas (5.20) ~ (5.27) to obtain the Laplacian matching function $C_{Lap}(t)$ for Laplacian statistic.

To confirm peak-value location consistency between $C_{Gau}(t)$ and $C_{Lap}(t)$ some empirical observations have been made. Figure 5-5 shows some of the results.



(a) Auto matching[2] of studio data



(b) Cross matching between audience and studio data

Figure 5-5: Consistency between $C_{Gau}(t)$ *and* $C_{Lap}(t)$

[2] By *auto matching* we mean a signal match itself, a term comparable with auto-correlation. Similarly, we define *cross matching* to be a signal match another, a term comparable with cross-correlation.

The maximum observed difference of the peak-value location between $C_{Gau}(t)$ and $C_{Lap}(t)$ is equal to one finest scan resolution, equivalent to about 2.6 ms. This is not significant for the matching quality since, as will be shown in Chapter 6, the matching function is about 3-time over-sampled in the finest resolution scan.

5.2.4. Hierarchical (coarse-fine) searching

As can be seen in Figure 5-5, the peak-width, above its floor noise, of the matching function is about 30~40 samples. This situation suggests that, instead of scanning directly with the finest scan step, we can first scan quite roughly to locate the peak area and then finely scan only the detected peak neighbourhood.

Based on the above observation, a three-stage coarse-fine searching-matching process is adopted as illustrated in Figure 5-6.

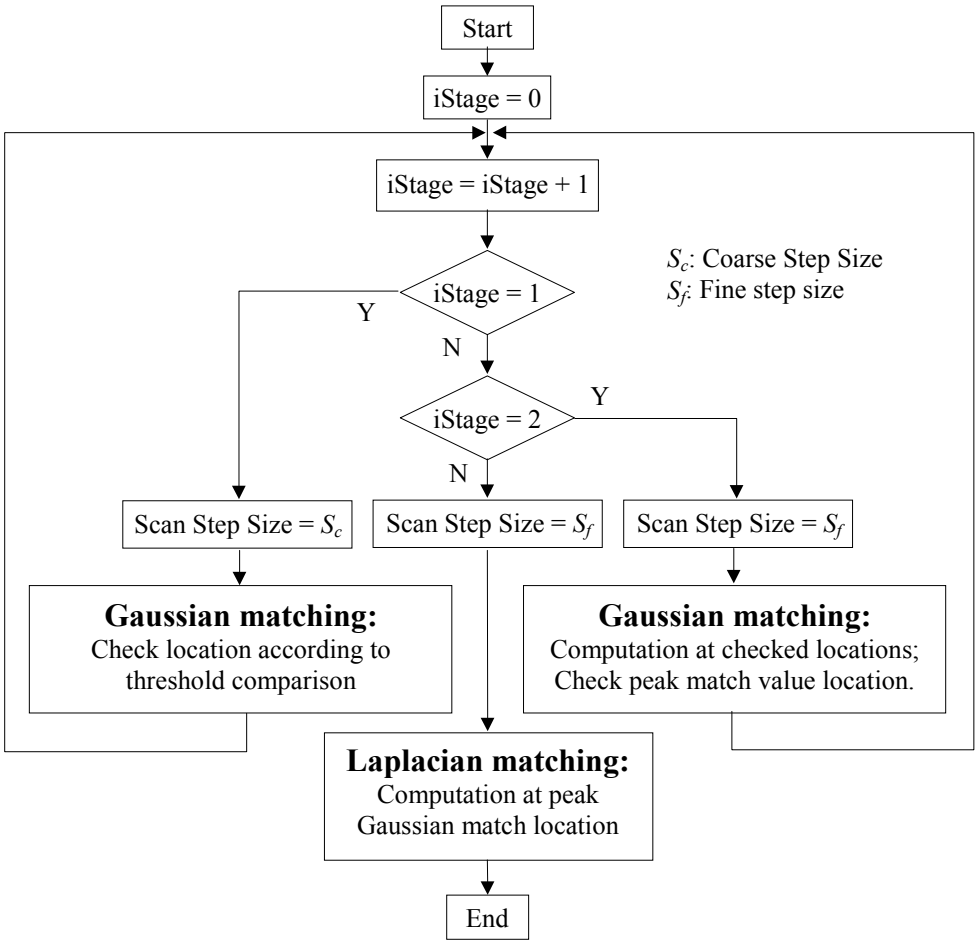


Figure 5-6: Hierarchical (coarse-fine) searching scheme overview.

The process starts with a Gaussian coarse scan (iStage = 1). At this stage it computes the matching value at each coarse scan step S_c and checks interesting location for the next stage according to a threshold comparison. At the second stage (iStage = 2) it performs fine scan (step size S_f) but computes the matching value only within the $\pm S_c/2$ neighbourhood of the time location previously checked "interesting". At the end of the second stage a unique "interesting" time location is checked where the Gaussian statistic gives the maximum matching value. At the last stage (iStage = 3) it performs the same fine scan (step size S_f) but computes the matching value only on the unique checked time location employing the robust Laplacian statistic.

Clearly, the fine scan step size S_f and the coarse scan step size $S_c \geq S_f$ directly affect the computational load of this search scheme. If the scan range is N_s and, at the threshold level, the peak-width of a matching function is N_p, then the computational complexity, in terms of the number of match value computation, can be estimated roughly by (5.34)

$$N_m = \frac{N_s}{S_c} + \frac{N_p}{S_f} \qquad \text{with} \quad S_c < N_p \tag{5.34}$$

Notice that, in general, we have $N_s >> N_p$. Consequently, most match value computations are distributed in coarse scan stage. The condition $S_c < N_p$ should be asserted since $S_c > N_p$ will make the peak detection random. This will affect the overall classification quality. Furthermore, the fine scan step size S_f will also affect the match value: too large S_f will bring the match value down. In Chapter 6 we will discuss the experimental scan step size determination.

The threshold value for Gaussian statistic is obtained, using training database, so as to obtain 0 % false-reject-rate (a secure criterion). Chapter 6 gives the details of its determination.

5.3. Summary

In this chapter, based on likelihood ratio, we have built a new Laplacian distance metric *Delta* taking into account the class variance. It results in a generalised Manhattan norm. In this metric the class vari-

ances is combined into a unique parameter β, called weighting factor, that will be estimated in an overall classification performance optimisation process. The introduction of the weighting factor β allows the metric tuning to fit different noisy situations.

We have explained how the quantization noise, introduced during audience data compression, can be compensated by the "insensitive interval" method.

The problem of scale retrieval (or loudness estimation), arising in the Laplacian distance metric, is resolved by running a distance minimisation process. The minimised distance has been normalised to derive the final Laplacian match value C_{Lap}. This distance minimisation process has been run in a band-wise manner to make the system invariant against the frequency dependent sound power damping.

We have experimentally demonstrated the validity and advantage of using conventional Gaussian template matching. And finally, the effective Gaussian-Laplacian coarse-fine searching mechanism has been constructed.

Chapter 6
System evaluation, results

Two closely related tasks will be considered in this chapter to accomplish the overall classification system design: the decision threshold determination and system performance evaluation. Some important simulation results will be given, including weighting factor β optimisation, influence of audience data compression, parameter determination of studio data compression, determination of the audience data recording duration and matching scan step size determination. The training database construction will equally be briefly described here.

6.1. Decision based on matching certainty coefficient

The matching certainty coefficients or match values, C_{Lap} and C_{Gau} developed in chapter 5, quantify the level of similitude between a test pattern (audience data) and a reference pattern (studio data). However, to decide whether the test pattern contains the reference pattern, we need a threshold match value.

As pointed out in chapter 2, this threshold match value can be obtained by solving a 1-variate (matching value), 2-class (class-1: accept, class-2: reject) supervised classification problem where a training database is indispensable. In many literatures such a problem is often called *binary hypothetic test problem*.

6.1.1. General consideration, tools

Let us consider a generic match value C_m as a random variable with

$$C_m = \begin{cases} C_{Lap} & \text{Laplacian statistic} \\ C_{Gau} & \text{Gaussian statistic} \end{cases}$$

Let c_a and c_r represent the *accept* class and *reject* class respectively. The conditional PDF $p(C_m|c_a)$ and $p(C_m|c_r)$ of class c_a and c_r can be experimentally estimated by running the matching process on the training database where the class labels are known. The resulting estimates of these PDFs may take a form similar to that illustrated in Figure 6-1a where the $p(C_m|c_r)$ curve is plotted upside-down for better illustration. In this figure a threshold value $C_{m,thr}$ of C_m is placed to separate the C_m axis into two subspaces labelled R_a and R_r. We want to make the following decision: if incoming C_m is in the region R_a we class it into class c_a, if C_m is in R_r we class it into c_r. Two error probabilities can be associated with each decision, as the greyed areas *far* or *frr* show. The area *far*, called *false accept rate* (or *false alarm rate* in general decision theory), is the right tail area of $p(C_m|c_r)$ from $C_{m,thr}$. The area *frr*, called *false reject rate* (or *missed detection rate*), is the left tail area of $p(C_m|c_a)$ from $C_{m,thr}$.

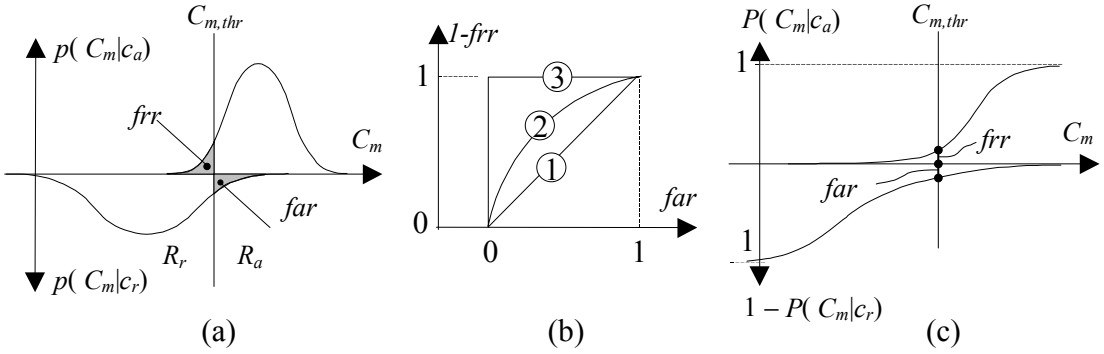


Figure 6-1: ROC curve

Clearly, the mutually related quantities *far* and *frr* are functions of the position of $C_{m,thr}$ and the overlap situation of $p(C_m|c_a)$ and $p(C_m|c_r)$. To reveal these relations, the so-called ROC (Receiver Oper-

ating Characteristic, a term issued from the communication community) curve is often used as shown in Figure 6-1b. In this illustration 3 ROC curves are given and each of them demonstrates a particular $p(C_m|c_a)$ and $p(C_m|c_r)$ overlap situation. The ROC curve 1 is a straight line $1 - frr = far$ corresponding to completely overlapped $p(C_m|c_a)$ and $p(C_m|c_r)$ (identical), therefore the worst situation for a classification task. Curve 3 represents the ideal situation where the two classes are completely separated. In this case it is possible to appropriately place $C_{m,thr}$ to achieve 0-error classification. Curve 2 represents a realistic situation. This ROC curve presentation gives a clear overview of the overall quality of the system but has the drawback that $C_{m,thr}$ is not explicit.

Figure 6-1c shows another ROC representation where the quantities far and frr are presented as explicit functions of $C_{m,thr}$. The two curves $P(C_m|c_a)$ and $1 - P(C_m|c_r)$ are the left tail probability of the accept class c_a and the right tail probability of the reject class c_r respectively. That is, they are derived from

$$P(C_m|c_a) = \int_{-\infty}^{C_m} p(x|c_a)\,dx \quad \text{and} \quad P(C_m|c_r) = \int_{-\infty}^{C_m} p(x|c_r)\,dx$$

It is interesting to note that if we consider the axe C_m being $C_{m,thr}$ these curves represent directly far and frr, that is

$$frr(C_{m,thr}) = P(C_{m,thr}|c_a) \quad \text{and} \quad far(C_{m,thr}) = 1 - P(C_{m,thr}|c_r)$$

For this reason, we call the curves $P(C_m|c_a)$ far curve and $1 - P(C_m|c_r)$ frr curve.

Using Figure 6-1, different decision criteria can be readily stated. Adopting a decision criterion is to place the $C_{m,thr}$ bar somewhere along the axis C_m to achieve a specific goal. For example, if we seek to minimise the total classification error probability $far + frr$, we should place this bar at intersection point of the densities $p(C_m|c_a)$ and $p(C_m|c_r)$. But in general, for different applications, far and frr have different implication or importance.

Until now in this subsection our discussion has not taken into account the class *a priori* probabilities $P(c_a)$ and $P(c_r)$. As we have discussed in chapter 2, these *a priori* probabilities will play the role of weighting $p(C_m|c_a)$ and $p(C_m|c_r)$, therefore also weighting *frr* and *far*, according to the occurrence frequency of the classes c_a and c_r. Unfortunately, it is difficult to obtain these *a priori* probabilities in the audience measurement application. Their estimation will depend on the number of active media emissions, their receivabilities, and above all, they will also dependent on the audience rate. In order to avoid the difficulty of *a priori* probability estimation, we will adopt Neyman-Pearson type decision criterion in the subsequent development because it can make the corresponding decision threshold less sensible to the hidden *a priori* probabilities. We are conscious that by doing so the system will deviate from the MAP criterion (2.4) and will therefore be conducted to a suboptimal system in terms of overall classification error probability. However, the system remains in the general Bayes decision frame [McDo95] [Kay98].

Armed with the tool ROC we can now evaluate the system performance and elaborate the matching threshold value. But before this let us briefly discuss the database construction.

6.1.2. Database construction

The database composition should reflect the real media audience condition. Large bias can lead to an incorrect performance evaluation. Furthermore, the number of entries in the training matching list for each class c_a and c_r plays an important role in the reliability of the classification error estimation. To avoid an excessively detailed discussion we will mention only the essential characteristics of the used database.

Reference signals and noisy audience signals

The selected reference sound signals are:

1. March music;
2. Woman sings German song;
3. Man sings German song;
4. Woman's speech;
5. Woman's and man's speech in group;

6. Man's speech;
7. Hard rock music;
8. Yodel music;
9. Classical orchestral music;
10. Pop and soft rock by a woman;
11. Pop and soft rock by a group;
12. Pop and soft rock by a man;
13. Violin concerto;
14. Xylophone;
15. Toilet flush noise + baby crying.

The number of selected reference data is limited because of the consideration of simulation time during the system design.

The selected disturbances cover a variety of noises and distortions as listed below.

- Additional noise:
  - Non-structured:
    - White noise
    - High speed car motor (5000 cycle/minute)
    - Low speed car motor (1000 cycle/minute)
    - Cloth rubbing the microphone
    - Rain falling onto a metal roof
    - Wind
  - Structured:
    - All the reference sounds used as noises
    - In a crowed bar
    - Noise in a kitchen, i.e. clattering of dishes
- Distortion:
  - Filter:
    - Low-pass filtered
    - High-pass filtered
    - Microphone covered by thick cloth
    - Source sound coming from outdoors with the door closed
  - Other:
    - Harmonic distortion
    - Echo

The audience noisy signals are combinations of the reference signals with the disturbances listed above. The database at our disposal is not made up of exhaustive combinations but only about 480 of them. This number is not enough to give a reliable performance evaluation of our 168-feature[1] classification problem [Theo99]. This lack of training data should be fulfilled by database replications as will be explained later on.

About 60 percent of these combinations are of the additional noise type. And 75 percent of this type is of 0 dB signal-to-noise-ratio (SNR), the others are of various SNR ranging from –3 dB to –10 dB.

The remaining 40 percent of the reference-disturbance combinations are of the distortion type. 70 percent of them are of the filter type, the others are of the harmonic distortion and echo type.

In the original database, signals are sampled at 12 kHz to allow the study on sensor parameter requirements. It is decimated to 3 kHz signals before entering into the classification program.

Matching list

In order to produce the ROC curves of the system, we have to form two kinds of matching lists corresponding to the accept-class c_a and the reject-class c_r.

The list of c_a is a collection of audience-reference-pair specifications where the audience contains the reference. The number of entries for this list is limited to the number of audience signals in the database (480).

For the list of c_r, since it should be filled by audience-reference-pair specifications where the audience does not contain the reference, the number of entries could be $n-1$ times larger than that for the list of c_a, where $n=15$ is the number of reference signals. However, it is not necessary to include all of them (several thousands). Experience shows that 200 of them are enough.

Database replications

The experimental ROC curves have random character due to insufficient size of the database and other system level randomness along the whole computation path conducting to matching values. In order to make

[1] corresponding to 4-second recording duration. see section 6.3.3 for explanation.

the results reliable and interpretable we have to run the algorithm on several replications of the database. The statistics of the results, issued from all these executions, can give more stable indications of some general tendencies.

To this end, 50 replications of the database are produced by taking the successive decimation phases during the 12 kHz to 3 kHz decimation operation. That is, the replications are obtained by a small time shift of $T = 1/12000 = 0.083\,\text{ms}$.

6.1.3. Decision and performance in the Laplacian case

The design goal for the robust Laplacian statistic can be stated in the following way: We seek to construct a system that minimises the false-reject-rate frr under the constraint of ensuring a practical zero false-accept-rate far. That is, at $C_{Lap,thr}$ the far is practically zero. This is a Neyman-Pearson criterion[2]. The decision threshold value $C_{Lap,thr}$ determined under this criterion will be less sensible to the variation of the a priori probability. A "practical zero" is used because of the theoretically unbounded tails of density.

In practice the continuous density function can only be approximated by a discrete sample statistic (histogram) with a limited number of samples. An experimental ROC presentation for the dedicated system is given in Figure 6-2. In order to facilitate the illustration, the far curve is modified so that all points to the left of $C_{Lap,thr} = 0$ are brought to $C_{Lap,thr} = 0$.

[2] Under Neyman-Pearson criterion, we choose a maximum acceptable far and design the system that seeks to minimize frr [McDo95] [Kay98].

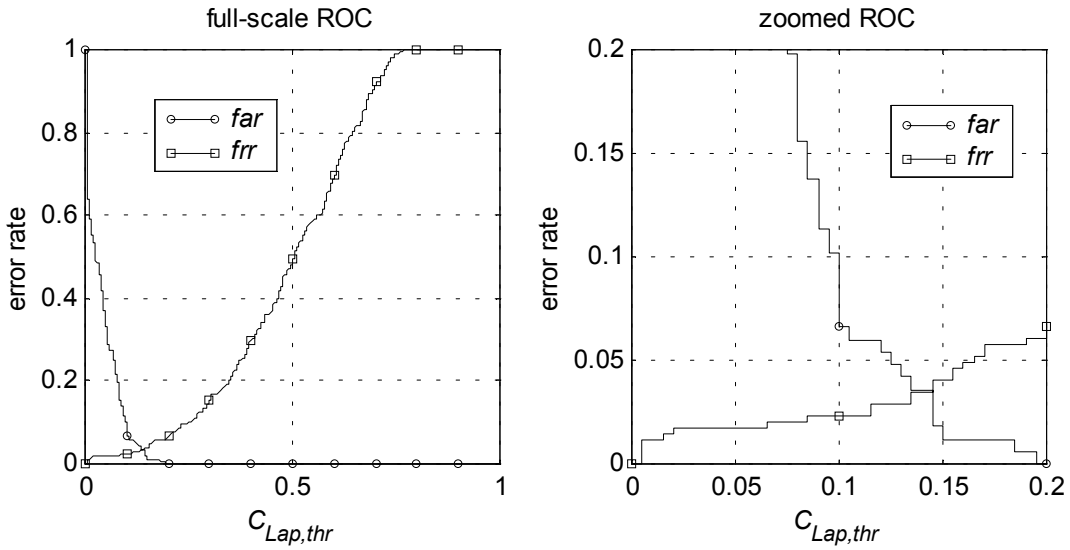


Figure 6-2: Practical ROC plot example

The "practical zero" is solved by rejecting the part of the *far* curve beneath 0.04 then linearly extrapolating the remaining part of the curve as shown in Figure 6-3.

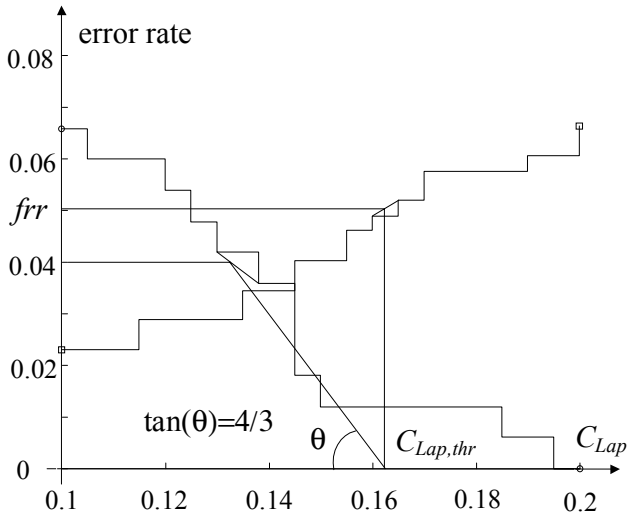


Figure 6-3: Extrapolation of far curve

The slope of the extrapolation line, –4/3, is an experimental choice. It is meant to reflect the average tangent of the *far* curve at 0.04. Because of the discrete nature of the curve, the 0.04 point is obtained by linear interpolation. Finally, the resulting $C_{Lap,thr}$ is used to define the corresponding *frr* value which in turn indicates the system performance.

As pointed out earlier, the resulting statistical quantities (far, frr, $C_{Lap,thr}$) themselves should be treated as random variables. More reliable results should be presented by a statistic of a repeated execution on different database replications. The results of the simulation on 50 database replications are presented in Figure 6-4 for the match value threshold $C_{Lap,thr}$ and in Figure 6-5 for frr indicating the system performance. These simulations are executed using optimised parameter set ($\beta = 0.8$, $\gamma = 0.9$, with quantization error correction).

As can be seen in Figure 6-4, the experimental estimations of $C_{Lap,thr}$ fluctuate around a mean value of about $C_{Lap,thr,mean}$ with a sample standard deviation of about $\sigma_{Lap,thr}$. Their numerical values are given in Table 6-1. In order to observe the influence of this fluctuation to the system performance, three situations are considered for frr estimation as shown in Figure 6-5:

(a) use $C_{Lap,thr,min} = C_{Lap,thr,mean} - \sigma_{Lap,thr}$ as effective $C_{Lap,thr}$;

(b) use $C_{Lap,thr,mean}$;

(c) use $C_{Lap,thr,max} = C_{Lap,thr,mean} + \sigma_{Lap,thr}$.

While a slight performance difference can be observed from this figure, the maximum frr is below 4 percent. The numerical results are given in Table 6-2.

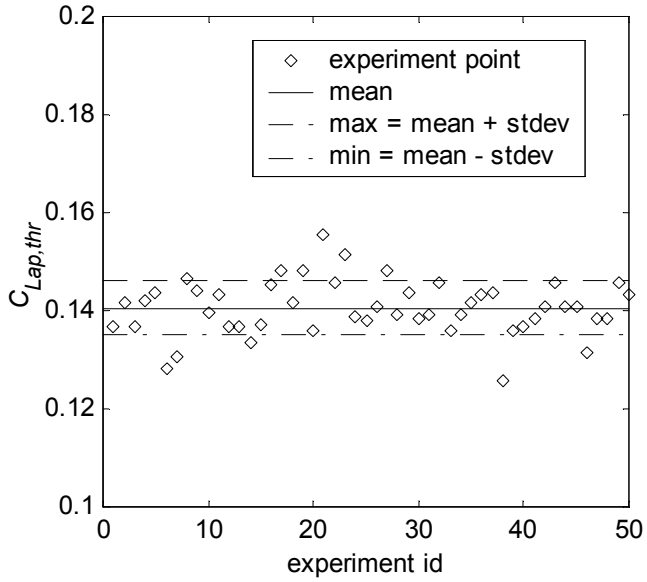


Figure 6-4: Simulation result for match value threshold $C_{Lap,thr}$

Table 6-1: Numerical results for $C_{Lap,thr}$

mean	0.1405
standard deviation	0.0056

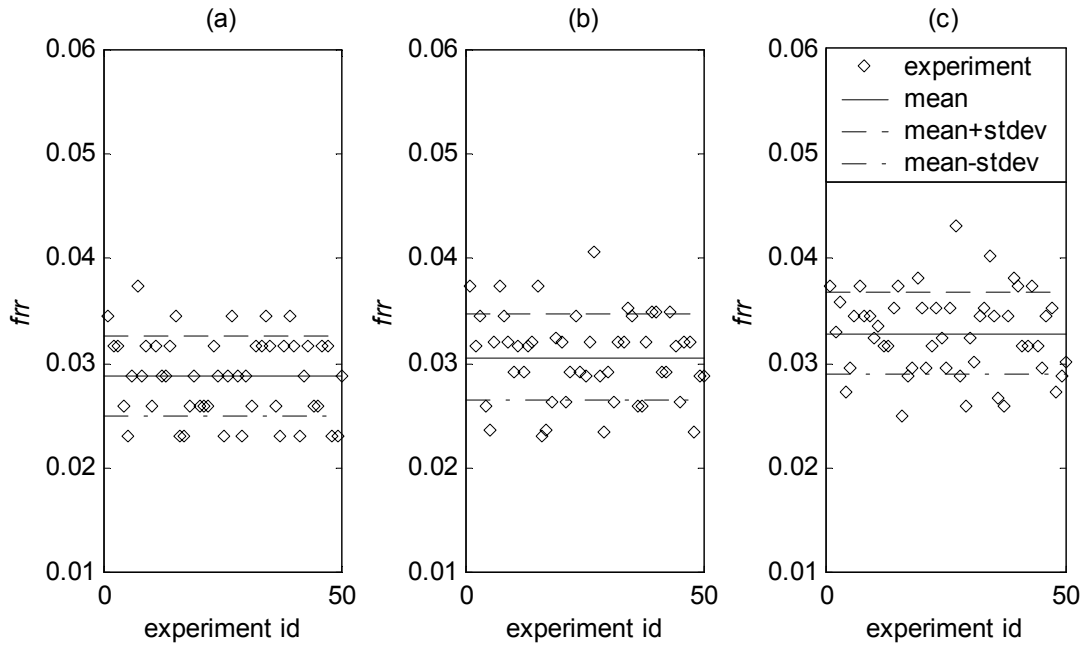


Figure 6-5: Simulation result for frr: (a) use minimum match value threshold, (b) use mean match value threshold, (c) use maximum match value threshold.

Table 6-2: Numerical result for frr

	(a)	(b)	(c)
mean	0.0288	0.0308	0.0328
standard deviation	0.0038	0.0038	0.0040

In the subsequent discussion, if not declared specially, we assume always $C_{Lap,thr,mean}$ to be effective $C_{Lap,thr}$.

6.1.4. Gaussian matching threshold value

As described in Section 5.2.4, in the coarse Gaussian matching scan, we need a threshold value to check the interesting locations. The design goal for this Gaussian statistic is different from the Laplacian one. Our

concern here is to find a threshold match value that ensures a practical zero false-reject-rate *frr*. We ignore the false-accept-rate *far*.

Similar as in the case Laplacian, the "practical zero" *frr* is obtained by rejecting the part of the *frr* curve below 0.04 and linearly extrapolating the remaining part of the curve. The slope of the extrapolation line is 4/10, approximately reflecting the average tangent of the *frr* curve at 0.04. The results of simulation using purely Gaussian statistic (with $\gamma = 0.9$) are collected in Figure 6-6 for the match value threshold $C_{Gau,thr}$.

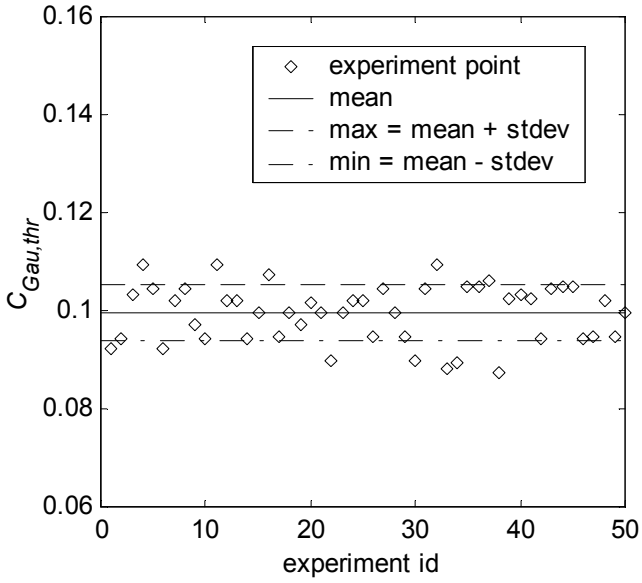


Figure 6-6: Experimental match value threshold for Gaussian coarse searching stage

The numerical results are listed in Table 6-3. The effective $C_{Gau,thr}$ can therefore take 0.08 or even 0.07 to ensure a 0 *frr*.

Table 6-3: Numerical result for $C_{Gau,thr}$

mean $C_{Gau,thr}$	0.0995
$C_{Gau,thr}$ standard deviation	0.0057

6.1.5. Performance comparison

In order to appreciate the advantage of Laplacian statistic over Gaussian statistic, comparative presentations are given in Figure 6-7 and Table 6-4, imposing 0 *far* condition and using a mean match value threshold in both statistic cases. Note that here we discuss on a pure Laplacian statistic and a pure Gaussian statistic, that is, no mixed model coarse-fine searching is involved.

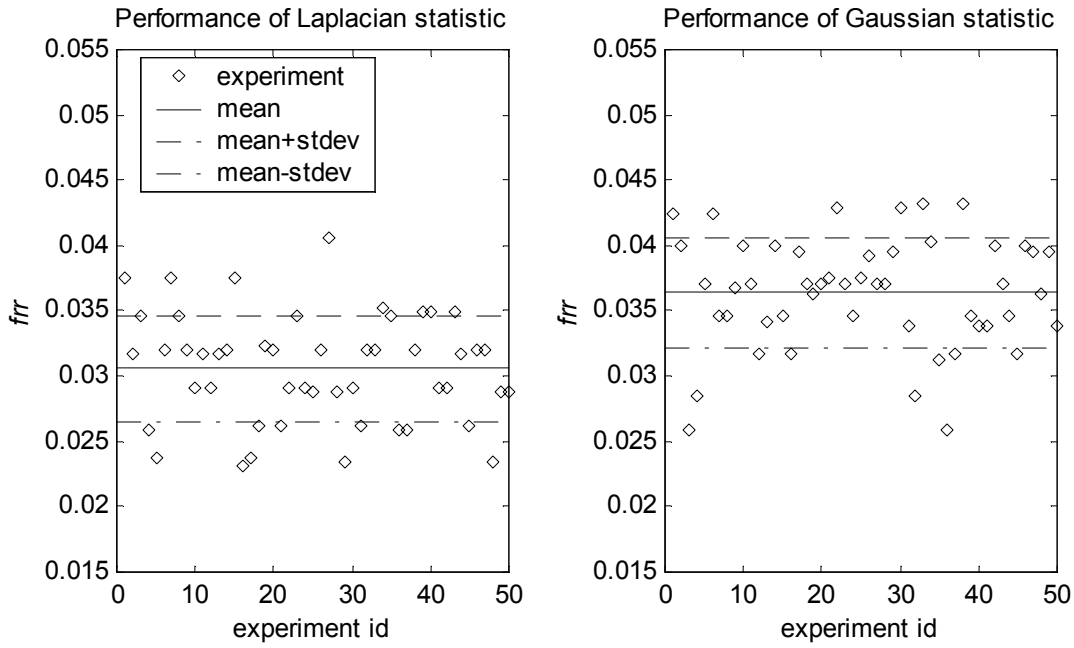


Figure 6-7: Performance comparison between Laplacian statistic and Gaussian statistic

Table 6-4: Numerical comparison between Laplacian and Gaussian statistic

	Gaussian	Laplacian
mean *frr*	0.0364	0.0308
frr standard deviation	0.0042	0.0041

About 16% false rejection is avoided using robust Laplacian statistic over the traditional Gaussian statistic. This is a significant improvement.

6.2. Other simulation results

6.2.1. Influence of the audience data compression and determination on the weighting factor β

As pointed out in Sections 4.1.2 and 5.1.2, the audience data compression by quantization will introduce additional noise and this noise can be handled with the "insensitive interval" method. In this subsection we will see, through some simulation results, how this quantization noise influences the matching process and the adopted noise cancellation method is effective. We will also see that the optimal estimation of the weighting factor β in robust distance measure can be affected by the compression scheme.

The simulations are run under the following compression-compensation cases:

(a) The code-decode stage in the audience data path is skipped, so there is no quantization noise;

(b) The compression is engaged in the audience data path but without any quantization noise compensation at the matching stage;

(c) The audience data are compressed, and quantization noise is compensated using the "insensitive interval" method (Section 5.1.2) at the matching stage.

In (b) and (c) we use $\gamma = 0.9$ for gamma companded quantization. All simulations have been done on the 50 replications of the database to avoid desultory conclusions.

Let us start from presenting the auto matching (see Section 5.2.3 for definition) results to see the pure influence of the quantization noise, caused by the audience data compression, without worrying about the influence of other noises. These simulation results, in terms of match value, are illustrated in Figure 6-8. The common weighting factor is set to $\beta = 0.8$ for all three compression-compensation schemes. In Figure 6-8 the selected signals are the reference signals from the training database. The signal id numbers are the same as in Section 6.1.2.

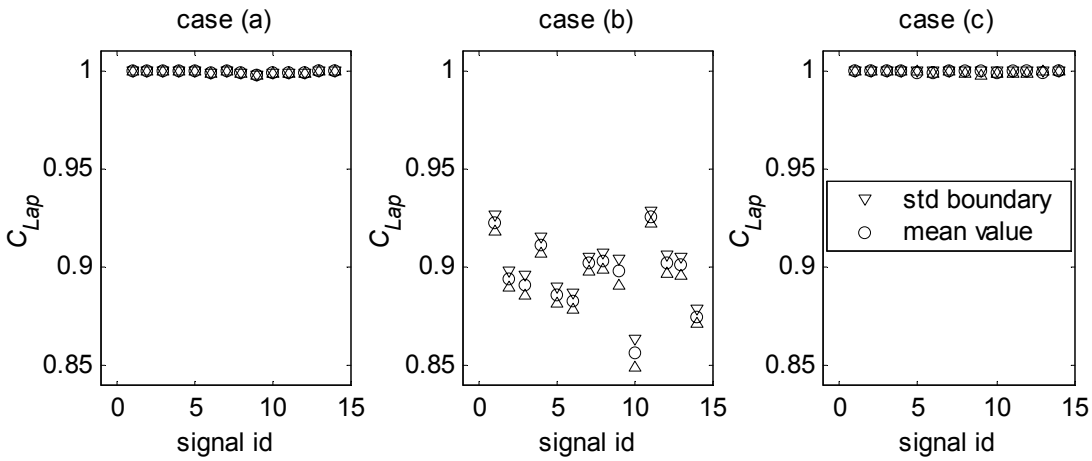


Figure 6-8: Influence of audience data compression to auto-match value.

In case of cross matching (see Section 5.2.3 for definition), the situation is messy. We have to use ROC to obtain general tendency information. In order to reveal the influence of the compression quantization to the weighting factor optimisation, the results of simulation are presented in *frr* versus β format as shown in Figure 6-9. The corresponding $C_{Lap,thr}$ versus β results is presented in Figure 6-10.

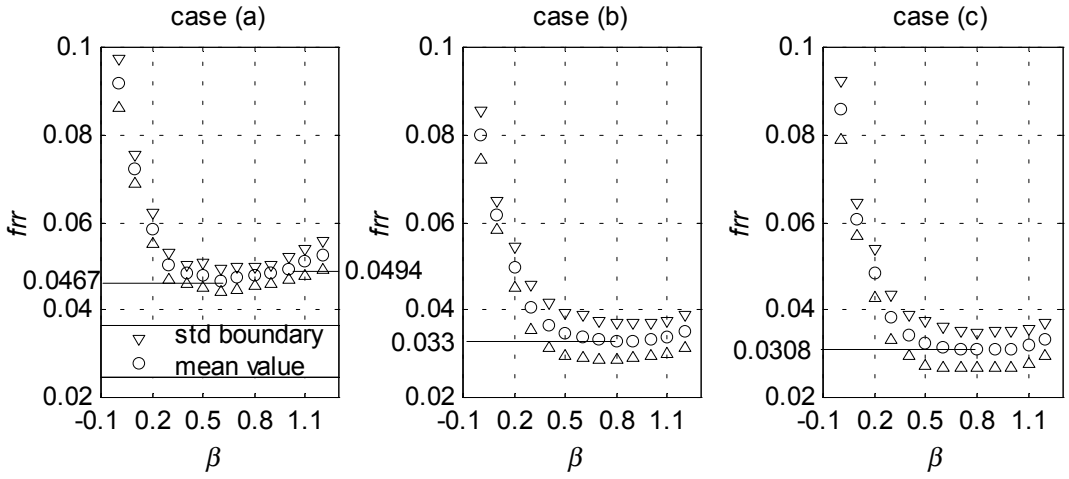


Figure 6-9: Optimal β estimation.

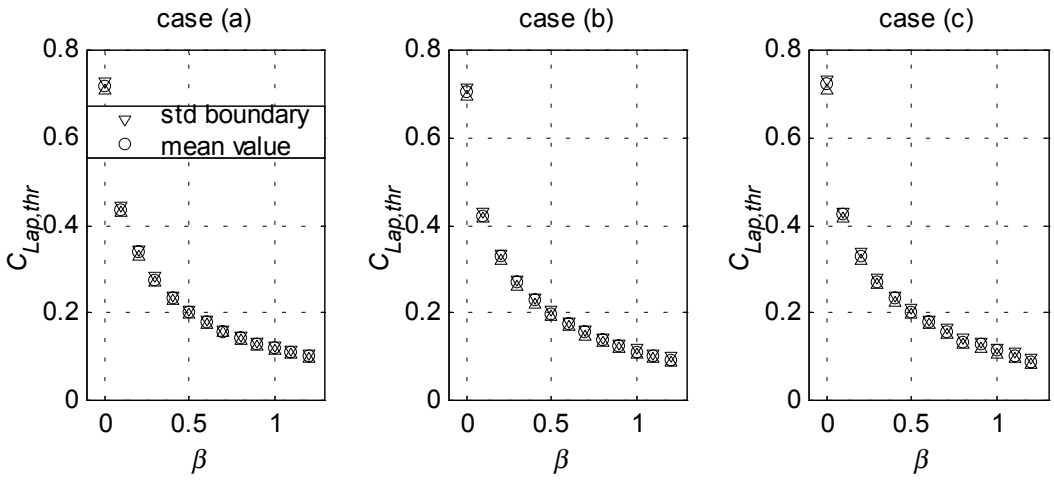


Figure 6-10: Corresponding threshold match value $C_{Lap,thr}$ versus β

First of all, we remark from Figure 6-9 that, in all cases (a), (b) and (c), the *frr* - β curve shows a clear and unique local minimum. This means that an optimal β exists and corresponds to the β minimising *frr*.

Comparing case (a) and (b) of Figure 6-9, we remark that the *frr* - β curve has a general "surprising" shift, vertically towards 0, representing a 30 % classification error reduction! This means that the quantization in audience data compression has a positive effect on the class separability: while this quantization introduces the quantization noise to the feature vector, it also eliminates the other noises presented in lower amplitude. This is a further argument in favour of the quantization method in the selection of the audience data compression scheme. Another remarkable change of the *frr* - β curve from case (a) to case (b) is the optimal β having shifted to the right. This means that σ_1^2, the variance of the class under test, is statistically reduced. These two remarkable changes are consistent.

From case (b) to (c) of Figure 6-9, we observe a further improvement of the classification quality due to the quantization error compensation. The classification error is reduced by about 7 %.

We remark also, from (a) to (b) and (c) the *frr* - β curve become flatter near its minimum. This is a desirable result because the system becomes less sensible to the mismatch of β.

The fact that the $C_{Lap,thr}$ - β curve remains practically unchanged over the three cases of Figure 6-10 indicates that the influence of audience data compression and quantization compensation occurs mainly in the accept class represented by *frr* curve of ROC.

In spite of the relation (5.17), requiring $\beta \in (0,1)$, We have added the simulation points with $\beta = 1.1$ and $\beta = 1.2$ to show the variation tendency of the *frr* - β curve at the point $\beta = 1.0$.

6.2.2. Studio data compression parameters

In section 4.3 we have seen the possibility of compressing the studio data using the decimation-interpolation method. Here, we present the related simulation results providing useful information for the parameter determination. As shown in Figure 6-11, the results are presented in *frr* versus design parameter-set ID format. Table 6-5 gives the meaning of these parameter-set ID.

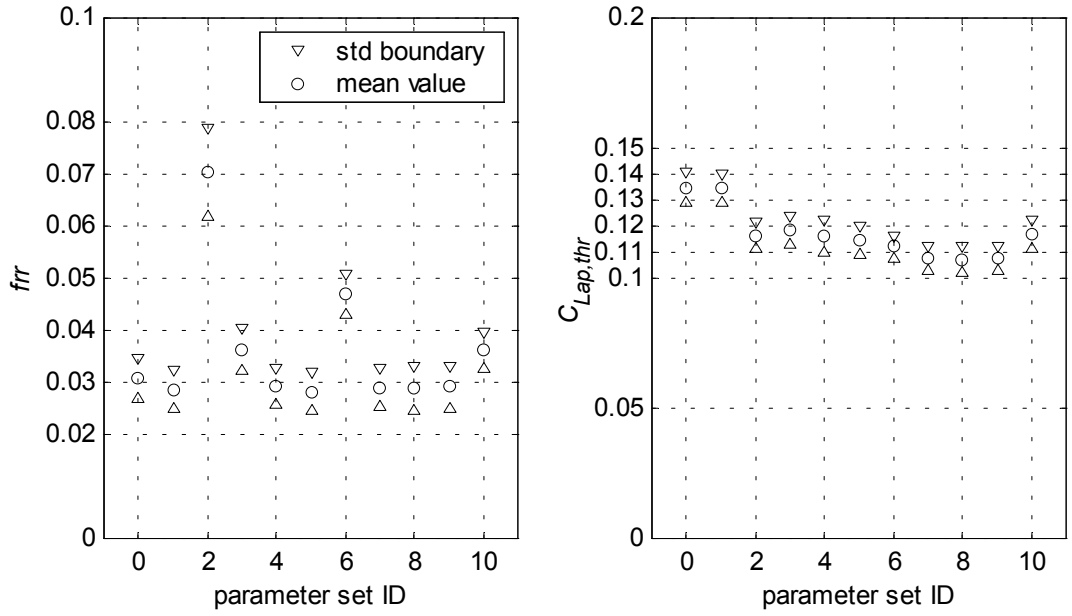


Figure 6-11: System performance versus studio data compression scheme

Table 6-5: Studio data compression parameter set

p.s. ID	Comment
0	$lp2$, $lp3$, $lp4$ all use 48-tap trapezium, $F=1$
1	$lp2$, $lp3$, $lp4$ all use 48-tap boxcar; $F=1$
2	$lp2$, $lp3$, $lp4$ all use 48-tap boxcar; $F=48$
3	$lp2$, $lp3$, $lp4$ all use 48-tap boxcar; $F=24$
4	$lp2$, $lp3$, $lp4$ all use 48-tap boxcar; $F=12$
5	$lp2$, $lp3$, $lp4$ all use 48-tap boxcar; $F=6$
6	$lp2$ uses 48-tap boxcar, $lp3$, $lp4$ use 96-tap hamming; $F=48$
7	$lp2$ uses 48-tap boxcar, $lp3$, $lp4$ use 96-tap hamming; $F=24$
8	$lp2$ uses 48-tap boxcar, $lp3$, $lp4$ use 96-tap hamming; $F=12$
9	$lp2$ uses 48-tap boxcar, $lp3$, $lp4$ use 96-tap hamming; $F=6$
10	$lp2$, $lp3$, $lp4$ all use 96-tap hamming; $F=48$

For the meaning of $lp2$, $lp3$, $lp4$ and F please refer to Section 4.2.

These simulation results confirm the feasibility of the studio data compression by decimation and interpolation. We can obtain compression rate of up to at least 24 (ID 7) without additional computational complexity in the feature extraction program of the portable audience recording unit.

Comparing the results for different parameter-sets indicates that the system is sensible to the additional aliasing and imaging noise at the studio data side during this studio data compression.

6.2.3. Determination of portable unit recording duration

In section 3.4.3 we have mentioned that feature vector length is directly related to the portable unit recording duration. The following simulation results, illustrated in Figure 6-12, give some hints for this recording duration determination. As can be seen, 4-second recording duration gives quite a good performance and the system starts to saturate thereupon. 4-second recording is used in all simulations (except the one in Figure 6-12) presented in this thesis and it is also 4-second recording that is used in the actual implementation.

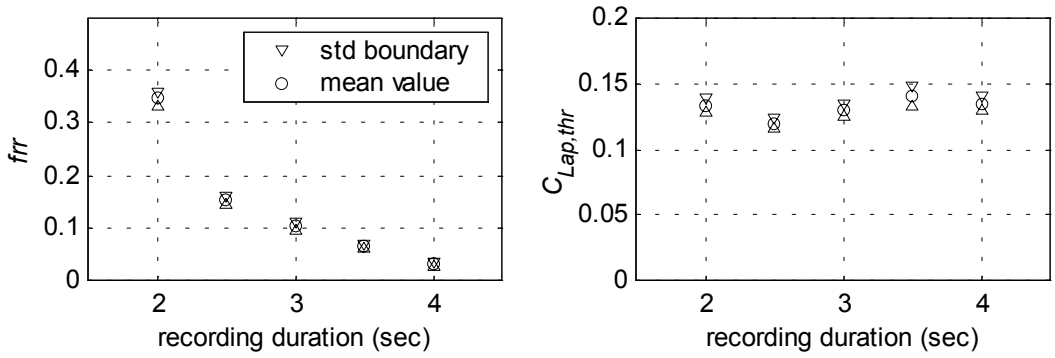


Figure 6-12: System performance versus portable unit recording duration

Further increase of performance by enlarging the recording duration is possible but has to be payed by computational complexity and data storage.

4-second recording duration corresponds to a 168-feature feature vector after the feature generation using the proposed scheme.

6.2.4. Matching scan step size determination

In section 5.3.4 we have discussed the coarse-fine mixed model searching scan method and pointed out that the coarse scan step size S_c and the fine scan step size S_f play an important roles for the matching quality and computational complexity. In this subsection we will present the experimental determination of these scan step sizes.

Let us start our discussion with the fine scan step size S_f. The matching scan can be viewed as a down sampling process applied to the matching function with the scan step size as down sampling factor. The spectrum analysis of a typical matching function shows that a matching function is a low-pass signal (Figure 6-13). Its band width occupies only about 1/3 of the entire band, i.e., it is about 3 times over-sampled. This means that, without match value degradation, S_f can take up to 3 corresponding to $3 \cdot T_{ch}$ in time resolution, where T_{ch} is the channel signal time resolution at the end of band splitting (see Section 3.2.2).

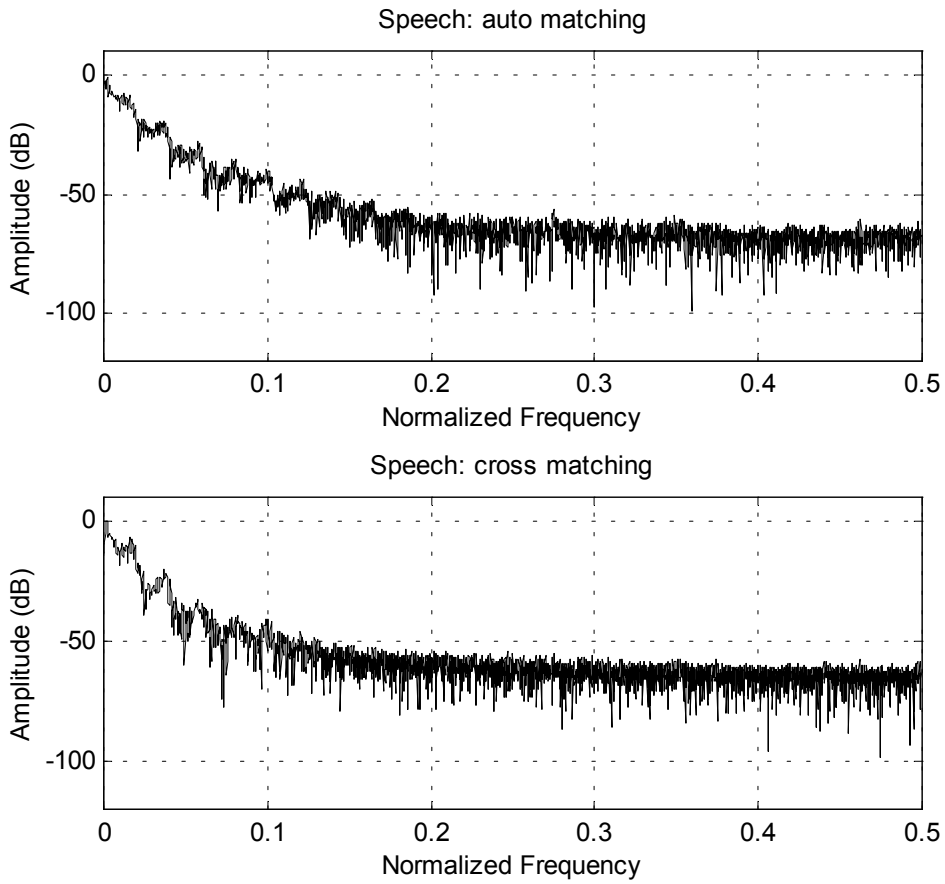


Figure 6-13: Frequency response of matching function

The coarse scan step size S_c is mainly determined by the peak-width of the matching function (Figure 5-9). However, exhaustive observation of matching functions is highly impractical. Once again, we have to resort to ROC statistic.

The first simulation fixes $S_f = 1$ and generates the *frr* values versus S_c ranging from 2 to 52 (to be large enough). The results are plotted out in Figure 6-14 (a).

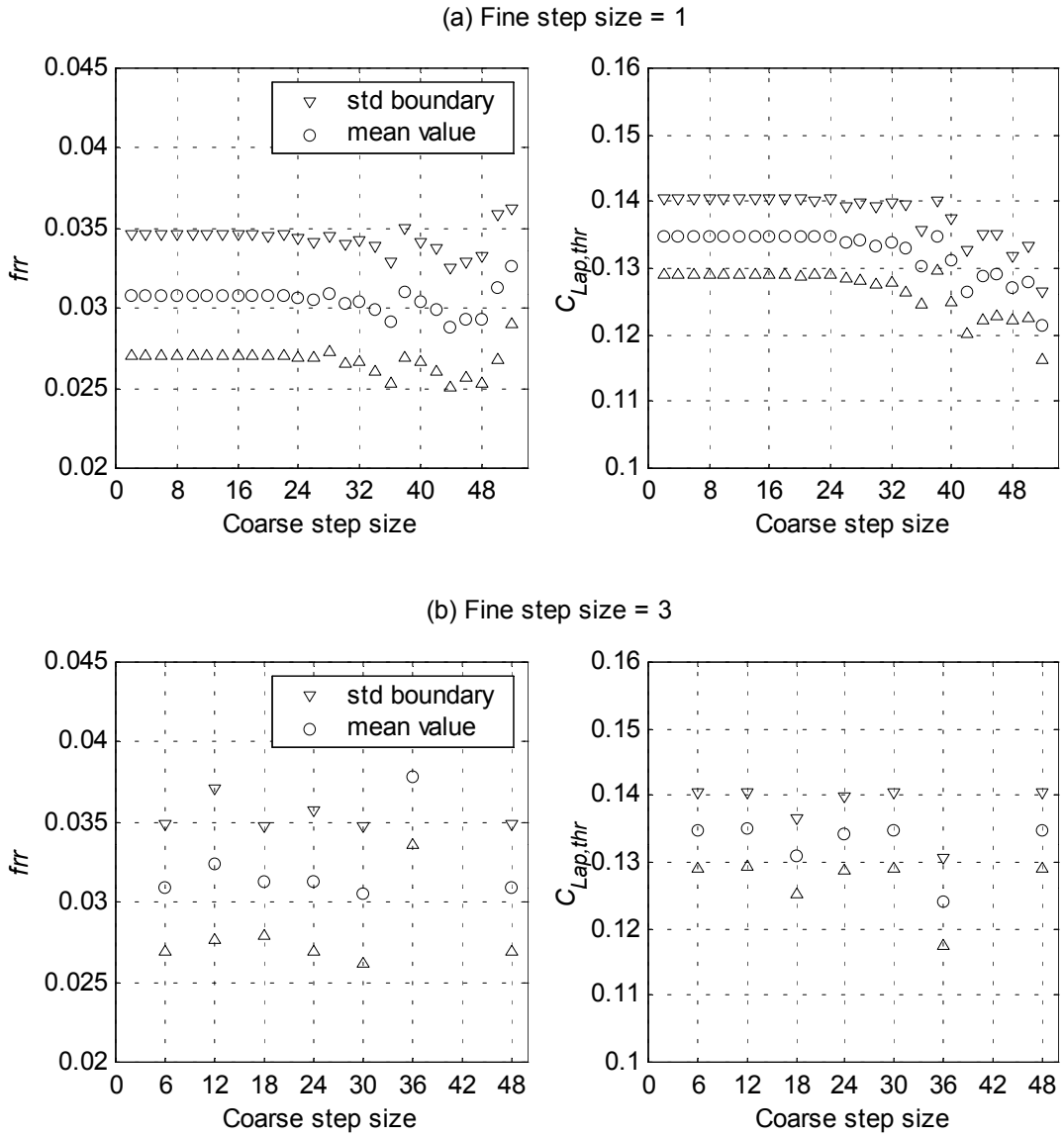


Figure 6-14: frr and $C_{Lap,thr}$ versus coarse scan step size.

As can be seen, the mean frr value remains constant over the interval [2, 24] and from then on it starts to swing due to random peak detection as explained in section 5.3.4. This oscillation makes the classification quality non-predictable and should thus be avoided. The maximum S_c before this oscillation roughly corresponds to one half of the peak-width of the matching function from the observation made in Chapter 5.

In order to reveal an eventual inter-dependency between S_c and S_f, we produce the same plot but with $S_f = 3$. The simulation results are shown in Figure 6-14 (b). We note, from this figure, the similar phenomenon as in Figure 6-14 (a) except that the oscillation starts earlier.

What we can conclude from this second simulation is that the discussion about maximum S_f being 3 holds but increasing S_f enhances the randomness of peak detection.

To conclude this subsection, we would like to point out that it is more judicious to keep S_f being 1 and increase S_c as long as the application can tolerate the corresponding classification error. This is because, from expression (5.33) and Figure 5-9, most match value computations are located in the coarse scan stage.

6.2.5. Influence of the sqrt function

In Section 3.4.2 we have mentioned that a square root operation applied on the smoothed channel energy can improve the matching quality. Figure 6-15 gives the simulation result favouring this proposition.

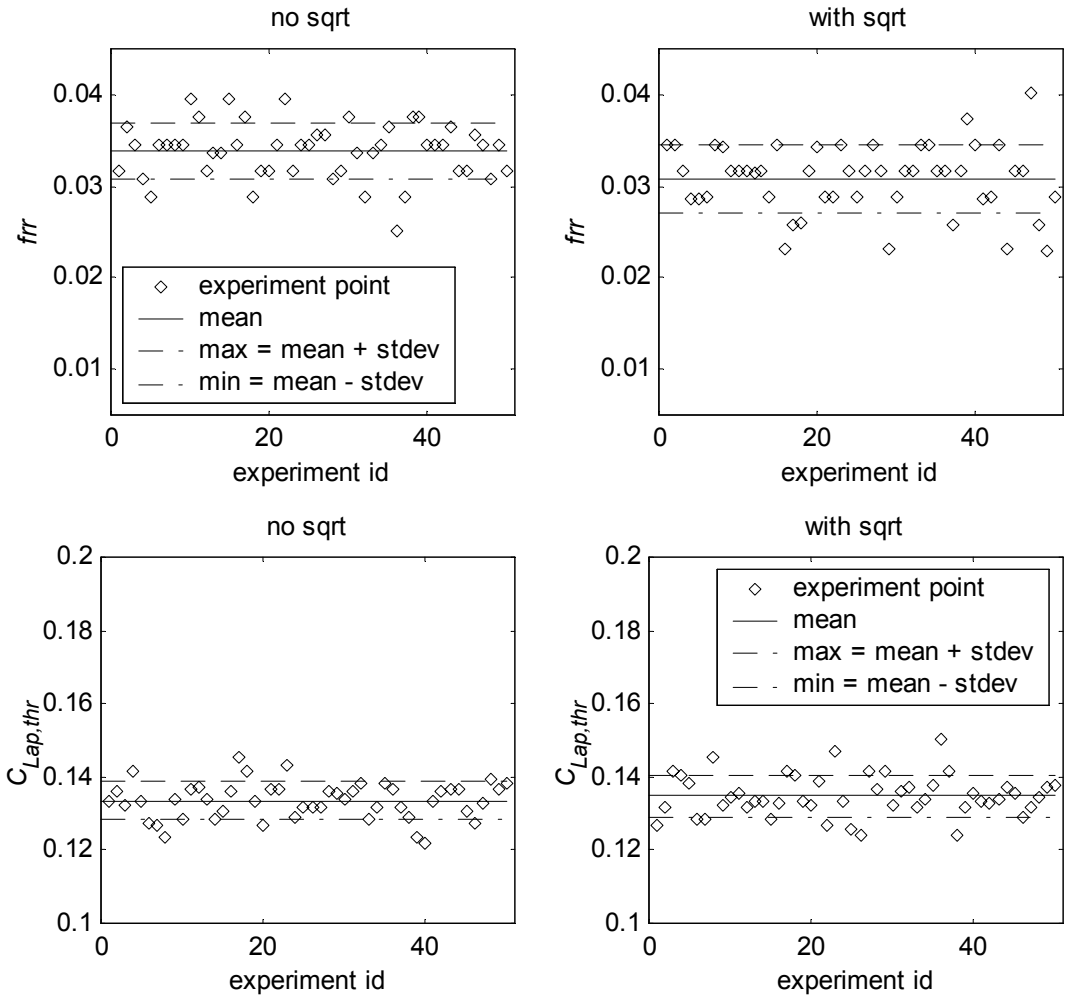


Figure 6-15: Influence of sqrt function on frr and $C_{Lap,thr}$

The improvement due to this operation is about 10 % *frr*. Unfortunately, we have nothing more convincing than this experimental result to support the use of the square root function. Perhaps, it contributes to the reduction of the noise distribution mismatch.

6.3. Summary

In this chapter we have shown that the system performance evaluation and matching threshold determination can be accomplished jointly by solving a 1-variate, 2-class supervised classification problem using ROC tool.

We have demonstrated, by simulation, that the proposed robust Laplacian template matching approach can achieve the audience measurement with a false rejection error rate of $3.08^{\pm 0.4}$ % if we place the match value threshold so as to ensure ~0 % false accept error rate. Compared to the conventional Gaussian template matching, 16 % error reduction is obtained.

In this chapter we have also presented other important simulation results that have contributed to the system design. Namely the following:

- The simulation has shown a positive effect of the audience data quantization on the matching quality: 30 % error reduction has been observed!
- The quantisation noise due to audience data compression is effectively compensated by the proposed "insensitive interval" method. 7 % error reduction is realised.
- The studio data can be compressed by a factor of up to 24 without degradation of the system performance and without additional complexity.
- In the coarse-fine searching process, the coarse scan step size can be increased up to 24 if the fine step size remains at 1.

Chapter 7
Implementation considerations

This chapter discusses some important implementation issues such as multirate technique for decimation and interpolation in feature extraction, real time implementation in portable audience recording units, matching process control and special considerations on the scale retrieval operation.

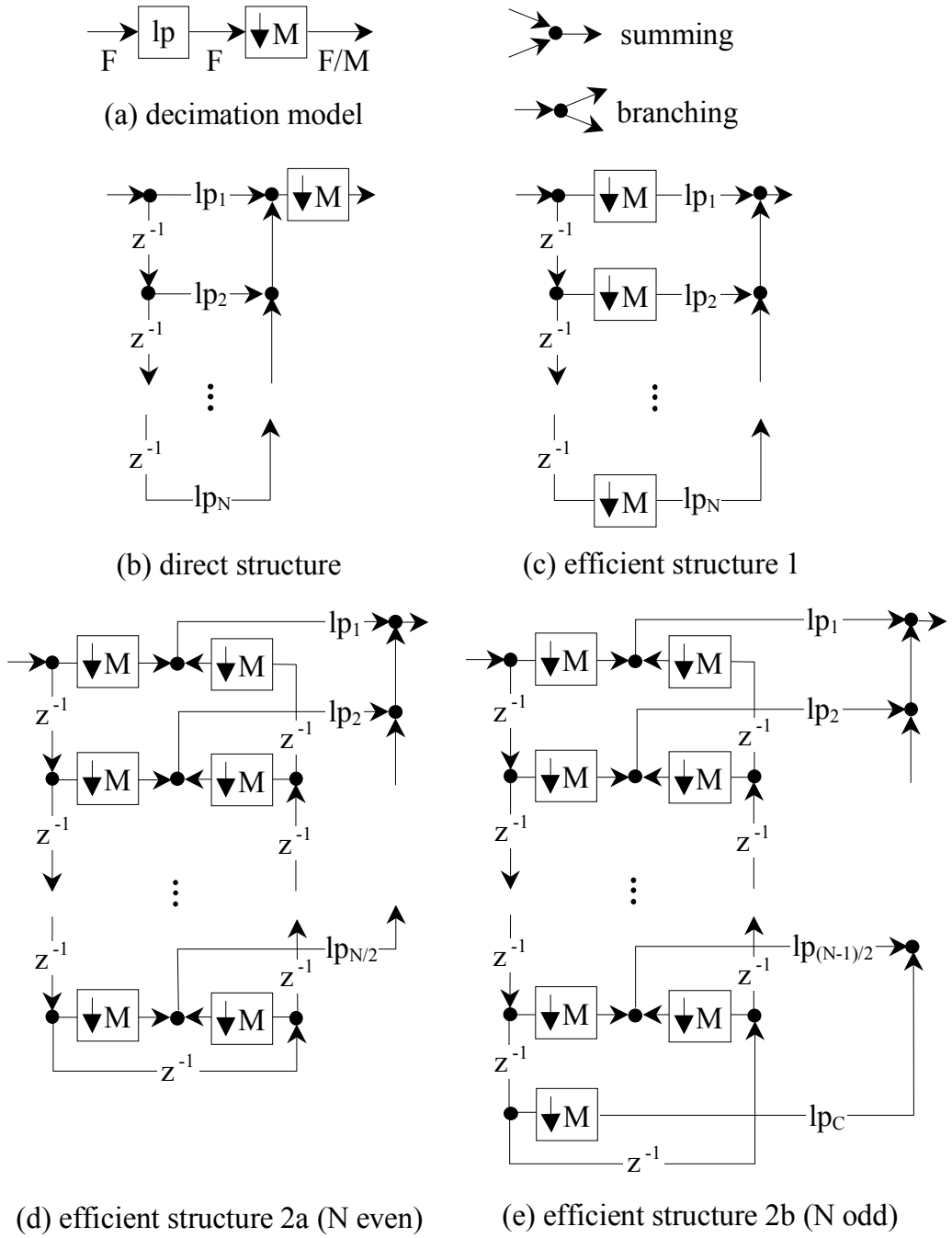
7.1. Feature extraction implementation

7.1.1. Decimation with multirate techniques

The decimations discussed in chapter 3 (Figure 3-7 and Figure 3-9) are computationally inefficient since the filtering operations are performed on the high sampling rate side. Using well-known multirate techniques, this situation can be greatly improved. Let us give three relevant structures for efficient decimation implementation as shown in Figure 7-1. Similar techniques also exist for interpolation [Croc83].

The decimation model (Figure 7-1a), called *decimator*, is composed from a filter lp and a down sampler by M. It converts the input sampling rate from F to F/M. If this model is implemented directly it will result in the structure shown in Figure 7-1b, where lp_i (i = 1,2,…,N) represents multiplication by i^{th} filter coefficient lp_i, z^{-1} represents delaying of one sample, the small solid circle at each node of the networks represents summing or branching as shown at the top of the figure. This structure is

inefficient because M-1 of every M output samples of the filter lp are computed to no avail.



(a) decimation model

(b) direct structure

(c) efficient structure 1

(d) efficient structure 2a (N even)

(e) efficient structure 2b (N odd)

Figure 7-1 Efficient decimation structure from multirate signal processing theory

According to network topological equivalence the down sampler can be moved so that the structure in Figure 7-1c is obtained. This change makes the structure efficient by a factor of M in computational

complexity, since it computes only the necessary output of filter lp. If the impulse response of the filter lp possesses a symmetry, we can use the structures given in Figure 7-1d or Figure 7-1e, according to N being even or odd respectively, to earn a further factor of 2 in computational complexity.

The structure of Figure 7-1e is suitable for our dyadic tree filter bank of Figure 3-7. Applying this structure to *lp*1 decimation with M = 2 and removing the zero coefficient branches, we obtain the efficient structure for *lp*1 decimation as shown in Figure 7-2.

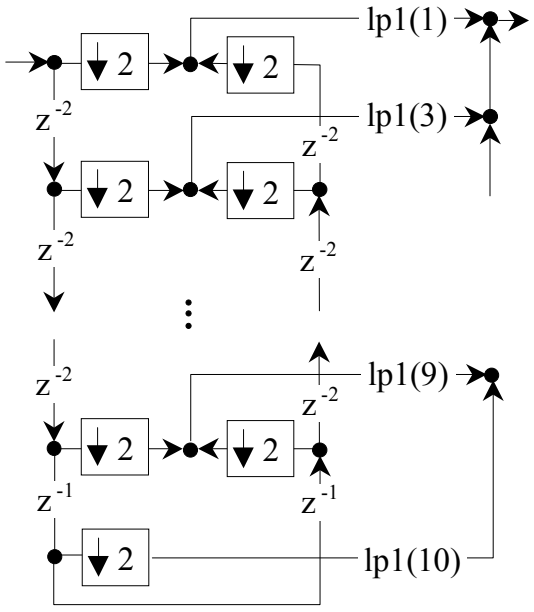


*Figure 7-2 Efficient structure for lp*1 *decimation*

For *hp*1 decimation, since its coefficients are a negative version of those of *lp*1 except for the central one which is identical to that of *lp*1, we need merely change lp1(10) to –lp1(10) in Figure 7-2 in order to obtain a negative version of the output of *hp*1. The sign does not pose any problem since the next operation is squaring the samples for energy extraction. Concretely, we first compute the part of lp1(1) to lp1(9) once, then we add respectively the lp1(10) term and the – lp1(10) term in order to obtain the output of *lp*1 and the negative version of the output of *hp*1.

For the *lp*2 decimation of Figure 3-9, we can directly use the structure of Figure 7-1d with M = N = 48.

7.1.2. Real time implementation in portable audience recording units

In portable audience recording units, for one recording session, a huge amount of data (12000 samples for 4-second recording) will be injected into the feature extraction algorithm. This can bring about a troublesome situation for RAM size. A possible solution to this problem is the use of real time processing. In fact, the problem is accentuated only in the subband decomposition and energy extraction. The data will be dramatically reduced after these operations.

7.1.3. Processor related optimisation

With the help of algorithmic level optimisation (feature generation scheme, filter choice, efficient multirate implementation, etc.), a huge amount of computational complexity can be saved. However, a first real-time execution of the feature extraction algorithm on the selected DSP processor (TMS320-C50 16-bit fixed point DSP from Texas Instrument) required as much as 12 MIPS in computation power. This poses a serious problem when choosing a commercial small sized battery for a wristwatch. Hence a processor related optimisation is desirable.

The implementation experiment of the dedicated algorithm showed that the processor related optimisation can further reduce the computational complexity by a factor of up to 6. This optimisation can be realised in the following sensible aspects:

- Subroutine call: the less the better because of the context save and restore;
- Memory allocation: fixed-size table rather than dynamic allocation by malloc() etc. because of addressing and subroutine call;
- Loop: the less the better notably inner loop because of the need of multi-cycle instruction for managing a loop;
- Parameter of subroutine: best without any because of the need for special management of stack memory;
- Variable type: float or double should be avoided because of special multi-cycle instruction for floating-point arithmetic on fixed-point DSP.

Similar optimisation can also be made for the matching process on the target processor.

In the real time execution the arrival of an incoming sample is signalised to the program by interruption. The rigid constraint of the real time implementation is that corresponding computations should be terminated between two such adjacent signalizations. Otherwise, some of the incoming data may be discarded. The implementation experiment shows that if the program processes the data in a group by group manner, the code can be arranged to run more efficiently. The group size is fixed to 8 in our implementation. This is managed by a circularly updated buffer.

7.2. Matching process implementation

7.2.1. Time relations

The time relation between a audience data frame and a studio data frame is an important aspect for implementation due to the time stamp error. In this subsection we will elucidate this. To this end, we first define the following quantities:

- Tp : nominal audience recording start time;
- Ts : nominal studio recording time;
- TEp : audience recording time stamp error;
- TEs : studio recording time stamp error;
- TWp : audience recording duration;
- TWs : studio recording duration;

where we should have, as pointed out in section 5.2.1, $TWp < TWs$ in order to allow the compensation of time stamp error by a matching scan. Figure 7-3 illustrates the time relation between a audience data frame and a studio data frame, viewed from the absolute physical time reference.

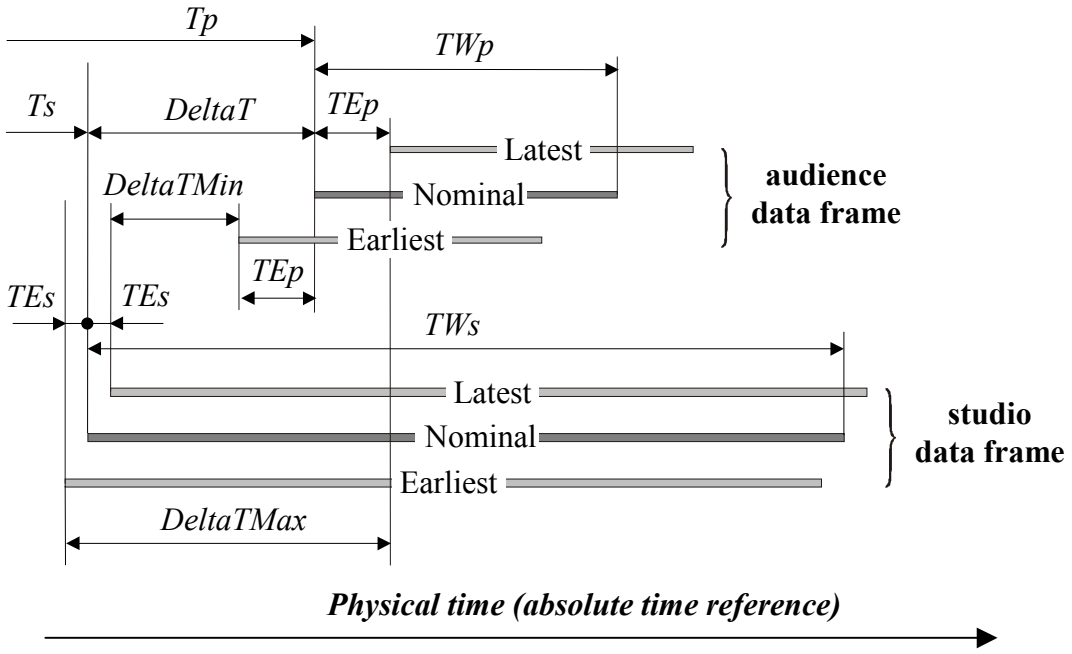


Figure 7-3: Time relation of data frames viewed from absolute time reference

Let us focus on the time difference between these two data frames. Its nominal value $DeltaT$, minimum value $DeltaTMin$ and maximum value $DeltaTMax$ can be computed as follows.

$$DeltaT = Tp - Ts \tag{7.1}$$

$$DeltaTMin = (Tp - TEp) - (Ts + TEs) = DeltaT - (TEp + TEs) \tag{7.2}$$

$$DeltaTMax = (Tp + TEp) - (Ts - TEs) = DeltaT + (TEp + TEs) \tag{7.3}$$

These two last quantities determine the necessity of searching and the size of the searching range. To understand this more clearly, let us observe the time relation in the relative time reference, that is, the studio data frame, as shown in Figure 7-4. Notice that the difference between the earliest and latest audience frame is enlarged, compared with that in absolute time reference, due to studio time error TEs.

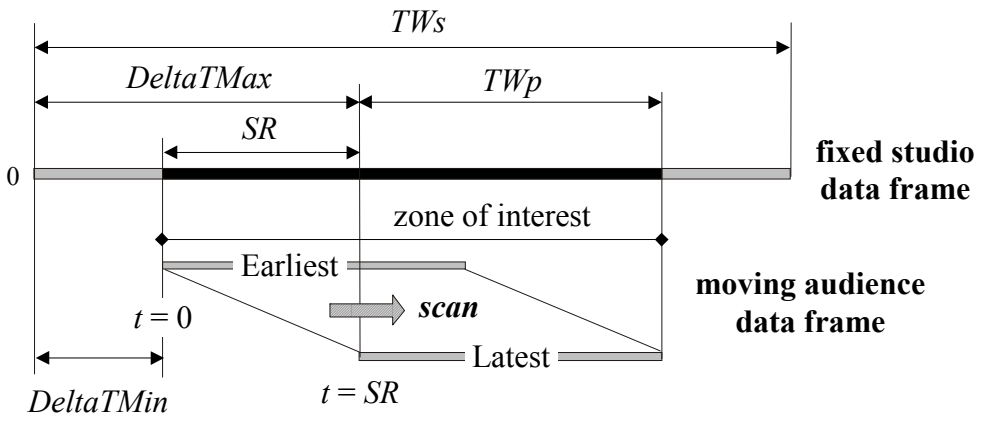


Figure 7-4: Time relation of data frames viewed from relative time reference

The scan range SR in Figure 7-4 can be calculated by

$$SR = DeltaTMax - DeltaTMin = 2 \cdot (TEp + TEs) \tag{7.4}$$

This time relation image clarifies the scan mechanism and different constraints can be deduced in order to make a valid data set and searching scan.

If $DeltaTMin < 0$, we can assume a poor accuracy of the clock device calling for an increase in the ratio TWs / TWp, that is, for more margin in the studio recording duration to absorb this inaccuracy. However, this is not a crucial condition to run the matching process because there is still chance that the audience record is contained in the studio record. So in this case we can force $DeltaMin = 0$.

A similar reasoning can be made for the case where $DeltaTMax + TWp > TWs$, that is, the scan may run out of range. In this case, we can force $DeltaTMax = TWs - TWp$. Nevertheless, in all cases, the relation $DeltaTMax \geq DeltaTMin \geq 0$ should be asserted.

7.2.2. Time resolution

The final feature vector extracted as described in chapter 3 determines the searching time resolution of the matching process (Figure 7-5).

It must be recalled that the finest distinguishable time domain structure is limited by the energy averaging time $T_{av} = 128$ ms (section 3.3.1) and the finest searching scan resolution is determined by the chan-

nel signal time resolution $T_{ch} = 2.67$ ms (section 3.2). The data presented to the matching process, for a *K*-sample audience data at an arbitrary scan position, can be illustrated as in Figure 7-5.

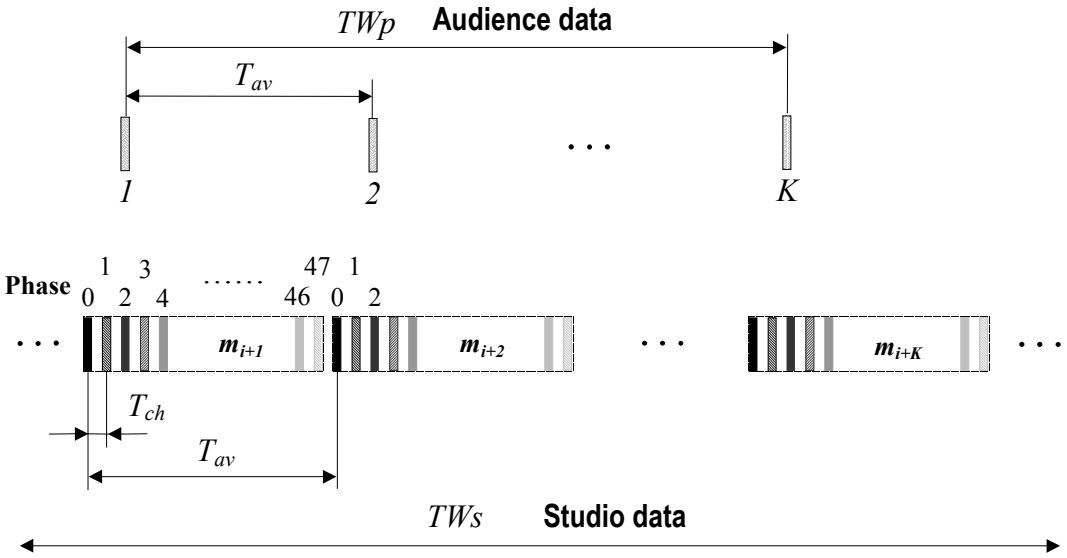


Figure 7-5: Time resolution and decimation phase relation

As can be seen the studio data is presented in a higher time resolution allowing finer searching scan. As explained in section 3.3.2, this higher resolution for studio data is obtained by keeping all decimation phase of energy smoothing decimation. To emphasize this fact, we have illustrated, in Figure 7-5, the studio data train in a grouped sub-frame manner and indicated these sub-frames by the index $\boldsymbol{m}_i$. Thus, while the scan step size can take values as fine as T_{ch}, the matching distance should be computed using a down-sampled studio data. The down sample factor is obviously $M_{sm} = 48$.

7.2.3. Scan control registers map

Let us have a look at the matching process control through a numerical example.

With 1 ppm (part per million) watch precision 1 week stand-alone functioning the *TEp* can go up to 0.6048 second in the worst case. Assuming the same precision for the station, and $DeltaT = 3\,\mathrm{s}$, the related scan control quantities become (see Figure 7-4): $DeltaTMin = 1.7904\,\mathrm{s}$ and $DeltaTMax = 4.2096\,\mathrm{s}$.

The equivalent quantities in terms of sample index:

$$iMin = \text{floor}(DeltaTMin / T_{ch}) = 671$$
$$iMax = \text{ceil}(DeltaTMax / T_{ch}) = 1579$$

At the first stage of the coarse-fine search, the scan boundaries $iMin_c$ and $iMax_c$ should be rounded according to the coarse scan step size S_c (7.5). In the second and third stage, these boundaries must ensure the coverage of the interval $(iMin, iMax)$ and the relation with the coarse scan (7.6). In the cases where the finest scan step size $S_f = 1$ and the coarse scan step size $S_c = 4$, we have, for stage 1

$$\begin{aligned} iMin_c &= \text{ceil}(iMin / S_c) \cdot S_c - \text{floor}(S_c / 2) = 670 \\ iMax_c &= \text{ceil}(iMax / S_c) \cdot S_c - \text{floor}(S_c / 2) = 1578 \end{aligned} \tag{7.5}$$

but for stage 2 and 3

$$\begin{aligned} iMin_f &= \text{ceil}(iMin / S_f) \cdot S_f = 671 \\ iMax_f &= \text{floor}(iMax / S_f) \cdot S_f = 1579 \end{aligned} \tag{7.6}$$

To ensure correct scan control, using the rounding and alignment formulas (7.5) and (7.6), the coarse scan step size S_c should be a multiple of S_f. Notice that, after these rounding operations, the relation $iMin_c < iMax_c$ should be asserted. Using the above numerical values of various parameters, the proposed coarse-fine searching scan mechanism can be illustrated by its control registers, as shown in Figure 7-6.

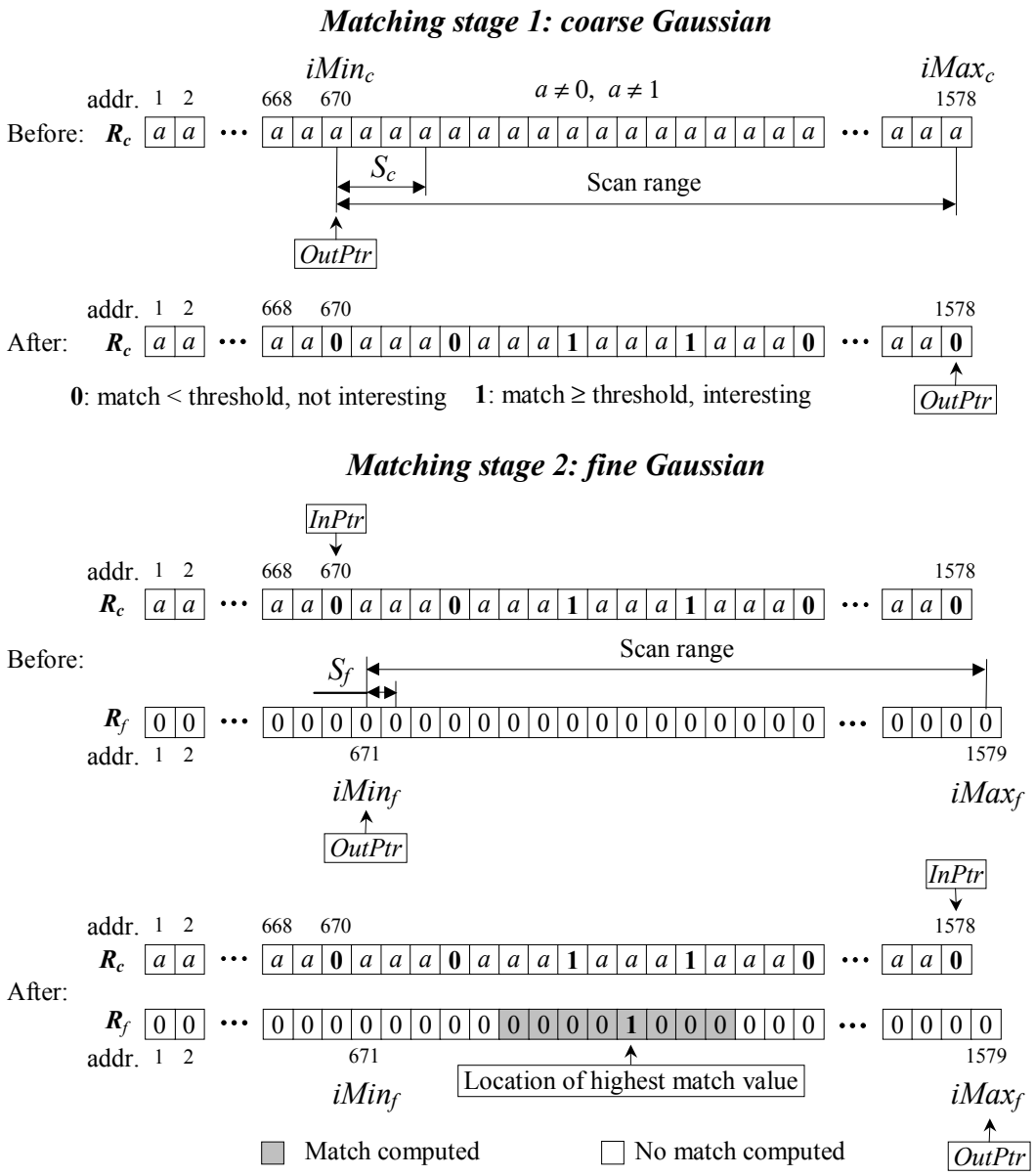


Figure 7-6: Scan control registers

As can be seen, two registers $\boldsymbol{R}_c$ and $\boldsymbol{R}_f$, two pointers *InPtr* and *OutPtr*, are used to manage the searching process. For addressing convenience, the memory allocation of these two registers starts from 1 rather than from the corresponding lower limiters, $iMin_c$ and $iMin_f$, of the scan range. Therefore, the lengths of $\boldsymbol{R}_c$ and $\boldsymbol{R}_f$ are $iMax_c$ and $iMax_f$ respectively.

At the first matching stage, the cells of the register $\boldsymbol{R}_c$ are initialised to an arbitrary value a other than 0 or 1, and the pointer *OutPtr*, pointing to $\boldsymbol{R}_c$, is initially placed at $iMin_c$. From there, *OutPtr* steps up by S_c until $iMax_c$. The Gaussian match value C_{Gau} is computed at every such *OutPtr* position and compared with the Gaussian threshold $C_{Gau,thr}$. If $C_{Gau} < C_{Gau,thr}$, the algorithm writes a 0 in the corresponding cell of $\boldsymbol{R}_c$, otherwise, it writes a 1. If there is no 1 written in any cell of $\boldsymbol{R}_c$ the whole matching process can stop here and the Laplacian match value C_{Lap} is forced to -1. On the contrary, if there is at least one 1 in $\boldsymbol{R}_c$, the matching process continues.

At the second matching stage, the register $\boldsymbol{R}_c$ keeps its value from the first stage and the register $\boldsymbol{R}_f$ is initialised to 0 in every cell. The pointer *OutPtr* is used for $\boldsymbol{R}_f$ addressing and the pointer *InPtr* is used for $\boldsymbol{R}_c$ addressing. During the scan, *OutPtr* steps up by S_f from $iMin_f$ to $iMax_f$ while the position of *InPtr* is updated according to

$$InPtr^* = \text{ceil}\left(OutPtr^* / S_c\right) \cdot S_c - \text{floor}\left(S_c / 2\right) \tag{7.7}$$

where * denotes the position of the pointer. At every *OutPtr* position, the algorithm reads the cell value of $\boldsymbol{R}_c$, pointed by *InPtr*, and computes the Gaussian match value C_{Gau} only if this read value is 1 (greyed area at matching stage 2 of Figure 7-6). A 1 will be written to $\boldsymbol{R}_f$ in the cell where the Gaussian match value reaches the maximum among all computed C_{Gau}.

At the third matching stage, the pointer *InPtr*, pointing to $\boldsymbol{R}_f$, steps up by S_f from $iMin_f$ until the pointed cell contains a 1, where the unique robust Laplacian match value C_{Lap} is computed.

7.2.4. Special consideration for scale retrieval process

In section 5.1.3 we have explained that in Laplacian statistic we have to retrieve the common scale s between audience data and studio data and this is realised within a band-wise error minimisation process

$$ErrMin_k = \min_s \sum_{j=1}^{N_l} Delta_j \tag{7.8}$$

where $Delta_j$ is the robust distance measure defined in (5.22), N_l is the number of features within one subband and k is the subband index. The candidates for this scale are $s = f_j / r_j$ for any $j \in [1 \cdots N_l]$, where f is the audience data feature and r is the studio data feature.

Usually, this scale retrieval process can be implemented in a band wise double loop.

```
initialise ErrMin_k to large;
for j from 1 to N_l
    Err_k = 0 ;
    s = f_j / r_j ;
    for i from 1 to N_l
        Err_k = Err_k + Delta_i
    end loop i
    if Err_k < ErrMin_k
        ErrMin_k = Err_k ;
    end if
end loop j
```

This scale retrieval is a heavy task because in order to obtain the minimised band-wise match error $ErrMin_k$ we have to compute N_l time $\sum_{j=1}^{N_l} Delta_j$.

It is possible to lighten this process because the curve Err_k versus s exhibits a unique local minimum. To convince ourselves, it suffices to take a look at the experimental plot of this relation as presented in Figure 7-7 for both the auto match case (with noiseless audience data) and the cross match case (with noisy audience data).

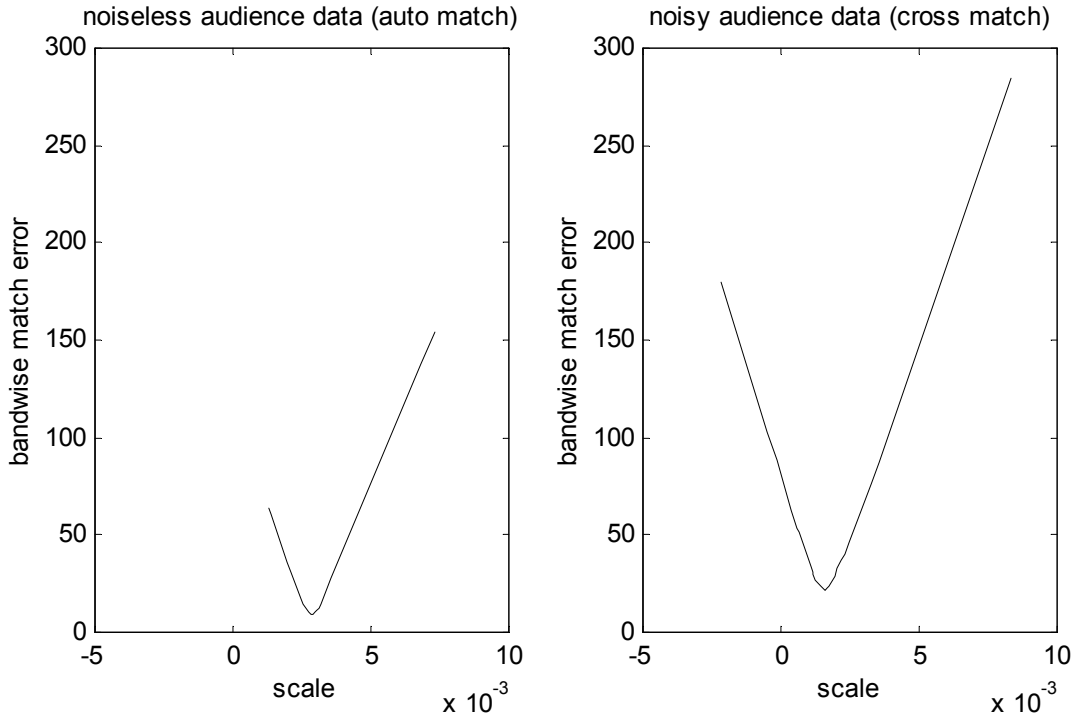


Figure 7-7: Match error versus potential scales

This situation suggests the following modifications to the above error minimisation process:

1) sort the candidate scale set $\{f_j/r_j\}_{j=1}^{N_l}$ in an ascending order before the process;
2) stop the process when $Err_k > ErrMin_k$.

The corresponding modifications of the pseudo-code are straightforward. With these modifications, the computation of the Laplacian match value can be greatly sped up.

7.3. Summary

In this chapter, we have discussed the use of multirate signal processing technique to gain further computational efficiency in the filtering operations.

We have pointed out that a real time implementation of feature extraction program in portable audience recording unit is a necessity because of the limited available memory in target DSP processor.

We have listed some sensible issue for processor related optimization to reduce the computational complexity for a possible gain of up to factor of 6.

With regard to the matching process, we have clarified the time relation between the audience data frame and studio data frame, as well as the time resolution the searching scan can achieve.

We have then given a numerical example on the use of searching scan control registers.

Finally, we have explained how the heavy computation of scale retrieval process (loudness estimation) in robust Laplacian statistics can be lightened using sorted set of scale candidates.

Chapter 8
Conclusion

In this final chapter we will give the appreciation and critiques to the present work so as to draw some useful conclusions thereupon.

8.1. Achievements

In the present work, a template matching based acoustic signal recognition system is constructed for automated TV/radio audience measurement application. This system is characterised by its simple and efficient feature extraction scheme, its robust Laplacian distance metric and its effective coarse-fine searching scan mechanism.

We have shown that the feature generation strategy has been to extract the variation of the local energy contour in the most informative frequency range (375 to 1'500 Hz). It has been realised by subband decomposition followed by energy smoothing and derivation. We have demonstrated that the white type noise and echo can be cancelled and the small time shift invariance can be obtained at this pre-processing stage. Combined with further data compression, the total data reduction factor has reached as high as 285 in portable audience recording unit.

The elaboration of the new robust Laplacian distance metric has started from the Laplacian likelihood ratio and resulted in a generalised Manhattan norm. In this metric, a unique parameter β, being function of class variances, has allowed to tune the distance metric so that the overall classification performance is optimised. Using this new distance metric,

the matching certainty coefficient has been derived from the normalised version of the minimised distance which in turn has been a result of signal loudness estimation (scale retrieval) running in a distance minimisation process. This distance minimisation process has been performed in a band-wise manner to make the system invariant against the frequency dependent sound power damping.

We have demonstrated experimentally the validity and advantage of incorporating conventional Gaussian template matching. And finally, the effective Gaussian-Laplacian alternated coarse-fine searching mechanism has been constructed.

8.2. Performance and product

The crucial requirements to the system mentioned in chapter 1 have been fulfilled as follows.

(1) Portable audience recording unit:

Using TMS320-C50 16-bit fixed point DSP, the final optimised feature extraction program for the portable audience recording unit has resulted in an executable file of 5.5 k words code size, 1 k words data memory use and 1.8 MIPS computational complexity. The compressed audience data occupies about 1 Mbyte low cost external memory (Flash) for one week functioning if it operates 4 seconds recording per minute. The portable audience recording unit is finally packed into a wristwatch with low-power system design (Figure 8-1).

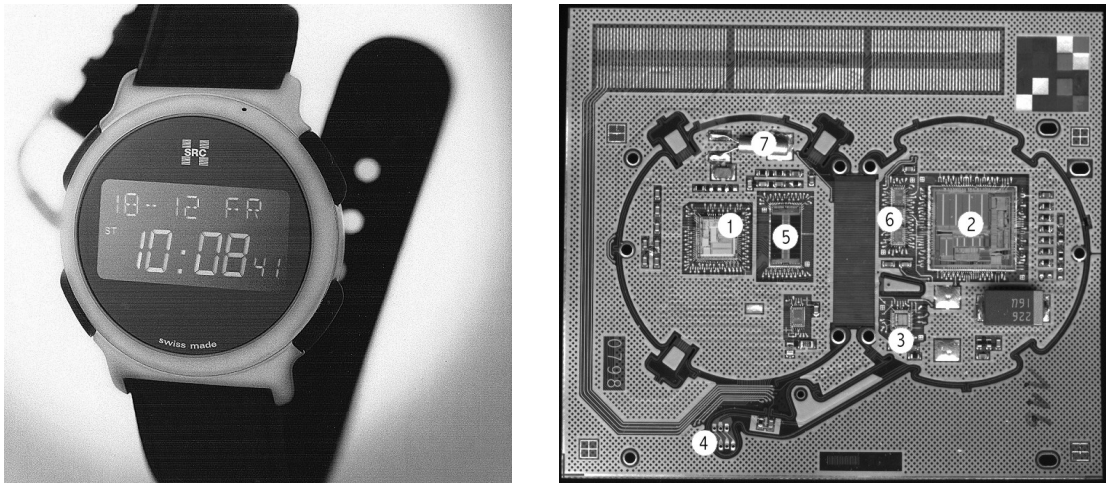


Figure 8-1: Audience recording watch and its Printed-Circuit Board

In Figure 8-1 we identify the following main components: 1. processor Seiko playing the role of watch controller and task scheduler; 2. TMS320-C50 16-bit fixed point DSP; 3. 14-bit low-power AD converter from Institute of Microtechnology, University of Neuchâtel; 4. microphone connection; 5. 1-Mbyte Flash memory for compressed data storage; 6. control logic; 7. quatz.

(2) Matching quality:

The simulation results have shown that the proposed robust template matching algorithm applied to the available training audience database exhibits a false rejection rate of 3.08±0.4 % while ensuring a pratically 0 % false accept rate (see Section 6.1.3). Comparing with the conventional Gaussian template matching, 16 % error reduction is obtained.

The system is invariant against various disturbances including harmonic distortion, echo, time shift, white noise, coloured and structured noise.

(3) Matching computation load:

Considering a realistic example of 100 studios and 1000 portable audience recording units. For non-stop recording during 24 hours with 1 record per minute, there will be $24 \times 60 \times 100 \times 1000 = 144'000'000$ matching operations that have to be executed. With typical configuration, a 400 MHz Pentium II® processor can perform on average about 540 matchings per second. By typical configuration we mean: the recording duration is of 4-second for portable audience recording unit and of 10-second for studio recording unit; full scan of 6-second for each matching operation; coarse scan step size is 8 and fine scan step size is 1. The above matching job will last 74 hours running on such a machine. In order to provide continuous audience rate information over a long period, we have to resort to the parallel computing technique or other high speed computing means.

8.3. Discussion and future work

While this pattern recognition based audience measurement approach is industrially proved to be successful due to its measurement passiveness and its adaptability to media technical environment, the huge

amount of computation load and huge data storage and transfer need serious consideration.

The further improvement of classification quality are limited or could be influenced by the following factor:

- statistical model mismatch;
- feature independence mismatch;
- class separability due to feature selection (time-frequency resolution, recording duration and sampling rate);

Experiments have shown that when the signal to noise ratio is less than –5 dB the match value is significantly degraded.

The time location of the peak match value in matching process can be taken into account to enhance the matching quality since a normal audience should possess time continuity. However, some very repetitive sound signal like techno music can be troublesome.

The simulation results have shown that the matching quality has a sensitive impact with the audience data compression by quantisation. A study of the matching quality versus number of bit of the quantizer can suggest an optimal choice.

It could be an interesting alternative to move the studio recording unit to the audience info study service. In other words, instead of recording the reference signal before broadcast in studio, we record the reference signal after broadcast via high quality receiver in the audience info study institution. This will facilitate the reference data storage and transfer. The impact of doing so, in terms of matching quality, should not be significant if the added signal distortion is not too important.

Bibliography

[Arbi] www.arbitron.com

[Benn48] W.R. Bennett, "Spectra of Quantized Signal", Bell System Technical Journal, 27: 446-472, July 1948.

[Bend93] Julius S. Bendat & Allan G. Piersol, "Engineering Applications of Correlation And Spectral Analysis", John Wiley & Sons, 1993.

[Croc83] Ronald E. Crochiere & Lawrence R. Rabiner, "Multirate Digital Signal Processing", Prentice Hall, 1983.

[Duda73] Richard O. Duda & Peter E. Hart, "Pattern Classification And Scene Analysis", John Wiley & Sons, 1999.

[Drap98] N.R. Draper, H. Smith, "Applied Regression Analysis", John Wiley & Sons, 1998.

[Flie94] N.J. Fliege, "Multirate Digital Signal Processing", John Wiley & Sons, 1994.

[Jeli97] Frederick Jelinek, "Statistical Methods for Speech Recognition", The MIT Press, 1997.

[Kais94] Gerald Kaiser, "A Friendly Guide to Wavelets", Birkhäuser, 1994.

[Kay93] Steven M. Kay, "Fundamentals of Statistical Signal Processing, Volume I, Estimation Theory", Prentice Hall, 1993.

[Kay98] Steven M. Kay, "Fundamental of Statistical Signal Processing, Volume II, Detection Theory", Prentice Hall, 1998.

[McDo95] Robert N. McDonough & Anthony D. Whalen, "Detection of Signals in Noise", Academic Press, 1995.

[Mix95] Dwight F. Mix, "Random Signal Processing", Prentice Hall, 1995.

[Papo91] Athanasios Papoulis, "Probability, Random Variables, and Stochastic Processes", Boston: McGraw-Hill, 1991.

[Poyn96] Charles Poynton, "A Technical Introduction to Digital Video", John Wiley & Sons, 1996.

[Proa96] John G. Proakis & Dimitris G. Manolakis, "Digital Signal Processing", Prentice Hall, 1996.

[Sayo96] K. Sayood, "Introduction to data compression", San Francisco: Morgan Kaufmann, 1996.

[Sebe77] George A. F. Seber, "Linear Regression Analysis", John Wiley & Sons, 1977.

[Sebe89] George A. F. Seber & Christopher J. Wild, "Nonlinear Regression", John Wiley & Sons, 1989.

[Theo99] Sergios Theodoridis & Konstantinos Koutrombas, "Pattern Recognition", Academic Press, 1999.

[Vaid93] P.P. Vaidyanathan, "Multirate Systems and Filter Banks", Prentice Hall, 1993.

[Vett95] Martin Vetterli & Jelena Kovacevic, "Wavelets and Subband Coding", Prentice Hall, 1995.

[Well99] Richard B. Wells, "Applied Coding And Information Theory For Engineers", Prentice Hall, 1999.

[Zeln94] Glenn Zelniker & Fred J. Taylor, "Advanced Digital Signal Processing", Marcel Dekker Inc., 1994.