Error Resilient Compression

of

Digital Video Data

Manuscript

Frank Spaan

Error Resilient Compression

of

Digital Video Data

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,
op maandag 3 april 2000 te 13.30 uur

door

Francisco Henricus Petrus SPAAN

doctorandus in de algemene sterrenkunde
geboren te Utrecht

Dit proefschrift is goedgekeurd door de promotoren:
Prof.dr.ir. J. Biemond
Prof.dr.ir. R.L. Lagendijk

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. J. Biemond,	Technische Universiteit Delft, promotor
Prof.dr.ir. R.L. Lagendijk,	Technische Universiteit Delft, promotor
Prof.dr. F.C. Schoute,	Technische Universiteit Delft
Prof. dr. J. H. T. H. Andriessen,	Technische Universiteit Delft
Dr.ir. E.F.A. Deprettere,	Technische Universiteit Delft
Ir. R.H. Koenen,	KPN Research, Leidschendam
Prof.dr.ir. C.H. Slump,	Universiteit Twente

Het promotieonderzoek heeft plaatsgevonden in het kader van het project *Mobiele Multimedia Communicatie* van de Nederlandse organisatie voor Wetenschappelijk Onderzoek.

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Spaan, Franciscus Henricus Petrus

Error Resilient Compression of Digital Video Data / F.H.P. Spaan
Delft: Technische Universiteit Delft,
Faculteit der Informatietechnologie en Systemen,
Subfaculteit der Elektrotechniek.
Thesis Technische Universiteit Delft. – With ref. – With summary in Dutch.
ISBN 90xxxxxxxx
Subject headings: video compression / error resilience.

Foutrobuuste Compressie

van

Digitale Videodata

Contents

PREFACE

This thesis describes the research that was carried out by Frank Spaan at the Delft University of Technology at the Faculty of Information Technology and Systems, in the Information and Communication Theory Group.

The structure of this thesis is largely based on papers [26], [105], [117], [122], [123], [125], [126], [127], [128] and [156]. For reading Chapters 1 and 2, which consist of the introduction and the context within which the work has been carried out, no specific background is required. For the remainder, however, the reader is assumed to have a background in video compression.

The content of this thesis can roughly be divided into two main parts. The first part consists of Chapter 1 through 3 and addresses the context of the work and the Mobile Multi-media Communication project. The second part, Chapters 4 through 7, addresses the state-of-the-art in existing error resilient video compression techniques and our contributions to this field.

The work reported here has been integral part of the Mobile Multi-media Communication project [11], DEL.3671, funded by the Dutch technology foundation S.T.W., part of the Dutch research council N.W.O..

SUMMARY

Communication plays an important role in our lives. There is almost no aspect of our activities without one form or another of communication. There are many ways in which people can communicate with each other, for instance using language or visually. Visual communication has its own special characteristics which make it different from other kinds of communication. For visual communication, as for most types of communication, technical means have been developed, an example is television. Television and video as a medium of communication have rapidly become a part of Western culture. The video signal is changing more and more from analogue to digital, and when working with digital video, one of the things one encounters first is the large amount of data that is necessary to represent the image content, which is a problem for both storage and transmission of video. Therefore compression of the digital video data is mandatory, for which several compression standards exist. *Mobile* visual communication is not common yet but is an important topic in many research institutes, arisen from the recent tendency of people wanting to communicate anytime, anywhere. The combination of *compressed* video data to be communicated to and from a *mobile* user poses challenging problems.

The Mobile Multi-media Communication project of the Delft University of Technology tries to find solutions to these problems. There are five research disciplines involved: the application, the user interface, the video compression, the network protocol and the transmission. Although there is co-operation between these areas, each of the research areas produces their own results. In order to integrate these results, co-operation was stimulated in the first phase of the project, and in a later phase an integrated concept was developed for a future mobile multi-media communication system, to be operational five years from now. Another effort was the common experiment environment where combined experiments were performed.

For video compression, the problem which is specific for mobile video communication is the presence of transmission errors that corrupt the compressed bit stream. The standard compression techniques are not sufficiently equipped to deal with these errors. Therefore, compression techniques are being developed today to reduce the impact of errors. However, they are not sufficient for our purposes. In this thesis we present new approaches that we have developed and that each have their advantages.

We have modified a H.263 compression standard to make it more error resilient. Instead of using the previous frame for prediction, a prediction frame is used which is composed of several previous frames which smoothes the impact of errors. This technique is not dependent on knowledge whether or where any errors have occurred. Also an error detection algorithm is applied which searches in the decoded reconstructed frame for errors. The locations of the errors are signalled to the encoder which intracodes at that position a macro block, some frames later due to the delay. In the meantime at the decoder error concealment is applied. Results show a substantial improvement in visual quality at the same bit rate. A comparison of part of this technique with the error resilience tools of MPEG-4, showed that the proposed technique outperforms MPEG-4 in peak signal-to-noise ratio, at the cost of an increase in bit rate.

The trade-off in terms of resulting visual quality and bit rate between this approach and the error resilient network protocol that was also developed in the Mobile Multi-media Communication project has also been addressed. As this involves a complex multi-dimensional problem, an experimental approach was chosen. The aim was to have a high image quality in the presence of errors while keeping the increase in bit rate as low as possible. We looked at the effect of application of a finer quantisation, of error resilient source coding, of error resilient channel coding and of combinations of these three. We investigated the case of single-bit errors at different bit errors rates and of packet loss and burst errors also at different error situations. For the whole spectrum of error situations, the combined error resilient source channel coding technique appeared to be the best solution. For single-bit errors only the best solution depends on the bit error rate. For packet loss and burst errors error resilient source coding is necessary, possibly combined with channel coding. For a simple system channel coding only would already cover most situations in the case of single-bit errors.

In current coding techniques, such as the MPEG-4 standard, the coding of video objects and their shapes is important. Not many error resilient shape coding techniques exist. We developed a new technique which is based on polar co-ordinates and spectral decomposition. Its error resilience is shown both analytically and experimentally, as well as its efficiency in bit rate. The computational load is low.

Generally speaking, it is not possible to have mobile digital video communication without some kind of enhanced error resilience. While for our purposes error resilience is not sufficiently incorporated in the existing compression standards, we have shown examples of feasible new techniques.

Summary of the thesis: “Error Resilient Compression of Digital Video Data”.

Frank Spaan, Delft, September 1999.

Chapter One

1 INTRODUCTION

This chapter serves as an introduction to the thesis. In the first section, we discuss the framework. In Section 1.2, we state the problem this thesis addresses, and in Section 1.3 its contribution to the field of digital video compression research. Section 1.4 is an outline of the thesis.

1.1 Introduction

Communication plays an important role in our lives. This is as much true today as it has been in the past and will be in the future. After language, visual communication is one of the most important ways of communicating. Television and video, which are technical means for visual communication, have become an integral part of modern Western culture. Despite the recent development of video on demand and interactive television, this kind of visual communication still remains a kind of communication that is nearly exclusively one-way. Two-way interactive real time visual communication other than face to face meetings apparently is still something of the, possibly near, future.

Another aspect of communication that is gaining more and more importance in Western culture is communication anytime, anywhere. We have recently witnessed a rapid a development in voice communication, especially mobile voice communication. If we combine the visual and mobile aspects, we deal with mobile video communication, which, including other media as well, leads to the challenging subject of mobile multi-media communication.

This challenge already starts at the basis, because, although visual and mobile communication are separately in demand, it is not clear yet what the actual demand will be for a combination of the two. Furthermore, it is unclear what the user-interface for a mobile visual communication system will look like. There are also more technical challenges. The problem of dealing with a large amount of data as in video communication becomes even greater when the data has to be communicated to and from a *mobile* user.

To deal with these problems, an interdisciplinary approach is necessary, as well as in-depth research of specific topics. The different research areas that have to be covered each have their own scientific challenges and technical problems to address. In the research area of digital video for mobile multi-media communication the specific topic that has to be addressed concerns the compression of the digital video data in the mobile situation. This is the subject of this thesis.

1.2 Problem Statement

Until recently the major part of the television and video signals has been analogue, but in the past decade the video signal has become digital. This opened up the possibility of extensive manipulation of the signal. Compression of the data, which is now possible, is important because the amount of data associated with video is very large, and transmission bands for communication are getting filled up. Many researchers all over the world have worked and are still working on solving this problem.

When working with digital video, a large amount of data is necessary to represent the image content; the input data in Figure 1. This is a problem for both storage and transmission of video [107]. The well-known solution to this problem is to rewrite the information in such a way that the size of the data is reduced. In other words, the digital video data is compressed. Before displaying the image, decompression to the original data is necessary in exactly the inverse way the compression was carried out. This principle of compressing a large amount of data to a small amount and decompressing it again is shown in Figure 1.

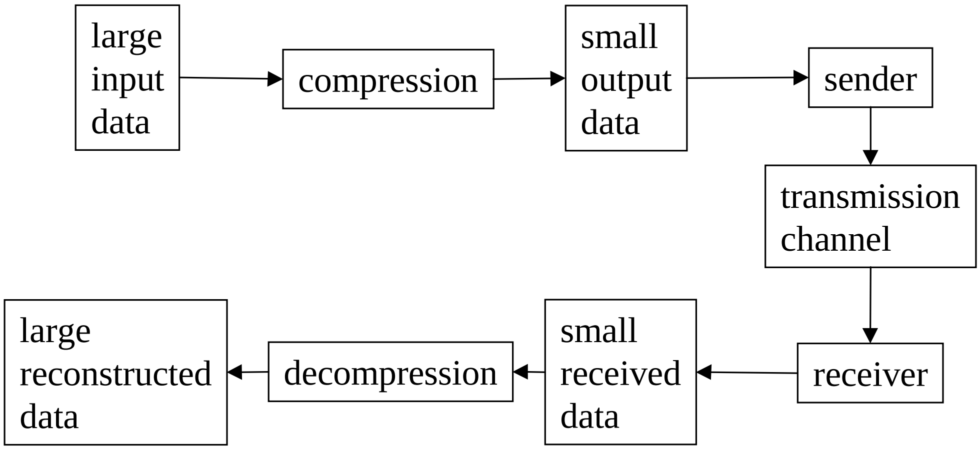


Figure 1 The principle of compression.

Such compression-decompression techniques, the so-called codecs, were developed in the past decades. MPEG-2 [4][132], H.263 [16] and MPEG-4 [42][70] are examples of such compression standards. They are very efficient and recently not much improvement has been obtained in efficiency, even though still a lot of people are working on this subject. Why then more research on compression? In the framework of the Mobile Multi-media Communication project the user is mobile and part of the communication link is wireless. This requires a different approach in the sense that the inherent unreliability of a wireless link can and will cause errors in the compressed data, which is disastrous for the decoding process. Although some solutions to this problem are already described in the literature, these are not sufficient for our purposes.

1.3 Scientific Contribution

The scientific contribution of the work described in this thesis lies in the field of mobile multi-media communication in general and error resilient video compression in particular. However, an exhaustive treatment of every aspect of these fields does not lie within the scope of this research. Nevertheless, different aspects of the research fields have been addressed. These are of interdisciplinary research, defining a quantitative concept for a mobile multi-media communication system, new error resilience techniques for both frame based video compression algorithms and shape compression algorithms, and the trade-off between error resilient source and channel coding.

The interdisciplinary research carried out in the Mobile Multi-media Communication project resulted in showing the importance of the user interaction in the field of video compression. It became also clear that the applicability of breaking down a communication system into independent Open Systems Interconnect layers is limited. This is because the research results for video compression depend on the results in the other research areas, which means that the Open Systems Interconnect layers can not be treated independently.

In the course of the Mobile Multi-media Communication project we introduced a quantitative concept for a mobile multi-media communication system if it were to be in operation five years from now. The reason for making this concept was twofold. The first reason was to make sure that the research results of the different research areas were not conflicting with each other in such a way that in a real world application such a system could never be developed. The second reason was to find out what such a system would look like, especially concerning the quantitative aspects. In defining the concept we showed the viability of such an integrated mobile multi-media communication system. This concept can be used as an indication of a probable future situation in mobile multi-media communication.

For video compression, the problem which is specific for mobile video communication is the presence of transmission errors that corrupt the compressed bit stream. The standard compression techniques are not sufficiently equipped to deal with these errors. Therefore, compression techniques are being developed today to reduce the impact of errors. However, they are not sufficient for our purposes. In this thesis we present new approaches that we have developed and that each have their advantages.

We have modified a H.263 compression standard to make it more error resilient. Instead of using the previous frame for prediction, a prediction frame is used which is composed of several previous frames which smoothes the impact of errors. This technique is not dependent on knowledge whether or where any errors have occurred.

For the cases when error detection is necessary, we found that error detection in the image right before display is effective to maintain the visual quality, which we point out as our optimisation criterion. In the modified H.263 codec we apply an error detection algorithm which searches in the decoded reconstructed frame for errors. The locations of the errors are signalled to the encoder which intracodes at that position a macro block, some frames later due to the delay. In the meantime at the decoder error concealment is applied. Results show a substantial improvement in visual quality at the same bit rate. A comparison of part of this technique with the error resilience tools of MPEG-4, shows that the proposed technique outperforms MPEG-4 in peak signal-to-noise ratio, at the cost of an increase in bit rate.

The trade-off in terms of resulting visual quality and bit rate between this approach and the error resilient network protocol that was also developed in the Mobile Multi-media Communication project is addressed. As this involves a complex multi-dimensional problem, an experimental approach is chosen. The aim is to have a high image quality in the presence of errors while keeping the increase in bit rate as low as possible. We look at the effect of application of a finer quantisation, of error resilient source coding, of error resilient channel coding and of combinations of these three. We investigate the case of single-bit errors at different bit errors rates and of packet loss and burst errors also at different error situations. For the whole spectrum of error situations, the combined error resilient source channel coding technique appears to be the best solution. For single-bit errors only the best solution depends on the bit error rate. For packet loss and burst errors error resilient source coding is necessary, possibly combined with channel coding. For a simple system channel coding only would already cover most situations in the case of single-bit errors.

In current coding techniques, such as the MPEG-4 standard, the coding of video objects and their shapes is important. Not many error resilient shape coding techniques exist. We present a new technique which is based on polar co-ordinates and spectral decomposition. Its error resilience is shown both analytically and experimentally, as well as its efficiency in bit rate. The computational load is low.

1.4 Thesis Outline

The work described in this thesis addresses several questions that are important in the fields of mobile multi-media communication and video compression.

The first question we try to answer is: why is research on mobile multi-media communication necessary? These and other questions of a general nature are dealt with in Chapter 2. In Sections 2.1 and 2.2 we describe the perspective in which mobile visual communication has to be placed. This will make clear what motivations lie behind the research for mobile visual communication; what is essential and what is not, and for which direction research in this area should be carried out. In Section 2.3 we describe how the approach to technical solutions can be made, and which projects world-wide are trying to develop such solutions. Conclusions are drawn in Section 2.4.

One of such projects is the Mobile Multi-media Communication project within which the work that this thesis describes has been performed. This leads us to another question: why the Mobile Multi-media Communication project? The why and how of the project will be described in Chapter 3. The research areas that are covered by the project and the results that have been obtained are described in Sections 3.2 through 3.6. A separate section is dedicated to the technical environment that was developed to carry out experiments in the project: Section 3.7. We state our conclusions in Section 3.8.

The remainder of this thesis concerns the subject of compression of digital video data in the context of mobile multi-media communication. Now yet another question has to be answered, which is addressed in Chapter 4: what are the technical problems and solutions? In Section 4.1 first the problems are described which are encountered when transmitting compressed video data over a wireless link. In Section 4.2 the existing techniques to solve these problems are described. We conclude this chapter with Section 4.3, in which also the problems that still remain are described.

To solve part of these remaining problems, we developed two specific techniques. In Chapter 5 a new technique for increasing the error resilience of hybrid codecs, especially H.263, is described. In Chapter 6 the trade-off between source coding using this technique and channel coding with respect to error resilience and efficiency is addressed. In Chapter 7 a new error resilient and efficient technique for compression of the shapes of video objects is described.

Now we come to the final question: have the questions we have posed here been sufficiently answered, which questions do still remain and are there new questions? We deal with this final question in the Discussion, Chapter 8.

Chapter Two

2 CONTEXT

In this chapter we describe the context in which the research that has been carried out should be placed. Since the subject is mobile multi-media communication, we start with communication in Section 2.1. We then proceed to the aspects of visual and mobile communication in Section 2.2 and reach the area of the technological means to aid communication in Section 2.3. We conclude this chapter with Section 2.4.

2.1 Communication

There is almost no aspect of our lives that does not involve one form of communication or another. There are many ways in which people can communicate with each other. The earliest forms of communication between living creatures that evolved were probably mainly tactile and visual [155]. Although sound was used for communication between creatures at various levels of evolution only man has the capacity to produce more than an elementary form of speech. We can define man as a communicator of information. The outstanding characteristic of man is not simply being a tool-using creature, but is the ability to communicate information to an other human being. It is this capability of communication of very complex ideas between human beings which has led to our present state of semi civilisation [155].

Many people have attempted to analyse communication and developed theories about how people communicate. Even as early as the fourth century BC, Aristotle addressed aspects of communication when he wrote down, although probably he did not invent them, the five canons of rhetoric: inventio, dispositio, elocutio, memoria, and pronuntiatio [44]. They are important because they form the basis of many contemporary studies in the field of communication. The five canons of rhetoric describe the essential processes necessary to the production of a message. Inventio refers to the thought, reasoning and background that go into a message. Dispositio refers to the arrangement of the ideas in a particular order. Elocutio is the style of the message, including the choice of words and phrase. Memoria or memory is not used as much nowadays as it has been in the past. Pronuntiatio refers to the delivery, including the non-verbal cues that accompany the message.

Cicero analysed the speech and felt it has six parts: introduction, narration, division, proof, refutation, and conclusion. He also discussed the gestural behaviours that should accompany one’s verbal message.

Today, giving a straightforward definition of communication is difficult [82]. Much research is done in this field, however, and most first divide communication into verbal and non-verbal communication [44]. Another division that we find often in literature is a division into fifteen distinct conceptual components of communication [40]. Differing selections of the components are incorporated in many communication models [155]. There are three types of communication theories: general, thematic and context theories [44]. However, because describing all these theories is beyond the scope of this thesis, we will give examples of communication instead.

We take a viewpoint that slightly differs from the usual one; we give examples of communication from the area of human culture consisting of the following areas: everyday life, science, art, and religion. After this we try to find common denominators. Also, we have to focus on the user, the human being. The technical solutions we are about to investigate concern the means for and not the goal of communication. The goal is determined by the user; what are his or her main reasons for communication?

We now we give examples of communication, limiting ourselves in these examples to communication between living beings:

- In *everyday life*, communication is everywhere and an almost infinite number of examples could be given. However, the most important and most commonly used way people communicate with one another is through conversation, meeting face to face. Like we saw in Aristotle and Cicero, not only the language but also the non-verbal methods of communication are important when meeting a face to face. One can indicate many different reasons for communicating, like social, logistical and business reasons, and for entertainment, and more. Communication by means of technology has become more and more important and has grown rapidly in the past one or two centuries, for instance in the form of letters, newspapers, the telephone and television.
- In *science* exactly what communication is, is not obvious at first sight. In general, a scientist has to deal with the world and its facts. He or she wants to translate this pile of facts, which seems to be chaotic at first, into a more general form, showing general rules and laws. So the scientist deals, by observation and stimulation, with the fundamental structure of the world that is impersonal and eternal. In our restriction this is actually not part of the area of communication. Here personal influences are avoided and therefore nowadays mostly technical equipment is used. The output of this equipment can be interpreted in a consistent way by any person. An important aspect of communication arises when we look at the communication between scientists. When they communicate they do not communicate directly with each other but do this via the subject of their communication. This subject is something having, or should have, an impersonal and eternal status. For instance, if two scientists discuss the area of a circle, the subject is not affected nor will change by their intentions. In education communication is more obvious than in science, as the teacher tries to communicate knowledge to the pupil. Many media are used here, also as a result of recent technical developments.
- In *art* communication is again different. The artist tries to communicate a message through his or her work either consciously, subconsciously or superconsciously. The latter means putting a message in a work of art that can only be understood consciously by an initiate. This message can be personal or a general truth. It is usually a kind of communication that is from one person to many people. Also often, like in painting or composing or writing, the communication is asynchronous because the time and place of creation is different from the time and place of perception of the work by the art lover. The technical means and media used in art are usually any means available, that is, any existing means will be used to communicate the message.
- In *religion* people try to communicate with a deity. This we consider to be inside the limits we posed for the area of communication we address. In this kind of communication people are trying to re-establish the, now partly lost, relation with the spiritual world. The original meaning of the word religion, or religio, comes close to re-establishment. To this end usually a fixed place and time are necessary, and all means of communication are used. The communio in Christian religion comes very close to the idea of communication.

At first sight, it seems as if the examples show very diverse types of communication. In order to find common aspects in these examples, we now try to find a general way of looking at communication. We can see from the examples that in general the reason for a person to communicate is to share, either unilaterally or reciprocally, meaningful information with another person, or persons or being.

We will now try to analyse what happens in the process of communication. The steps we are about to describe are shown, one way only, in Figure 2. Suppose person A observes a part of the world. The observation can be done in many ways, but usually it requires one of the human senses. The part of the world that is observed can be for instance an object, an impression, a feeling or an idea. It is important to notice that the term "world" here is not confined to the physical world but can also refer to the inner world of the person that is communicating. For obtaining the observation, the world has to be interpreted by A and based on that, a selection of a part of the world has to be made. This selection yields a message content to be communicated. Let us now say that A wants to communicate the message content to someone else, person B. To this end, the information is put into a kind of code, a language for instance. The encoded message content can consist of spoken or written verbal symbols but also a visual representation, etcetera. Because B has to be able to understand the message, the kind of coding has to be known by A as well as B; a common framework is necessary. After having prepared the message, it has to be put into the real world; not doing so would require some kind of telepathy in order to be able to communicate. A stimulus in the real world is required to put the encoded message content into a certain form. For instance, in the case of the spoken word this is the voice. The stimulus can be extended or enhanced by technical means.

Now person B has to find out what A is trying to communicate. To make B understand, A has to 'pronounce' properly, while B has to have properly working 'ears' and the message must not be corrupted on the way. Whatever way of and means for communication we choose, the communication of the information is never reliable for one hundred percent. One of the ways the effects of such corruptions of the information and meaning can be reduced, is by means of interaction: “what did you say?”. In such cases it is important to know that something went wrong in the communication, or else there will be a misunderstanding. The importance of feedback for any communication is well known [20].

Summarising, we can indicate the following topics for person A:

- The world, either inner or outer, is taken as the starting point.
- A observes the world, interprets that observation and selects a part which is to be communicated to B; the message content.
- A encodes the message content, based on the common framework.
- A puts the encoded message into a certain form, using a stimulus.
- Rules that indicate how the conversion and de-conversion from world to information is carried out are or have been agreed upon.
- As an extension or enhancement of this stimulus a technical means can be used by A.

Person B has to go through more or less the same procedure but in reverse order. B uses a sense instead of a stimulus, decodes the message content, interprets it and puts it into B's own context. When B wants to given some feedback, the same procedure has to be followed by B as A did.

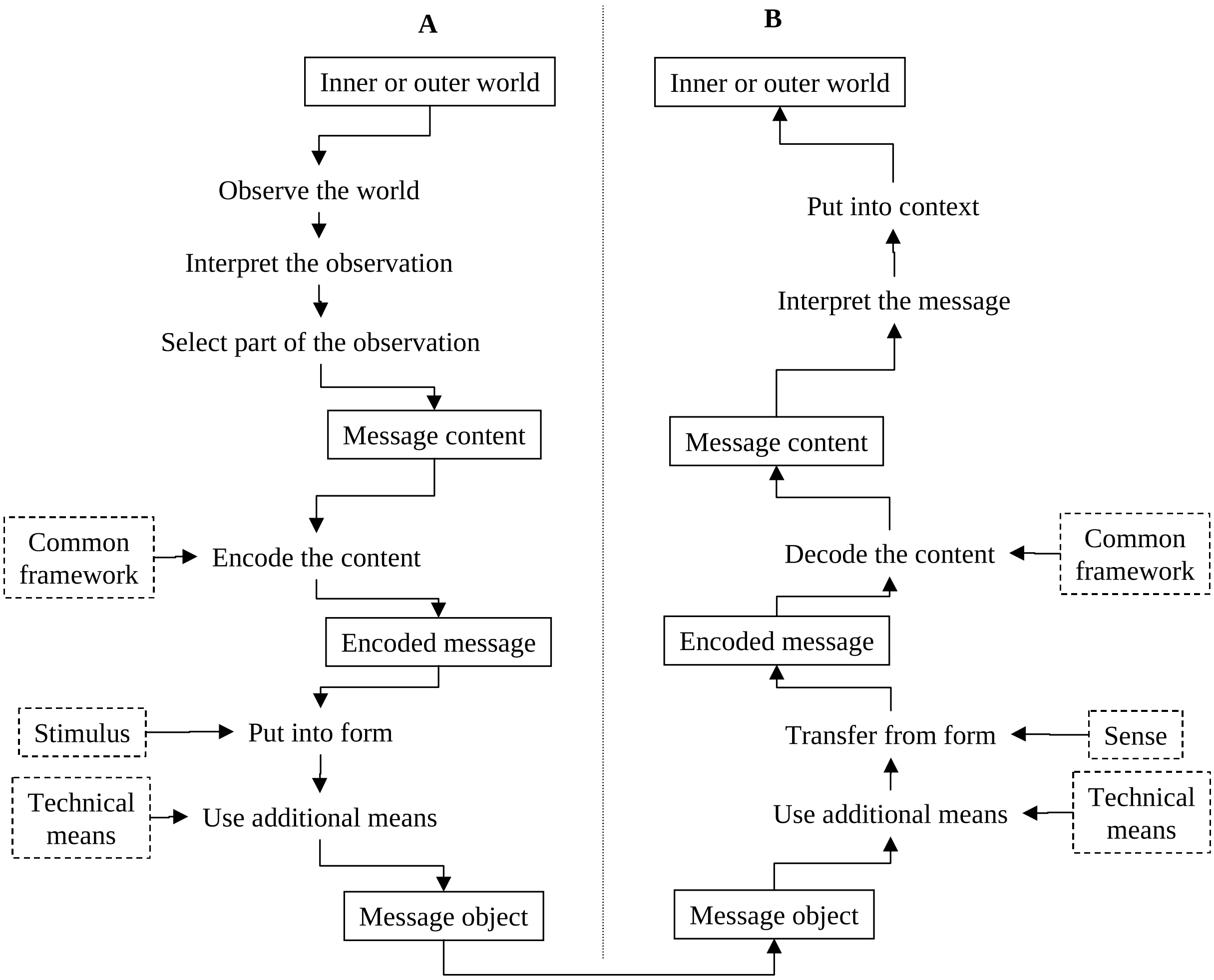


Figure 2 The communication block diagram. Only the communication path from A to B is shown.

This kind of approach to communication is common in communication theory [44][82]. However, our system is slightly different; we explicitly include the technical means, which does not exist in most communication theories.

There are many ways to communicate, and each requires one or more of the human senses. An important division can be made by looking at how the information is perceived when we use them. This can be either in a sequential way or parallel. For example, using language, we perceive the information mostly in a sequential way. If we were just given the words without any order, the message would be difficult to understand. An example of parallel perception is visual communication. For instance, looking at a picture can give much information, but the order is now spatial. We now see that the same division in sequential versus parallel can also be indicated using the terms time versus space, respectively. A video sequence is an example of a medium that combines both. The time – sequential aspect is also present in a video sequence.

Summarising we can say that there are many ways to communicate, each with its own characteristics. However, we have to focus our attention and we do this on a type of communication which has two specific qualities: mobile visual communication. The most important characteristic we already mentioned: it is a parallel way of communication which can be used anytime, anywhere. In the following section we go futher into the subject of mobile visual communication.

2.2 Visual and Mobile

We now focus on using visual data as the medium. Why is visual communication important? There are several ways to communicate visually: one could think of gestures for instance, but here we concentrate on image-type visual data. Visual communication is a parallel type of communication; at one glance one can get a kind of overview, which is not possible using a sequential way of communicating. To give the same information sequentially, a lengthy description is necessary: “a picture paints a thousand words”. Also when a situation or problem with many interdependencies has to be communicated, visual communication is more efficient; a sequential presentation would be unclear and probably mainly try to achieve the effect of an image, ending with “do you get the picture?”.

Technical solutions for the extension and enhancement of the visual senses and stimuli have evolved from painting via making prints, photography, film, television and video to digital video. Especially in the context of video communication three motivations for using this type of communication can be indicated:

- Video makes the transfer of non-verbal information possible mainly through gestures, facial expressions and physical attitude.
- Video gives information about the environment and the presence of people.
- Video gives information about subjects. Two people at different locations looking at the same image and discussing it belong to this category too.

Recently we can also observe a tendency of people wanting to communicate anywhere at anytime. Why is mobile communication important? It enables a better way of communicating in the sense that one can communicate more often, in more situations. Face to face meetings are not always possible, and means have to be found to enable the communication. If a mobile person wants to communicate with someone, the physical distance between the two has to be covered, with something other than fixed connections. However, this extension of the distance also makes corruption of the message more probable. Technical solutions have also been developed. The solutions to problems specific to visual communication anytime anywhere have evolved from reproduction, television and wireless communication to digital mobile visual communication.

A number of points can now be indicated that are important for research on mobile visual communication and that motivate the framework within which we carried out the research described in this thesis:

1. Communication is an integral and important part of human culture.
2. Visual communication has characteristics that cannot be fully replaced by other ways of communication.
3. The role of the user, the human being, in communication is important.
4. Communication is always unreliable to a certain extent.
5. Interaction and knowledge about corruptions are both important.

2.3 Technology

Up to now we described and tried to analyse communication in more or less general terms. Concerning this generality, it has even been suggested that, like thermodynamics, communication theory will always remain conceptual and guiding rather than definitive [155]. So far, communication theory has not led to any startling new practical inventions; it has merely confirmed the inventions which already existed before the theory. These inventions however, are important. We start with a brief historic overview.

The history of civilisation can be regarded as the history of improved means of communication [155]. Primitive humans used some form of visual or aural signalling like drums, smoke signals, mirrors, lights, flags or whistles [52]. Later visual and aural communication was relegated to printed or written form in letters, books, magazines and newspapers, and was conveyed by hand messenger, on foot, on horse or camel back, by wagon, locomotive, steamboat or automobile, and finally by aircraft. The information was physically transported for dissemination. The locomotive entered the scene at the turn of the nineteenth century, the automobile near the end of the nineteenth century, and the aeroplane shortly thereafter. This form of physical dissemination prevails to this day.

Electrical methods of communication appear to have been used, be it crude, by fish and by eels already very early in the evolutionary scale [155]. In human communication however, only in the 1830s the dissemination process was first enhanced by electrical and electronic means with the invention of the telegraph, using explicit coding of the information. In the 1870s the invention of the microphone, telephone, and phonograph allowed the dissemination of aural information directly without conversion to such a code. In the late nineteenth century, the radio signal was discovered which led to wireless telegraphy at the turn of the twentieth century, and in essence this eliminated the need for telegraph wires between sending and receiving terminals. Within the first twenty years of the twentieth century, amplitude modulation radio was developed and the wireless telephone came into being. The additional development of the radio signal through its modification to other forms of modulation such as frequency, phase and pulse code, for broadcasting of aural and visual information, point-to-point communication, data transmission, navigation and radar led to the highly developed communication sector in use today.

The explicit coding of information also has a long history. Various forms of shorthand for reporting rapid speech were developed. The one invented by Tyro in 63 BC for recording the speeches of Cicero was used until the Middle Ages. Julius Caesar used a displaced alphabet code for written communication. Polybius described a Greek method of signalling used about 300 BC. The 25 letters of the alphabet were arranged in five columns and five rows. The respective row and column numbers were indicated separately by the number of torches displayed on top of each of the towers. By the late 1700s a chain of 220 semaphore stations was used to signal over a distance of 1900 miles at a rate of one alphabetic symbol per minute.

In many places the use of combinations of only two different symbols to communicate complex ideas was developed. For example, long and short smoke signals were used by Red Indians, high and low pitched drums by Africans. In the 1830s the telegraph was invented, which also uses a more or less binary code.

We now have to leave history behind and see what is happening today in the field of communication techniques. We focus on the current status of the state-of-the-art technical solutions. The topic we address, communication technology, corresponds to the technical means in Section 2.1, enhancing the mobile multi-media communication.

The starting point for our technical solution should incorporate the points of importance stated in Section 2.2. For the technical field this means that:

1. Research on communication is important.
2. Research on visual communication is important when its parallel characteristics are required.
3. Aspects concerning the human user have to be incorporated.
4. Communication errors have to be expected.
5. Interaction between sender and receiver has to be possible, and error detection is important.

These points play a role in the research on technical solutions for mobile multi-media communication carried out at the Delft University of Technology: the Mobile Multi-media Communication project.

This project is not unique: research and development in this field are taking place all over the world. Before presenting the approach of the Mobile Multi-media Communication project, we first give a concise overview of European activities. Within the European Advanced Communications Technologies and Services programme (ACTS) [1], there are four European Union funded research and development projects, namely the Magic Wireless ATM Network Demonstration (WAND), the ATM Wireless Access Communication System (AWACS), the System for Advanced Mobile Broadband Applications (SAMBA) and MEDIAN (Wireless broadband CPN/LAN for professional and residential multi-media applications). Table 1 summarises these European projects.

ACTS project / Parameter	WAND	AWACS	SAMBA	MEDIAN
Frequency	5 GHz	19 GHz	40 GHz	61.2 GHz
Data rate	20 Mb/s	70 Mb/s	2x41 Mb/s	155 Mb/s
Modulation	OFDM 16 carriers 8PSK	OQPSK coherent detection	OQPSK	OFDM, 512 carriers DQPSK
Cell radius	20-50 m	50-100 m	6x200 m 60x100 m	10 m
Radio access	TDMA/TDD	TDMA/TDD	TDMA/FDD	TDMA/TDD

Table 1 Summary of European ACTS projects

There are also projects in the same research field outside Europe [39][60][89][104][120]. In the U.S.A., the Seamless Wireless Network and the Broadband Adaptive Homing ATM Architecture are two major projects in BELL Laboratories.

Wireless ATM Network is of C&C research laboratories of the Japanese company NEC in the U.S.A.. In Japan, the Communication Research Laboratory is running several Research and Development projects, such as a broadband mobile communication system in the SHF band from 3 to 10 gigahertz, with a channel bit rate up to 10 megabits per second and an indoor high-speed wireless local area network in the SHF band with a target bit rate of up to 155 megabits per second.

We can illustrate the difference between the Mobile Multi-media Communication project and the aforementioned European projects. Figure 3 shows the bandwidth and mobility area covered by the projects. The Mobile Multi-media Communication project aims at covering the '155 megabits per second and mobile' area unlike any other project. The problems and solutions particular for research in this area are described in Chapter 3 of this thesis, where we present the Mobile Multi-media Communication project.

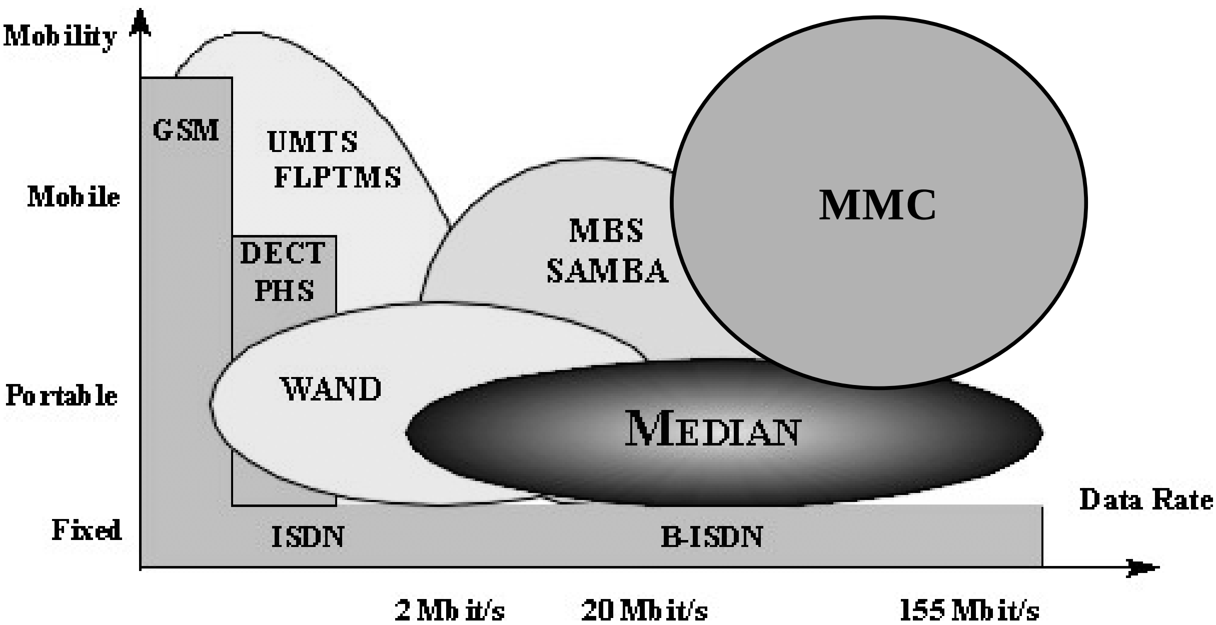


Figure 3. The position of the Mobile Multi-media Communication project.

2.4 Conclusion

In this chapter we followed a path from the past to the present and from general forms of communication to technical solutions for mobile visual communication. We have seen that visual communication is important and that finding solutions for the problems associated with the combination of mobile and visual is not easy.

From a general approach to the way communication takes place we derived five points of importance and applied those points to our case of finding technical solutions for mobile multi-media communication. The first two points, that say that research on visual communication is important, we already have incorporated by taking this specific research subject. The fourth point, saying that errors are important, we incorporated very soon after the start of the research when it became clear that this is the main problem for compression in the mobile situation. Third and fifth point, saying that the user is important and that interaction and error detection are important, we used and found to be valid during our research, see also Sections 3.6.2 and 5.3.2.

Finally, we looked at European and worldwide development in the field of research on mobile multi-media communication. We mentioned the Mobile Multi-media Communication project as one of the approaches followed at the Delft University of Technology. In the next chapter this project will be described extensively.

Chapter Three

3 THE MOBILE MULTI-MEDIA COMMUNICATION PROJECT[1]

In this chapter we go further into the context within which this work was carried out. We describe the Mobile Multi-media Communication project and the different research areas it consists of. Especially the results of the research on link and network protocols will be used in the chapters to follow. Furthermore we describe the video compression research area, introducing the subject and showing the interaction with the other research areas. This will serve as an introduction to the subject of error resilient video compression which is the subject of the remainder of this thesis.

We first introduce the Mobile Multi-media Communication project in Section 3.1. Then the five research disciplines which are part of the project are described one by one in Section 3.2 through Section 3.5. The research area of video compression is treated more extensively in Section 3.6, as this is the subject of this thesis. How the results of the research areas have been integrated in an overall technical concept is also described as part of the different sections. The experiment environment that was set up and used, also for the integration, is described in Section 3.7. We conclude with Section 3.8.

[1] This Chapter is partly based on papers by Spaan et alteres [128], by Schoute [117] and by Prasad et alteres [105].

3.1 Introduction

In February 1996 the Mobile Multi-media Communication project was started at the Delft University of Technology. It is a four-year project, covering several research areas, each of which represented by a Ph.D. student and a professor related to the area. Some more staff and technical support people complete the team. The faculties involved are Information Technology and Systems (subfaculty of Electrical Engineering, involving the *Information and Communication Theory Group* and the *Telecommunication and Traffic-Control Systems Group*), Design, Engineering and Production (subfaculty of Industrial Design Engineering, involving the *Section Information and Communication*), and Technology, Policy and Management (subfaculty of Technology and Society, which includes the *Department of the Psychology of Work and Organisation*).

The scope of research of the Mobile Multi-media Communication project is mobile, and therefore partly wireless, communication through different media, like video, audio, text, graphics, tactile information, and is aimed at the professional user. An important feature of the project is finding solutions for the problems in this field of research through interdisciplinary research. For solving these problems, spanning many fields of research, this is imperative. Although perhaps such unified approaches are not one hundred percent feasible, it is still worthwhile to try to realise them [155].

A more detailed technical framework for the research was specified during the development of the project. The limits this framework implies are not fixed, but show in what kind of framework the mobile multi-media system might operate. The main points are:

- The application area is that of professional users, who are mostly mobile and who use multi-media, that is video, audio and data.
- The bandwidth is large, around 155 megabits per second, and the transmission frequency in the 60 gigahertz range.
- There is an assumed but not specified connection between the mobile multi-media communication system and a backbone. This backbone is assumed to be ideal: 1 gigabit per second, delay limited only by the speed of light, and having a bit error rate of 10^{-12}. Traffic other than from the mobile multi-media communication system is assumed to have no influence on the mobile multi-media communication system.
- The hardware of the mobile multi-media communication system is assumed to be very reliable. The reliability of the received information will be as high as possible, although errors cannot be prevented in the case of mobile real-time communication.

The following topics are *not* addressed:

- Environmental influences like the weather, and lighting, which determines for instance how well illuminated the scene is which in turn determines the performance of the camera and the quality of the video images.
- Multi-point to multi-point communication.

- The routing of the information; that is, where it is determined which user gets to see which information. Some other network routing aspects were addressed although not extensively.
- Three-dimensional video.
- Application specific data like generation of cardiograms.
- The hardware aspects like power, reliability and required number of floating-point operations per second were addressed to some extent, however, a thorough investigation of all hardware was not carried out.
- The way synchronisation is achieved throughout the system.
- Legal and detailed economical aspects, although some cost aspects have been taken into consideration, mainly in the research on the application aspects and on the link and network protocols.

In the Mobile Multi-media Communication project, the research field of mobile multi-media communication is divided into five different research areas, as the aim is to cover all aspects.

- Section 3.2 presents the results of the research, and tries to answer the question where and how mobile multi-media communication will be needed and used most in the first applications. This brought up the concept of expertise at distance.
- Issues related to the user interface were examined. We found that visualising information for a user who also receives all kinds of information from the real world around him or her is problematic. Especially the transparancy of presented visual informatino is a problem we tackled, and discuss in Section 3.3.
- Link and network protocol issues are also addressed. The most important of these are how to give each mobile user the required bandwidth, and dealing with transmission errors while keeping the delay low: Section3.4.
- Section 3.5 addresses the physical transmission issues, like wireless transmission in the 60 gigahertz range and modulation techniques for the required large bandwidth and low delay.
- The research on video compression aspects has resulted in this thesis. A first outline of these aspects starts in Section 3.6.

In each of these research areas results are produced, which can be in conflict with each other. In order to come up with a concept for a complete future system, we combined these results in a consistent and realistic way, qualitativley and also quantitatively. In the description of the research areas in the following sections, the outcome of this effort has been incorporated. As the basis for the qualitative and quantitative description of the different research areas, a complete system has been taken with a targeted time of operation lying within five years from now. As a result, the viability of a consistent and working mobile multi-media communication system around the targeted time is made clear. The actual values of some of the parameters and circumstances that can have an important impact on the actual concept are not known yet, and therefore the concept is intended to present, not a final solution, but a realistic and quantitative picture of a possible complete system.

Hereafter the five research areas of the Mobile Multi-media Communication project are presented. For each area first the main research issues are discussed and the results described. Then, again for each area, a description of the parameters associated with that area is given for a future integrated mobile multi-media communication system. A summary of all the parameters of the integrated system and their values is given in Appendix A. Note that only the outcome of the parameters is given and not the discussion path that was followed obtaining them.

3.2 Expertise at Distance

Given the success of mobile telephony, it is not strange to think of adding a small television screen to the mobile phone showing the face of the person at the other end of the connection. However, there are no indications that users of future mobile multimedia terminals are waiting for something like a portable videophone. Instead of proposing a solution to a problem that may not exist, it seems more appropriate to study what kind of needs different types of future users might have. Therefore situations were addressed in which Mobile Multimedia Communication would probably be used first. The next research subject was the type of work performed in such situations and which types of communication could be helpful in performing the required tasks. This was studied in order to find out which type of audio, video and data, and therefore bandwidth and quality requirements, are a useful innovation to mobile telephony systems. In earlier communication systems, usage was usually pioneered by professionals and professional services that could clearly benefit from the new systems, in spite of the initially high price of the first units. Therefore a study was started to investigate in which professional situations mobile multi-media communication would be most worthwhile. A literature search was carried out, media experts were interviewed and key informants in typical mobile sectors like transport, emergency services and maintenance field services were contacted. Also an approach was used that was more based on theory [53][110] to establish the fundamental aspects of task execution in which visual information from somewhere else might be most needed.

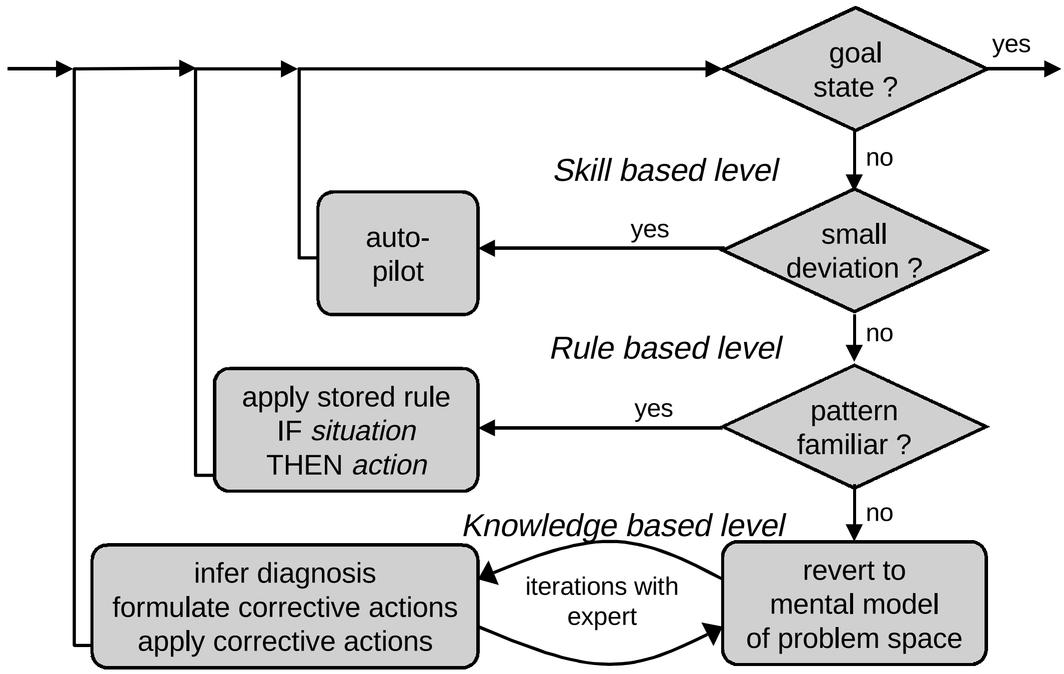


Figure 4. Execution levels of human action

It turned out that the work situations that benefit most from mobile multi-media communication are those in which the worker encounters a complex problem that cannot be solved by known procedures and for which the worker lacks expertise. These situations are typically found in emergency services, medical assistance and disaster management. They are also common in maintenance services dealing with complex equipment. It would be a great help if the person on the spot could use not just a voice or data link to consult an expert, but also a video connection so he or she can show the problem to the expert and the expert can give visual directions. The common denominator in these applications can be summarised as "expertise at a distance". The diagram of Figure 4 classifies the different levels at which human actions are executed. The control loops range from automatic control, the inner loop, to decisions for which additional expertise is required; the outer loop. The ability of mobile multi-media communication to deliver expertise at distance identifies this outer loop as the place where it will show first to be worth the costs, because the other loops can be easily supported by cheaper existing technologies and for the outer loop hardly any technology exists. Therefore, mobile multi-media communication is expected to be used first in situations where unexpected, unstructured, complex problems need to be solved.

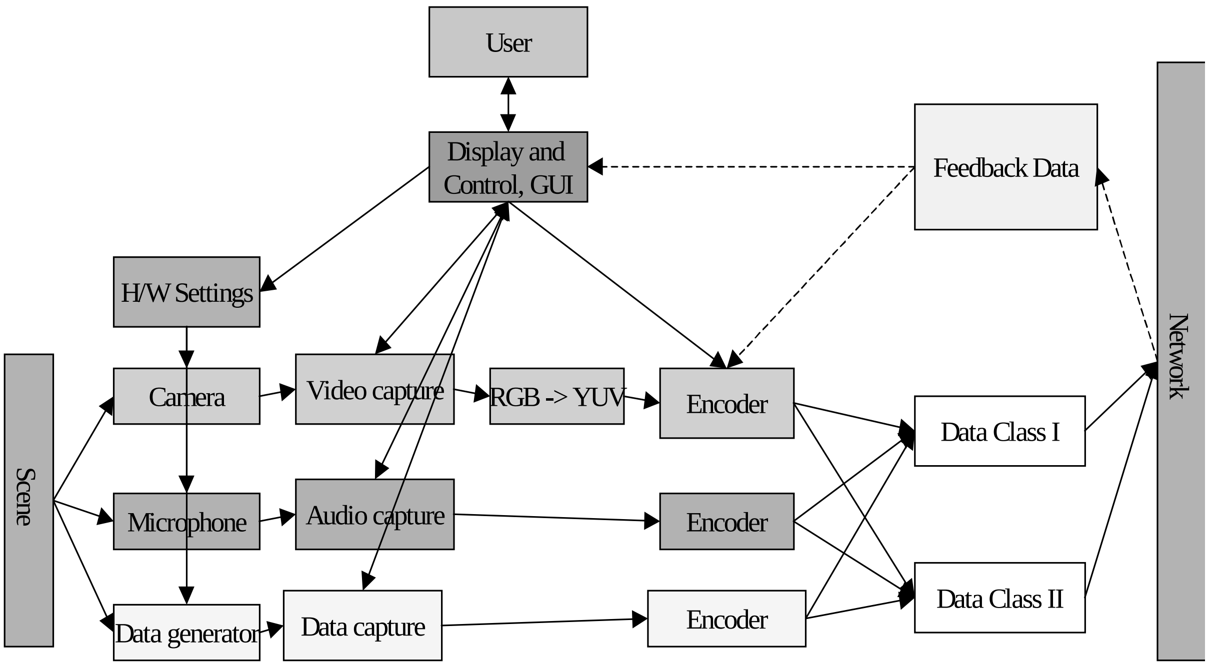


Figure 5. The block scheme of the media in the Mobile Multi-media Communication project.

Reasoning from this viewpoint, it can easily be shown that the emphasis in development of applications should not be on showing the facial expressions of the communicators, but rather on making available a common platform of visual and other information related to the situation at hand that can be manipulated by both the expert and the person on the spot. Important results from this user-oriented study are that real-time audio-visual communication must be accommodated by the system, as well as a reliable and high-speed data-exchange. Highly specialised compression methods, for instance for head-shoulder scenes, are much too limiting for the identified usage. Instead, high-quality and highly error resilient audio-visual compression methods are necessary. Also, the incorporation of the “user-in-the-loop” concept in the entire communication system is of crucial importance to make sure that not only the system but also the user can make a proper trade-off between audio, video and data exchange when the channel conditions are not ideal. For instance, if the system's performance drops, the user must be able to choose between high quality stills versus low quality, low delay moving images. Finally, since both the remote expert and the person on the spot must be able to see and manipulate the common data and visual information, a full two-way communication system is needed.

If we now look at the technical system, we can say that the used media will be video, audio and data. The data can be a town plan, cardiograms, medical images, text, and so on. The maximal video resolution will be in the order of 1000x1500 pixels at 50 hertz, the audio will be stereo and up to 20000 hertz. The data resolution is application specific. The users can change, in the case of a bad channel, the media system parameters, such as zoom, pan, tilt, focus, shutter speed, colour, encoding parameters, contrast, frame rate, resolution, sound sensitivity and resolution, and which media are used. This can be done by a user at either side of the link or automatically. A block scheme of the media system is shown in Figure 5.

3.3 User Interface and Transparency

This research area of the Mobile Multi-media Communication project strives to develop the requirements and user specifications for the professional's wearable computer. The mobile person on the spot must have a terminal of which the visual aspects, especially the problem of transparency, are the most challenging. The interaction with the computer is of a mixed visual, aural and tactile nature. The emphasis in the Mobile Multi-media Communication project is on video information and as a consequence also on the visual aspects of the professional-computer interaction.

Use of mobile multi-media communication in the context of "expertise at distance" raises three major problems in user interface design:

1. The type of equipment to be used, especially for the person-on-the-spot.
2. The design of the human-computer interface proper.
3. The interference of mobile multi-media communication and images with visual and other information present on the spot.

The visual images that professionals receive from their terminals are added to what they see around them: images from their real world environment. For example, users may have head-mounted displays through which they can see real objects with, in overlay, additional visual information. This transparent information can not be projected in just any part of the field of view. Figure 6 shows the visual field for the left eye. The field of view can roughly be divided into three ecologically sensitive areas. The "country side" area depicts the part of the visual field that is sensitive for ego-motion. If something appears or moves in that area, it can give users the impression they are moving themselves and in case of a head-mounted display they might lose balance. The lightened area on the left depicts the area where more sensitivity for approaching objects is found. If something appears or moves in that area one might get the impression that something is approaching and one might instinctively duck away. The text area, at the side of the nose, is used for looking at details. Transparency problems can also be expected in case of auditory information in noisy environments and in case of tactile feedback where the professional touches real objects. The research effort is focused on questions like 'what information should be placed in what part of the visual field?’ and ‘would monocular or binocular displays be more appropriate, and what are the problems with either one of them?’.

Figure 6. The human visual field has ecologically sensitive areas, the left eye view is shown. The shape of the "country side" in the visual field depicts the area sensitive for the motion. The text area on the nasal side is used for looking at details. The high-lightened area depicts where more sensitivity for approaching objects is found.

Another research question relates to the required characteristics of the overlayed information in relation to the real-world images. Answers to this question give information on, for instance, how the amount of image detail in the overlay information interferes with the real-world images. Such information can help to find out what kind of transmission or compression-based errors are still tolerable for the professional to adequately perform the job under different conditions. This translates directly into requirements for the compression and coding techniques to be developed, which is again related to the underlying protocols and transmission methods.

In an integrated application system to be operational five years from now, the user-interface proper will be very application specific. However, we do foresee that in the mobile situation the user will have a head-mounted display for the visual and aural input, and a wearable roller-ball ‘mouse’ and roll-up keyboard for input to the system, as well as a hand-held or mounted camera for visual communication. For less demanding applications other, for instance less costly, user-interfaces can be used like a laptop, headset and camera. The antenna is envisaged to be smaller than 1 centimetre, or a flat patch of 5 x 5 centimetre; in both cases two of them are required to obtain omni-directionality because part of the body of the user can be in the line of sight between this antenna and the sender. The total end-to-end delay has to be kept lower than 100 milliseconds; the geographical distance is therefore limited. The resolution of the media is high, unless in a very specific situation a less costly solution is acceptable [96].

3.4 Link and Network Protocols

Several standards for link and network protocols exist [1][5][6][7][8][9][95]. However, within the framework of our project, specific goals need to be met. The protocol for the wireless channel must be able to counter transmission errors dependent on the requested quality of service. This can be done by forward error correction and automatic repeat request at the link layer [25][34][38][69] [78][81][100], but this has consequences for the delay and throughput performance that is experienced by the higher layers. Another important function of the protocol is the flexible assignment of bandwidth involving multiple base stations that communicate simultaneously with one mobile user, according to the virtual cellular concept [27]. The general situation is depicted in Figure 7.

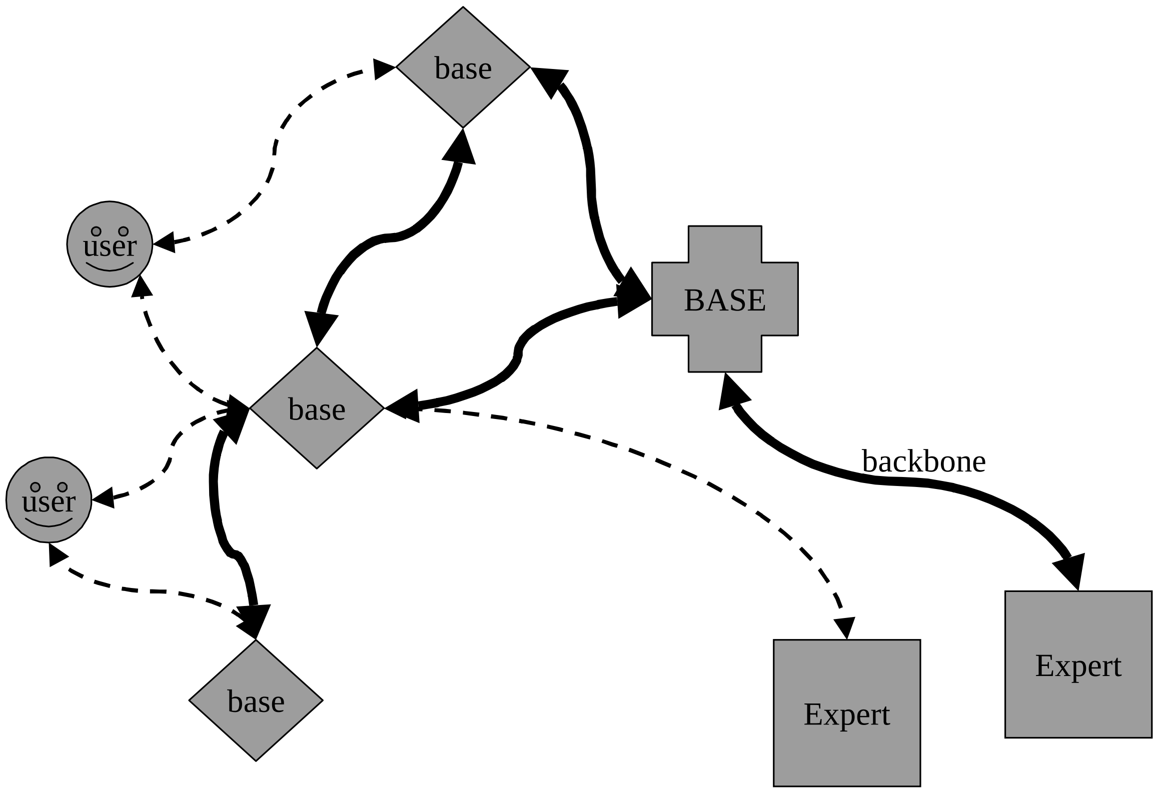


Figure 7 An overview of the connections.

To fulfil those requirements, a hybrid time division multiplexing - frequency division multiplexing technique is chosen combined with a hierarchical structure of radio-data units, fragments, packets and frames. The hierarchical approach allows for easy future expansion, both in terms of the allocated bandwidth and technological advances in modulation and coding. The hierarchical structure is shown in Figure 8, and consists of the following items:

- A *radio data unit* consists of one forward error correction code word. In the system, a (128,106) extended Bose-Chaudhuri-Hocquenghem code is used. Note that in the case of orthogonal frequency division multiplexing one radio data unit is one orthogonal frequency division multiplexing symbol. The transmission scheme is presented in the next section.
- Four radio data units are concatenated to one *fragment*. A fragment is the smallest group of bits over which the automatic repeat request algorithm can

request a retransmission. Each fragment contains 512 bits, of which at least 88 are overhead. Additionally, a fragment may contain a cyclic redundancy check.

- One *packet* consists of one header fragment and seven data fragments. The packet is the smallest unit that can be sent from one station to another. One packet, however, may contain data of multiple qualities of service, because not all of the packet has to be involved in the automatic repeat request.
- A *frame* is the largest unit of data on the radio link. It consists of lots of packets. In other words; there are n x m slots, in $n \geq 1$ frequency bands and $m \geq 2$ time slots. The header gives data like the slot allocation. Each slot in the frame holds one packet; there are n header packets, not to be confused with packet headers, and (m - 1) x n data or user or payload packets.

By using both fragments and packets, one can keep the size of a block requested by the automatic repeat request small. This is good for noisy links, as the duration of a transmission to one station is relatively long, which is good for synchronisation. The target is to keep the transmission time of one fragment shorter than the time over which the channel characterisation changes significantly. This time is known as the coherence time. The total frame transmission time must be small compared to the allowed delay of the data, especially the part requested by the automatic repeat request. The hybrid time division multiplexing - frequency division multiplexing, together with the hierarchical structure, gives sufficient quality of service for the intended multi-media application. Of course, it requires a transmission scheme that supplies sufficient bandwidth to the system.

In an integrated application system, to be operational five years from now, the link from the mobile user to the fixed network will be established using several base stations. They can be fixed, for instance in lampposts or along the freeway or at an airport or a densely populated urban area. They can also be mobile and set out for instance in a disaster area. There is one base station per cell, and there are mobile users, up to 4 per cell, if each one requires 155 megabits per second sustained. With lower bit rates and a lower duty cycle on/off behaviour a lot more users can be served; up to 100. This situation is also the situation when several users are using high-resolution variable bit rate video, because the peaks in the bit rate, in the case of intracoding, will mostly not occur at the same time for different users. There is also a connection to the backbone, for instance using standard high bandwidth equipment in a car or helicopter. The supported speed of the users is up to 540 kilometres per hour to support communication to and from speed trains also. The maximum data rate per user is 155 megabits per second, and per cell 1 gigabits per second of which about 600 megabits per second is reserved for content, and about 10 gigabits per second between base stations.

The link protocol is designed to generate a bit stream that can also be supported by low cost terminals. Users communicate with several base stations at the same time. The users can also request any desired traffic class, quality of service, or bandwidth within the limits stated above. Information about the network status is also passed on to the higher system layers. The delay of the link is about 5 milliseconds, but this depends on the desired quality of service, that is it depends on the number of allowed retransmissions. After multiplexing, first the link buffers the incoming data according to traffic class, after which a fragment of 53 bytes is put together. Seven fragments and a header are then put into a packet and queued for transmission, which is time triggered. Forward error correcting codes are added on the basis of radio data units of 106 bits, generating 128 bit units that are transmitted. Automatic repeat request information, that is acknowledge, negative acknowledge or undefined, is included in the packet header. This same kind of information coming in from the other side contains information about which buffers have to be re-queued for transmission.

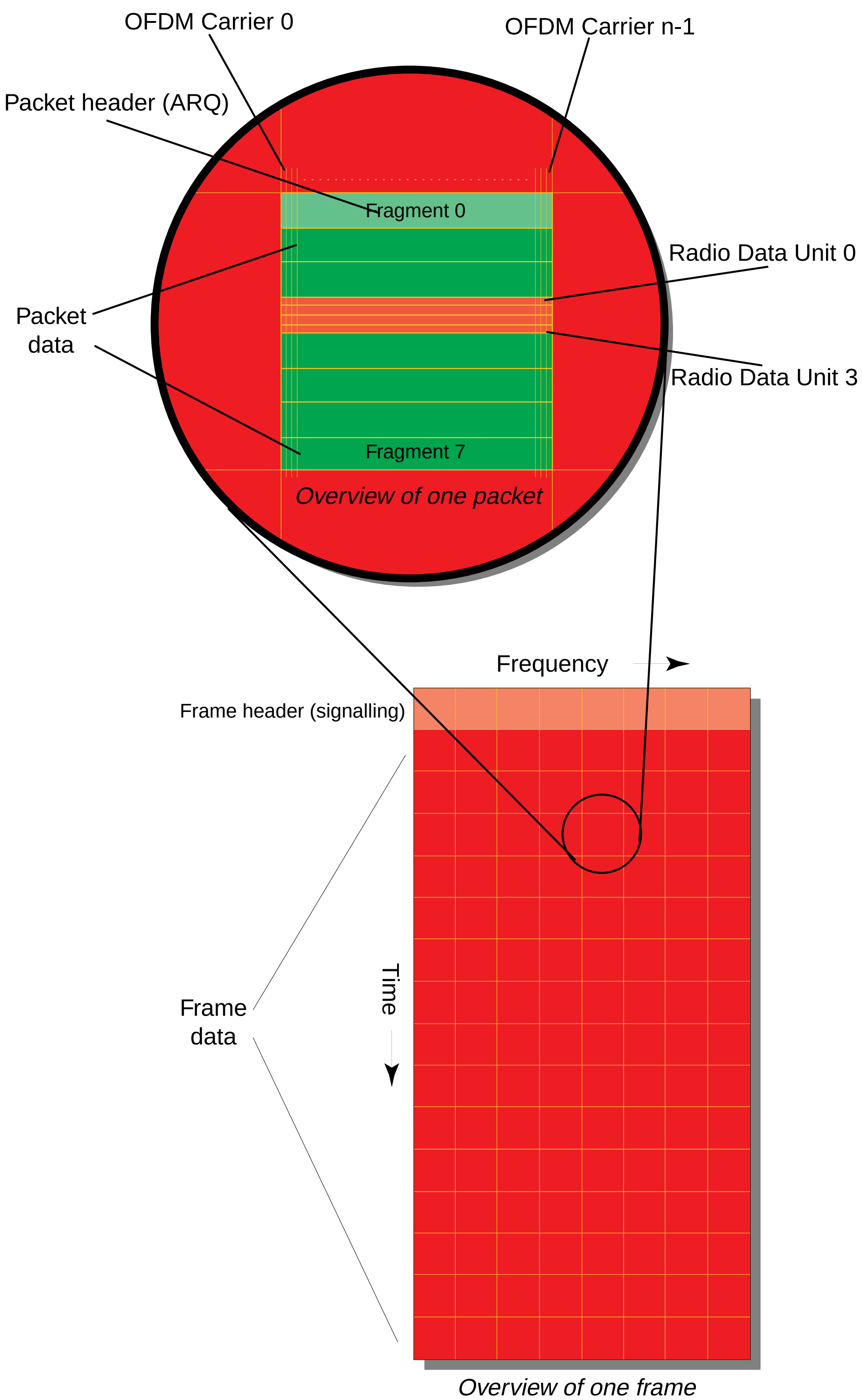


Figure 8 The hierarchical structure of a radio data unit, fragments, packets and frames.

The number of allowed retransmissions depends on the traffic class; this can be for instance 2 retransmissions for video data and 20 for the other data. If after the maximum number of retries the received packet is still not correct, it is discarded and information is lost. Priorities can be set to enforce different qualities of service. This is effectuated by defining for each stream of fragments the maximum number of automatic repeat request retries, the priority, because packets are first filled with high priority fragments, and the expiration time. With these three control parameters graceful degradation is effectuated when transmission conditions worsen.

Concerning the media communication layer a low error rate is expected, which can be achieved because of the available high bandwidth, apart from severe burst errors every now and then. The output bit stream has a variable bit rate. The use of a high video resolution at a variable bit rate and with low delay is made possible by the high bandwidth. The high bandwidth ensures that even even when most of the image is intracoded, like at scene changes, the great amount of data need not be buffered. The delay of the video part is at least in the order of one row of macro blocks but can be up to one frame. The encoded bit stream is split into different traffic classes, depending on the vulnerability of the information. The bit stream is error resilient up to at least 10^{-3} single bit errors in the bit stream to be decoded.

3.5 Transmission

Radio transmission is the foundation of any wireless system. To introduce the subject we show in Figure 9 the available frequency bands in gigahertz for the wireless broadband communications in Europe, USA and Japan [105]. The study group Multi-media Mobile Access Communications, set up by the Japanese Ministry of Post and Telecommunications and the Asynchronous Transfer Mode Forum standardised the new Mobile Broadband System. The European Telecommunications Standardisation Institute (ETSI) standardised the new system for the wireless broadband communications for mobile multi-media.

One can see in Figure 9 that 5 gigahertz and 60 gigahertz are the commercially important bands. All over the world spectrum allocations have been made around 5 gigahertz and 60 gigahertz for the wireless broadband multimedia communication networks. Through the co-operation with the Netherlands Organization for Applied Scientific Research (T.N.O.) and by observing the emerging availability of millimetric wave monolithic integrated circuits, exploration of transmission around 60 gigahertz is possible, where, particularly with the much smaller cells, spectrum appears to be abundant.

With 155 megabits per second the chip rate of code division multiple access will be unrealistically high and the bit duration will be smaller than the delay-spread. Therefore the prime transmission method will be orthogonal frequency division multiplexing. However, the method is hybrid; it also includes time division multiplexing. The width of each radio data unit is equal to the number of bits that are transmitted in parallel through orthogonal frequency division multiplexing, which is implemented with an inverse fast Fourier transform. The full radio path is shown in Figure 10.

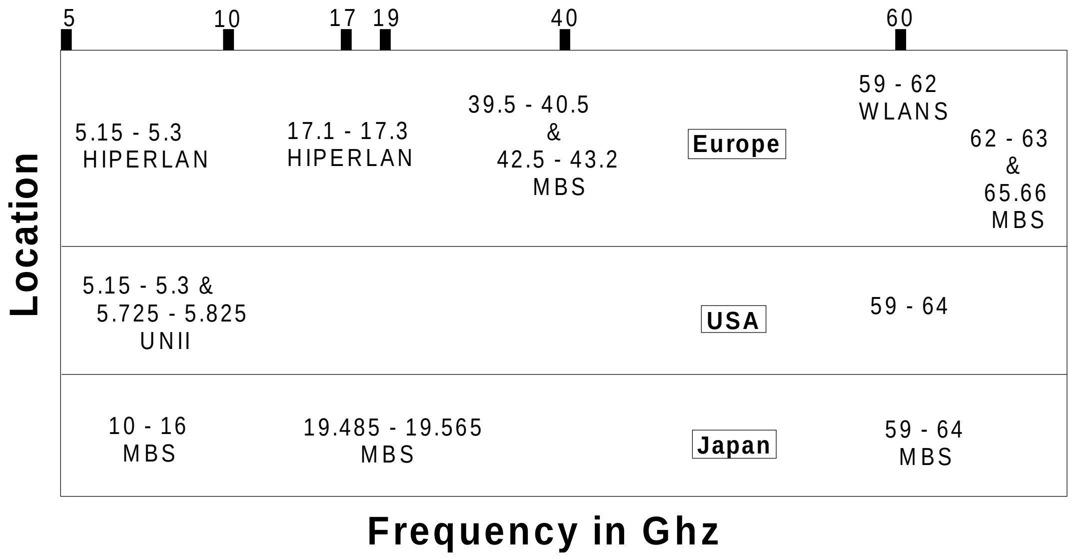


Figure 9. Frequency bands in gigahertz for wireless Broadband Communications. HIPERLAN: High Performance Local Area Networks, MBS: Mobile Broadband System, WLAN: Wireless Local Area Network, UNII: Unlicensed National Information Infrastructure.

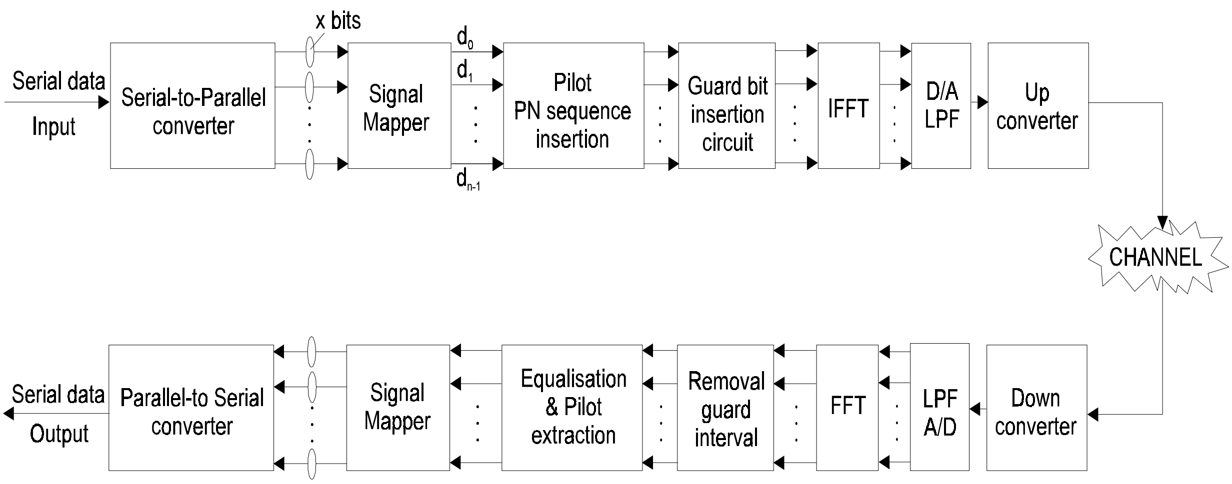


Figure 10. A block diagram of the orthogonal frequency division multiplexing system using a pilot PN sequence and guard bit insertion.

In an integrated application system, to be operational five years from now, the radio link uses a carrier frequency of 60 gigahertz, which creates cells of about 150 meter radius. Because of the small cells a line of sight is usually available and Rice fading is assumed. The bit error rate is about 10^{-4} to 10^{-2}, having additive white Gaussian noise characteristics and there will also be error bursts. However, this data about the error behaviour is uncertain due to the many parameters involved: antenna, power, modulation technique, noise characteristics, etcetera. The incoming serial data is converted to parallel data after which an inverse 128-point fast Fourier transform is applied. Then this is put through a low pass filter and up-converted to the carrier frequency. The modulation technique is orthogonal frequency division multiplexing. The system uses a total bandwidth of about 100 megahertz to transmit 155 megabits per second. In less costly terminals this will be down to 15 megabits per second. The required power will be in between .01 and .1 Watt [84].

3.6 Video Compression

In this section a first outline is given of the research area of video compression, which is the subject of the remainder of this thesis. In Section 3.6.1, we first describe the current situation: existing video standards. Then we give the results of the co-operation between the other research areas and video compression: they are discussed in Section 3.6.2. The integration into a total system is discussed Section 3.6.3.

3.6.1 Existing Standards

In the field of video compression, several standards were developed in the past decades, and the development of new standards is still going on. We briefly describe some of the existing standards.

MPEG-1 [3] is intended for moderately low bit rates, mostly used for a format of 352 x 288 pixels, 4:2:0 and 25 frames per second progressive. In general, the standard specifies a coded representation that can be used for compressing video sequences to bit rates around 1.5 megabits per second for both sequences of frames of 625 lines and 525 lines. MPEG-1 was developed to operate principally from storage media, offering a continuous transfer rate of about 1,5 megabits per second. Nevertheless it can be used more widely than this because a general approach was taken for its development.

MPEG-2 [4] is a very versatile but complex compression standard which builds on the video compression capabilities of the MPEG-1 standard. It offers a wide range of coding tools which have been grouped in profiles to offer different functionalities. These tools are indicated in Table 2. On the right side of the table the levels are indicated, which determine the bit rate and frame size. Only the combinations marked with an "X" are recognised by the standard. Since the final approval of MPEG-2 Video in November 1994, one additional profile has been developed. This uses the existing coding tools of MPEG-2 Video but is also capable to deal with pictures having a colour resolution of 4:2:2 and a higher bit rate. Even though MPEG-2 Video was not developed with studio applications in mind, a set of comparison tests carried out by MPEG confirmed that MPEG-2 Video can at least be characterised as good. The 4:2:2 profile has been finally approved in January 1996 and is now an integral part of MPEG-2 Video.

	Simple	Main	SNR scalable	Spatial scalable	High	Multi-view	4:2:2
High level		X			X		
High-1440 level		X		X	X		
Main level	X	X	X		X	X	X
Low level		X	X				

Table 2 The MPEG-2 video profiles and levels.

MPEG-4 [70] has been approved in 1998 and aims at compression as well as manipulation of video sequences and video objects. It provides standardised ways to

- Represent units of aural, visual or combined content, called "media objects". These media objects can be of a natural or an artificial synthetic origin; this means they can have been recorded with a camera or microphone, or have been generated with a computer.
- Describe the composition of these objects to create compound media objects that form audio-visual scenes.
- Multiplex and synchronise the data associated with media objects, so that they can be transported over network channels, providing a quality of service appropriate for the nature of the specific media object or objects.
- Interact with the audio-visual scene generated at the receiver's end.

MPEG-4 audio-visual scenes, as shown in Figure 11, are composed of several media objects, organised in a hierarchical fashion. At the different levels of the hierarchy, we find primitive media objects, such as

- Still images, for instance as a fixed background.
- Video objects, for instance a talking person without the background.
- Audio objects, for instance the voice associated with that person.

MPEG-4 standardises a number of such primitive media objects, capable of representing both natural and synthetic content types, which can be either two-dimensional or three-dimensional. In addition to the media objects mentioned above and shown in Figure 11, MPEG-4 defines the coded representation of objects such as:

- Text and graphics.
- Talking synthetic heads and associated text used to synthesise the speech and animate the head.
- Synthetic sound.

A media object in its coded form consists of descriptive elements that allow the system to manipulate the object in an audio-visual scene as well as associated streaming data, if needed. In its coded form, each media object can be represented independent of its surroundings or background. The coded representation of media objects is as efficient as possible, taking into account the desired functionalities. Examples of such functionalities are error resilience, easy extraction and editing of an object, or having an object available in a scaleable form.

We turn again to Figure 11 which explains the way in which an audio-visual scene in MPEG-4 is described as composed of individual objects. The figure contains compound media objects that group primitive media objects together. Primitive media objects correspond to leaves in the descriptive tree, while compound media objects encompass entire subtrees. As an example: the visual object corresponding to the talking person and the corresponding voice are tied together to form a new compound media object, containing both the aural and visual components of that talking person. Such grouping allows authors to construct complex scenes, and enables consumers to manipulate meaningful objects or sets of objects. More general, MPEG-4 provides a standardised way to describe a scene, allowing for example to

- Place media objects anywhere in a given co-ordinate system.
- Apply transforms to change the geometrical or aural appearance of a media object.
- Group primitive media objects in order to form compound media objects.

- Apply streamed data to media objects in order to modify their attributes, for instance to a sound, a moving texture belonging to an object or animation parameters driving a synthetic face.
- Change, interactively, the user's viewing and listening points anywhere in the scene.

The scene description is built on several concepts from the Virtual Reality Modelling language (VRML); on its structure and the functionality of object composition nodes. These have been extended to enable the aforementioned features.

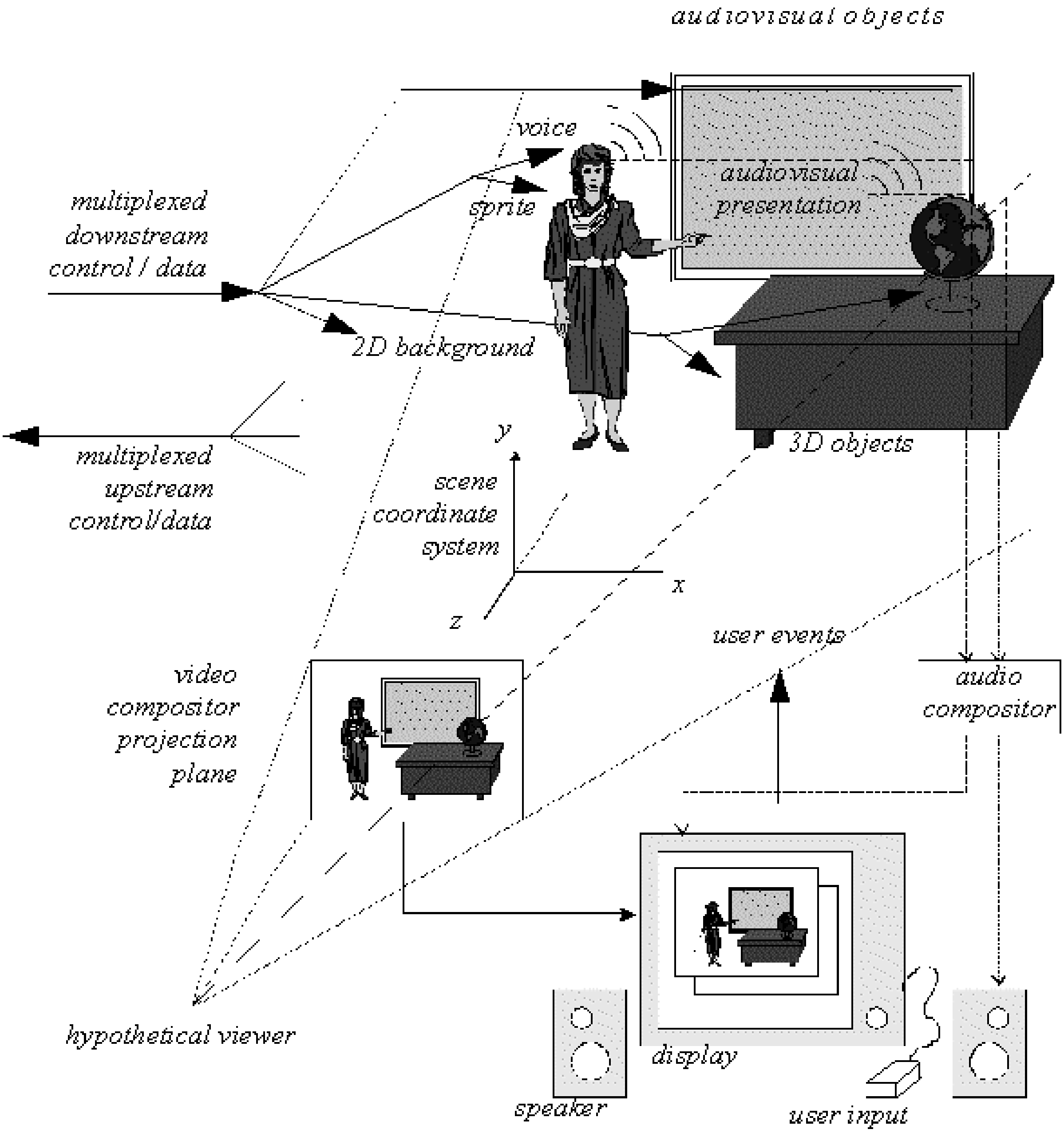
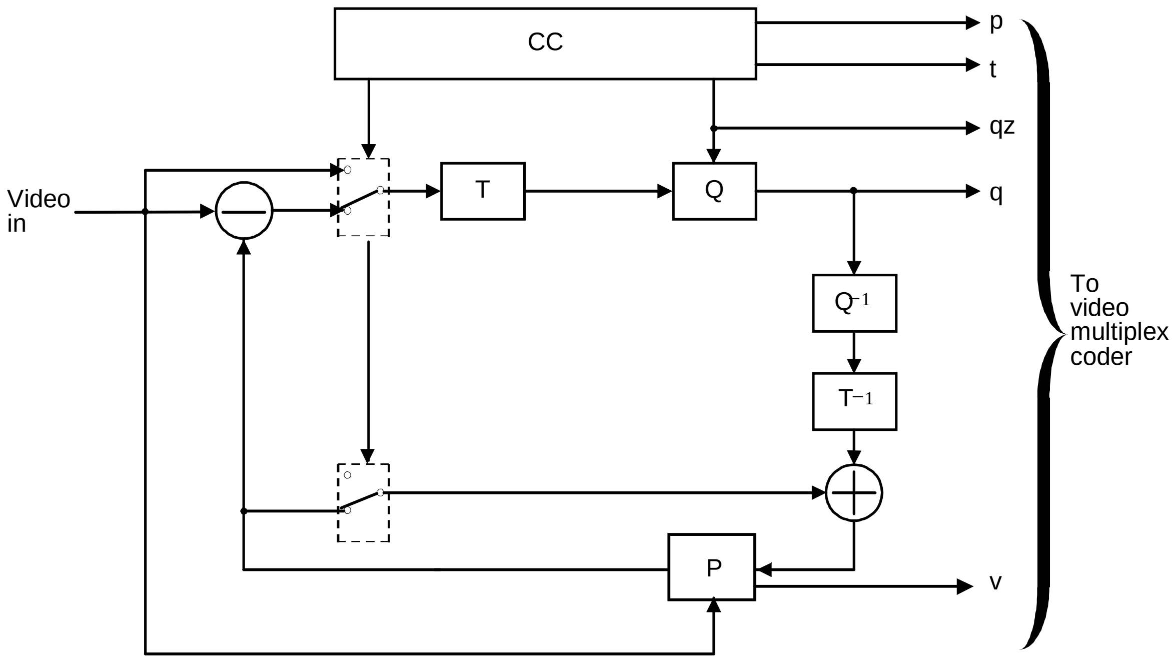


Figure 11 An example of an MPEG-4 scene.

MPEG-7 [90] is being developed and should become a standard in 2001. Instead of aiming at efficient compression of video sequences, it seeks to provide standardised core technologies allowing description of visual data content in multi-media environments. This is not easy, given the broad spectrum of requirements and targeted multimedia applications, and the great number of visual features of importance in such a context. In order to achieve this goal, in MPEG-7 the following will be standardised:

- Descriptors: representations of features that define the syntax and the semantics of each feature representation.
- Description Schemes, which specify the structure and semantics of the relationships between their components, which may be both descriptors and description schemes.
- A Description Definition Language, to allow the creation of new description schemes and, possibly, descriptors, and to allow the extension and modification of existing description schemes.



T	Transform
Q	Quantizer
P	Picture Memory with motion compensated variable delay
CC	Coding control
p	Flag for INTRA/INTER
t	Flag for transmitted or not
qz	Quantizer indication
q	Quantizing index for transform coefficients
v	Motion vector

Figure 12 A simplified block diagram of the H.263 video compression standard.

H.261 [15] is a standard for video telephony. Accepted signal formats are the common intermediate format (CIF), that is progressive 352 x 288 pixels at 29.97 frames per second with 4:2:0 chrominance subsampling, and quarter common intermediate format, with the possibility to skip up to three consecutive frames. The small image format at ten frames per second is normally used for video telephony applications. The compression algorithm uses hybrid compression with motion compensation and predicted frames on a macro block basis.

H.263 [16] is a standard which aims at very low bit rate compression. In Figure 12 a simplified block diagram is shown. The image format can be from a quarter to 16

times the content of the common intermediate format. The high coding efficiency is achieved by means of fractional pixel motion vector accuracy, motion vectors pointing outside the picture, motion compensation on blocks instead of on macro blocks, arithmetic coding and overlapped block motion compensation. In overlapped block motion compensation, each pixel in an 8 x 8 luminance prediction block is a weighed sum of three prediction values, divided by 8, with rounding. In order to obtain the three prediction values, three motion vectors are used: the motion vector of the current luminance block, and two out of four "remote" vectors:
- The motion vector of the block at the left or right side of the current luminance block.
- The motion vector of the block above or below the current luminance block.

We have described some of the most important existing compression standards. However, none of these standards are very suitable for video compression in mobile multi-media communication. Why this is so is described in Chapter 4. We developed more suitable techniques in our research on video compression:
- An error resilient version of the H.263 standard was developed; the algorithm is described in Chapter 5.
- The aspects of combining this technique with the error resilient network protocol developed in the Mobile Multi-media Communication project are addressed in Chapter 6.
- A new shape coding technique which is error resilient was developed, this technique is described in Chapter 7.

3.6.2 Interdisciplinary Research

Part of the research on video compression carried out in the Mobile Multi-media Communication project was interdisciplinary; done in co-operation with researchers from the other areas. This co-operation between the research areas, which has taken place from the very beginning of the project, yielded important results. These results were taken as input to develop the video compression solutions further. The results will now be described.

Co-operating with researchers studying the application of mobile multi-media communication, we found that it was important that the user can adapt the compression algorithm to his or her needs and that the user gets information from the algorithm about the status of the compression process and possibilities to change this. Such a user of a mobile multi-media communication system, behind his or her terminal, can adapt some of the compression parameters in order to direct the impact of the link imperfections to where they will do the least damage. The user will do this according to his or her special demands, situation and channel characteristics of that moment. An automated operation is therefore not possible. Adaptations in case of a decreasing link quality or capacity can be the following:
- Decreasing the frame rate, that is, lower temporal resolution.
- Decreasing the frame size, that is, lower spatial resolution.
- Switching to black and white, that is, lower colour resolution.
- Effecting coarser quantisation, that is, lower resolution in the spatial frequency domain.

One or more of these actions result in bit rate reduction. The bits that then become available can be used for one of the error resilience techniques. The important thing is that one can not predict which of these adaptations the user will want and therefore user interaction in the compression is an inherent part of a mobile multi-media communication system. Although this is an important subject, experiments on this subject could not be performed within the framework of this thesis. We therefore consider this to be a subject for future research.

Taking into account what is mentioned above about the user interaction, it is also clear that the division of the system into different layers according to the Open Systems Interconnect model, shown in Figure 13, as it is commonly done, cannot be done at all times. The "user interaction" is a good example, as this will have an effect through several Open Systems Interconnect layers. For instance, information about the state of the physical channel that is found in the physical layer has to be available to the user, and therefore to the application layer.

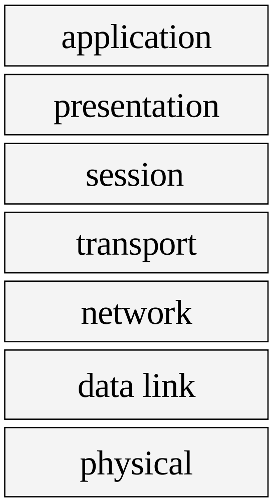


Figure 13. The division of a system in Open Systems Interconnect layers.

Co-operating with researchers studying the link and network protocol for mobile multi-media communication, we found another interesting result. This concerns the trade-off between applying an error resilient video compression algorithm and applying an error resilient network protocol using as a criterion the efficiency and error resilience. Which of these error resilience techniques should be used, depends on the situation. This is the subject of Chapter 6.

3.6.3 Integration Aspects

After the development of a compression algorithm for mobile multi-media communication, which is described in Chapter 5, the integration was addressed of the results of the research on video compression into an integrated application system, to be operational five years from now. In such a system, error resilient video compression is applied, by coding with respect to a composed prediction frame composed of several previous images, instead of the last image, and by detecting errors in the frame to be displayed and requesting macro blocks to be intra-coded in some future frame. A reference image composed of five previous images is used, or an other number depending on the network situation and error rate. Feedback information about the errors detected in the image is transmitted from the decoder to the encoder. In a point to multi-point situation the information of all the feedback channels is used for the encoder adaptation. The amount of source coding and channel coding, which both provide error resilience, is determined automatically on the basis of the information about the network situation.

In the course of the Mobile Multi-media Communication project, it became clear a large bandwidth and high compression of the video data are both available. This is a situation that is different from most communication situations today. It was and is a point of discussion whether the bandwidth is too high, or the compression too high, or both, or that applications for such bandwidth and compression have yet to be found. However, one of the results of the research on the application of mobile multi-media communication is that high resolution and high quality video probably will be important, especially for expertise at distance. This is supported by research on the use of video in interpersonal communication [20][21] which shows that the environment and context of the person or object in the video scene are important. In communication, not only the content is important but also the aspects concerning the interpersonal relations between the people communication. Especially for such aspects so-called analogical communication is necessary and this kind of communication requires richness and therefore at least high resolution. We therefore assume that the bandwidth of 155 megabits per second will be used fully. This yields the following parameters for the video part of the integrated concept.

The compression ratio C is about 20. The available bandwidth of 155 megabits per second supports such a video stream: 1000 x 1500 pixels at 50 hertz means 1.8 gigabits per second uncompressed. When this has to be compressed to 155 megabits per second this means that C should be at least 12, when no delay is allowed. This can then be used to handle the bit rate peaks that occur when information is being intracoded. This in turn means that the value of $C = 12$ refers to a peak bit rate and C should therefore be at least 12. This way the bit rate can be handled also when mainly intracoding is used when there is a lot of motion in the scene, or when there are many errors.

3.7 The Experiment Environment

Within the framework of the Mobile Multi-media Communication project, a common experiment environment was developed. It serves to link the contributions of the researchers and is a vehicle for demonstrating to other researchers and interested parties the state-of-the-art in the different areas that have been discussed in the previous parts of Chapter 3. It also helps to get a feel for the different mono-disciplinary contributions in a multi-disciplinary context.

The individual parts are integrated into an experiment environment that is used with professional test persons. The output of the first tests was used to create a more realistic, but controlled environment. The user interface, the video compression algorithm and the channel model, which includes protocols and transmission, provide an example of the technical environment that people might be working in when they use mobile multi-media communication. This was followed by experiments in which various parameters can be manipulated.

The scheme in Figure 14 gives an overview of the set-up. A more detailed overview of the structure is given in Figure 15, also showing the location of the hardware and software and the two phases in which the structure was built.

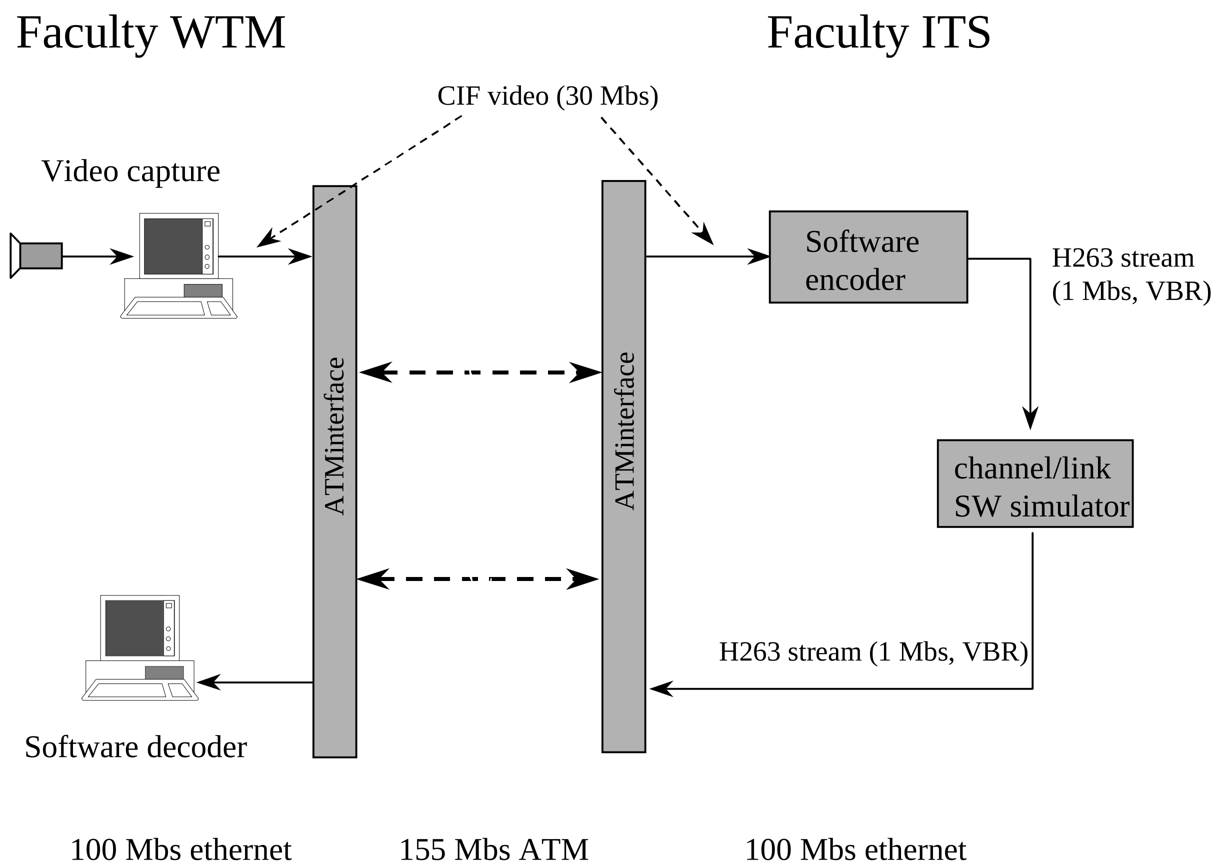


Figure 14 A simplified block scheme of the experiment environment.

The actual implementation of the experiment environment was not trivial, as it involved connecting several buildings and providing real-time video compression and channel and network simulations. In a controlled room in one building the actual experiments with the test persons were carried out, while in the other building the compression and channel and network simulations were performed in real time. The buildings were connected through a dedicated link using asynchronous transfer mode (ATM). The following gives some implementation parameters of the final set-up, which included:

- Four quad Pentium Pro systems.
- A Pentium II with ATM that does the network routing for the four quad systems. This system also has a 50 gigabyte RAID system to store simulation results and video sequences.
- A Pentium with ATM that routes network traffic to and from the laboratory with the control room.
- A Pentium Pro with ATM that is a file server.
- A dual Pentium Pro that is a Web server for intra-project work co-ordination.

All systems run Linux 2.1.90, except the Web server, which runs Windows NT. Full duplex unshared Fast Ethernet connections were used, as well as shared single duplex connections. Plain every day ethernet was used for emergencies. Finally, 155 megabit per second ATM was also incorporated and connected to a Cisco ATM switch. This set-up makes it possible to perform calculations on high volumes of data that come in from the ATM link, move it fast from one quad machine to another and back to the ATM link.

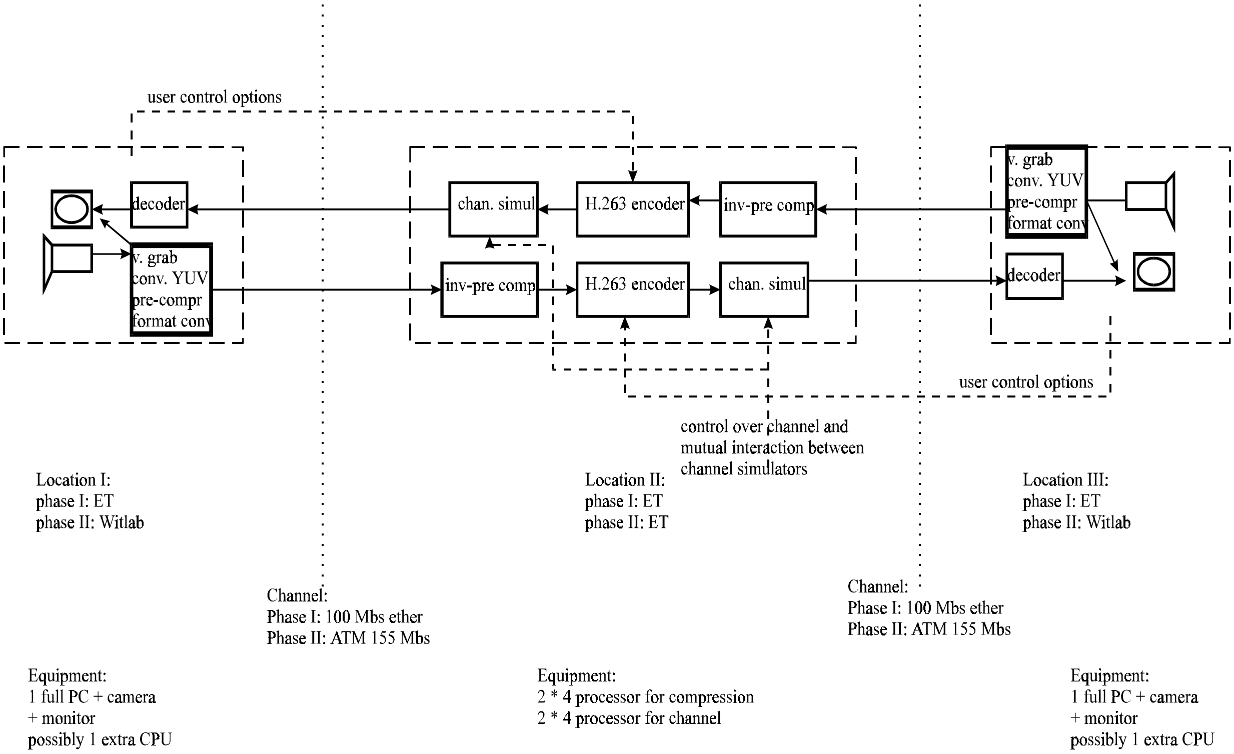


Figure 15. The experiment environment.

3.8 Conclusion

In this chapter we described the Mobile Multi-media Communication project. We described the framework and the different research areas. Attention was also given to integration into a concept for a possible system to be operational five years from now.

We then introduced the area of video compression, starting with a description of existing standards. Further, we addressed the interdisciplinary aspect of the research, from which two important conclusions resulted. First, that the user is an important part of the compression system. This topic should be addressed further in future research. The second conclusion is that the systems of open systems interconnect layers cannot always be used in the development of a compression system.

Finally we described the experiment environment that was used to carried out experiments. It also made possible and stimulated the integration of the research of the different disciplines.

This chapter concludes what we might call the first part of the thesis in which the context of the research was described. In the remaining chapters we address the subject of error resilient video compression. The next chapter starts with this by first looking into what the problems are concerning compression of digital video data in the mobile wireless situation and then describing existing solutions to these problems.

Chapter Four

4 Overview of Error Resilient Video Compression

The subject of this chapter is video compression in mobile multi-media communication. The chapter starts what we might call the second and last part of the thesis. This part concerns video compression. Its content is more technical than the previous chapters and therefore some background in video compression is assumed. We start with the problem statement in Section 4.1, then describe existing solutions in Section 4.2 and conclude with Section 4.3. For some of the remaining problems we have developed solutions, which are described after this chapter.

4.1 Problem Statement

In video communication, like any other kind of communication, information must be conveyed from a source to a destination [155]. There are imperfections in all methods of conveying information, and these introduce the possibility of misunderstanding or, in the case of video communication, corrupted video data. In the past decades, video signals were first mainly analogue but now become more and more often digital. The transmission was and is mostly wireless, except in studio-like situations. Any errors during the wireless transmission does not affect the uncompressed analogue signals very much, but for compressed digital video data errors are indeed an important problem. Up to recently this problem was hardly addressed; only in the recent video compression standards MPEG-4 [42][70] and H.263+ [65] some error resilience is incorporated.

In our field of research, video compression in mobile multi-media communication, part of the transmission is done over the wireless part of the link, and errors can occur. In this situation three kinds of errors can be identified:
- Bit errors: a single bit is flipped or lost.
- Burst errors: a certain number of bits is corrupted or lost.
- Packet loss: the network packet which contains video data is not received at the receiver, for instance due to packet loss in a congested network [108][137][158].

For the first type of errors the main parameter is the probability of the occurrence of an error, the bit error rate. In the case of burst errors, apart from the bit error rate, the burst length is also an important parameter. For the third type of errors, the packet loss rate is important.

The effect of any type of error on the compressed stream of digital video data can easily be disastrous. If we look at an H.263 or similar hybrid codec we can identify the following possible effects of an error:
- A variable length code word is misinterpreted as a different code word of different length. This means that the code words after the error cannot be decoded or will be misinterpreted.
- Only the value, and not the length, of the variable length code word is misinterpreted. This means that the following code words can be decoded.
- There is an error in a part of the stream with fixed length code words, here also the following code words can be decoded.

The first type is the worst type of error because one bit error can cause a substantial part of the compressed video data stream to be misinterpreted or uninterpretable. Unfortunately, because of the efficiency of compression using entropy coding, most of the data is coded with variable length code words.

The following effects of an error on the image reconstructed at the decoder are possible:
- A DCT coefficient gets a wrong value, which leads to a wrong image content in a block.
- A motion vector gets a wrong value, which also leads to a wrong image content in macro block.

- A value in the header gets a wrong value, which can lead to misinterpretation of the whole image content, depending on which part of the header is affected.

All of these effects will become larger with every new frame because the previous frame, with the erroneous content, is used as a prediction for the new frame. This error propagation is worse when there are many motion vectors. It can be corrected when a macro block is intracoded, thereby not using the prediction. In scenes with a lot of motion, more macro blocks will be intracoded and the effect of error propagation on such sequences is therefore less.

Which of these effects will occur depends on the composition of the data of the compressed stream; how many transform coefficients and motion vectors are present and so on. The composition in turn depends on the compression parameters that were used and the image content, for example:

- When the original sequence contains a lot of motion, not many motion vectors will be present.
- When the compression is very high, the number of DCT coefficients will be low with respect to the number of motion vectors and the header size.

The effect of errors in the case of compressed shapes of video objects depends largely on the used technique. If a sequential description of the shape is used, the result of an error is usually that the video object is lost. This is because after the error has occurred, the remaining part of the description of the shape is lost. The effect of errors in the case of the more error resilient shape compression using polar co-ordinates is discussed in detail in Chapter 7.

Because of the above effects, it is important to find a technique to reduce the impact of errors. However, before trying to counter the effect of the errors, we first have to determine what our goal is in doing so. Since we are dealing here with video communication, the optimisation criterion we choose is the visual quality of the displayed sequence at the decoder side. Optimisation of the visual quality in the presence of transmission errors should preferably not lead to an increase in bandwidth, to an increase in end-to-end delay or more stringent hardware requirements. However, almost all error resilience techniques have to act upon loss of information and therefore will require extra bits.

There are mainly two ways to handle transmission errors: using an error resilient compression algorithm, "source coding", and using an error resilient link and network protocol, "channel coding". The subject of Chapter 5 and 7 of this thesis concerns source coding techniques, while the trade-off between an error resilient source and channel coding is addressed in Chapter 6. However, before we discuss the development of new techniques, we first address the different existing error resilience techniques. We do this in the next section.

4.2 Existing Solutions

To combat the effect of transmission errors in the compressed stream of video data, several error resilient video compression techniques have been developed. They are either based on adaptation of the encoder, see Section 4.2.1, or on adaptation of the decoder, see Section 4.2.2, or both, thereby inducing interactivity, see Section 4.2.3.

We address general joint source channel coding in Section 4.2.4, and the error resilience tools of the MPEG-4 standard in Section 4.2.5.

4.2.1 Encoder Based

In this section we describe existing encoder based error resilient source coding techniques:

Layered coding [24][30][50][109][157][160]. For this technique, the compressed bit stream is split into different streams of different importance, as we can see in Figure 16. The more important streams can be sent first, or via a well-protected communication link, which is shown in the right part of the figure. Both techniques reduce the probability of transmission errors in the stream. However, it is imperative to have knowledge about which part of the original stream is important. The stream partitioning is often done in the frequency domain. The first part of the data, either in the frequency or spatial domain, usually consists of a rough presentation of the data, after which successive amplitude refinement and/or spatial/temporal resolution refinement is possible using the following data. It is also possible to use different levels of transmission power to enhance the error resilience of the layer containing the basic information [64][66]. Some techniques make use of the layered structure of the data by using for the prediction in the video compression algorithm only the data that was sent via the stream containing the basic information [50].

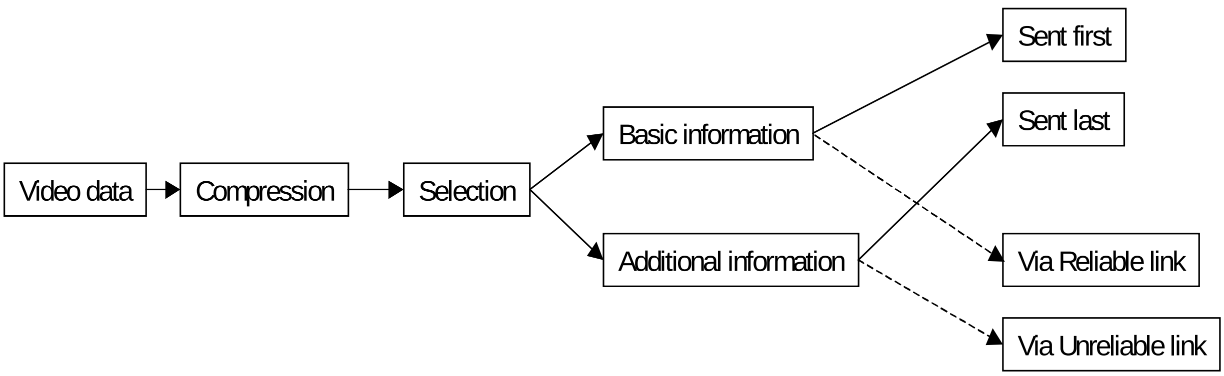


Figure 16 A simplified block scheme of layered coding.

Multiple description coding [43][99][135][145][152]. For this technique, several different bit streams are generated, each carrying a rough description of the data and a different refinement. These streams are transmitted through several physically different channels or paths, thereby increasing the probability that some data will be received. This is shown in Figure 17, where the different bit streams and channels are indicated using labels A, B and C. Even when only one single physical channel exists between the source and the destination, the path can be divided into several virtual channels by for instance time interleaving or frequency division, as this will reduce the probability of an error occurring in the same data segment for the different virtual channels. There are different versions of multiple description coding:

- *Multiple description scalar quantisation* [29][138][139][140]. Two descriptions of each quantisation value are generated, which combined give the exact level, but separately each can act as a rough quantiser result. A simple implementation of this approach is using two quantisers whose decision regions shift by half of the quantiser interval with respect to each other. If each quantiser has a bit rate of *R*, the reconstruction error from two descriptions is equivalent to that of a single R + 1 bit quantiser. In the absence of channel failure, a total of $2R$ bits are required to match the performance of a single quantiser with R + 1 bits. Therefore, the loss of coding efficiency is quite significant for large values of R. More sophisticated approaches can improve this.

- *Correlation inducing linear transforms* [37][56][83][97][146]. A correlating transform is applied to the data of which the redundancy has previously been removed by the compression process. If this transform for instance correlates pair-wise, it produces two groups of data with a pair wise correlation of each of the symbols in the two groups. The data within one group should be uncorrelated. The data of the two groups are sent via different transmission channels. If data of one of the groups is missing, it can be estimated from the data in the other group. This can for instance be applied to the uncorrelated coefficients of the Karhunen Loeve transform which are then again transformed using a pair wise correlating transform and sent via different channels.
- *Subsampling* [135][138][140][145]. A simple way of producing multiple equally important descriptions is by subsampling the original image, coding the subimages independently and transmitting the compressed data via different channels. If one subimage is lost or damaged, it can be recovered based on the correlation between the subimages. The coding efficiency is low because a substantial part of the correlation in the image cannot be exploited by the coding scheme. A similar approach can be taken not in the spatial domain but in the frequency domain.

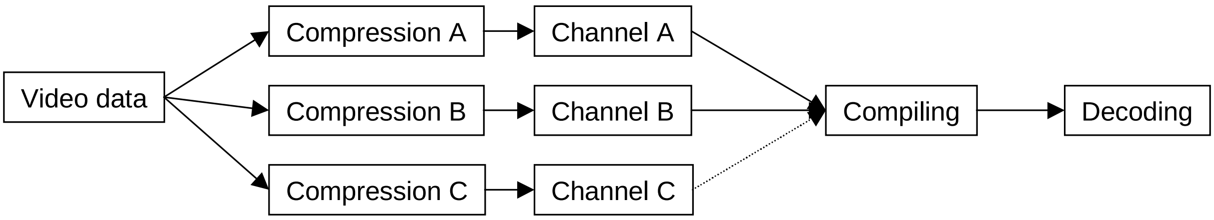


Figure 17 A simplified block scheme of the general multiple description coding.

Robust waveform coding [57][150]. In this technique, redundancy is added to increase the error resilience. Strictly speaking, layered coding and multiple description coding also belong to this category, as they both add some redundancy in the coded bit streams to provide the resilience to channel errors. Here we assume that the coded stream is transmitted over a single channel.

- One example is to send motion vectors for all blocks which can be used for error concealment when interpolation is necessary.
- One can also restrict the prediction domain, for instance confine it to one row of macro blocks, thereby reducing the impact of error propagation.

Robust entropy coding. This technique aims at decreasing the effect of errors in the very vulnerable variable length code words. There are several known techniques that accomplish this:

- A *synchronisation code word* [42][46][70][75][80][85][93] can be inserted. Such a word is unique and can be used to find a new starting point after loss of synchronisation. Such loss is very probable when variable length codes are used because once a code is corrupted it cannot be determined how long it was and therefore neither where the next word starts. Because the synchronisation word has to be unique and may not easily be emulated by mutilated parts of the video stream, it has to be large. Therefore, although it can be inserted every so many frames, or blocks, or bits, it can not be inserted too often.

- *Error resilient entropy coding* [36][65][112][132]. The variable length code words are rearranged in such a way that per block a fixed number of bits is used. The principle is shown in Figure 18. This makes resynchronisation possible without the use of synchronisation words, which are usually large. In this technique bit streams originating from individual blocks are distributed into slots of equal size, either fully or partially. Then, a predefined offset sequence is used to search for empty slots to place any remaining bits of blocks that are bigger than the slot size. This is done until all the bits are packed into one of the slots. At the decoder side the original data streams can be restored when the compression algorithm produces a coded stream in which it is known when the data of a block is complete. Furthermore the way the slots have been filled is known. The overhead of this technique turns out to be low.

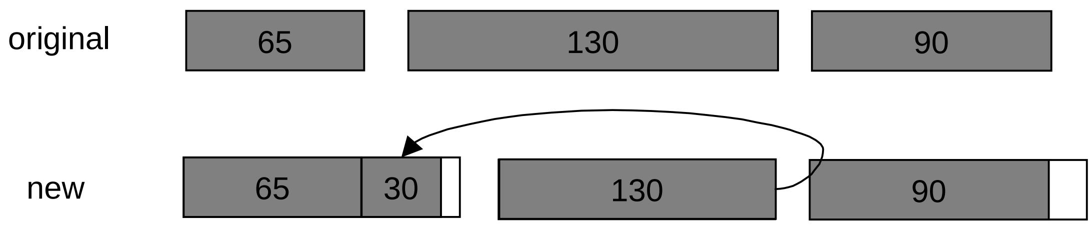


Figure 18 An illustration of the error resilient entropy coding technique. In the original situation, three parts of a stream exist, each one representing a block, which span 65, 130 and 90 bits respectively. In the new situation using error resilient entropy coding, the data is put into three equal slots of 100 bits. Now after each block resynchronisation is possible.

- *Reversible variable length codes*: such codes can be read either from right to left or left to right, which means that in the case of an error, one can advance to the next synchronisation word and then read back from there and decode up to the error. The principle is shown in Figure 19. This means that a lot of the otherwise lost data is recovered. Usually, there is no increase in bit rate because it is simply necessary to use a specific set of the different possible conversions to the variable length codes. If there are multiple errors between two synchronisation words, the data between the first and the last error is still lost.
- *Fixed length codes* [18]. When fixed length codes are used, the resynchronisation becomes easier because the starting point of the next code word is known, even when the current word cannot be interpreted. Since no entropy coding can be used the bit rate goes up and therefore this is only useful for extremely noisy channels. This technique does not work when data is lost and it is not known exactly how much is lost. This can, however, happen only in extreme situations. A combination of fixed and variable length codes can also be used. When doing so, in the region of the probability density function of the transform coefficients where the coefficients are almost equiprobable, that is around zero, the fixed length codes are used, while for the remaining coefficients, variable length code words are used. In the case of errors, the fixed length coded part of the data is available.

Different levels of error resilience [12] can be necessary when the channel conditions change over a longer period of time. This means for each of the described error resilience techniques an adaptation to the channel conditions is necessary.

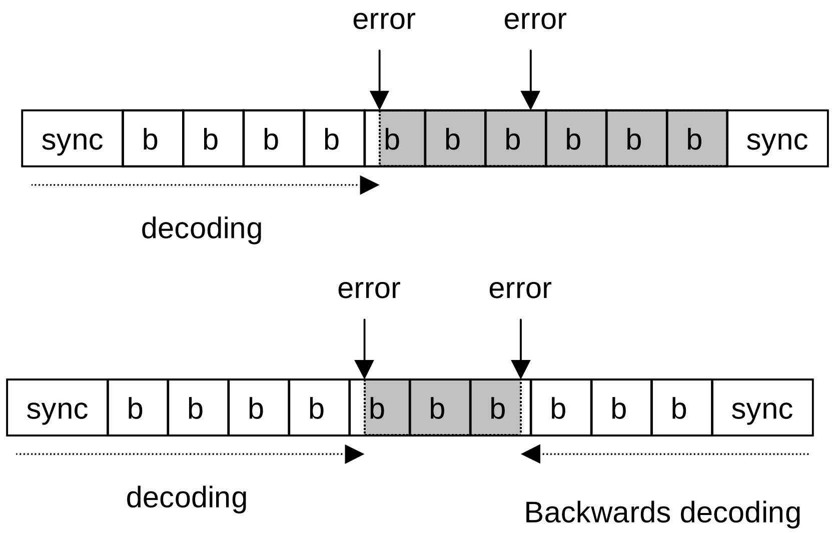


Figure 19 The principle of the use of reversible variable length codes.

4.2.2 Decoder Based

Here we give a description of existing decoder based error resilient source coding techniques. Most of them aim at recovering the damaged information and concealment of the effect of the errors.

Error detection [74][88][94][115]: many of the techniques described hereafter need to know if and where an error has occurred. This is a difficult problem, about which little is found in literature. Our new approach to this problem is described in Section 5.3.2. In general, the error can be detected in the link and network or in the decoder.

- Techniques to detect errors occurring in the link and network are well known and we discussed some of them in Section 3.4. It is, however, important that information about errors that have not been corrected successfully in the link is passed on to the video layer.
- At the decoder, the error can be detected in the decoding process in two places:
  - In the compression syntax; whenever inconsistent syntax is encountered, an error is assumed.
  - In the image; when features in the reconstructed image appear that look much like artefacts caused by errors in the channel, an error is assumed. For instance one can look for damage in a single DCT coefficient by examining the difference between the boundary pixels in a block and its four neighbour blocks. Assuming that the transition between blocks is smooth, the difference is small unless an error has occurred.

Motion compensated temporal prediction [48][67]. In this error concealment technique the decoder replaces the damaged macro block by its corresponding macro block in the previous frame, preferably motion compensated. In Figure 20 the corrupted block in the current frame, right, is replaced by the corresponding motion compensated block of the previous frame, left. Using the motion information is only possible if the motion vector corresponding to the macro block has remained uncorrupted. The technique for recovery of motion information in case this information is corrupted is described below.

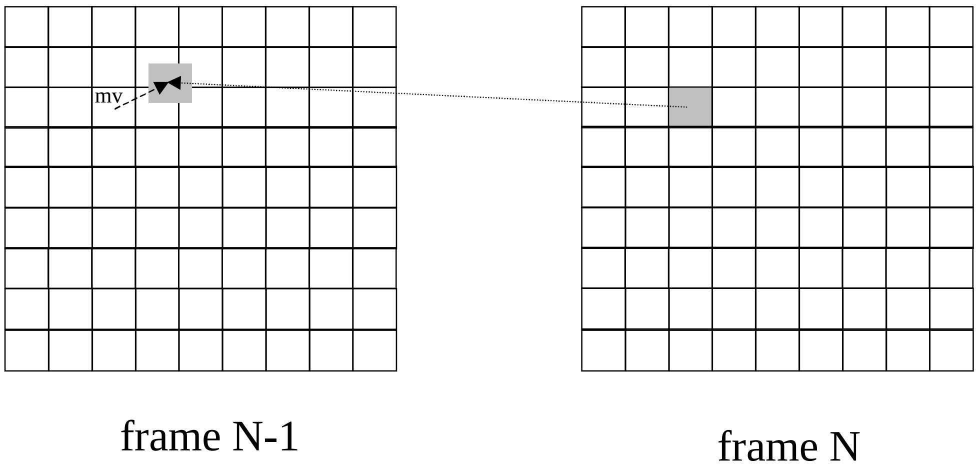


Figure 20 Error concealment in the motion compensated temporal prediction technique. The corrupted block in the current frame N, shown on the right, is replaced by the motion compensated corresponding block in the previous frame N-1, shown on the left.

Spatial and frequency interpolation [17][59][101][130]; the original macro block is as much as possible reconstructed by interpolation between the neighbouring macro blocks and by extrapolation from the previous macro block. This can be done either in the spatial or in the frequency domain. The principle is shown in Figure 21, where the interpolation in the same frame is shown on the left, while the additional extrapolation from the previous frame is shown on the right.

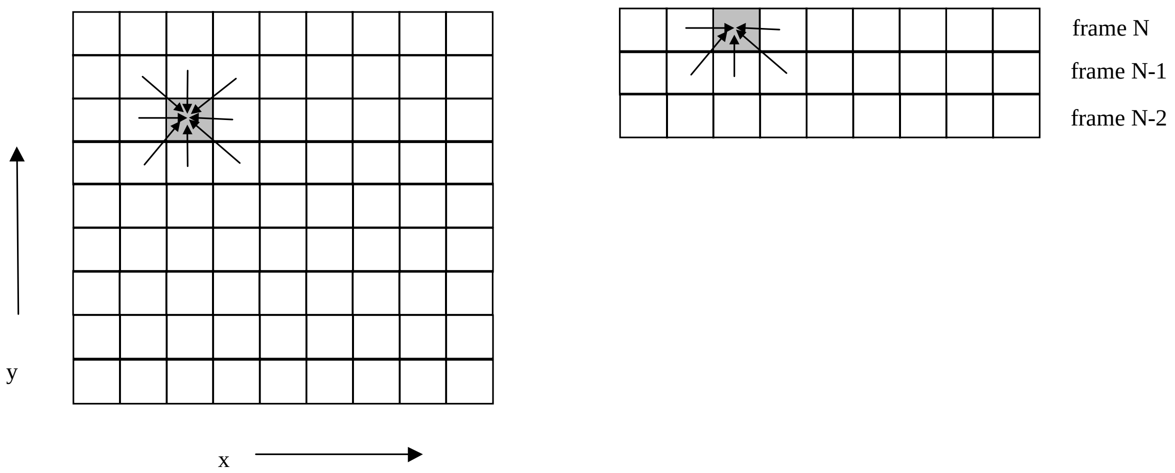


Figure 21 The technique of spatial and frequency interpolation. On the left is shown the current frame where the corrupted or missing data of the corrupted block is recovered by interpolation between neighbouring blocks. On the right one row is shown of the current frame and two previous frames. Data recovery is done by extrapolation from the block and neighbouring blocks, all in the previous frame.

Recovery of coding mode and motion information [55][76][92][130]. The same can be done as in spatial and frequency interpolation but now for recovery of the coding mode and motion vector information. The corrupted information is replaced by interpolated or extrapolated data from neighbouring blocks or from blocks of the previous frame.
Maximally smooth recovery [72][148][160]. This technique works on a block by block basis. To estimate the missing DCT coefficients in a block, it minimises a measure of spatial and temporal variation between adjacent pixels in the current block and in the spatially and temporally neighbouring blocks so that the resulting estimated video signal is as smooth as possible. In other words, in the current block for each pixel its neighbour is taken in a certain direction, and the difference is added to the measure which is to be minimised.

Projection onto convex sets [131]. The convex sets are derived through the requirement that the recovered block must have a limited bandwidth, either isotropically for a block in a smooth region or along a particular direction for a block containing a straight edge. With this method first a combined block is formed by including eight neighbouring blocks with the damaged block, as shown in Figure 22. Now this combined block is subject to an edge existence test by using the Sobel operator. The block is then classified either as a monotone block or as an edge block. Then two projection operators are applied to the combined block. The first operator enforces a limited bandwidth constraint. The second projection operator enforces a range constraint and truncates the output value. These two projection operations are applied alternatingly until the block does not change any more under further projections.

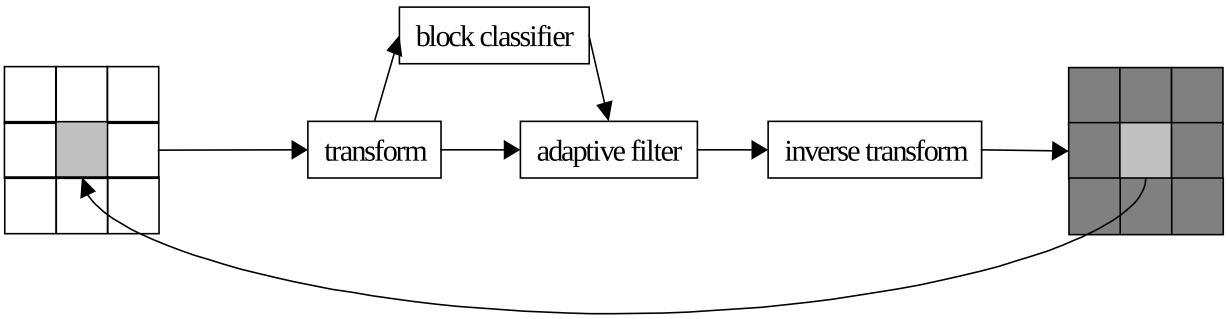


Figure 22 A simplified block scheme for the use of projection onto convex sets.

Fuzzy logic [79]. This can also be used. It is most useful to aid the recovery of high frequency components that normally cannot be recovered by the techniques presented above. However, this is computationally very intensive.

4.2.3 Interactive Techniques

Some error resilient coding techniques use both the encoder and the decoder. An important feature that can be exploited then is the ability of the encoder and decoder to interact with each other.

Back channel [61]. A link from the decoder back to the encoder is used to exchange information about the reception of the video data at the decoder. It can also be used to send information about the channel conditions to the encoder.

Selective encoding [91][143][154]. In the encoding process only data is used that is uncorrupted. There are three ways to implement this technique. They are shown schematically in Figure 23.

- After an error has occurred, the damaged area can be requested by the decoder to be intracoded in the first possible frame. After the reception of the intracoded image area the effect of any error in this area is eliminated because the intracoded macro blocks of which the area consists do not use any information from the previous frame. To do this properly, the whole affected area should be intracoded. In order to know which macro blocks have been affected by the error propagation, information like the motion vectors have to be taken into account.
- After an error has been detected at the decoder and information about this error has been sent back to the encoder, the affected area is not longer used for prediction until the block is intracoded again.
- A third method performs error concealment at the encoder in exactly the same way as the decoder. This will minimise the difference between their data. Since there is a delay between the encoder and decoder, this requires substantial recalculation of the data by the encoder.

The efficiency of these methods depends on the delay between the decoder and encoder and back.

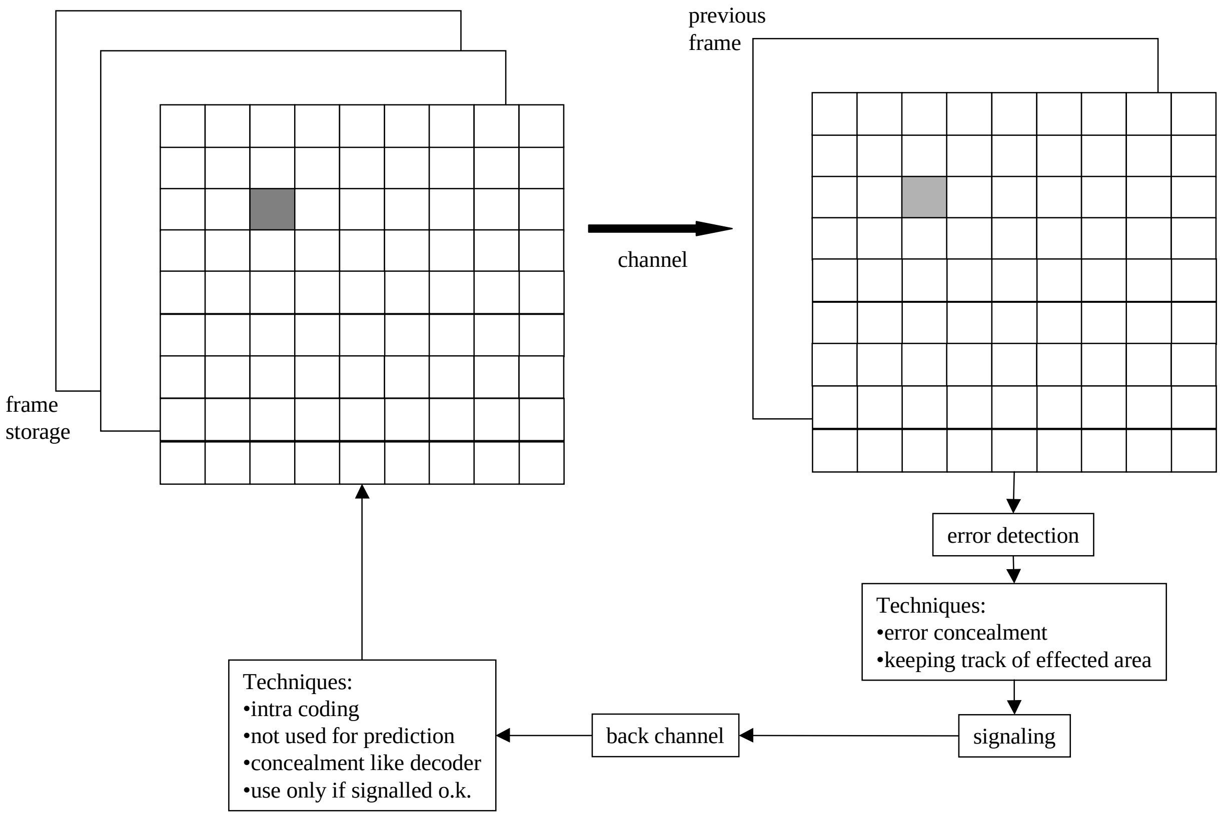


Figure 23 A simplified block scheme for selective encoding, data selection and retransmission without waiting.

Data selection [16]. In this technique, only data is used for prediction that has been signalled as being received without errors by the decoder. In Figure 23 we show the error detection, signalling and back channel that are used by this technique. The signalling can be done frame based or block based. Storage of several frames is necessary, how many depends on the delay.

Retransmission without waiting [49][159]. In this technique, the damage to a block is signalled to the encoder, but after concealment of the error the decoder proceeds, while keeping track of all the damaged pixels. Upon receipt of the new data, some time later, the correct image is calculated, which can be quite complex. The difference with selective encoding is that this is a retransmission of old but corrupted data, while selective encoding does not retransmit but changes the coding mode for future data for the same area in the frame. This technique is also shown in Figure 23.

Prioritised multi-copy retransmission [144]. In this technique multiple copies of the lost data are sent in a single network retransmission trial, as shown in Figure 24. In this figure we see a data packet which has an error after passing through the channel. The smallest retransmittable data size containing the error is requested to be retransmitted by the receiving side. This is then copied two times to be sent in one retransmission through the same channel. If there is an error in one or two of them, the data can still be decoded. This technique is efficient when not many retransmissions are allowed to avoid a long delay. This is especially important for situations where there are many unreliable links to pass in the communication link and many retransmissions would increase the delay too much. This can be the case for instance for internet video. A simplified block scheme is shown in Figure 24.

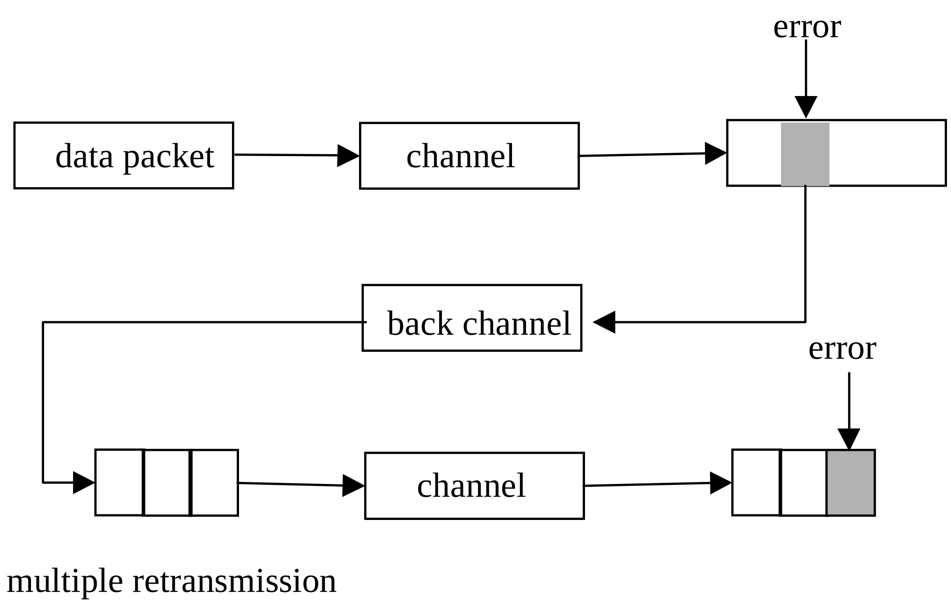


Figure 24 A simplified block scheme of the prioritised multi-copy retransmission technique.

4.2.4 Joint Source Channel Coding

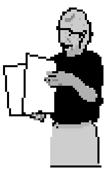
The error resilience techniques described in the previous sections can be combined into joint source channel coding. We can indicate two main approaches:

- **Adaptive bit allocation** [45][71][129][141]. The allocation of the available bits and the code-word mapping can be based on channel statistics. This way efficient coding can be achieved in the absence of errors, adding error resilience only where and when necessary.

- **General approach** [23]. Several theoretical approaches to the problem of optimising the joint source channel coding exist. We mention one interesting attempt here [23]. A channel is modelled by the bit error rate and code rate. For the source coder the rate distortion curves are available. This means that there is a common parameter determining the performance of the joint source channel coding, namely the bit error rate. This combination yields the distortion as a function of energy and rate, thereby simplifying the joint optimisation to the choice of the optimal energy and source coding rate for each symbol to minimise the distortion. The problem of optimising the joint source channel coding, however, is very complex. The solutions are therefore usually either highly theoretical or highly experimental. The theoretical solutions, however, mostly can not easily be applied real applications due to the necessary simplifications. We chose to follow an experimental approach and developed a new method to address the problem of the optimisation of joint source channel coding which is described in Chapter 6.

4.2.5 MPEG-4 Error Resilience

The recent standard for object based compression of digital video, MPEG-4, also incorporates some error resilience techniques [70]. Figure 25 outlines the basic approach of the MPEG-4 video algorithm to encode rectangular as well as arbitrarily shaped objects.



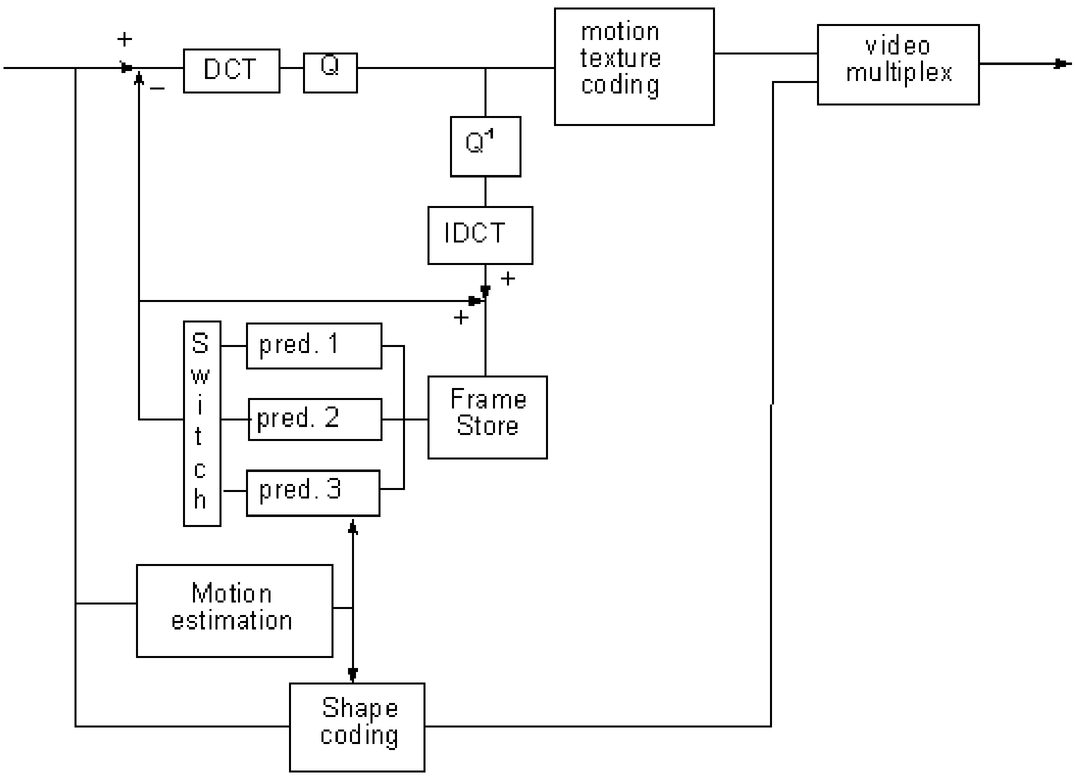


Figure 25 The outline of the MPEG-4 compression scheme.

The major difference with well-known hybrid transform coders is the ability to encode arbitrarily shaped objects, for which shape compression is also necessary. There are no error resilience tools in MPEG-4 specifically for the shape coding. The more general error resilience tools developed for MPEG-4 can be divided into three major areas: resynchronisation, data recovery, and error concealment. They will now be described.

Resynchronisation

The resynchronisation tools of MPEG-4 attempt to enable resynchronisation between the decoder and the bit stream after a residual error or errors have been detected. Generally, the data between the synchronisation point prior to the error and the first point where synchronisation is re-established is discarded. If the resynchronisation approach is effective at localising the amount of data discarded by the decoder, then the ability of other types of tools that recover data or conceal the effects of errors is greatly enhanced.

The resynchronisation approach adopted by MPEG-4, referred to as a video packet approach, is similar to the group of blocks structure utilised by the ITU-T standards of H.261 and H.263. In these standards a group of blocks is defined as one or more rows of macro blocks. At the start of a new group of blocks, certain information, called a group of blocks header, is placed within the bit stream. This header information contains a group of blocks start code, which is different from a picture start code, and allows the decoder to locate this group of blocks. Furthermore, the group of blocks header contains information which allows the decoding process to be restarted, that is resynchronise the decoder to the bit stream and reset all predictively coded data. The group of blocks approach to resynchronisation is based on spatial resynchronisation. That is, once a particular macro block location is reached in the encoding process, a resynchronisation marker is inserted into the bit stream. A potential problem with this approach is that since the encoding process produces a variable bit rate, these resynchronisation markers will most likely be unevenly spaced throughout the bit stream. Therefore, certain portions of the scene, such as high motion areas, will be more susceptible to errors, which will also be more difficult to conceal.

The video packet approach adopted by MPEG-4 is based on providing periodic resynchronisation markers throughout the bit stream. In other words, the length of the video packets are not based on the number of macro blocks, but on the number of bits contained in that packet. If the number of bits contained in the current video packet exceeds a predetermined threshold, then a new video packet is created at the start of the next macro block.

A resynchronisation marker is used to distinguish the start of a new video packet. This marker is distinguishable from all possible variable length code words as well as from the video object plane start code. In the MPEG-4 object-based coder, the video object plane is the equivalent of the frame in the non-object based coders, like H.263. Here, too, header information is provided at the start of a video packet. Contained in this header is the information necessary to restart the decoding process and it includes: the macro block number of the first macro block contained in this packet and the quantisation parameter necessary to decode that first macro block. The macro block number provides the necessary spatial resynchronisation while the quantisation parameter allows the differential decoding process to be resynchronised.

Also included in the video packet header is the header extension code. The header extension code is a single-bit that, when enabled, indicates the presence of additional resynchronisation information. It includes the modular time base, the video object plane temporal increment and the video object plane prediction type. This additional information is made available when the video object plane header has been corrupted.

It should be noted that when utilising the error resilience tools within MPEG-4, some of the compression efficiency tools are modified. For example, all predictively encoded information must be confined within a video packet so as to prevent the propagation of errors.

In conjunction with the video packet approach to resynchronisation, a second method called fixed interval synchronisation has also been adopted by MPEG-4. This method requires that video object plane start codes and resynchronisation markers, that is the start of a video packet, appear only at legal fixed interval locations in the bit stream. This helps avoiding the problems associated with start code emulations. That is, when errors are present in a bit stream it is possible for these errors to emulate a video object plane start code. In this case, when fixed interval synchronisation is utilised the decoder only needs to search for a video object plane start code at the beginning of each fixed interval. The fixed interval synchronisation method extends this approach to any predetermined interval.

Data Recovery

After synchronisation has been re-established, data recovery tools attempt to recover data that normally would be lost. These tools are not simply error correcting codes, but techniques for encoding the data in an error resilient manner. For instance, one particular tool that has been adopted by MPEG-4 is reversible variable length codes. In this approach, the variable length code words are designed such that they can be read in the forward as well as in the reverse direction.

An example illustrating the use of a reversible variable length code is given in Figure 26. We see the same principle as in “reversible variable length codes” of Section 4.2.1 and Figure 19, but now adapted to the MPEG-4 context. In our example, a burst of errors has corrupted a portion of the data. Generally in such a situation, all the data between the two synchronisation points would be lost. However, as shown in the figure, a reversible variable length code makes it possible to recover some of that data. It should be noted that the parameters *MB address*, *QP* and *HEC* shown in the figure represent the fields reserved in the video packet header for the macro block address, quantisation parameter and the header extension code, respectively.

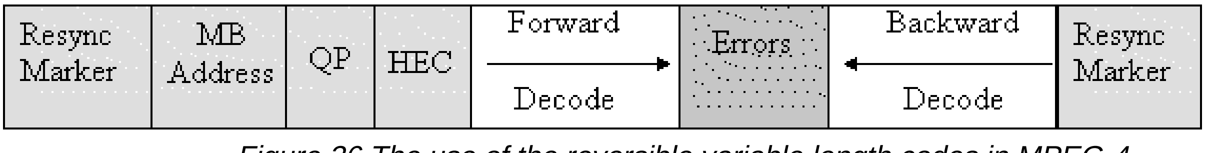


Figure 26 The use of the reversible variable length codes in MPEG-4.

Error Concealment

Similar to the error resilience tools discussed above, the effectiveness of an error concealment strategy is highly dependent on the performance of the resynchronisation scheme. Basically, if the resynchronisation method can effectively localise the error, then the error concealment problem can be solved more easily. For low bit rate, low delay applications the resynchronisation scheme of MPEG-4 provides very acceptable results with a simple concealment strategy, such as copying blocks from the previous frame. However, MPEG-4 has developed an additional error resilient mode that further improves the ability of the decoder to localise an error. Specifically, this approach utilises data partitioning as it separates the motion and the texture. This approach requires the insertion of a second resynchronisation marker between motion and texture information. If the texture information is lost, the approach utilises the motion information to conceal these errors. That is, due to the errors the texture information is discarded, while the motion information is used for motion compensation of the previous decoded video object plane.

4.3 Remaining Problems and Conclusion

We have described the main problem we encounter in compression of digital video data in mobile wireless communication; the occurrence of errors in the encoded bit stream during transmission. We defined as the optimisation criterion the quality of the image that is displayed at the receiving side while not increasing the value other important parameters like the bit rate. This criterion determines, to a certain extent, the approaches we take in the next chapters to find solutions for dealing with transmission errors.

We have addressed many techniques which aim at dealing with transmission errors, at different points in the communication path. Some of them are efficient in certain situations. However, these existing techniques still show some deficiencies, especially when we look at the situation of Mobile Multi-media Communication project. Therefore for some of them we developed new techniques in our research on error resilient video compression. Specifically, the following remaining problems are addressed in the next chapters.

- For most techniques, knowing that an error has occurred and where, is essential. This means that for using such techniques, an effective error detection algorithm is necessary. However, not many efficient error detection algorithms are available. We address this problem and present a solution in Chapter 5.
- An alternative is to use an error resilience technique that is independent of the knowledge of the presence and position of errors. We developed such a technique that we present also in Chapter 5.
- Another subject of research that generally has not be addressed much, is the trade-off between source coding and channel coding in the presence of transmission errors. We address this problem and evaluate it for our mobile multi-media communication situation in Chapter 6.
- The problem of error resilience for shape coding generally has been addressed only marginally. We present a shape coding technique that is error resilient and address its error resilience both qualitatively and quantitatively in Chapter 7.

.

Chapter Five

5 Error Resilient Modified H.263[2]

In this chapter we describe a new error resilient video compression technique that we have developed. It is based on the H.263 standard. We start in Section 5.2 by looking at the influence of errors on a simple differential pulse code modulation (DPCM) system. Then we proceed to a H.263 based coding algorithm. In Section 5.3 the different parts of our error resilient video compression technique are described. The results of experiments can be found in Section 5.4, including a comparison with MPEG-4 results. Section 5.5 concludes this chapter.

[2] This Chapter is based on paper [125] by Spaan et alteres.

5.1 Introduction

When wireless digital video communication is used, transmission errors occur in the compressed video data stream. To counter the effect of such errors, we developed a new error resilient video compression technique, based on the H.263 standard.

Most of the existing error resilient source coding techniques we have described in Section 4.2 aim at reconstructing the damaged signal, that is they try to reduce the corruptions after the damage has been done. We in turn tried to find a way to make the algorithm more robust in the sense that if an error occurs, the effect of the error will be smaller than it would be in the original decoding algorithms.

The main reason for the devastating effect of errors in the encoded stream is that most information is coded with respect to the previous frame. That means that if anything goes wrong in the current frame, all following frames will be damaged. One way to avoid this is using intracoding only, as has been suggested in literature, but this increases the bit rate substantially. The way we reduce the impact of the errors, is by using for the prediction not one frame, but an average of a number of previous frames. An error in one frame will then be smoothed by the averaging. A side effect is that the prediction is worse and the bit rate increases.

In order to investigate the viability of this technique, we first analyse the influence of the errors. However, since a theoretical approach of a complete hybrid codec is difficult, we first considered the influence of errors on a DPCM-like compression system. From the obtained results we developed an error resilience technique which is incorporated in our proposed error resilient version of an H.263 codec. In this technique, a composed prediction frame is used, with respect to which the differential frames are coded. This frame is composed of a number of previous decoded frames.

Furthermore, since our main objective is maintaining the quality of the *displayed frames*, we examined a technique which detects errors in the frame to be displayed, unlike other techniques, which aim to maintain the quality of the *link*. The positions of the errors are detected and sent to the encoder. When the encoder has received this information, some frames later, it will intracode the macro blocks that were indicated by the decoder. Since there probably will be a delay between the detection of the error and the reception of the intracoded macro block, error concealment is applied in the meantime.

The proposed technique is based on a standard hybrid codec [31] like H.263 [16] and requires as additional features one or more frame stores at the encoder and the decoder and a feedback channel with low requirements. The main additional computational load is at the decoder.

Hereafter, we first discuss the influence of errors on a DPCM system. Then we describe the different parts of our error resilient video compression technique. Finally we discuss the results of the experiments.

5.2 Differential Pulse Code Modulation

In this section we first calculate the variance of a reconstructed DPCM compressed signal in the presence of errors. Then we address a more special case for which we numerically optimise the DPCM coefficients given bit error rate.

5.2.1 Analysis

We consider a system consisting of a DPCM encoder and decoder, connected through a transmission channel, which introduces errors. A simplified block diagram is shown in Figure 27. The input signal that is fed into the system is *f(n)* and the difference signal that is to be quantised is *Δf(n)*. After quantisation the values are assigned variable length code words, as shown by the entropy coding block.

The coded signal is now transmitted through the channel which will introduce bit errors with a certain bit error rate. At the decoder entropy decoding and inverse quantisation are applied after which the DPCM decoding is applied, yielding the reconstructed signal $\tilde{f}(n)$.

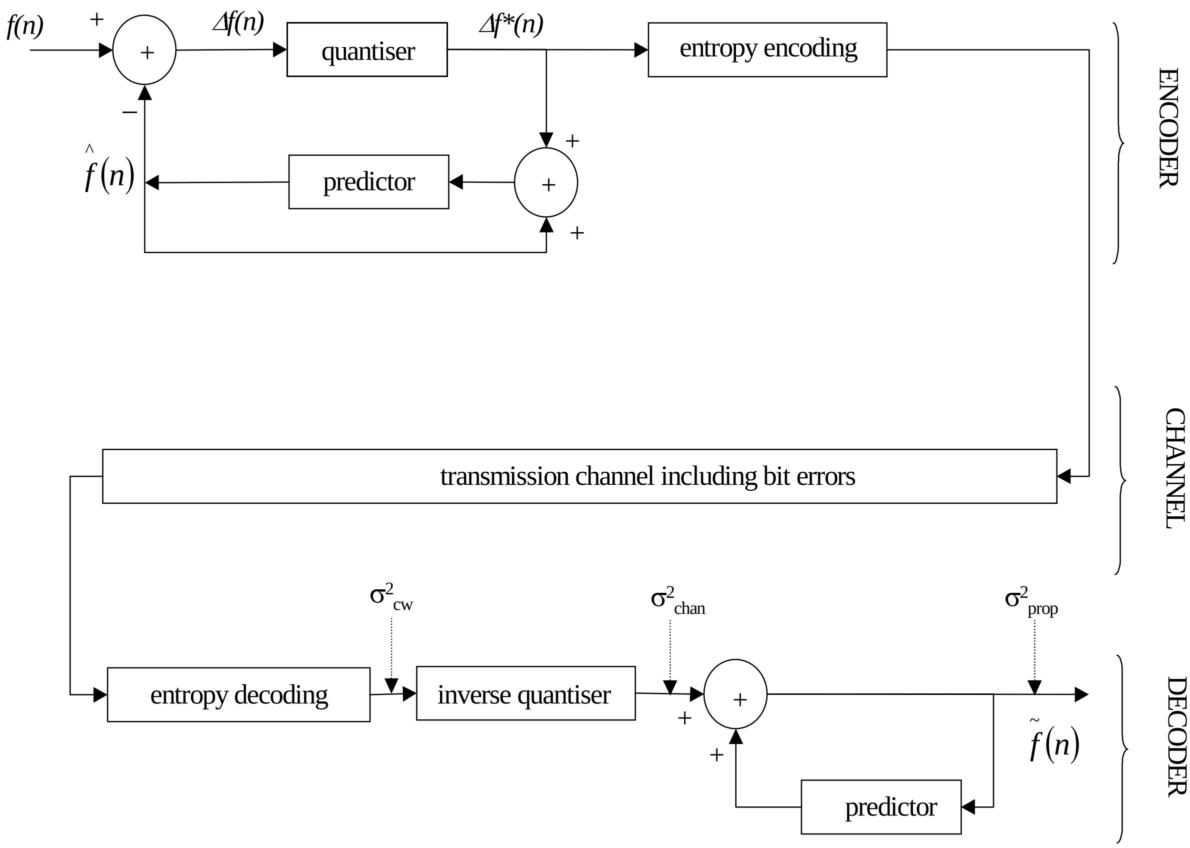


Figure 27 The DPCM coding system including transmission errors.

We now analyse this DPCM system. The aim of the analysis is to find the DPCM coefficients for which the contribution of the quantisation and transmission errors to the variance of $\tilde{f}(n)$ is minimal, given a bit error rate. To facilitate our analysis we first define the variance of the errors at different locations in the decoding part of the coding scheme shown in Figure 27. The contribution due to the transmission errors to the variance of the reconstructed signal at the decoder we call σ^2_{prop}. The same contribution but now right after inverse quantisation we call σ^2_{chan} and the contribution right after decoding of the code words we call σ^2_{cw}.

We assume that the effect of transmission errors is additive. The total variance of the errors in the reconstructed signal at the decoder σ^2_{tot} is then the sum of the variance caused by the quantisation errors and the variance caused by the transmission errors:

(1) $$\sigma_{tot}^2 = \sigma_q^2 + \sigma_{prop}^2$$

We treat these two contributions to the variance separately. We start with the quantisation errors.

Quantisation errors

We take a uniform quantiser with a quantisation step size of *S*. From literature [31] we know that the following approximation is valid for a wide range of bit rates:

(2) $$\sigma_q^2 = \frac{S^2}{12}$$

However, because the effect of errors is dependent on the number of bits in the compressed stream we want to have a constant bit rate *R*. In order to realise this we scale the quantiser to a signal of variance one. Its quantisation step size, now corresponding to the signal of variance one, we call S_{proto}

(3) $$S = S_{proto}\sigma_{\Delta f}$$

where $\sigma_{\Delta f}$ is the variance of the DPCM coded signal before quantisation as indicated in Figure 27. We can do this because we have the following relation known in literature [31]:

(4) $$R = \frac{{}^2\log(2\pi e)}{2} - {}^2\log\left(\frac{S}{\sigma_{\Delta f}}\right)$$

We have now obtained the following result for the contribution of the quantisation errors to the variance of $\tilde{f}(n)$:

(5) $$\sigma^2{}_q = \frac{S_{proto}^2\sigma^2{}_{\Delta f}}{12}$$

Having obtained this result, we now proceed to the determination of the contribution to the variance due to the transmission errors σ^2_{prop}.

Transmission errors

At the decoder side, the codewords, including transmission errors, are first decoded and inverse quantised. At this position in the decoding scheme, the contribution to the variance due to the transmissions errors is σ^2_{chan}. Following the decoding scheme the signal is then reconstructed using the DPCM system, which has a transfer function we call $H(\omega)$. This yields for the contribution to the variance due to transmission errors:

(6)
$$\sigma^2_{prop} = \sigma^2_{chan} \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{\left|1 - \sum_{m=1}^{A} h_m e^{(-j\omega m)}\right|^2} d\omega$$

We still do not know σ^2_{chan} but we know it is related to the variance caused by the transmission errors before inverse quantisation through

(7)
$$\sigma^2_{chan} = S^2 \sigma^2_{cw}$$

and we already now S^2 from equation (3).

As an intermediate result we can now combine the models for σ^2_q and σ^2_{prop} which yields the overall DPCM reconstruction error

(8)
$$\begin{aligned}\sigma^2_{tot} &= \sigma^2_q + \sigma^2_{prop} \\ &= S^2_{proto}\sigma^2_{\Delta f}(h_m)\left[\frac{1}{12} + \frac{\sigma^2_{cw}}{2\pi} \int_{-\pi}^{\pi} \frac{1}{\left|1 - \sum_{m=1}^{A} h_m e^{(-j\omega m)}\right|^2} d\omega\right]\end{aligned}$$

Here $\sigma^2_{\Delta f}(h_m)$ denotes the prediction error variance that is a function of the prediction coefficients h_m.

However, having obtained this intermediate result, we still have to find expressions for σ^2_{cw} and $\sigma^2_{\Delta f}$. We start with determination of σ^2_{cw} for which we have the following relation:

(9)
$$\sigma^2_{cw} = E[N(n)^2] - E[N(n)]^2$$

Here $N(n)$ is a stochastic process describing the errors in the amplitudes after decoding the code words, which can be corrupted. We assume that we know when an error has occurred and that all corrupted code words are mapped to the normalized signal amplitude zero. We can assume that $N(n)$ is an integer-valued independent identically distributed process with zero mean. The probability mass function of N has been derived to be

(10) $$P(N=i)=(1-(1-P_{ber})^{l_i})P(Cw=i)$$

The second part of the right half of equation (10) is the probability that a codeword represents quantiser level *i*. The first part is the probability that there is an error in the codeword which has length l_i. This part also includes the parameter P_{ber} determining the bit error rate. The length of the code word is

(11) $$l_i \approx -\lfloor \log_2 P(Cw=i) \rfloor$$

In order to determine the second part of (10), we assume an odd uniform quantizer and a Gaussian-distributed prediction error signal. The probability mass function for the code word of the corresponding integer-valued signal amplitudes is then

(12) $$P(Cw=i)=P(Cw=-i)=\int_{\frac{2i-1}{2}S_{proto}}^{\frac{2i+1}{2}S_{proto}} \frac{1}{\sqrt{2\pi}} e^{(-\frac{u^2}{2})} du \qquad i=0,1,2,...$$

Having obtained σ^2_{cw} we now proceed to determine the variance of the prediction error $\sigma^2_{\Delta f}$. This is straightforwardly given as

(13) $$\begin{aligned}\sigma^2_{\Delta f}(h_m) &= E\left[\left(f(n)-\sum_{m=1}^{A} h_m f(n-m)\right)^2\right] \\ &= \sigma_f^2 - 2\sum_{m=1}^{A} h_m R_f(m) + \sum_{k=1}^{A}\sum_{m=1}^{A} h_k h_m R_f(m-k)\end{aligned}$$

where $R_f(n)$ is the autocorrelation function of $f(n)$.

Combining equations (8), (9), (10), (11), (12) and (13), we have as the final result a system to describe the overall reconstruction error σ^2_{tot}. In order to obtain the best quality of the reconstructed signal, we need to minimize σ^2_{tot} with respect to the prediction coefficients h_m for a given bit error rate, a given bit rate determined by the quantizer step size, and a given correlation of the input signal.

However, we can not find an analytical solution for this system except for the case that the bit error rate is equal to zero. In that case, however, the minimization of equation (8) leads to the well-known Yule-Walker equations [31]. We therefore have to turn to a numerical approach for optimisation of the DPCM coefficients.

5.2.2 Numerical Optimisation

As we have seen, the set of equations we have so far still does not enable us to minimise the total variance σ^2_{tot} as a function of the DPCM coefficients and the bit error rate. Therefore we now have to make some assumptions.

We first model the input signal as a first-order autoregressive process with autocorrelation coefficient ρ.

(14) $$R_f(n) = \sigma_f^2 \rho^{|n|}$$

Furthermore, we use a predictor structure in which the predictor contains only two prediction coefficients, yielding the following relation:

(15) $$\hat{f}(n) = h_1 f(n-1) + h_2 f(n-2)$$

For the selected signal model, that is the autoregressive process with autocorrelation coefficient ρ, the optimal prediction coefficients are simply given as

(16) $$h_1 = \rho$$
$$h_2 = 0$$

Now we have a set of equations and assumptions that enable us to perform the numerical optimisation of the prediction coefficients given a bit error rate and correlation of the input signal.

Results

The actual values of the results depend on the specific choice of the input signal which makes comparison between results difficult. However, to avoid this problem we present our results in terms of the variance based signal-to-noise ratio, given as

(17) $$VSNR = 10 \cdot {}^{10}\log\left(\frac{\sigma_\phi^2}{\sigma_f^2}\right) \text{ (dB)}$$

In this equation σ^2_ϕ stands for the variance of the difference between the input signal at the encoder and the corrupted reconstructed signal at the decoder. This way we can disregard in all experiments the actual variance σ^2_f because we look at the changes relative to this variance.

The figures that follow show the variance based signal-to-noise ratio in a two-dimensional contour plot as a function of h_1, horizontal, and h_2, vertical. In the figures the upper right half is empty because for some values of h_1 and h_2 the system does not converge since the DPCM prediction filter should be stable and have all poles within the unit circle.

Through our simulations we obtained the following results. Figure 28 and Figure 29 show the variance based signal-to-noise ratio of the reconstructed signal for the case that the bit error rate is equal to zero but for different values for the correlation of the input signal. In Figure 28 we have $\rho = 0.5$, low correlation, in Figure 29 we have $\rho = 0.92$, high correlation. From these figures we clearly see that the optimum of the variance based signal-to-noise ratio is at $h_2 = 0$ and $h_1 = \rho$, as expected. In Figure 28 the contour lines are at a distance of approximately 0.05 dB with a maximum of 12.04 dB, and in Figure 29 at a distance of 0.13 dB with a maximum of 18.92 dB. In both cases the bit rate is 2 bits per point.

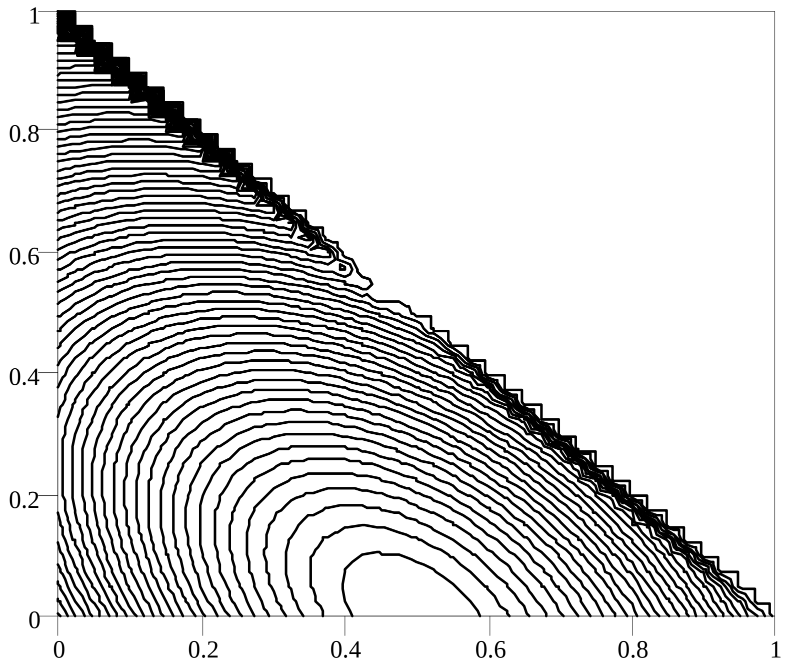


Figure 28 A contour plot showing the variance based signal-to-noise ratio as a function of h_1, horizontal, and h_2, vertical. In this plot the bit error rate is zero and the correlation low, $\rho = 0.5$.

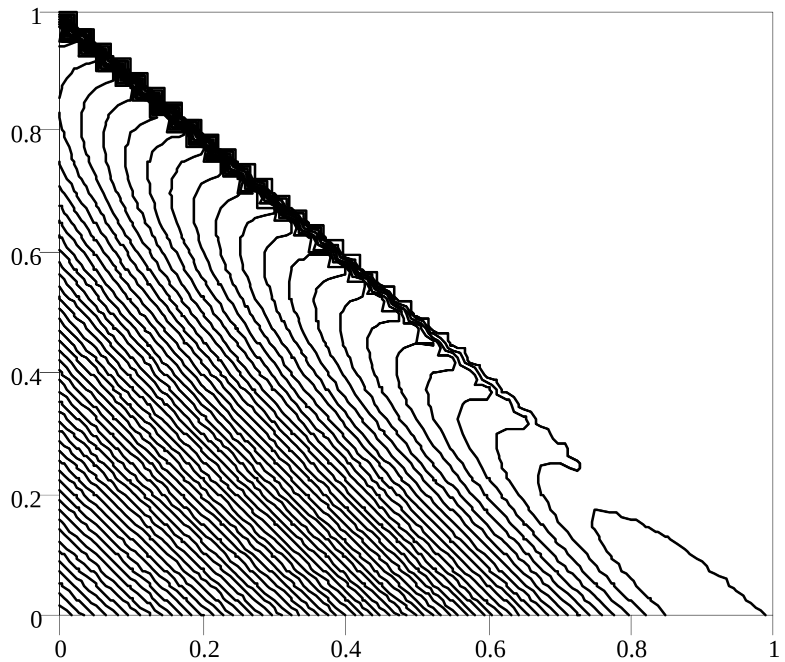


Figure 29 A contour plot showing the variance based signal-to-noise ratio as a function of h_1, horizontal, and h_2, vertical. In this plot the bit error rate is zero and the correlation high, $\rho = 0.92$.

In the following four figures we show the variance based signal-to-noise ratio for a bit error rate of 0.001 and 0.01, again for the two different cases of the autocorrelation coefficient and for a bit rate of 2.0 bits per point.

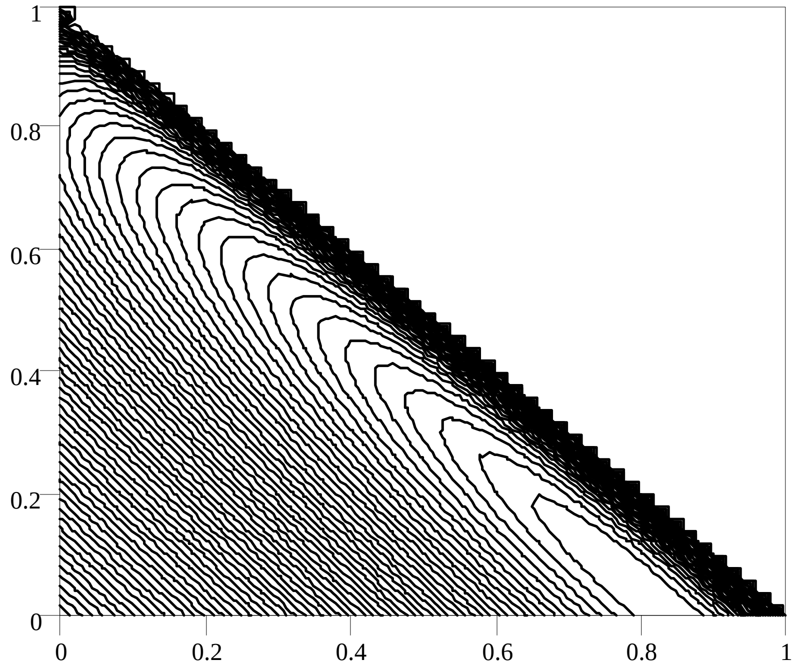


Figure 30 A contour plot showing the variance based signal-to-noise ratio as a function of h_1, horizontal, and h_2, vertical. In this plot the bit error rate is 0.001 and the correlation high, $\rho = 0.92$. The optimum is found at $h_1 = 0.80$ and $h_2 = 0.04$. The bit rate is 2 bits per point.

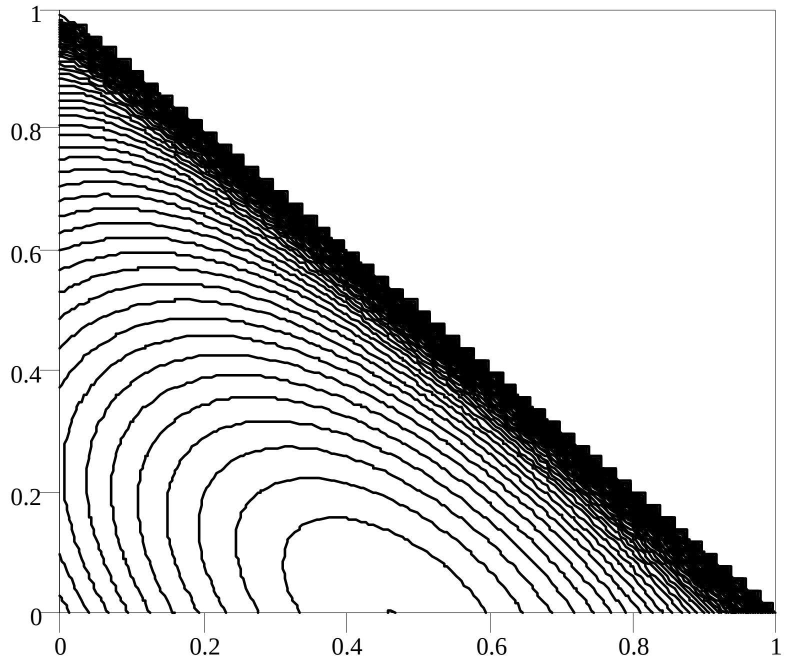


Figure 31 A contour plot showing the variance based signal-to-noise ratio as a function of h_1, horizontal, and h_2, vertical. In this plot the bit error rate is 0.001 and the correlation low, $\rho = 0.50$. The optimum is found at $h_1 = 0.46$ and $h_2 = 0$. The bit rate is 2 bits per point.

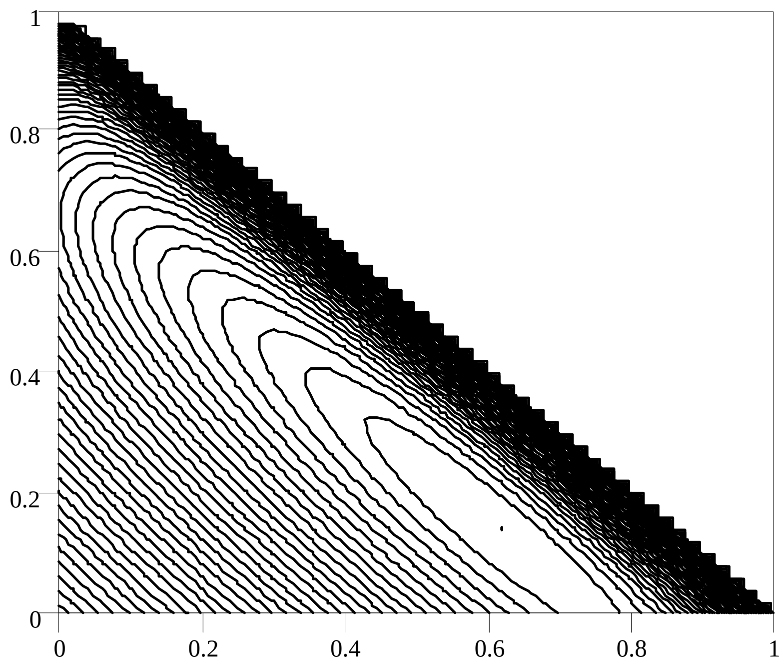


Figure 32 A contour plot showing the variance based signal-to-noise ratio as a function of h_1, horizontal, and h_2, vertical. In this plot the bit error rate is 0.01 and the correlation high, $\rho = 0.92$. The optimum is found at $h_1 = 0.62$ and $h_2 = 0.14$. The bit rate is 2 bits per point.

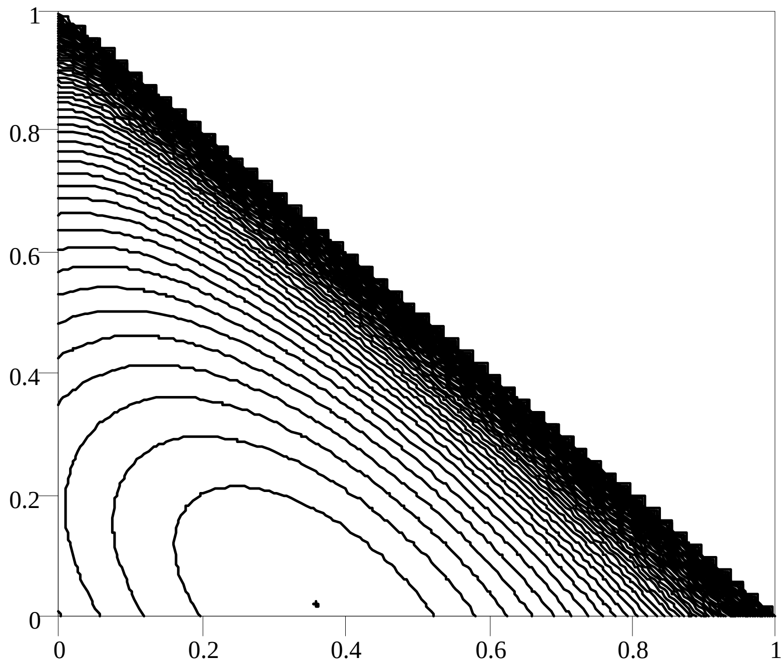


Figure 33 A contour plot showing the variance based signal-to-noise ratio as a function of h_1, horizontal, and h_2, vertical. In this plot the bit error rate is 0.01 and the correlation low, $\rho = 0.50$. The optimum is found at $h_1 = 0.36$ and $h_2 = 0.02$. The bit rate is 2 bits per point.

From these experiments we can conclude that the impact of the presence of channel errors is more severe for highly correlated signals than for less correlated signals.

From now on, we therefore concentrate on the case of high correlation; that is ρ = 0.92.

Figure 30 and Figure 32 clearly indicate that the optimal values for the prediction coefficients vary with the bit error rate. In the next experiment we consider the effect of a varying bit rate, by using three different bit rates, 0.7, 2 and 4 bits per point. Through numerical optimisation using a brute force optimization procedure, we determined the optimal values for the coefficients h_1 and h_2 for bit error rates of 0, 0.001, 0.01 and 0.1. The resulting optimal coefficients are plotted in Figure 34.

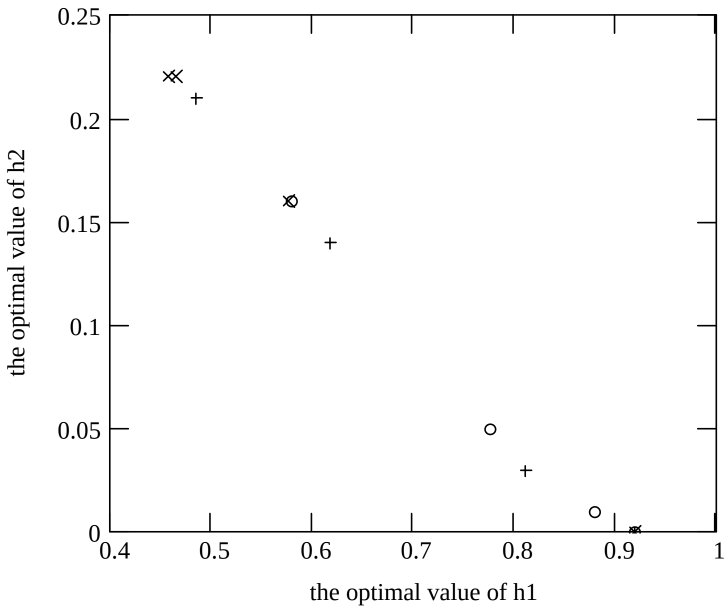


Figure 34 The optimal values of h_1 and h_2 for different bit rates and bit error rates. Shown are three series of four points. The series correspond to a bit rate of 4 bits per point, denoted by the 'x' symbol, a bit rate of 2 bits per point, denoted by the '+' symbol, and a bit rate of 0.7 bits per point, denoted by the 'o' symbol. The four points in each series are generated using, from left to right and top to bottom, bit error rates of 0.1, 0.01, 0.001 and 0.

Though we vary over a wide bit rate range, clearly the trend in all cases is the same, namely:

- With increasing bit error rate, more emphasis is placed on the sample weighted with h_2, in other words more information from the past is used. The reason for this is that the effects of channel errors need to be averaged out more and more to maintain a good signal quality.
- With increasing bit error rate, the sum of h_1 and h_2 becomes smaller, which means that information from the past is less relied on. The reason for this is that error propagation becomes increasingly dominant, which can only be stopped off by decreasing the impact of the prediction.

Though limited in scope, we can conclude from this experiment that with increasing bit error rate more information from the past should be taken into account in the DPCM predictor. Therefore, the predictor in the above experiment should be replaced by a higher order version that includes more prediction coefficients. Another conclusion is that when the bit error rate increases, in the prediction the weight of information of the past should decrease.

To investigate this further we now consider another DPCM predictor, which has the following structure:

$$\hat{\tilde{f}}(n) = \sum_{k=1}^{A} \frac{c}{A} \tilde{f}(n-k) \tag{18}$$

Essentially, this is a predictor that averages *A* previous data samples, and weighs them with a factor *c*. Observe that for $c = 1$, the resulting prediction filter will have poles on the unit circle, which will lead to instability of the integral in equation (8). We therefore cannot consider a value of *c* of exactly one but we will use a value close to one. In practical coding systems, however, such instability will not occur and a value of one can be used.

In this example we confine ourselves to the case of coding a first-order autoregressive process with $\rho = 0.92$. In Figure 35 we show the results for $c = 0.98$, for four cases of the bit error rate: 0, 0.001, 0.01 and 0.1, the lowest bit error rate is shown using the 'x' symbol. Clearly, the optimal value of *A* depends on the bit error rate. This is shown in Table 3.

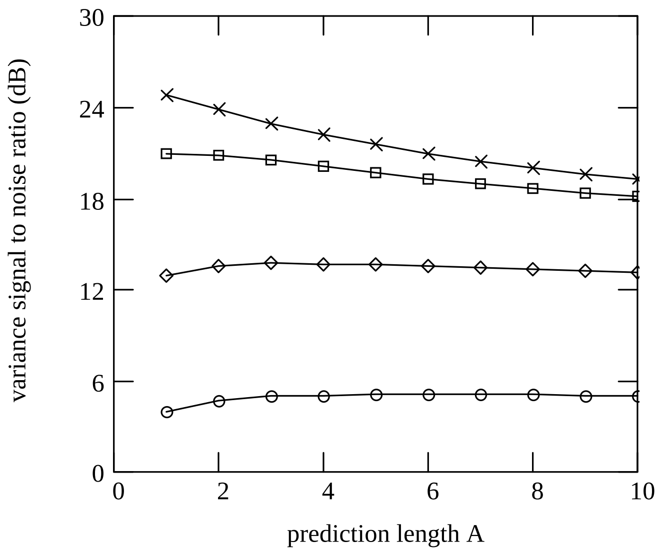


Figure 35 The variance signal-to-noise ratio as a function of the prediction length A, for different bit error rates. The values for the bit error rates are 0, 0.001, 0.01 and 0.1, denoted by the 'x', box, diamond and circle symbol respectively.

Table 3 The optimal value of A as a function of the bit error rate.

bit error rate	optimal value of A
0	1
0.001	1
0.01	3
0.1	6

Having obtained these results, we are encouraged to apply them to the way the prediction, that is the predicted image, is constructed in a hybrid codec like H.263. In such a codec the prediction is normally constructed by subtraction of the previous frame from the current frame. We can now try to construct the prediction by taking a weighed sum of the previous frame and the frames before and subtracting this from the current frame. We could try to optimise the weighing coefficients the same way we did for the DPCM system. However, for video applications this is very unpractical since one can hardly collect the required statistical information, that is the correlation function for each of the pixel positions, that is needed to optimize the set of equations in real time. Therefore we limit ourselves in the practical implementation of the H.263 encoder to using prediction coefficients that are all equal in value.

An other effect we observed is that for higher bit error rates the weigh factors should become smaller. For the modification of the H.263 codec we did not incorporate this because we had to confine ourselves to a limited amount of research within the framework of this thesis. However, this is a topic that is interesting for future research.

As we have seen, using a prediction frame composed of several previous frames increases the error resilience of an H.263 based compression algorithm. Unfortunately, this is not enough for most communication situations. Therefore, we developed additional techniques that increase the error resilience for the H.263 system. The complete set of techniques we developed to modify an H.263 based compression algorithm to make it more error resilient will be described in the next sections.

5.3 The Error Resilient Video Compression Technique

This section describes the different parts of the error resilient video compression algorithm. It is a modified version of the H.263 standard. For the implementation we used as a basis the H.263 2.0 version by Telenor [14]. Section 5.3.1 describes the composed prediction frame, 5.3.2 the adaptive request for intracoded macro blocks, how the errors are detected in the frame to be displayed and how the errors are concealed.

5.3.1 The Composed Prediction Frame

The effect of an error in the encoded bit stream can be devastating. One of the reasons for the impact of the errors on the decoded displayed sequence is that a wrong value in the current frame will be used as a reference with respect to which the next frame is decoded, and consequently that frame will be corrupted as well. However, if we use in the decoded frame not just one value, but an average of several previous values as the reference, the one wrong value will have less impact on the value in the next frame. This principle, which has been demonstrated for the DPCM case, is now used in the technique proposed to make H.263 more error resilient.

The frame with respect to which the current frame is encoded is not the previous frame; it is a newly constructed frame composed of a number, A, of previous frames, which are weighed and summed:

(19) $$f_{ref}(n) = \frac{1}{\sum_{k=1}^{A} c_k} \sum_{k=1}^{A} c_k \, f_{recon}(n-k)$$

In (19) $f_{recon}(n)$ stands for the n^{th} reconstructed frame and $f_{ref}(n)$ stands for the composed prediction frame. The difference with (18) is that in (19) the previous values each are weighed individually, while in (18) they are first averaged and then collectively weighed. However, in the remainder of this chapter we use an averaging of the last *A* frames, which means that the weighing factors c_i in (19) are all equal to 1. Alternatively, a recursive version of (19) can be used. The block diagram of the modified H.263 encoder is shown in Figure 36.

The result of using (19) is a decrease in the effect of an error in the reconstructed frame and its propagation. The prediction will be worse, however, because the average of *A* previous frames is usually not as similar to the current frame as the previous frame, especially when there are large motions in the video scene. The worse prediction leads to a lower compression ratio. Therefore, the performance of the system in terms of visual quality and bit rate depends on the choice of *A* as well as c_i. This has been shown for the second-order DPCM system in Section 5.2.

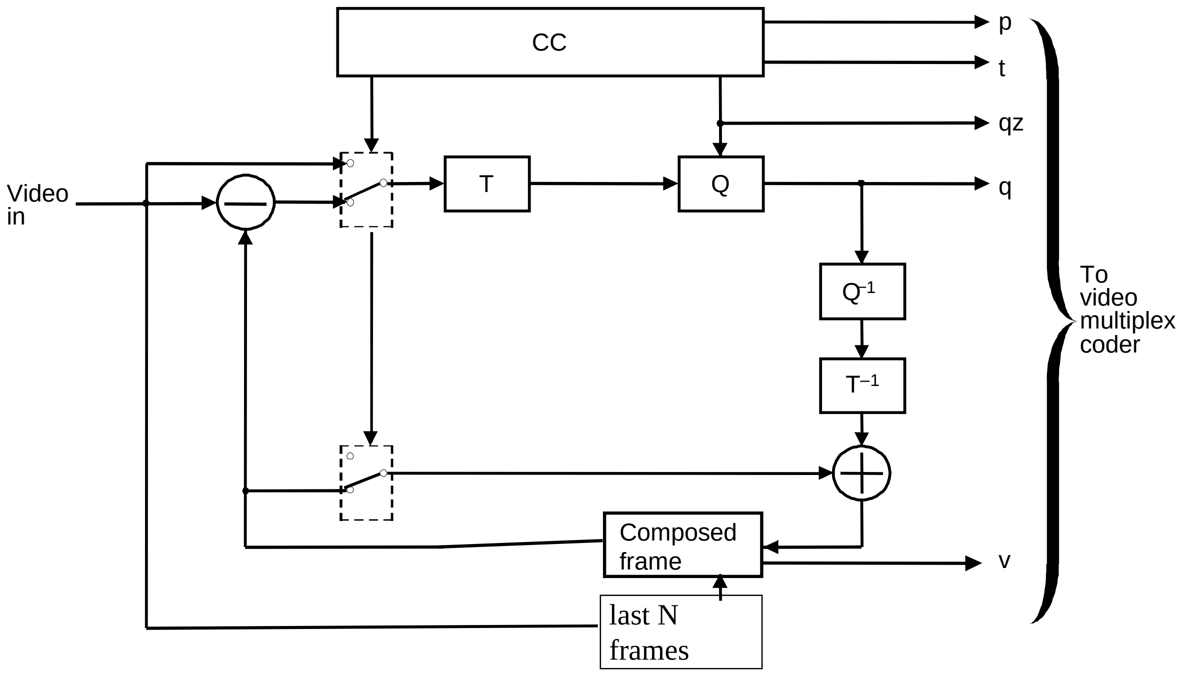


Figure 36 The modified H.263 encoder. The original encoder is shown in Figure 12.

Because in the absence of errors *A* = 1 is most effective, and at high bit error rates the disadvantage of a decrease in compression ratio is outweighed by the advantage of error resilience, *A* should be adjustable. In that case, information is required at the decoder on the composition of the prediction frame. This information has to be sent with each frame. This would require to be sent with each frame, information on how the composed prediction frame for this frame was composed. Although this method is not applied in the technique we describe in this chapter, we have applied it to shape coding. This is described in Chapter 7, where differential coding of the shape of video objects was made more error resilient using this very technique. For each shape, the number of previous shapes contributing to the composed prediction shape is transmitted, as well as the weighing factors.

We now want to address the dependency of the bit rate on the value of *A*. In the case of DPCM we have been able to produce some numerical results after theoretical analysis. However, a hybrid coding system like H.263 is much more complex then the DPCM case, for which we could not find a full analytical solution. Therefore we do not follow a theoretical approach here and have performed experiments.

The difference in bit rate for different values of *A* is shown in Figure 37. We see the bit rate as a function of the frame number of the Hall Monitor test sequence for *A* = 1, which is common intercoding, for *A* = 5, for *A* = 20 and for compression without prediction, which is intracoding. We see the lowest bit rate for intercoding, as expected. On this logarithmic bit rate scale we also see that the increase in bit rate with respect to intercoding for low values of *A* is moderate and still far removed from the bit rate for intracoding.

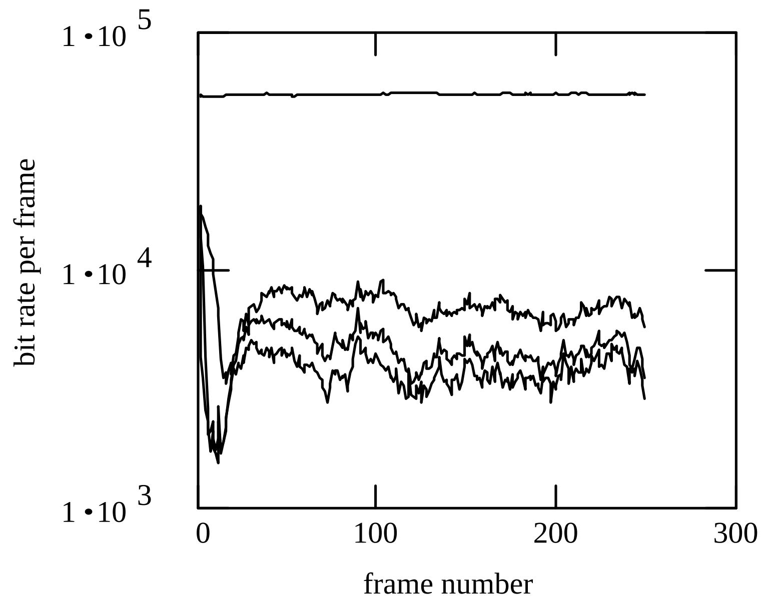


Figure 37 The bit rate for different coding modes. Shown from bottom to top are intercoding, coding with A = 5, coding with A = 20 and intracoding.

5.3.2 Adaptive Intracoding of Macro Blocks

In this section the second part of the error resilience technique is described. It consists of the following three elements. At the decoder, first the errors have to be detected in the frame that is reconstructed and ready to be displayed. Then, at the positions in the frame where these errors have occurred, intracoding of a macro block is requested, which will arrive some frames later, depending on the round-trip delay between encoder and decoder. In the meantime the errors are concealed. These three steps are described in more detail hereafter.

Error Detection

An important and perhaps the most critical part of the technique is the detection of the errors. Some techniques for error detection exist [74][88][94][115], also for hybrid codecs. However, because of our optimisation criterion, we want to detect the effect of an error in a macro block in the reconstructed image, right before display. We use several methods to accomplish this. Each of these methods searches, per macro block and per colour, for one of the following features:

1. Inconsistent H.263 syntax. However, only a few of the possible inconsistencies are checked.
2. Large intensity jumps coinciding with the macro block boundaries. Although such jumps are possible, they are very unlikely in a natural video sequence.
3. The combination of the following, conflicting, two features:
   - The macro block was not coded, because the difference with the previous macro block was almost zero.
   - In the reconstructed frame there is a big difference in average intensity between the macro block of the current and of the previous frame.

If one of these macro block evaluations is true, the macro block is marked corrupted. Especially when 2 and 3 are used, a macro block can be marked corrupted without having an error. However, we do not minimise such “false alarms”, at all costs, as not finding an error when there is one is worse. This is because this might decrease the image quality substantially, while the false alarm will only result in a minor increase in bit rate.

Intra Request

After error detection, a map is available with flags indicating which macro blocks are corrupted. This map consists of 396 bits for an image size of 352 x 288 pixels. This bit rate, which is less than 10 kilobits per second at 30 frames per second, can be reduced substantially using run length or entropy coding. The map is sent to the encoder using the feedback channel. This is shown in the simplified block diagram of the proposed technique in Figure 38. The encoder will now force intracoding of the flagged macro blocks as soon as possible and resets its flag for that macro block. In the case of high bit error rates, the number of forced intracoded macro blocks is limited by the encoder. The forced intracoding of the macro blocks at the signalled position will make sure that the error is fixed. Error propagation is dealt with indirectly by the decoder, because any error in the frame to be displayed is detected, regardless of its cause. This kind of error detection can therefore also lead to requests for intracoding of macro blocks which are not justified by the presence of a real error. Such an unjustified request for an intracoded macro block will only increase the bit rate a little, while an undetected error can reduce the image quality substantially.

In general, because of the intracoding of macro blocks, the bit rate will go up. In a fixed rate situation the image quality, that is the peak signal-to-noise ratio, will go down. This is because in that case there are fewer bits available to be allocated to the rest of the macro blocks. In the presence of errors, however, information is lost and restoration requires additional information, which in turn requires bits. It is therefore inevitable that the bit rate goes up or the quality goes down.

Error Concealment

Because of the round-trip delay, the corrupted macro block is not fixed immediately by an intracoded macro block. Therefore, error concealment has to be applied during the time between the corruption and the arrival of the intracoded macro block. The error concealment technique we use is replacement of the current corrupted macro block with the macro block of the previous frame, which we assume to be uncorrupted. Information about the motion vectors could also be used to aid the error concealment, but this is not implemented.

In our integrated error resilience technique, some features of such an error concealment technique become important:

- If an error in the bit stream forces the decoder to resynchronise at the next resynchronisation word, all macro blocks in between the macro block with the error and the next synchronisation word are requested to be intracoded.
- When the corruption of a macro block is not detected and therefore not concealed, using the composed prediction frame will maintain the image quality to a certain extent.
- The concealment of the error will also avoid multiple error detections at the position of the error for the duration of the delay.

A simplified block scheme of the complete system is shown in Figure 38. Above the horizontal dashed line the original coding system with encoder, transmission channel and decoder is depicted. Below this line we see the new elements. The encoder stores the last *A* frames and calculates the composed reference frame. It receives the map of corrupted macro blocks via the feedback channel. This map is produced in the decoder using the error detection. The error detection information is also used for error concealment. The decoder also stores the last *A* frames and calculates the composed reference frame.

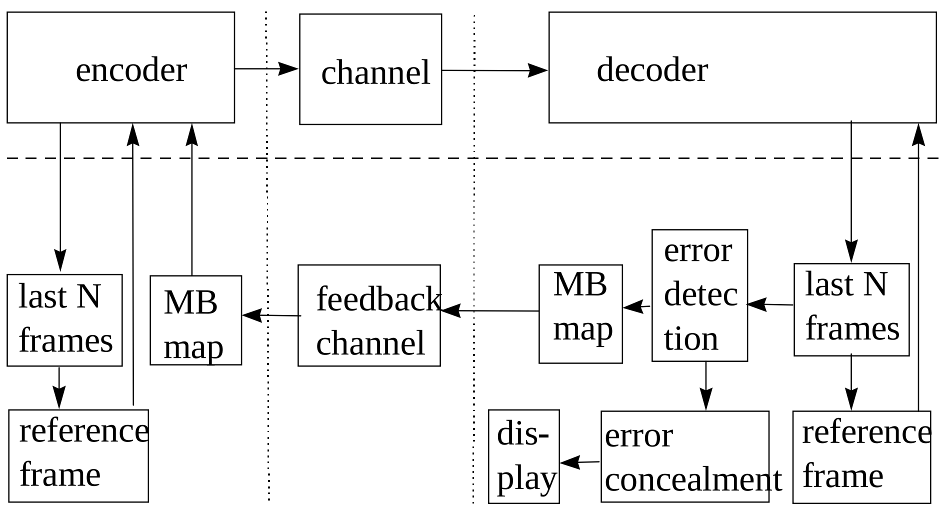


Figure 38. A simplified block scheme of the complete system.

5.4 Experiments

In this section we describe the set-up and results of the experiments that were carried out with the error resilient video compression algorithm. We also compare some of these results with MPEG-4 results. The algorithm is a modified version of H.263 and the implementation is based on the Telenor H.263 2.0 codec [14].

5.4.1 Experimental Set-up

We used the Hall Monitor, Foreman, Miss America and Car Phone test sequences in CIF format, which is 288 x 352 pixels and 4:2:0 chrominance subsampling at about 30 frames per second, and in quarter CIF format. We used the Telenor H.263 2.0 codec as a reference. In the reference algorithm we inserted a synchronisation word after every row of macro blocks. This improved the error resilience of the reference codec substantially. Since this is part of the H.263 standard, we think it should be part of the reference codec.

The error pattern we used to generate transmission errors was random. We did not impose errors on the feedback channel because it can easily be protected and errors in the macro block map have no great effect. For more comparable results we did not allow errors in the first two frames. The runs using a fixed bit rate were obtained using the rate control of the Telenor codec. We evaluated the results using peak signal-to-noise ratio, bit rate and subjective criteria. The peak signal-to-noise ratio *PSNR* is defined as

(20)
$$PSNR = -10 \cdot {}^{10}\log\left(\frac{\left(v_{orig} - v_{recon}\right)^2}{v_{\max}^2}\right)$$

where v_{orig} is the value in the original sequence, v_{recon} is the value after encoding, transmission including possible errors and decoding, and v_{max} is the maximum value that v can have. The units used for the *PSNR* are decibels (dB).

5.4.2 Results

The Hall Monitor sequence

We measured the peak signal-to-noise ratio of the Hall Monitor sequence in CIF format at different bit error rates. We used an average fixed bit rate of 250 kilobits per second, which is about 8300 bits per frame. We used also different values for *A*. The delay was fixed to 15 frames and the motion vector search range to 8 pixels.

Shown in Figure 39 are the peak signal-to-noise ratio results. Shown are the original sequence, that is the sequence coded with the reference codec, without errors, shown as the solid line, the original with errors, shown as + symbols, and the proposed error resilient technique with errors, shown as circular symbols. The delay is 15 frames, A = 5, the motion vector search range 8 pixels, the bit error rate 10^{-4} and the bit rate fixed at 250 kilobits per second. The average peak signal-to-noise ratio obtained by the reference codec without errors is 38.3 dB, due to quantisation noise and coding artefacts. Introducing errors brings this down to 21 dB after only a few seconds. The error resilient technique results in 34.8 dB on the average, with some fluctuations, which is an increase of 13.8 dB. An equally important result is that it produces a stable image quality, even though the average peak signal-to-noise ratio is less than the original.

A bit error rate of 10^{-3} gives results in the order of 17 dB for the standard codec and 31.5 dB for the error resilient technique. At a bit error rate of 10^{-2} the error resilient technique becomes unstable: the number of errors is higher than the number of intracoded macro blocks that reach the decoder without any error. A higher *A* could not prevent this.

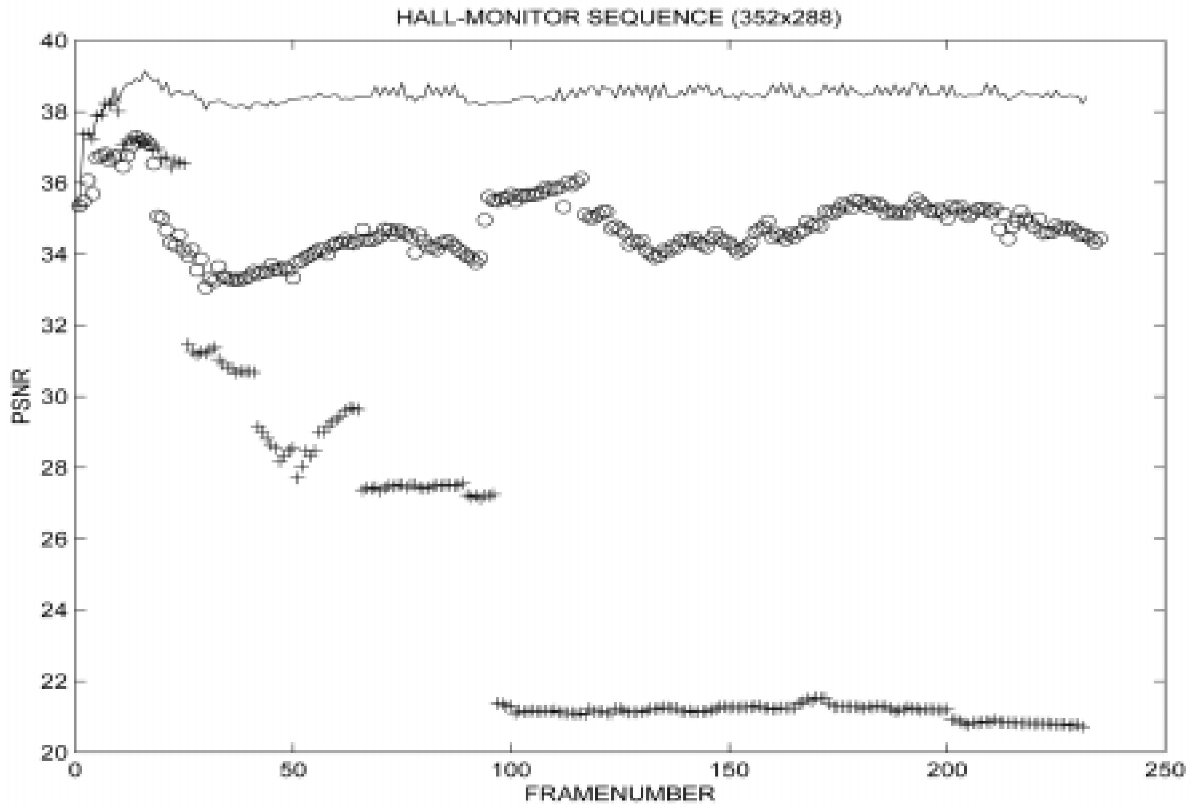


Figure 39. The peak signal-to-noise ratio of the Hall Monitor sequence. Original (solid line), with errors (+) and using the error resilient technique (o). The bit error rate was 10^{-4}.

The Car Phone and Foreman sequences

The Car Phone and Foreman sequences, which contain more motion, were run, now in quarter CIF image format. In Figure 40 is shown the Car Phone sequence at a bit error rate of 10^{-4}. Here we used a value for *A* of 3, a motion vector search range of 15, a delay of 3 frames and a fixed bit rate of 150 kilobits per second. Shown are the original sequence coded with the reference codec without errors, shown as the solid line, the original with errors, shown as + symbols, and the proposed error resilient technique with errors, shown as circular symbols. Using a quarter CIF image format instead of CIF results in a smaller amount of errors per frame, therefore overall the peak signal-to-noise ratios are higher. We also found with these sequences that when there is a lot of motion, like after frame 170 in Car Phone, the error resilient technique performs slightly less. This is due to a worse prediction because of the averaging and also due to less appropriate concealment because of the large motion. In this part of the sequence, the original codec recovers some, due to the many intracoded macro blocks that are generated when there is large motion and no proper motion vectors can be found.

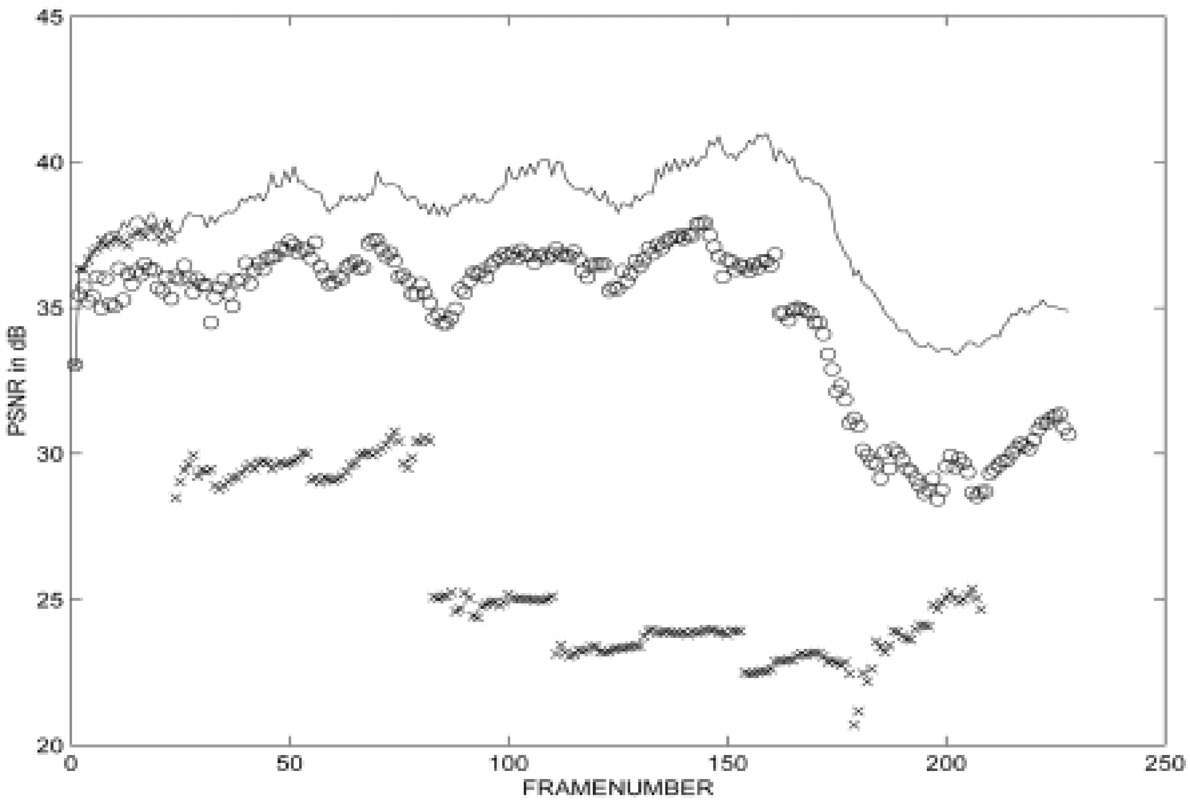


Figure 40. The peak signal-to-noise ratio of the Car Phone sequence. Original (solid line), with errors (x) and using the error resilient technique (o).

In the following paragraphs the Hall Monitor test sequence is used again in the experiments.

The number of requested macro blocks at different bit error rates

Figure 41 shows the number of requested intracoded macro blocks for different values of the bit error rate. Shown are the results for bit error rate zero as a solid line, for 10^{-4} using circular symbols, for 10^{-3} using x symbols and for 10^{-2} using points. The values for *A* were 1, 5, 11 and 15 respectively. The number of requested intracoded macro blocks increased with increasing bit error rate, as is to be expected.

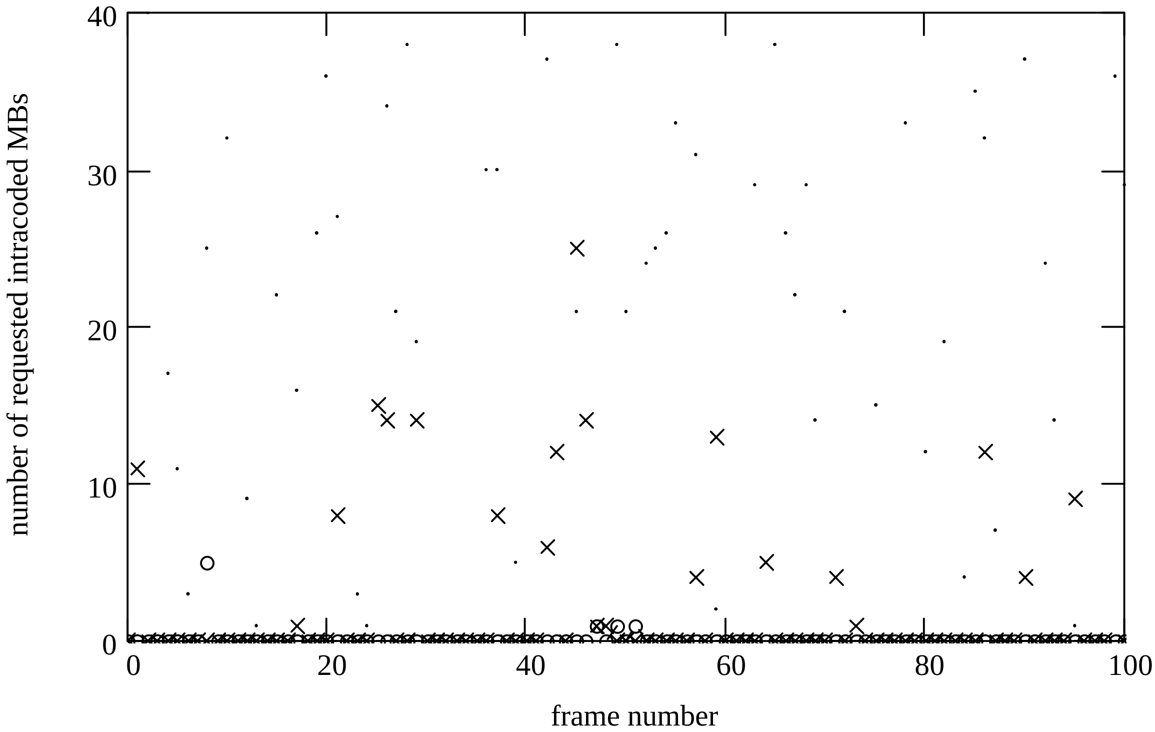


Figure 41 The number of requested intracoded macro blocks as a function of the frame number for different bit error rates. The used values for the bit error rate were 0, 10^{-4}, 10^{-3} and 10^{-2}, using a solid line and o, x symbols and points respectively.

The number of errors at different bit error rates

We also looked at the number of errors at different bit error rates. The results are shown in Figure 42. Shown are the results for bit error zero as a solid line, for 10^{-4} using circular symbols, for 10^{-3} using x symbols and for 10^{-2} using points. The used values for *A* were 1, 5, 11 and 15 respectively.

From this we can estimate how many macro blocks we expect to be requested to be intracoded. For instance, using a CIF sequence at 250 kilobits per second and a bit error rate of 10^{-4} we expect $8300 \cdot 10^{-4} = 0.8$ corrupted macro blocks per frame and 0.8 errors per frame. In the experiments we find an average of 6 macro blocks per frame, which is more than 7 times what we expected, and an average of 0.8 errors per frame, which is as expected.

Although the experimental value can be lower than the calculated value when multiple errors hit one macro block, this value can also be much higher, as any macro block that is corrupted by propagating errors will also be requested. Also the found number of requested intracoded macro blocks can be larger than expected due to the 'false alarms', which might be reduced by a more efficient error detection algorithm.

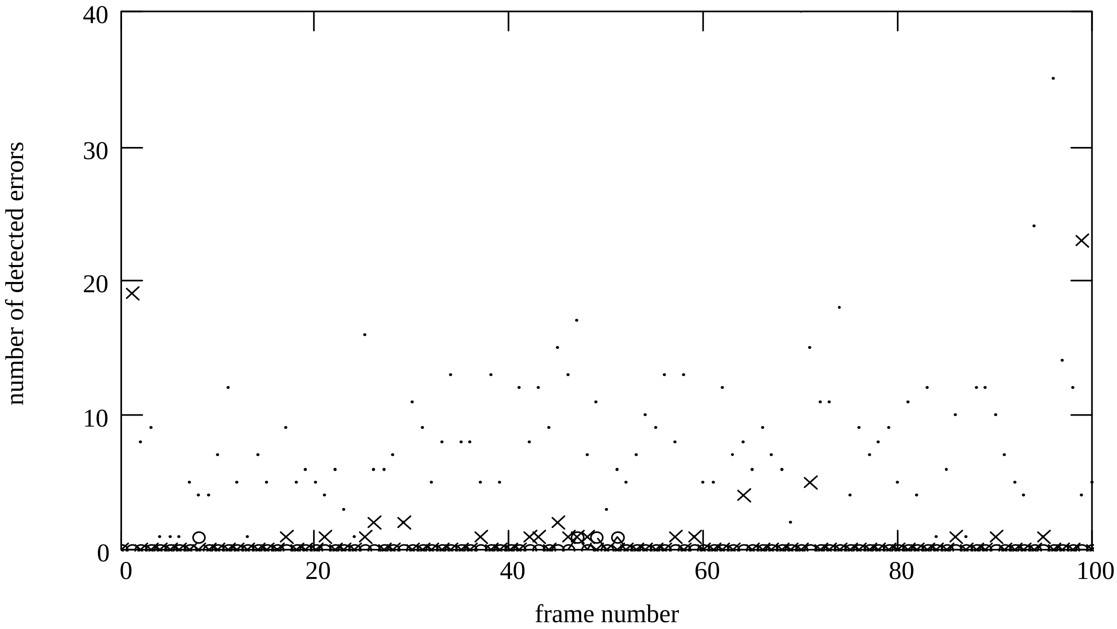


Figure 42 The number of detected errors as a function of the frame number for different bit error rates. The used values for the bit error rate were 0, 10^{-4}, 10^{-3} and 10^{-2}, using a solid line and o, x symbols and points respectively.

Variable bit rate

Up to now, we used a fixed bit rate. Now we consider results obtained using variable bit rate. First we look at the bit rate for different values of *A*. We can use the results shown in 5.3.1 about the increase in bit rate for different coding modes as a starting point. We now evaluate also values for *A* of 3 and 10 and we compare these with the bit rate for intercoding, that is, *A* = 1. Figure 43 shows the bit rate as a function of the frame number, at a bit error rate of 0, for different values of *A*. The result for *A* = 1 is shown as a solid line, for *A* = 3 using + symbols and *A* = 10 using circular symbols. We can see now in more detail that for almost every frame there is an increase in bit rate and that the actual increase differs per frame.

We also obtained results that show the bit rate as a function of the bit error rate. Figure 44 shows the bit rate for different bit error rates as a function of the frame number. Shown are the results for bit error zero as a solid line, for 10^{-4} using circular symbols, for 10^{-3} using x symbols and for 10^{-2} using points. The used values for *A* were 1, 5, 11 and 15 respectively. In these results the increase in the number of requested intracoded macro blocks is decreased by the use of a higher value for A at higher bit error rates. This is so because using more frames to compose the prediction frame decreases the effect of an error in the image and thereby the necessity to request an intracoded macro block decreases as well. On the other hand, increasing A will also cause an overall increase in bit rate as the prediction is worse, as is already shown.

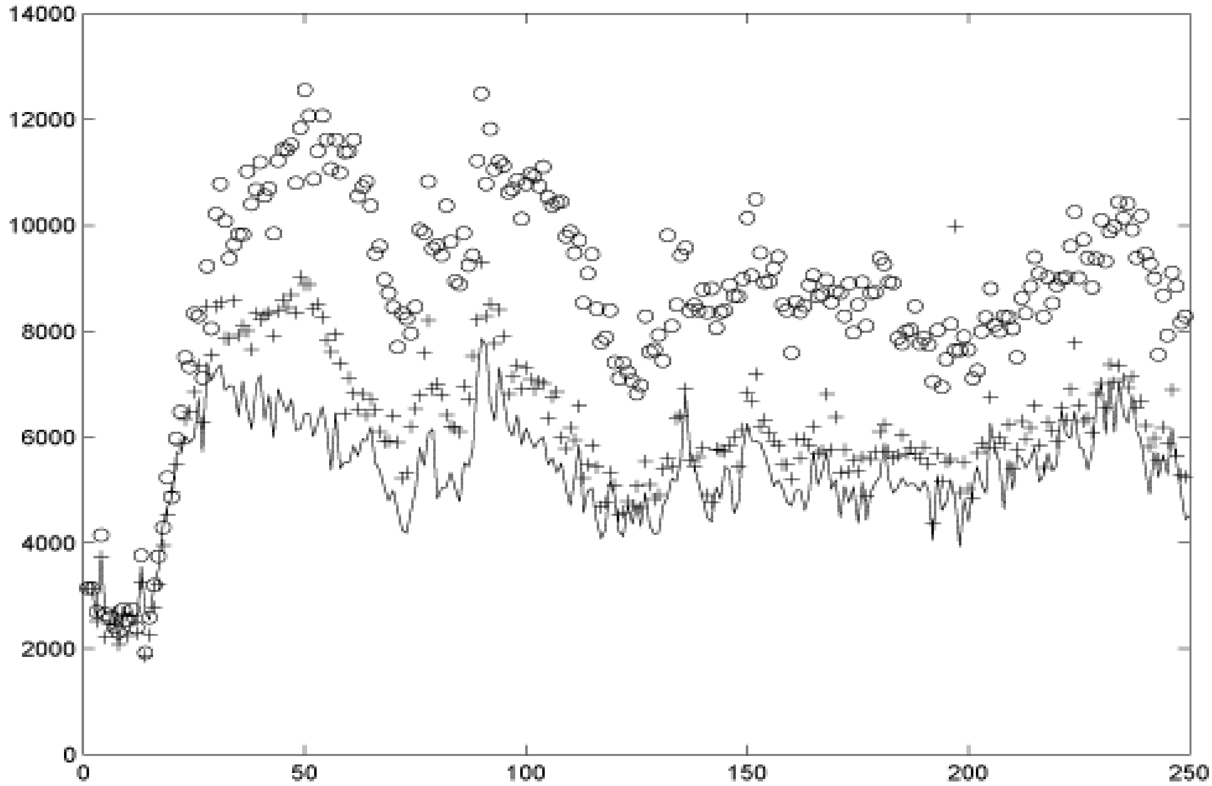


Figure 43. The bit rate for different values of A: 1 (solid line), 3 (+) and 10 (o).

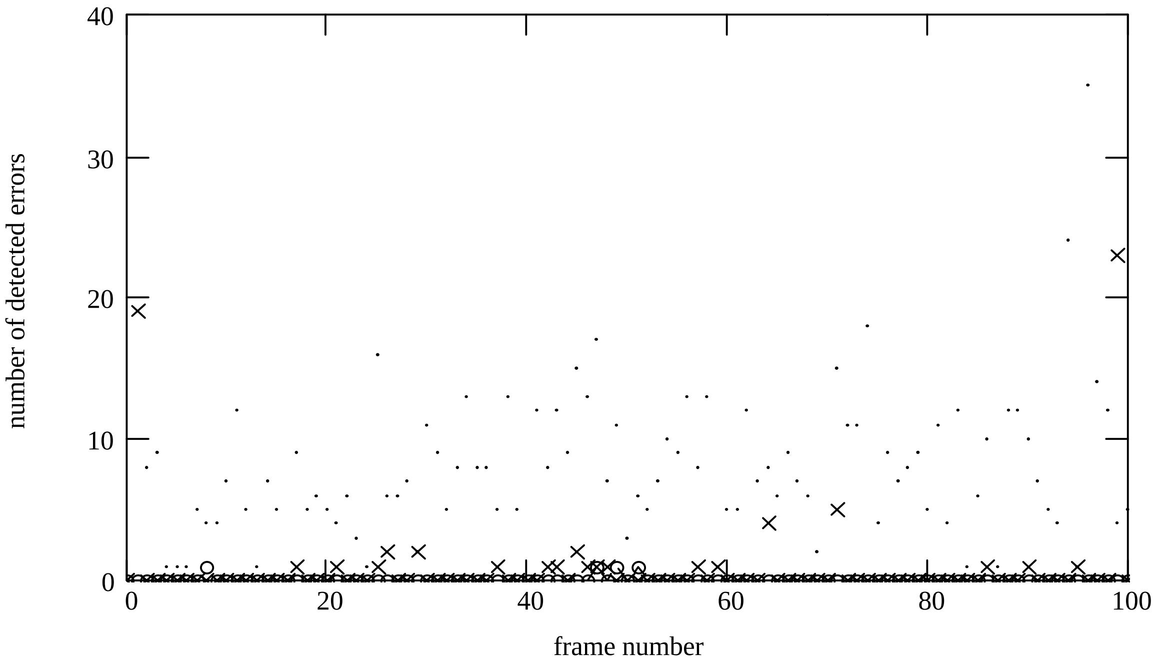


Figure 44 The bit rate for different bit error rates as a function of the frame number. Shown are the results for bit error zero as a solid line, for 10^{-4} using circular symbols, for 10^{-3} using x symbols and for 10^{-2} using points. The used values for A *were 1, 5, 11 and 15 respectively.*

The value of *A* as a function of the bit error rate

The parameter *A* is the number of previous frames used to compose the prediction frame used in the video compression algorithm. A high value for *A* will decrease the effect of the errors by a kind of smoothing, but it will also increase the bit rate, because the prediction is worse, especially when there is motion in the scene. This means that the optimal *A* in terms of both peak signal-to-noise ratio and bit rate varies with the bit error rate. When there are no errors, the absence of an increase in bit rate is most important and *A* should therefore be 1, the lowest value and corresponding to intercoding, while for large bit error rates the reduction of the impact of errors outweighs the disadvantage of the additional bit rate and *A* should be relatively high.

A thorough experimental evaluation would give a graph that we could use to determine *A* for every bit error rate. However, the optimisation criterion is not clear, since it is not obvious how the criteria of image quality and bit rate should be combined into one optimisation parameter, which would determine the optimal value for *A* at a certain bit error rate. *A* also influences other parameters, for instance the bit rate definitely increases when the frame rate decreases and *A* is high. Therefore the effect of *A* on these other parameters should also be incorporated in the determination of the optimal value for *A* as a function of the bit error rate.

This means that we cannot carry out an exhaustive experimental evaluation. However, from the experimental results described so far we already know some values and their effects on the performance of the error resilient compression algorithm. For instance, for the Hall Monitor, we varied *A* from 3 to 15 at bit error rates up to 10^{-3}. This showed that, at these bit error rates, a substantial increase in quality compared to *A* = 1, which is standard intercoding, can be obtained for *A* = 5, while a higher value does not increase the quality very much.

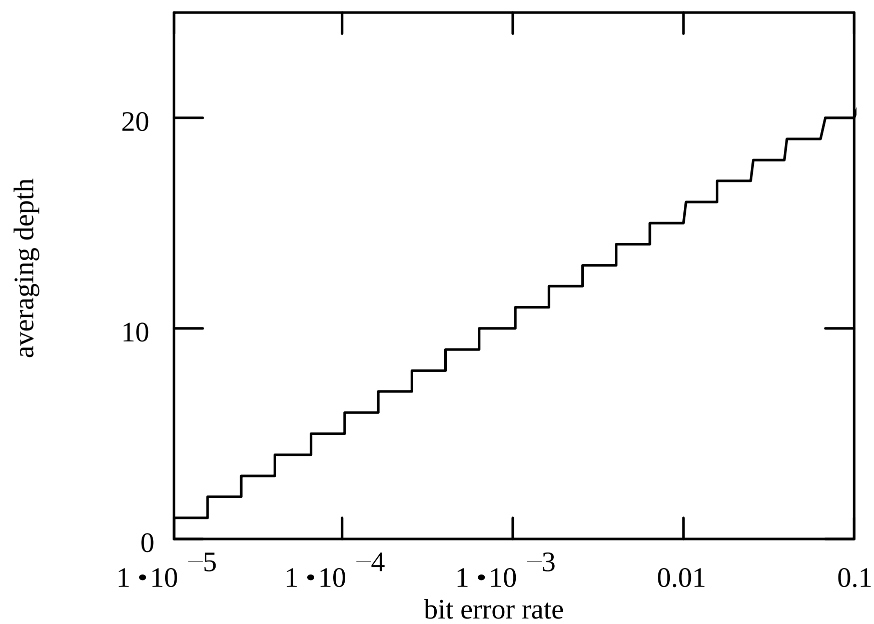


Figure 45 The value of the parameter A as a function of the bit error rate.

From these and other experimental values we pose the following relation, equation (21), between the bit error rate *ber* and *A*;

(21) $$A = \left\lfloor \left(5 \cdot {}^{10}\log(ber)\right) + 26 \right\rfloor$$

which is also depicted in Figure 45.

Visual evaluation

Since we have to evaluate the impact of errors on video, a subjective evaluation is important. We did not use a panel to judge the improvements in image quality because the visual effects are very obvious. Here we are dealing with improvements in the order of 10 dB and not 0.1 dB, as in cases where the improvement in compression efficiency is judged and not the effect of transmission errors.

Using the reference codec, the visual quality becomes unacceptable at a bit error rate of 10^{-5} after only a few seconds. At higher bit error rates, the visual quality degrades even faster.

Using the error resilient technique, results were invisible down to acceptable up to a bit error rate of 10^{-4}. An example of the difference in visual effects is shown in Figure 46. At a bit error rate of 10^{-3}, the image is definitely damaged, but still usable. At a bit error rate of 10^{-2} and higher, the image contains hardly any recognisable features.

The minor overall decrease in peak signal-to-noise ratio in the results of the proposed technique is for a large part extra quantisation noise due to the reduced number of bits available for the non-intracoded macro blocks. Although this difference is definitely noticeable, it is more acceptable than a similar decrease in peak signal-to-noise ratio in the results of the original codec, which is caused by some local misplaced or unrealistic macro blocks with abnormal colour and texture.

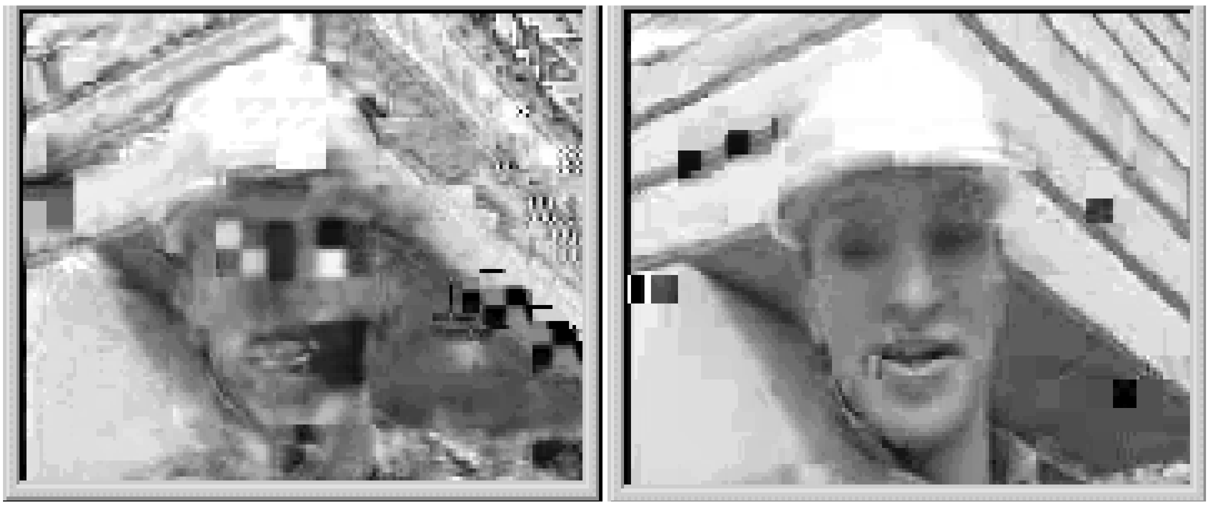


Figure 46. Examples of the Foreman sequence without and with additional error resilience, shown on the left and right respectively.

5.4.3 Comparison with MPEG-4

In [41], the error resilience and concealment performance for MPEG-4 frame based coding was addressed. We compare their results with our own results. We first describe how the MPEG-4 results were obtained.

Results have been obtained for different situations at a fixed overall bit rate. Unfortunately, the variance in bit rate was not shown in [41]. It is also unclear at what rate the first, intracoded, frame was coded and whether this rate is included in the bit rate limit that was used to obtain the results presented in [41]. The YUV peak signal-to-noise ratio is used for evaluation.

Within MPEG three kinds of residual error condition classes have been defined for the purpose of studying error resilience performance in the video layer [13]. On top of the video layer a multiplex layer is assumed. The three types of error condition classes are:

1. Uniformly distributed errors. This case corresponds to a wireless system where extensive interleaving is performed by the network and the errors are simply passed on to the video layer by the multiplex layer.
2. Burst errors. This case models the errors caused by a fading channel, where the multiplex layer is able to correctly decode the headers of the packets of the multiplex layer. The errors in the data itself are passed on to the video layer.
3. Packet loss errors. This case is representative for the situation where the multiplex layer is unable to decode a packet header and therefore all the data in the packet is lost. This case can also be representative of packet losses caused by time-outs that might occur in an internet-like application.

For each of these error classes one condition was chosen and evaluated; for type 1 this was a bit error rate of 10^{-3}. The MPEG-4 combined mode with data partitioning was used, which means that for each frame the motion and texture data is multiplexed at the macro block level and that the motion information and texture information are separated to increase the error resilience.

Starting from this situation, the performances when additionally backward decoding and when error concealment was used, were addressed separately.

We now compare our results with the MPEG-4 results. We do this only for the results for type 1 errors because in our current implementation results from runs using the other conditions would not be comparable. The test sequence was Foreman, at 10 frames per second, quarter CIF format. For the MPEG-4 runs, an average bit rate of 48 kilobits per second was used, which is about 500 bits per frame, with a total of 50 frames per run and a synchronisation word every 736 bits. The results of 10 runs were averaged.

This means we have in the MPEG-4 situation about 4915 bits per frame and 99 macro blocks per frame. This leads to about 50 bits per macro block and therefore to a synchronisation word about every 15 blocks, which is for every 1.36 groups of macro blocks. In our set-up we use one synchronisation word every 1 group of blocks.

At this bit rate and bit error rate we have 49 errors per second on average, so about 5 per frame, with many in the first frame because this frame is large. However, the errors do not show until after frame 10, therefore we did not allow errors in the first frame in our own runs.

We see in the MPEG-4 results in Figure 47 the same features both for application of the reversible variable length code words option and for application of the error concealment options. The sequence starts with 33 dB and after the first error, the peak signal-to-noise ratio drops about 14 dB in about 15 frames to 19 dB and does not recover the original image quality.

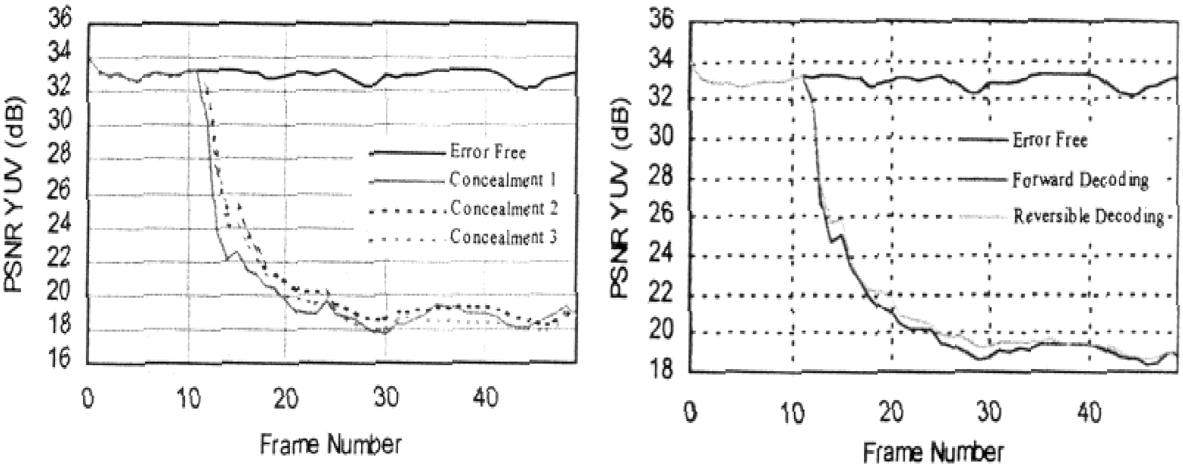


Figure 47 The MPEG-4 results as they were obtained in [41], for different concealment methods, shown on the left, and for reversible decoding, shown on the right.

We made similar runs, obtaining a bit rate of 5400 bits per frame and an error free peak signal-to-noise ratio of about 30 dB. This means that the version of H.263 we used is less efficient than the MPEG-4 frame video codec. We did not use the adaptive intracoding of macro blocks because in the MPEG-4 runs no back channel was available either. So what we compare here is the efficiency of the set of MPEG-4 options as we described above with the efficiency of our techniques of using a composed prediction frame and error concealment only.

We now look at the results we obtained for a bit error rate of 0 and 10^{-3}, shown in Figure 48. Furthermore, we made two runs with different values for the number *A* of used previous frames to compose the reference frame: 11 and 3. For *A* equal to 11 we see a drop of the peak signal-to-noise ratio like in the MPEG-4 results. The drop is faster but less deep, about 10 dB instead of 14 dB in less than 10 frames. So in peak signal-to-noise ratio our technique is superior to MPEG-4. The bit rate does increase, however, as opposed to MPEG-4, with a factor of more than 2 to about 13000 bits per frame. For *A* equal to 3 we see something similar, but the increase in bit rate is less, about 8800 bits per frame, while the peak signal-to-noise ratio is higher in the beginning but lower at the end.

This rough comparison between the MPEG-4 techniques and our technique where only error concealment and the composed prediction frame are used, shows that our technique is superior in peak signal-to-noise ratio at the cost of a higher bit rate. As our techniques to obtain the error resilience and the one used in MPEG-4 are very different, it is probably easy to combine these, which would result in a technique that is much more error resilient than each of the ones of which it consists. In other words, the error resilience of MPEG-4 can be improved by our technique.

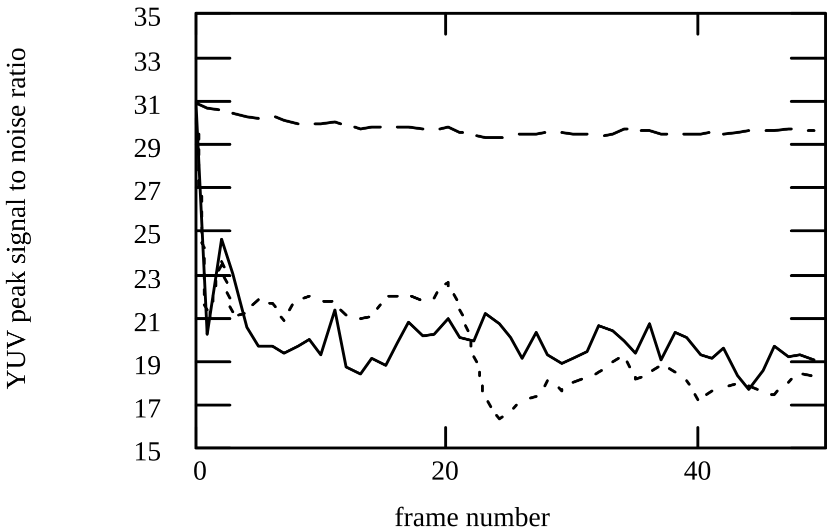


Figure 48 The peak signal-to-noise ratio as a function of the frame number for the Foreman sequence using parameters similar to the MPEG-4 performance evaluation. The bit error rate is 0, which is shown by the top curve, and 10^{-3}. The frame rate is 10 frames per second and the number of previous frames used for the composed prediction frame is 11, which is shown as a solid line and 3, shown as a dashed line.

5.5 Conclusion

In this chapter we presented a new error resilient compression technique based on the H.263 standard. We first showed the importance of taking, in the case of transmission errors, more than one previous value to compose the prediction using a simple DPCM system. This we then applied to the H.263. Furthermore, we introduced error detection, the request by the decoder of corrupted macro blocks to be intra coded in some future frame, and error concealment. We found that error detection is difficult but important, and we consider it a subject for future research.

The experimental results show that this error resilient compression technique performs well in maintaining the image quality while not increasing the bit rate substantially. The limit for this performance lies at a bit error rate of about 10^{-3}.

We also compared application of part of our technique with application of most of the MPEG-4 error resilience options. This turned out in favour of our technique, although the image quality also in our case was not high. This was mostly due to the high bit error rate and not using all of our error resilience techniques for better comparison.

In our technique we use a modified compression algorithm to handle the effect of the transmission errors. However, as we have seen in Chapter 4, we can also handle these errors in the link and network protocol. This is done in the research on the link and network protocol in the Mobile Multi-media Communication project. Both techniques can be combined into joint error resilient source channel coding. However, both techniques have a certain performance in error resilience, that is image quality, and in bit rate. The optimisation, that is which of these techniques performs best and should be used in which situation, is addressed in the next chapter.

.

Chapter Six

6 Joint Error Resilient Source Channel Coding

In this chapter we address the complex and partially still unsolved problem of optimising a combined error resilient source channel coding system in terms of image quality and bit rate. We have chosen to do this through an experimental approach.

After the introduction in Section 6.1 we first state the problem in Sections 6.2 and 6.3. Then we describe the framework, the set-up of the link protocol that we used and the video compression algorithm parameters. We do this in Section 6.4. The distribution of the bits over different error resilience techniques is dealt with in Section 6.5, which also describes the experimental determination of the optimal bit distribution. Then the important issue of errors caused by packet loss is dealt with in Section 6.6. Aspects of generalisation of the results are discussed in Section 6.7. We conclude this chapter with Section 6.8.

6.1 Introduction

In the previous chapter we presented a solution to the problems caused by errors in the stream of compressed video data. The errors occur mainly in the wireless part of the link between the encoder and the decoder. This suggests another solution: making sure there are no errors in this part of the link. This can be done by designing an error resilient link protocol, as we have seen in Section 3.4. Different disciplines investigate these two kinds of solutions, and therefore approach the subject differently as well. Even though we have the source channel separation theorem [119][142], it cannot easily be applied to our situation because of the complex system of the compression algorithm, link protocol, error resilience techniques and the influence of errors. Therefore, and since both error resilient source and channel coding have the same goal and work on the same problem, a joint solution can and should be considered. A first step is taken in the Mobile Multi-media Communication project, where we try to find a trade-off between both error resilience techniques developed in the project.

We now first state the problem. Then we describe the framework, the set-up of the link protocol that we used and the video compression algorithm parameters. The determination of some specific parameters necessary for the experiments is also discussed. After that the distribution of the available bits over the error resilience techniques is addressed. Then the errors caused by packet loss are described and experimentally addressed and finally aspects of generalisation of the results are discussed.

6.2 Problem Statement

The errors that will occur during the wireless transmission due to channel noise and network congestion can be anticipated and handled in two places in the communication link, namely in the network protocol and in the video compression algorithm. The problem is to establish criteria that determine the trade-off between error resilience due to protocol actions and due to advanced compression and decompression techniques. In both cases the error resilience comes at the price of data overhead. The question is which combination yields the smallest overhead for the highest error resilience and image quality. This depends on many parameters concerning the protocol and compression algorithm. Since many of the parameters are explicitly or implicitly dependent upon the characteristics of the actual communication situation, the optimal parameter settings will also be different for each individual communication situation.

To gain insight in the mutual parameter influences and the trade-off between various parameters, two approaches are known: the analytical approach and the experimental approach. For the first, one must simplify the problem greatly by choosing a theoretical situation or a very specific application. This leads to general results but application of these to a real situation is difficult. The second approach does produce realistic results but they are hard to generalise.

Since both an error resilient link protocol and an error resilient video compression algorithm are available in the project, we have chosen to carry out an evaluation using an experimental set-up. However, also in the experimental set-up, the results depend on many parameters, most of which are not independent. Furthermore, for each set of parameters, at least one test sequence has to be compressed, passed through the transmission simulation and decompressed, for which a high computational effort is required. Additionally, many data points, that is runs, are necessary to gather sufficient statistics to produce meaningful results. This problem of high dimensionality combined with computational intensity is the reason why a full search of the parameter space cannot be carried out. We deal with this problem in the next section, Section 6.3.

6.3 Bit Distribution

As we have stated, our main problem at this point is the multi-dimensionality combined with sparse data due to computational intensity. A solution to this problem is intelligent reduction of the number of parameters, and for this we return to our problem definition. What we actually want to know is on which error resilience technique we should spend our bits in order to get the best image quality at the lowest bit rate. Starting with a sequence that is compressed obtaining a basic image quality in the absence of errors, we can spend our bits on one or more of the following:

- An increased image quality, established by using a lower value for the quantisation parameter *Q*. This parameter determines the coarseness of the quantisation and thereby also the initial bit rate and peak signal-to-noise ratio.
- An error resilient link, established by retransmission of erroneous packets and the use of forward error correction, as described in Section 3.4.
- Error resilient video compression, which uses requesting intracoded macro blocks, a composed prediction frame, error detection and error concealment, as described in Section 5.3.

We can spend our bits on either one of these three methods for increasing the image quality in the presence of transmission errors, or on a combination of them. Although there are many parameters involved we can choose to fix these parameters to a reasonable value. If we now spend all our bits on one of the methods or on a combination of two or three of them, this leaves us with eight possible methods of spending our bits. The influence of the fixed parameters on the results can then be addressed afterwards by making the approximation that those parameters are independent. Following this approach, the determination of the influence of a parameter on the already obtained results is easier because it requires less computational effort as we only have to vary this parameter. The fixed values of the parameters have to be chosen well, and the approximation of independence must be probable. This is discussed in Section 6.7 on generalisation of the results.

How effective each one of the bit spending methods is in yielding a high image quality will depend on the method and on the bit error rate. The chosen method and the bit error rate will also have an effect on the bit rate. The peak signal-to-noise ratio of the image, which we use as a measure for the quality of the image, and the bit rate are therefore the two output parameters for our experiments. They both depend on the bit error rate and the method used.

We could have chosen to fix the bit rate using the rate control of the Telenor H.263 codec, thereby reducing the parameter space. However, even though this rate control produces a predetermined rate at the end of the sequence, the rate still varies much between frames. In turn this variation causes packet loss every time the bit rate of a frame peaks. Packet loss errors are severe and make determining of the influence of single-bit errors very difficult. Of course the effect of packet loss and burst errors is a very important issue, especially for real applications, and has to be explored, but this is done separately in Section 6.6. An example of the variation of the bit rate, without rate control, is shown in Figure 49. This variation is on the average sufficiently constant for our purposes.

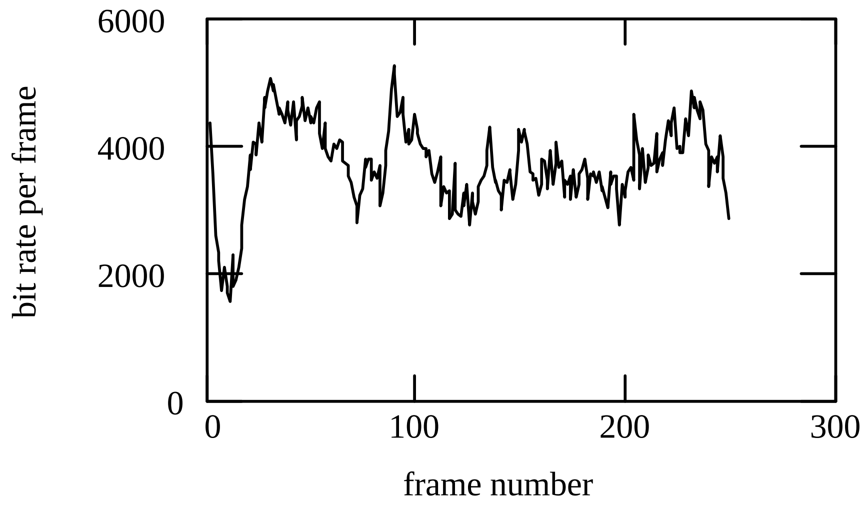


Figure 49 An example of the variation of the bit rate without rate control.

So this leaves us with a five-dimensional parameter space: the application of the three error resilience techniques, the bit rate and the peak signal-to-noise ratio. There are other parameters of the link protocol and compression algorithm, which still influence the performance of the system. However, the relative trade-off between channel and source coding can, at least initially, be addressed using the five-parameter set-up.

6.4 Experimental Set-up

First the specific fixed parameters that are used in our experiments are described and their values given. This is done for the general parameters, for the parameters for the link protocol and for the parameters for the compression algorithm. Finally special attention is given to the determination of the values of the quantisation parameter Q we need in our experiments.

6.4.1 General Parameters

We sum up the general parameters for the basic set-up.

- There is a basic amount of bits to make a rough image. This requires a coarse quantisation with a high value of the quantisation parameter Q. The additional image quality is obtained by using a low Q. The determination of the Qs is addressed in Section 6.4.4. Bits can also be spent on making the link or the compression algorithm more error resilient.

- The two main parameters that are varied are the bit error rate and which method we use for increasing the image quality.
- We look first at only single-bit errors. In a separate Section, 6.6, we look at errors caused by packet loss. Therefore, at first, several parameters are set to avoid burst errors and packet loss.

We use the following specific parameters for the experiments:

- The used test sequence is the Hall Monitor, an image of which is shown in Figure 50.
- The image format is common intermediate format: 288 x 352 pixels, 4:2:0 chrominance subsampling.
- The frame rate is 30 frames per second.
- The number of frames is 250.
- The round-trip delay that is used in the compression algorithm is 3 frames.
- The bit error rate for the first frame is 0 to make the results more comparable.
- The bit errors for the link protocol simulation are generated using an additive white Gaussion noise distribution. For the experiment runs where the link simulator is not used, a random bit error generator was used. This produces results very similar to those produced by means of an additive white Gaussian noise distribution, as no error correcting codes are used then.

Figure 50 An image from the Hall Monitor sequence.

6.4.2 Link Protocol Parameters

The used error resilient link protocol is the one developed in the Mobile Multi-media project, which was outlined in Section 3.4. We here describe some features of this protocol and the parameter settings that are used in our experiments.

- The first frame is set to have a bit rate which is tenfold the average bit rate. This is because the first frame has to be fully intracoded, which requires a high bandwidth, but only for this frame. It is assumed that the link is able to handle this peak in the bandwidth.
- The maximum link delay is set to 80 milliseconds to avoid packet loss. If this value were too small, some of the packets would never be sent because they would be considered overdue. This is especially the case when the bit error rate is high, because retransmission of erroneous packets has priority in this version of the protocol simulator. Note that this delay parameter is different from the delay parameter in the compression algorithm. The former deals with the delay in the wireless link, the latter with the overall delay, and will therefore always be higher than the former.
- The parameter that determines the maximum bit rate per second is set to a high value to avoid packet loss. In doing so we make the resulting bit rate an output parameter that has to be evaluated.
- When using the link protocol simulator the error generation uses an additive white Gaussian noise distribution. The width of this distribution determines the bit error rate.
- Only packets that never have been sent and are missing at the receiving side of the link are filled with ones, the other packets are sent on including the errors. Using zeros instead of ones might emulate a start code of the compression algorithm, which consists mainly of zeros. The link is not able to send a packet either when the bandwidth is already fully used or when a packet had to wait so long for transmission that it has become overdue. The latter can happen for instance when the bit error rate is very high and many other packets had to be retransmitted. If the maximum number of retransmission for a packet has been reached it will be sent, even if it still has errors.
- When using the link simulator, padding with ones is used to fill the last fragment of a frame. This makes the bit rate for experiment runs using the link simulator slightly higher.
- The maximum number of retries for sending the data packets over the link is set to three, that is one try and two retries maximum. When there are no errors, no retransmissions will be done so the bit rate only increases when necessary.
- The error correcting codes are always present when using the link simulator and in the current implementation enforce an increase in bit rate of about 21%.
- The performance of the link protocol is not dependent on the video content or video parameters, unlike the performance of the compression algorithm.

6.4.3 Compression Algorithm Parameters

The used error resilient compression algorithm is the one developed in the Mobile Multi-media Communication project, which is described in Section 5.3. We now describe the features of this algorithm and parameter settings that are used in our experiments.

- The quantisation parameter for the first frame is 31, the coarsest one. This is done in order to reduce the high bit rate for this frame, which is always intracoded. This is necessary, even when a tenfold bandwidth is available for this frame. The only exception is when the remaining part of the sequence is coded with very high quality; in that situation the bit rate for the first frame could be even be slightly lower than average.
- A synchronisation word is inserted every row of macro blocks. This value is chosen because the minimum of once per frame is not enough when errors might occur, and the maximum of one every macro block creates an unacceptable bit rate increase. This insertion is also used when the error resilient source coding algorithm is not used, because this way to increase the error resilience is allowed within the H.263 standard.
- The coding mode is intercoding, unless the error resilient compression algorithm is used. In that case, a number of *A* previous frames is used to compose the prediction frame. *A* is dependent on the bit error rate. The determination of *A* is discussed in Section 5.4.2.
- The coefficients for composing the prediction frame are all taken equal, this means an averaging of the *A* previous frames.
- The maximum number of macro blocks that are requested to be intracoded is 396, which means that intracoding of all macro blocks can be requested.
- No extra features of the H.263 standard, as described in the Annexes of the standard, are used [16].
- The range of the search for a motion vector we chose is 8 pixels; an average value between the minimum of 0 and maximum of 15.
- Error detection is applied only when the error resilient compression algorithm is used.
- Error concealment is applied only when the error resilient compression algorithm is used, and is done on the basis of the results of the error detection. An erroneous macro block is replaced with the one on the same position in the previous frame.

6.4.4 Determination of Q

We want to determine a *Q* value for which a minimum quality image is obtained and a *Q* value which makes a clear difference in visual quality with the low *Q* image.

We will use these two values to distinguish between spending bits on the image quality, using a low *Q*, and not, using a high *Q*. A low *Q* will result in a higher peak signal-to-noise ratio and a higher bit rate.

The range of possible values for *Q* is from 1, yielding high quality and a high bit rate, through 31, yielding low quality and a low bit rate.

To support our choice, we made figures Figure 51 and Figure 52, which show the peak signal-to-noise ratios and bit rates for 5 different values of *Q*.

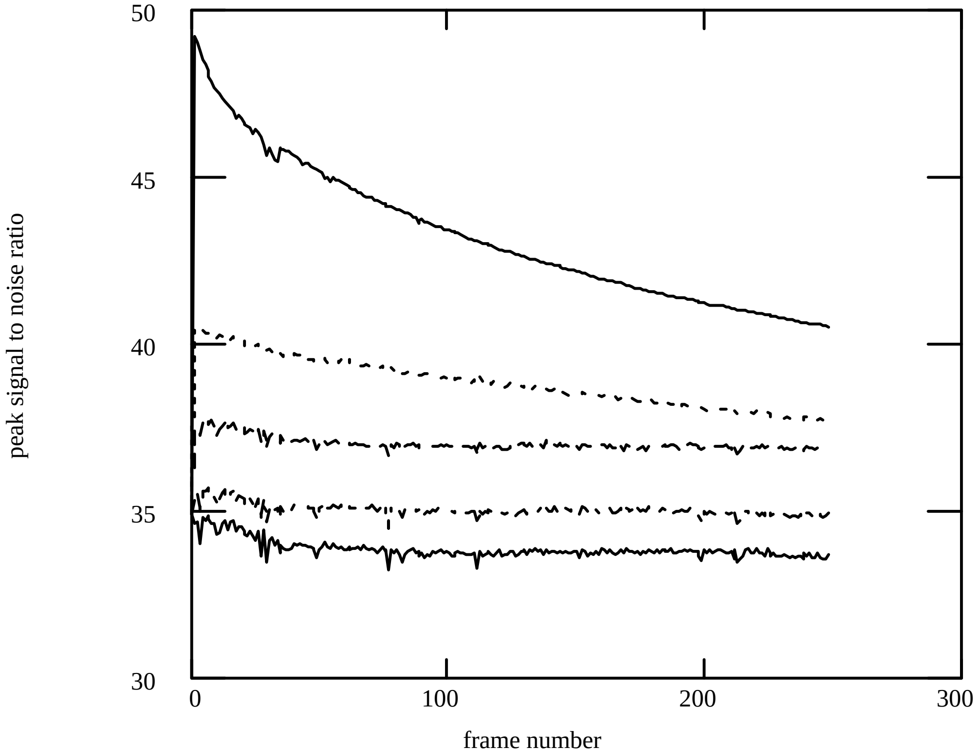


Figure 51 The peak signal-to-noise ratio as a function of the frame number for different values of Q: 1 (top), 5, 10, 20 and 31 (bottom).

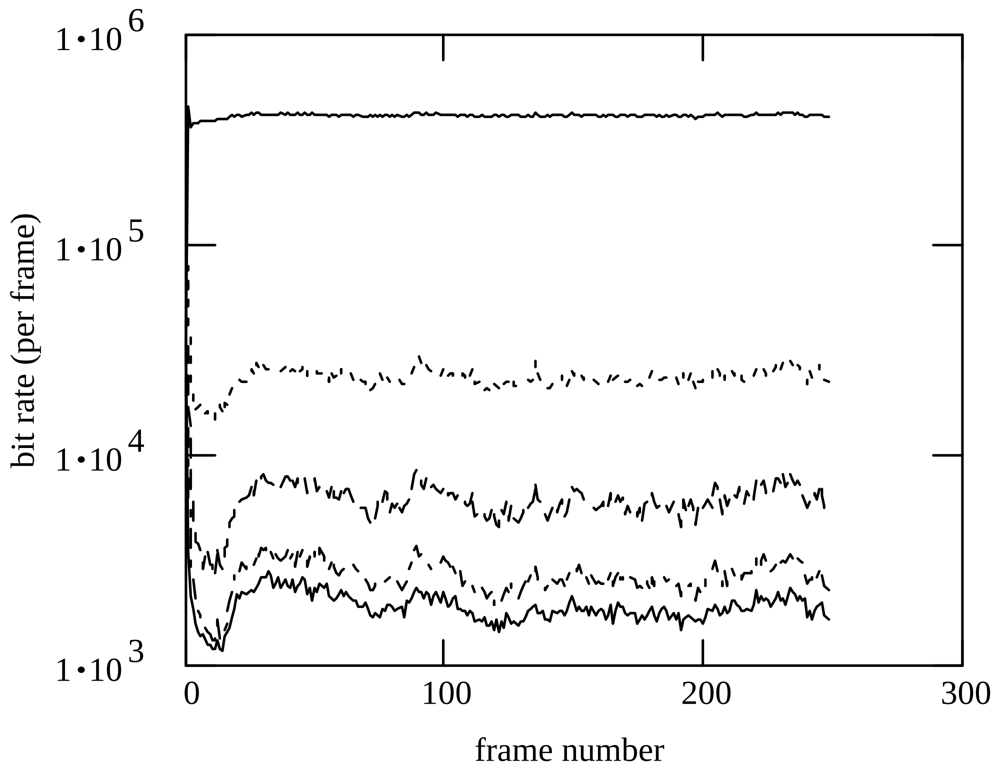


Figure 52 The bit rate, logarithmic, as a function of the frame number for different values of Q: 1 (top), 5, 10, 20 and 31 (bottom).

From these graphs we determined the peak signal-to-noise ratio and bit rate as a function of the quantisation parameter, see Figure 53 and Figure 54.

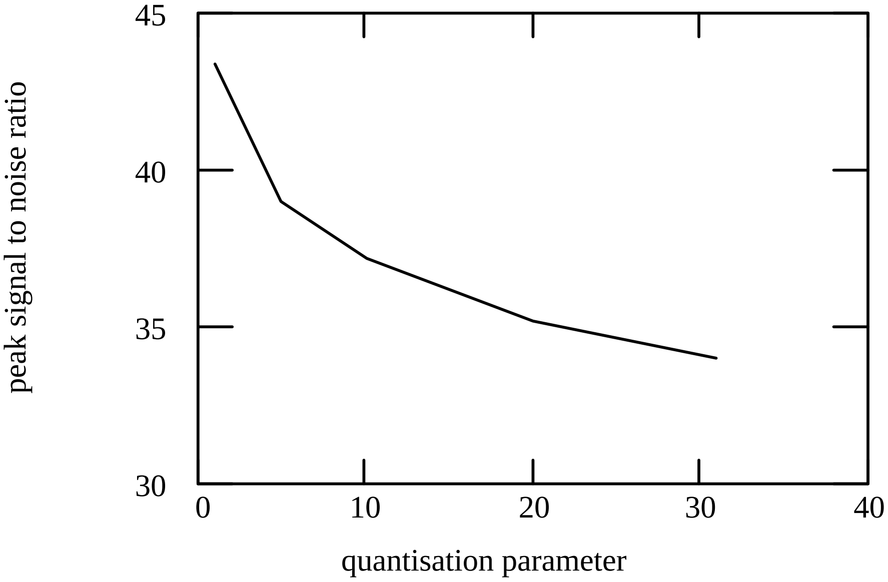


Figure 53 The average peak signal-to-noise ratio as a function of the quantisation parameter.

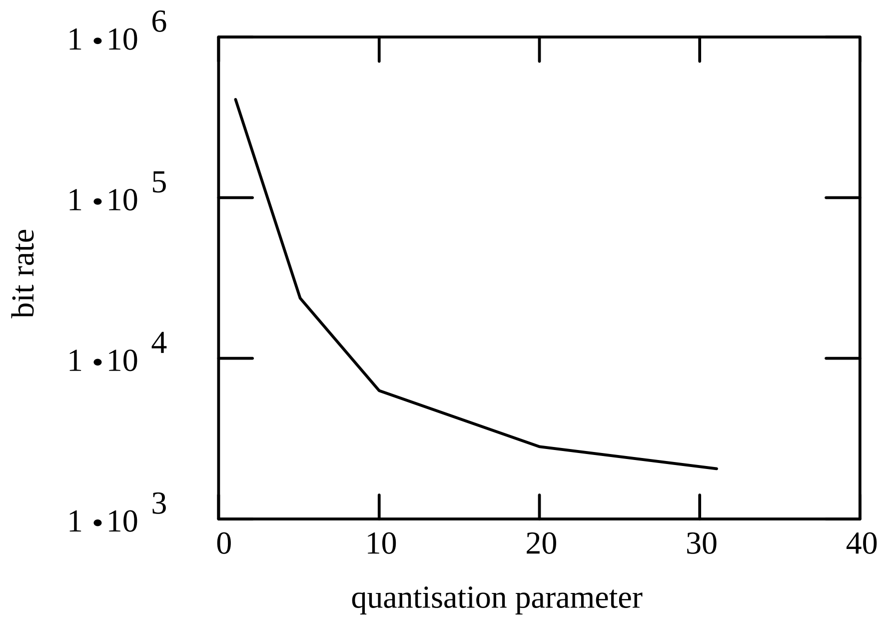


Figure 54 The average bit rate as a function of the quantisation parameter.

From these results we can deduce the exponential relation between the peak signal-to-noise ratio increase and the bit rate increase, both relative to the result for Q = 31. For the peak signal-to-noise ratio we used subtraction, because a peak signal-to-noise ratio is already relative, and for the bit rate we used division. We used equation (22),

$$I_{rel}(Q) = \frac{S(Q) - S(31)}{R(Q) / R(31)} \tag{22}$$

where I_{rel} is the relative increase, *S(Q)* the peak signal-to-noise ratio using quantisation parameter *Q*, and *R(Q)* the bit rate using quantisation parameter *Q*. This way we obtained Figure 55.

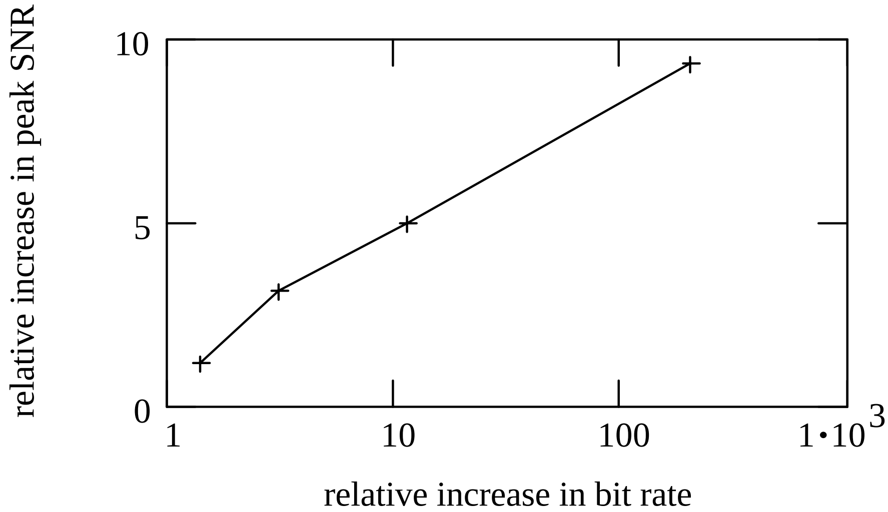


Figure 55 The relative increase in peak signal-to-noise ratio as a function of the relative increase in bit rate (logarithmic).

We can see for instance that, to obtain an increase in the peak signal-to-noise ratio of 5 dB, we need to increase the bit rate by a factor of 10. In other words, we either need a lot of extra bits or have only a slightly better image quality. To be not too far off the bit rate increase due to the link protocol or compression algorithm, we choose to work with the following two values:

$$Q_{basic} = 15 \qquad Q_{high} = 5$$

This is also close to the value of Q = 10, which is usually taken to obtain an average image quality.

6.5 Optimal Bit Distribution

In Section 6.4 we described our experimental set-up. In this section we describe the experimental results that can be used to determine the optimal distribution of the available bits over the different error resilience techniques. As a function of the bit error rate, we determine the peak signal-to-noise ratio and the bit rate for each of the eight combinations of the three methods for increasing the image quality. A fixed solution for the optimal bit distribution can not be given, since in each communication situation the requirements for bit rate and image quality can be different. Therefore, the results we obtain form a guide line which error resilience method to use in what circumstances. We use the bit rate and the peak signal-to-noise ratio as evaluation parameters, while we vary the bit error rate and the method used to increase the image quality.

The results are organised as follows. We have three error resilience techniques; image quality, that is decrease quantisation coarseness, an error resilient link protocol and an error resilient compression algorithm. Each of these techniques are either used or not used, so we have eight combinations, called methods, each yielding an average peak signal-to-noise ratio and an average bit rate. These results are obtained for different values of the bit error rate, which gives us two graphs for each method. This could give us two plots, one for the peak signal-to-noise ratio and one for the bit rate, each with eight graphs. However, for the purpose of a clearer presentation, we present the results with and without the additional image quality in different plots. We now show our results and describe for each plot the most important features that can be observed.

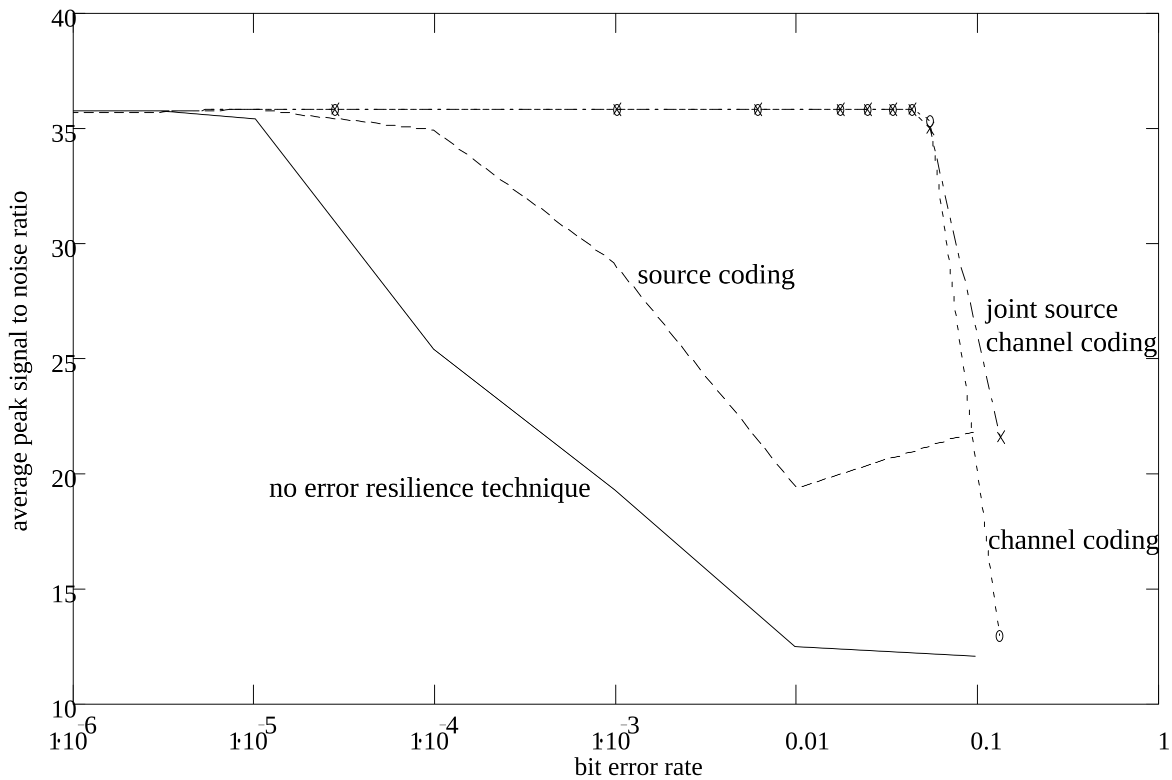


Figure 56 The average peak signal-to-noise ratio as a function of the bit error rate for different error resilience methods: none, presented by a solid line, error resilient source coding, presented by a dashed line, error resilient channel coding, presented by a dot-dashed line with circular symbols, and combined error resilient channel and source coding, presented by a dotted line with cross type symbols. For the graphs in this plot, no additional image quality is used (Q = 15).

Ad Figure 56: For low bit error rates the peak signal-to-noise ratio is hardly effected. At 10^{-4}, however, for the method not using any error resilience technique it has already decreased by 10 dB, and at $2*10^{-3}$ the source coding also. The channel coding and the combined channel source coding maintain a maximum peak signal-to-noise ratio up to very high bit error rates, also because in our set-up two retransmissions are allowed for each data packet. These results have to be compared with the next figure, which shows the bit rate increase for each of these methods. For very high bit error rates, none of the techniques work really well; the combined source channel coding is the least dramatic, although this method too drops more than 10 dB at a bit error rate of 0.13.

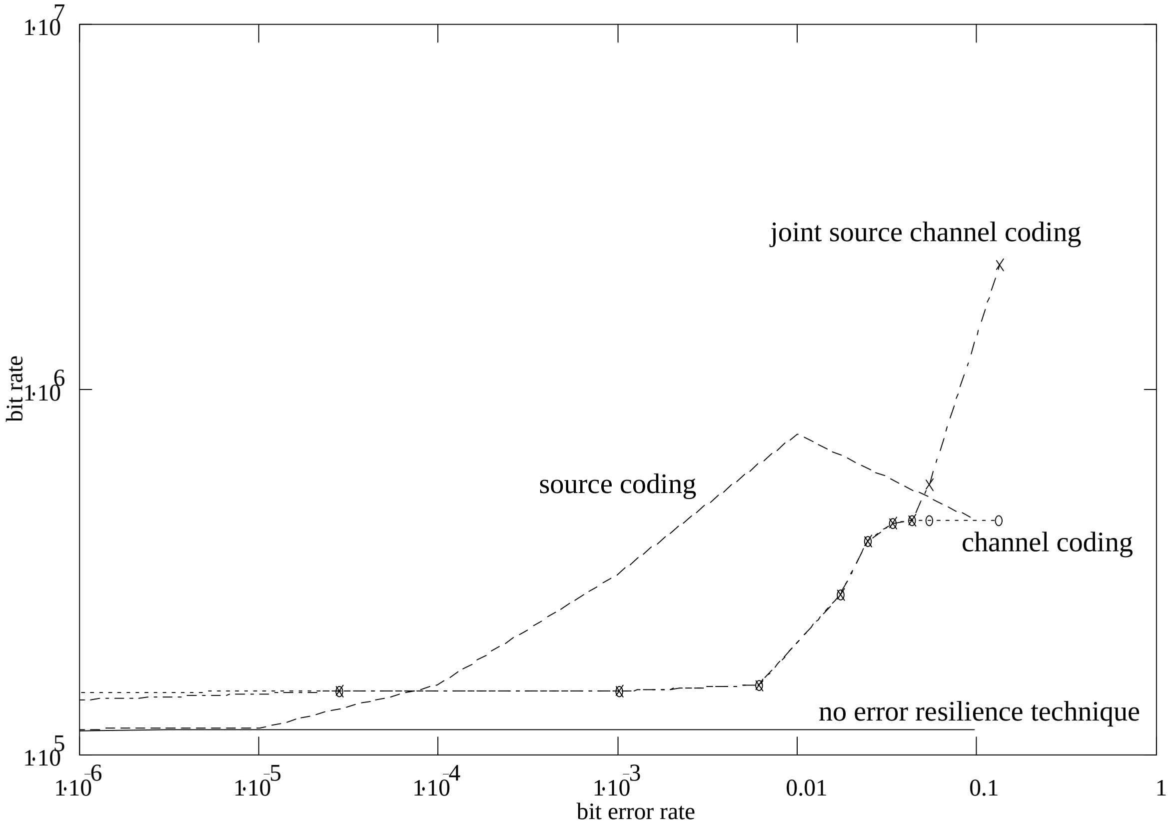


Figure 57 The bit rate as a function of the bit error rate for different error resilience methods: none, presented by a solid line, error resilient source coding, presented by a dashed line, error resilient channel coding, presented by a dot-dashed line with circular symbols, and combined error resilient channel and source coding, presented by a dotted line with cross type symbols. For the graphs in this plot, no additional image quality is used (Q = 15).

Ad Figure 57: The bit rate for the method not using any error resilience technique is constant and minimal, of course. The bit rate for the error resilient source coding increases smoothly, first slowly, then faster and faster to almost a factor of 8, and then it decreases again. Up to almost 10^{-4}, the bit rate is lower than the bit rate for channel coding. For the decrease after 10^{-2} we have no explanation, except that at bit error rates of 10^{-2} and higher, the number of requested intracoded macro blocks, which mainly cause the increase in bit rate, is unpredictable because the image quality is so low. For the error resilient link protocol and the combined source channel coding, the bit rate has a higher minimal value due to the error correction codes. From $5*10^{-2}$ up, the retransmissions of packets also increase the bit rate substantially. For the highest bit error rates the bit rate for combined source channel coding goes up rapidly to high values because the source coding part has to deal with the errors that the channel coding can not deal with now.

The next figures give the same results, but now with additional image quality, that is *Q* is 5.

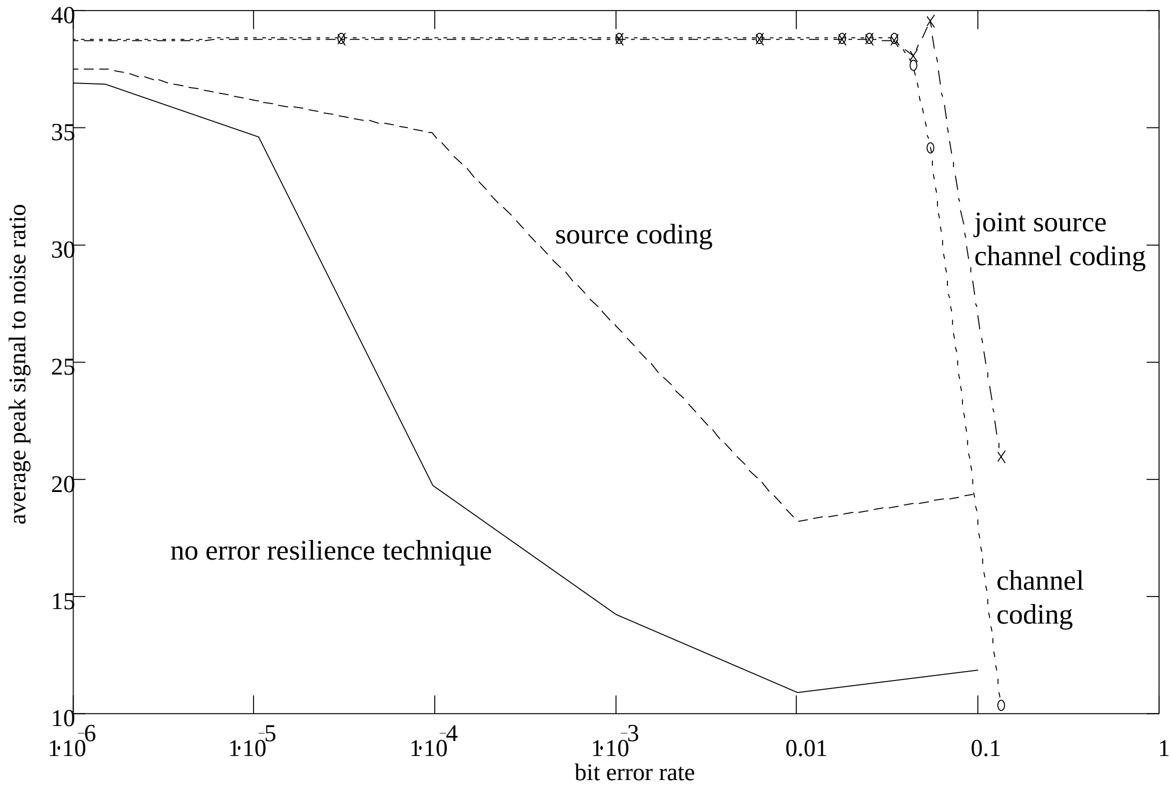


Figure 58 The average peak signal-to-noise ratio as a function of the bit error rate for different error resilience methods: none, presented by a solid line, error resilient source coding, presented by a dashed line, error resilient channel coding, presented by a dot-dashed line with circular symbols, and combined error resilient channel and source coding, presented by a dotted line with cross type symbols. For the graphs in this plot, additional image quality is used (Q = 5).

Ad Figure 58: At low bit error rates we already see a degradation in image quality for the method not using any error resilience technique and for error resilient source coding. This is because due to the higher bit rate, the probability that any error is present in the compressed test sequence is higher than in the case of the coarse quantisation. Channel coding and combined source channel coding are performing well again. For the highest bit error rates, the peak signal-to-noise ratio for the channel coding decreases dramatically, while the combined technique first can maintain the image quality: the errors that the link protocol could not correct through retransmissions are dealt with by the compression algorithm. However, at a somewhat higher bit error rate the combined method also collapses, although the peak signal-to-noise ratio is about 10 dB higher than the channel coding. The fact that the combined method collapses so soon after the channel coding collapses is due to the fact that the channel coding can either correct most or all of the errors or very few to none, due to the structure of the protocol. The feature of the combined method being somewhat better than the channel coding only is now more pronounced than in the case of *Q* = 15 because the absolute number of errors is higher due to the higher bit rate and the number of retransmissions is therefore higher, too. Again, we have to compare these results with the increase in bit rate shown in the next figure.

Up to a bit error rate of 10^{-5}, the method using a lower value for *Q*, and therefore a higher image quality, outperforms all other methods in peak signal-to-noise ratio with a *Q* of 15. At higher bit error rates one of the error resilience methods is necessary.

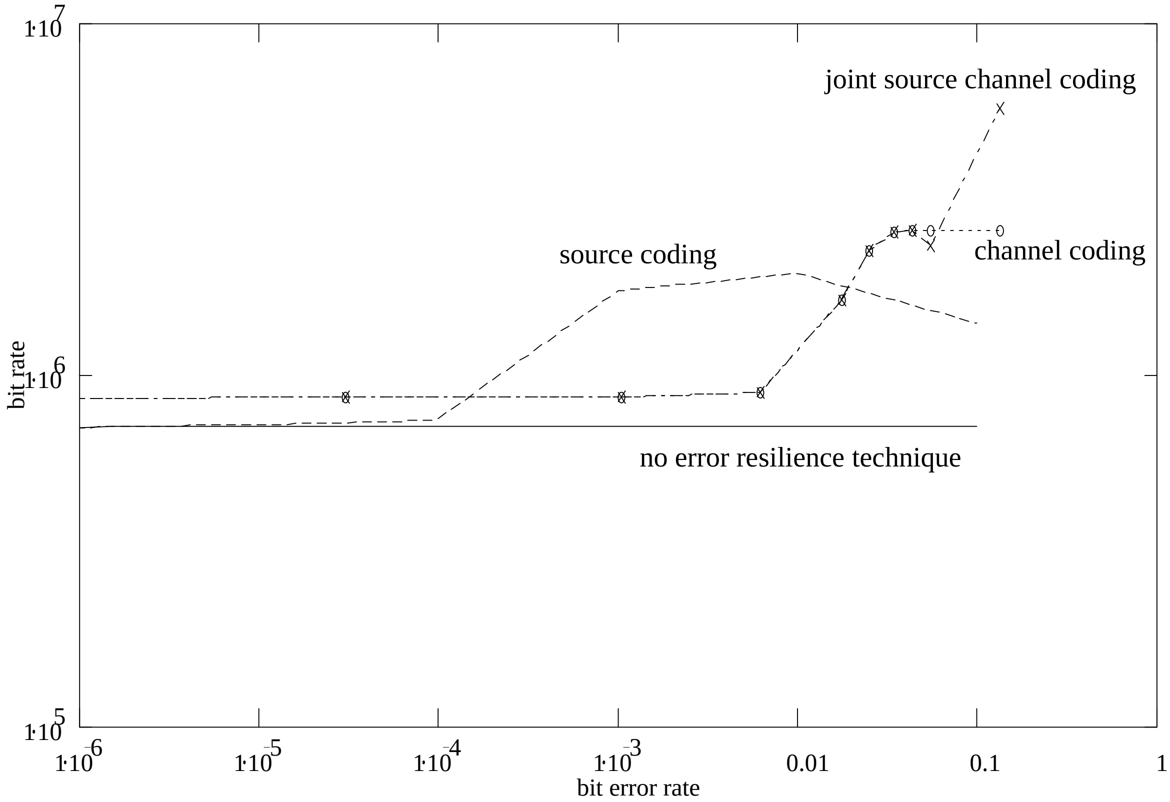


Figure 59 The bit rate as a function of the bit error rate for different error resilience methods: none, presented by a solid line, error resilient source coding, presented by a dashed line, error resilient channel coding, presented by a dot-dashed line with circular symbols, and combined error resilient channel and source coding, presented by a dotted line with cross type symbols. For the graphs in this plot, additional image quality is used (Q = 5).

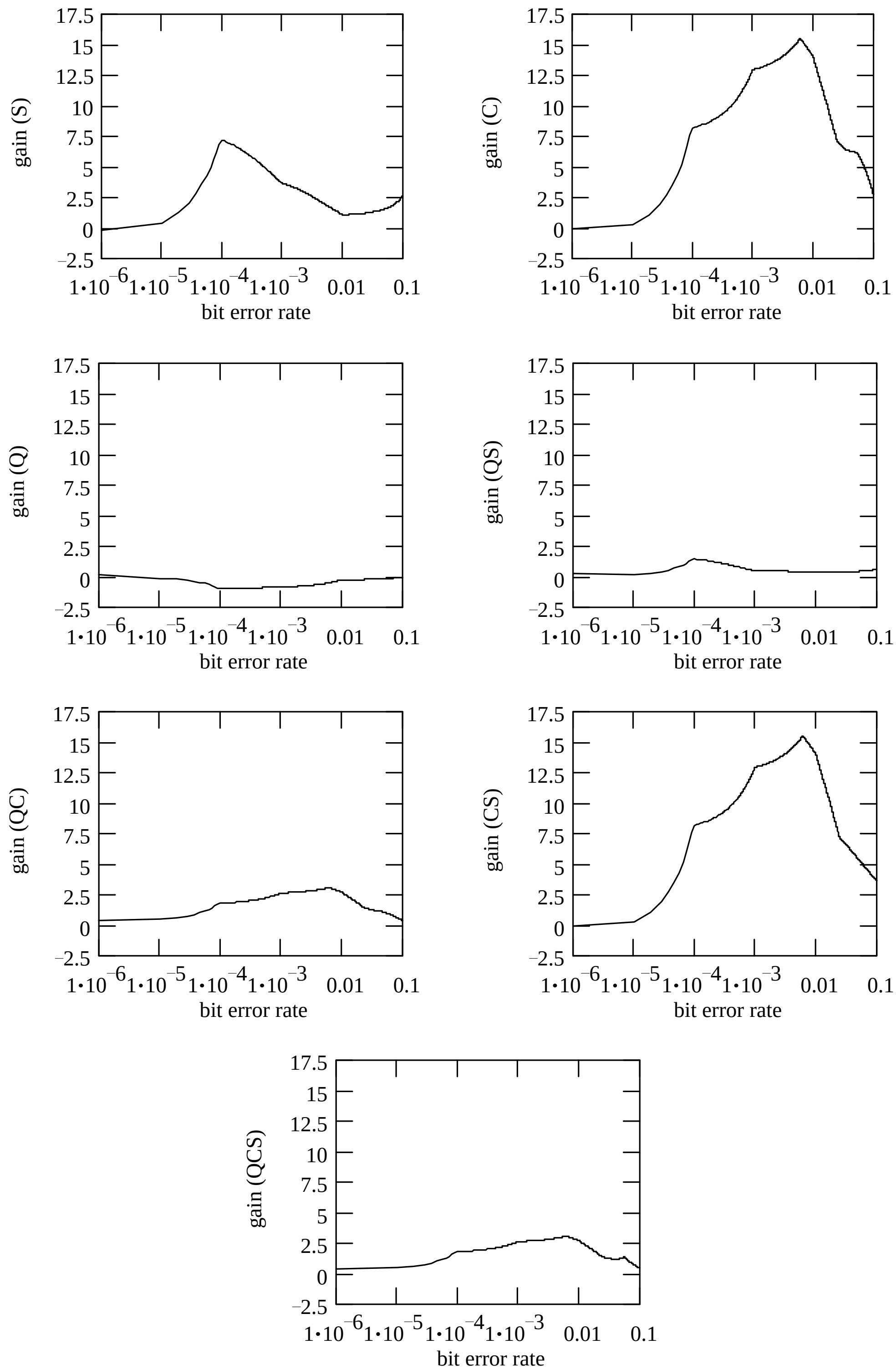
Ad Figure 59: We see more or less the same features as in the plot for *Q* = 15: a constant value for the method not using any error resilience technique, an increasing value for source coding, a higher minimal value for channel coding. For channel and combined channel and source coding at high bit error rates, we also see the same effect as in the plot for *Q* = 15. The bit rate is overall higher than for *Q* = 15 as expected.

Just as we did when we determined the *Q* value for the experiments, we make for each error resilience method a graph of the relative increase, the gain, in peak signal-to-noise ratio divided by the increase in bit rate, with respect to the method not using any error resilience technique at *Q* = 15. For the following results we used

(23)
$$gain(method) = \frac{S(method) - S(notechnique)}{R(method) / R(notechnique)}$$

to determine the gain; we use the same notation as for the Q determination. In (23) *notechnique* stands for not using any additional technique to increase the image quality. The results are shown in Figure 60.

Figure 60 The seven graphs below show the increase in signal-to-noise ratio divided by the relative increase in bit rate as a function of the bit error rate. The results for each method are plotted separately. If a technique has been used to obtain the result, it is indicated by a letter: Q for additional image quality, S for source coding and C for channel coding.



Using these graphs we can derive the following results. Notice that from equation (23) we can see that negative values arise from a decrease in signal-to-noise ratio when there is an increase in bit rate.

- We see that in general channel coding and combined source channel coding perform well, and source coding a little less. At higher bit rates, that is with additional image quality, these values drop, which is also shown in the graph of gain(Q) which is mainly negative, so adding bits in the presence of transmission errors on average actually decreases the peak signal-to-noise ratio.
- On the whole, we see that peaks in these graphs are mainly found in the area of bit error rate 10^{-4} through 10^{-2}, which means that spending bits on error resilience techniques in that area is most effective.
- In the gain(Q) graph, the gain is low for low bit error rates. This means that, even when there are only a few or no errors, applying a lower Q to increase the image quality does not increase the peak signal-to-noise ratio very much. This is similar to what we have seen for the Q determination in Section 6.4.4.

The maximum gain, irrespective of the method used is shown in Figure 61. We see that spending bits on error resilience techniques is most effective around a bit error rate of 10^{-2}; this is where the channel coding reaches its limit. From about $2*10^{-5}$ up, the techniques are efficient, below this value the gain is minimal.

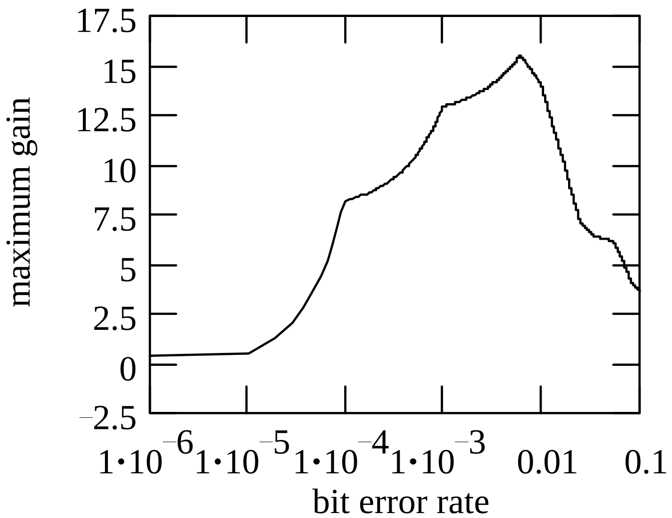


Figure 61 The maximum gain, irrespective of the used method.

We now summarise the conclusions that we obtained from the results of the experiments. These results concern single-bit errors. For extremely low bit error rates, when the error is not noticeable, the additional image quality should be applied, even though the gain is low. It also has been shown that for the low bit error rates, a lower bit rate, so no additional image quality, reduces the impact of errors. For moderately low bit error rates, about up to 10^{-4}, the source coding technique is best because it roughly maintains the image quality at a lower bit rate than the channel coding, which also maintains the image quality. For higher bit error rates, that is up to $2*10^{-2}$, channel coding is very efficient. For lower bit error rates it is also superior if the increased bit rate is no problem or when no image degradation is allowed. At the highest bit error rates, more than $2*10^{-2}$, the combined source channel coding is superior, although at bit error rates higher than about 10^{-1} even these results are not very good. This is depicted in Figure 62. We have not been able to find a similar approach to the joint source channel coding problem in literature and therefore we can not present a comparison with results from literature here.

no error resilience technique; low image quality or high image quality	error resilient source coding	error resilient channel coding	combined error resilient source channel coding

10^{-8} 10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 10^{-0}

--> bit error rate -------------------------------------->

Figure 62 The optimal error resilience technique as a function of the bit error rate, for single-bit errors.

For a simple communication system that should be error resilient as well, channel coding only would probably be a good average choice. For channel coding, the error resilience is very good up to high bit error rates, although the bit rate for very low bit error rates is higher than necessary and at very high bit error rates a higher error resilience could be achieved using additionally error resilient source coding.

It is important to notice that these results have been obtained for single-bit errors only. In practice, however, packet loss and burst errors will also occur and are an important cause for the degradation of the image quality in many communication situations. We address the error resilience in terms of the peak signal-to-noise ratio and bit rate for packet loss and burst errors in the next section. The results there have to be added to the main results we obtained up to now, as shown in Figure 62.

Finally, we have to remark the following. The results we have shown have been obtained with a specific set of parameters using a specific error resilient compression algorithm and a specific error resilient link protocol. It is therefore necessary to address the extension of the results to a more general situation. This is done in Section 6.7.

6.6 Packet Loss and Burst Errors

Up to now, we were looking at single-bit errors only. Now we want to deal with burst errors and packet loss. Note that in the MPEG-4 definition of residual error condition classes [13], these errors are indicated as type 2 and type 3 errors. Of these two types of errors we address mainly packet loss, because this type of error is the one that differs most from single-bit errors and is the most severe. Packet loss can occur when

- There is a peak in the bit rate. This will cause a bit rate momentary exceeding the bandwidth and packets have to be dropped. Whether the bandwidth limit is exceeded also depends on the time period over which the bandwidth limit is calculated, which could be for instance one frame. A slow increase in bit rate could be compensated by changing the quantisation parameter *Q* when the bit rate starts to go up and recompression of the data with the changed *Q*. However, according to the ITU standard [16], the *Q* can be changed steeply only per frame

and only moderately within a frame, to avoid coding artefacts within the frame. Therefore, handling the peak in the bit rate by means of *Q* is in most cases not possible.
- The transmission packet is overdue, according to the delay constraint. This occurs when the bit error rate is high and there are many retransmissions of other packets, for which the packet might have to wait.
- There is network congestion. This occurs at the network level. This is an important issue for video over the internet [33][35].

Notice that when the number of retransmissions would exceed the maximum, the erroneous packet is transmitted as it is, and the packet is not dropped, even though it does contain errors. This occurs when the bit error rate is very high or the number of allowed retransmissions low.

In our experimental set-up, the size of the error, which is due to dropping packets, is one or more fragments, as defined in Section 3.4. A fragment consists of 424 bits. The information in a fragment can consist of several macro blocks, depending on the quantisation parameter *Q*, the coding mode and the video content. For instance, if we would have a bit rate of 3960 bits per frame we would have 10 bits per macro block and therefore about 42 macro blocks per fragment, which is about two rows of macro blocks in a CIF image. However, it is difficult to predict the actual number of corrupted macro blocks in the reconstructed image when a fragment is lost. This is for a number of reasons:
- The number of bits can be very different for each macro block. For instance, if there are areas in the video scene without motion, many of the macro blocks can be coded with just one bit to indicate no change, while on the other hand an intracoded macro block can fill up a whole fragment.
- The loss of a fragment might cause degradation in only a few macro blocks in the image because if, for example, information on “no change” of a macro block is lost, this will not degrade the image very much. If on the other hand intracoded data is lost, the loss of this information will degrade the image substantially.

Once a packet is dropped, the link cannot restore the situation or handle the effects of the errors. A dropped packet is not left out at the receiving side, but stuffed with ones, not zeros, as these could cause emulation of a synchronisation word, which uses mainly zeros.

These kinds of errors, for instance the size of a row of macro blocks, can be handled by the error resilient video compression technique. For this corrupted row of macro blocks all macro blocks are requested to be intracoded in some future frame. This will, however, at some point give rise to a substantial increase in bit rate. This increase in bit rate could in turn be well beyond the bandwidth and more packets will be dropped, which leads to more requests for intracoding of macro blocks, and so on.

This process can be stopped if the following technique would be used. When the number of requested intracoded macro blocks exceeds a certain value, only the *Q* value for that frame should be set to a low value. The next frame is then coded normally. Coding artefacts will then be introduced in this frame due to the very coarse quantisation. The artefacts will, however, be hardly visible because they occur only in one frame. The next frame will still have a bit rate that is somewhat higher because the prediction frame, which is the frame with the coarse quantisation, has a poor quality. This increase is, however, moderate. This way at the encoder side *Q* would be made dependent on the number of macro blocks that are requested to be intracoded by the decoder.

In our experiments, however, we wanted to examine the ability of the error resilient video compression algorithm to deal with an error caused by packet loss, apart from other coding features that arise when packet loss occurs. Therefore the set-up was more or less opposite to the situation described above. We limited the bandwidth only for one frame in the sequence and evaluated the signal-to-noise ratio and bit rate when only channel coding was used and when combined source channel coding was used. We first did this for frame 20 of the test sequence. The result is shown in Figure 63. The limited bandwidth for frame 20 resulted in 27 fragments lost, each consisting of 424 bits. Unlike the example described above, we now have a relatively high bit rate and the 27 lost fragments now consist of about 7 rows of macro blocks. If only channel coding is used the peak signal-to-noise ratio immediately drops by about 6 dB and the original value is not recovered, while the additional use of source coding results in a drop of about 3 dB in the first few frames after the introduced packet loss, but the original value is almost completely recovered after less than 10 frames. This shows the efficiency in error resilience of the source coding algorithm in the presence of packet loss.

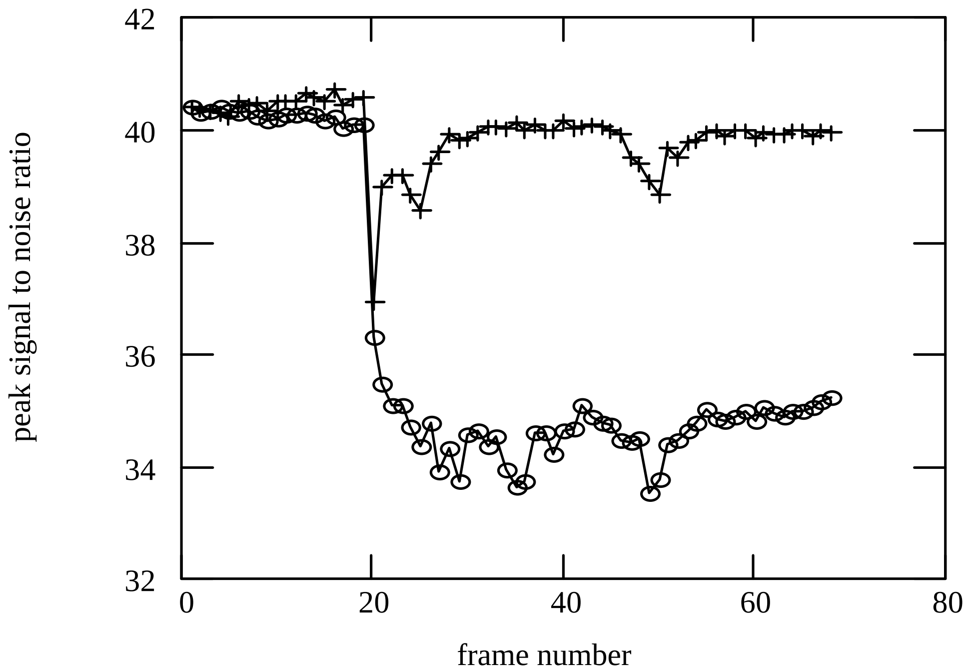


Figure 63 The peak signal-to-noise ratio as a function of the frame number. Packet loss was induced in frame 20. Shown are the results for channel coding, denoted as circles, and for combined source channel coding, denoted as crosses.

We also induced more packet loss in the same way, each time with shorter intervals. In the complete test sequence of 250 frames, we induced packet loss in frame 20, then after 50 frames in frame 70, again after 50 frames in frame 120, then with an interval of 10 in frame 130, then in frame 170 and 173, having an interval of 3 frames, and finally in the 3 consecutive frames 220, 221 and 222. The results are shown in Figure 64.

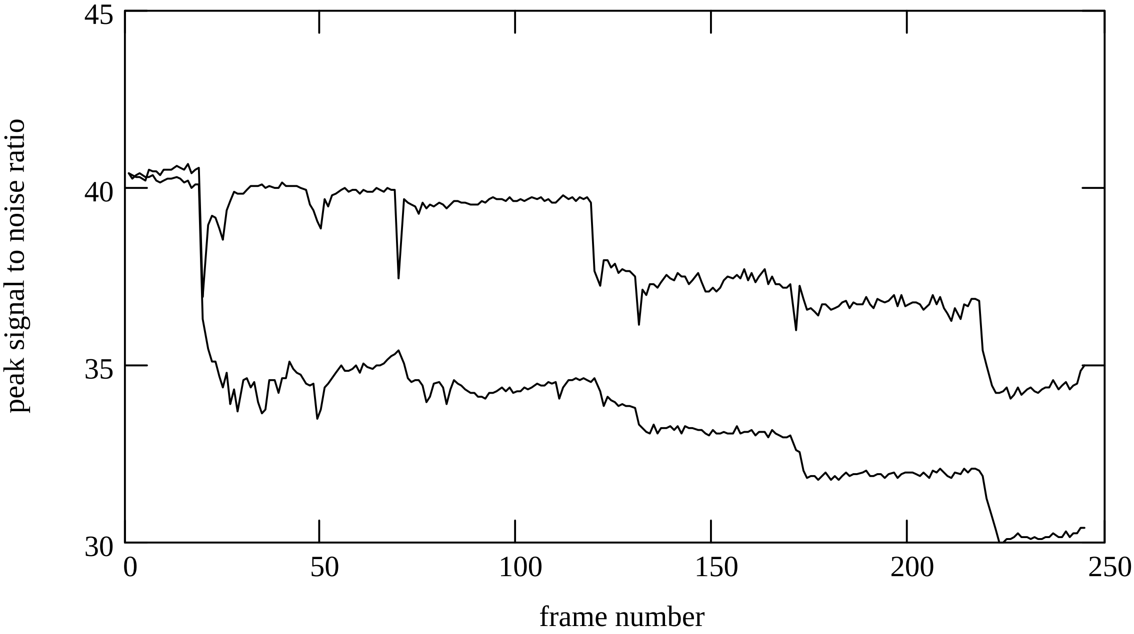


Figure 64 The peak signal-to-noise ratio as a function of the frame number. Packet loss was induced in frames 20, 70, 120, 130, 170, 173, 220, 221 and 222. Shown are the results for channel coding, bottom, and for combined source channel coding, top.

We see that the peak signal-to-noise ratio for the channel coding only drops more and more as packets are lost. It is not affected by the error in frame 70; apparently the bandwidth there is not constrained by our limit. This is opposed to the case for the combined source channel coding, which has a slightly higher bit rate, see also Figure 65. The combined technique recovers from the packet loss in frames 20 and 70, but not in frame 120, where apparently the error affected a vulnerable area in the video data stream, like a header. It does recover from the error in frame 130 and somewhat less from the errors in frames 170 and 173. The three consecutive frames with packet loss are also prove to be damaged too severely also for this technique and the peak signal-to-noise ratio drops substantially.

Figure 65 shows the bit rate as a function of the frame number for each of the two techniques. We see that overall the bit rate is higher for the combined source channel coding technique. We can also see the peaks for some of the frames containing errors caused by packet loss, like frame 70 and frame 130. The overall average increase in bit rate for the combined technique is less than the variance in bit rate for both techniques. This means that if the link can handle the bit rate variance, it is probably also able to handle the occasional increase in bit rate when burst errors occur and the combined technique is used. However, an increase in the average bit rate has to be expected.

Summarising we can say that for bursts in a single frame, the combined error resilient channel source coding technique performs well and the image quality recovers almost completely within several frames. The increase in bit rate due to the use of combined technique is in the order of the variance in bit rate present in the original coded bit stream. Since the error resilient channel coding cannot deal with packet loss and burst errors at all, these results apply not only to the combined error resilient channel source coding technique but also to error resilient source coding only. For packet loss and burst errors, the performance of the error resilient source coding depends on the number of errors as we have seen, but will be the best solution for all error rates because none of the other techniques can handle such errors, except for the combined source channel coding which has a somewhat higher bit rate.

We now see that the optimal technique apparently depends not only on the bit error rate but also on the types of errors that are expected in the communication situation. If no information is available about the number and type of errors to be expected, the combined error resilient channel source coding technique offers the best solution, since it performs well for almost all of the range of bit error rates in the single-bit error case but also performs well in the case of the occurrence of packet loss and burst errors.

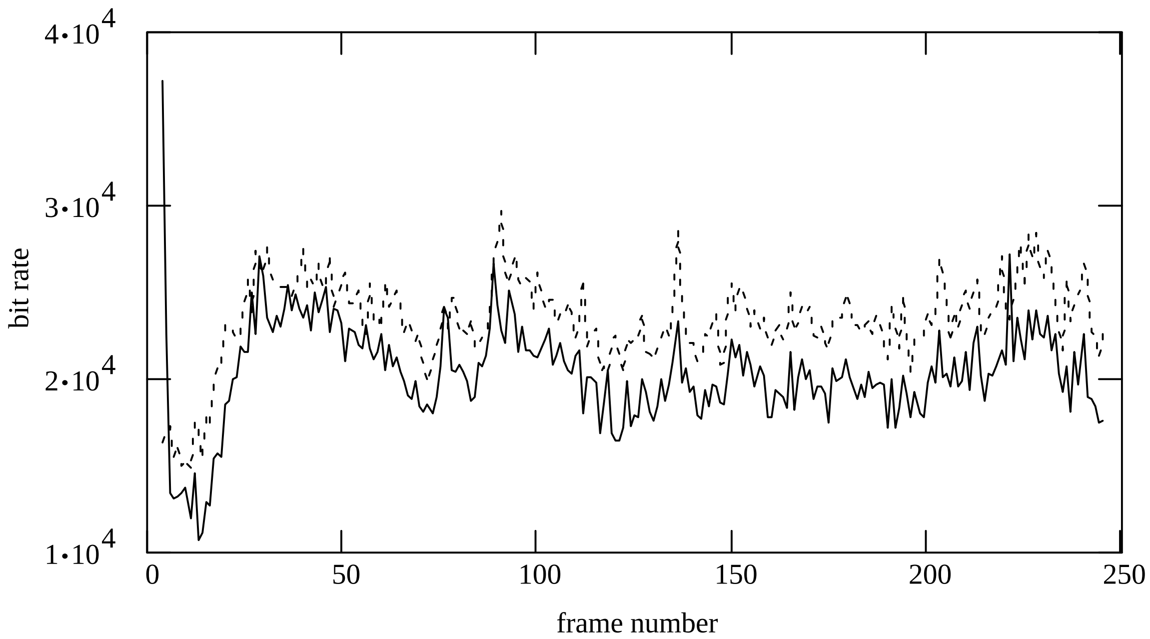


Figure 65 The bit rate per frame as a function of the frame number corresponding to Figure 64.

6.7 Generalisation

We obtained the results using a certain kind of compression algorithm, a certain kind of link protocol and certain values for several parameters that were fixed at the beginning of the experiments. In this section we address the problem of the generalisation of the results.

One way to generalise results is to switch to theoretical analysis, which would involve the problems we mentioned earlier. Another way is to run the experiments again, but now varying all the parameters. However, this is computationally far too intensive, especially where it concerns obtaining enough data per varying parameter to support statistical evaluation tools.

The way we approach the generalisation problem here is to look separately at each of the parameters and predict in which way the results we have obtained would be different if this parameter had a different value. Such a prediction is mostly based on earlier, limited, experiments where this parameter was changed and on considerations concerning the structure and set-up of the system and the experiments. The independent treatment of the parameters is only allowed when the parameters do not depend on each other. This is an assumption we make here. It is partly supported by the earlier, limited, experiments. For the cases in which the assumption of independence is clearly not correct, we try to take into account the interdependencies as much as possible.

We first address the influence of the general parameters, then the influence of the parameters of the error resilient link protocol and finally the influence of the parameters of the error resilient video compression algorithm. It is important to keep in mind that the following treatment of the influence of the parameters has a preliminary character.

6.7.1 The General Parameters

We present a list of the fixed general parameters that were used, and a prediction of the influence on the experiment results if the parameter had had a different value.

- The test sequence. A different sequence content with, for instance, more motion would have the effect of a higher bit rate, less use of motion vectors and more intracoded macro blocks. This might lead to a sequence that is slightly less sensitive to errors, because especially intercoding and motion vectors are very vulnerable.
- The number of encoded frames. A higher number of encoded frames would statistically produce more significant results, and would also reduce the influence of the intracoded first frame, which has a high bit rate. The computational intensity would obviously go up.
- The frame format. A smaller frame size results in less bits per frame, and therefore in less errors per frame, at the same bit error rate. On the other hand, if one macro block is affected a relatively larger part of the frame is affected, because the absolute size of one macro block is fixed to 16 by 16 pixels. Therefore, the influence of the frame format is complex.
- The frame rate. A smaller frame rate results in a worse prediction and therefore less motion vectors and more intracoding. It also results in a smaller bit rate and therefore less errors. For the error resilient video compression algorithm, the prediction using the composed prediction frame becomes worse.
- The round-trip delay. A higher delay would result in a less effective error resilience for the source coding technique, because the time between the request for an intracoded macro block and the arrival would be longer, and, as the error concealment is not perfect, the remaining errors propagate over a longer period of time.
- The error generation. A type of error generation with, for instance, a much more bursty character would result in less error resilient channel coding, as this technique does not handle such errors well.

- The block size. This is always fixed to 8 by 8 pixels, while four blocks make up one macro block. A smaller macro block size would require more block overhead and motion vectors but would also increase the effectiveness of the error resilient source coding. This is because for each error, of a whole macro block an intracoded version is requested, and a smaller macro block would decrease the probability that of a correct image area an intracoded version is requested.

6.7.2 The Link Protocol

We present a list of the fixed link protocol parameters that were used and a prediction of the influence on the experiment results if the parameter would have had a different value.

- The bandwidth. Unlike in the real world situation, we did not limit the bandwidth we used in our experiments and the bit rate was used as an output parameter. Although rate control is imperative in real world situations, designing such a rate control is complex, and its influence on the results hard to predict. Limiting the bit rate to a certain value, for instance for every frame, would result in dropping of packets whenever the compression algorithm does not sufficiently compress the frame, for instance due to a lot of sudden motion in the scene. A solution is to use rate control in the compression algorithm. However, the rate is then controlled through the quantisation parameter, which can only be changed unlimitedly for a whole frame. Within one frame the changes are limited. So if the size in bits of a compressed frame would turn out to be too large, it would have to be compressed again with a higher value for the quantisation parameter. However, normally there is no time for this due to real time constraints, and we did not use such a rate control in our experiments.
- The maximum number of retries. Increasing this number would result in a link with less errors at the cost of an increase in bit rate. This would especially affect the high bit error rate area because we have allowed in our experiments a number of retries which is already enough to handle errors up to high bit error rates.
- The maximum link delay. A decrease of this delay would result in packets to be overdue and dropped. The effect of this is an increase in errors caused by packet loss.
- In our experiments, we did not have layered coding. In the case of layered coding, we would have channels with a different quality of service, and therefore different error resilience through different lengths of error correcting codes. In that case, the use of long or short error correcting codes can be adapted to the actual bit error rate. This way, the channel coding would also become suited for use in the area of more moderate bit error rates.
- The link protocol. Most link protocols, like the one in our experiments, use error correcting codes and retransmission of erroneous packets. Therefore, substantial differences in protocol structure and their effects on the experiment results are not expected.

6.7.3 The Compression Algorithm

Here we present a list of the fixed compression algorithm parameters that were used and a prediction of the influence on the experiment results if the parameter would have had a different value.

- The maximum number of requested intracoded macro blocks. Decreasing this number would especially affect the high bit error rate area, when a greater number of macro blocks is corrupted than the number of macro blocks that can be requested to be coded intra. However, limiting this number results also in a limit in bit rate increase, as the intracoded blocks usually require more or many more bits. This can be important for bit-rate-constrained situations.
- The number of frames *A* used to compose the prediction frame. If we would use a different relation between *A* and the bit error rate, for instance with an overall higher value for *A*, this would result in a slightly better error resilience, but also in a worse prediction and therefore a higher bit rate.
- The error detection. Until now we used a limited number of error detection criteria. Increasing the size and effectiveness of the error detection would definitely increase the error resilience. However, how to do this is not straightforward and would also increase the computational complexity and therefore, in the worst case, also have an effect on the real time constraints.
- The quantisation parameter. Performing the experiments with an overall increased quantisation parameter would mostly increase the bit rate, and thereby the number of errors. However, it would also decrease the effect of an error due to the reduced compression ratio, thereby representing a certain part of the image with more bits and decreasing the effect in one of them.
- The coding mode. In our experiments we roughly have three coding modes:
  - Intracoding, for the first frame and for some macro blocks.
  - Intercoding, for low bit error rates and for when the error resilient source coding algorithm is not used.
  - Using the composed prediction frame, for low bit error rates and when using the error resilient source coding algorithm, is used.

  We could for instance force intracoding for the whole test sequence. This would result in more error resilience but also in a substantial increase in bit rate. Using intercoding only would decrease the bit rate but also decrease the error resilience.
- The video compression algorithm. If another algorithm would be used, the outcome of the experiments would depend to a large extent on the principle it is based on. Some approaches are known in literature, for instance for subband coding [51][58][147].

6.8 Conclusion

In this chapter we addressed through an experimental approach the problem of joint error resilient source channel coding. We had to reduce our parameter space by fixing several parameters and taking the resulting bit rate as an output parameter.

Specifically we addressed which error resilience technique to use in which communication situation. The problem is to get a high image quality as well as a low bit rate. We looked at the effect of application of a finer quantisation, of error resilient source coding, of error resilient channel coding and of combinations of these three. We investigated the case of single bit errors at different bit errors rates and of packet loss and burst errors also at different error situations.

We can summarize the results in the following points:

- For the whole spectrum of error situations, the combined error resilient source channel coding technique is the best solution.

- For single bit errors only the best solution depends on the bit error rate. From very low to very high bit error rates we found as best solutions:
  - No error resilience technique and possibly a fine quantisation, but only for extremely low bit error rates.
  - Error resilient source coding.
  - Error resilient channel coding.
  - Combined error resilient source channel coding.
- For packet loss and burst errors error resilient source coding is necessary, possibly combined with channel coding.
- For a simple system channel coding only would already cover most situations in the case of single bit errors

Notice that most of these results originate from the fact that channel coding is an all or nothing method, while the performance of the source coding changes gradually. Most of the conclusions are summarised in Figure 66. Notice that the bit error rates shown apply to the single bit error case.

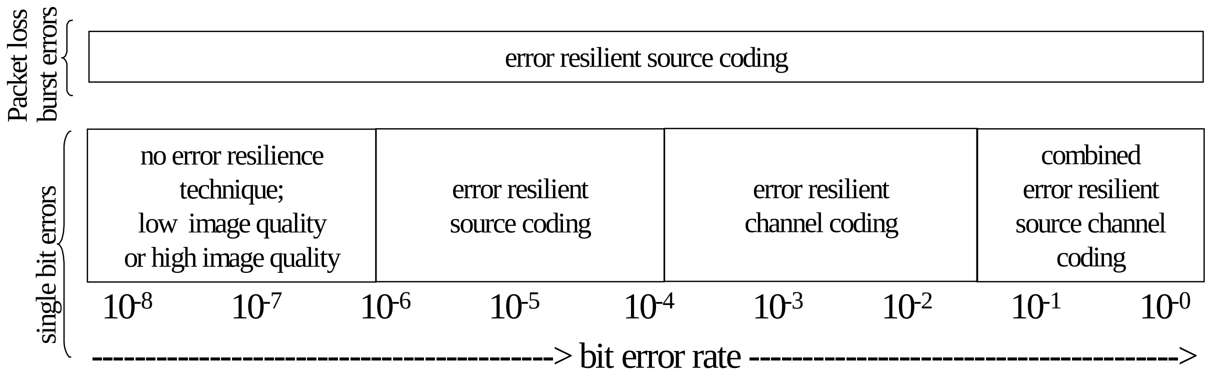


Figure 66 The optimal error resilience technique for different types of errors.

We also addressed the generalisation of these results by looking at the influence of changing the parameters that we fixed during our experiments. We mention the following parameters which probably have the largest influence:

- The content of the video sequence determines the composition of the compressed video stream.
- The frame format and frame rate determine how much visual information a certain number of bits represents.
- Fixed bit rate compression is more realistic than our experiments but als hard to interpret.
- Changing the link protocol parameters would change the performance of the error resilient channel coding and thereby the results shown in Figure 66.
- Optimisation of the source coding could increase the performance of the error resilient source coding and thereby the results shown in Figure 66, although probably less than the previous point.

The source coding algorithm we used up to now, based on H.263, is a common hybrid algorithm, compressing video frames. In recent developments, however, we also see compression of arbitrarily shaped video objects, which is called object based compression. Compression of such objects requires the compression of the texture and the shape of the object. Also for these new techniques error resilience is important. In the next chapter we address the problem of error resilient shape coding.

Chapter Seven

7 Error Resilient Shape Coding[3]

We now leave the frame based video compression techniques and go object based coding. We address specifically compression of the shape of video objects. Again, our aim is to increase the error resilience with respect to existing solutions.

In this chapter we describe our new error resilient and efficient shape coding technique. After an introduction in Section 7.1, we start in Section 7.2 by discussing existing shape coding techniques. In Section 7.3 we discuss the new error resilient shape coding technique. In Section 7.4 we discuss how to deal with transmission errors, and in Section 7.5 experimental results are presented. We discuss implementation aspects in Section 7.6, and conclude with Section 7.7.

[3] This Chapter is based on papers [122], [123], [126] and [127] by Spaan et alteres.

7.1 Introduction

Up to now we have only discussed error resilient video compression techniques for frame based codecs. However, recently techniques have been developed for compressing arbitrarily shaped video objects. This opens up two possibilities: increasing the compression efficiency and increasing the possibilities for manipulating the content of video images. However, this requires that also the shapes of the video objects are described, compressed and transmitted.

In a wireless communication environment we have to anticipate that data will be lost, introducing errors in the reconstruction of the shape, even when channel coding is used. Error resilient compression is then mandatory, also for object shapes. Most current approaches [10] and the development of MPEG-4 [70] have hardly addressed this issue.

In this chapter we propose a new shape compression technique which is error resilient, also when there is no knowledge about when or where errors have occurred in the bit stream. Error resilience can be achieved by minimising the impact of errors and ensuring graceful degradation in case an error becomes noticeable.

Our shape-coding technique combines several approaches to compression and error resilience:

1. A polar co-ordinates system is used for representing the shape. By using this co-ordinate system we induce graceful degradation.
2. Spectral decomposition is applied to the polar functions. After spectral decomposition, the result of an error is a global deformation instead of a local deformation or a missing segment of the reconstructed shape.
3. Intercoding using a reference shape composed of several previous shapes. This new technique temporally smoothes the impact of errors.

The following four techniques we use are more general:

4. Fixed-length coding of the quantisation levels is applied. This improves the resynchronisation capabilities of the decoder.
5. Prioritised transmission is used. This reduces the error impact.
6. Intrarefresh is incorporated in the technique to reduce the error propagation.
7. Error detection and concealment are applied. These also minimise the error propagation.

Besides error resilience, also high efficiency, that is high compression, an adjustable bit rate and a low computational load are desirable. We achieve high efficiency firstly by using polar co-ordinates instead of Cartesian co-ordinates. Secondly we use energy packing by applying a discrete cosine transformation (DCT) to the polar functions. A predefined bit rate or distortion is obtained by means of an iterative/recursive bit rate allocation algorithm on the DCT transformed polar co-ordinates. The computational load of these coding operations is moderate.

The technique starts with the Cartesian $x(i)$ and $y(i)$ shape co-ordinate functions, which are transformed to the domain of polar co-ordinates yielding an $r(i)$ and a $\varphi(i)$ function. Then these functions are transformed by means of a discrete cosine transform, which packs the energy in only few coefficients. Next quantisation is applied and the coefficients of both transformed co-ordinate functions are multiplexed and transmitted. Decoding is the inverse process. A simplified block diagram of the technique is shown in Figure 67.

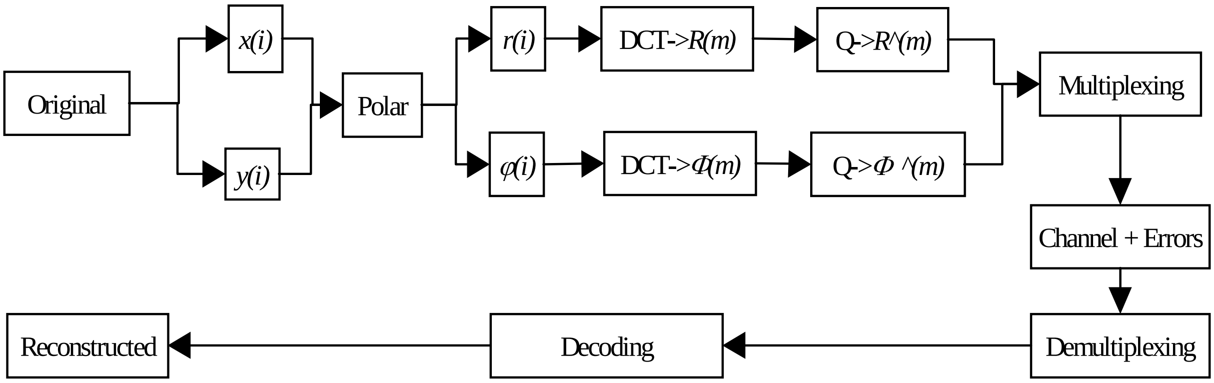


Figure 67. A simplified block scheme of the technique.

Hereafter we first discuss existing techniques and then we discuss the shape encoding technique using intracoding. Intercoding is also applied, but we explain the principles of our technique first on the basis of intracoding for simplicity. Then we discuss how to deal with transmission errors, and experimental results are presented. Finally we address implementation aspects.

7.2 Existing Shape Coding Techniques

There are many ways known in literature to compress shapes, based on different principles. The most commonly used efficient coding techniques are:

Splines [32]. The shape is split into segments, each of which is approximated by a polynomial. Starting points and control points are coded. Although this technique is usually computationally intensive, a substantial improvement on the computation time was obtained by analysing the curvature function of the original shape to obtain a, suboptimal, control point assignment [156].

Chain coding [47]. Each shape point is coded by giving the direction in which to proceed from the previous shape point. This requires 3 bits per shape point, but entropy coding can further reduce the rate.

Modified Modified Read (MMR) [62]. In this run-length-coding technique the binary image is scanned and the position of the pixel where the intensity changes is coded.

Baseline based [77]. The distance to the horizontal co-ordinate axis is coded as well as the positions, the *x*-co-ordinates, where the shape turns.

Vertex based [10]. The shape is described by coding vertices, constructed by taking the largest distance to the previous vertex.

Morphological skeleton [118]. The form enclosed by the shape is described using simple forms that are indicated by their type, centre and size.

There are two more techniques, which have some similarities to the technique we propose in this chapter:

Centroid Based [10]. This technique is, like our technique, based on polar co-ordinates, but it uses a continuously changing polar co-ordinate φ. This does not lead to good results because some parts of the shape, for instance where the shape is parallel to the local radius, are coded less accurately.

Fourier descriptors [102]. This technique seems at first sight somewhat similar to application of a DCT to polar co-ordinates. However, they are different: the Fourier descriptor is applied to the complex position function

(24)
$$z(i) = x(i) + jy(i)$$

while in our case the transform is applied to each of the polar functions.

All these techniques are error sensitive. This is mainly due to the fact that each pixel, or part of representation of the shape, is coded with respect to the previous one. So if there is an error at a certain point, the remainder of the shape is lost. Hierarchical coding, which has been proposed for some of these techniques leads to an increase of the error resilience. However, although some of these techniques are very efficient and fast, none of them emphasises error resilience as much as we think is necessary for wireless communication.

7.3 Polar Co-ordinates

We take a certain shape point with Cartesian co-ordinates *(x,y)* with index *i*, which ranges from 0 through *N*, the number of shape points. The polar co-ordinates of this shape point are the angle $\varphi(i)$ and the radius *r(i)* with respect to a fixed centre point (c_x, c_y):

(25)
$$r(i) = \left((x(i) - c_x)^2 + (y(i) - c_y)^2\right)^{1/2}$$

(26)
$$\varphi(i) = \arctan\left(\frac{(y(i) - c_y)}{(x(i) - c_x)}\right)$$

The origin of the co-ordinate system, (c_x, c_y), usually lies inside the closed shape. The way this centre point is determined for each shape is shown in Section 7.6.

We prefer the description of the shape of objects in polar co-ordinates over that in the usual Cartesian co-ordinates. This is because the polar co-ordinates representation is more appropriate and efficient for the description of two-dimensional closed objects that originate from projection of three-dimensional objects onto the two-dimensional image plane [98]. Although there are other ways in which shapes are created, in video communication most of the shapes will originate from real, three-dimensional objects. Such shapes will in most cases represent the outside of an object and will therefore enclose a circular disc-shaped area without shape points. In the case of polar co-ordinates this leads to a reduced range of values for the *r(i)* function.

Although for object-based coding the shapes need to be closed, this is not required for our technique.

To illustrate the increase in efficiency by means of polar co-ordinates instead of Cartesian co-ordinates, we look at the variance of the co-ordinate functions. The variance of the Cartesian *x(i)* and *y(i)* co-ordinate functions is approximately proportional to the diameter of the shape, denoted by L_1 in Figure 68. The variance of the polar co-ordinate function *r(i)* is approximately proportional to the radius of the shape minus the radius of the enclosed circular area without shape points, denoted by L_2 in Figure 68. Since the allowed reconstruction error is the same in both cases, for instance one pixel, polar co-ordinates require fewer quantisation levels, and therefore are more efficient. A similar argument, though less easily visualised, can also be given for the phase function.

From an error resilience point of view, another advantage is that, because the basic form of polar co-ordinates is circular, coarse quantisation or transmission errors will introduce overall rounded artefacts. Furthermore, the shape as a whole will degrade towards a circle. Both features give visually more acceptable results than the arbitrarily shaped local artefacts resulting from most other techniques.

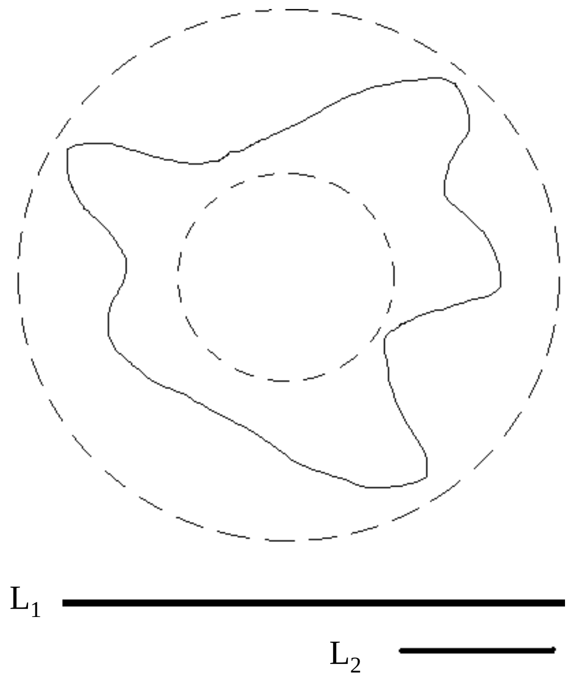


Figure 68. An illustration of the increase in efficiency when polar co-ordinates are used instead of Cartesian co-ordinates.

7.3.1 Shape Decomposition

An example of a shape and its *r(i)* function are given in Figure 69 and Figure 70, respectively. Figure 69 shows a shape of image 45 of the Hall Monitor sequence: the original (grey) and reconstructed shape (black, overlay, without post-processing) are shown. Figure 70 represents the *r(i)* co-ordinate function corresponding to the shape shown in Figure 69.

Figure 69. Shape number 45 of the Hall Monitor sequence in CIF format. The original shape is shown in grey and the reconstructed compressed shape is shown in overlay in black, without post-processing.

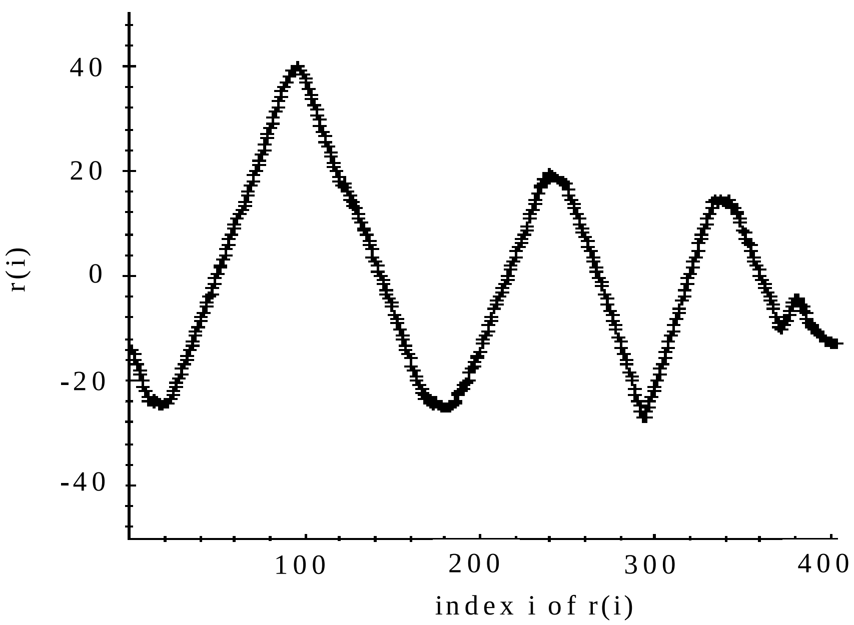


Figure 70 The r(i) co-ordinate function corresponding to the shape shown in Figure 69.

Before the spectral decomposition we have to apply pre-processing, which consists of the following steps. First, for *r(i)* the mean value is subtracted. Then we apply to $\varphi(i)$ phase unwrapping and finally remove the linear phase component to increase the compression efficiency:

(27) $$\varphi(i) = \varphi(i) - \left(\frac{i2\pi}{N}\right)$$

After pre-processing, the $\varphi(i)$ function is smoother and has a smaller variance, except for unusual shapes. Although the values of the pre-processed *r(i)* and $\varphi(i)$ are different from the original *r(i)* and $\varphi(i)$, we use the same notation for simplicity.

We now spectrally decompose the polar co-ordinate functions. We do this for the following reasons:

- We prefer a globally degrading effect of transmission errors, which is less destructive than local degradations. The result of the spectral decomposition is a global description of the shape, which results in global degradation.
- The error resilience is increased by means of prioritised transmission, which is enabled by the spectral decomposition.
- The efficiency is increased because the spectral decomposition packs the energy in just a few coefficients after quantisation.

For the spectral decomposition we use the discrete cosine transform of a length equal to the number of shape points. After transformation of the *r(i)* and $\varphi(i)$ functions, we have obtained the *R(m)* and $\Phi(m)$ functions, respectively. As an example, the first 100 DCT coefficients of the *R(m)* function of the shape in Figure 70 are shown in Figure 71. We can already see that in order to obtain a well-approximated reconstructed shape at the decoder, not many of the coefficients need to be transmitted.

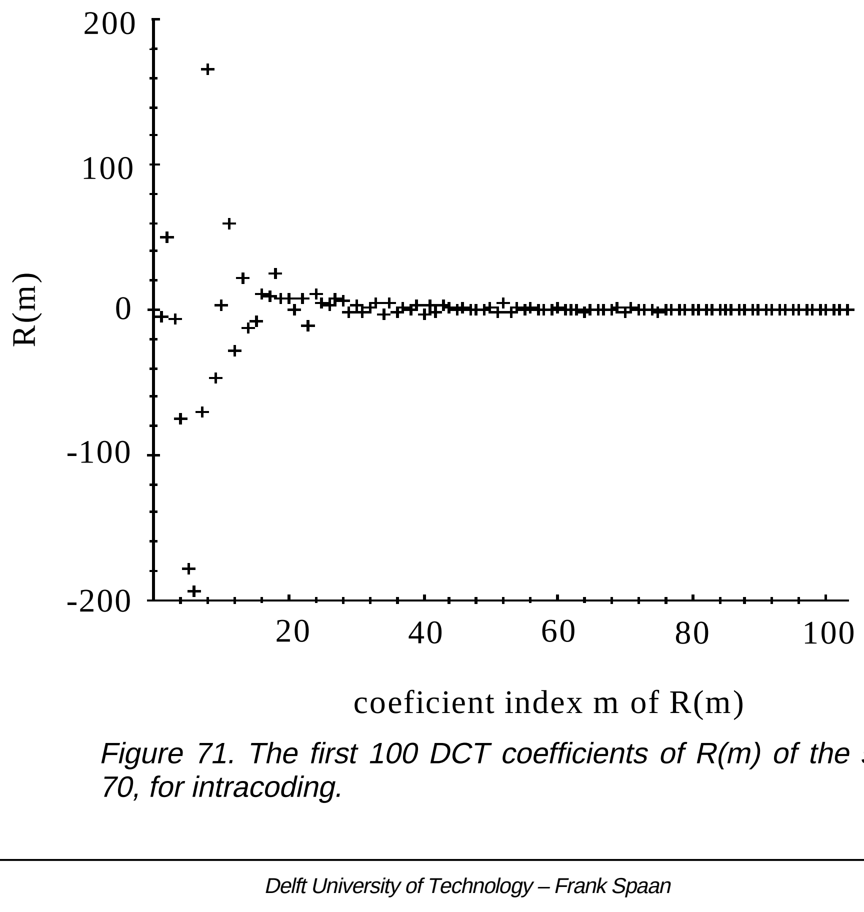


Figure 71. The first 100 DCT coefficients of R(m) of the shape in Figure 70, for intracoding.

7.3.2 The Reconstruction Error

After transformation, the DCT coefficients have to be quantised and the available bits have to be allocated. To this end, we need a bit allocation criterion function. This function reflects how quantisation errors in the DCT coefficients influence the error in the spatial reconstruction. We define the error in the spatial shape reconstruction as the sum of the geometrical distances between the original and reconstructed shape points:

(28) $$E^2 = (\Delta x)^2 + (\Delta y)^2$$

where

(29) $$\Delta x = x_{reconstructed} - x_{original}$$

and similarly for *Δy.* The squared geometrical error can now be rewritten as

(30) $$E^2 = (\Delta r)^2 + 4(r\Delta r + r^2)\sin^2\left(\frac{\Delta\varphi}{2}\right)$$

To arrive at this result we used the following relations:

(31) $$\Delta x = \Delta(r\cos\varphi) = \Delta r(\cos\varphi) + r\Delta\cos\varphi = (\cos\varphi)\Delta r - r(\sin\varphi)\Delta r$$

(32) $$\Delta y = \Delta(r\sin\varphi) = \Delta r(\sin\varphi) + r\Delta\sin\varphi = (\sin\varphi)\Delta r + r(\cos\varphi)\Delta r$$

Since the reconstruction errors will be small, we can use a second-order Taylor expansion as an approximation for the sinus function in (30):

(33) $$\sin\varphi \cong \varphi \qquad \text{for } \varphi \ll 1$$

We can then approximate (30) by

(34) $$E^2 = (\Delta r)^2 + (\Delta(r\varphi))^2 \qquad \text{for } \Delta\varphi^2 \ll 1 \text{ and } \Delta r \ll r$$

The conditions for which (34) is valid are normally met, unless extremely coarse quantisation is used for *r* and φ.

In a rate-distortion optimal approach *R(m)* and *Φ(m)* are quantised in such a way that the error is minimal at a given bit rate. As the DCT is a unitary transform, minimising the sum of the squared errors in *R(m)* and *Φ(m)* gives the same solution as minimising the sum of the squared errors in *r(i)* and *φ(i)* [107], so

(35) $$E^2 = (\Delta r)^2 + (\Delta\varphi)^2 = (\Delta R)^2 + (\Delta\Phi)^2$$

However, equations (30) and (35) show that the squared geometrical error is different from the above squared error. We solve this problem by using *rφ(i)* (*r* times *φ*) as a variable instead of *φ(i)*, and we use equation (34) for the error determination. The DCT will then be applied to *r(i)*, yielding *R(m)*, and to *rφ(i)*, yielding *RΦ(m).* Figure 72 shows geometrically that *r(i)* and *rφ(i)* are locally perpendicular and will have a correlation close to zero. They will also have about equal ranges of values, roughly equal to the distance in pixels.

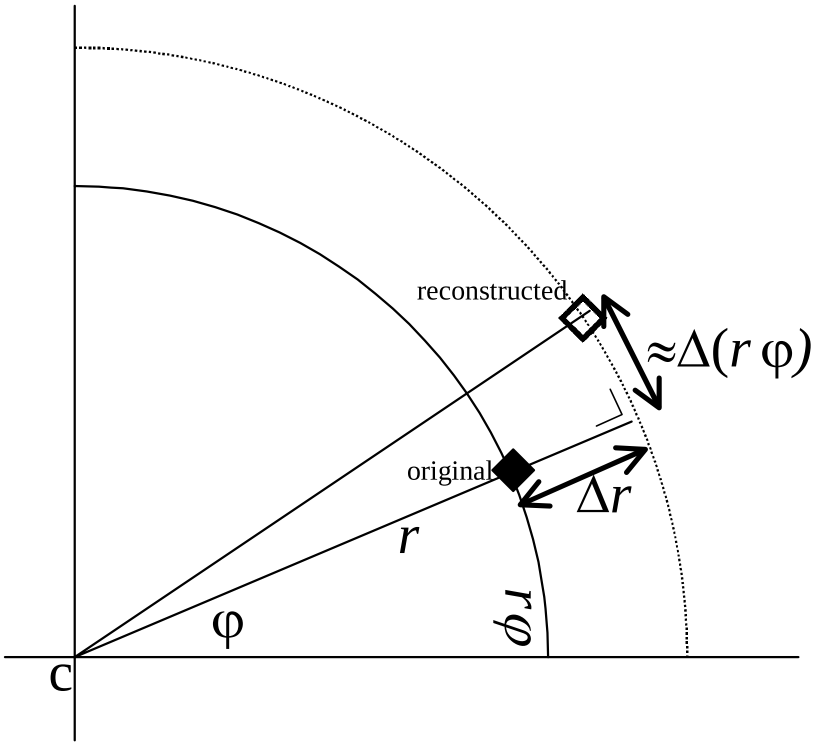


Figure 72. An illustration of the reconstruction error when using (r,rφ) co-ordinates.

7.3.3 Quantisation and Bit Allocation

We apply to the transformed co-ordinate functions *R(m)* and *RΦ(m)* a uniform quantiser with step size *S*, which determines the coarseness of the quantisation. The *R(m)* and *RΦ(m)* are quantised separately. By controlling *S* for each function, we can change the distortion and bit rate. We can reduce the number of bits further by assigning a maximum number of bits to each index *m* which decreases with increasing *m*. This reduces the bit rate because, although the quantisation step size *S* is constant for all coefficients, the maximum allowed quantified value decreases with increasing DCT index.

From information and rate distortion theory we expect an exponential dependency of the variance of each coefficient on the index. Therefore the maximum number of bits $d_{max}(m)$ for each *R(m)* or *RΦ(m)* DCT coefficient is made exponentially dependent on their standard deviation. This yields for *R(m)*

$$d_{\mathrm{R,max}}(m) = {}^{2}\log\left(\sqrt{\mathrm{var}(R(m))}\right) \tag{36}$$

A similar result can be obtained for *RΦ(m)*. The number of bits assigned to *R(m)* can be different from that assigned to *RΦ(m)*. To get an estimate of the standard deviation, we use the following model for the standard deviation of *R(m)*:

$$\sqrt{\mathrm{var}(R(m))} = a_R e^{-b_R m} \tag{37}$$

and similarly for *RΦ(m)* with parameters $a_{R\Phi}$ and $b_{R\Phi}$. This model reflects that, with increasing *m*, the variance of the DCT coefficients decreases, so that by (36) fewer bits are assigned. Furthermore, the model (37) is specified by two parameters, which need to be adapted for a given shape. Figure 73 shows the average behaviour of the variance of *R(m)* for the test sequence 'Hall Monitor' for intracoding.

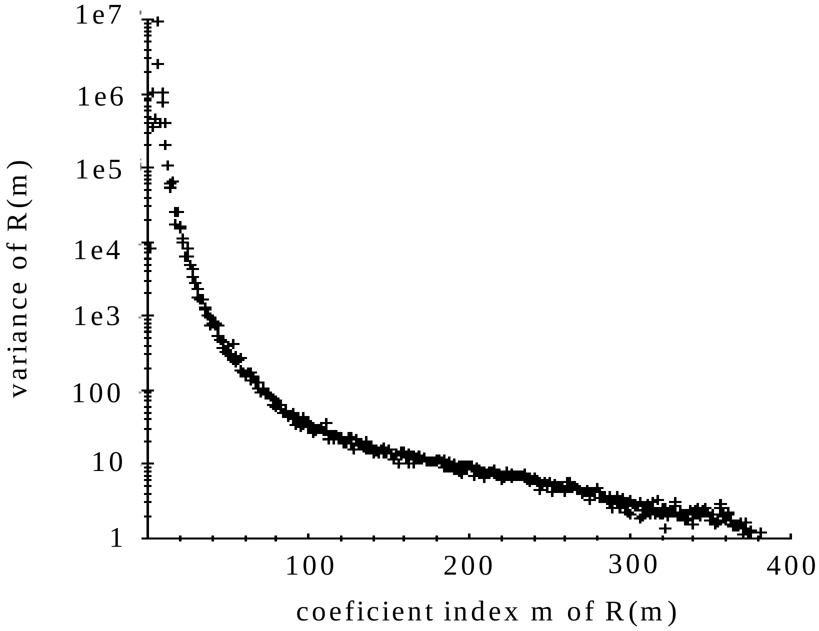


Figure 73. The average of the variance of R(m) for the test sequence 'Hall Monitor', for intracoding.

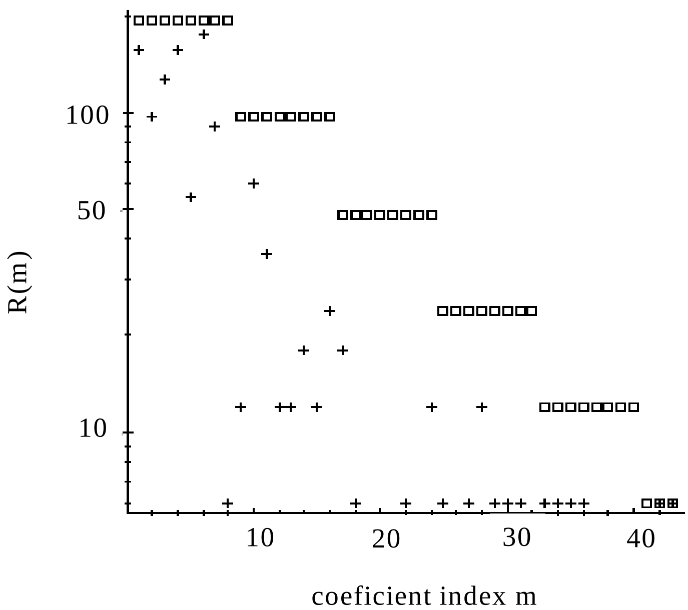


Figure 74. A logarithmic plot of the first 50 values of R(m), denoted as (+), and $d_{R,max}(m)$, denoted as squares.

For each shape, for both *R(m)* and *RΦ(m)* and for each value of the quantisation step size *S*, we have to determine the parameters *a* and *b* in (37). We take *a* as the largest value of the coefficients of the transformed co-ordinate function, and for *b* we try a limited number of values. One of the results is shown in Figure 74, where a logarithmic plot is shown of the first 50 values of *R(m)*, denoted by (+), and $d_{R,max}(m)$, denoted as squares.

To determine the optimal value for the quantiser step size *S* and the parameter *b*, we can use an exhaustive search. However, a more elegant approach is to use the convex hull property of the rate distortion relation. One of the obtained rate distortion plots is shown in Figure 75, which demonstrates this convex hull property. Shown are the obtained rate-distortion values using a full search of possible *S* and *b* values. We start either at a very low rate or at a very low distortion and use a greedy algorithm to obtain the approximately best solution close to the maximum rate or maximum distortion [114][151].

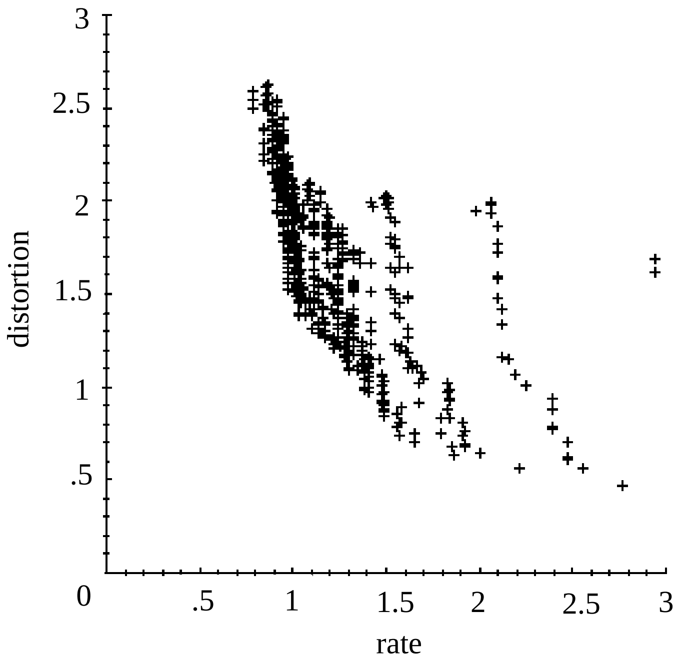


Figure 75. An example of an obtained rate-distortion plot. The horizontal axis represents the bit rate per shape point and the vertical axis the distortion value in pixel distances.

The found parameters *S* and *b* are then transmitted to the decoder with each shape and for each of the co-ordinate functions. The quantisation levels are transmitted using fixed-length codes for error resilience. These fixed-length codes representing the quantisation levels of the DCT coefficients of the co-ordinate functions are multiplexed and concatenated. Many of the levels are zero, and if after index *m* all are zero, only the levels up to index *m* will be transmitted.

7.3.4 Intercoding

The shape coding method described so far operated on shapes in individual frames. Since no reference is made to other coded shapes, it can be characterised as an intracoding technique. The proposed method does, however, also allow for coding of shapes with reference to other already coded shapes, which can be characterised as an intercoding technique. Instead of transmitting the DCT coefficients of the current shape, we take the difference of each DCT coefficient with the DCT coefficient with the same index of a predicted shape. Normally, the predicted shape would be the corresponding shape from the previous image. However, if there is an error in the previously decoded shape, this error will propagate through all the following shapes, with the same energy, that is the same error, as in the shape in which it first occurred. Therefore, it is important to reduce the impact of the error.

To realise this, we encode a current shape, not with respect to the previous shape, but with respect to a composed prediction shape. This shape is composed, coefficient-wise, of *z* previous shapes; each weighed with a weigh factor w_k:

$$R_{reference}(m) = \sum_{k=1}^{k=z} \left(w_k R_k(m)\right) \Big/ \sum_{k=1}^{k=z} \left(w_k\right) \qquad (38)$$

The number of shapes used and their weigh factors can be different for each shape. Although this technique reduces the impact of the error, it also makes the prediction worse. Therefore, the weigh factors are made dependent on the bit error rate and also decrease with decreasing frame number. The latter means that for instance the shape of frame N – 3 is weighed heavier than the shape of frame N - 4. The dependency on the bit error rate is important because, if there are no errors, only the previous shape is taken into account to compose the reference shape. If there is a high bit error rate, the average of for instance the previous 5 shapes is taken.

The actual encoding of the difference between the current shape and the composed prediction shape is identical to the coding of $R(m)$ and $R\Phi(m)$ in the intracoding case. The main difference is that for the same reconstruction error E^2 using equation (34), a coarser quantisation can be used, yielding a smaller bit rate.

A further increase of the error resilience is achieved by intracoding the shapes with a certain temporal refresh rate. This increases the bit-rate, but will also reduce error propagation.

In the remainder of this chapter we refer to the technique described above, which uses a composed prediction frame as the prediction, as the error resilient technique and use the term intercoding for coding with respect to the previous frame only.

7.4 Transmission Errors

Due to wireless transmission, delay limits and network congestion, there will be errors in the received encoded bit stream. In this section we evaluate the impact of these errors on the error resilient shape coding technique. For the experiments we used random errors induced by flipping bits in the original bit stream.

Before we present our calculations, we first describe the way we find a relation between the bit error rate and the relative number of shapes that will be lost because of a large reconstruction error. How large this reconstruction error is, depends on the impact of the transmission error, which in turn depends on where in the coded shape the error occurs. If the error occurs in the header, the shape is considered lost. If the error occurs in one of the coded coefficients, the impact depends on the position of the error in the bit stream, because on the average the variance of the coefficient decreases with the index according to equation (37).

In other words, if we know the index of the corrupted coefficient, we can estimate the reconstruction error. The value of this index depends on the bit error rate and on the number of bits that are untouched before the error occurs at such a bit error rate. Given such a number of untouched bits, the index to which this corresponds depends on the number of header bits and the number of bits per coefficient, which follows equation (36). So, given a bit error rate, we can estimate the index of the coefficient that is corrupted and thereby the reconstruction error. Vice versa, we can set a maximum allowed reconstruction error and find the bit error rate up to which we obtain, on average, acceptable reconstructed shapes.

We first calculate the location in the bit stream of the coded shape of the first error we can still allow. We look at a part of the bit stream representing one average shape, as shown in Figure 76. The average number of header bits we denote as *H*. The number of coefficients used to represent the shape is N_u, the total number of shape points we denote as *N.* If there is an error somewhere in the coefficients, the index of this corrupted coefficient is denoted by m_c.

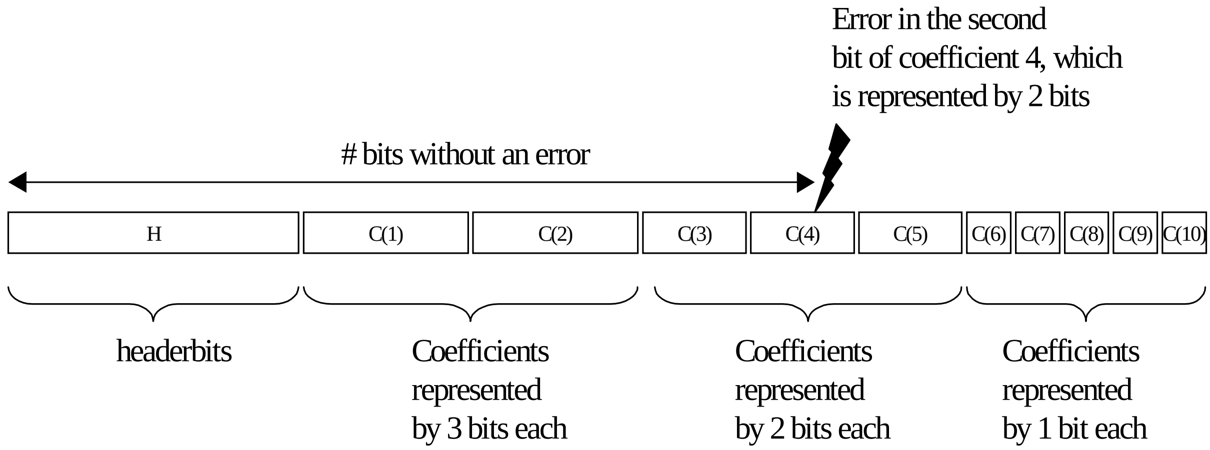


Figure 76 A simplified example of the structure of a compressed shape. The multiplexing of the R(m) and RΦ(m) coefficients is not shown.

The effect of an error in one of the DCT coefficients depends on its original and degraded or erroneous value. Both are limited by the parameters of the exponential function which determines the number of bits used to represent the coefficient. If this number, including the sign bit, is *d*, and the error consists of flipping one of the *d* bits, the energy of the such an error will be on the average E_b

(39)
$$E_b = S\left(\sum_{j=0}^{d-2}\frac{1}{d}2^j + \frac{1}{d}\,\frac{\sum_{j=0}^{2^{d-1}-1} j}{2^{d-1}}\right)$$

Here *S* denotes the quantisation step size. The first part of this equation represents the probability of an error in one of the bits, excluding the sign bit, multiplied by its value caused by the bit flip. The second part of this equation represents the probability of an error in the sign bit times the average of the value represented by the other bits. The number of used bits *d* we know from the function used by the quantiser (36). Therefore, we can find an expression for the actual error by substituting (36) in *(39*). This error will cause a reconstruction error. Because the

DCT is an energy-preserving transformation, we assume that the energy of the error is equal to the summed squared reconstruction error:

(40) $$E_b^2 = \sum_{j=1}^{j=N} E_a^2 = NE_a^2$$

We can solve the system (34), (35), (37), *(39*) and (40) numerically for m_c.

The probability that no errors occur before the m_c^{th} coefficient at a certain bit error rate R_{be} depends, among other things, on how many bits on average there are in the shape representation up to the m_c^{th} coefficient. This in turn depends on the number of header bits and the number of bits per coefficient. The latter depends on the average exponential function used for the bit reduction in the quantiser. The values for *a* and *b* are experimentally determined and the values for *R(m)* and *RΦ(m)* are taken equal here. Taking into account the multiplexed transmission of the data, the number of bits *B* up to m_c is equal to:

(41) $$\begin{aligned} B &= H + 2\sum_{m=0}^{m_c} {}^2\log\left(ae^{-bm}\right) = \\ &H + \left\{2\ln(a)(m_c+1) - b(m_c+1)^2 + b(m_c+1)\right\}/\ln(2) \end{aligned}$$

Now we calculate the probability that the first error occurs before m_c. The probability P_r of any number of random errors to be present anywhere before m_c at a bit error rate of R_{be} is equal to one minus the probability of no errors:

(42) $$P_r = 1 - \left(1 - R_{be}\right)^B$$

Alternatively, if we only allow a fraction P_r of the shapes to be affected worse than the allowed additional error, the expression for the allowed bit error rate R_{be} becomes:

(43) $$R_{be} = 1 - \sqrt[B]{\left(1 - P_r\right)}$$

where *B* is given in (41). We will evaluate these analytical results experimentally in Section 7.5.

7.5 Experimental Results

For the experimental results presented hereafter the segmented Hall Monitor, Dancer and Akiyo sequences in CIF format were used. Of these sequences, the Hall Monitor is the most demanding to compress because of the motion and the relatively unusual shapes, and Akiyo the least demanding having little motion and relatively smooth shapes.

We coded the sequences and randomly introduced errors by flipping a bit using a random number generator. The error resilience of the decoder was increased by resynchronisation, that is searching for the next shape starting code whenever an error occurs. Some error detection was also applied which enabled error concealment.

7.5.1 Compression Efficiency and General Results

We first look at the obtained bit rates using the error resilient technique, without introducing errors. We coded Akiyo, which has large shapes and little motion. The resulting compression is very efficient as expected: a mean value of 0.29 bits per shape point at a mean euclidian reconstruction error of 0.57. This is shown in Figure 77.

In the same figure some bit rate peaks occur about every 30 shapes. These occur whenever a shape is forced intracoded to prevent error propagation in the event of transmission errors. Without this, the mean bit rate would be even lower. The data shown were obtained using a composed prediction shape composed of 5 previous shapes. For Shadow this was somewhat worse, 0.95 bits per shape point, although large parts of the sequence were coded with about 0.6 bits per shape point. For the Hall Monitor sequence an average result of 1.3 bits per shape point was obtained with an average mean Euclidean reconstruction error of 1.1 pixel distance. The higher average is due to the implemented inefficient way of coding very small shapes.

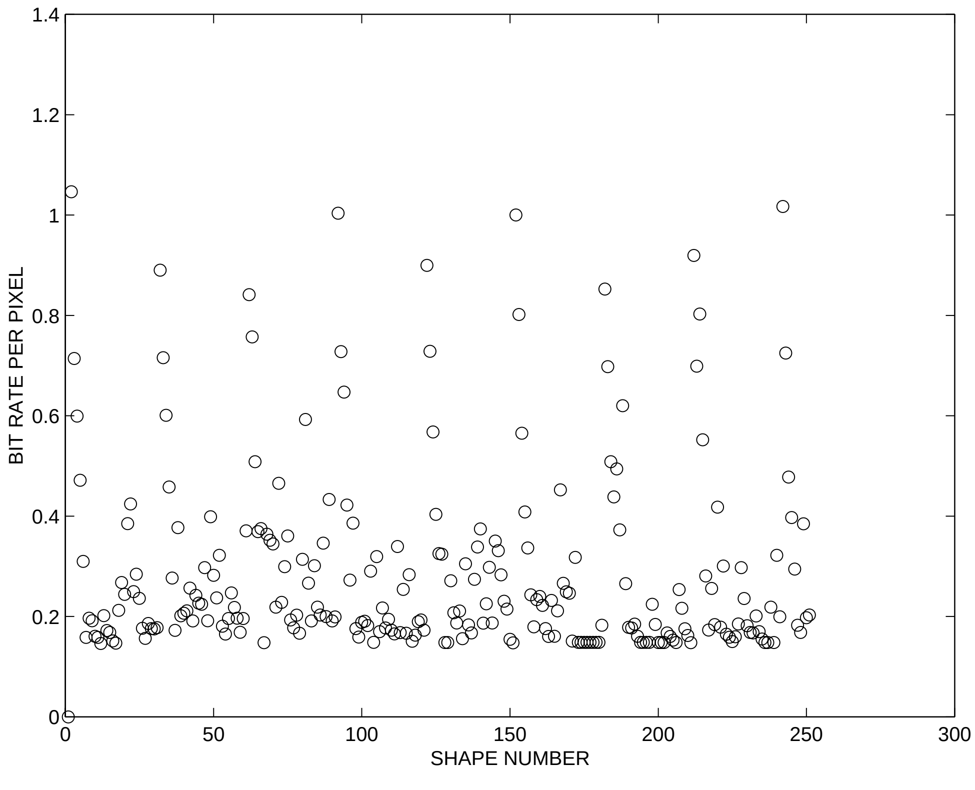


Figure 77. The bit rate per shape point as a function of the shape number obtained using the Akiyo sequence.

Using intercoding instead of intracoding gives in general an efficiency improvement by a factor of about 2, depending on the quality of the prediction. This is illustrated by the low average bit rate in Figure 77 and the peaks caused by the higher bit rate for the regularly intracoded shapes in the same figure.

At very course quantisation, as an example for either very low bit rate or high error rate, the deformation of the shape is smooth and non-local, as shown for the shape in frame 45 in Figure 69.

7.5.2 Error Resilient Intracoding

We first address the experimental results of the intracoded Hall Monitor sequence. We can compare these with our analytical results, which are valid for intracoding only. We use the following values: the average number of header bits H = 124, the parameters of the exponential distribution a = 126 and b = .108 , the average number of shape points N = 231, the quantisation step size S = 5. In Figure 78 are shown the analytically obtained results denoted by a + symbol. We used a bit error rate of 10^{-4}. We plot the number of lost shapes as a function of the allowed additional error. Also shown are the experimental results denoted by an o symbol. We see a discrepancy for higher values of the additional error, which is possibly mainly due to the fact that we used average values for *S, N*, *a* and *b* and not their distributions. For large values of *N,* for instance, the value of the DCT coefficients can be higher and therefore, in the event of an error, large additional reconstruction errors can occur, which are not present in the presented analytically obtained values.

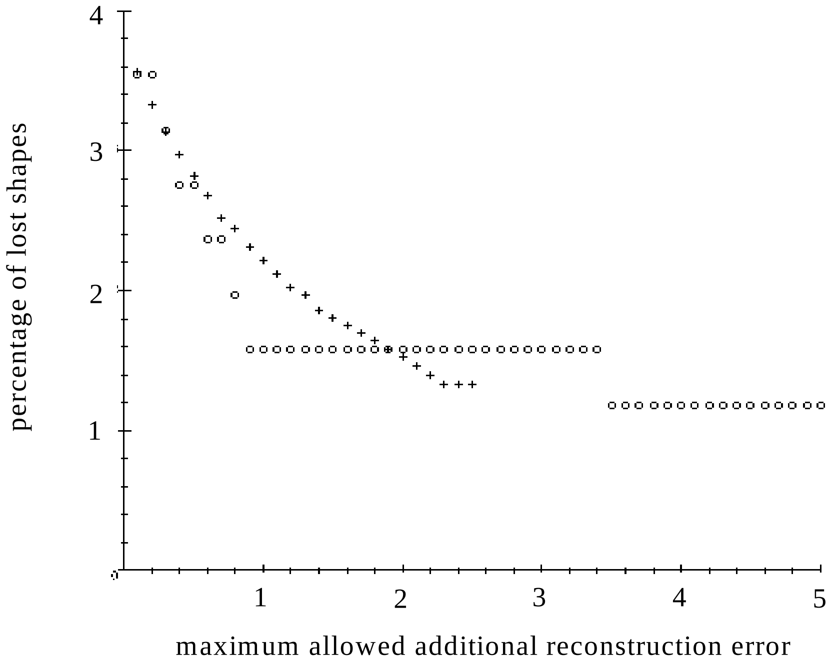


Figure 78. The percentage of lost shapes as a function of the maximum allowed additional reconstruction error. Shown are analytical (+) and experimental (o) results for a bit error rate of 10^{-4}.

Now we compare this with standard techniques. At a bit error rate of 10^{-4} we have introduced 23 errors and about an equal number of shapes would have been lost if standard techniques had been used. The error resilient technique, using intracoding only, produces 5 lost shapes at an allowed additional error of 1 pixel distance, which is an improvement by a factor of 4.6. Note that some of the error resilient techniques, namely using the composed prediction shape, error detection and concealment, and intrarefresh, were not used when these values were obtained, so this can be improved.

Besides experiments using random bit error patterns, some limited experiments were carried out using a simulator for the transmission channel and link protocol, which gives more realistic bit error patterns [27][28]. Such patterns have a more bursty character, and the error resilience results were according to this. Most of the time good quality shapes were obtained while many shapes are lost when there is a burst.

7.5.3 Error Resilient Intercoding

Now we address the question whether coding using the error resilient technique, that is by means of a composed prediction shape, gives more error resilient results than just using the previous shape for the prediction, that is intercoding. The results for both cases using the Hall Monitor test sequence at a bit error rate of 10^{-3} are shown in Figure 79. Shown is the number of lost shapes as a function of the maximum allowed additional reconstruction error. On average the error resilient technique gives better results by a factor of 1.6.

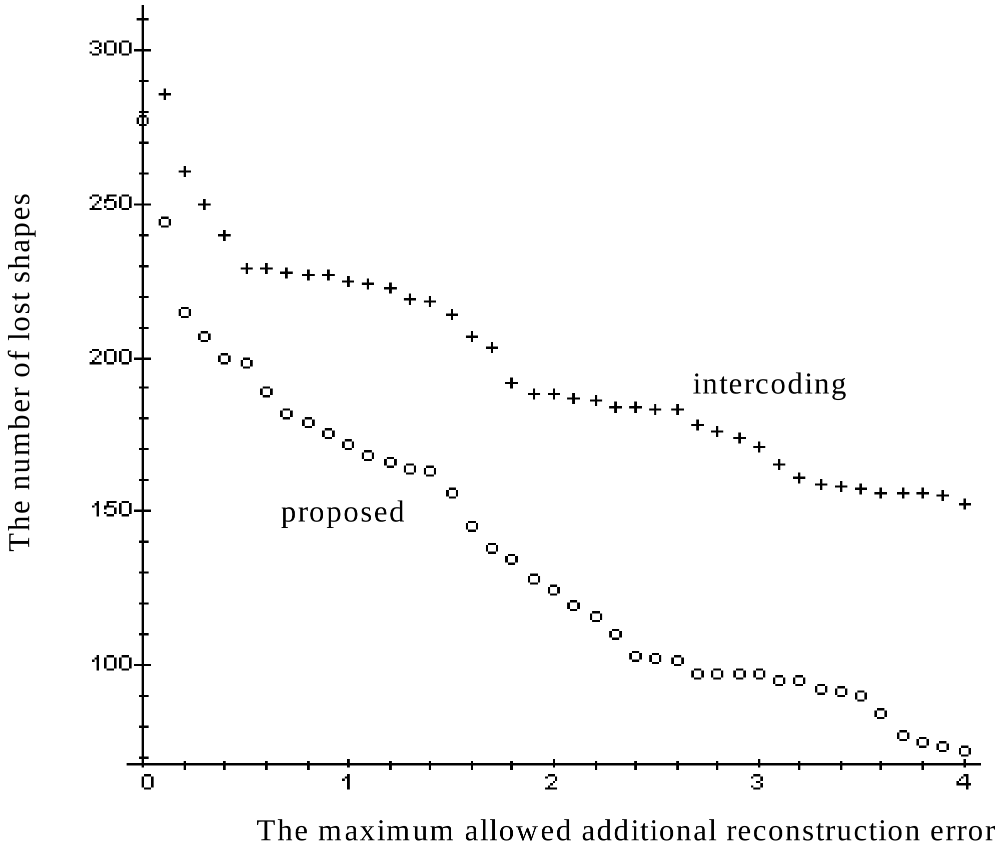


Figure 79. The number of lost shapes as a function of the maximum allowed additional reconstruction error, in pixel distances. Shown are the results for intercoding (+) and using the error resilient technique (o) at a bit error rate of 10^{-3}. The total number of coded shapes was 468.

Experiments on the impact of the propagation of the error using intercoding, so without using the composed prediction shape, show that this is constant. Once there is an error, it will not increase or decrease substantially over the shapes in the next images. This low propagation of the error is due to the fact that only the absolute error is transferred to the next shape, which means that if the effect of the first occurrence of the error is small, it will stay small in the next images. However, it will not disappear until a shape is intracoded again; in the meantime new errors that occur will be added to the existing error. This is overcome by intracoding shapes with a certain frequency.

7.5.4 Bit Rate Comparison

We now compare the bit rates for the different techniques. For the Hall Monitor sequence the bit rate for the intracoding, intercoding and the error resilient technique is 1.7, 1.3 and 1.3 bits per shape point, respectively. The relatively high rates are partially due to the number of very small shapes in the sequence, which cannot be coded with very few bits in the current implementation. The small difference between intercoding and the error resilient technique indicates that, at least in this example, the error resilient technique is as efficient as intercoding. Lower absolute values for the bit rate were already shown in Section 7.5.1.

7.6 Implementation Aspects

We have implemented the error resilient shape coding technique we described in the previous sections. For this implementation the following aspects were important:

- As input, we assume to have a closed, eight connective shape from a frame of a video sequence. The centre of the shape, which we need for transformation to polar co-ordinates, is the average of the x and y co-ordinates of the shape. Having tried other ways of determining the shape, like maximising the minimal distance to the shape, we have found the averaging of the co-ordinates a simple but fast and robust technique giving constant results, which is important for our technique.
- For both intercoding and the error resilient technique, it is essential that the current shape is linked to the corresponding shape of the previous frame, or to shapes from multiple previous frames when using a composed prediction shape. To accomplish this, the position of the centre of the current shape is matched with the centres of the stored shapes of the previous image or images. If there is no match at all, the shape is taken to be new and will be intracoded. The thresholds for the matching were determined experimentally.
- We have to find a starting point for the co-ordinate functions that is such that the starting points of consecutive shapes do not differ very much. We take the right most point having the same y co-ordinate as the centre. Furthermore we make sure that we run the shape in the same direction as the previous one.
- It is necessary, as a post-processing step, to make the shape connective again because both co-ordinates, $r(i)$ and $\varphi(i)$, are treated separately and the resulting point is not necessarily attached to the previous shape point. This is especially the case when coarse quantisation is applied and when transmission errors occur.
- Finally, we present in Table 4 a summary of the information that is contained in the header of the compressed shape.

Table 4 The information in the header of the compressed shape.

	Remarks	**Number of used bits (for example)**
Start code	15 x 1 + 1 x 0	16
Image number	Modulo 256	8
Shape number	Maximum of 8 shapes / image	3
Number of shape points	Maximum 1024	10
Number of used coefficients per co-ordinate function	Maximum 1024 each	20
Number of bits per coefficient per co-ordinate function	32 levels each	8
Quantisation parameter S per co-ordinate function	Maximum 1024 each	14
Quantisation parameter b per co-ordinate function	Maximum 512 each	10
Number of weigh factors	Maximum 16	3
Centre co-ordinates	Maximum 512 each	18
r-mean	Maximum 1024	9
Sign of linear phase		1
Intra / inter flag		1
Total		122

7.7 Conclusion

In this chapter we described a new shape coding technique. We first looked into existing shape coding techniques, which show little error resilience. The main reason for developing a new technique was to improve this. Several different error resilience techniques are now incorporated in our technique, of which polar coordinates, the disrete cosine transform and the composed prediction shape are the main new elements. We also descibed the determination of the reconstruction error and the bit allocation. We have been able to describe the improved error resilience analytically in terms of the number of lost shapes as a function of the allowed reconstruction error.

We have verified the results experimentally, which showed that the correspondence between the analytical and experimental results is not optimal yet. This could be a topic for further research, although the difference is not very large.

We have also fully implemented the developed shape coding technique and found that it performs well in terms of computational load, bit rate, and error resilience. The compression of very small shapes is in this implementation inefficient, however.

.

8 DISCUSSION

One of the first things we showed in this thesis is that mobile visual communication has characteristics that can hardly be replaced by other ways of communication. Furthermore, an application area was indicated: complex or unexpected problem solving in a professional environment. However, it is still not clear when or whether mobile visual communication will ever be as successful as for instance mobile telephony is today. For the moment the necessary financial investment is high and the potential utilisers are still sceptical.

Interdisciplinary research brought us results that could hardly have been obtained otherwise: for instance, showing the importance of the user interaction and the limits of the applicability of the Open Systems Interconnect layer model, and producing the results on joint source channel coding. On the other hand, it is also clear that for many other results the interdisciplinary part of the research did not contribute decisively.

We introduced a quantitative concept for a system if it were to be in operation five years from now, and showed the viability of such an integrated mobile multi-media communication system. Defining the concept, however, we needed to assume certain conditions, and if these turn out to be different this could make the system less probable. The most important assumed parameters under discussion are the transmission conditions, the availability of the capacity of the back bone, the high carrier frequency, the synchronisation, the low delay, the environment and the error- and traffic-statistics. Nevertheless, integrating results of different parts of the project is a usefull effort because it is a possible substitute for producing a complete system with technology that does not exist yet and because it shows the weak and strong points of the projects results.

We see error resilience as the main problem for video compression in mobile multi-media communication. For most error resilience techniques error detection is crucial. However, error detection is difficult and not one hundred percent successful and therefore a technique should be used or added that is independent of error detection. To this end we analysed a DPCM system with respect to transmission errors. We found that the optimal choice of the coefficients depends on the bit error rate: the number of used coefficients should increase with increasing bit error rate. We used this result to increase the error resilience of an H.263 based codec by using a composed reference frame. Furthermore we found that the value of the DPCM coefficients should decrease with increasing bit error rate. We have not been able to further investigate this. However, some first results show that this effect might turn out to be more important than choosing the correct number of coefficients. Therefore this is an important topic for further research.

Using the composed reference frame does not increase the error resilience of the H.263 codec sufficiently. Therefore, we still need additional techniques and these use error detection. We found that error detection in the image right before display is effective in trying to maintain the visual quality, which we defined as our optimisation criterion. The error detection technique we developed and combined with the error concealment and adaptive request for intracoding of macroblocks showed good results in trying to maintain the image quality at the same bit rate.

We compared this error resilient video compression technique with the error resilient protocol developed in the Mobile Multi-media Communication project to find out which of these techniques should be used as a function of the bit error rate. To this end we performed experiments and from the results we derived a guide for which technique to use in what circumstances. The optimisation parameters were the bit rate and the image quality. This guide indicates that for single bit errors the choice of the optimal technique depends on the bit error rate. For packet loss type of errors, the error resilient compression algorithm is superior, and for more simple applications the error resilient network protocol is a good choice. We also see that trying to increase the image quality by using a finer quantisation in the presence of transmission errors actually decreases the quality. Furthermore, using an error resilience technique is most effective in the 10^{-4} through 10^{-2} bit error rate range. This is altogether an important result, especially for applications, and we mark this topic as one that should be addressed and expanded in future research. We obtained these results experimentally and they are therefore hard to generalise. However, the method for deriving the guide is one that can be easily applied to other systems.

The shape coding techniques that can be found in literature are not or hardly error resilient. We developed an error resilient as well as efficient technique. The error resilience and its limits were shown analytically as well experimentally. Obtaining a predefined bit rate or distortion is also possible. The current implementation is also efficient, except for small contours, and optimisation could well lead to a technique that outperforms current techniques.

In the end, a new compression technique is only useful if it becomes incorporated in a standard. It is therefore important that researchers and current and future standardisation committees know of this and other work on video compression and their results. Although the current MPEG effort is in the direction of the description of the content of visual information, we see error resilience in video communication still as a subject of attention for future standards. We hope the results of our research, described in this thesis, may contribute to this.

APPENDIX A SUMMARY OF THE INTEGRATED SYSTEM PARAMETERS

In this Appendix we summarise the parameters of the concept of the integrated application system, to be operational five years from now:

The application area:
- Professional mobile users.
- Multi-media.
- Increase of efficiency or quality of outcome.
- Unexpected or complex situations.
- Remote expertise.

The user-interface:
- Modular and application specific.
- High quality.
- Example:
    - Head mounted display.
    - Wearable mouse and keyboard.
    - Mounted or hand-held camera.
    - 'Patch' antennas.
    - End to end delay smaller than 0.1 seconds.
    - Less advanced user interface when possible.

Video compression:
- 1000 x 1500 pixels and 50 frames per second.
- Low bit error rate, smaller than 10^{-3}, error bursts.
- Error resilient compression, compression factor about 20.
- Trade-off with channel coding.
- Control of media system parameters.
- Different traffic classes.

The link and network protocol:
- Base stations, fixed or 'mobile'.
- Up to 4 users per cell, up to 540 kilometres per hour.
- 155 megabits per user, 1 gigabit per second per cell raw.
- Backbone connection.
- Bit stream is compliant with low cost terminals.
- User demanded qualities of service.
- Buffering, forward error correcting codes and retransmissions, delay up to 5 milliseconds.

Transmission:
- Data conversion serial to parallel, 128 point fast Fourier transform.
- Orthogonal frequency division multiplexing.
- 155 megabits per second, using 100 megahertz.
- Low cost terminals possible, 15 megabits per second.
- Power between 0.01 and 0.1 Watt.
- 60 gigahertz carrier, 150 meters cell radius.
- Rice fading, additive white Gaussian noise errors.
- Bit error rate between 10^{-4} and 10^{-2}.

Items outside the scope of the concept:
- Environment influences like rain, night, snow, and scene illumination.
- Multi-point to multi-point communication.
- The routing of the information.
- Three-dimensional video.
- Application specific data.
- Hardware aspects concerning power, reliability and required number of floating point operations per second.
- The way synchronisation is achieved throughout the system.
- Legal aspects.
- Detailed economical aspects.

Assumptions:
- Available and reliable hardware.
- Transmission errors possible
- Available connection to backbone of 1 gigabit per second, delay is the speed of light, the bit error rate 10^{-12}.
- No interfering traffic.

BIBLIOGRAPHY

[1] "Advanced Communications Technologies and Services", http://www.de.infowin.org/.

[2] "Frame structure for a 64 to 1920 kbit/s channel in audio-visual teleservices", International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.221, 1993.

[3] "Information technology - Generic coding of moving pictures and associated audio for digital storage media at up to 1,5 Mbit/s - part 2: Video", ISO/IEC 11172-2, 1993.

[4] "Information technology - Generic coding of moving pictures and associated audio information - part 2: Video", ISO/IEC/ DIS 13818-2, 1994.

[5] "Media stream packetisation and synchronisation on non-guaranteed quality of services LANs", International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.225.0, 1996.

[6] "Multiplexing protocol for low bit rate multi-media communication", International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.223, 1995.

[7] "Narrow-band visual telephone systems and terminal equipment", International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.320, 1993.

[8] "Telephone systems and equipment for local area networks which provide a non-guaranteed qualtity of service", International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.323, Visual, 1996.

[9] "Terminal for low bit rate multi-media communication", International Telecommunications Union, Geneva, Switzerland, ITU-T Draft Recommendation H.324, 1997.

[10] “Description of Error Resilience Core Experiments”, ISO/IEC JTC1/SC29WG11, N1473, Maceio, Brazil, November 1996.

[11] “Mobile Multi-media Communication project”, http://www.mmc.tudelft.nl.

[12] “Multiplexing protocol for low bit rate mobile multi-media communications”, International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.223/Annex A-Annex C, 1998.

[13] “Report on error conditions for error resilience core experiments”, ISO/IEC JTC1/SC29/WG11 N1223, Florence, MPEG meeting, March 1996.

[14] “Telenor”, http://www.telenor.no.

[15] “Video codec for audiovisual services at p x 64 kbits”, International Telecommunications Unios, Geneva, Switzerland, ITU-T Recommendation H.261, 1993.

[16] “Video coding for low bitrate communication", International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.263, 1998.

[17] Aign, S., Fazel, K., “Temporal and spatial error concealment techniques for a hierarchical MPEG-2 video codec”, Proceedings Globecom ’95, pages 1778-1783.

[18] Alatan, A.A.,”Joint Utilization of Fixed and Variable-Length Codes for Improving Sychronization Immunity for Image Transmission”, Proceedings ICIP-98, Chicago, October 1998.

[19] Aldid, J., et alteres, "Magic into reality, building the WAND modem", Proceedings ACTS Mobile Communication Summit '97, Acelborg, Denmark, October 1997, pages 734-744.

[20] Andriessen, J.H.T.H., "Cooperation at a distance; understanding and evaluating computer supported interaction at work", Private communications.

[21] Andriessen, J.H.T.H., "De werkelijkheid van videovergaderen; sociale en psychologische voorwaarden", Private communications.

[22] Anker, F.W.G. van den, Arnold, A.G., “Multimedia in the work environment: towards a task- and user-centred approach”, International Congress Multimedia Minded, Lanaken, Belgium, 1996.

[23] Appadwedula, S., et alteres, “Joint source channel matching for wireless image transmission”, Proceedings of the International Conference on Image Processing, Chicago, October 1998.

[24] Aravind, R., Civanlar, M.R., Reibman, A.R., “Packet loss resilience of MPEG-2 scalable video coding algorithms”, Transactions on Circuits and Systems for Video Technology, volume 6, pages 426-435, October 1996.

[25] Ayanoglu, E., Pancha, R., Reibman, A.R., Talwar, S., “Forward error control for MPEG-2 video transport in a wireless ATM LAN”, ACM/Baltzer Mobile Networks Applications, volume 1, number 3, pages, 245-258, December 1996.

[26] Bakker, J.D., F.H.P. Spaan, “Establishing a Trade-off between Error Robust Network Protocols and Error Robust Video Compression Algorithms”, Proceedings 19th Symposium on Information Theory in the Benelux, 1998.

[27] Bakker, J.D., Prasad, R., “Efficient Protocols for a Radio Link in a Wireless Multimedia Environment”, Proceedings PIMRC ’97, 1997.

[28] Bakker, J.D., Schoute, F.C., Prasad, R.,” An air interface for high bandwidth cellular digital communications on microwave frequencies”, Proceedings VTC’98, 1998.

[29] Batllo, J.C., Vaishampayan, V.A., ”Asymptotic performance of multiple description transform codes”, IEEE Transactions on Information Theory, volume 43, pages 703-707, March 1997.

[30] Bosveld, F., “Hierarchical Video Compression using SBC” Ph.D. Thesis, Delft University of Technology, 1996.

[31] Breeuwer, M., Veldhuis, R., “An introduction to source coding”, Prentice Hall, 1993.

[32] Brigger, P., Kunt, M., “Morphological Shape Representation for Very Low Bit Rate Video Coding”, Signal Processing – Image Communication 7, pages 297-311, 1995.

[33] Cen, S., et alteres, “A distributed real time MPEG video audio player”, Proceedings Fifth International Workshop on Network and Operating System Support for Digital Audio and Video”, Durham, NH, U.S.A., pages 151-162, April 1995.

[34] Challapali, K., et alteres, “The grand alliance system for US HDTV”, Proceedings IEEE, volume 83, pages 158-174, February 1995.

[35] Chen, Z., et alteres, “Real time video and audio in the world wide web”, Proceedings Fourth World Wide Web Conference, 1995.

[36] Cheng, N.T., Kingsbury, N.G., “The EREC: an efficient error resilient technique for encoding positional information on sparse data”, IEEE Transactions on Communication, volume 40, pages 140-148, January 1992.

[37] Chung, D., Wang, Y., “Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms”, unpublished.

[38] Cochennec, J.Y., “Method for the correction of cell losses for low bit rate signals transport with the AAL type 1”, ITU-T SG15 Document AVC-538, July 1993.

[39] Correia, L.M., R. Prasad, "An overview of wireless broadband communications", IEEE Communications Magazine, volume 35, pages 28-33, January 1997.

[40] Dance, F.E.X., “The concept of communication”, Journal of Communication 20, pages 201-210, 1970.

[41] Ducla-Soares, L., Pereira, F., "Error resilience and concealment performance for MPEG-4 frame-based video coding", Signal Processing: Image Communication 14, pages 447-472, 1999.

[42] Ebrahimi, T., “MPEG-4 video verification model 8.0”, ISO/IEC JTC1/SC29WG11 N1796, July 1997.

[43] El-Gamal, A.A., Cover, T.M., “Achievable rates for multiple descriptions”, IEEE Transactions on Information Theory, volume IT-28, pages 851-857, November 1982.

[44] Emmert, P., Donaghy, W.C., “Human Communication”, Addison-Wesley Series in Communication, 1981.

[45] Farvardin, N., Vaishampayan, V.A., “Optimal quantizer design for noisy channels: an approach to combined source-channel coding”, IEEE Transactions on Information Theory, volume IT-38, pages 827-838, November 1987.

[46] Ferguson, T.J., Ranowitz, J.H., “Self-synchronising Huffman codes”, IEEE Transactions on Information Theory, volume IT-30, pages 687-693, July 1984.

[47] Gasull, A., Marques, F., “ Lossy Image Contour Coding with Multiple Grid Chain Code”, Workshop on Image Analysis and Sythesis in Image Coding, Berlin, Germany, pages B4, October 1994.

[48] Ghanbari, M., “Cell loss concealment in ATM video codes”, IEEE Transactions on Circuits and Systems for Video Technology, volume 3, pages 238-247, June 1993.

[49] Ghanbari, M., “Postprocessing of late cells for packet video”, IEEE Transactions on Circuits and Systems for Video Technology, volume 6, pages 669-678, December 1996.

[50] Ghanbari, M., “Two layer coding of video signals for VBR networks”, IEEE Journal on Selected Areas in Communication, volume 7, pages 801-806, June 1989.

[51] Gonzalez-Rosiles, G., Cabrera, S.D., Wu, S.W., “Recovery with lost subband data in overcomplete image coding”, Proceedings ICASSP ’95, Detroit, MI, U.S.A., volume 2, pages 1476-1479, 1995.

[52] Gregg, W.D., "Analog and Digital Communication", Wiley & Sons, 1977.

[53] Hacker, W., "Arbeitspsychologie; Psychische regulation von Arbeitstätigkeiten", Bern, 1986.

[54] Haskell, B.G., Puri, A., A.N. Netravalli, "Digital Video: An Introduction to MPEG-2", Chapman and Hall, New York, 1997.

[55] Haskell, P., Messerschmitt, D., “Resynchronisation of motion compensated video affected by ATM cell loss”, Proceedings ICASSP ’92, San Francisco, CA, U.S.A., volume 3, pages 545-548, March 1992.

[56] Hemami, S.S., “Reconstruction optimized lapped orthogonal transforms for robust image transmission”, IEEE Transactions on Circuits and Systems for Video Technology, volume 6, pages 168-181, April 1996.

[57] Hemami, S.S., Gray, R.M., “Image reconstruction using vector quantised linear interpolation”, Proceedings ICASSP ’94, Australia, volume 5, pages 629-632, May 1994.

[58] Hemami, S.S., Gray, R.M., “Subband coded image reconstruction for lossy packet networks”, IEEE Transactions on Image Processing, volume 6, pages 523-539, April 1997.

[59] Hemami, S.S., Meng, T.H.Y., “Transform coded image reconstruction exploiting interblock correlation”, IEEE Transactions on Image Processing, volume 4, pages 1023-1027, July 1995.

[60] Honcharenko, W., et alteres, "Broadband wireless access", IEEE Communications. Magazine, volume. 35, pages 20-26, January 1997.

[61] Horne, C., Reibman, A.R., “Adaptation to cell loss in a two layer video codec for ATM networks”, Proceedings Picture Coding Symposium, March 1993.

[62] Jain, A.K., “Fundamentals of Digital Image Processing”, pages 548-551, Prentice-Hall, 1989.

[63] Jayant, N.S., Noll, P., "Digital coding of waveforms", Prentice Hall, 1984.

[64] Kansari, M. et alteres, “Low bit rate video transmission over fading channels for wireless microcellular systems”, IEEE Transactions on Circuits and Systems for Video Technology, volume 6, pages 1-11, February 1996.

[65] Kawahara, T., Adachi, S., "Video transmission technology with effective error protection and tough synchronization for wireless channels", Proceedings IEEE International Conference on Image Processing, Lausanne, Switzerland, pages 101-104, November 1996.

[66] Khansari, M., Vetterli, M., "Layered transmission of signals over power constrained wireless channels", Proceedings Proceedings IEEE International Conference on Image Processing, Washington DC, U.S.A., volume 3, pages 380-383, October 1995.

[67] Kieu, L.H., Ngan, K.N., "Cell loss concealment techniques for layered video codecs in an ATM network", IEEE Transactions on Image Processing, volume 3, pages 666-677, September 1994.

[68] Kim, H.J., JP. Linnartz, "Virtual cellular structure: a new wireless communications architecture with multiple access ports", *Proc VTC '94*, pages 1055-1059, 1994

[69] Kinoshita, T., Nakahashi, T., Maruyama, M., "Variable bit rate HDTV codec with ATM cell loss compensation", IEEE Transactions on Circuits and Systems for Video Technology, volume 3, pages 230-237, June 1993.

[70] Koenen, R. (Ed.), Overview of the MPEG-4 Version 1 Standard, ISO/IEC JTC1/SC29WG11 N2725, Seoul, South Korea, March 1999.

[71] Kurtenbach, J., Wintz, P.A., "Quantizing for noisy channels", IEEE Transactions on Communication Technology, volume CT-17, pages 291-302, April 1969.

[72] Kwok, W., Sun, H., "Multi-directional interpolation for spatial error concealment", IEEE Transactions on Consumer Electronics, volume 39, pages 455-460, August 1993.

[73] Lagendijk, R.L., Biemond, J., Quist, C.P.: 'Low Bit Rate Video Coding for Mobile Multi-media Communications', Proceedings EUSIPCO-96, 1996.

[74] Lam, A.M., Reibman, A., "An error concealment algorithm for images subject to channel errors", IEEE Transactions on Image Processing, volume 4, pages 533-542, May 1995.

[75] Lam, W.M., Reibman, A.R., "Self-synchronising variable length codes for image transmission", Proceedings ICASSP '92, San Francisco, CA, U.S.A., volume 3, pages 477-480, March 1992.

[76] Lam, W.M., Reibman, A.R., Liu, B., "Recovery of lost or erroneously received motion vectors", Proceedings ICASSP '93, Minneapolis, U.S.A., volume 5, pages 417-420, April 1993.

[77] Lee, S.H., "Technical Description S4 Geometrical Representation Method S4e: Baseline Based Shape Coding", ISO/IEC JTC1/SC29/WG11, 1996.

[78] Lee, S.H., Lee, P.J., Ansari, R. "Cell loss detection and recovery in variable rate video", Proceedings Third International Workshop on Packet Video, Morriston, March 1990.

[79] Lee, X., Zhang, Y.Q., Leon-Garcia, A., "Information loss recovery for block based image coding techniques – a fuzzy logic approach", IEEE Transactions on Image Processing, volume 4, pages 259-273, March 1995.

[80] Lei, S.M., Sun, M.T., “An entropy coding system for digital HDTV applications”, IEEE Transactions on Circuits and Systems for Video Technology, volume 1, pages 147-154, March 1991.

[81] Lin, S., Costello, D.J., "Error control coding: fundamentals and applications", Englewood Cliffs, NJ, Prentice Hall, 1983.

[82] Littlejohn, S.W., “Theories of Human Communication”, Wadworth Inc., 1983.

[83] Malvar, H.S., Staelin, D.H., “The LOT: transform coding without blocking effects”, IEEE Transactions on Acoustics, Speech and Signal Processsing, volume 38, pages 553-559, April 1989.

[84] Matic, D., “Indoor and outdoor frequency measurements for mm-waves in the range of 60 GHz”, , H.Harada and R. Prasad, IEEE Vehicular Technology Conference (VTC) '98, Ottawa, Canada 18-21 May, 1998.

[85] Maxted, J.C., Robinson, J.P., “Error recovery for variable length codes”, IEEE Transactions on Information Theory, volume IT-31, pages 794-801, 1985.

[86] Mikkonen, J., et alteres, "Emerging wireless broadband networks", IEEE Communication Magazine, February 1988.

[87] Mitchell, J.L., et alteres, "MPEG Video Compression Standard", Chapman and Hall, New York, 1996.

[88] Mitchell, O.R., Tabatabai, A.J., “Channel error recovery for transform image coding”, IEEE Transactions on Communications, volume COM-29, pages 1754-1762, December 1981.

[89] Morinaga, M., M. Nakagawa, R. Kohno, "New concepts and technologies for achieving highly reliable and high capacity multimedia wireless communications system", IEEE Comm. Mag., volume 35, pages 34-40, January 1997.

[90] MPEG-7 Requirements Group, “MPEG-7, context and objectives”, version 10, ISO/IEC JTC1/SC29WG11 N2460, Atlantic City, U.S.A., October 1998.

[91] Mukawa, N., Kuroda, H., Matuoka, T., “An interframe coding system for video teleconferencing signal transmission at a 1,5 Mbit/s rate”, IEEE Transactions on Communications, volume COM-32, pages 280-287, March 1984.

[92] Narula, A., Lim, J.S., “Error concealment techniques for an all-digital high definition television system”, Proceedings SPIE Conference on Visual Communication Image Processing, Cambridge, MA, U.S.A., pages 304-315, 1993.

[93] Neuman, P.G., “Self-synchronising sequential coding with low redundancy”, Bell Systems Technical Journal, volume 50, pages 951-981, March 1971.

[94] Ngan, K.N., Stell, R., "Enhancement of PCM and DPCM images corrupted by transmission errors", IEEE transactions on communications, volume COM-30, pages 257-265, January 1982.

[95] Ohta, N., "Packet video: modeling and signal processing", Norwood, MA, Artech House, 1994.

[96] Oosterhoff, F.H., “Exploring binocular rivalry in transparency situations”, ECVP perception supplement 1998, Oxford, England, August 1998.

[97] Orchard, M.T., Wang, Y., Vaishampayan, V.A., Reibman, A.R., “Redundancy rate distortion analysis of multiple description coding using pairwise correlating transforms”, Proceedings IEEE International Conference on Image Processing, Santa Barbara, CA, U.S.A., volume 1, pages 608-611, October 1997.

[98] Otterloo, van, “A contour-oriented approach to digital contour analysis”, Ph.D. Thesis, Delft University of Technology, 1988.

[99] Ozarow, L., “On a source coding problem with two channels and three receivers”, Bell Systems Technical Journal, volume 59, page 1921, December 1980.

[100] Parthasarathy, V., Modestino, J.W., Vastola, K.S., “Design of a transport coding scheme for high quality video over ATM networks”, IEEE Transactions on Circuits and Systems for Video Technology, volume 7, pages 358-376, April 1997.

[101] Pennebaker, W.B., J.L. Mitchell, "JPEG Still Image Data Compression Standard", Van Nostrand Reinhold, New York, 1993.

[102] Philips, W., “Adaptive coding using warped polynomials”, Proceedings ICASSP, 1996.

[103] Pinches, C. L.; Kuijpers, E. A.; Spaan, F., "A study on focusing in telescience using the TelePODI breadboard", January 1991.

[104] Prasad R., "Wireless Broadband Communication Systems", IEEE Communications Magazine, volume 35, page 18, January 1992.

[105] Prasad, R., et alteres, “A Multi-disciplinary approach towards new Mobile Multimedia systems”, private communication.

[106] Prisoli, F.D., R. Velt, "Design of medium access control and logical link control functions for ATM support in the MEDIAN system", Proc. ACTS Mobile Comm. Summit '97, Acelborg, Denmark, October 1997, pages 734-744.

[107] Rabbani, M., Jones, P.W.,”Digital Image Compression Techniques”, SPIE, 1991.

[108] Ramamurthy, G., Raychaudhuri, D., “Performance of packet video with combined error recovery and concealment”, Proceedings IEEE Infocomm ’95, pages 753-761, 1995.

[109] Ramchandran, K., Ortega, A., Uz, K.M., “Multi-resolution broadcast for digital HDTV using joint source channel coding”, IEEE Journal on Selected Areas in Communication, volume 11, pages 6-23, January 1993.

[110] Rasmussen, J., "Skills, rules, and knowledge: signals, signs, and symbols, and other distictions in human performance models", IEEE Transactions on Systems, Man and Cybernetics, volume 13, page257, 1983.

[111] Reason, J., "Human error", New York: Cambridge, University Press, 1990.

[112] Redmill, D.W., Kingsbury, N.G., “The EREC: an error resilient technique for coding variable length blocks of data”, IEEE Transactions on Image Processing, volume 5, pages 565-574, April 1996.

[113] Rheinschmitt, R., A. de Haz, M. Umehinc, " AWACS MAC and LLC functionality", Proceedings ACTS Mobile Communication Summit '97, Acelborg, Denmark, October 1997, pages 745-750.

[114] Riskin, “Optimal Bit Allocation via the Generalized BFOS Algorithm,” IEEE Trans. on Inform. Theory, volume 37, pages 400-402, March 1991.

[115] Rose, K.M., Heiman, A., "Enhancement of one-dimensional variable length DPCM images corrupted by transmission errors", IEEE transactions on communications, volume 37, pages 373-379, April 1989.

[116] Sadka, A.H., Eryurtlu, F., Kondoz, A.M.: ‘Modified H.263 Video Coding Algorithm for Mobile Multimedia Communications’. Proceedings Second Erlangen Symposium, 25 April 1997.

[117] Schoute, F.C., "Mobile Multimedia Communication", Proceedings European Microwave and Wireless '98, Amsterdam, the Netherlands, October 1998.

[118] Serra, J., “Image Analysis and Mathematical Morphology”, Academic Press, 1982.

[119] Shannon, C.E., “A mathematical theory of communication”, Bell Systems Technical Journal, volume 27, pages 379-423, 623-656, 1948.

[120] Silva, J.S. da, et alteres, "Mobile and personal communications: ACTS and beyond", PIMRC '97, Helsinki, Finland, September 1997.

[121] Spaan, F. H. P., de Jager, C., Nieuwenhuijzen, H., Kondo, Y., "Micro-turbulence in the upper photosphere of Alpha Persei (F5 Ib) derived from ultraviolet spectral observations", Astronomy and Astrophysics (ISSN 0004-6361), volume 185, numbers 1-2, pages 229-232, October 1987.

[122] Spaan, F.H.P., “Error resilient shape compression based on polar coordinates and the discrete cosine transform”, submitted to Journal of Visual Communication and Image Representation, California, U.S.A.

[123] Spaan, F.H.P., “Error-robust contour-coding using polar coordinates”, Proceedings, ASCI '98, 1998.

[124] Spaan, F.H.P., Kramer, A.J., “Analysis of the alignment of a Schlieren system for tele-operation”, Proceedings VIIIth European Symposium on Materials and Fluid Sciences in Micorgravity, Brussels, Belgium, 12-16 April 1992, ESA SP-333, pages 435-440, August 1992.

[125] Spaan, F.H.P., Lagendijk, R.L., Biemond J., “Error robust video coding based on H.263”, Proceedings IEEE International Conference of Image Processing, Chicago, U.S.A., October 1998.

[126] Spaan, F.H.P., Lagendijk, R.L., Biemond J., “Error-robustness of polar contour-coding*”, Lecture Notes in Computer Science 1425*, David Hutchison and Ralf Schaefer (Eds.), Proceedings Third European Conference on Multi-media Applications, Services and Techniques ECMAST ’98, Springer Verlag, Berlin, pages 402-414, May 1998.

[127] Spaan, F.H.P., Lagendijk, R.L., Biemond J., “Shape coding using polar coordinates and Fourier transformation”, Proceedings IEEE Conference On Image Processing 1997, Santa Barbara, CA, U.S.A, volume 1, pages 516-519, October 1997.

[128] Spaan, F.H.P., Lagendijk, R.L., Biemond, J., "A Concept for a Mobile Multimedia Communication system", Proceedings IEEE Workshop on Multi Media Signal Processing, Copenhagen, Denmark, September 1999.

[129] Spilker, J.J. Jr., “Digital Communications by Satellite. Englewood Cliffs, NJ, U.S.A., Prentice Hall, 1997.

[130] Sun, H., Challapali, K., Zdepski, J., “Error concealment in a digital simulcast AD-HDTV decoder”, IEEE Transactions on Consumer Electronics, volume 38, pages 108-117, August 1992.

[131] Sun, H., Kwok, W., “Concealment of damaged block transform coded images using projections onto convex sets”, IEEE Transactions on Image Processing, volume 4, pages 470-477, April 1995.

[132] Swann, R., Kingsbury, N.G., “Transcoding of MPEG-2 for enhanced resilience to transmission errors”, Proceedings IEEE International Conference on Image Processing, Lausanne, Switzerland, volume 2, pages 813-816, November 1996.

[133] Talluri, R., “Error-Resilient Video Coding in the ISO MPEG-4 Standard”, IEEE Communications Magazine, June 1998, pages 112-119.

[134] Tekalp, A.M., "Digital Video Processing", Prentice Hall, Upper Saddle River NJ, 1995.

[135] Tom, A.S., Yeh, C.L., Chu, F., “Packet video for cell loss protection using deinterleaving and scrambling”, Proceedings ICASSP ’91, Toronto, Canada, pages 2857-2850, April 1991.

[136] Torres, L., Kunt, M., (Eds.) “Video Coding, the Second Generation Approach”, Kluwer Academic Publishers; 1996.

[137] Turletti, T., Huitema, C., “RTP payload format for H.261 video streams”, IETF RFC 2032, October 1996.

[138] Vaishampayan, V.A., Domaszewicz, J., “Design of entropy constrained multiple description scalar quantizers”, IEEE Transactions on Information Theory, volume 40, pages 245-251, January 1994.

[139] Vaishampayan, V.A., “Application of multiple description codes to image and video transmission over lossy networks”, Proceedings Seventh International Workshop on Packet Video, Brisbane, Australia, pages 55-60, March 1996.

[140] Vaishampayan, V.A., “Design of multiple description scalar quantizers”, IEEE Transactions on Information Theory, volume 39, pages 821-834, May 1993.

[141] Vaishampayan, V.A., Farvardin, N., “Optimal block cosine transform image coding for noisy channels”, IEEE Transactions on Communication, volume 38, pages 327-336, March 1990.

[142] Vembu, S., Verdu, S., Steinberg, Y., “The source channel separation theorem revisited”, IEEE Transactions on Information Theory, volume 41, pages 44-54, January 1995.

[143] Wada, W., “Selective recovery of video packet loss using error concealment”, IEEE Journal on Selected Areas in Communication, volume 7, pages 807-814, June 1989.

[144] Wang, Y., "Error Control and Concealment for Video Communication: A Review", Proceedings of the IEEE, volume 86, no. 5, May 1998.

[145] Wang, Y., Chung, D., “Non-hierarchical signal decomposition and maximally smooth reconstruction for wireless video transmission”, Mobile Multi-media Communications, Goodman, D. and Raychaudhuri, D., Editors, New York, U.S.A., Plenum.

[146] Wang, Y., Orchard, M.T., Reibman, A.R., “Multiple desription image coding for noisy channels by pairing transform coefficients”, Proceedings IEEE First Workshop of Multi-Media Signal Processing, Princeton, NJ, U.S.A., pages 419-424, June 1997.

[147] Wang, Y., Ramamoorthy, V., “Image reconstruction from spatial subband images and its applications in packet video transmission”, Signal Processing, volume 3, numbers 2-3, pages 197-229, June 1991.

[148] Wang, Y., Zhu, Q.F., Shaw, L., “Maximally smooth image recovery in transform coding”, IEEE Transactions on Communications, volume 41, pages 1544-1551, October 1993.

[149] Wen, G., Villasenor, J., “A Class of Reversible Variable Length Codes for Robust Image and Video Coding”, Proceedings International Conference on Image Processing, Santa Barbara, CA, U.S.A, volume 2, pages 65-68, October 1997.

[150] Wenger, S., “Video redundancy coding in H.263+”, Proceedings AVSPN, Aberdeen, United Kingdom, September 1997.

[151] Westerink, C., Biemond, J., Boekee, D.E., “An Optimal Bit Allocation Algorithm for Sub-Band Coding,” Proceedings ICASSP, New York, U.S.A., pages757-760, 1988.

[152] Wolf, J.K., Wyner, A., Ziv, J., “Source coding for muliple descriptions”, Bell Systems Technical Journal, volume 59, pages 1417-1426, October 1980.

[153] Wu, G., et alteres, "A wireless ATM oriented MAC protocol for high-speed wireless LAN", Proc. PIMRC '97, Helsinki, Finland, September 1997, pages 198-203.

[154] Yasuda, H., et alteres, “Transmitting 4 MHz TV signals by combinational difference coding”, IEEE Transactions on Communications, volume COM-25, pages 508-516, May 1977.

[155] Young, J.F., "Information Theory", Butterworth & Co., 1971.

[156] Zaletelj, J., Pecci, R., Spaan, F., Hanjalic, A., Lagendijk, R.L. “Rate Distortion Optimal Contour Compression Using Cubic B-Splines”, Proceedings European Signal Processing Conference, Rhodes, Greece, September 1998.

[157] Zhang, Y.Q., Liu, Y.J, Pickholtz, R.L., “Layered image transmission over cellular radio channels”, IEEE Transactions on Veh. Technology, volume 43, pages 786-796, August 1994.

[158] Zhu, C., “RTP payload format for H.263 video streams”, IETF draft, March 1997.

[159] Zhu, Q.F., “Device and method of signal loss recovery for real time and/or interactive communications”, U.S. Patent 5 550 847, August 1997.

[160] Zhu, Q.F., Wang, Y., Shaw, L., “Coding and cell loss recovery for DCT based packet video”, IEEE Transactions on Circuits and Systems for Video Technology, volume 3, pages 248-258, June 1993.

.

List of Abbreviations

Most of the abbreviations explained below arose from having to use them in a figure or table.

ATM	Asynchronous Transfer Mode
AWACS	ATM Wireless Access Communication System
BC	Before Christ
B-ISDN	Broadband ISDN
CIF	Common Intermediate Format
Codec	Compression Decompression system
CPN	Customer Premises Network
CPU	Central Processing Unit
dB	decibel
DCT	Discrete Cosine Transform
DECT	Digital Enhanced Cordless Telephony
DPCM	Differential Pulse Code Modulation
DQPSK	Differential QPSK
Dr	Doctor
Drs	Doctorandus
FDD	Frequency Division Duplex
FLPTMS	Future Public Land Mobile Telecommunications Service
GHz	Gigahertz
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HIPERLAN	High Performance LAN
HW	Hardware
Ir	Engineer
ISDN	Integrated Services Digital Networks
ITS	Information Technology and Systems
LAN	Local Area Network
LPF	Low Pass Filter
MB	Macro Block
Mb	Megabit
MBS	Mobile Broadband System
MEDIAN	Wireless Broadband CPN/LAN for Professional and Residential Multimedia Applications
MPEG	Moving Pictures Expert Group
OFDM	Orthogonal Frequency Division Multiplexing
OQPSK	Offset QPSK
OSI	Open Systems Interconnect
PhD	Physical Doctor
PHS	Personal Handyphone Systems
Prof	Professor
PSK	Phase Shift Keying
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter CIF

QPSK	Quadrature PSK
RAID	Redundant Arrays of Independent Disks
RGB	Red Green Blue
s	second
SAMBA	System for Advanced Mobile Broadband Applications
SHF	Super High Frequency
SW	Soft Ware
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
UMTS	Universal Mobile Telecommunications System
UNII	Unlicensed National Information Infrastructure
VBR	Variable Bit Rate
VSNR	Variance based Signal to Noise Ratio
WAND	Wireless ATM Network Demonstration
WLAN	Wireless LAN
WTM	Technology and Society
YUV	Luminance Chrominance U and V

SAMENVATTING IN HET NEDERLANDS

Communicatie speelt een belangrijke rol in ons leven. Er is bijna geen aspect van onze activiteiten te bedenken waar niet een of andere vorm van communicatie een rol bij speelt. Er zijn veel manieren waarop mensen met elkaar communiceren, zoals via taal of visueel. Visuele communicatie heeft bepaalde eigenschappen die het anders maken dan andere vormen van communicatie. Er zijn voor visuele communicatie, net als voor andere vormen van communicatie, technische middelen ontwikkeld, een voorbeeld is televisie. Televisie en video als medium voor communicatie zijn een deel van de Westerse cultuur geworden. Het videosignaal is steeds vaker digitaal in plaats van analoog en wanneer je met digitale video werkt, is een van de eerste dingen die je merkt de grote hoeveelheid data die nodig is om de beeldinhoud te representeren. Dit is een probleem voor zowel opslag als het overzenden van videodata. Daarom is compressie van de video data nodig, waar verschillende compressie standaarden voor bestaan. *Mobiele* visuele communicatie is nog niet algemeen in gebruik maar desondanks een belangrijk onderwerp in veel onderzoeksinstituten. Dit komt voort uit de recente tendens van de mens om op elk moment en overal te willen kunnen communiceren. De combinatie van *gecomprimeerde* videodata die verzonden moet worden naar en van een *mobiele* gebruiker zorgt voor problemen en uitdagingen.

Het Mobiele Multi-media Communicatie project van de Technische Universiteit Delft richt zich op het vinden van oplossingen voor dergelijke problemen. Vijf onderzoeksdisciplines zijn hier bij betrokken: de toepassing, de gebruikersinterface, de video compressie, het netwerk protocol en de transmissie. Hoewel er samengewerkt wordt, produceert ieder gebied zijn eigen onderzoeksresultaten. Om de integratie van de resultaten te bevorderen is de samenwerking gestimuleerd in het begin van het project en in een latere fase is een geïntegreerd concept ontwikkeld voor een toekomstig Mobiele Multi-media Communicatie systeem dat over vijf jaar operationeel zou kunnen zijn. Een andere stimulans was de gezamenlijke experimenteeromgeving waar geïntegreerde experimenten werden uitgevoerd.

Voor video compressie is het probleem dat specifiek optreedt bij mobiele video communicatie het optreden van fouten tijdens het overzenden van de gecomprimeerde bitstroom. De bestaande compressietechnieken zijn niet voldoende uitgerust om met deze fouten om te kunnen gaan. Daarom worden er tegenwoordig compressietechnieken ontwikkeld om het effect van de fouten te reduceren. In dit proefschrift worden hier voor nieuwe technieken gepresenteerd die elk hun voordelen hebben.

Wij hebben een standaard H.263 compressiesysteem zodanig aangepast dat het beter bestand is tegen fouten. In plaats van het vorige beeld te gebruiken als voorspelling voor het huidige, een voorspellingsbeeld wordt gebruikt dat is samengesteld uit een aantal vorige beelden. Dit smeert het effect van de fouten uit. Deze techniek werkt onafhankelijk van kennis over het aanwezig zijn van fouten. Daarbij is ook een foutdetectiesysteem aanwezig wat zoekt naar fouten in het gedecodeerde en gereconstrueerde beeld. Dit stuurt de posities van de fouten in het beeld naar de encoder die op die positie een macroblok intracodeert, welliswaar enige beelden later vanwege de vertraging. In de tussentijd worden door de decoder de effecten van de fouten zoveel mogelijk verborgen. De resultaten laten een behoorlijke verbetering in visuele kwaliteit zien bij een gelijk blijvende bit rate. Vergelijking van een deel van deze techniek met een deel van de technieken in de moderne MPEG-4 standaard laten zien dat onze techniek beter is, zij het in beperkte mate en ten koste van een verhoogd aantal bits.

Er is ook gekeken naar de vergelijking tussen deze techniek en het foutrobuuste netwerkprotocol dat eveneens ontwikkeld is in het Mobiele Multi-media Communicatie project. De kriteria waren de resulterende beeldkwaliteit en het aantal benodigde bits. Omdat dit een complex mulitdimensioneel probleem is werd een experimentele benadering gekozen. Het doel was een hoge beeldkwaliteit te krijgen terwijl de verhoging van het aantal benodigde bits zoveel mogelijk beperkt werd. We keken naar het effect van het toepassen van een fijnere kwantisatie, van een foutrobuust compressiealgoritme, van een foutrobuust netwerkprotocol en van combinaties van deze drie. We hebben gekeken naar het effect van gewone bitfouten en van samengestelde fouten in verschillende fout situaties. Voor het geheel van de situaties is de combinatie van het compressiealgoritme en het netwerkprotocol het beste. Voor enkele bitfouten hangt de keuze van de best methode af van de foutkans en dus van de kwaliteit van het transmissiekanaal. Voor samengestelde fouten is het compressiealgoritme nodig, eventueel in combinatie met het netwerkprotocol. Voor een eenvoudige oplossing is het netwerkprotocol het beste dat de meeste situaties aan kan, zeker voor wat betreft de enkele bitfouten.

In de huidige compressietechnieken zoals de MPEG-4 standaard, is het comprimeren van video objecten en hun contouren belangrijk. Er bestaan weinig foutrobuuste contourcompressiealgoritmen. Wij hebben een nieuwe techniek ontwikkeld die is gebaseerd op pool coördinaten en spectrale decompositie. Het bestand zijn tegen fouten wordt zowel analytisch als experimenteel aangetoond. Ook is deze techniek efficient voor wat betreft compressie en vereist weinig rekenkracht.

In het algemeen gesproken is het niet mogelijk mobiele digitale video communicatie te hebben zonder een vorm van additionele foutrobuustheid. De foutrobuustheid van de huidige technieken is voor ons doel onvoldoende en wij hebben dan ook enkele haalbare nieuwe technieken laten zien die beter tegen fouten bestand zijn.

Samenvatting van het proefschrift: “Foutrobuuste Videocompressie van Digitale Videodata”.

Frank Spaan, Delft, September 1999.

ACKNOWLEDGEMENTS

There are many people whom I wish to thank for making the work and this thesis a success and of whom only a few are mentioned here:

The promotores *Jan Biemond* and *Inald Lagendijk*. My colleague-researchers of the Mobile Multi-media Communication project team: *Fred van den Anker*, *Jan-Derk Bakker*, *Dusan Matic*, *Faye Oosterhoff* and also the post-docs *Carlos Rodriguez Casal* and *Hiroshi Harada*, and *Tineke Egyedi*. The people without whom the work could not have been carried out: *Eric van Doorn*, *Lia van den Enden*, *Danilo Tromp*, *Annet Bosch-van Dijk*, *Ben van den Boom* and many other people of *the 11th floor*. The project leader of the Mobile Multi-media Communication project: *Frits Schoute*. The people who contributed to this thesis as author or co-author of one of the papers on which parts of this thesis are based but have not been mentioned yet: *Ramjee Prasad*, *Laurens ten Horn* and *Arnold Vermeren*. The members of the promotiecommissie whom have not been mentioned yet: the *Rector Magnificus*, *Erik Andriessen*, *Ed Deprettere*, *Rob Koenen* and *Cees Slump*. My wife *Esther*.

A special thanks to Inald Lagendijk and Jan Biemond who made it possible for me to write my thesis in a much tighter time-schedule than originally planned.

.

CURRICULUM VITAE

Francisciscus Henricus Petrus (Frank) Spaan was born in Utrecht, the Netherlands, on December 27th, 1961. In 1980 he graduated from the gymnasium at the Christelijk Lyceum in Zeist. He studied astronomy at the Rijksuniversiteit Utrecht in Utrecht specialising in astronomical instruments and graduated in 1989 [121]. He then worked at the National Aerospace Laboratory at the department of Space Systems. The subject was the development of instruments for experiments to be conducted in micro-gravity [103][124]. In 1992 a change of interest effectuated a switch in the direction of art, art therapy and music therapy. In 1996 he turned to science once more and joined the Delft University of Technology as a Ph.D. student in the Information and Communication Theory Group at the Department of Information Technology and Systems. The subject was video compression in the context of mobile multi-media communication. As from September 1999 he studies in Stuttgart, Germany, to become a priest of the Christian Community.