JOURNAL ON

COMMUNICATIONS

VOLUME XLV. MAY-JUNE 1994

IMAGE PROCESSING I.

TUTORIAL PAPERS

REGULAR PAPERS

TRANSFORMS, TRANSFORM CODING

IMAGE COMPRESSION, CODING I.

Followed on the back cover.

EDITORIAL

Two special issues of the JOURNAL ON COMMUNICATIONS are devoted to the International Workshop on Image Processing: Theory, Methodology, Systems, and Applications, which was held on June 20-22, 1994, at the International Trade Center, Budapest, Hungary. It was sponsored by the European Association for Signal Processing EURASIP, the IEEE Hungary Section and Hungarian Radiocommunications Corporation ANTENNA HUNGÁRIA.

The objective of this Workshop was to provide a forum for discussion of new theoretical developments in image coding/processing and current as well as future applications to real-world problems. The progress of digitalization of visual media and telecommunication networks, supported by the advance of VLSI signal processing technologies, has given birth to the widespreading fields of new image and video services, which are connected one and all with broadband communications. The technical and economical effectiveness of the most new applications strongly depends on both the image/video processing methods, equipments and their accessibility through broadband communication services.

The experts of 18 countries presented their new results. In addition, 2 tutorials were given by well known scientists in the field, and also a round table was organized about the state-of-the-art of Multiresolution Image Processing.

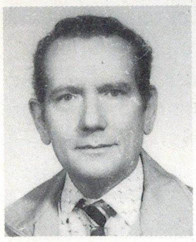
The most important topics included

- Image compression, coding
- Transforms, transform coding
- Filtering
- Restoration, recognition
- Image analysis
- Texture analysis
- Multidimensional processing
- Remote sensing.

A small technical exhibition was organized concurrently with the workshop presenting new results in processing methods and in their simulations.

The Workshop and this special issue could not be realized without the substantial help and activity of the Scientific Society for Telecommunications and of the Sponsors. Thanks are due to the members of Local Program Committee who helped the organization, and last but not least to the editor in chief, A. Baranyi, for giving me the opportunity and all support necessary to compile this special issue.

K. FAZEKAS



Kálmán Fazekas received the M. Sc. degree in electrical engineering in 1962 and in teacher-engineering in 1968 from the Technical University of Budapest. He is associate Professor in the Department of Microwave Telecommunications at the Technical University of Budapest. His research interests and activity includes image coding, image processing and space communications. In the last decade, the main sponsors of his research works were the Hungarian Academy of Sciences, the Hungarian PTT, the National Committee of Technical Development. He participated during the last two decades in the development of several image digitizer equipments, digital image transmission system for surveillance, digital codecs, etc. He is the author and coauthor of 8 text books, more than 20 longer research studies, and 80 papers and conference presentations. He is the member of the New York Academy of Sciences from 1993. He is the member of several societies: IEEE (Senior Member), EURASIP, SPIE, HTE (Sceintific Society Society for Telecommunications in Hungary), OPAKFI (Optical, Acoustical and Filmtechnical Society in Hungary), MMEV (World Federation of Hungarian Engineers and Architects). At present, he is the secretary of the Joint COM, ED, AP, MTT Chapter of IEEE Hungary Section. He participates in the work of COST 229 project, he is the member of the Management Committee of this project. He was the member of teaching staff of the ERASMUS Short Course in spring 1992, at TU Delft. He is professional expert in the field of digital transmission of information and of digital image processing, and of microelectronics. He got for his professional activies in 1981 the award of Minister of Education, for a text book series in 1986 the award of the Rector at the Technical University of Budapest and for the best paper in 1988 and in 1993 the Pollák-Virág award from HTE. He is the coordinator of several scientific cooperation between Dept. of Microwave Telecom. and the following foreign partners: Tu Delft, Telecom Paris, TU Kosice (Slovakia), TU Vigo (Spain), TU Ljubljana (Slovenia), etc. in the field of digital image transmimssion. He was the organizer of several workshops, conferences and meetings, reviewer for EUSIPCO and EUROMICRO. He has served as the Chairman of the COST 229 Workshop on Intelligent Terminals and Source and Channel Coding, in September 1993, in Budapest.

DIGITAL HDTV SYSTEM COMPRESSION

Z. S. BOJKOVIĆ

UNIVERSITY OF BELGRADE
VOJVODE STEPE 305, BELGRADE, YUGOSLAVIA

The purpose of this paper is to present digital high definition televison DHDTV system compression which is one of the most important current problem in modern television. First, various standards, from the DHDTV system compression point of view, will be presented and discussed. The second part of the paper addresses the problem of the MPEG image data compression algorithm. A technical overview of MPEG algorithm, a robust video compression and transport system will be presented. Spatial redundancy reduction, temporal redundancy reduction as well as error concealment strategy are provided, too. Next, the attention is focussed on DHDTV coding approach including bit allocation for subband compression. Finally, advanced DHDTV as a layered digital system will be considered.

1. INTRODUCTION

High Definition Television HDTV is one of the most important current development today when society is rapidly moving into the information age. It has a strong impact on various technologies like microelectronics, image processing, display techniques, optical transmission, magnetic recording and others. This is of great significance not only for TV customer electronics but also for medical data processing communication techniques and so on. HDTV is gaining increasing attention not only for broadcasting but also for a wide range of other applications. Therefore a nation-wide HDTV transmission and distribution network must be developed in order to offer economical HDTV services to the customer.

Digital transmission is a promising method for the delivery of high definition television signals, especially when a high quality signal is desired at the receiver. This is particularly attractive for the future broadband network where different services such as a voice, data and video will share resources in a common network. Of course, it is assumed that the network will be fiberglass-based and will offer wide bandwidth capability. Such a network will represent a culmination of the current evolution in which different services will be offered with integrated access, transport switching and network management. In view of this, it is desirable and it turns out to be feasible, to significantly compress the bandwidth of the row HDTV signal while maintaining high quality.

HDTV will display images with about 1000 scan lines on screen that have aspect ratios of 16:9 instead of the current 4:3. Luminance and chrominance will be properly separated for excellent colour rendition, while sound will be in stereo at compact-disc quality. As for viewers, they will have dramatically enhanced viewing. To secure it, broadcasters will need to transmit far more information per second to the receiver than they do today (1). To facilitate this goal, engineers are concentrating on developing systems that deliver programming directly to homes from high-power satellites, using direct-broadcast-satellite delivery requiring reception with very small dish aerials.

Improving the quality of television poses a series of novel technological challenges. Since exceptionally large bandwidth is needed for transmission of HDTV signals in either analog or digital form, products using this technology will require sophisticated, real time signal processing and compression techniques. They will also require large amounts of memory and specialized logic circuitry, all in VLSI. Furthermore, HDTV will help bring about a new generation of computers in the form of multimedia workstations. For these reasons, the market for HDTV is likely to be a driving force for technologies like semiconductors, computers and telecommunications.

At a meeting in Luxembourg in June 1993. the Ministers for Telecommunications adopted the Community plan of action for introducing HDTV in Europe, thereby opening a financial package of 228 million ECU over 4 years. This sum will be increased to 405 million ECU by finance from other sources, preferably from the private sector, though without routing out the use of public funds. The agreement reached concerns a Council resolution the development of technology and standards in the field of HDTV services, with a framework agreement attached for the plan of action. Any extra costs incurred by manufactures and broadcasters as a result of the introduction of such sources will be financed by the plan, which is envisaged for a period of 4 years, ending on 30 June 1997 [2].

Many compression techniques have been developed to reduce the bit rate for digital image transmission, maintaining, at the same time, a good fidelity between the original and the compressed image. Some of the most powerful compression algorithms are based on the Discrete Cosine Transform (DCT). In this technique compression is attained exploiting the spatial and temporal relation of the original image pixels [3]. Data compression techniques will be extremely important in the development of HDTV. In this case, the information needs a very high compression rate. Further, good quality is a fundamental request for HDTV images. Hence, sophisticated compression algorithms are required to avoid a significant distortion in the compressed images. Further degradation is more evident in compressed images, since compressed data are generally much more sensitive to noise than correlated source data. The correlation of original data, made during the compression step, introduces a high correlation between noise samples and pixels of the received and decompressed image. Hence, errors in few compressed data spread over many pixels of the reconstructed image. Degradation will be as more evident as more complex is the compression technique.

First part of this paper deals with standard activities. Various standards for digital HDTV system compression will be taken into consideration. The second part (Sections 3 and 4) deals with image data compression from the point of view of MPEG algorithm, a robust video compression and transport system for digital HDTV will be presented, too. Digital HDTV approach including bit allocation for subband compression for HDTV will be addressed in Sections 5 and 6. Section 7 seeks to provide advanced digital HDTV as a layered digital system, while Section 8 summarizes the paper.

2. STANDARDIZATION

At the moment, various standards are under discussion for studio equipment, transmission and for display of the TV signal. Japan, Europe and United States are competing to establish a world standard for the next generation of TV system. Standard is necessary whenever there are more than one solution competing for the same problem.

Japan's MUSE and Europe's HDMAC are still based on analog transmission whereas in the US the discussion on all digital systems comes up. The development of Broadband Integrated Services Digital Network (BISDN) also tends towards the consistent expansion for digital techniques in the studio transmission.

The Japanese HDTV system is the result of long range efforts started in late seventies within a programme called "high vision", especially under the leadership of the state owned broadcast company and NHK (Nippon Hoso Kyokai) [4]. The first prototype demonstration was available around 1986. Simpleminded compression scheme gives the name of the system MUSE standing for multiple subnyqist sampling in coding. This system has been designed for satellite broadcasting in the bands of 12 GHz. The transmission system adopted is analog frequency modulation requiring a transmission bandwidths of 27 MHz for a base band within 8 MHz. The video signal of the MUSE system has 1125 lines, 30 frames and 60 fields per second. It is an interlaced scan. The video signal is processed component by component, chrominance signal being compressed through so-called time compressed integration. Compression by a factor of 4 is used to fit chrominance signal in the line blanking interval. The digital sound is inserted in the field blanking interval. Probably, the greatest merit of the new MUSE system is its existence. In 1991, they already had daily broadcast for about one or two hours a day, which has been increased to 8 hours since January 1992. Its second merit is that it triggered an international competition between Europe, Japan and the US for the development of new television system.

The European answer to the MUSE system was the establishment of a European project called EUREKA '95 which came up with an equivalent system called HDMAC [1]. In contrast with the MUSE system which is not compatible with the existing TV systems, like NTSC, PAL or SECAM, HDMAC has been introduced in 1988 with the argument of compatibility. It refers to compatibility with a D2-MAC system, which is even today not yet fully operational. The D2-MAC is a component system in which luminance and chrominance signals are sampled and time compressed to fit within 64 ms of one video scan line, including audio and data. Because of time compression, the initial bandwidths of 5,6 MHz and 1,6 MHz for luminance and chrominance, respectively, increase to 8,4 and 4,8 requiring a global sampling frequency of 20,25 MHz. The D2-MAC system is the result of an evolution which lasted for about 10 years, starting with A-MAC, B-MAC and so on. On the other hand the HDMAC system is an upscaled version of D2-MAC. Accordingly by undersampling a high resolution video signal one can bring it down to D2-MAC format. The input to HDMAC system is a video signal with 1250 pictures lines interlaced, scanned with 50 fields and 25 frames. This signal is sampled. The resulting picture is than decimated into 4 subsets using odd and even numbered lines and columns. After some spatial arrangement, this subset became the fields of the D2-MAC system, which could be recombined to make the interlaced HDMAC frames. This scheme is for the still picture case. The initial signal is processed to determine three categories of areas in each picture: fix areas, slowly moving areas and moving areas. Each area is analyzed according to these three hypothesis and processed for compression using a trade-off between time resolution and space resolution. The results are then compared to the original area and a decision is made to put the appropriate level where the similarity is the highest. Both MUSE and HDMAC systems have very simple data compression scheme.

It is required that American HDTV should be based on terrestrial transmission and should fit within the 6 MHz bandwidths as presently used for regular NTSC, with no interference to the existing NTSC system. In early 1989, there were many systems being developed in the US (Advanced Compatible Television developed by David Sarnoff Center and Thomson Consumer Electronics, the North American Philips developed another system based on NTSC, as well as the GLEN system, the Zenith system, two versions of the MUSE system, one being incompatible and the other one compatible with NTSC receivers). These systems were mainly hybrid combining some analog and some digital signal processing and modulation. The most up-to-date system was the early MITCC system based on subband coding and proposed by Schreiber. The FCC opened a competition to find the best HDTV system for American needs. Namely, engineers from the General Instrument Corporation came up with the first all digital HDTV system. Out of 6 systems competing today for FCC, 4 of them are all digital. These 4 systems share many common principles. Important point of these systems are data compression and channel coding. Data compression is important to reduce the number of bits necessary to transmit and to represent the original data. Because compressed data is much more sensitive to noise existing in the transmission channel, compressed data must be protected. This is done with the channel coding. The data compression can be subdivided into spatial data compression and time data transmission. The first refers to the removal or the attenuation of the spatial correlation existing between picture points of the same frame, whereas the second one refers to the redundancy existing from one frame to the next.

The linear transform coding techniques introduced 25 years ago allow to compact the energy of the image signal in two specific areas of a transform domain and then assigning bits to each transform coefficient in a very specific manner compressions around 10 to 1 or 15 to 1. This technique has been improved so that today recommendation from international standardization bodies exist on the label of JPEG (Joint Photographic Experts Group). Basically, the input image is subdivided into blocks of 8x8 pixels. Each block is transformed by DCT. The transform coefficients are quantized according to some specifications and then coded for transmission. The decoder implements the inverse operations to reconstruct the picture. The JPEG scheme is information lossy because of the quantization of transformed coefficients.

To reduce temporal redundancy, it is easy to realize that from one frame to the next. There is no need to code each frame independently from the preceding one. The common principle used is to fully encode the first frame and then code only the changes appearing in the forthcoming frames with respect to the first one. These changes are best described if the objects moving in the scenes are identified. A good approximation of this problem is to trace the motion of small image blocks (for example 8x8 from one frame to the next). The previous frame is showing the corresponding position of each block. One step further is to take into account the previous situation, to predict where the given block will be in a third frame. If a prediction is good, there is no need to transmit additional information. If there is a slight prediction error, then it is sufficient to transmit this small error with a very reduced number of bits, to recover fully the position of this block in the third frame. This technique is known as motion compensation. A better description would be the coding of the prediction error. Namely, using the previous techniques, one can produce a sequence of error images containing the prediction error of block motions from one frame to the next. This is called the displaced frame difference sequence. A common procedure is to view each frame of this new error sequence as an image and apply the spatial decorrelation technique using for example a DCT. Recommendation called MPEG (Moving Picture Experts Group) is issued by the ISO (International Standardization Organization) for video coding of about 1,5 Mb/s.

Among the American systems the Zenith-consortium designed a system which uses the previous principles. It has a very elaborate motion detection scheme, which uses hierarchy so that a coarse motion analysis is made in all cases and then depending on the bit available for transmission, more refined motion analysis is implemented. The spatial redundancy is attenuated using the DCT. The so-called Digicypher system proposed by General Instruments Corporation uses also similar principles. One level motion compensation reduces the time redundancy followed by a DCT for spatial redundancy reduction. Quantized coefficients are then encoded prior to quadrature amplitude modulation QAM. The Advanced Digital HDTV (AD-HDTV) uses the MPEG recommendation adapted to the high definition format. They called it MPEG ++. The originality of this system is the introduction of priorities to the data to be transmitted. Very important data receives high priority and is coded with high security as opposed to low priority data which is less protected for transmission. The modulation used is again the QAM. Finally, among digital HDTV systems, the new system proposed by MIT has to be mentioned. Here, a subband analysis is introduced to reduce the spatial correlation of the data.

3. IMAGE DATA COMPRESSION

The key to reducing the bit rates is image compression. it depends on sending nothing that is not visible to the human eye and on exploiting the inherent redundancy of the video signal. Image compression techniques are concerned with reduction of the number of bits required to store or transmit images without any appreciable loss of information. The degree to which images may be compressed while still allowing satisfactory reproduction after storage or transmission in compressed form is therefore, crucially dependent upon there correlation properties.

Two widely used digital video compression techniques are predictive coding and transform coding. Predictive schemes compress each pixel by quantizing the difference between a predicted value from its actual value. Transform coding, particularly using the DCT has been established as one of the most powerful approaches. images are first separated into square blocks of typical size 8x8. Each of the blocks is DCT transformed, resulting in another 8x8 blocks, whose coefficients are then quantized and coded. Most of the quantized DCT coefficients and up having zero value resulting in high compression. Applying the inverse DCT on the quantized DCT coefficients recovers an approximate version of the original block.

4. MPEG COMPRESSION ALGORITHM

The MPEG compression algorithm achieves compression in three stages: bandwidth reduction, subjectively adapted lossy compression and a final stage of lossless compression. The first stage consists primarily in matching the source resolution to the target bit rate and reducing the chrominance resolution to a subjectively satisfying ratio. The second stage, the compression algorithm itself, removes the spatial and temporal redundancy by means of waveform analysis and subjectively adapted quantization. The third stage losslessly maps the resulting information into a bitstream by way of a syntax combining fixed length and variable length codes.

The elementary concept in MPEG is a picture. A picture corresponds to a non-interlaced video signal and covers one instant in time. The picture rate in MPEG is flexible. The rates of 24,25 and 30 Hz are central to MPEG. Converting video to the MPEG source format is the first phase of the compression algorithm. It involves temporal decimation, horizontal filtering and decimation as well as chrominance decimation.

The MPEG source format uses the same colour space (Y, Cr, Cb) as the CCIR Recommendation 601. Both the luminance and the colour difference signals are represented with 8 bits. Unlike the signals, in CCIR 601, vertical decimation of the chrominance becomes as natural as horizontal decimation. MPEG limits itself to 2:1 decimation of the chrominance in both the horizontal and the vertical dimensions.

4.1. Spatial redundancy reduction

Both still image and prediction error signals have a very high spatial redundancy. There are many the redundancy reduction techniques usable to this effect. Block based techniques are preferred because of the block based nature of the motion compensation process. Transform coding techniques with a combination of visually weighted scalar quantization and run length coding have been preferred. The technique to perform intraframe compression with the DCT has basic similarities in MPEG, JPEG and nx64 kbps and consists of three stages:

- computation and transform coefficients,
- quantization of the transform coefficients, and
- conversion of the transform coefficients into run-amplitude pairs after reorganization of the data in a zig-zag scanning order. The transform coding, quantization as well as run length coding are shown in Fig. 1.

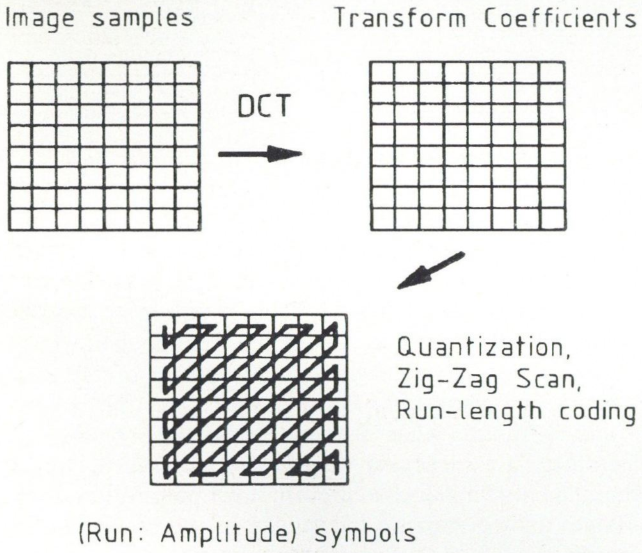


Fig. 1. Transform coding, quantization as well as run-length coding

The DCT has inputs in the range (−255, 255). On the other hand, output signals are in the range (−2048, 2047) providing enough accuracy even for the finest quantizer. The accuracy of the inverse transform is determined according to the CCITT H.261 standard specification [5], [6].

Quantization of the DCT coefficients represents the combination of quantization and run length coding which is responsible for most of the compression. Through quantization, the encoder can match its output to a given bit rate. Finally, adaptive quantization is one of the key tools to achieve visual quality.

Subjective perception of quantization error varies with the frequency. Thus, it is advantageous to use coarser quantizers for the higher frequencies. It is possible to design a particular quantization matrix for an application or even for an individual sequence. A customized matrix can be stored as context together with compressed video.

A masking model can be derived on the basis of smooth area detection, edge detection and texture analysis. The regions of complex textures exhibit the highest degree of masking. A large quantization error might be allowed without perceptible degradations. Psychovisual quantization consists in concentrating the quantization errors where they are less visible and quantizing more accurately in the regions of the picture where the visibility of errors is high.

The signal from intra-coded blocks should be quantized differently from the signal resulting from prediction or interpolation. Intra-coded blocks contain energy in all frequencies and are very likely to produce blocking effects. Prediction error type blocks contain predominantly high frequencies and can be subjected to much coarser quantization. The difference between intra blocks and differentially coded blocks results in the use of two different quantizer structures. Namely, while both quantizers are near uniform, their behavior around zero is different. Quantizers for intra-coded blocks have no dead zone, i.e., the region that gets quantized to the level zero is smaller than a stepsize, while quantizers for non intra blocks have a large dead zone.

Quantized coefficients are ordered along a zig-zag path and runs of zeros are identified. In order to further increase the compression inherent in the DCT, variable length coding is used. A Huffman table for the DCT coefficients is used to code events corresponding to a pair (run, amplitude). Only those codes with a relatively high probability of occurrence are coded with a variable length code. The less likely events are coded with an escape symbol followed by fixed length codes in order to avoid extremely long code words and reduce the cost of implementation.

4.2. Temporal redundancy reduction

Temporal redundancy is typically exploited by using motion compensation to predict each image frame and then compressing the difference between the predicted and actual frame using DCT coding. Motion compensation is an important element of high compression. Video scenes typically contain repeated frames of objects that are essentially unchanged from frame to frame, except for some displacement due to their motion. Motion related coding operations can improve the performance of video compression. Motion estimation is beneficial only as long as it is accurate. Otherwise, it make create severe quality problems. To minimize receiver complexity, motion vectors are typically evaluated at the encoder site from the original signal and then sent as side information to the decoder.

Three types of pictures are considered in MPEG: intra pictures (I), predicted pictures (P) and interpolated pictures for bidirectional prediction (B). In all cases when a picture is coded with respect to a reference model, motion compensation is used to improve the coding efficiency. The relationship between the three picture types is presented in Fig. 2. Intra pictures provide access points for random access, but only with moderate compression. Predicted pictures are coded with reference to a past picture (intra or predicted) and will in general be used as a reference for future predicted pictures. Bidirectional pictures provide the highest amount of compression but require both a past and a future reference for prediction. In addition, bidirectional pictures are never used as reference. As an example, an intra-coded picture is inserted every 8 frames. The ratio of interpolated pictures to intra or predicted pictures is 3 out of 4.

A motion compensated prediction is in particular the basis of most compression algorithms for visual telephony such as CCITT standard H.261. This algorithm assumes that "locally" the current picture can be modeled as a translation of the picture at some previous time. This means that the amplitude and the direction of the displacement need not be the same everywhere in the picture. As for the motion information, it is a part of the necessary information to recover the picture and has to be coded appropriately.

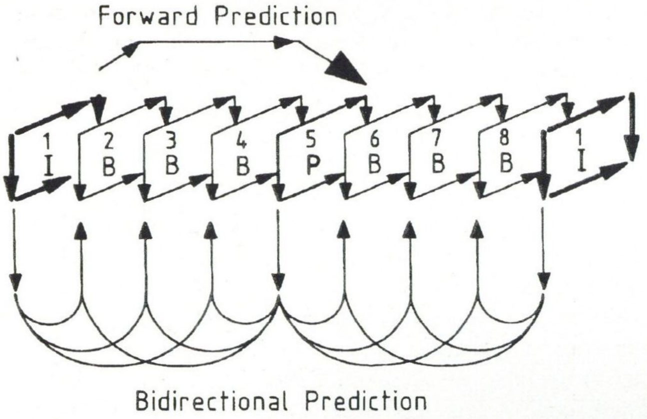


Fig. 2. Relationship between intra, predicted and interpolated pictures

Motion compensated interpolation is a technique that helps satisfy some of the application dependent requirements. It improves random access and reduces the effect of errors, while at the same time it contributes significantly to the image quality. Motion compensated interpolation i.e. bidirectional prediction in MPEG terminology presents a series of advantages. For example, the compression obtained by interpolative coding is very high. Some other advantages of bidirectional, i.e. temporal interpolation include that. a) uncovered areas can be properly predicted from the "future" reference, b) the effect of noise can be decreased by averaging between the past and the future reference pictures, c) there is no error propagation between prediction and coding, d) increasing the number of B-pictures between references decreases the correlation of B-pictures with the references as well as the correlation between the references themselves.

Finally, the motion estimation covers a set of techniques used to extract the motion information from a video sequence. The MPEG specifies how to represent the motion information: one or two motion vector pair 16x16 subblock of the picture depending on the type of motion compensation: forward predicted, backward predicted, average. However, the MPEG does not specifies how such vectors are to be computing. Block matching techniques are likely to be used. In these techniques, the motion vector is obtained by minimizing a cost function increasing the mismatch between a block and each predictor candidate.

4.3. Layered structure

The syntax of a MPEG video bitstream contains 6 layers presented in Table 1.

The MPEG syntax defines a MPEG bitstream as any sequence of binary digits consistent with the syntax [6]. The bitstream must satisfy particular constraints so that the bitstream is decodable with a buffer of an appropriate size. Every bitstream is characterized at a sequence layer by two fields: bit rate and buffer size. The buffer size specifies the minimum necessary to decode the bitstream within the context of the video buffer verifier.

Table 1.

LAYERS IN THE MPEG SYNTAX

number	layer	function
1	sequence layer	random access unit: context
2	group of pictures layer	random access unit: video coding
3	picture layer	primary coding unit
4	slice layer	resynchronization unit
5	macroblock layer	motion compensation unit
6	block layer	DCT unit

4.4. Error concealment strategy

In MPEG compression, video frames to be coded are formatted in groups of pictures consisting of a sequence intra-coded (I), predictive coded (P) and bidirectionally predictive coded (B) frames. The structure of MPEG implies that if an error occurs within I frame data, it will propagate through ail frames in the group of pictures. Similarly, an error in a P-frame will effect the related P- and B-frames, while B-frame errors will be isolated. Therefore, it is desirable to develop sophisticated error concealment techniques to prevent error propagation from I-frames and to improve the quality of the reconstructed pictures.

Two approaches have been used for I-frame concealment: temporal replacement and spatial interpolation. Temporal replacement can provide high resolution image data as the substitute to the lost data. However, in motion areas a significant difference might exist between the current intra-coded frame and the previously decoded frame. In this case, temporal replacement will produce large distortion unless some motion-based process can be applied at the decoder. This type of processing is not always available since it is a computationally demanding task. In contrast, a spatial interpolation approach synthesizes the lost data from the adjacent blocks in the same frame. In spatial interpolation, the intra-frame redundancy between blocks is exploited. A potential problem of blurring remains due to insufficient high order DCT coefficients for active areas. To address this problem, a two stage adaptive error concealment technique has been developed and evaluated. Namely, error concealment process is performed in two steps: codeword or frequency domain concealment and error concealment in the video decoder. The algorithm based on temporal replacement gives good results in still or motion areas, while causes large shearing in the fast moving areas. In the adaptive algorithm, the first concept is to apply decisions to smaller regions of the picture, using concealment modes on a block by block basis. This reduces shearing and block artefacts in the concealed areas. Since I-frame concealment is both the most difficult and the most subjectively important, an adaptive concealment strategy has to be applied. This concealment uses some simple measures obtainable at the decoder in order to arbitrate between spatially based concealment and temporally based concealment. For P- and B-frame, major concealment strategy is to resume the lost motion information and then to use temporal replenishment with motion compensation.

Error concealment algorithms are essential for many practical video transmission scenarios characterized by occasional data loss due to thermal noise, channel impairments, network congestion and so on. Scenarios of current importance include terrestrial HDTV, teleconferencing via packet networks, TV/HDTV over fiber optic asynchronous transfer mode systems, etc.

5. HDTV CODING

Many HDTV DCT coding algorithms have been proposed when CCIR has selected the DCT coding algorithm as the standard TV coding algorithm [8]. Concurrently, a new concept of coding algorithm is subband coding SBC. The advantages of SBC scheme are: a) each band can select the optimum coding algorithm, b) quantization noise generated in a particular band is not allowed to spread to other bands, c) parallel processing can be applied to each band. Based on the SBC algorithm, many HDTV subband coding algorithms are proposed in the open literature [9], [10], [11], [12].

The subband approach consists of splitting a signal into bands, using different techniques to code the bands, transmitting the code in one or more channels and performing the reverse processing in the receiver mainly by band interpolation. The compatibility between different standards, mainly between TV and the future HDTV is an important requirement to coding techniques and can be flexibly managed in a subband systems. Furthermore, there are additional requirements related to terrestrial digital transmission, among which are graceful degradation and error resilience in the case of a low signal amplitude at the receiver. This feature also can be accommodated in a subband system quite efficiently.

Based on the SBC algorithm, two HDTV subband coding algorithms are proposed, i.e.

- subband DPCM for HDTV distribution service, and
- subband DCT for HDTV primary distribution service.

In subband DPCM for HDTV contribution service, the subband DPCM coding scheme can encode original 1,2 Gb/s HDTV signals without any distortion. The coder configuration is shown in Fig. 3. Input signals are decomposed into four bands in horizontal and vertical directions. The architecture of the short kernel subband filtering reduces the entropy of the subband signals, while maintaining the original picture quality. Further entropy reduction is achieved by using DPCM coding [11]. Computer simulation results show that it is possible to apply the subband coding scheme to distortion-free video transmission at 600 Mb/s in a Synchronous Transfer Mode STM network. In the subband DCT for HDTV primary distribution service algorithm, the quadrature mirror filters QMF in the first stage decompose the input signal into two bands in the horizontal direction. On the other hand, the second stage filters decompose the two bands into four bands (LL, LH, HL, HH) in the vertical direction. The block diagram of the HDTV adaptive subband DCT coder is shown in Fig. 4.

Key:

MUX: Multiplexer
VLC: Variable Length Coding
Hp: High-pass Filter and Down Sampling
Lp: Low-pass Filter and Down Sampling
DPCM: Differential Pulse-Code Modulation

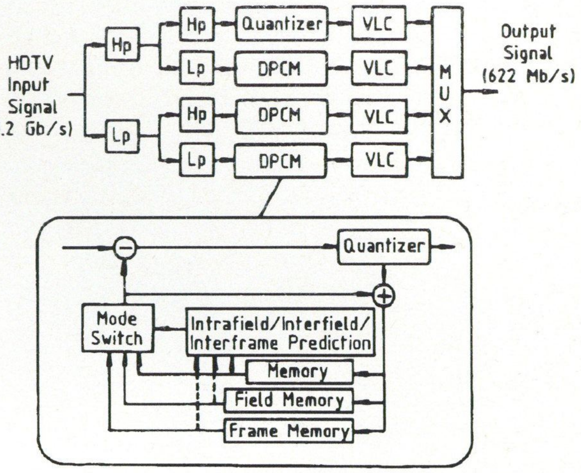
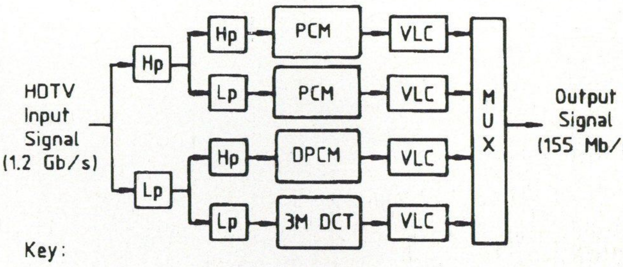


Fig. 3. Subband DPCM for HDTV coding



Key:

DPCM: Differential Pulse-Code Modulation
MUX: Multiplexer
Hp: High-pass Filter and Down Sampling
Lp: Low-pass Filter and Down Sampling
VLC: Variable Length Coding
3M DCT: 3 Mode Discrete Cosine Transform
PCM: Pulse Code Modulation

Fig. 4. HDTV adaptive subband DT coding

The HDTV original sample picture "Fashion show" and each decomposed signal of the sample picture are presented in Fig. 5. All picture element values except for LL, are shifted to a predefined level for the convenience of the observer. Adaptive DCT is applied to the LL band. To maximize the bit rate reduction efficiency in the LL band signal, adaptively selection of the DCT coding according to the intra-field, inter-field and motion compensated interframe signals was proposed [13].

To further reduce the information bit rate, nonuniform length and run length codings are applied to the quantized signal. In the nonuniform lengths coding stage, a higher level signal is assigned a longer length code. In addition, run length coding reduces the length of consecutive zero level signals. Since the LH signal contains the lower band signal in the horizontal direction and the higher band signal in the vertical direction, its pixel to pixel correlation is high. Therefore, the proposed algorithm applies one dimensional DPCM coding to signals on the same horizontal line of the LH band. Since the HL and HH signals show little correlation among pixels, those two band signals are coded by pulse code modulation PCM having a dead zone. The performance of the proposed coding algorithm was evaluated in terms of signal to noise ratio SNR as well as bit/pel ratio using a picture coding simulator and the HDTV sample picture "Fashion show". The average SNR was 38 dB, while the average entropy was 1,08 bit per pixel (bpp). This corresponds to an information bit rate of about 120 Mb/s.

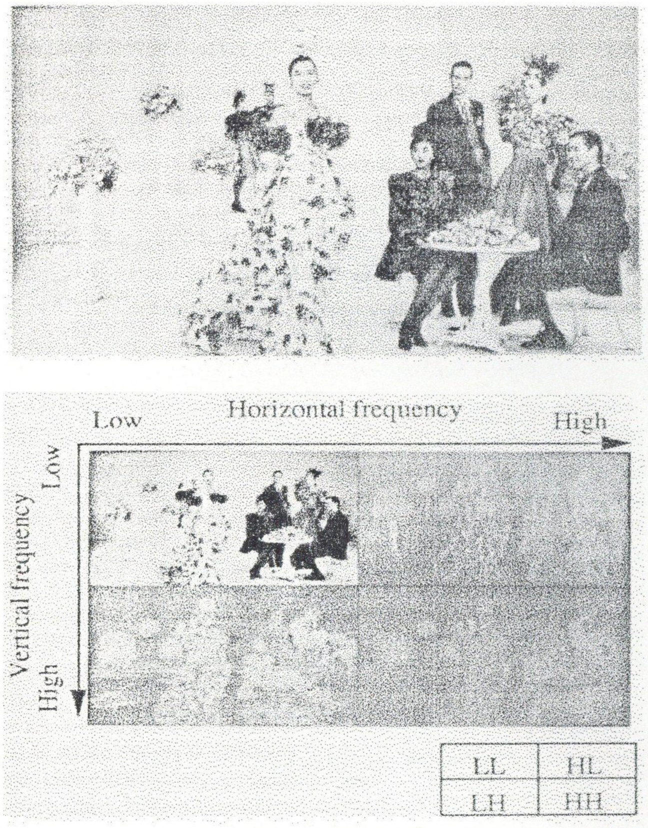


Fig. 5. HDTV decomposed signals ("Fashion show")

Since rapid expansion of optical fiber lines and network digitization are currently being carried out, and construction of BISDN is anticipated, HDTV digital transmission will be dominant in the near future. With BISDN in mind, CCITT has recommended a unified new synchronous digital hierarchy SDH. In SDH, the synchronous transport module-1, STM-1, rate of 155,52 Mb/s is a good candidate for HDTV transmission.

Recently, an HDTV bit rate reduction codec was developed aimed at the transmission of a HDTV signal for distribution use. This codec can perform the coding of a 1125 lines/60 Hz HDTV video signal accompanied with 4-channel sound signal at about 133 Mb/s and transmit it at the STM-1 rate of 155,52 Mb/s in the synchronous digital hierarchy. The sampling frequencies are selected considering the sample relation with the studio standard as well as the required bandwidth for HDTV distribution. A hybrid DPCM/DCT coding scheme is employed as a bit rate reduction algorithm, where intra-field 8x8 DCT is first performed and then inter-frame DPCM is carried out in the DCT coefficient domain. An adaptive intra-field/inter-frame mode selection is performed only for low-frequency DCT coefficients. The intra-field mode is always used for high frequency coefficients because the inter-frame correlation of the high frequency coefficient is fairly weak. Computer simulation experiments were carried out to examine the performance of this coding scheme. From these experiments, it was confirmed that this codec can transmit HDTV at the STM-1 rate with picture quality satisfactory for distribution use.

6. BIT ALLOCATION FOR SUBBAND COMPRESSION OF HDTV

A major component coding schemes is allocating the bits for encoding various subbands. Some of the papers reported in the SBC of images involved the decision of the bit allocations on a perceptual basis [10], [14]. In [14], [15], [16] M equisized subbands were generated by QMF's and encoded on an objective basis. For sufficiently bandlimiting filters, it has been found that the overall distortion D in the reconstructed image can be written as the sum of separate distortions, i.e. reconstruction error variances, various channels. We optimize the bit allocations r_k to have

$$\min D = \sum_k \sigma_{rk}^2$$

such that

$$1/M \sum_k r_k \leq R, \quad r_k \geq 0 \qquad (2)$$

where the subscripts are indexes for the subbands. This assumption is valid for QMF's which are half band symmetric filters whose low-pass and high-pass filters are mirror images of each other. On the other side, the low-pass and high-pass perfect reconstruction filters are not mirror images of each other. When using such nonsymmetric filters, the quantization noise in various subbands will not be equally weighted in the reconstruction, even if the subbands are of same size and bandwidth. Thus, in order to make a valid bit allocation, we introduce the scaling factors ω_k in (1) and we have that

$$\min D = \sum_k \omega_k \sigma_{rk}^2$$

such that

$$1/M \sum r_k \leq R, \quad r_k \geq 0 \qquad (2)$$

where ω_k takes into consideration the filter set being used in the generation of subbands. The weighted factor ω_k represents the energy contribution of the k-th channel in the reconstruction when unit variance noise is input to that channel in the synthesis filter bank.

In the case of M subbands of equal bandwidth, each subband has been subsampled by sqrt (M) in each dimension. Assuming error free transmission and the use of PCM or DPCM coding of individual subbands and assuming a constant quantizer performance factor, equation (2) can be presented in the form

$$\min D = \sum_{k=1}^{M} \varepsilon_q^2 2^{-2r}k\ \omega_k \sigma_k^2$$

such that

$$a/M \sum_{k=1}^{M} r_k \leq R, \quad r_k \geq 0 \qquad (3)$$

where the ω_k's are the weighting factors, while σ_k^2 is the variance of subband k if PCM is used and the variance of the prediction error if DPCM is used to code the subband. Using Lagrangian multipliers to minimize the reconstruction error variance with the overall bit rate constraint given in (3), we have the approximately optimal bit assignment in the form

$$r_{k\ \mathrm{opt}} = R + \frac{1}{2}\log_2 \frac{\sigma_k^2 \omega_k}{\left[\prod_{j=1}^{M} \sigma_j^2 \omega_j\right]^{1/M}} \,. \qquad (4)$$

In order to avoid negative bit allocation, a quickly converging iterative algorithm is used. By allocating bits based on the mean squared error criterion, the longer filters gave the best quantitative performance. The background noise is the most objectionable distortion and the bit rate of 1,25 bpp is needed to reduce it to an acceptable level

7. ADVANCED DIGITAL HDTV SYSTEM COMPRESSION

Advanced Digital HDTV (AD-HDTV) is a layered digital system that consists of:

- MPEG ++ video compression
- MUSICAM audio compression
- prioritized data transport format
- spectrally shaped QAM.

AD-HDTV's layered digital system approach provides interoperability at all of its layers. This includes interoperability among picture and sound formats, compressed video and audio data streams, various packet formats and a variety of transmission media.

AD-HDTV's MPEG ++ compression simultaneously provides high quality HDTV pictures and forms the basis of AD-HDTV's reliable and robust performance as a simulcast system. MPEG video compression was selected as the basis for AD-HDTV, based on its outstanding picture quality and wide acceptance. It also has important capabilities to edit and search compressed video on digital storage media.

Luminance and colour difference blocks covering the same spatial region in a picture are organized together in a macroblock. It comprises for 4 luminance (Y_0-Y_3) blocks and two colour difference blocks (Cr and Cb) as shown in Fig. 6. A macroblock is the unit of motion compensation and adaptive quantization. An integer number of consecutive macroblocks is organized to form a slice. A slice represents the boundary within which differential coding of macroblocks parameters (motion vectors, DC coefficients, etc.) is performed. Each slice has its own header information and can be independent of other slices. An entire picture is thus composed of a cell of slices.

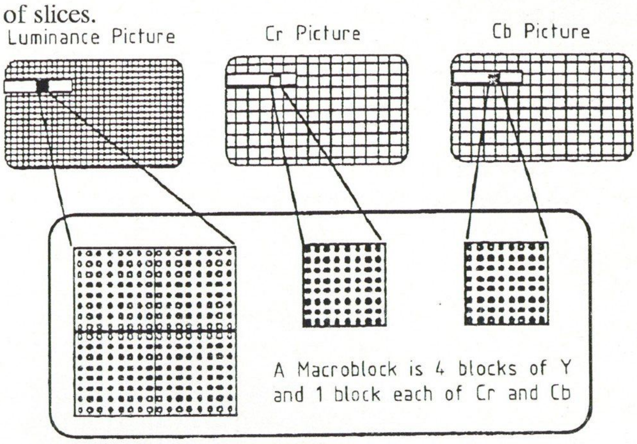


Fig. 6. Entire picture composed of a set of slices

AD-HDTV's MPEG ++ approach is fully compliant with MPEG, but adds important robustness that is essential to survive the transmission bit errors that will occur during simulcasting. To overcome the serious artifacts that typically result from errors occurring in critical bits, MPEG ++ prioritizes a MPEG codeword stream, dividing it into two separate video data streams: a) high priority data that is essential to make viewable pictures and b) standard priority data that is additionally required for high quality HDTV pictures. Specifically, MPEG ++ is a post processing layer that splits a standard MPEG codeword stream into two separate codeword streams. The high priority codeword subset constitutes a viewable picture and is the basis for providing robustness under heavily impaired transmission conditions.

In MPEG compression, blocks of pixels from 1 or more frames are transformed into a set of video data structures, including control bits, motion vectors and DCT coefficients. These data structures are coded to achieve a compressed representation of video. MPEG ++ prioritization identifies the important codewords needed to make viewable pictures which are transmitted as a high priority bit stream. The remaining codewords that are additionally required to make a full quality HDTV picture are transmitted as a standard priority bit stream. AD-HDTV delivers high quality HDTV services to its coverage areas, which is defined by the reception of both standard priority and high priority data. With two separate data streams, additional reliability and robustness is provided by transmitting the MPEG ++ high priority video data at a higher power level-ensuring the reception of viewable pictures under virtually all conditions.

8. CONCLUSION

To create highly effected multimedia communication services high bit rate reduction methods are being studied by the MPEG of the ISO with the aim of creating a useful standards. These coding methods can be applied to HDTV signals. Many HDTV digital coding methods have also been studied and it is shown that it is possible to code HDTV signals to achieve a coding bit rate of 15-30 Mb/s. When highly efficient bit reduction techniques are employed various HDTV communication services can be provided cheaply to the customers.

By using the subband coding concept, that has the potential to bridge NTSC and HDTV, the following basic HDTV coding algorithms can be recommended:

- subband DPCM for HDTV contribution service with a transmission bit rate of about 600 Mb/s
- subband DCT for HDTV primary distribution service with a transmission bit rate of about 120 Mb/s.

Some new subband schemes for HDTV secondary distribution service are now being studied. Computer simulation results show that the SNR is about 38 dB when the average entropy is 1 bpp.

AD-HDTV provides important interoperability features at every layer of its architectures. It also provides a broad scope of features and services, allowing opportunities to tailor services to specific markets. AD-HDTV MPEG ++ compression simultaneously provides high quality HDTV pictures and forms the basis of AD-HDTV's reliable and robust performance.

REFERENCES

[1] M. Kunt: "HDTV", Plenary Lecture delivered at the 1991, *IEEE International Symposium on Information Theory*, Budapest, Hungary, June 1991.

[2] DG XIII Telecommunications, Information Market and Exploitation of Research: "Introduction of High Definition Television: Council accepts action plan", *XIII Magazine News Review*, Issue No. 2/93, No. 11, p. 3.

[3] K. R. Rao, P. Yip: Discrete cosine transform algorithms, advantages, applications, Academic Press, Inc. (1990)

[4] Y. Ninomiya et al.: "HDTV", NHK Technical Monograph, No. 32, June 1982.

[5] CCITT Recommendation H.261: "Video codec for audio visual services at px64kb/s ", 1990.

[6] Committee Draft of Standard ISO 11172: "Coding of Moving Pictures and Associated Audio", ISO/MPEG 90/176, December 1990.

[7] H. Sun, J. Zdepski: "Adaptive error concealment algorithm for MPEG compressed video", *Proceedings SPIE*: Visual communications and image processing'92, Vol. 1818, 18-20 November 1992, Boston, pp. 814-824.

[8] K. Sawada: "Coding and transmission of high definition television-technology trends in Japan", *Proceedings ICC'90*, No. 320. 3, April 1990.

[9] J. W. Woods, D. O. O'Neil: Subband Coding of Images. *IEEE Trans. on ASSP*, Vol. 34, No. 5 (October 1986), pp. 1278-1288.

[10] H. Gharavi, A. Tabatabai: Subband Coding of Monochrome and Color Images. *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 2 (February 1988), pp. 207-214.

[11] K. Irie, R. Kishimoto: "A study on perfect reconstructive subband coding", *IEEE Trans. on CAS Videotechn.*, Special issue on Signal Processing for Advanced Television, March 1991.

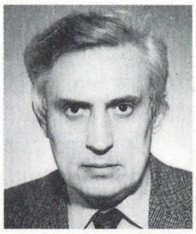
[12] R. Ansari, D. Le Gall: "Advanced television coding", in Subband Image Coding, edited by J. W. Woods, Kluwer Academic Publishers (1991), pp. 280-287.

[13] Y. Okumura, K. Irie, R. Kishimoto: "High quality transmission system design for HDTV signals", *Proceedings ICC'90*, No. 325. 2, April 1990.

[14] T. Kronander: "Some aspects of perception based image coding", Ph.D. dissertation, Linkoping Univ., Linkoping, Sweden 1989.

[15] N. Tanabe, N. Farvaradin: "Subband image coding using entropy coded quantization over noisy channels", Technical Rep., UMIACS-TR-89-86, University of Maryland, August 1989.

[16] Westernik, P. H.: "Subband Coding of Images ", Ph.D. dissertation, Delft University Technology, Delft, The Netherlands, 1989.



Zoran S. Bojković received the B.S., M.S., and Ph.D. degrees, all in electrical engineering, from the University of Belgrade in 1964, 1974 and 1978, respectively. In 1965 he joined the ISKRA Company, Kranj in the Republic of Slovenia working as a Project telecommunication engineer. Since 1969 he has been with the University of Belgrade where he is currently a Professor of Electrical Engineering. In 1986 he was a visiting Professor at Stanford University, Stanford, USA, whereas in 1993 he was a visiting Professor at the University of Texas at Arlington, USA. He has taught a number of courses in Electrical Technology, Telecommunication Systems and Networks planning, design, control and maintenance, Image Processing and Image Compression. He published many papers and books, some of them in four editions. His interest are in the areas of image coding as well as image data compression with applications in Broad band Integrated Services Digital Networks. He is a member of IEEE and EURASIP.

ITERATED FUNCTION SYSTEMS FOR IMAGE AND VIDEO CODING*

M. H. HAYES

GEORGIA TECH LORRAINE
2-3 RUE MARCONI
57070 METZ, FRANCE

Iterated Function Systems (IFS's) are receiving a lot of attention in the literature as a new technique for signal and image coding. Much of this interest stems from the fact that an IFS is simple in form and yet capable of representing complicated signals, many of which closely resemble those found in nature. In this paper, we present a tutorial overview of the basic ideas involved in using iterated function systems for image and describe some of the current methods and approaches that are currently being used for IFS coding of video sequences.

1. INTRODUCTION

Signal compression is an important problem in applications requiring the transmission of high bandwidth signals over a low bandwidth channel as well as in applications requiring efficient storage of large amounts of information. In order to develop efficient (high compression) coders with acceptable signal quality, it is important to exploit the properties of the signal that is to be compressed and to take into account the limitations and characteristics of the observer that will be processing the decoded signal. For example, in many applications, it is possible to represent a complicated signal parametrically, based on the physics of the signal generation process. An acoustic tube model for the vocal tract, for example, implies an all-pole model for speech. When the number of bits necessary to store the parameters related to the model is less than the number of bits required to represent the signal, then the signal may be compressed. In other applications, a signal may be characterized by repetitions or patterns that may be compressed by encoding the pattern and defining the repetition pattern. One form of repetition or redundancy is represented in the property of self similarity in which one part of the signal may be closely approximated by a suitable transformation, such as scaling, of another part of the signal. It has been asserted that most "natural" signals possess the property of self-similarity [1], [2]. Self-similar signals may be compressed in a manner similar to repetitive signals.

In this paper, we explore the use of an iterated function system (IFS) for image and video coding. These coders are designed to exploit the self-similarity property that has often been ascribed to "natural" images. Due to the self-similarity in a single image and the similarity between successive frames in an image sequence, iterated function systems may also provide the basis for an efficient video coding system. In developing the basic design of an IFS image coding system, we begin with a review of the theory of fixed points, contraction mappings, and the definition of an IFS in Section 2. In addition, we describe how this theory has been applied to the problem of iterative signal reconstruction. In Section 3 we describe how an IFS may be used to encode an image. Beginning with a block encoding system proposed by Jacquin, we proceed to discuss some modifications and extensions of this method to improve the coder efficiency and speed. In Section 4 we look at an orthonormal basis approach to IFS image coding and in Section 5 we describe the encoding and decoding process. Finally, in Section 6, we describe how an IFS coder may be extended and generalized to encode video sequences.

2. BACKGROUND

The theory of iterated function systems has its roots in the theory of fixed points and contraction mappings. As we will soon see, this theory has also played an important role in solving signal reconstruction problems using iterative techniques. We begin, therefore, with an overview of fixed points and the Banach fixed point theorem.

2.1. Fixed Points

Let $\mathcal{U}$ be a complete metric space with a distance metric d and let f be a function or mapping from a subset, A, of $\mathcal{U}$ into $\mathcal{U}$. If the mapping f has a point $x* \in A$ which is invariant under f, i.e., $f(x^*) = x^*$, then x^* is called a *fixed point* of f. A mapping may have any number of fixed points. For example, the mapping $f(x) = -x$ from the real numbers, R, into R has a unique fixed point, $x^* = 0$. On the other hand, every point in R is a fixed point of the mapping $f(x) = x$, whereas the map $f(x) = x + 1$ has no fixed points. With the appropriate set of constraints on f, however, it is possible to insure that f has a unique fixed point. One such constraint is that f be a contraction mapping which is defined below

Definition: If A is a subset of $\mathcal{U}$ and if f maps A into itself and if there is a constant σ, with $0 < \sigma < 1$ such

$$d[f(x), f(y))] \leq \sigma d[x, y]$$

then f is called a *contraction mapping.*

A contraction mapping is characterized by the property that it brings points closer together. For example, let x^* be a fixed point of f so that $f(x^*) = x^*$. It then follows from the definition above that

$$d[f(x), x^*] \leq \sigma d[x, x^*]. \tag{1}$$

Therefore, the effect of applying f to an arbitrary point $x \in \mathcal{U}$ is to bring it closer to the fixed point x^*. In fact,

* This work was supported by the Joint Services Electronics Program, Grant No. DAAH-0493G0027.

iterating by successively applying f to x we find that, after k iterations,**

$$d[f^{(k)}(x), x^*] \leq \sigma^k d[x, x^*]. \qquad (2)$$

Since $\sigma < 1$, as $k \rightarrow \infty$ the distance between $x_k = f^{(k)}(x)$ and x^* goes to zero

$$\lim_{k \rightarrow \infty} d[f^{(k)}(x), x^*] = 0. \qquad (3)$$

An important property of contraction mapping is that they have a unique fixed point. To see this, suppose that there were two fixed points, ${x^*}_1$, ${x^*}_2$. It would then follow that

$$d[{x^*}_1, {x^*}_2] = d[f({x^*}_1), f({x^*}_2)] \leq \sigma d[{x^*}_1, {x^*}_2]. \quad (4)$$

Since $\sigma < 1$, this will be true only if ${x^*}_1 = {x^*}_2$. These results are summarized in the *Fixed Point Theorem* below [3].

Fixed Point Theorem: If f is a contraction mapping with a contraction factor σ on a closed subset A of a complete metric space $\mathcal{U}$, then there is a unique fixed point $x^* \in A$. Furthermore, any sequence of iterates, $x_k = f^{(k)}(x_0)$ will converge to x^* with

$$d[x_k, x^*] \leq \sigma^k d[x_0, x^*]. \qquad (5)$$

As an example, consider the function

$$f(x) = \frac{1}{3}x + \frac{2}{3}. \qquad (6)$$

Clearly, f is a contraction mapping with a contraction factor $\sigma = 1/3$. Therefore, it follows from the fixed point theorem that f has a unique fixed point. To find the fixed point, we solve the fixed point equation, $f(x^*) = x^*$, and find that $x^* = 1$.

A useful inequality that may be derived from Eq. (6) is as follows. From the triangle inequality, which must hold for any metric d, we have

$$\begin{aligned} d[x_0, x^*] &\leq d[x_0, f(x_0)] + d[f(x_0), x^*] \\ &\leq d[x_0, f(x_0)] + \sigma d[x_0, x^*]. \qquad (7) \end{aligned}$$

Therefore,

$$d[x_0, x^*] \leq \frac{1}{1-\sigma} d[x_0, f(x_0)]. \qquad (8)$$

This inequality is useful since it allows one to place a bound on the distance between x_0 and the fixed point x^* in terms of the distance between x_0 and $f(x_0)$ without having to know exactly what the fixed point is. As we will see, this inequality is a special case of the Collage Theorem discussed below [2].

The Fixed Point Theorem has played an important role in the development and theory of iterative signal reconstruction algorithms using the *method of successive approximations*. In fact, as we will soon discover, an interesting similarity exists between iterative reconstruction algorithms and IFS signal modeling. The two problems, in fact, may be thought of as dual of each other. In order to understand this duality, we present a brief overview of the signal reconstruction problem. To make the discussion general, signals will be represented as symbols without any arguments, such as x and y, so that they may be considered to be one-dimensional signals such as speech, two-dimensional signals such as images, or higher-dimensional signals such as video sequences.

The problem of signal reconstruction is concerned with the recovery of a signal x from a distorted observation,

$$y = D(x). \qquad (9)$$

The distortion, represented by the operator D, may be nonlinear and, quite possibly, noninvertible. Examples of important signal reconstruction problems that may be cast into the framework of Eq. (9) include deconvolution, phase-retrieval, and band-limited extrapolation [6], [7]. An approach that has been studied extensively for estimating x from y is to use the observation Eq. (9) to set up the *fixed point equation*

$$x = x + \lambda[y - D(x)] = F(x), \qquad (10)$$

where λ is a parameter that is chosen to optimize some criterion [7]. The signal reconstruction problem is thus reformulated as one of finding the fixed point of the operator F in Eq. (10).

In addition to the observation equation in Eq. (9), the signal that is to recovered is often known to satisfy some constraints such as positivity (image processing), and finite support or duration in either the time or frequency domain (time-limited or band-limited signals). Representing such a constraint in terms of a constraint operator, C, the reconstruction problem now becomes one of finding the solution to a pair of equations

$$\begin{aligned} y &= D(x) \\ x &= C(x). \qquad (11) \end{aligned}$$

With a constraint operator C, the fixed point equation becomes

$$x = C(x) + \lambda[y - DC(x)] = F(x). \qquad (12)$$

If the operator F is a contraction mapping then, for any initial starting point or initial guess for x, denoted by x_0, the sequence of iterates formed by applying the method of successive approximations,

$$x - k = F(x_{k-1}) = F^{(k)}(x_0) \qquad (13)$$

will converge to the unique fixed point of F. Thus, a signal x is mapped by D to an observation y. A fixed point equation is then established, using (12), and a sequence of iterates x_k is formed to produce an estimated of x.

In solving the *signal reconstruction inverse problem* by means of the iterative algorithm (13), three important issues arise. The first concerns the conditions under which the sequence x_k converges to a fixed point of F. Whether or not x_k converges to x^* depends upon many factors including the properties of the function F, the choice of the initial condition x_0, and whether or not F has a unique fixed point. The second issue concerns the relationship between the fixed points of F and the solutions to Eq. (11). It is clear, for example, that any solution to Eq. (11) will also solve the fixed point Eq. (12). However, it is not necessarily true that all fixed points of F will necessarily satisfy (11). The fixed point iteration, therefore, may produce solutions that do not satisfy (11).

** The notation $f^{(k)}(x)$ means the k^{th} iterate of x by the function f. For example, $f^{(3)}(x) = f(f(f(x)))$.

A variety of interesting questions also arise when $D(x)$ is singular (noninvertible). For example, if x_k converges to a fixed point then, since the solution to (11) is not unique, how close is the fixed point to the desired solution x? Finally, the third issue concerns the effects of errors on the solution. Several possible sources of error include truncation of the sequence of iterations, inaccuracies in the measurements of y or in the assumed form of the distortion and constraint operators, and imperfections in the implementation of the iteration.

2.2. Iterated Function Systems

Having introduced the idea of a contraction mapping and having considered some of their features and properties, it is now possible to generalize and define what is meant by an Iterated Function System, or IFS. An IFS is a finite set of contractive mappings, f_i, defined on a complete metric space, $\mathcal{U}$,

$$f_i : \mathcal{U} \to \mathcal{U} \tag{14}$$

for $i = 1, 2, \ldots, P$ with

$$d[f_i(x), f_i(y)] \leq \sigma_i \cdot d[x, y], \tag{15}$$

where $0 \leq \sigma_i < 1$. The transformation formed using all of the maps of the IFS is written as

$$F(B) = \bigcup_{i=1}^{P} f_i(B), \tag{16}$$

where $B \subset \mathcal{U}$. Just as with a single contraction mapping, an IFS has a unique set of points, called the *attractor*, that is invariant under F. The attractor is given by

$$A^* = \lim_{k \to \infty} F^{(k)}(A_0), \tag{17}$$

where A_0 is some subset of the metric space $\mathcal{U}$. As an example, consider the pair of contraction mapping (4)

$$\begin{aligned} f_1(x) &= \frac{1}{3}x + \frac{2}{3} \\ f_2(x) &= \frac{1}{3}x \end{aligned} \tag{18}$$

that is formed by adding one function, $f_2(x) = \frac{1}{3}x$, to that given in Eq. (6). In this case, with A_0 the closed interval [0,1] over the set of real numbers, the attractor is the *Cantor Set*, $\mathcal{C}$. The Cantor set is a *perfect set* (that is, $\mathcal{C}$ is closed and every point of $\mathcal{C}$ is a limit point [5]) that contains an uncountable number of points. Note that, compared with the fixed point of Eq. (6), the fixed point (attractor) of the pair in Eq. (18) is remarkably rich. The Cantor set may also be generated by the successive deletion of middle third open intervals, i.e., $\mathcal{C}$ is the limit of A_k as $k \to \infty$ with [5]

$$\begin{aligned} A_0 &= [0, 1] \\ A_1 &= [0, 1/3] \bigcup [2/3, 1] \\ A_2 &= [0, 1/9] \bigcup [2/9, 3/9] \bigcup [6/9, 7/9] \bigcup [8/9, 1] \\ &\vdots \end{aligned}$$

A theorem of interest related to an IFS is the *Collage Theorem* which is as follows,

Collage Theorem: Let A^* be the attractor of an IFS and let A be any compact set. Then***,

$$d(A, A^*) < \frac{1}{1-\sigma} d(A, F(A)).$$

As with Eq. (8), the importance of this theorem lies in the ability to place a bound on the distance between a set A and the attractor of the IFS by measuring the distance between A and $F(A)$, without having to find the attractor A^*.

At this point, we should be asking the following question: What do contraction mappings and IFS attractors have to do with image compression? The answer lies in the observation that it is possible for a very complex set to be represented as the attractor of a very simple IFS. Recall, for example, that the Cantor set is the attractor of an IFS that consists of only two affine contraction mappings. Another example is the *Sierpinski Gasket* shown in Fig. 1 which is a binary image, defined by the set of points (x, y) that have a value of one. This image is the attractor of the IFS consisting of three maps

$$\begin{aligned} f_1\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ f_2\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ f_3\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} \end{aligned}$$

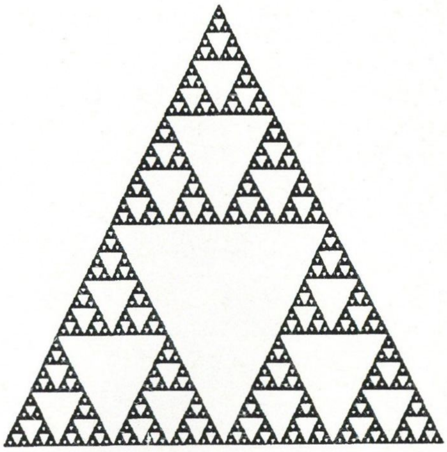


Fig. 1. The Sierpinski gasket that is generated using an IFS consisting of three contractions mappings

Therefore, given an image x that is to be encoded, if we can find a set of contractive maps, f_i, that have an attractor that is close to x, then we may consider using the parameters of the maps as the *code* for generating the image. Given an IFS, it is straightforward to find the attractor. For example, we may simply iterate, forming the sequence of "points"

$$A_F(A_{k-1})$$

*** Here, $d(A, A^*)$ is a measure of the distance between the two set A and A^*. Typically, the distance metric used is the Hausdorf metric [2].

for some initial set $A_0 \subset \mathcal{U}$. The difficult problem is solving the *inverse problem*, i.e., finding the maps that have an attractor that is close to a given image x.

At this point we may illustrate the parallel that exists between IFS coding and iterative signal reconstruction. In IFS coding, we are given an attractor (the signal to be coded) and the problem is to find the contraction mapping f_i associated with the given attractor. In iterative signal reconstruction, on the other hand, the problem is to find contraction mapping that will produce, upon iteration, an unknown attractor (the signal to be reconstructed).

3. IFS IMAGE CODING

In this section we describe the general approach that is used to code an image using an IFS. Although many of the techniques described in this section have also been applied to coding one-dimensional signals such as speech, electrocardiograms, and seismic data [8], [9], [10], the focus of this section will be on image coding. In Sect. 6 we extend this to video coding.

A digital image, $x(m,n)$, is a function of two discrete variables, m and n. It will be convenient, however, to consider an image as a function of two continuous variables, $x(u,v)$, and to represent this image as a 3-D *graph*, i.e., a triplet

$$x = \begin{bmatrix} u \\ v \\ z \end{bmatrix}, \tag{19}$$

where u and v represent the spatial coordinates of a given *pixel* and $z = x(u,v)$ represent the intensity.

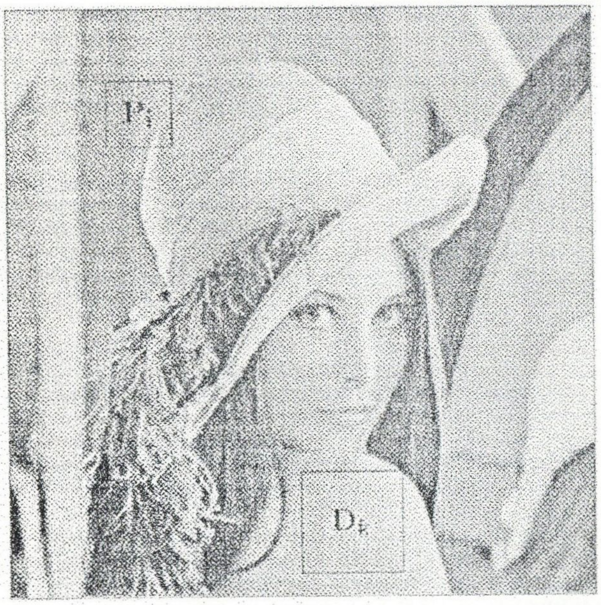


Fig. 2. An original image, $x(m,n)$

As noted in the introduction, an iterated function system attempts to reduce the information rate in an image by exploiting the similarities that exist within different regions of the image. Consider, for example, the image shown in Fig. 2. Since certain regions of the image tend to be similar to other regions, one may consider "modeling" or "approximating" one region of the image, for example P_i, with an appropriately transformed piece extracted from another region such as D_k. This, in effect, is what an iterated function system is designed to accomplish: to represent an image as a collection (*collage*) of suitably transformed versions of itself. Over the set of "real" image, however, one must determine what types of transformations will produce other "real" images. Clearly, these transformations should include translations (delay), rotations and reflections (time reversal), shrinking and stretching (time scale modification), scaling and inversion (gray-scale transformations).

In IFS image coding, a *block coding* approach is typically used. Thus, the image that is to be encoded is initially segmented or partitioned into a set of range (or target) blocks, $P_i(m,n)$. It is then the task of the FS encoder to find a domain block $D_k(m,n)$ and a transformation f_i that produces the best approximation to the given range block $P_i(m,n)$,

$$O_i(m,n) \approx f_i[D_k(m,n)].$$

Therefore, a block IFS image coding system consists of the following steps:

1. Partition the image into a set of range (target) blocks, P_i.
2. Search for an appropriate domain (library) block D_k that is to be used to approximate P_i.
3. Find the transformation f_i which, when applied to D_k, produces the best approximation to P_i.

Clearly, without any restriction on the form of the transformation, f_i, the IFS encoder will be computationally intractable. Therefore, the transformations must be constrained. One constraint that is typically applied in IFS coders is that the maps be affine transformations [11], [12]. In terms of the vector $\mathbf{x}$, an affine map is of the form

$$f_i(\mathbf{x}) = \mathbf{A}_i\mathbf{x} + \mathbf{b}_i,$$

where $\mathbf{A}_i$ is a 3×3 matrix and $\mathbf{b}_i$ is 3×1 *translation* vector. For example, an affine map that has been used for image coding is the following

$$f_i\left(\begin{bmatrix} u \\ v \\ z \end{bmatrix}\right) = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & \alpha_i \end{bmatrix} \begin{bmatrix} u \\ v \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ \beta_i \end{bmatrix}. \tag{20}$$

Note that this mapping is composed of a pair of decoupled transformations. The first,

$$f_{i,1}\left(\begin{bmatrix} u \\ v \end{bmatrix}\right) = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} = \mathbf{T}\begin{bmatrix} u \\ v \end{bmatrix} + \mathbf{t}_i$$

defines how regions (sub-image domains) in the image $\mathbf{x}$ are mapped under f_i. In most cases, the parameters in $\mathbf{T}_i$ are further constrained to consist only of simple transformations such as a spatial scaling (decimation) by a factor of 2,

$$\mathbf{T}_i = \begin{bmatrix} 0,5 & 0 \\ 0 & 0.5 \end{bmatrix},$$

and rotations by 90°, 180°, or 270°

$$\mathbf{T}_i = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}; \ \mathbf{T}_i = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}; \ \mathbf{T}_i = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

The second transformation,

$$f_{i,2}(z) = \alpha_i z + \beta_i$$

corresponds to a linear scaling of the intensities by α_i plus an offset β_i. In the IFS image coding system proposed by Jacquin [12], the domain blocks are constrained to go through a limited set of transformations, including

1. **Spatial contraction:** Typically this is set at 2:1. (In addition to constraining the set of transformations, a spatial contraction of 2:1 simplifies the search for the domain block D_k. Specifically, if the range blocks are $p \times p$ pixels in extent then the domain blocks are constrained to a size of $2p \times 2p$ pixels.)
2. **Isometric transformations:** One of 8 possible transformations including rotation, reflection, circular shift (thus, only 3 bits are required to specify the transformation.)
3. **Grey-scale transformation:** These include scaling, absorption (setting the intensity equal to a constant), translation, and inversion.

Even with these restrictions on the set of allowable transformations, however, it is still necessary for the encoder to find a suitable domain block, D_k, for each range block P_i along with a transformation f_i, that minimizes the distance between P_i and $f_i(D_k)$. Without any restrictions on the *pool* of available domain blocks, the library of possible blocks that need to be searched may be very large and, as a result, the search for the optimum domain block may become prohibitively time-consuming. For example, consider an image $x(m,n)$ that is $N \times N$ pixels in extent. If the domain blocks have a size of $N_d \times N_d$ pixels, then there are a total of $(N - N_d + 1)^2$ possible domain blocks. With $N = 512$ and $N_d = 16$ this implies that, for each range block, there are 247,009 different domain blocks to choose from. As a result, the *domain block pool* is normally pruned before searching for the optimum block. The domain block pool may be trimmed in a number of different ways. For example, the domain pool may be trimmed by sliding a window of size $N_d \times N_d$ pixels across the image in increments of $\delta_h > 1$ pixels in the horizontal direction and $\delta_v > 1$ pixels in the vertical direction. With $\delta_h = \delta_v = N_d$, for example, the domain pool is constrained to non-overlapping blocks which reduces the number of domain blocks by a factor of N_d^2. In the coder of Jacquin [12], the increments δ_h and δ_v were chosen to be equal to either N_p or $N_p/2$, where N_p is the size of the range blocks. To further prune the search, Jacquin also classifies each domain block according to block geometry as either a shade block, an edge block, or a midrange block. Then, for each range block, each element in the domain pool of the same type is analyzed for the optimum transformation (which depends on the domain-pool classification). Another possibility for pruning the domain pool is to constrain the domain blocks that may be used to represent a given range block P_i to one that lies in close proximity to P_i [10], [13]. The motivation for this approach is that pixel values in an image tend to be more correlated when they are in close proximity to each other.

Each of the methods described above for constraining the form of the affine maps in the IFS encoder and the approaches used to reduce the size of the domain pool, are attempts to make the IFS encoder more efficient from a computational point of view. However, each of these approaches affect, in turn, the fidelity of the decoded image. Therefore, approaches have also been considered for increasing the signal-to-noise ratio (SNR) in the decoded image. For example, several different methods of using adaptive block sizes have been investigated [10], [15], [16]. Although a large block size increases the compression since a large block size implies fewer maps and, thus, fewer map parameters that must be stored, a large block size will also, in general, result in a lower quality image. Therefore, by allowing the block sizes to vary and to conform to the geometrical characteristics of the image, increases in the SNR are possible. Another approach that has been considered for improving the quality of the encoded image is to use nonlinear maps and nonlinear addresses [10], [14]. The increased complexity in finding the map parameters and the increased bit rates necessary to represent these mappings, however, do not seem to result in an improved IFS image coding system. In the next section we provide another interpretation of the IFS image coding system which will lead to a more general coding system.

A number of variations and modifications to the IFS coder developed by Jacquin have been proposed and studied. Some of these modifications have been briefly outlined above. We now describe how the affine transformations of the IFS image coder may be viewed as subspace approximations to the range blocks. Specifically, the IFS map parameters are obtained by finding the best approximation of each range block by an element of a two-dimensional subspace that is formed from a *domain vector*, which depends on the domain block pixel values, and a *fixed translation vector* [17].

To develop the subspace approximation approach to IFS encoding, recall that the affine transformation given in Eq. (20) may be viewed as a mapping consisting of two parts. The first, represented by the parameters a_i, b_i, c_i, d_i and f_i, defines a mapping from the set of $N_d \times N_d$ pixels in the domain block $D_k(m,n,)$ to a set of $N_p \times N_p$ pixels, which we denote by the vector $\mathbf{d}_i$. This mapping, from the domain block to $\mathbf{d}_i$, typically consists of spatial contractions and isometric transformations. The second transformation then scales the elements of $\mathbf{d}_i$ (the pixel intensities) by a factor α_i and adds a bias of β_i. Therefore, the affine mapping in Eq. (20) may be written as

$$\hat{\mathbf{p}}_i = \alpha_i \mathbf{d}_i + \beta_i \mathbf{v}_0, \qquad (21)$$

where $\mathbf{v}_0$ is a vector of pixel values, with each value being equal to one. Thus, $\hat{\mathbf{p}}_i$ may be viewed as an approximation of the range block, $\hat{\mathbf{p}}_i$, using a linear combination of two *basis vectors* — the *image dependent* basis vector $\mathbf{d}_i$ and the *values fixed* basis vector $\mathbf{v}_0$. It is important to note, however, that it is the fixed point (attractor) of the transformations defined by the values of α_i and β_i that determine to pixel values of the decoded image. Therefore, $\hat{\mathbf{p}}_i$ should not be considered to be the output of the decoder in representing the sub-image $\mathbf{p}_i$. However, since the collage theorem states that the closer each of the $\hat{\mathbf{p}}_i$'s are to $\mathbf{p}_i$, the closer the attractor will be to the original image, the encoder seeks to minimize the distance $d[\mathbf{p}_i, \hat{\mathbf{p}}_i]$.

Given the approximation in Eq. (21), it is possible to consider using other basis vectors or, perhaps, more than two vectors. For example, Ramstad et. al. [17) proposed using three fixed basis vectors in addition to the image dependent basis vector. Thus, each range block $\mathbf{p}_i$ is approximated using a subspace approximation of the form

$$\hat{\mathbf{p}}_i = \alpha \mathbf{d}_k + \beta_0 \mathbf{v}_0 + \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2,$$

where $\mathbf{v}_0$ is a vector consisting of all ones as before, $\mathbf{v}_1$ is a vector having values that increase linearly in the horizontal (u) direction, and $\mathbf{v}_2$ is a vector that increase linearly in the vertical (v) direction. These three fixed basis vectors define a plane and is equivalent to the affine transformation

$$f_i\left(\begin{bmatrix} u \\ v \\ z \end{bmatrix}\right) = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ \beta_1 & \beta_2 & \alpha \end{bmatrix} \begin{bmatrix} u \\ v \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ \beta_0 \end{bmatrix}.$$

With the IFS encoder thus formulated in terms of a subspace approximation, it is how possible to make the encoding process more efficient by orthonormalizing the basis vectors and finding the optimum set of constants, α, β_0, β_1, β_0 by projections [17].

In addition to increasing the number of fixed basis vectors, $\mathbf{v}_k$, we may also use multiple domain blocks as follows [10], [18], [19]

$$\hat{\mathbf{p}}_i = \sum_{k=1}^{K} \alpha_k \mathbf{d}_k + \beta_0 \mathbf{v}_0 + \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2.$$

Clearly, the use of multiple domain blocks allows for a much richer set of basis vectors. However, the difficulty in using more than one image dependent basis vector compared to using multiple *fixed* basis vectors is that the search required to find the *optimum* set of domain vectors becomes geometrically more time consuming. For example, with a domain pool size of P and two domain vectors, $\mathbf{d}_1$ and $\mathbf{d}_2$, the encoder must find the best *pair* of domain vectors out of the domain pool. Therefore, the search time becomes proportional to P^2 instead of P in the case of a single domain block. In the following section we present a new type of IFS image coder that uses an orthonormal basis to alleviate this problem.

4. AN ORTHONORMAL BASIS APPROACH

Each of the previous approaches to IFS image coding have been search-based, requiring that a set of domain blocks be evaluated for each range block. With the orthonormal basis approach presented in this section, a set of orthonormal basis vectors are created by the Gram-Schmidt algorithm and the range blocks are coded by projecting the blocks onto this basis. The advantage of using an orthonormal basis is two-fold. First, only one search is required for a given set of domain blocks. Thus, once a suitable collection of domain blocks have been selected, encoding the range blocks is based on this fixed subset of the domain blocks. The second advantage is that the encoding process is relatively fast since the encoding may be performed by simply projecting the range block onto the orthonormal basis. As with any compression technique, we are concerned with reducing the dimensionality of the data to be stored. Thus we wish to find a smaller subspace in which to accurately represent each range block. The goal in determining the orthonormal basis will be to create a basis which allows each range block to be accurately represented whith a minimum number of the basis vectors. By reducing the dimensionality of each of the range blocks, compression is achieved.

Beginning with the three fixed vectors, $\mathbf{v}_0$, $\mathbf{v}_1$, and $\mathbf{v}_2$, suppose that we add Ns domain vectors, $\mathbf{d}_i$, $i = 1, \ldots, N_S$, thus forming a set of $D = N_S + 3$ vectors. For simplicity, we will assume that these vectors are linearly independent. With $N \times N$ range blocks, if $D = N^2$ then these vectors form a a basis and any range block may be represented as a linear combination of these basis vectors,

$$\mathbf{p}_i = \beta_{0,i}\mathbf{v}_0 + \beta_{1,i}\mathbf{v}_1 + \beta_{2,i}\mathbf{v}_2 + \sum_{k=3}^{N_S-1} \alpha_{k,i}\mathbf{d}_k = \mathbf{B}\mathbf{a}_i, \tag{22}$$

where

$$B = [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{d}_1, \ldots, \mathbf{d}_{N_s}] \tag{23}$$

is a matrix containing the basis vectors and $\mathbf{a}_i$ is a vector containing the weights $\alpha_{k,i}$ and $\beta_{k,i}$. In order to facilitate the computation of the weight vector for a given range vector $\mathbf{p}_i$, we may orthonormalize the basis using the Gram-Schmidt algorithm. With the resulting orthonormal basis vectors placed in the columns of a matrix $\mathbf{Q}$, the representation for a given range block $\mathbf{p}_i$ becomes

$$\mathbf{p}_i = \sum_{k=0}^{N_S-1} w_{k,i}\mathbf{p}_k = \mathbf{Q}, \mathbf{w}_i. \tag{24}$$

Once the orthonormal basis has been constructed, the coding process is straightforward. Specifically, Eq. (24) defines the representation of a given range vector, $\mathbf{p}_i$, in terms of the weight vector, $\mathbf{w}_i$, and since $\mathbf{Q}$ is orthonormal, the weight vector may be determined as follows

$$\mathbf{w}_i = \mathbf{Q}^T \mathbf{p}_i. \tag{25}$$

The only remaining task to implement the coder is an algorithm to select the Ns domain vectors. The characteristics that these vectors should have are:

- They should produce a contraction mapping in the IFS,
- They should be as nearly orthogonal as possible, and
- They should allow, each range vector to be represented with as few vectors as possible

Several methods have been developed based on these three criteria and are discussed below.

4.1. Covariance Method

Rather than searching through the relatively large set of domain vectors to find the appropriate set of basis vectors, $\mathbf{d}_i$, in the first approach, which we call the *covariance method*, the range vectors are first analyzed to determine the optimal basis vector directions. Then, the domain vectors are searched to find the vectors that are most closely aligned in each of these directions. This approach has two advantages: first we are primarily interested in representing the range vectors, thus the directions for the basis vectors should be based on the range vectors themselves; secondly, there are far fewer range vectors than domain vectors and the computational task of determining the basis vectors is reduced.

The covariance method begins by projecting all of the range vectors onto the subspace, S^0, that is orthogonal to the subspace spanned by the *a priori* vectors, $\mathbf{v}_0$, $\mathbf{v}_1$, and $\mathbf{v}_2$. At the k^{th} iteration, denoting the i^{th} projected range vector by $\mathbf{s}_i^k$, the next basis vector direction is chosen to be the vector, $\mathbf{s}_l^k$ that is most highly correlated with all of the other vectors, $\mathbf{s}_i^k$. Thus the vector, $\mathbf{s}_l^k$, that maximizes

$$C_i = \sum_{j=1,j\neq i}^{N_R} | < \mathbf{s}_i^k, \mathbf{s}_j^k > | \tag{26}$$

is selected, where $| < \mathbf{s}_j^k, \mathbf{s}_i^k > |$ is the absolute value of the inner product between $\mathbf{s}_j^k$ and $\mathbf{s}_i^k$ and where N_R is the number of range blocks. Once the kth basis vector direction has been determined, the remaining vectors, $\mathbf{s}_i^k$, are projected onto the subspace that is orthogonal to $\mathbf{s}_l^k$ using the projection operator

$$P_{S_k} = I - \mathbf{s}_l^k(\mathbf{s}_l^{k^T}\mathbf{s}_l^k)^{-1}\mathbf{s}_l^{k^T}. \tag{27}$$

The selected basis vector direction is saved as $\mathbf{t}_k$ and the process is repeated until the required set of N_S vectors are obtained. Essentially this procedure performs a Gram-Schmidt orthogonalization of the range vectors. However, the vector used in each step or the orthogonalization is the vector that has the largest correlation with the other remaining vectors. In this manner, the set of N_S *direction vectors*, $\{\mathbf{t}_i\}_{i=1}^{N_S}$, are determined that best represent the subspace S^0. Since these *direction vectors* are orthogonal, the last two criteria in the list above are satisfied.

The last step is to use the direction vectors to select the domain vectors that will be used in the IFS coder: The procedure here is to find the set of domain vectors that are most closely aligned with the direction vectors. Since the order in which the $\mathbf{t}_i$ vectors are selected is important, with the most significant vector coming first, the same order is used to find the domain vectors. Specifically, beginning with $\mathbf{t}_1$ and progressing through to $\mathbf{t}_{N_S}$, at the kth step the domain vector is selected that is most closely aligned with $\mathbf{t}_k$ in the sense of maximizing the projection

Since it is possible for one domain vector to be the largest component for more than one direction vector, each domain vector is only allowed to be used once. In summary, this algorithm proceeds as follows:

1. Loop NS times to find the *direction vectors*, $\mathbf{t}_i$
   (a) Find vector with largest correlation using equation (26)
   (b) Save vector as $\mathbf{t}_i$
   (c) Project remaining into subspace using equation (27)
2. Loop Ns times for each $\mathbf{t}_i$ to find best domain vector
   (a) Find domain vector with largest component in direction of $\mathbf{t}_i$ using Eq. (28)
   (b) Save vector and eliminate from list of domain vectors

The final result is the set of domain vectors that are to be used in the IFS image coder.

4.2. K-Means Based Approach

Instead of searching *incrementally* over the range vectors for a set of N_S direction vectors, another approach would be to use a clustering algorithms. Therefore, another approach that we considered is to use the K-means clustering algorithm to group the range vectors into a fixed number clusters and then a single representative vector, the mean vector, is chosen to represent each cluster. In the K-means algorithm, since $\mathbf{s}_i$ and $-\mathbf{s}_j$ are very different vectors and, since we are interested in finding a set of basis vectors, it is preferable to treat these two vectors as being equivalent. Therefore, a simple modification of the data prior to running the K-means algorithm is performed to eliminate this problem. Specifically, all of the vectors are forced to reside in the same half-space. The half-space that was selected is the one which includes the positive axes, and any vector that is outside of this half-space is negated. A vector may be easily be checked to see if it lies in this half-plane by examining the sign of the sum of the elements in each vector,

$$h_j = \sum_{i=1}^{D} \mathbf{s}_j[i]. \tag{29}$$

If $h_j < 0$, then the negative of that vector was used. In addition, since the magnitude of each of the range vectors is not important for a set of basis vectors, each vector was also normalized prior to running the K-means routine. Once the N_S direction vectors have been determined, the domain vectors were searched to find the set which match these *direction vectors*.

5. THE IFS ENCODER AND DECODER

The IFS encoder using an orthonormal basis is based on Eq. (25) which provides the weights that specify the vector in the rotated coordinate system. In order to achieve compression, however, the weights must be quantized and, ideally most of the weights should be set to zero. The encoding algorithm consists of the following steps:

1. Determine the domain vectors to use by one of the above methods.
2. Form the B matrix and use the Gram-Schmidt algorithm to get Q.
3. Determine the map parameters, $\mathbf{w}_i$, with Eq. (25).
4. Save those weights which exceed a threshold T_w.

The final encoded image consists of the indices for the N_S domain vectors and the quantized weights for each map which exceed the given threshold. By adjusting the threshold, the accuracy of the reconstructed image can be controlled. In addition, the quantization approach can be different for the *a priori* vectors, which will tend to have a different distribution as compared to the weights for the $\mathbf{b}_i$ vectors.

The IFS decoder, on the other hand, is based on Eq. (24). The reconstruction process is much simpler than the encoding procedure, and begins with any initial image, V_0, and iteratively performs the following steps:

1. Gather the basis vectors, $\hat{\mathbf{d}}_i$ from the image.
2. Form B as given in Eq. (23).
3. Apply the Gram-Schmidt algorithm to B to get Q.

4. Compute each $\mathbf{p}_i = Q\mathbf{w}_i$ and save in the image.
5. Go to step 1.

The image typically converges in a few iterations.

We conclude this section with an example. Shown in Fig. 2 is an original 512×512 image that is to be encoded. Using the covariance method, this image was coded using 0.44 bpp with a PSNR of 30.5 in the decoded image [10], [19]. The result is shown in Fig. 3. The basis vectors that were used are shown in Fig. 4. In the first figure, the blocks are shown prior to the Gram-Schmidt process, and in the second figure the orthonormal vectors are presented. The results obtained with the K-means algorithm were similar.

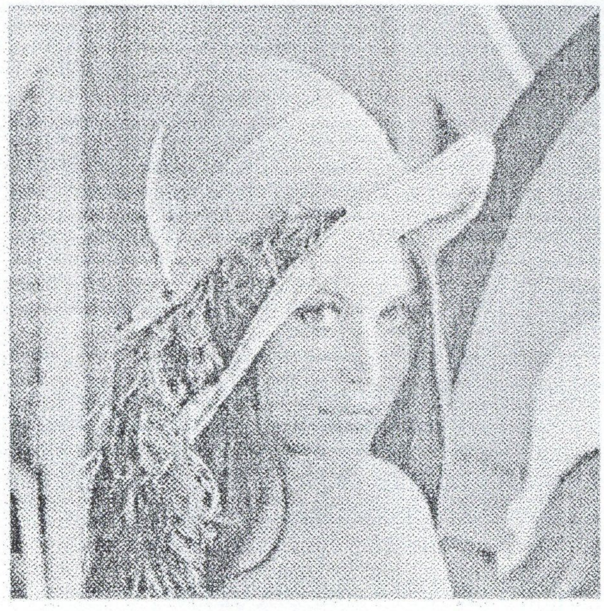


Fig. 3. LENA image coded with orthonormal basis method at 0.44 bpp, PSNR=30.SdB

6. IFS CODING OF VIDEO

It is possible to extend the techniques of image coding using an IFS to code an image (video) sequence. However, with an additional dimension, there are several different approaches that may be taken. The most straightforward approach would be to partition a video sequence into non-overlapping three-dimensional range sequences as is done for a single image. Then, for each three-dimensional range block, a search would be performed over a region extending in space and time for the best domain block that would be mapped, by an IFS, to the given range block. Another approach is to view the video sequence as a sequence of independent frames and apply an IFS coding algorithm to each frame individually. In both of these approaches, however, the computational burden associated with searching for the best contraction mapping would be prohibitive.

Another approach that has been proposed is to partition the video sequence into three-dimensional range blocks, each consisting of a variable number of rectangular blocks that belong to consecutive frames along the same motion trajectory [21]. The variability of the depth in the range blocks is introduced to take into account the effect of occlusion. The IFS map for the first block in any range is found by performing a full search for the best domain block in the corresponding frame. For each of the remaining blocks in the range, the IFS map is found by performing a limited search around the projection of the domain block associated with the first block on the corresponding frame along the motion trajectory. In this case, some of the parameters that describe the IFS map for the first block can be propagated to the following blocks in the range.

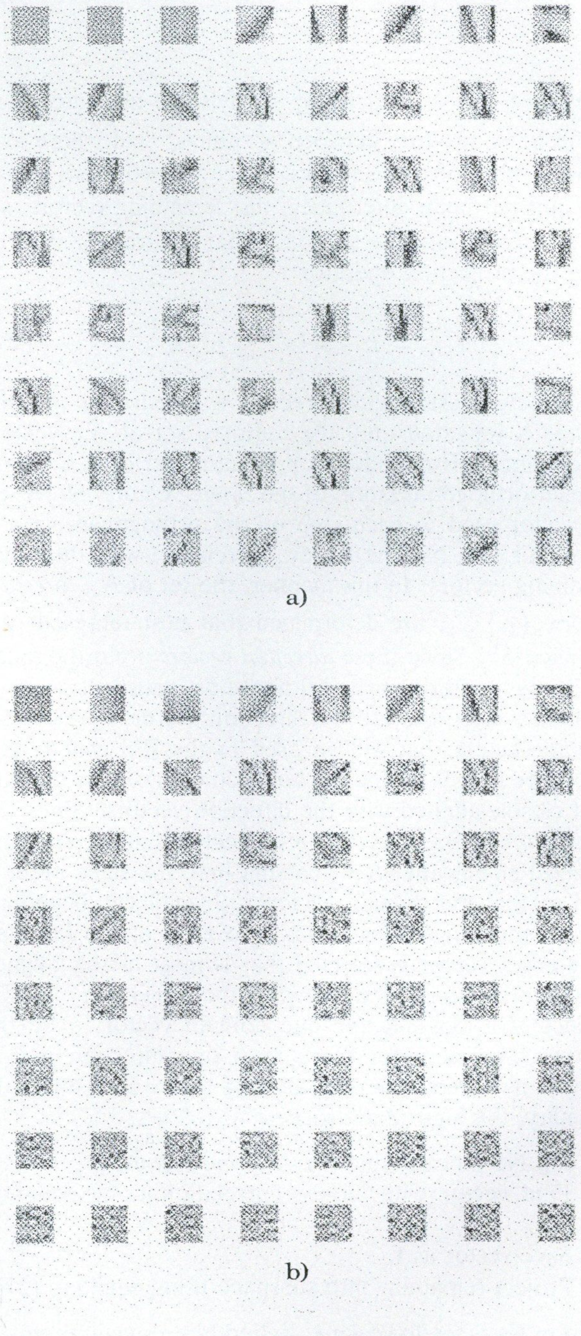


Fig. 4. Basis vector blocks for covariance method. (a) Before Gram-Schmidt orthogonalization and (b) After Gram-Schmidt orthogonalization

This approach exploits the correlation that exists between successive blocks along the direction of motion by predicting the IFS maps of a given range block with that of a parent range block along the trajectory of motion. In effect, this extends the assumption of near intensity constancy along the motion trajectory to IFS maps. Since frame to frame motion is relatively limited in scope, it is computationally much more efficient to perform motion estimation for a matching range block in a previous frame, followed by a refinement of the predicted map in the current frame. The preliminary results of this technique are quite promising [21].

7. SUMMARY AND CONCLUSIONS

In this paper, we have provided an overview of the basic principles and concepts involved with using iterated function systems for image and video coding. Although the IFS approach to coding has some potentially exciting possibilities, at this point it is not clear what types of signals or images are best suited to IFS encoding. Initial IFS coding systems were computationally intensive, requiring long searches and perhaps operator intervention. Fortunately, this is changing as the encoding and decoding algorithms are becoming more efficient. Furthermore, the use of a multiple domain IFS image coding system provides added flexibility and potentially increased performance. The issues that must be addressed in the design of these systems include

1. Techniques for selecting the optimum domain vectors.
2. Efficient algorithms for finding the optimum set of mapping parameters, α_k and β_k.
3. Methods for insuring that the mapping F is a contraction, or that it is at least eventually contracting [20].
4. A rendering algorithm to find the attractor of the IFS.

At this point, the evaluation of multiple image domain IFS coders are only in the preliminary stage. Although the results obtained so far are encouraging, further testing and analysis will be necessary before it will be possible to draw any definite conclusions about their efficiency. In spite of the remarkable progress in IFS image coding that has been made over the last several year, it also unfortunately appears that, with a few exceptions, the image compressions that have been reported are not dramatically better than those obtained using more traditional encoding systems such as JPEG or VQ. The research that is performed over the next few years should answer the question of whether or not IFS encoders will be able to satisfy the needs of a broad-based market.

REFERENCES

[1] B. B. Mandelbrot: "The Fractal Geometry of Nature", New York, N. Y.: W. H. Freeman and Co., 1982.

[2] M. Barnsley: "Fractals Everywhere", New York, N. Y.: Academic Press, Inc., 1988.

[3] A. W. Naylor, G. R. Sell: "Linear Operator Theory in Engineering and Science", Springer-Verlag, New York, 1982.

[4] M. Barnsley, L. P. Hurd: "Fractal Image Compression", Wellesley, MA., AK Peters, 1993.

[5] W. Rudin: "Principles of Mathematical Analysis", McGraw-Hill, New York, 1976.

[6] M. H. Hayes: "Inverse Problems: An Overview", *J. Soc. Instr. and Control Eng.*, Vol. 15, No. 12, pp. 1089-1094, Dec. 1986, JAPAN (invited paper).

[7] R. W. Schafer, R. M. Mersereau, M. A. Richards: "Constrained iterative restoration algorithms", *Proc. IEEE*, Vol. 69, No. 4, pp. 432-450, April 1981.

[8] D. S. Mazel, M. H. Hayes: "Using iterated function systems to model discrete sequences", *IEEE Trans. on. Signal Processing*, Vol. 40, No. 7, pp. 1924-1934, July, 1992.

[9] D. Mazel: "Fractal modeling of time series data", Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA., 1991.

[10] G. Vines: "Signal modeling with iterated function systems", Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA., 1993.

[11] A. E. Jacquin: "A Fractal theory of iterated Markov operators with applications to digital image coding", Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA., 1989.

[12] A. E. Jacquin: "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," *IEEE Trans. on Image Processing*, Vol. 1, pp. 18-30, January 1992.

[13] G. Vilnes, M. H. Hayes: "Adaptive IFS Image Coding with Proximity Maps", *Proc Int. Conf. on Acoust., Speech, Sig. Proc.*, Vol. 5, pp. 349-352, April 1993.

[14] G. Vines, M. H. Hayes: "Nonlinear address maps in a one-dimensional Fractal model", *IEEE Trans. on Signal Processing*, Vol. 41, pp. 1721-1724, April 1993.

[15] G. Vines, M. H. Hayes: "Map search strategies for IFS image compression algorithms", *Quatorzieme Colloque sur le Traitment du Signal et des Images*, Juan-les-Pins, France, Sept. 1993.

[16] L. Thomas, F. Deravi: "Pruning of the transform space in block-based Fractal image compression", *Proc. Int. Conf. on Acoust., Speech, Sig. Proc.*, Vol. 5, pp. 341-344, April 1993.

[17] T. A. Ramstad, G. E. Øien, S. Lepsøy: "An Inner Product Space Approach to Image Coding by Contractive Transformations," in *Proc. ICASSP*, Vol. 4, pp. 2773-2776, 1991.

[18] M. Gharavi-Alkhansari, T. S. Huang: "A Fractal-based image block-coding algorithm", *Proc. Int. Conf. on Acoust., Speech, Sig. Proc.*, April 1993.

[19] G. Vines, M. H. Hayes: "Orthonormal basis approach to IFS image coding", *Proc IEEE Multidim. Sig. Proc. Workshop*, Cannes, France, September 1993.

[20] Y. Fisher, E. W. Jacobs, R. D. Boss: "Iterated transform image compression", *NOSC TR-1408*, Naval Oceans Systems Center, San Diego, CA., 1991.

[21] B-B. Paul, M. H. Hayes, "Fractal-based compression of motion video sequences", submitted for publication in *1994 IEEE Int. Conf. on Image Proc.*

Monson H. Hayes received his Sc.D. Degree from M.I.T. and is currently a Professor in the School of Electrical and Computer Engineering at Georgia Tech. He received the 1983 IEEE Senior Award for the author of a paper of exceptional merit from the Acoustics, Speech, and Signal Processing (ASSP) Society and in 1984 received the Presidential Young Investigator Award for outstanding teaching and research potential. He has been the Chairman of the Digital Signal Processing Technical Committee of the ASSP Society of the IEEE, an Associate Editor for the IEEE Transactions on ASSP, a member of the ASSP Administrative Committee, Secretary-Treasurer of the ASSP Publications Board, and Chairman of the Publications Board of the Signal Processing Society. He is the author of nearly 100 journal and conference papers and was elected Fellow of the IEEE in 1992.

TRANSFORMS, TRANSFORM CODING

KLT-BASED CODING OF I-FRAMES IN STEREOSCOPIC IMAGE SEQUENCES

K. I. DIAMANTARAS, N. GRAMMALIDIS and M. G. STRINTZIS

INFORMATION PROCESSING LABORATORY ARISTOTLE UNIVERSITY OF THESSALONIKI
THESSALONIKI, 54006, GREECE

1. INTRODUCTION

The present DISTIMA coding schemes [2] use only P-frames for the right channel channel sequence. The purpose of the work reported here is to investigate the method by which right channel I-frames might also be profitably used alongside the I-frames of the left channel. Specifically the report compares the standard MPEG-like DCT coding of the right channel I-frames with the (optimum) PCA-based coding based on the eigenvectors calculated at the decoder side using a reconstruction of the corresponding left-channel I-frame.

2. KLT VS. DCT

The basic coding scheme recommended in MPEG [5], [3] is Discrete Cosine Transform coding. Although DCT is suboptimal it enjoys two advantages [6], [4]: (a) it is close to optimal for most images and (b) it is a fixed basis, orthogonal transform therefore there is no need to send the transform basis to the decoder. In this work we investigate the use of non-DCT transform bases for the coding of images. In particular we investigate the use of the ltarhunen-Loeve transform basis which is known to be the optimal linear transform. Similar approaches have been reported elsewhere [1] for non-stereo image sequences. The major problem of KLT is that the basis is image-dependent so we are faced with the issue of either sending basis vectors to the decoder or have the decoder estimate them. In section 3 we choose the latter approach and develop different techniques to improve the estimation of the best transform basis. Section 4 shows the results of our experiments in a test stereo sequence. Finally, in section 5 we conclude showing some direction for future improvements.

3. OPTIMUM I-FRAME CODING

Let $X = [x_{ij}]$ be an 8×8 matrix representing a block to be coded. The DCT transform performs the following mapping

$$Y = TXT', \tag{1}$$

where $T = [t_{ij}]$, $i = 0, \dots, 7$, is the DCT basis matrix

$$t_{ij} = c_i \cos\left(\frac{(2j+1)i\pi}{16}\right), \quad c_i = \begin{cases} 1/\sqrt{8} & \text{if } i = 0 \\ 1 & \text{otherwise} \end{cases}$$

and T' denotes the transpose of T. The transform coefficients form the 8×8 matrix Y. If we let x, y, be the 64-dimensional vectors formed by stacking the columns of X, Y, on top of each other then (1) can be rewritten equivalently as

$$y = (T \otimes T)x, \tag{2}$$

where $\otimes$ is the matrix Kronecker product. Since T is orthogonal (namely $TT' = I$), $(T \otimes T)$ is also orthogonal, thus DCT falls into the general class of orthogonal linear transforms

$$y = Ax, \quad AA' = I. \tag{3}$$

It is well known [4] that Karhunen-Loeve is the optimal transform of type (3) for two reasons: (a) the transform coefficients y_i are uncorrelated, so we do not waste bit-rate to code correlated information, and (b) KL packs the most energy of the signal x, in the fewest possible transform coefficients, i.e. for any $m < 64$ the partial energy $\sum_{i=1}^{m} E\{y_i^2\}$ is maximum for KL among all orthogonal transforms ($E\{\cdot\}$ denotes statistical expectation).

The rows of the optimal KL matrix A^* are the (normal) eigenvectors of the autocorrelation matrix $R = E\{xx'\}$ of x,

$$A^* = [e_1 e_2 \dots e_{64}]', \quad Re_i = \lambda_i e_i, \tag{4}$$

where we assume that the eigenvalues are arranged in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{64} \geq 0$.

One would like to code the images using the optimal KL transform rather than DCT. However, there is a serious problem: the KLT basis is not fixed — it is image-dependent — and therefore it is not known to the decoder. Although some bit-rate is saved by using KLT instead of DCT we face the issue of transmitting the transform basis. Adding the bit-rate required to send the basis to the decoder normally not only erases the savings of KLT but can even result in higher bit-rates compared to DCT.

3.1. Camera calibration

We can attack the problem by using some estimate of A^* at the decoder and the encoder. Thus we avoid altogether sending the eigen-basis of KLT, with the hope — depending on how good the approximation is — that we still retain some performance advantage over DCT.

In the case of stereoscopic image sequences it is assumed that an I-frame in the left channel has approximately the same statistics as its corresponding I-frame in the right channel, after camera calibration to correct for systematic luminance differences between the two cameras. Using a linear calibration model we assume that the images

$$x^r_{ij} \quad \text{(right channel)}$$

$$ax^l_{ij} + b \quad \text{(calibrated left channel)}$$

have similar statistics for some scalar parameters a, b. For blocks this assumption translates into the following formula

$$R^r = E\{x^r x^{r'}\} = E\{(ax^l + bu)(ax^l + bu)'\}, \tag{5}$$

where x^r, $x^l \in \mathbb{R}^{64}$, denote the blocks in the right, left channel respectively, and $u = [1 \ \cdots 1]' \in \mathbb{R}^{64}$. Eq. (5) implies

$$E\left\{\left(I - \frac{uu'}{64}\right) x^r x^{r'} \left(I - \frac{uu'}{64}\right)\right\} =$$

$$= E\left\{\left(I - \frac{uu'}{64}\right)\left(ax^l + b_1 u\right)\left(ax^l + b_1 u\right)'\left(I - \frac{uu'}{64}\right)\right\}, \tag{6}$$

$$E\left\{\left(x^r - \frac{uu'}{64}x^r\right)\left(x^r - \frac{uu'}{64}x^r\right)'\right\} =$$

$$= a^2 E\left\{\left(x^l - \frac{uu'}{64}x^l\right)\left(x^l - \frac{uu'}{64}x^l\right)'\right\}, \tag{7}$$

or more concisely,

$$R_0^r = a^2 R_0^l, \tag{8}$$

where

$$R_0^r \equiv E\{\tilde{x}^r \tilde{x}^{r'}\}, \quad \tilde{x}^r = x^r - \frac{uu'}{64}x^r \tag{9}$$

$$R_0^l \equiv E\{\tilde{x}^l \tilde{x}^{l'}\}, \quad \tilde{x}^l = x^l - \frac{uu'}{64}x^l. \tag{10}$$

If $x_1 \ldots, x_{64}$, are the pixels in a block x then the vector

$$\tilde{x} = x - \frac{uu'}{64}x = \begin{bmatrix} x_1 - \frac{1}{64}\sum_{i=1}^{64} x_i \\ \vdots \\ x_{64} - \frac{1}{64}\sum_{i=1}^{64} x_i \end{bmatrix}$$

is equal to the block x after subtracting the DC term $\frac{1}{64}\sum_1^{64} x_i$. Using (8) we estimate the correlation matrix R_0^r of hte AC right blocks with the scaled correlation matrix $a^2 R_0^l$ of the AC left blocks. The scale parameter a^2 is not known. Observe however that we are only interested in the eigenvectors of R_0^r, and they are not affected by simple scaling. Thus the KLT basis can be estimated from the eigenvectors of R_0^l and there is no need for estimating neither a nor b.

3.2. The proposed transform

The transform which we shall use is identical to DCT in the DC term, but differs in the AC terms. For the AC terms we shall use the KLT basis of R_0^r which, since R_0^r is unknown to the decoder, is estimated from the eigenvectors of R_0^l.

The matrix $\left(I - \frac{uu'}{64}\right)$ is the orthogonal projector matrix for the subspace $\mathcal{L}\perp u$, so $R_0^r u = 0$ and $R_0^l u = 0$. Thus the vector $\frac{1}{8}u$ is a normal eigenvector of R_0^r (and R_0^l) corresponding to the zero eigenvalue. In general, the remaining 63 eigenvalues are nonnegative, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{63} \geq 0$ and their corresponding eigenvectors of R_0^r are orthogonal to $\frac{1}{8}u$ because $R_0^r$is symmetric, and so they form the basis for the AC part.

The total transform matrix is

$$\tilde{A} = \left[\frac{1}{8}u\; e_1\; e_2\; \cdots\; e_{63}\right]'. \tag{11}$$

$\tilde{A}$ is orthogonal, so

$$x = \frac{1}{8}uy_1 + \sum_{i=2}^{64} e_{i-1}y_i$$

$$y_1 = \frac{1}{8}u'x, \quad y_i = e_{i-1}'x, \quad i = 2, \ldots, 64.$$

3.3. Coding

The coding is done exactly according to DISTIMA. The only difference is that the I-frames of the right channel are coded with the approximated KLT transform explained above. The transform coefficients are quantized with exactly the same quantizers proposed in DISTIMA for the DCT coefficients, with the same quantization matrix Q_I sorted into the following 1-dimensional array

$$\begin{aligned}\tilde{Q}_I = [&8, 16, 16, 16, 19, 19, 22, 22, 22, 22, 22, 22, 24, 26, 26, 26,\\ &26, 26, 26, 27, 27, 27, 27, 27, 27, 27, 27, 29, 29, 29, 29, 29,\\ &29, 29, 29, 32, 32, 34, 34, 34, 34, 34, 34, 35, 35, 35, 37, 37,\\ &38, 38, 38, 40, 40, 40, 46, 46, 48, 48, 56, 56, 58, 69, 69, 83]\end{aligned} \tag{12}$$

For each macroblock we make the field/frame DCT decision as prescribed by DISTIMA, while for simplicity we use a fixed mquant parameter. We are only interested in I-frames and as a result all macroblocks are intra macroblocks.

3.4. Segmentation

Images typically include various regions of different texture: for example, objects in the foreground may be characterized by different statistics than the background. This nonstationary nature of images is ignored if we use a uniform transform basis for the whole image. A further improvement would require that we partition the image into segments and assign a single transform basis per segment. The level of sophistication of such a segmentation algorithm may be anything from very simple to very complex with different advantages and disadvantages in each case.

We have investigated two different approaches:

1. *Straightforward partitioning:* cut the image into rectangular segments of equal sizes (or as close to equal as possible). This is the simplest segmentation approach. The advantage is that no additional bit-rate is required to make the segmentation known to the decoder. The disadvantage is that segments do not correspond (in general) with regions of uniform statistics. Still if the estimation of the eigenvectors was perfect the performance should improve over the globally uniform basis case and the more parts the better. On the other hand, the more parts the smaller they become, and therefore, the bigger the proportion that has changed from the left eye to the right eye, and as a result the poorer the estimate of the basis. Also the more parts we have the longer it takes to compute all the different eigenbases. So there is a trade-off between the benefit from specializing the eigenbases and the damage from loosing eigenbase estimation accuracy. As we shall see in the experiment section the balance point of this trade-off corresponds to small numbers of parts.
2. *Segmentation based on the variance of the macroblocks.* In this approach we compute the variance of each macroblock in the right channel and create the histogram of these values which is used to partition the macroblocks into (not necessarily connected) groups. We do this by first locating the valleys in the histogram and count the number of parts into which the histogram is partitioned by them. If these parts are less than desired we stop there. Otherwise, we merge the smallest consecutive parts together until we have as many parts as desired.
The advantage of this approach is that it usually gives regions with better statistical coherence than the straightforward approach. The disadvantage is that the segmentation is not known to the decoder and thus we have to spend bits to make it known. If we have N parts, direct coding would require $\lceil \log_2 N \rceil$ bits per macroblock (or $\lceil \log_2 N \rceil$ /256 bits per pixel). For example, if we have 4 segments, we can represent the class of each macroblock using just 2 bits per macroblock (or 0.008 bits per pixel) without using entropy coding.
The same trade-off as in the straightforward case occurs here as well. For a certain number of segments increasing the number of segments will only make the performance worse.

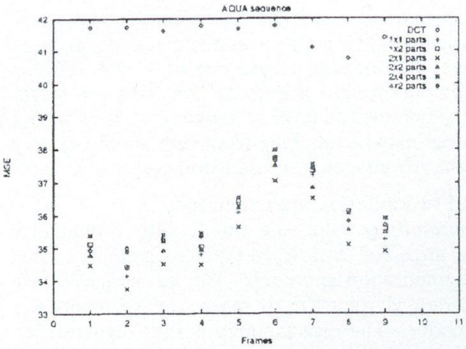
Notice that in none of these approaches (or any other segmentation approach) does the encoder need to send the eigenbases. Once the decoder knows the left image and the segmentation, it can estimate the eigenvectors on its own as with the single segment case.

4. EXPERIMENTAL RESULTS

We conducted experiments for a test sequence with rather small disparities between left and right eye ("aqua") and good calibration between the two eyes.

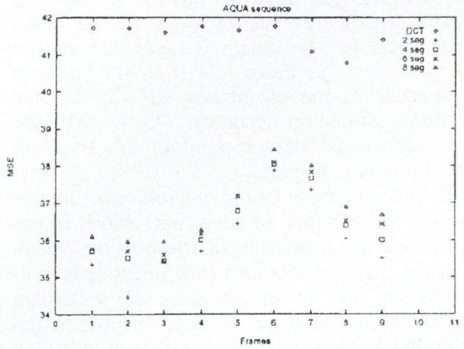
We implemented the DISTIMA protocol for I-frames using DCT for comparison purposes. In addition we implemented the KLT-based algorithm described above using (a) straightforward segmentation with different partitions: 1×1 (no segmentation), 1×2, 2×1, 2×2, 2×4, 4×2, and (b) variance-based segmentation with different numbers of segments: 2, 4, 6, and 8 (1 segment is the same as the 1×1 partition of case (a)). We measured and compared the entropy of the transform coefficients in each case and we also measured the square error of the quantization. We ignored all other bits (header sequences, etc.) since they are the same for all methods.

Fig. 1 shows the results of the experiments with the straightforward segmentation. We see that we have significant improvement of the square error over DCT, without significant difference in the expected bit-rate. We even have less entropy.



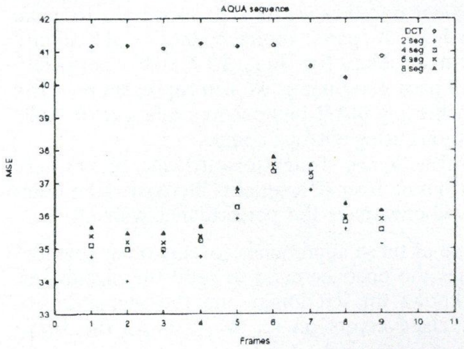
dct entropy:	59.36
1x1 entropy:	57.26
1x2 entropy:	57.25
2x1 entropy:	56.73
2x2 entropy:	56.83
2x4 entropy:	57.09
4x2 entropy:	56.60

Fig. 1. Straightforward segmentation: the aqua sequence. The partitions tried are 1×1, 1×2, 2×1, 2×2 *and* 2×4. *The entropy for each case is measured in bits per block.*



dct entropy:	59.36
2 seg entropy:	57.11
4 seg entropy:	56.90
6 seg entropy:	56.76
8 seg entropy:	56.76

Fig. 2. Variance-based segmentation: the aqua sequence. We experimented with 2, 4, 6 and 8 segments. The entropies are in bits per block.



dct entropy:	58.55
2 seg entropy:	56.35
4 seg entropy:	56.09
6 seg entropy:	56.04
8 seg entropy:	55.98

Fig. 3. Variance-based segmentation: the calibrated aqua sequence. The entropies are in bits per block.

Fig. 2 shows the corresponding results for the variance-based segmentation. We still observe that we enjoy some advantage. Finally Fig. 3 shows the corresponding results when using explicit calibration along with the variance-based segmentation scheme. In this case we calibrate the right channel by a simple linear transformation:

$$r' = a \cdot r + b \tag{13}$$

The parameters a and b are determined so that they adjust the right channel so as to have the same mean and variance as the left channel [7]. This calibrated version of the right channel, which statistics approximate those of the left channel, is coded more efficiently using KLT since the accuracy of the estimation of the eigenbasis, which is based on the luminances of the left channel, increases.

In this case the calibrated image estimate R^r (correlation matrix) is directly used rather than R^r_0. The eigenvectors of R^r are the globally optimal KLT basis. Results show a significant decrease of the mean square error over the previous cases. We also noticed that the result of using DCT with this calibrated version of the right channel is also improved. Sample images and errors are displayed in Fig. 4.

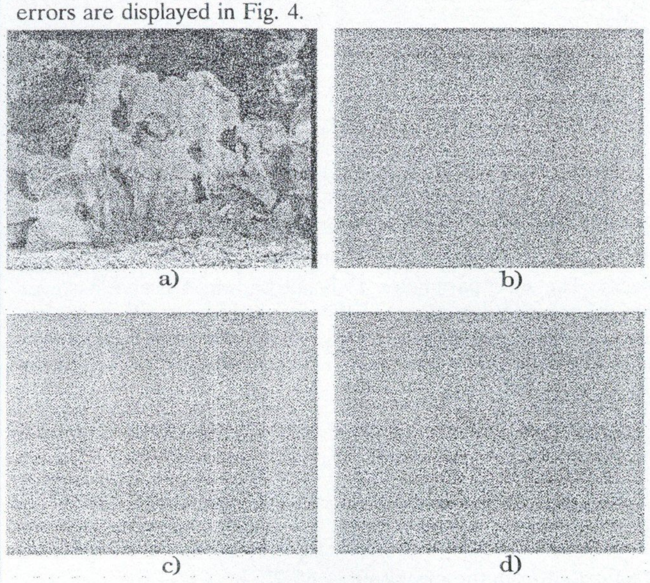


Fig. 4. Images from the "aqua" experiment: (a) Original image (right channel); (b) Error image for DCT coding; (c) Error image for KLT-based coding with no segmentation; (d) Error image for KLT-based coding using variance-based segmentation with 2 segments. Error images are enhanced to show differences.

5. CONCLUSIONS AND FUTURE DIRECTIONS

Overall the methods outlined here are most successful for stereo sequences that have good balancing between the left and right camera. Poor calibration results in deteriorating performance. A disadvantage of the proposed method is the additional computational cost for the eigenvector estimation. Still the overhead is not large since there exist very fast eigen-computation techniques. However as shown by the results in figures 1-6, DCT gives very good results and does not suffer from this disadvantage. Thus it is not clear at this point whether the payoff of the optimal method is high enough to warrant its use as a DCT replacement for the coding of moving image sequences.

Various lines of improvement of this work can be pursued in the future. Some ideas are outlined below.

- *Use of more sophisticated calibration models.* The presence of disparity between the left and the right images results to errors in the estimation of the eigenbasis of the right image at the decoder when a model of the left image is used. The use of higher order calibration models or models that compensate for the presence of disparity would improve the estimation of the eigenbasis of the right image at the decoder.

Furthermore the use of more sophisticated segmentation schemes may be examined.

- *Use quantization matrices designed for KLT statistics.* So far all the experiments were conducted using the MPEG-provided quantization matrices (Q_I, Q_N) which were designed for the DCT coefficient statistics. Of course the design of KLT quantization matrices would require extensive experimentation with more than just 3 test sequences.
- *Improve eigen-basis estimation* by using both the corresponding frame from the other channel and the previous frame of the same channel.
- *Using non-DCT bases for coding predicted frames.* Typically the prediction error images resemble edge maps. That is because the DCT error is higher at high frequencies, namely where there are abrupt changes in the luminance. Therefore error images have special statistics. Using KL along with vector quantization for coding and transmitting the basis matrix A^* of error images may prove superior to traditional DCT coding.

REFERENCES

[1] K. I. Diamantaras, S. Y. Kung: "Compressing Moving Pictures using the APEX: Neural Principal Component Extractor". *Proceedings, IEEE Neural Networks for Signal Processing Workshop*, Baltimore MD. September 1993.

[2] R. Franich, R. Lagendijk, R. Ter Horst: "Codec Specification for Hardware Implementation". RACE Project # R2045: Digital Stereoscopic Imaging and Applications (DISTIMA), December 1992.

[3] D. Le Gall: "MPEG: A Video Compression Standard for Multimedia Applications". *Trans. ACM*, April 1991.

[4] A. K. Jain: "Fundamentals of Digital Image Processing". Prentice Hall, Englewood Cliffs, NJ, 07632, 1989.

[5] "Coded Representation of Picture and Audio Information". Test Model Editing Committee Draft ISO-IEC/JTC1/SC29/WG11 MPEG 92/NO245, July 1992.

[6] A. N. Netravali, B. G. Haskell: Digital Pictures, Representation and Compression. Plenum Press, 1988.

[7] R. E. H. Franich: Balance Compensation for stereoscopic sequences. RACE Project # R2045: Digital Stereoscopic Imaging and Applications (DISTIMA).

A COMPARATIVE STUDY ON SOME SUBBAND TRANSFORMS FOR STILL IMAGE COMPRESSION

D. MILOVANOVIĆ
UNIVERSITY OF BELGRADE
DEPT. OF ELECTRICAL ENGN.
BULEVAR REVOLUCIJE 73, YU-11000
BELGRADE, YUGOSLAVIA

A. SAMĆOVIĆ
UNIVERSITY OF BELGRADE
DEPT. OF TRAFFIC AND
TRANSPORT ENGN. VOJVODE STEPE 305
YU-11000 BELGRADE, YUGOSLAVIA

Z. BOJKOVIĆ
UNIVERSITY OF BELGRADE
DEPT. OF TRAFFIC AND
TRANSPORT ENGN. VOJVODE STEPE 305
YU-11000 BELGRADE, YUGOSLAVIA

1. INTRODUCTION

Subband coding SBC has recently become an efficient technique for high bit-rate compression [1]. It also seems to be effective in high quality image reproduction, because it does not suffer from block distortion as in the case of transform coding. The SBC scheme is also suitable for hierarchical coding, because lower resolution images can naturally be obtained in the tree structured subband decomposition process [2].

In SBC an image is first filtered to create a set of images, each of which contains a limited range of spatial frequencies. These images are called the subbands. Since each subband has a reduced bandwidth compared to the original full-band image, they may be downsampled. This process of filtering and subsampling is termed the analysis stage. The subbands are then encoded using one or more coders. Different bit rates or even different coding techniques can be used for each subband, thus taking advantage of the properties of the subband and/or allowing for the coding errors to be distributed across the subbands in a visually optimal manner. Reconstruction is achieved by upsampling the decoded subbands, applying appropriate filters, and adding the reconstructed subbands together. This is termed the synthesis stage. The motivation for this approach is that the subbands can be encoded more efficiently than the original image. The key elements in a SBC system are: the analysis and synthesis filtering bands and the coding techniques applied to the subbands [3].

Attractive SBC advantages motivated this presentation. The first part deals with some theoretical aspects concerning same subband transforms, as well as entropy coding. The second part of the paper seeks to provide a model for a comparative study on some subband transforms for still image compression. Finally, some numerical as well as simulation results will be reported.

2. SUBBAND TRANSFORMS

Subband transforms are a subclass of linear transforms which offer useful properties for these applications. These transforms are generally computed by convolving the input signal with a set of bandpass filters and decimating the results. Each decimated subband signal encodes a particular portion of frequency spectrum, corresponding to information occurring at a particular spatial scale. To reconstruct the signal, the subband signals are upsampled, filtered, and then combined additively. For purposes of coding, subband transforms can be used to control the relative amounts of error in different parts of the frequency spectrum. Most filter designs for subband coders attempt to minimize the "aliasing" resulting from the subsampling process. In the spatial domain, this aliasing appears as evidence of the sampling structure in the output image. An ideal subband system incorporates "brick-wall" bandpass filters which avoid aliasing altogether. However, such filters produce ringing in the spatial domain which is perceptually undesirable.

As an example, consider the block discrete cosine transform DCT as a subband transform [4]. Computing a DCT on non-overlapping blocks is equivalent to convolving the image with each of the block DCT basis functions and then subsampling by a factor equal to the block spacing. The Fourier transform of the basis functions indicates that each of the DCT functions is selective for a particular frequency subband, although it is clear that the subband localization is rather poor. Thus, the DCT qualifies as a subband transform.

3. ENTROPY CODING

The entropy coding EC consists of variable word length (VWL) coder to encode the non-zero quantized values and a run length coder to encode their corresponding locations [5]. This is achieved by considering each scan line as a consecutive black and white run. The black run corresponds to zero and white runs to non-zero values. In this way only non-zero values need to be transmitted. The VWL code set is designed according to the average statistics of all bands so that only one VWL code set can be used for all bands, at the expense of a slight loss of efficiency. It should be noted however, that based on earlier experiments VWL coding of non-zero PCM does not significantly improve the coding performance, particularly in the case of intraframe. Therefore, it may be more convenient to select the number of quantized levels within the active range in such a way that fixed length codes can be efficiently applied.

The individual subimages are first partitioned into non overlapping blocks. The scanning is performed on a block-by-block basis starting from the first block on the upper left and continuing in the horizontal direction until the last block (upper most right) is scanned. The resulting bit stream is then run length coded and transmitted together with non-zero PCM coded values. This process continues in the same manner until the last strip of blocks in each subimage is scanned and coded. For the block size of $m \times n$, the above arrangement can be viewed as transforming $M \times N$ image into $P \times Q$ where $P = M \times n$, $Q = N/n$. As for the manner in which each block is scanned, there are normally three scanning techniques. There are horizontal, vertical and diagonal scanning.

4. SIMULATION RESULTS

In our experiment we first measured an entropy of subimages after the decimation of the test "LENA" image (Fig. 1.). The calculated entropies of those subbands are presented in Table 1.

Table 1.

subbands	11	12	21	22
H [bpp]	7,69	4,68	4,78	5,10

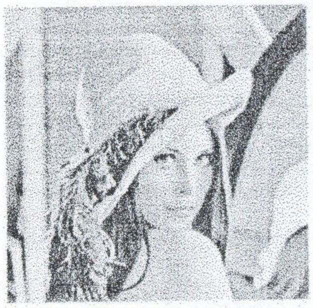


Fig. 1.

Assuming the 256 gray-scale value image, one can calculate the entropy as

$$H = -\sum_{k=0}^{255} p_k \quad ld \; p_k$$

where p_k is probability of a k-th gray-scale value in an image. One cannot compress an image below this value, not loosing an image visibility.

One can conclude from the Table 1 that the greatest entropy, as well as the most energy is concentrated in the lowest subband of the image. Therefore, in order to achieve the redundancy reduction, the most care should be devoted to a coding of the lowest subband.

In the first one, the DPCM coder with 4 bits is applied in the subband 11, while in the subbands 12 and 21 is used the same coder with 3 bits step size. The highest 22 subband is coded by the 3 bits PCM coder. This DPCM coder has an optimal, linear prediction for a previous pixel in the same line and a pixel on the same position for a previous line. A quantizer in the DPCM loop is a nonuniform, optimal one. This coding leads to PSNR = 25.86 dB, with the bit rate 3.25 bpp (2.52 with EC). Here, EC means the estimated subband entropy using 8-bit linear quantizer. Our major asssumption for EC is the independent pixel entropy coding. In comparison, DPCM over full-band image gives PSNR = 27.45 dB with 4 bpp (EC 3.65 bpp). The obtained image is presented in Fig. 2.

The second process involves DCT coder in the lowest subband (4 bits), DPCM coding in the subband 12 (3 bits) and PCM in 21 and 22 with 3 bits, respectively. A separable 2D DCT with 4x4 pixel blocks is taken, while the transformation coefficients are coded by the uniform quantizer with 4 bits. We obtained PSNR = 24.85 dB with 3.25 bpp (EC 1.15 bpp). DCT coding over full-band image gives PSNR = 27.52 dB with the bit rate 4 bpp (EC 0.39 bpp). For this case, the simulated image is shown in Fig. 3.

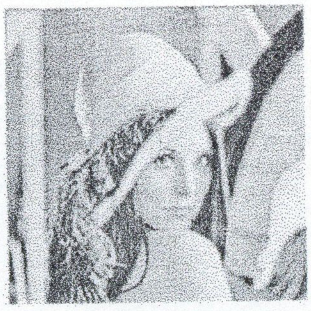


Fig. 2.

These results are shown in Table 2.

Table 2.

coder	BR/BR_{ec}	CR/CR_{ec}	PSNR (dB)
DPCM	4/3.65	2/2.19	27.45
DCT	4/0.39	2/20.5	27.52
SBC-DPCM	3.25/2.52	2.46/3.17	25.86
SBC-DCT	3.25/1.15	2.46/6.96	24.85

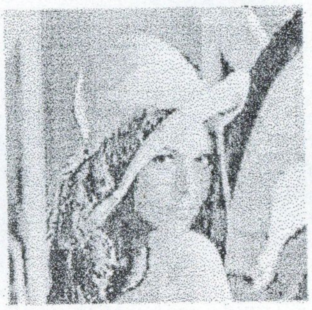


Fig. 3.

Here BR and BR_{ec} represent the bit rates, without and with the entropy coding, $CR=8/BR$ is the compression ratio without, and $CR_{ec} = 8/BR_{ec}$ with the entropy coding, while $PSNR= 10\log(255/\sigma^2_{\rm rec})$ is the peak signal-noise ratio, where $\sigma^2_{\rm rec}$ is the reconstruction value variance.

5. CONCLUSION

In an attempt to suggest a model for still image compression using the comparative study of some subband transforms, we combined the ideas of SBC, EC and transform coding. A coder choice was made on the basis of entropy measuring in subbands. It can be seen that SBC eliminates blocking effects visible in classical DCT use. Combined SBC with DCT gives good compression ratio with an acceptable picture quality. On the other hand, PSNR is a little bit smaller (with the presence of the noise granularity) when SBC-DCT is used instead of DCT over full-band image.

REFERENCES

[1] J. W. Woods, D. O. O'Neil: "Subband Coding of Images", *IEEE Trans. on ASSP*, Vol. 34. No. 5. (October 1986), pp. 1278-1288.

[2] P. H. Westernik, J. Biemond, D. E. Boekee: Evaluation of image subband coding scheme, Signal Processing IV: "Theories and Applications," *Proc. EURASIP*, 1988. pp. 1149-1152.

[3] H. Gharavi: "Subband coding algorithms for video applications: Videophone to HDTV-conference", *IEEE Trans. Circuits and systems for video technology*, Vol. 1. (June 1991), pp. 174-183.

[4] K. R. Rao, P. Yip "Discrete cosine transform algorithms, advantages, applications", Academic Press, Inc. (1990).

[5] H. Gharavi: "Subband coding of video signals, in Subband Image Coding", edited by J. W. Woods, Kluwer Academic Publishers, (1991). pp. 229-272.

MOTION ESTIMATION FOR WAVELET TRANSFORM CODING

L. BÖRÖCZKY and P. CSILLAG

KFKI RESEARCH INSTITUTE FOR MEASUREMENT AND COMPUTING TECHNIQUES
P. O. BOX 49, H-1525 BUDAPEST, HUNGARY

1. INTRODUCTION

Estimation of motion between successive frames of an image sequence has received a lot of attention in different fields of image processing including digital video and dynamic scene analysis. High temporal correlation of intensities along the motion trajectory can successfully be utilized for motion-compensated (MC) video coding, MC image sequence restoration, MC frame interpolation, etc. Applications of dynamic scene analysis include robot vision, traffic monitoring, biomedical imagery and remote sensing.

Frame-to-frame object motion in an image sequence can be represented by a 2D timevarying vector field consisting of the local displacement vectors. This vector field is often referred as motion field, which is unknown in general and has to be estimated based on the given time-varying image sequence. Generally, uniresolution motion estimation algorithms (such as smoothness-based, parametric) using two consecutive frames of full resolution have the property of slow convergence rate even with complex methods [1]. In the worst case, they are not at all suitable to estimate fast motion (large displacement vectors). To overcome this problem, multiresolution motion estimation schemes have been proposed [2], [3].

In video coding motion estimation plays an important role, because its use can reduce the redundancy of the image sequence. Recently, wavelet theory has been proven to be a useful tool in multiresolution representation of video signals and it can be used efficiently for coding [4], [5]. Basically, wavelet transform decomposes the video signal into a set of subbands, which can be coded easier than the original signal by adapting the coding parameters to each band and matching them to the human visual characteristics at different frequencies. Especially, at low bitrate video coding wavelet/subband based methods can be applied successfully, because they eliminate the blocking artifacts, that appear in traditional standardised DCT coding. For MC wavelet coding multiresolution motion estimation is especially suitable, because this signal representation has the inherent property of multiresolution image representation.

In this paper a predictive multiresolution motion estimation scheme for wavelet video coding is proposed. It is based on the estimation of motion fields for all subbands using an advanced propagation strategy, which exploits the spatiotemporal smoothness of motion fields. In Section 2 principles of MC wavelet coding are outlined. After a brief discussion of multiresolution motion estimation a predictive multiresolution scheme for MC wavelet video coding is presented. Experimental results obtained for real-life video sequences are given in Section 4.

2. MOTION-COMPENSATED WAVELET VIDEO CODING

The application of wavelet transform to image/video coding can be considered to be similar to subband coding. In this framework, at first the input signal is split into several subsignals by a set of analysis filters. Each subsignal is then subsampled, quantized and coded. At the receiver the decoded subband signals are upsampled and combined through a set of synthesis filters. The wavelet decomposition and reconstruction can also be specified by quadrature mirror filter (QMF) pairs, that correspond to the scaling and wavelet functions [4]. A typical filterbank and its subband splitting scheme are shown in Figs. 1 and 2.

For illustration of multiresolution wavelet-based image representation Fig. 3 shows frame 41 from the "Trevor White" sequence and its 3-level wavelet transform. The filter coefficients used are shown in Table 1 [4].

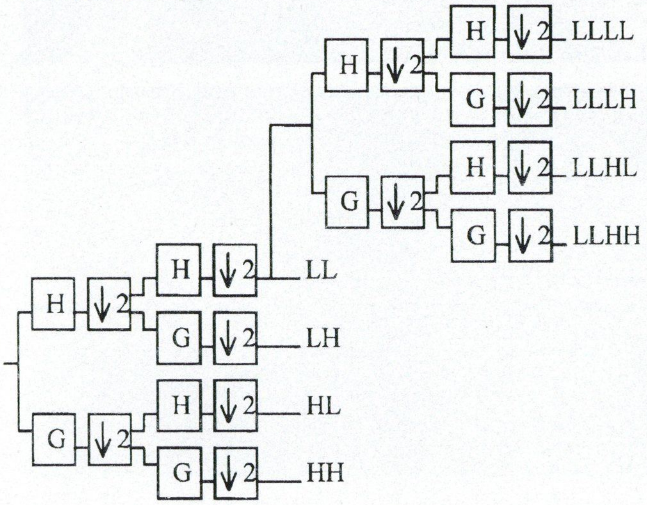


Fig. 1. Example of a filterbank

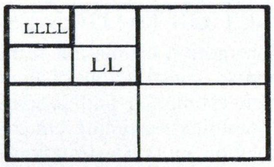


Fig. 2. Splitting an image into 7 frequency bands

Table 1. Coefficients of the filter; h(-n)=h(n)

n	h(n)	n	h(n)	n	h(n)
0	0.542	4	0.023	8	0.006
1	0.307	5	0.030	9	0.006
2	-0.035	6	-0.012	10	-0.003
3	-0.78	7	-0.013	11	-0.002

In an interframe hybrid motion-compensated DPCM/wavelet transform scheme motion compensation can be applied to either the original video signal before the wavelet decomposition or to all subbands after the transform. It was shown that the scheme based on wavelet decomposition of the original video signal followed by motion compensation generally outperforms the other approach in terms of both the signal-to-noise ratio and subjective evaluations [6]. Therefore, our main objective is to estimate motion fields for all frequency bands.

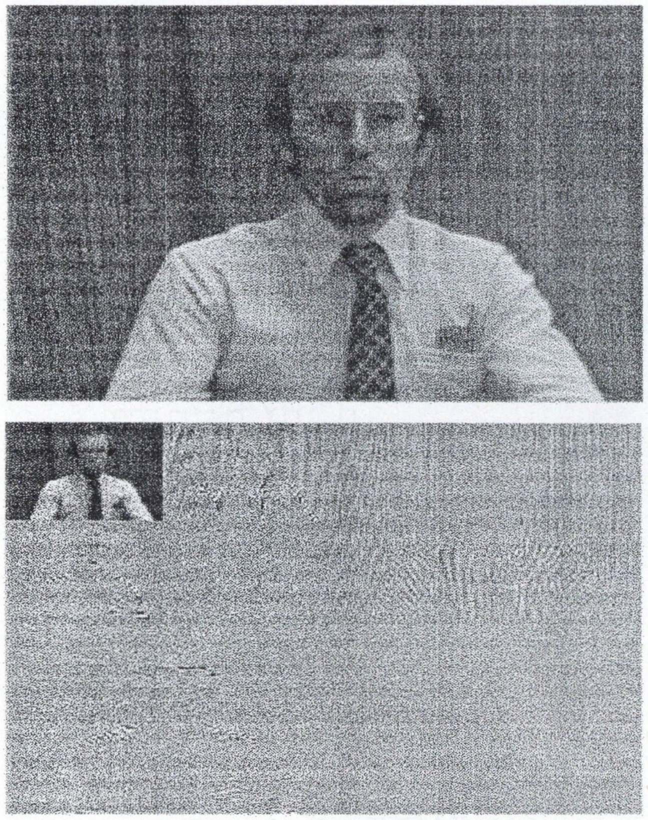


Fig. 3. Frame 41 of image sequence "Trevor White" and its 2-level wavelet decomposition

3. MULTIRESOLUTION MOTION ESTIMATION

The multiresolution motion estimation framework consists of three basic components: construction of multiresolution image representation, motion estimation and propagation strategy for refinement of the motion fields estimated in coarser resolutions.

Generally, uniresolution motion estimation algorithms can be adapted to a hierarchical computational framework without difficulties and requiring only minor modifications. Both nonparametric and parametric methods were investigated in multiresolution environment and their performances with respect to robustness as well as to convergence speed have outperformed the single level algorithms [2], [3], [7].

However, efficiency of these estimation procedures is also determined by the propagation strategy used for refinement of the motion field through the levels of the image pyramid. Design of sophisticated propagation strategy can be considered as a task to develop an advanced prediction strategy for the initial estimate of the motion field. Then it has to be updated within the actual level of the image pyramid by a chosen motion estimation algorithm [8]. A good initial estimate will reduce the magnitude of the unknown update vector, consequently a simple estimation algorithm can be applied for the correction vector field.

For this propagation/prediction strategy it is worth to consider the spatiotemporal smoothness of the motion field [9], [10], which increases mainly the robustness of the motion estimation algorithm. This predictive multiresolution motion estimation approach can be applied for MC wavelet coding and it will be discussed in the following sections.

3.1. Motion estimation

The motion estimation algorithms applied in MC video coding are basically blockmatching and pel-recursive approaches. For our experiments a Wiener-based pel-recursive motion estimator was selected [11]. It operates in a scanwise causal way through the images and is based on a prediction/update principle. At each pixel the following iteration is performed:

$$\mathbf{d}_{i+1}(\mathbf{x}, t) = \mathbf{d}_i(\mathbf{x}, t) + \mathbf{u}_i, \qquad (1)$$

where $\mathbf{d}_i(\mathbf{x}, t)$ is the current estimate for the motion field vector at spatiotemporal location $(\mathbf{x}, t)$, $\mathbf{u}_i$ is an update vector, $\mathbf{d}_{i+1}(\mathbf{x}, t)$ is the new estimate and i is the iteration index. Three strategies need to be specified, such as determination of the initial estimate $\mathbf{d}_0(\mathbf{x}, t)$ to start the iterations, the computation of the update vector $\mathbf{u}_i$ at every iteration and the control of the iteration process. The extension of this algorithm to multiresolution motion estimation can be carried out by determination of the initial estimate $\mathbf{d}_0(\mathbf{x}, t)$ using the propagation strategy of the hierarchical framework.

3.2. Predictive multiresolution motion estimation for wavelet coding

For MC wavelet coding of image sequences, the motion vector field has to be estimated for each frequency band. This can be implemented in several variations of the predictive multiresolution motion estimation. The motion vector for the different frequency bands can be calculated separately, jointly, or the motion vectors calculated for some subbands can be used for all other subbands with proper scaling. Zhang et al. [5] have shown that the most efficient technique is to calculate different vector fields for the different subbands, and the initial estimates of a subband can use the results of another band. This scheme can further be improved by using not only the motion vectors calculated for the current frame, but the ones obtained in the previous frames, too.

We have also proposed a multiresolution motion estimation algorithm for subband coding in [7]. In that scheme spatiotemporal prediction with a Wiener-based pel-recursive displacement estimator was applied.

Experimental results showed that spatiotemporal extension improves the performance of the algorithm, however this approach was not yet extended to the other subbands.

In order to estimate the motion fields corresponding to each subband efficiently and accurately the spatiotemporal prediction strategy proposed in [12] was applied for the lowpass filtered (LL) subbands at each level:

$$\mathbf{d}_0 = f(\mathbf{d}_s, \mathbf{d}_{pl}, \mathbf{d}_t), \qquad (2)$$

where $\mathbf{d}_0$ is the initial estimate for the iteration, $\mathbf{d}_s$ is the spatial motion vector calculated from the vectors obtained around the actual pixel, $\mathbf{d}_{pl}$ is the (properly scaled) motion vector in the previous level and $\mathbf{d}_t$ is the temporal motion vector, calculated in the same level of the pyramid in the previous frame. A linear combination was selected as function f. It contains two parameters, P and Q. P is the ratio of the vector $\mathbf{d}_{pl}$ in calculation of the motion vector only from the two current images:

$$\mathbf{d}_c = P\mathbf{d}_{pl} + (1 - P)\mathbf{d}_s \qquad (3)$$

and Q is the ratio of the temporal part, when the previous motion field pyramid is considered as well

$$\mathbf{d}_0 = Q\mathbf{d}_t + (1 - Q)\mathbf{d}_c. \qquad (4)$$

The final expression for function f is:

$$\mathbf{d}_0 = Q\mathbf{d}_t + (1 - Q)(P\mathbf{d}_{pl} + (1 - P)\mathbf{d}_s). \qquad (5)$$

The parameters of P and Q of the linear equation were determined experimentally for a given image sequence (see Fig. 4 and 5), and its typical values are around (1/3).

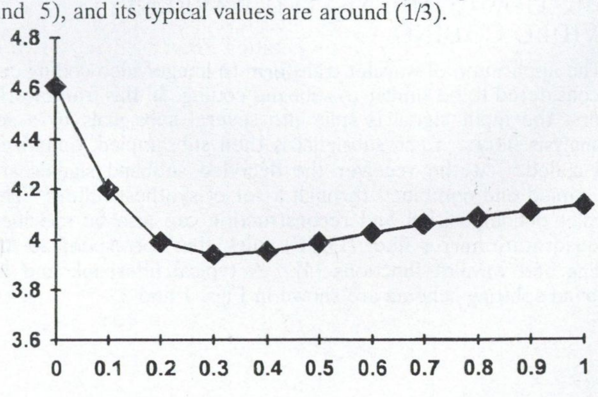


Fig. 4. Mean-square-error of the algorithm as a function of P for frames 31 – 32 of image sequence "Trevor White"

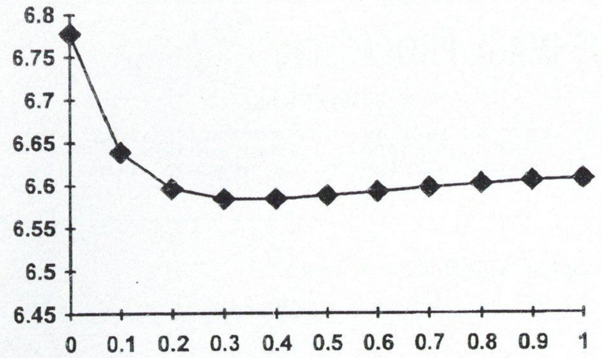


Fig. 5. Mean-square-error of the algorithm as a function of Q for frames 33–35 of image sequence "Trevor White"

For the other subbands (LH, HL, HH) a simpler expression can be used. The initial estimate for these bands is calculated as follows:

$$\mathbf{d}_0 = g(\mathbf{d}_s, \mathbf{d}_{ll}), \qquad (6)$$

where $\mathbf{d}_0$ is the initial estimate for the iteration $\mathbf{d}_s$ is the spatial motion vector calculated from the vectors obtained around the actual pixel and $\mathbf{d}_{ll}$ is the motion vector calculated for the lowest frequency band in the same frame and the same level of the pyramid. The chosen function is also a linear combination:

$$\mathbf{d}_0 = R\mathbf{d}_{ll} + (1-R)\mathbf{d}_s \qquad (7)$$

and the parameters for the different bands (R_{lh}, R_{hl}, R_{hh}) can be determined using similar experiments that were used for estimation of P and Q.

4. EXPERIMENTAL RESULTS

The proposed motion estimation algorithms were tested and verified on a real-life image sequence. The odd fields of the well-known sequence "Trevor White" were selected, and the three algorithms for the low frequency band (the one, using only full resolution images, the one using multiresolution image representation with $P = 0.3$ and the one utilising spatiotemporal propagation strategy $P = 0.3$, $P = 0.35$ were compared). The results are given on Fig. 6.

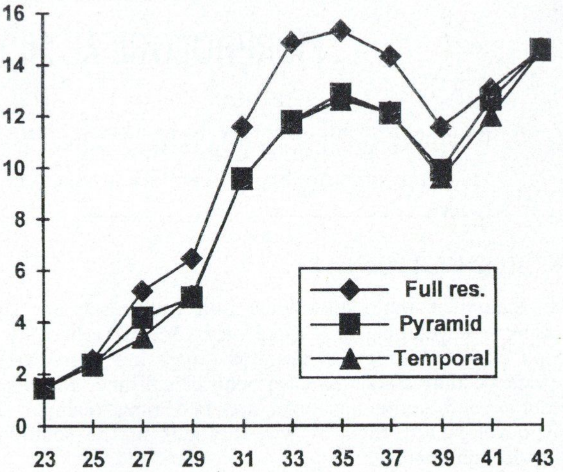


Fig. 6. Mean-square-error of the three algorithms

It is clearly visible, that the pyramidal and the spatiotemporal algorithms significantly outperform the full resolution one, and at some frames (where the motion is smooth enough) the spatiotemporal algorithm results in a smaller mean-square error, then the pyramidal one.

5. CONCLUSION

In this paper an advanced interframe and interlevel propagation strategy was proposed for multiresolution motion estimation in a wavelet image sequence coding environment. Experimental results showed, that the proposed predictive mutiresolution algorithms significantly outperform the one using only full resolution images.

In future research we will focus on testing the proposed motion estimation scheme with an advanced segmentation algorithm which can improve its performance even further. The extension of the proposed multiresolution motion estimation to MC interpolation for low-bit rate wavelet video coding will be also a topic of our future research.

REFERENCES

[1] Q. Wang, L. J. Clarke: "Motion Estimation and Compensation for Image Sequence Coding", *Signal Processing: Image Communication*, Vol. 4. No. 2. April, 1992. pp. 161-174.

[2] W. Enkellmann: "Investigations of Multigrid Algorithms for the Estimation of Optical Flow Fields in Image Sequences", *CVGIP*, Vol. 43. 1988. pp. 150-177.

[3] P. Anandan, J. R. Bergen, K. J. Hanna, R. Hingorami: "Hierarchical Model-Based Motion Estimation", in Motion Analysis and Image Sequence Processing, ED. M. I. Sezan and R. L. Lagendijk, Kluwer Academic Publ., 1993. pp. 1-22.

[4] S. G. Mallat: "Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11. No. 7. July, 1989. pp. 674-693.

[5] S. Zafar, Y. Q. Zhang, B. Jabbari: "Multiscale Video Representation Using Multiresolution Motion Compensation and Wavelet Decomposition", *IEEE Journal on Selected Areas in Comm.*, Vol. 11. No. 1. 1993. pp. 24-35.

[6] Y. Zhang, S. Zafar: "Motion-compensated transform coding for color video compression", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 2. No. 3. 1992. pp. 285-296.

[7] L. Böröczky, P. Csillag: "Multiresolution Motion Estimation with Spatiotemporal Prediction", "Proc of COST # 229 WG 1+2 Workshop on Adaptive Medhods ans Emergent Techiques for Signal Processing and Communications" *Ed. by D. Docampo*, A. R. Figueiras, June 16-18. 1993, Bayona la Real (Vigo), Spain pp. 290-294.

[8] L. Böröczky, P. Csillag, K. Fazekas: "Hierarchical Motion Estimation for Multiresolution Image Coding", submitted paper to the final document of COST # 229 project.

[9] Y. Q. Zhang, S. Zafar: "Predictive Block-Matching Motion Estimator for TV Coding — Part II: Inter-Frame Prediction", *IEEE Trans. on Broadcasting*, Vol. 37. No. 3. 1991. pp. 102-105.

[10] J. N. Driessen, L. Böröczky, J. Biemond: "Pel-Recursive Motion Field Estimation from Image Sequences", *Journal on Visual Communications and Image Representation*, Vol. 2. No. 3. Sept., 1991. pp. 259-280.

[11] J. Biemond, L. Looijenga, D. E. Boekee, R. H. J. M. Plompen: "A Pel-Recursive Wiener-Based Displacemet Estimation Algorithm", *Signal Processing*, Vol. 13. No. 4. Dec., 1987. pp. 339-412.

[12] P. Csillag, L. Böröczky, K. Fazekas: "Predictive Multiresolution Motion Estimation and its Application to Wavelet Image Coding", will be published in Proc of 4th COST # 229 WG. 1 and 2 Worshop on Adaptive Methods and Emergent Techniques for Signal Processing and Communications, Ljubljana, Slovenia, Apr., 5-7. 1994.

MORPHOLOGICAL FILTERS FOR IMAGE PROCESSING

Z. FAZEKAS

DEPT. OF INFORMATION TECHNOLOGY, KFKI RESEARCH INSTITUTE
FOR MEASUREMENT AND COMPUTING
TECHNIQUES
P. O. BOX 49, BUDAPEST, HUNGARY, H-1525

K. FAZEKAS

DEPT. OF MICROWAVE TELECOMMUNICATION
TECHNICAL UNIVERSITY OF BUDAPEST
GOLDMAN TÉR 3. BUDAPEST, HUNGARY, H-1111

1. INTRODUCTION

The theory of multidimensional and multiresolution signal processing is developing at a rapid rate. Video technology has made an explosive progress during the past few years, various image/video coding standards have been established. Multimedia applications and communications are becoming reality. This speedy development requires new methods in the solution of signal processing problems.

In general, the signal decomposition techniques are useful tools for many signal processing problems. The choice of a suitable technique is determined by the point of view of real-rime realization and optimality. Optimality refers here to perfect interband decorrelation and alias-free split simultaneously for multirate signal processing. The decomposition of the signal spectrum into subbands provides the mathematical basis for two important and desirable features in signal analysis and processing. First, the monitoring of signal energy components within the subbands or subspectra is possible. The subband signals can then be ranked and processed independently. Second, the subband decomposition of the signal spectrum leads naturally to multiresolution signal decomposition via multirate signal processing in accordance with the Nyquist sampling theorem.

Multiresolution structural image representation and decomposition schemes typically apply linear filters with progressively increasing spatial extent to generate a sequence of images with progressively decreasing resolution. When linear filters fail to produce satisfactory results, which they often do in image processing applications, the alternative would be to pick a nonlinear filter. Among the nonlinear filter classes which are becoming increasingly popular is the class of morphological filters. Morphological filters are based on the theory of mathematical morphology developed by Matheron and Serra. The filters exploit the geometric rather than the analytic features of signals. Morphological filters have been widely used in digital signal processing for a number of years. They found extensive applications in several areas including image compression coding (e.g. subband coding), biomedical image processing, shape recognition, edge detection, image restoration, image enhancement, etc. Morphological filters are relatively simple and very effective for subband decomposition and directional filtering.

Fundamentally, mathematical morphology represents signals as sets and a morphological operation consists of a set transformation which transforms a set into another set. In this paper, first the mathematical morphological operations are reviewed, and after that subband decomposition applications are shown.

2. MATHEMATICAL MORPHOLOGY

Mathematical morphology provides an effective approach to the analyzing of digital images. The four basic operations in mathematical morphology are erosion, dilation, opening, and closing. Appropriately used, these operations tend to simplify image data while preserving shape characteristics and eliminate irrelevancies. An image can be represented by a set of pixels, the morphological operations deal with two images: the original data to be analyzed and a structuring element, which is analogous to the kernel of a convolution operation. Each structuring element has a shape which can be thought of as a parameter to the operation. Both binary and gray-level images can be processed effectively by morphological operations.

2.1. Binary Image Morphology

Let the original image X and the structuring element B be subsets of the Euclidean n-space E^n. The dilation of X by B is denoted by $X \oplus B$ and is defined as

$$X \oplus B = (c \in E^n | c = a + b \text{ for some } a \in X \text{ and } b \in B)$$

or

$$X \oplus B = \bigcup_{b \in \mathbf{B}} (X)_b,$$

where $(X)_b$ is the notation for a translation of the image X by the vector b. The erosion of X by B is denoted by $X \ominus B$ and is defined as

$$X \ominus B = (c \in E^n | \text{ for every } b \in B, \text{ there exists an } a \in X \text{ such that } c = a - b)$$

or

$$X \ominus B = \bigcap_{b \in \mathbf{B}} (X)_{-b}.$$

The deffinition means that the image is translated by the vector $-b$, and all translated images are ANDed together.

An opening is defined as an erosion followed by a dilation by the same structuring element and is written as

$$X \circ B = (X \ominus B) \oplus B$$

and finally a closing is defined as

$$X \bullet B = (X \oplus B) \ominus B.$$

Openings on an image with a structuring element B can be pictured by moving B inside all the shapes in an image and marking only those places where B fits. Similarly, closings on an image with a structuring element B can be pictured by moving B around the outside of an image with the result that the concave corners are rounded and the convex corners remain square.

2.2. Gray-level Morphology

Morphological concepts can be extended to gray-level images. In the gray-level morphology, gray-level images are visualized as 3D landscapes. The structuring elements of the gray-level morphology are operations on a gray-level image by a spherical structuring element is equivalent to sliding a sphere across the gray-level surface.

Let the image $X(x)$ be represented as a function of coordinates x. The analytical definitions of the basic 3D morphology operations are as follows:

$$X \oplus B = D(x) = \max_{b \in \mathbf{B}} [X(x - b) + B(b)]$$

$$X \ominus B = E(x) = \min_{b \in \mathbf{B}} [X(x + b) - B(b)],$$

where $B(b)$'s are weights that are a function of b.

The opening and closing operations for gray-level images are defined as follows:

$$X \circ B = (x \ominus B) \oplus B$$

$$X \bullet B = (x \oplus B) \ominus B.$$

3. SUBBAND DECOMPOSITION OF IMAGES USING MORPHOLOGICAL FILTERS

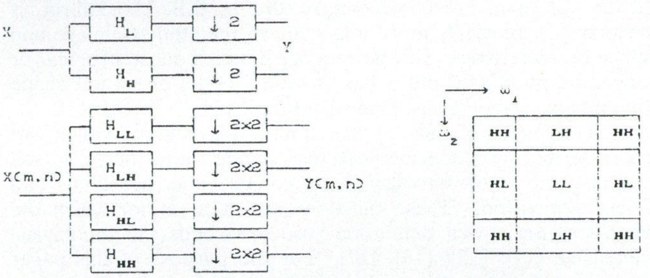
Subband coding of images is a procedure in which a digital source is filtered to decompose it into a desired number of non-overlapping frequency bands. These frequency subbands are each down-sampled, which effectively demodulates them to baseband. This new set of subband sources is coded for digital transmission.

The subband splitting is also used for the solution of compatibility between different image formats of existing video services. By splitting the source signal into several hierarchical video standard layers, a receiver selects and decodes only those layers suitable for its display monitor. A compatible coding system is nothing but a multiresolution coding system. The spatio-temporal spectrum of each standard video signal is a 3D function. This requires a nonrectangular 3D spatio-temporal subband decomposition.

Sequential alternating application of the morphological operations of opening and closing by means of the same structuring element removes details of the image that are small relative to this structuring element. These alternating sequential filters are called morphological low-pass filters. The 1D high-pass filters can be constructed with the original signal X and the complement of the low-pass filtering $[-H_L(X)]$. The 2D analysis/synthesis filters in the case of four-subband splitting are designed by a separable product of the above 1D morphological filters (Table 1).

Table 1.

1 D Filter Bank	2D Filter Bank	
H_{L}: closing [opening (X)]	$H_{\mathrm{LL}}: H_{\mathrm{L}}^{v}[H_{\mathrm{L}}^{h}(X)]$	$F_{\mathrm{LL}}: F_{\mathrm{L}}^{v}[F_{\mathrm{L}}^{h}(Y)]$
$H_{\mathrm{H}}: X - H_{\mathrm{L}}(X)$	$H_{\mathrm{LH}}: H_{\mathrm{L}}^{v}[H_{\mathrm{H}}^{h}(X)]$	$F_{\mathrm{LH}}: F_{\mathrm{L}}^{v}[F_{\mathrm{H}}^{h}(Y)]$
F_{L}: dilation (Y)	$H_{\mathrm{HL}}: H_{\mathrm{H}}^{v}[H_{\mathrm{L}}^{h}(X)]$	$F_{\mathrm{HL}}: F_{\mathrm{H}}^{v}[F_{\mathrm{L}}^{h}(Y)]$
$F_{\mathrm{H}}: Y - F_{\mathrm{L}}(Y)$	$H_{\mathrm{HH}}: H_{\mathrm{H}}^{v}[H_{\mathrm{H}}^{h}(X)]$	$F_{\mathrm{HH}}: F_{\mathrm{H}}^{v}[F_{\mathrm{H}}^{h}(Y)]$



HH	LH	HH
HL	LL	HL
HH	LH	HH

Fig. 1.

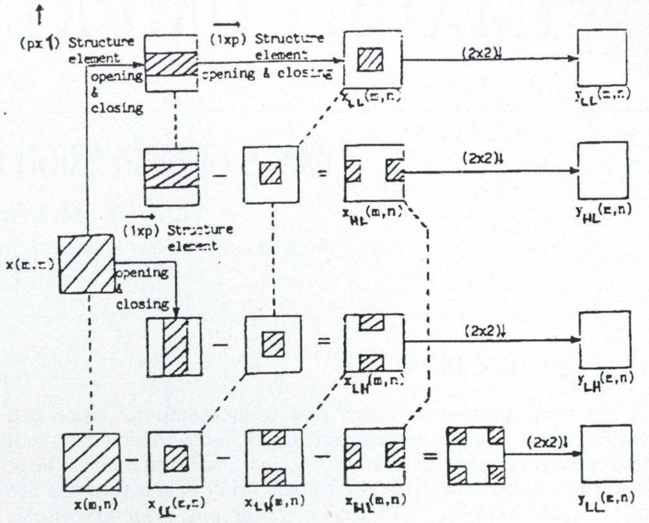


Fig. 2. Block digram of analysis filter bank for four-subband splitting

The block schemes of these filter banks are shown in Fig. 1. and Fig. 2. With mathematical morphology filter technologies, the complexity of the filter design and implementation is greatly reduced.

Because of the limited extent, here only the most trivial example is given. Other details of our work and further examples will be presented at the workshop.

REFERENCES

[1] P. Maragos: "Tutorial on advances in morphological image processing and analysis", *Optical Engineering*, Vol. 26. No. 7. July, 1987. pp. 623-632.

[2] A. Toet: "A morphological pyramidal image decomposition", *Pattern Recognition Letters*, Vol. 9. No. 4. May, 1989. pp. 255-264.

[3] J. Serra (Guest Editor): "Special Issue on Advances in maghematical morphology", *Signal Processing*, Vol. 16. No. 4. April, 1989.

[4] T. W. Pai, J. H. L. Hansen: "An improved image coding algorithm using morphological operator theory", *IEEE Conf.*, CH2977/91. 1991. pp. 2761-2764.

[5] S. C. Pei, F. C. Chen: "Subband decomposition of monochrome and color images by mathematical morphology", *Optical Engineering*, Vol. 30. No. 7. July, 1991. pp. 921-933.

[6] Y. C. Hsueh: "Mathematical morphology on *l*-images", *Signal Processing*, Vol. 26. No. 2. Febr., 1992. pp. 221-241.

[7] F. Y. C.Shin, O. R. Mitchell: "A mathematical morphology approach to Euclidean distance transformation", *IEEE Trans. on Image Processing*, Vol. 1. No. 2. pp. 197-204.

[8] A. C. P. Loui, A. N. Venetsanopoulos, K. C. Smith: "Morphological autocorrelation transform: a new representation and classification scheme for two-dimensional images", *IEEE Trans, on Image Processing*, Vol. 1. No. 3. July, 1992. pp. 337-354.

[9] F. Cheng, A. N. Venetsanopoulos: "An adaptive morphological filter for image processing", *IEEE Trans. on Image Processing*, Vol. 1. No. 4. Oct., 1992. pp. 533-538.

[10] D. G. Daut, D. Zhao: "A flaw detection method based on morphological image processing", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 3. No. 6. Dec., 1993. pp. 389-398.

[11] Q. Wang, M. Gabbouj, Y, Neuvo: "Root properties of morphological filters", *Signal Processing*, Vol. 34. No. 2. Nov., 1993. pp. 131-148.

[12] S. C. Pei, F. C. Chen: "3D spatiotemporal subband decompositions for hierarchical compatible video coding by mathematical morphology", *Image Communication*, Vol. 6. No. 1. March, 1994. pp. 83-99.

IMAGE COMPRESSION, CODING I.

IMAGE COMPRESSION BY FRACTAL GEOMETRY

A. FLAUTO, M. NAPPI, and S. VITULANO

CATTEDRA DI INFORMATICA MEDICA, ISTITUTO DI MEDICINA INTERNA M. ARESU
UNIVERSITA DEGLI STUDI DI CAGLIARI
CAGLIARI, ITALY

1. INTRODUCTION

The main problem in image processing, computer vision and graphics is to handle a great amount of data. This entails not only great amounts of memory to be used but also a huge computing time. Since 16-24 bit color printers and scanners are now readily available, and true color graphic adapters display stunning pictures, and last but not least, hardware prices have been reduced, the main aim of the research teams is now moving toward the design of data compression algorithms with very good performances. In fact a single 800 by 600 pixel true color image requires 1.44 MB of memory, while an uncompressed 10/second video clip with 30 frames/second at 320 by 200 pixels in true color requires an enormous disk space (about 57.6 MB). Nevertheless, a good compression ratio is no more the only qualitative feature of a data compression algorithm.

The great performances provided by optical storage devices such as CD-ROM and CD-I are not enough to support multimedia systems. They also need very good data compression algorithms, possibly integrated into dedicated hardware (printers, plotters, mother-boards).

The data compression techniques are classified into two classes: redundancy reduction and entropy reduction.

A redundancy reduction operation removes the redundancy in such a way that it can be subsequently reinserted into the data. Thus, redundancy reduction is always a reversible process. On the contrary, an entropy reduction operation results in a reduction of information. The lost information will never be recovered, so an entropy reduction operation is irreversible. Codings of the first kind are: Huffman coding [1], [2], Run-Length coding [3], Lempel and Ziv coding [3], etc.. Of the second kind are: Transform coding (Fourier [4]), Cosine [5], [6], [7], Hadamard [4], B-Tree Overlapping [8] and others [9], [10], [15].

The redundancy reduction is error-free (lossless), but it cannot reduce the storage very much; on the other hand, entropy reduction can achieve high compression ratios but it is not error-free (lossy).

The availability of many data compression algorithms gives the user the possibility of choice, but it also means not having an integrated and consistent environment. Today we need a standard methodology accepted all over the world. For this reason, a newer standard is emerging from Joint Photographic Expert Group or JPEG, a standard-setting body established by ISO (*International Standard Office*) and CCITT (*Comité Consultatif International Telegraphique et Telephonique*).

JPEG is a standard methodology for image compression which uses essentially the DCT (*Discrete Cosine Transform*) and also Huffman coding [6], [7]. JPEG's goal is to look for the best methods for image compression among those ones proposed by research centres and to adopt them as an international standard. This method requires $n \log n$ time in the compression and decompression step.

New data compression algorithms come out and are often based on new concepts. One of these is fractal geometry. It has been formulated by B. Mandelbrot [11]. A very interesting fractal geometry application is the one proposed by M. Barnsley using self-similarity. He made the observation that all real-world images are rich in affine redundancy; that is, under suitable affine transformations, large bits of a real-world image look like smaller bits of it. This observation led him to the realization of the fractal-transform process for image compression [15]. Barnsley's method has yielded a compression ratio of more than 10,000:1. However, it requires a huge computing time during the compression process, while the decompression process is completed in a few seconds. In this scenario we'll propose a new algorithm that assures a reasonable compromise between resource requirements (space and time complexity) and the quality of the compressed image.

2. FRACTAL GEOMETRY

In analysing, generating and compressing the complex shapes of natural objects, the fundamental problem is how the surface representing the object is described efficiently. Infact all man-made objects can be described by means a set of simple shape such as spheres, cylinders and so on [12], [13]. Of course, such an approach is not suitable to represent natural and irregular shapes, because it require a huge amount of data and time computing. In the last years Fractal Geometry, due to B.B. Mandelbrot, is becoming increasingly more important in the study of image and shape characteristics. This new theory has introduced irregular or stochastic primitives and it has provided model of natural shape (mountains, planets) based on such primitives.

Fractals are a family of mathematical functions, which are characterised by numerous features; among these there are self similarity, used by Barnsley in its compression algorithms and fractal dimension. These functions are as a set for which the Hausdorff-Besicovich dimension strictly exceeds the topological dimension [14], [15], [16], [17]. For the purpose of this paper we are interested to the irregular shapes and fractal dimension of this object. The fractal dimension is a real number rather than an integer and it measures the "roughness" of the fractal object. For example, a polyline have got a fractal dimension greater than 1 (line dimension) and smaller 2 (plane dimension).

Some result obtained from visual perception experiments are very interesting. After ten naive subject were shown sets of fifteen pictures with varying fractal dimension, these experiments have verified that the mean value of the subject's estimates of perceptual roughness had a nearly perfect correlation with the objects' fractal dimension. The fractal dimension, therefore, could be a discriminating and a topic parameter into human perceptive system for the pattern recognition.

An useful mathematical model for such fractals is a fractional Brownian function (fBf), which is a generalization of classical Brownian motion (fBm) [18], [19], [20]. The fBf $f(x)$ is a real-valued random function such formalized:

$$Pr\left\{\frac{f(x+\Delta x)-f(x)}{||\Delta x||^{H}}<t\right\} \quad \forall x \Delta x, \qquad (1)$$

where **x** represents a point N-dimensional Euclidean space R^N (for a 3D graph surface such as considered an image $N = 2$) and $F(t)$ is a commutative distribution function of a random variable t. The parameter H is a constant and lies in the range [0,1], with a Gaussian distribution. The fractal dimension is determined using the following equation:

$$D = N + 1 - H = 3 - H. \qquad (2)$$

In our model of fBf we assume that the function $F(t)$ is zero-mean Gaussian distribution $N(0, o\sigma^2)$ with variance σ^2 so formalized:

$$F(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-s^2}{2\sigma^2}\right) ds. \tag{3}$$

By the previous discussion we can determine a relation between H and the expected value of the difference of function values over the fixed value distance $||\Delta x||$:

$$E\left[|f(x+\Delta x) - f(x)|\right] \cdot ||\Delta x||^{-H} = C \tag{4}$$

that is equivalent to:

$$\log E\left[|f(x\Delta x) - f(x)|\right] - H \log ||\Delta x||^{-H} = \log C. \tag{5}$$

By means the Eq. (3) we can determine the value C:

$$C = \frac{2}{\sqrt{2\pi}}\sigma. \tag{6}$$

The values H and C are constant and this, considering Eq. (5), suggest that a plot of $E\left[|f(x+\Delta x) - f(x)|\right]$ as a function of $||\Delta x||$ on a log-log scale lies on a straight line. Then H is the slope of the considered straight line.

Related to these considerations a new method of digital image compression has been designed by the authors and is described in the following section.

3. THE PROPOSED METHOD

The proposed method is based on simulation of natural shape by means of statistical information, fractal dimension and few other spatial features. The compression goal is achieved through the selection of a minimal set of information contained in the image. This information must be enough for the image reconstruction by means of stochastic primitives.

The decompressed image is very similar to the original one rather than the same, which means performances and compression ratio very good together with a good quality image.

The compression process runs on two phases: the first provides the deleting of the redundancy contained in the original image using the lossy triangular coding; the second processes the error image, that for its random features can be considered as a fbf, generated by the previous coding and use a powerful solution to the problem of modelling errors as sample path of stochastic processes of one or more variables. During this second step, the best model input values are computed and are inserted in the compressed image together with the triangular code.

The proposed method considers an image as a discrete surface, i.e., a finite set of points in 3D space. Let $F(x,y)$ be a non negative discrete function of two discrete independent variables. The image, as it is illustrated in Fig. 1, can then be considered as the surface $A = \{(x,y,c)|c = F(x,y)\}$, so that for each $P \in A, (x,y)$ represents the projection of P onto the $X-Y$ plane, while c is the pixel's colour (height) [21].

Our goal is to approximate A by a discrete surface $B = \{(x,y,d)|d = G(x,y)\}$, defined by means of a finite set of polyhedrons. Each polyhedron has a right angled triangle (RAT) face on the $X-Y$ plane, three lateral faces orthogonal to the $X-Y$ plane and a RAT upper face approximating A [18]. The surface B is made by the upper faces of the polyhedrons. The triangular coding defines a particular decomposition or segmentation of the original image in regions [22], [23]. The decomposition follows an iterative process. The first step on the iteration subdivides the whole image into two right-angled triangles by drawing the main diagonal.

By means the heights of the vertices of each triangles all points can be generated by a linear interpolation. On both triangles an uniformity predicate is valued with the original one. If the predicate is not verified the triangle is subdivided again. The triangle subdivision is made by drawing the height related to the hypotenuse (so doing right-angled triangles are always created). The topological information related to all subdivision are stored in a B-tree structure [24], [25], [26]. The leaves of the tree store the spatial position of each triangle in the image and the pixel values its vertices. The coding of such a tree is stored in a binary string obtained by a breadth first visit that assign the value 0 to a leaf node and the value 1 to the others; this represents the first part of information stored in the compressed image.

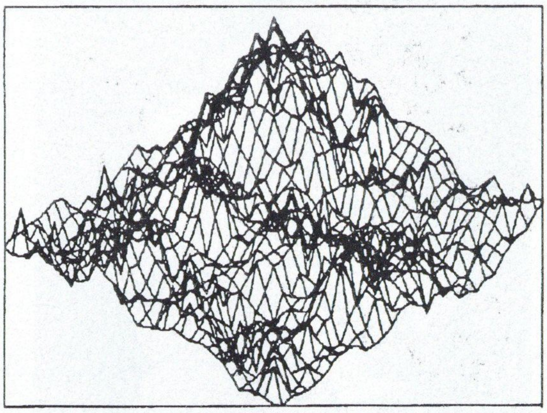


Fig. 1. An image can be considered as a 3D surface

By mean this technique we can remove from the image the regular pattern, i.e. the textures. The error signal is obtained from the difference between input image and triangular coding image.

The second phase is composed of the following two steps:

- Extraction of fractal-based features from the error.
- Interpolation of the error by using the extracted features.

The first step executes a statistical analysis of the error image obtained after the triangular coding and it values the stochastic model input parameters: mean, variance, fractal dimension and some typical point of the error image. These input parameters are not the same for the whole image; really the original image is segmented in region by means of a fractal predicate and the pervious parameters are calculated for each region.

In the second step the stochastic model generates a connection of irregular primitives (one for each region) and in this way it similes the shape being in the error image. This step is applied for all regions, the leaves of the B-tree, that have a significant fractal predicate. At the end of the simulation the quality degree of the decompressed image is values together with the compression ratio. If this two output parameters are not enough good a greater or smaller amount of the typical points of the error image are selected again if we want a better quality image or a greater compression ratio respectively. The stochastic process at the heart of the model used in the second step is the fractional Brownian motion (fBm). The model uses also random number generator. The random numbers are generated having the same probability distribution as the one owned by the signal values to be simulated. The fBm is implemented using the recursive subdivision algorithm. The stochastic model input parameters are stored in the computer image after the triangular code.

The decompression process start with the execution of the inverse triangular coding. In this way a partial image is generated. Then, by means of the input data contained in the second part of the compress image, the stochastic model can run and it generates an error image with the same statistic features as the true error image. The error image generated by the model is added to the partial image obtaining the final decompressed image.

The behaviour of the algorithm is now illustrated by means an example (24 bit true color) relative to a picture representing the Madonna con Bambino of Mantegna. In Fig. 2 the input pictures is illustrated, while in Fig. 3 the output after triangular coding and in Fig. 4 after fractal interpolation with compression ratio=18.0.

The performance of the proposed technique been evaluated by experimenting with a great number of different gray-level and true color pictures. For true color images (RGB format), a particular conversion might be utilized: the image is converted to a colour space with separate luminance and chrominance channels. This is done because the human eye is far more sensitive to luminance information (Y) than it is to chrominance information (Cb and Cr); by separating them, it is possible to compress the chrominance information more than the luminance before the perceived image quality suffers [6].

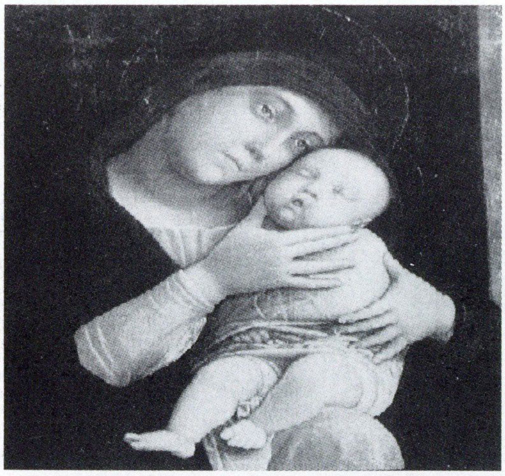


Fig. 2. Original Madonna con Bambino picture

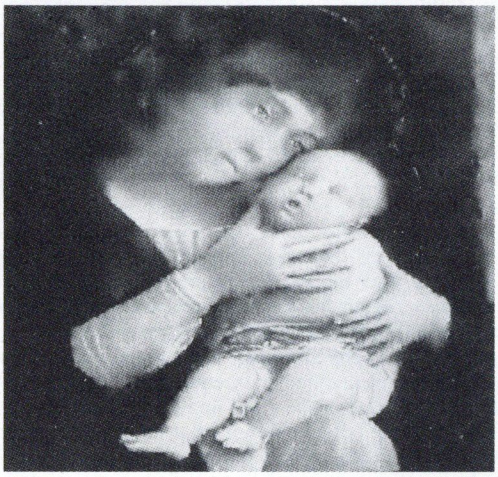


Fig. 3. Madonna con Bambino picture after triangular coding processing

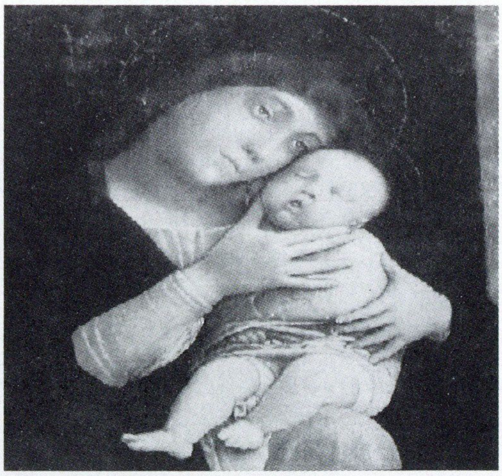


Fig. 4. Madonna con Bambino picture after error reconstruction by fractal method with compression ratio of 18.0

The algorithm has demonstrated to be time efficient. For the compression step, it requires a time proportional to $a \cdot n$, where n is the number of pixels and a expresses the mean number of subdivisions per pixel. An upper bound for a is therefore $\log n$, yelding a worst case time $O(n \log n)$. The decompression step, instead, requires $\Theta(n)$ time to restore the whole image.

REFERENCES

[1] R. G. Gallager: "Information Theory and Reliable Communications", John Wiley, New York, 1968.

[2] R. G. Gallager: "Variations on a Theme by Huffman", *IEE Trans. on Inf. Theo.*, Vol. 14, No. 6. November, 1978.

[3] T. A. Welch: "A Technique for High-Performance data Compression", *Computer*, June, 1984.

[4] R. C. Gonzales, P. Wintz: "Digital Image Processing", Addison Wesley, 1977.

[5] A. Wright: "Image Compression: the DCT approch", *Image Processing*, Winter 1990.

[6] G. K. Wallace: "The JPEG Still Picture Compression Standard", *Communication of the ACM*, April, 1991.

[7] H. S. Malvar: The LOT: "Transform Coding without blocking effects", *IEEE Trans. Acoust. Speech, and Signal Processing*, Vol. 37, pp. 553-559, Apr., 1989.

[8] L. J. Dimento, S. Y. Berkovich: "The Compression Effects of the Binary Tree Overlapping Method on Digital Imagery" , *IEEE Trans. on Communication*, Vol. 38. No. 8. August, 1990.

[9] S. G. Mallat: "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 11. pp. 672-693, 1989.

[10] R. J. Clarke: "Transform Coding of Images", New York: Academic, 1985.

[11] B. B Mandelbrot: "The Fractal Geometry of Nature", W. H. Freeman & Company, San Francisco, 1980.

[12] A. P. Pentland: "Fractal Based Description of Natural Scenes", *IEEE Transaction on Pattern Analysis and Machine Intell.*, No. 6. November, 1984.

[13] K. Falconer: "Fractal Geometry: Mathematical Foundation and Application", John Wiley & Sons, 1990.

[14] A. Fournier, D. Fussel, L. Carpenter: "Computer rendering of stochastic models", *Communication of the ACM*, Vol. 25. No. 6. June, 1982.

[15] M. F. Barnsley, A. D. Sloan: "A Better Way to Compress Images", *BYTE*, January, 1988.

[16] B. B. Mandelbrot: "The Fractal Geometry of Nature", W. H. Freeman and Company, San Francisco, 1980.

[17] A. P. Pentland "Fractal Based Description of Natural Scenes", *IEEE Trans. on PAMI*, Vol. 6. No. 6. November, 1984.

[18] A. Fournier, D. Fussell, L. Carpenter: "Computer Rendering of Stochastic Models", *Communication of the ACM*, Vol. 25. No. 6. June, 1982.

[19] N. Yokoya, K. Yamamoto, N. Funakubo: "Fractal Based Analysis and Interpolation of 3D Natural Surface Shapes and their Application to Terrain Modelling", *Computer Vision, Graphics and Image Processing*, No. 46. pp. 284-302, 1989.

[20] S. S. Chen, J. M Keller, R. M. Crownover: "On the Calculation of Fractal Features from Images", *IEEE Trans on PAMI*, Vol. l5. No. 10. October, 1993.

[21] M. Brady, J. Ponce, A. Yiulk, H. Asada: "Describing Surfaces", *Computer Vision, Graphics and Image Processing*, 32, 1985, pp. 1-28.

[22] Y. F. Wang, J. K. Aggarwal: "Surface Reconstruction and Representation of 3-d Scenes", *Pattern Recognition*, Vol. 19. No. 3. 1986.

[23] C. Lin, M. J. Perry: "Shapes Description Using Surface Triangulation", *Proceedings of the Workshop on the Computer Vision: Representation and Control, IEEE*, New York, 1982.

[24] J. Ponce, O. Faugeras: "An object Centered Hierarchical Representation for Three Dimensional Objects", *Computer Vision, Graphics and Image Processing*, 38, 1987, pp. 1-28.

[25] H. Sameth: "The Quadtree and Hierarchical Structures", *ACM Computer Surveys*, 1984.

[26] H. Sameth, M. Tamminen: "Computing Geometric Proprietes of Images Represented by Linear Quadtree", *IEEE Trans. Pattern Anal. Intelligence*, Vol. 7. No. 2. March, 1985.

ADAPTIVE FRACTAL IMAGE CODING IN THE FREQUENCY DOMAIN

K. U. BARTHEL and T. VOYÉ

INSTITUT FÜR FERNMELDETECHNIK, TECHNISCHE UNIVERSITÄT BERLIN
EINSTEINUFER 25, D-10587 BERLIN, GERMANY

1. INTRODUCTION AND OVERVIEW

The principle of fractal image coding consists in finding a construction rule that produces a fractal image which approximates the original image. Redundancy reduction is achieved by describing the original image through contracted parts of the same image (self transformability). Fractal image coding is based on the mathematical theory of iterated function systems (IFS) developed by Barnsley [1]. Jacquin [2] was the first to propose a block-based fractal coding scheme for gray-level images. In [3] we have shown that the coding performance can be greatly improved by applying a vector quantization to the optimal luminance transformation and using a better geometrical search scheme. In this paper we describe a new luminance transformation in the frequency domain. With this transformation the coding efficiency can be further enhanced. At the decoder fewer iterations are needed to reconstruct the image. In section 2., we briefly present the principle of a block based fractal image coder. An improved codebook design and an adaptive geometrical search scheme are described in section 3. The proposed new luminance transformation is presented in section 4. The description of the new coder can be found in section 5. Finally, in section 6., we present some results and discuss the merits of the new coding scheme.

2. THE PRINCIPLE OF A FRACTAL BLOCK-CODER

The image to be encoded is partitioned into non-overlapping square blocks. $R_{i,j}$ is the image block at the position (i, j) and is called a range block.

The task of a fractal coder is to find a good approximation for all range blocks. Each range block is approximated by a transformed larger block $D_{l,k}$, of the same image (domain block) as shown in Fig. 1.

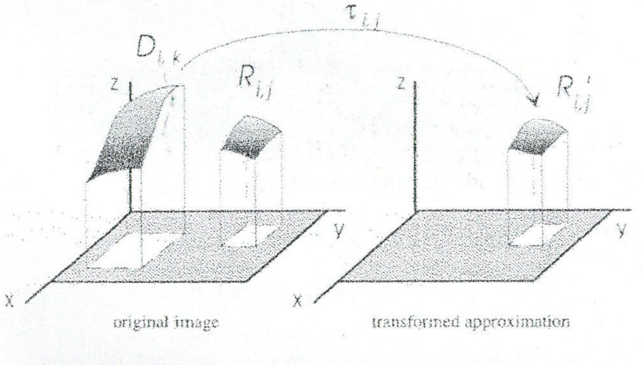


Fig. 1. Approximation of a range block through a transformed domain block

The transformation $\tau_{i,j}$ combines a geometrical transformation and a luminance transformation. The geometrical transformation is an affine linear transformation that consists of a spatial contraction and a position shift that maps the domain block to the position of the range block. The domain block that has been scaled down to the size of the range block is referred to as *codebook block*.

Jacquin proposed a 1st order luminance transformation that scales the dynamic range and changes the brightness of the pixel values of a codebook block.

In matrix form $\tau_{i,j}$ can be expressed as follows:

$$\tau_{i,j}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}k_{11}&k_{12}&0\\k_{21}&k_{22}&0\\0&0&a\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}+\begin{bmatrix}\Delta x\\\Delta y\\b\end{bmatrix}\qquad(1)$$

z denotes the pixel intensity of an image at the position x, y. (a, b, $k_{m,n} \in \mathbb{R}$).

Only the transformations of each range block have to be transmitted to the decoder. The set of all transformations can be seen as the *fractal code* for the original image. This code, iteratively applied to any initial image, generates the reconstructed image. To ensure the convergence at the decoder the transformations $\tau_{i,j}$ have to be contractive. This means:

$$\left|\det\begin{bmatrix}k_{11}&k_{12}\\k_{21}&k_{22}\end{bmatrix}\right|<1 \text{ and } |a|<1.\qquad(2)$$

The process of fractal encoding is lossy. The approximation error ε, that is determined at the coder increases during the decoding process since the decoding process since the codebook block are generated at the decoder from the fractal reconstruction image which is not free of errors. If the scaling factor a is assumed constant, the upper bound for the approximation error after the decoding is given by $\varepsilon/(1-|a|)$.

As the total number of transformations has to be kept low, hierarchical coders with variable range block sizes are used. If the approximation error for a large range block exceeds a given level, this block is split into up to four smaller range blocks for which additional transformations are determined.

For high coding efficiency well chosen coding parameters in combination with efficient coding of the fractal transformation parameters are necessary.

3. GEOMETRICAL TRANSFORMATION

The search for a geometrical transformation can be seen as a search in a codebook that contains the set of contracted domain blocks. Coding efficiency strongly depends on the construction of this codebook. Another important aspect is the order in which this codebook is searched.

When constructing the codebook, the set of all possible affine-linear transformations (Eq. (1): $k_{m,n}$, Δx, and Δy), has to be reduced to a suitable subset. As digital images are sampled images with a given spatial resolution not all affine linear transformations are possible. The size ratio of range to domain block is usually chosen to $1:2$ in x- and y-directions. A smaller contraction ratio allows a better approximation of range blocks but results in a higher error propagation at the decoder. Using higher contraction ratios leads to decreasing similarities between range and codebook block.

To assure contractivity the codebook blocks are generated from the filtered and sub-sampled original image. Jacquin proposed a simple averaging filter. We obtained better coding results using a 10-tap antialiasing filter with a cut-off frequency below $\pi/2$.

We determined an efficient search path by examining the distribution of codebook block positions that yield the best approximation for a given range block. Very often the best codebook block corresponds to the domain block directly above or close to the position of the range block to be encoded. This fact can be used for an optimized adaptive search scheme. The codebook blocks being the most probable are examined first. The search path has the form of a spiral and starts with the codebook block directly above the range block (Fig. 3). By introducing *search regions* a variable length of the search path is possible. The search is aborted at the end of each search region if the approximation error is below a threshold value. This search scheme reduces the encoding time and the average search index. Fig. 2 shows the probability density function of the search indices. In the given example an image was encoded with an 8 bit geometrical codebook as shown in Fig. 3. The entropy of the search indices is reduced if additional smaller search regions with error thresholds are introduced. Note that we use a relative addressing of the codebook blocks and a variable domain blick shift.

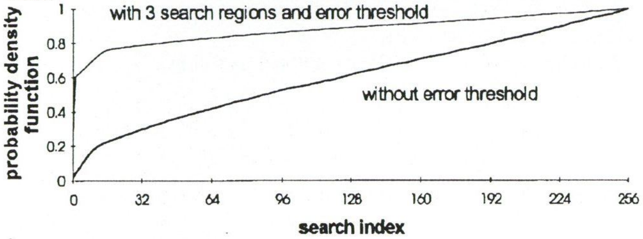


Fig. 2. Probability density function of the geometrical codebook indices using a search scheme as shown in Fig. 3. The maximum search region was 8 bit. The additional search regions used a search width of 0 and 4 bits.

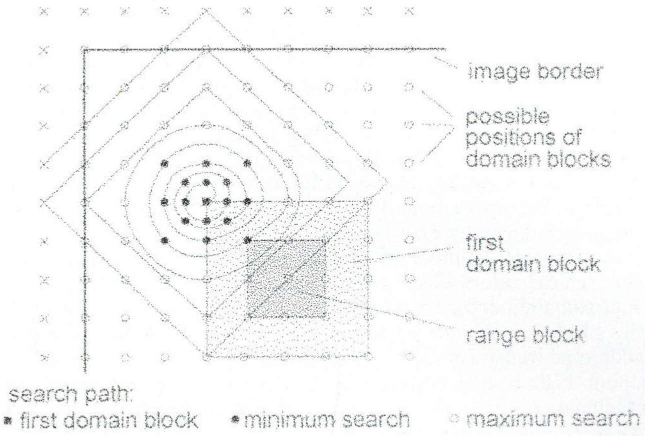


Fig. 3. Adaptive search scheme using a minimum search region

4. LUMINANCE TRANSFORMATION

4.1. Problems of 1^{st} order luminance transformations

The 1^{st} order luminance transformation λ_1 proposed by Jacquin scales the dynamic range (a) and changes the brightness of the pixel values (b) of a codebook block g:

$$\lambda_1(g) = a \cdot g + b. \tag{3}$$

This 1^{st} order transformation has two disadvantages:

- Only small and 'simple structured' range blocks can be approximated well.
- The convergence at the decoder is poor. In particular if a high approximation error is tolerated at the coder, the error propagation at the decoder is very high. In this case the number of iterations necessary to decode the reconstruction image will rise.

4.2. Modified 1^{st} order luminance transformation

Fig. 4 shows the distribution of optimal non-quantized a/b-values obtained from a fractal coder using the conventational 1^{st} order luminance transformation.

From Eq. (3) it can be seen that the b-offset serves to adjust the scaled means of the codebook blocks and is dependent on a. Decreasing a-values generally require increasing b-values. This leads to a triangular shaped a/b-distribution (Fig. 4).

There is a strong accumulation of the a/b-values in the region of a-values near 1. This indicates that the dynamic range of most codebook blocks is kept almost constant. Scaling values close to 1 have the disadvantage that they result in a high error propagation at the decoder. The a/b-distribution and the fact that the means represent the largest energy component of the codebook blocks result in a high upper error bound. Avoiding the scaling of the codebook means by large a-values reduces the error propagation at the decoder.

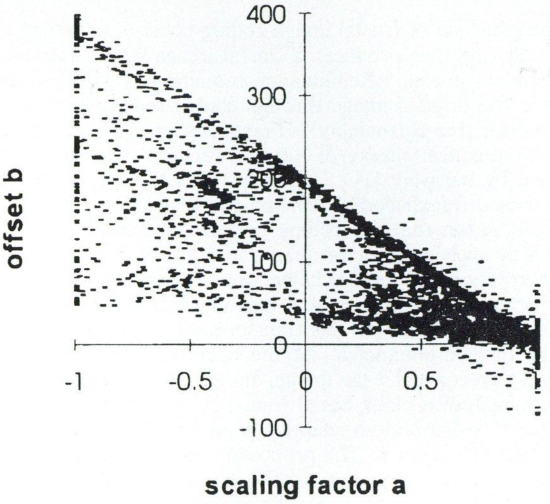


Fig. 4. Distribution of the optimal a/b-coefficients using the conventional 1^{st} order luminance transformation

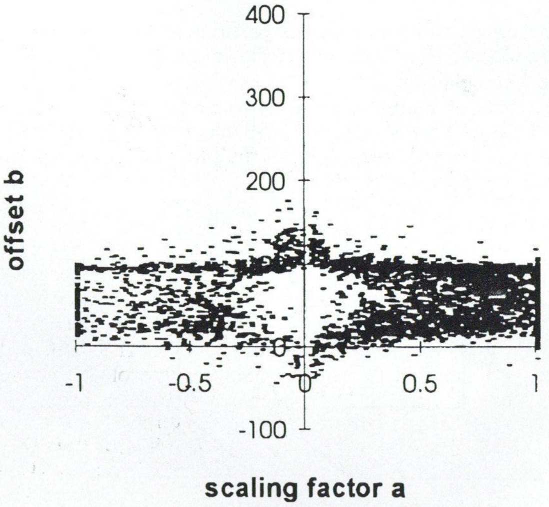


Fig. 5. Distribution of the optimal a/b-coefficients using the modified 1^{st} order luminance transformation

We propose a simple modification of the luminance transformation. We decorrelate the a/b-values by only scaling the dynamic part of the codebook blocks. With this modified transformation a similar approximation of the range blocks is possible. With a well-chosen factor a_0 a lower upper error bound at the decoder can be achieved.

$$\lambda_{1\text{mod}}(g) = a \cdot (g - \mu_g) + a_0 \cdot \mu_g + b. \qquad (4)$$

The constant factor a_0 can be chosen from 0 to 1 and is found as a compromise: For $a_0 = 1$ the variance of the a/b-coefficients reaches its minimum, but the luminance transformation is not contractive anymore. If a_0 is set to 0 we obtain a minimal error propagation and a minimal decoding time at the decoder. In this case the variance of the a/b-distribution is maximum. Our studies have shown that with quantized a/b-coefficients the best coding results are reached for $a_0 = 0.5$. Fig. 5 shows the distribution of optimal non-quantized a/b-values of the *modified* 1^{st} *order luminance transformation*.

Fig. 6 compares the convergence at the decoder for a critical part of the "Clown"-image. We compare the conventional and the modified luminance transformation with quantized and non-quantized coefficients. It can be seen that the modified luminance transformation outperforms the conventional transformation in the reconstruction error and the number of iterations needed. Decoding examples are shown in Fig. 7. The significance of the artifacts as shown in Fig. 7b is image dependent and they only occur with quantized parameters. By using the modified luminance transformation these artifacts can always be avoided.

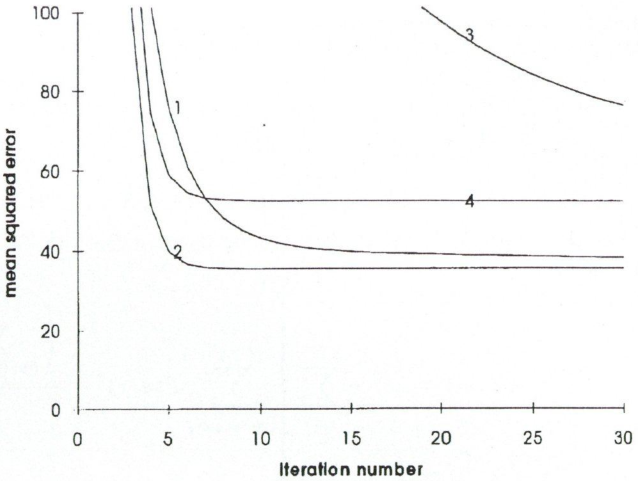
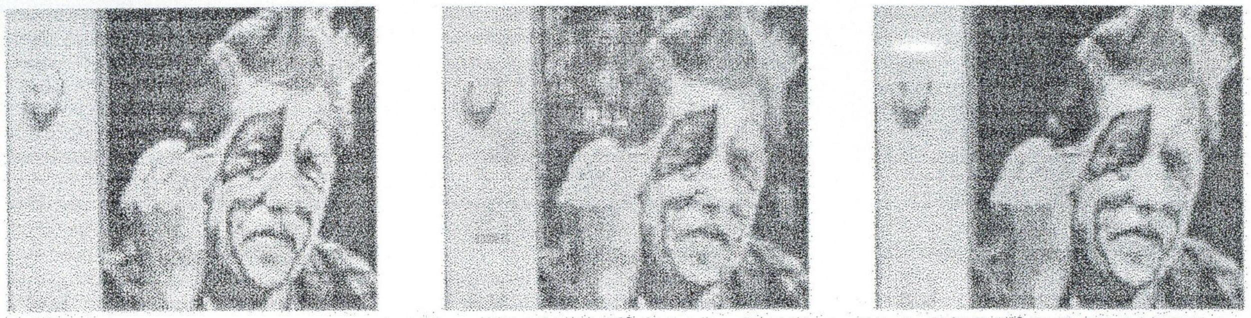


Fig. 6. Comparison of the conventional and the modified 1^{st} *order luminance transformation: conventional: 1 non-quantized, 3 quantized, modified: 2 non-quantized, 4 quantized*



a) original

b) conventional 1^{st} order luminance transformation (10^{th} iteration)

c) modified 1^{st} order luminance transformation (10^{th} iteration)

Fig. 7. Original and decoded images using the conventional and the modified 1^{st} *order luminance transformation with quantized* a/b*-values. a) original; b) conventional* 1^{st} *order luminance transformation* (10^{th} *iteration); c) modified* 1^{st} *order luminance transformation* (10^{th} *iteration)*

4.3. High order luminance transformations in the frequency domain

Any improvement in the approximation of range blocks will improve the image quality and can reduce the total number of transformations needed to describe the fractal approximation of the image to be encoded.

One approach to do this is to use additional 'basic codebook blocks' [4], such as simple polynominal blocks. We feel that such an approach is not very promising because these simple blocks are easy to encode with the fractal coder itself.

Another possibility is the use of squared and cubic scaling of the pixel intensities of the codebook blocks. The optimal scaling parameters are difficult to determine because of the dependency of the parameters on each other. A further problem is to guarantee the contractivity of such a transformation.

A high order luminance transformation has to fulfill the following conditions:

- To enable their individual adaptation the transformation coefficients should be independent of each other.
- To assure a control of the contractivity, the requirements for the contractivity should be controllable independently by the transformation coefficients.

Our proposal for a high order luminance transformation is an extension of our modified 1^{st} order luminance transformation:

First we transform all range and codebook blocks via the discrete cosine transform (DCT). In the frequency domain we obtain the energy compacted spectra of range and codebook blocks. Then by individually setting or scaling the spectral values of the codebook block $G(u,v)$ we can approximate the spectrum of the range block $F(u,v)$.

$$\lambda(g) = IDCT\left(\bigcup_{u=0}^{N-1}\bigcup_{v=0}^{N-1} a(u,v)\cdot G(u,v) + b(u,v)\right)$$

$$G(u,v) = DCT(g), \quad f(u,v) = DCT(f), \qquad (5)$$

whereby N denotes the size of the blocks, the IDCT is the inverse DCT.

Many coding schemes are possible using subsets of this *general luminance transformation* (5). If all spectral values were set or scaled, the number of transformation parameters to be transmitted would increase drastically. However, many range blocks can be approximated with low order luminance transformations.

In this paper we propose a coding scheme using one or more scaling factors for the dynamic part of the codebook spectrum. For a luminance transformation of order K we merge subsets of the spectral values to non-overlapping regions $\mathbf{R}_1$ to $\mathbf{R}_K$. The mean is approximated the same way as with the modified 1^{st} order luminance transformation.

If using a 1^{st} order luminance transformation, all dynamic coefficients are scaled with only one scaling factor (a_1). A 2^{nd} order luminance transformation has got three regions, so the dynamic part of the spectrum is scaled with two coefficients (a_1, a_2). For luminance transformations of order 2 and higher various *frequency domain partitions* are possible. Fig. 8 shows some examples of partitions for 2, 3, and 4 regions.

$$\lambda_K(g) = IDCT\left(\bigcup_{u=0}^{N-1}\bigcup_{v=0}^{N-1}\begin{cases} a_0 \cdot G(u,v) + b & \text{if } u=0, v=0 \\ a(u,v) \cdot Gu, v & \text{else}\end{cases}\right)$$

$$a(u,v) = a_i \quad if \quad (u,v) \in \mathbf{R}_i \quad i =, \ldots, K. \tag{6}$$

For every region $\mathbf{R}_i$ the optimal scaling factor $a_{\mathrm{i\,opt}}$ can be evaluated:

$$a_{i\,opt} = \frac{\sum_{(u,v)\in R_i} \{G(u,v) \cdot F(u,v)\}}{\sum_{(u,v)\in R_i} G(u,v)^2}$$

$$b_{opt} = F(0,0) - a_0 \cdot G(0,0). \tag{7}$$

To assure contractivity $a_{i\,opt}$ must not exceed a value of 1. a_0 again is set to 0.5. If the a_i/b-parameters are limited or quantized, we get an approximation error plus a quantization error:

$$e(\mathbf{F},\mathbf{G}) = \Delta b^2 + \sum_{i=1}^{K}\left[\sum_{(u,v)\in R_i} F(u,v)^2 - \frac{\left(\sum_{(u,v)\in R_i} F(u,v)G(u,v)\right)^2}{\sum_{(u,v)\in R_i} G(u,v)^2} + \Delta a_i^2 \cdot \sum_{(u,v)\in R_i} G(u,v)^2\right]$$

$$\Delta a_i = a_{i\,opt} - a_i \quad \Delta b = b_{opt} - b. \tag{8}$$

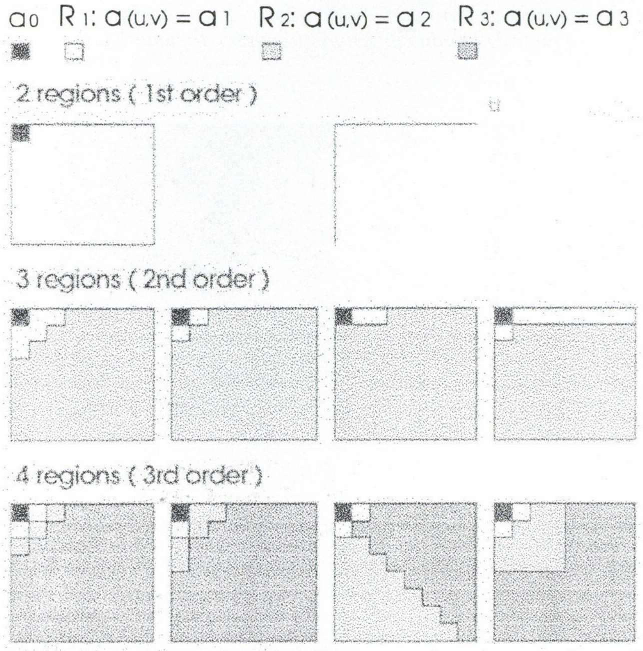


Fig. 8. Examples of different frequency domain partitions for luminance transformations of 1^{st}, 2^{nd}, *and* 3^{rd} *order (block size* 8×8*)*

5. DESCRIPTION OF THE CODER

Before describing our new coding scheme, we suggest some important modifications of the conventional fractal coding scheme.

5.1. Improvements of the fractal coding scheme

Partial approximation

To reduce the total number of transformations generally hierarchical coding schemes with variable range block sizes are used. In a first step, transformations for the largest range blocks of the highest hierarchy level are determined. If the approximation error is too high for any of the four range blocks of the next hierarchy level, the large range block containing these smaller blocks is split into sub blocks. For these sub blocks additional transformations are determined. The transformation for the large range block is kept if the number of additional transformations does not exceed two.

The total number of transformations can be significantly reduced if new transformation parameters are determined for the remaining part of the large range block. The coding procedure for large range blocks can then be described as follows: The transformation of a large range block and the resulting errors in the sub blocks are determined. The sub block that is responsible for the highest error component is excluded and a new transformation for the remaining 3/4-*block* is searched. If necessary, this procedure is repeated for the 3/4-block and leads to a 1/2-*block*. Many large range blocks that were totally split using the conventional scheme can now be coded as 1/2- or 3/4-blocks.

Codebook-update and a/b-update

One problem of fractal image coding is the error propagation at the decoder. It results from the fact that at the decoder the codebook is generated from the reconstructed image whereas at the coder the codebook is generated from the original image. The error propagation at the decoder can be reduced if the coder codebook is updated with the coded versions of the range blocks.

The *a/b-update* is comparable to the codebook-update. At the end of the coding process the best possible approximation of the original image is known. Now the coder could start coding the image again and again getting a better and better approximation of the decoder codebook. As this increases the coding time we propose to keep the geometrical transformations, but to redetermine the best a/b-values. The a/b-update can be repeated. We found that 1 to 2 a/b-updates are useful.

Using the modified luminance transformation and the described update procedures, the error propagation can be reduced and a slightly higher coding efficiency is obtained. The increase of the decoding error can be reduced to approximately $1-4$ % of the coding error compared to more than 10 % using the conventional.

5.2. Coder description

We use a coder with a three level hierarchy with range block sizes of 16×16, 8×8, and 4×4 pixels. For the quantization of the luminance transformation we apply a vector quantization (VQ) technique. We use an adaptive search algorithm to determine the order of the luminance transformation and the search region that is used for the geometrical transformation.

For each hierarchy level we define a set of *search classes*. A search class contains a fixed search region and a luminance transformation with fixed order and VQ-codebook size. These search classes are searched successively. If the approximation error after searching one class fulfills a given *search stop criterion* (error threshold) the search is aborted, otherwise the next search class is examined. To obtain good coding efficiency the bit costs are increased during the search. This assures to encode a range block with the lowest necessary rate. Simulations have shown that it is useful to increase both the VQ-codebook size and the search width.

If even with the maximum search class no good approximation can be found then this transformation is rejected and additional transformations are determined using the partial block approximation. For smaller range blocks this scheme is repeated until the highest search class of the lowest hierarchy level is reached. As the splitting criterion we check all errors of the smallest range block size.

The advantage of this coding scheme is that we can locally adapt the bitrate to the image contents. There is no classification of the range blocks done before the coding process. Fig. 9 shows the search classes of the lowest hierarchy level (block size 4 × 4 pixels). A complete set of coding parameters is shown in Table 1.

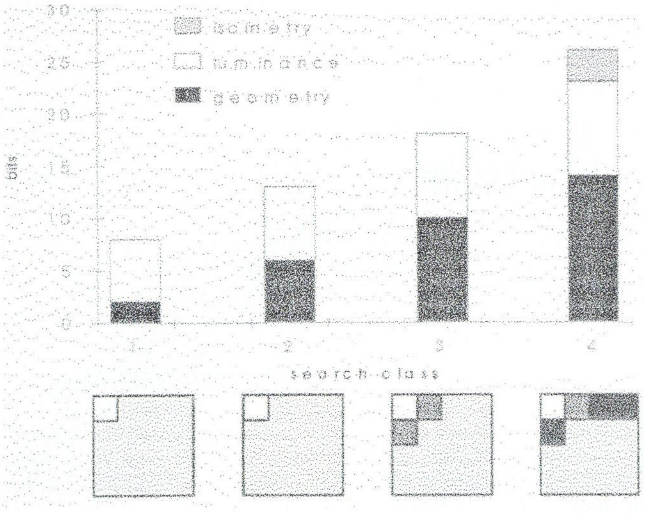


Fig. 9. Adaptive search scheme combining geometrical and luminance transformation (search classes 1 to 4). The applied frequency domain partitions of the luminance transformations are shown.

Parameters to be transmitted to the decoder are:

- splitting partition of range blocks of the higher hierarchy levels,
- search class,
- geometrical index of the codebook block and the isometry (if used) and
- codebook index of the luminance transformation VQ (scaling factors a_1 to a_K and the offset b). (The search class of a block and the splitting partition are entropy-coded.)

The image quality respectively the bitrate can be controlled over a large range by only adjusting the error thresholds. For very low bitrates however, the block sizes have to be enlarged to 32 × 32, 16 × 16, and 8 × 8 pixels.

Table 1. Coding parameters used for the coding results shown in Table 2.

hierarchy level	block size	search class	split error threshold	search stop error threshold	*luminance transformation*		*geometrical transformation*
			[msq]	[msq]	order	VQ codebook size [bits]	codebook size (search) [bits]
3	16	*1*	60	15	1st	8	3
		2			1st	8	7
2	8	*1*	90	30	1st	8	3
		2			1st	8	7
1	4	*1*	-	120	1st	6	2
		2			1st	6	7
		3			2nd	8	10
		4			3rd	9	14 + 3(isom.)

Table 2. Coding results for the LENA image (512 × 512 pixels).

level and search class	classification [bits]	partition [bits]	geometry [bits]	luminance [bits]	sum [bits]	number of blocks	product [bits]
3__x	2	-	-	-	**2**	350	**700**
3_1	2	-	3	8	**13**	313	**4069**
3_1x	2	3.13	3	8	**16.13**	148	**2387**
3_2	3	-	7	8	**18**	68	**1224**
3_3x	3	3.13	7	8	**21.13**	145	**3063**
2__x	2	-	-	-	**2**	378	**756**
2_1	2	-	3	8	**13**	517	**6721**
2_1x	3	3.22	3	8	**17.22**	236	**4063**
2_2	2	-	7	8	**17**	394	**6698**
2_2x	3	3.22	7	8	**21.22**	340	**7214**
1_1	1	-	2	6	**9**	1301	**11709**
1_2	2	-	7	6	**15**	461	**6915**
1_3	3	-	10	8	**21**	371	**7791**
1_4	3	-	17	9	**29**	225	**6525**
						total :	**69835**
				PSNR = 33.45 dB		0.266 bpp	

k__x : level *k*, block is totally split;
k_j : level *k*, search class *j*, block is not split;
k_jx : level *k*, search class *j*, block is partially split (the splitting partition is additionally coded)

6. SIMULATION RESULTS AND CONCLUSION

We have proposed a new block-oriented fractal coding scheme using an adaptive search scheme with an extended luminance transformation in the frequency domain. This transformation is able to better approximate codebook blocks to range blocks and has a better convergence at the decoder.

The bitrate is reduced because fewer transformations are needed to describe the fractal approximation of the image to be encoded. The subjective quality of images coded with our new scheme is superior compared to conventional fractal coded images. Blocking artifacts are reduced and detailed structures are better preserved.

In our simulations we used a hierarchical fractal coder with variable block sizes. Our results show that the 'LENA'-image (512 × 512 pixels) can be coded at the rate of 0.1 bpp to yield a peak-to-peak SNR of 30 dB. Fig. 9 shows the coder performance compared to JPEG.

Due to the high number of parameters detailed investigations are needed to achieve optimal coding efficiency. With optimized parameters and better codebooks for the luminance transformation further improvements are to be expected.

Many different coding schemes are possible using the general luminance transformation expressed in Eq. (5). A new efficient coding scheme unifying fractal and transform coding will be presented in a further publication.

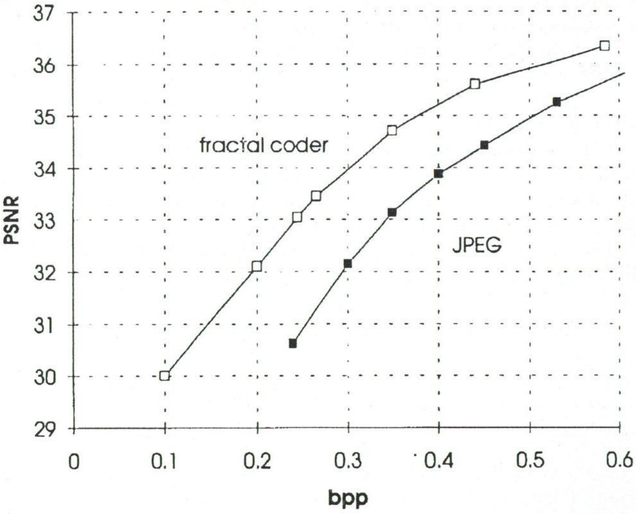


Fig. 10. Coding results of the new fractal coding scheme compared to JPEG. (Image: LENA 512 × 512 pixels)

REFERENCES

[1] M. F. Barnsley: "Fractals Everywhere", New York, Academic Press, 1988.
[2] A. Jacquin: "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transforms", *SPIE*, Vol. 1360, Visual Communications and Image Processing'90.
[3] K. U. Barhel, T. Voyé, P. Noll: "Improved Fractal Image Coding", *Proceedings of PCS'93*, section 1.5.
[4] M. Gharavi-Alkhansari, T. S. Huang: "A Fractal-Based Image Block-Coding Algorithm", *Proceedings ICASSP'93*, V. 345-348.
[5] E. W. Jacobs, Y. Fisher, R. D. Boss: "Image Processing. A study of the iterated transform method", *Signal Processings 29* (1992), pp. 251-263.

3D MEDICAL DATA COMPRESSION*

L. KONDIS, C. CHRISAFIS, H. SAHINOGLOU, and M. G. STRINTZIS

INFORMATION PROCESSING LABORATORY, ARISTOTLE UNIVERSITY OF THESSALONIKI
THESSALONIKI 54006, GREECE

1. LOSSLESS 3D MEDICAL DATA COMPRESSION

The increasing demand for storage and transmission of medical images has made data compression essential. There are two types of image compression: (1) lossless, or reversible compression, and (2) lossy, or irreversible compression. The former implies that no information is lost after the reconstruction of the image, while the latter implies that the compressed image cannot be exactly reconstructed and that some information is lost. Lossy image compression can produce compression ratios that are about five times as high as those that lossless image compression can produce, with very little loss in image quality [1]. However, in some cases, lossless compression of medical images is required.

The methods presented here are: Differential Pulse Code Modulation (HINT) using autocorrelation model, DPCM using computed autocorrelation and Hierarchical Interpolation (HINT).

In the methods presented here, the voxel values are first decorrelated to remove the statistical redundancy in the image. The decorrelated data are then encoded using a variable length coder. In this paper we will focus on decorrelation.

Generally [1], decorrelation methods can be divided into three classes:

- transform decorrelation methods,
- predictive decorrelation methods,
- multiresolution decorrelation methods.

Transform decorrelation methods are best suited for lossy compression and they do not perform satisfactorily in lossless decorrelation. Transform decorrelation methods will not be covered here.

Multiresolution decorrelation methods are also unsuited for lossless compression except for hierarchical interpolation (HINT) which will be covered here. Predictive decorrelation methods perform well in lossless compression.

The most widely used predictive decorrelation method is differential pulse code modulation (DPCM).

1.1. *Differential Pulse Code Modulation (DPCM)*

DPCM using Autocorrelation model

This method estimates the value of a voxel by using a weighted sum of the previously coded voxels. If $f(i,j,k)$ represents the original image, the prediction in 3D DPCM is given by the equation:

$$\hat{f}(i,j,k) = \sum_{p,q,r \in W} a(p,q,r) f(i-p, j-q, k-r), \tag{1}$$

where W is the prediction window which defines the previously coded elements that are used in the estimation. The differential image $u = f - \hat{f}$ is transmitted where $\hat{f}$ is rounded to the nearest integer.

If the image is assumed to be a stationary random field with zero mean and an exponentially decaying autocorrelation function of the form:

$$R(k,l,m) = \sigma^2 \rho_1^{|k|} \rho_2^{|l|} \rho_3^{|m|} \tag{2}$$

the optimum m.s linear prediction (the one that minimizes the variance of the differential image) is:

$$\begin{aligned}\hat{f}(i,j,k) =& \rho_1\rho_2\rho_3 f(i-q, j-1, k-1)\\ &- \rho_1\rho_2 f(i-1, j-1, k)\\ &- \rho_2\rho_3 f(i, j-1, k-1)\\ &- \rho_1\rho_3 f(i-1, j, k-1)\\ &+ \rho_1 f(i-1, j, k)\\ &+ \rho_2 f(i, j-1, k) + \rho_3 f(i, j, k-1)\end{aligned} \tag{3}$$

as easily seen by application of the orthogonality principle [2].

DPCM using computed autocorrelation

In general, if $R[k,l,m,]$ is computed using

$$\hat{R}(p,q,r) = \begin{cases} \frac{1}{MNL} \sum_{m=0}^{M-1-p} \sum_{n=0}^{N-1-q} \sum_{l=0}^{L-1-r} f(m+p, n+q, l+r) f(m,n,l), & \text{for } p \geq 0, q \geq 0, r \geq 0 \\ \frac{1}{MNL} \sum_{m=0}^{M-1-p} \sum_{n=0}^{N-1-q} \sum_{l=-r}^{L-1} f(m+p, n+q, l+r) f(m,n,l), & \text{for } p \geq 0, q \geq 0, r < 0 \\ \frac{1}{MNL} \sum_{m=0}^{M-1-p} \sum_{n=-q}^{N-1} \sum_{l=0}^{L-1-r} f(m+p, n+q, l+r) f(m,n,l), & \text{for } p \geq 0, q < 0, r \geq 0 \\ \frac{1}{MNL} \sum_{m=0}^{M-1-p} \sum_{n=-q}^{N-1} \sum_{l=-r}^{L-1} f(m+p, n+q, l+r) f(m,n,l), & \text{for } p \geq 0, q < 0, r < 0 \\ \hat{R}^T(-p,-q,-r), & \text{for } p < 0, \text{ every } q-r, \end{cases} \tag{4}$$

then the coefficients a of

$$\hat{f}(i,j,k) = \sum_m \sum_n \sum_l a(m,n,l) f(i-m, j-n, k-l) \tag{5}$$

*This material is based on work supported by the RACE 2045-DISTIMA project.

satisfy

$$R(p,q,r) = \sum_m \sum_n \sum_l a(m,n,l) R(p-m, q-n, r-l). \quad (6)$$

1	5	3	5	1	5	3	5	1
5	4	5	4	5	4	5	4	5
3	5	2	5	3	5	2	5	3
5	4	5	4	5	4	5	4	5
1	5	3	5	1	5	3	5	1

Fig. 1. Pixel classification scheme for 2D hierarchical interpolation. The block size is 4×4.

If m, n, l take the values 0 or 1, $(m,n,l) \neq (0,0,0)$, Eq. (6) leads to a 7×7 linear system of equations with $a(m,n,l)$ to be the unknowns.

This method can be easily extended to 4 dimensions, using 15 prediction coefficients and thus leading to a 15×15 linear system of equations.

In storing or transmitting of medical images, the prediction coefficients have to be quantized and transmitted or stored along with the differential image.

1.2. Hierarchical Interpolation (HINT)

Table 1. Comparison of 2D and 3D decorrelation methods

Method	Entropy	Entropy after Arithmetic Coding
Original Entropy	3.65	3.55
2D DPCM using autocorrelation model	3.54	3.46
2D Adaptive DPCM	3.28	3.31
2D DPCM using computed autocorrelation	3.23	3.25
2D HINT (4×4)	3.26	3.29
2D HINT (8×8)	3.26	3.29
3D DPCM using autocorrelation model	3.34	3.29
3D DPCM using computed autocorrelation	3.17	3.13
3D HINT ($4 \times 4 \times 4$)	3.18	3.19
3D HINT ($8 \times 8 \times 8$)	3.18	3.19

Hierarchical interpolation [1], [3], [4], is easily explained and understood by referring to Fig. 1 (2D version). First, the "1" pixels are transmitted or stored using a method like DPCM. Thus, a low-resolution version of the original image is obtained. Then, the "2" pixel values are estimated from the "1" pixels by linear or median interpolation. The estimates are rounded to the nearest integer and subtracted from the original pixel values; the differences are transmitted or stored. In the following steps, the "3", the "4" and the "5" pixels are estimated from the previously coded surrounding pixels and the differences are stored or transmitted.

The reconstruction is done in a similar way. The "1" elements are first decoded and the "2" elements are estimated from them. The actual values are obtained by adding the corresponding differences to the estimates. The values of the remaining pixels are recovered in a similar way.

Fig. 1 shows the pixel classification with a 4×4 block size. A 8×8 block is also used. This method is free of parameters and that gives it an advantage over DPCM.

Hierarchical Interpolation can be extended to take advantage of the correlation between slices [1]. A $4 \times 4 \times 4$ block size is used. The 3D data are first subsampled by a factor of 4^3. Then, in a way similar to the 2D case, the rest of the voxels are estimated by 2D and 3D interpolation of the previously coded voxels. A $4 \times 4 \times 4$ block size may also be used.

1.3. Experimental Results

An MRI multislice image of the head was used to evaluate the various decorrelation methods. The slices were filtered to remove noise. The distance between slices was 1mm. In Table 1 the performance of the examined methods is tested. There were 8 quantization levels (bits) and the size was $256 \times 256 \times 33$. 2D decorrelation methods were also used. DPCM and HINT can be modified to be used with 2D images. Adaptive DPCM [3] was also used. In 2D methods, each slice was treated as an individual image. In the 2D DPCM using computed autocorrelation, a set of coefficients was evaluated for each slice. In practice, this will pose a problem since too many coefficients will have to be quantized and transmitted. However, this method performs clearly worse than the 3D version in which only 7 coefficients have to be quantized and transmitted. Thus, in practice, the 3D version is preferred. As seen, the best methos indicated by these results is the 3D version of DPCM using computed autocorrelation with $4 \times 4 \times 4$ 3D HINT as second best. In all cases, the error image was coded using arithmetic coding.

2. LOSSY COMPRESSION FOR PROGRESSIVE TRANSMISSION OF 3D DATA

2.1. Description of the method

A novel wavelet transform is defined for the compression of 3D data. This transform is based on nonseparable wavelets whose form allow the construction of a very efficient algorithm, requiring only 2 additions and a division by 2 for each pixel. This can be achieved without any multiplications, allowing for a very efficient implementation and giving a significant speed advantage over encoding algorithms.

The algorithm has, by nature, a multiresolution form that can be exploited to achieve a progressive transmission effect. In the algorithm implementation therefore after the transform the coefficients are quantized with level-dependent thresholds and transmitted.

The second advantage of this algorithm is that the global topology of the inputs needed to produce the coefficients of a pixel, lend themselves to the creation of tree structures that partition the pixels of the image. These tree structures are found to have the property that the majority of the zero coefficients are grouped under common subtrees. This allows a progressive transmission scheme where for each coefficient we also transmit the information of whether it is root of a zero subtree. Correspondingly, coefficients that belong to zero subtrees rooted higher levels are omitted from transmission. This means that we need not transmit zero information for almost all zero coefficients, that cover all but few of the pixels. This allows a significant increase in coding efficiency when compared to independent level by level encoding of the coefficients.

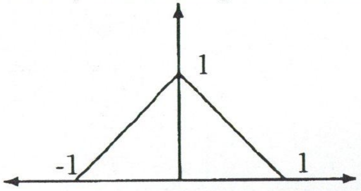


Fig. 2. The basic 1D wavelet

Experimental results show that reasonably good compression rates can be achieved. A compression of about 10 for 30 db PSNR is achieved, even when the values of the quantized coefficients are transmitted using PCM. Greater gains are possible when entropy coding methods, such as arithmetic coding is employed for transmission of the coefficients. This is a result of the ease of transmitting the zero quantized coefficients, that are the majority of the coefficients transmitted.

An additional advantage of the method is that it can be extended to cover any number of dimensions. Thus it is straightforward to adapt the method to cover the case of moving MR image, by employing the 4D version of the algorithm. The advantages that are available in the 2D and 3D cases are also available in this case as well.

2.2. The Wavelet expansion form

The orthogonal wavelet representation [19] of a signal is based on translations and dilations of a single scaling function $\Phi(x)$. In general a function $f(x)$ can be approximated by a series of functions of the form $\frac{1}{\sqrt{a}}\Phi\left(\frac{x-a}{a}\right)$. In general this form is overcomplete, with more coefficients than signal points. This happens because several wavelets, each with different dilation, can be centered in the same point. In the case discussed here though, the high-speed pyramid discussed for 2D in [20], we have a complete wavelet representation, by allowing at each point only the wavelet with the lowest dilation to have a non-zero value. For discrete k-D functions the expression becomes:

$$x(\overrightarrow{m}) = \sum_{\overrightarrow{n} \text{ in the support of } x} \phi_{\overrightarrow{n}}(\overrightarrow{m})\alpha(\overrightarrow{n}), \tag{7}$$

where $\phi_{\overrightarrow{n}}$ are reconstruction discrete wavelets formed by a continuous wavelet by dilations and translations, and the dilation is determined by $\overrightarrow{n}$. By the appropriate choice of the reconstruction wavelets the evaluation of the coefficients $\alpha(\overrightarrow{m})$ can be done very fast in some cases, as discussed in [8]. The reason the method is called the highspeed pyramid is that its implementation is done with a very fast algorithm that has a pyramid form, which gives pyramid coefficients α of the form:

$$\alpha^k(\overrightarrow{n}) = \begin{cases} 0 & \text{if all elements of are even} \\ \alpha(2^{b-k}\overrightarrow{n}) & \text{otherwise.} \end{cases} \tag{8}$$

To completely specify how the method works, it suffices to choose an appropriate function $\Phi(x)$. The one used here is the triangular wavelet defined as:

$$\Phi(x) = \begin{cases} 1-|x| & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

shown in Fig. 2. This relation is very important because it is the central in the development of the fast algorithms. Note that this wavelet is also considered in [21] for the pyramid method developed there.

2.3. An overview of pyramid methods

A pyramid method [20], [13], [15] is based on successive approximations of an image with a set of other images of lower resolutions, forming a pyramid. The main advantage of that form of representation of an image is its ability to implement progressive transmission of an image, which is profitable in such situations where we are interested on just a characteristic of an image and not the whole image. Suppose $or\ g$ is the original image, pyr_k is the k level of a pyramid having $b+1$ levels and pyr_b is the level with the higher resolution, then the most general form of a pyramid is:

$$pyr_k = REDUCE[pyr_{k+1}]$$

$$\hat{pyr}_{k+1} = EXPAND[pyr_k], \tag{10}$$

where $\hat{pyr}_k$ is an estimation of the k level pyramid (notice that is not the image itself). All multiresolution approaches are based in the above principles. The critical point about it, is the appropriate selection of the functions REDUCE, EXPAND. The above methods have been used mainly in 2D data, but they can very easily expanded in 3D data. A 3D multiresolution method can either be separable or not.

2.4. Relation to the Laplacian pyramid

The Laplacian pyramid [21] the best known pyramid method. It is a separable approach in multiresolution representations so it can be easily extended in 3D (as done, for instance in [22]). For this reason the one dimensional version will be described.

Consider first the reduction method. This method uses a filter followed by subsamplings. The filter has a parameter α that can be any value in the interval (0, 1). The operation is repeated on each level, until the lower resolution is produced. The expansion method interpolates first with zeros and then uses the same filters that the reduction did. Adding the appropriate error then recovers the original image. Then method used to encode the coefficients was entropy coding. The reason is that the entropy of each level was computed and found to be small enough to allow large compaction rates, and the use of entropy coding achieves compaction close to these rates. The results produced to this approach were encouraging, and they were extended and modified by many researchers. In addition the fact that they are based in the use of filters allow us to base the analysis of the method in the well developed framework, and predict its behavior and performance.

The relation between the pyramid approach is that it can be shown that each resolution level error coefficients contribute to the final level through a wavelet, whose dilation depends on the resolution of the level being expanded. One issue that appears then is whether the limit of these discrete wavelets in continuous time would be a continuous function. This property is called regularity. In general for every pyramid method that needs to employ this theory as its theoretical background, the regularity property must be shown. For the method discussed in this paper the regularity property is automatically satisfied because the method starts by choosing the wavelet to be used. This avoids the problem that the wavelet employed to be continuous.

3. THE USE OF THE HIGH-SPEED PYRAMID

The main advantages of that multiresolution representation are the computational simplicity and the fact that there is no need for extra space for the data above the first level. It can be proved [8] that the reduction and expansion processes are:

$$\alpha^{k+1}(\overrightarrow{n}) = \begin{cases} undefined & \text{if all components of } \overrightarrow{n} \text{ are even} \\ x^{k+1}(\overrightarrow{n}) - \frac{x^k(\overrightarrow{n}-\overrightarrow{m})+x^k(\overrightarrow{n}-\overrightarrow{m})}{2} & \text{if at least one of the components of } \overrightarrow{n} \text{ is odd} \end{cases} \tag{11}$$

$$x^{k+1}(\overrightarrow{n}) = \begin{cases} x^k(\frac{\overrightarrow{n}}{2}) & \text{if all components of } \overrightarrow{n} \text{ are even} \\ \sigma^{k+1}(\overrightarrow{n}) + \frac{x^k(\overrightarrow{n}-\overrightarrow{m})+x^k(\overrightarrow{n}+\overrightarrow{m})}{2} & \text{if at least one of the components of } \overrightarrow{n} \text{ is odd.} \end{cases} \tag{12}$$

Where $\overrightarrow{m} = f(\overrightarrow{n})$ in the manner discussed in [8]. It is easy to see that the computational requirements for coding are the same as for decoding, except that one of the pixel additions becomes a subtraction. The process requires data of dimensions $M_1 \times M_2 \times M_3$ where $M_i = (B_i - 1)2^b + 1$. At the lower resolution of the pyramid the dimensions of the data are $B_1 \times B_2 \times B_3$. The above formula take advantage of the correlation on all 3 dimensions of the data, is very fast to implement it in both hardware and software and by using some coding scheme based on tree structures is profitable form very low bit rates of transmission. By the appropriate selection of a threshold we can have the desired bit rate and the equivalent PSNR. Of course by having a lower bit rate the PSNR is getting lower.

When coding the values the process can take advantage of the structure of the algorithm to produce tree structures that aid in encoding, by choosing the appropriate one of the pixels $\overrightarrow{n} \pm \overrightarrow{m}$ as parent of pixel $\overrightarrow{n}$. These tree structures are found to have the property that the majority of the zero coefficients are grouped under common subtrees. This allows a progressive transmission scheme where for each coefficient we also transmit a bit of position information of whether it is the root of a zero subtree. Correspondingly, coefficients that belong to zero subtrees rooted higher levels are omitted from transmission. This means that we need not transmit zero information for almost all zero coefficients, that cover all but few of the pixels. This allows a significant increase in coding efficiency when compared to independent level by level encoding of the coefficients.

4. EXPERIMENTAL RESULTS

To illustrate the compaction properties of the algorithm, it was applied to a $257 \times 257 \times 33$ MR data. After coding, a level dependent quantization factor was used to uniformly quantize the coefficients so that the total error introduced in the wavelet representation, when using the quantized coefficients, produces a good peak SNR (PSNR). This is done by choosing a smaller quantization threshold for the positions where the support of the corresponding wavelet is larger. This approach assigns smaller thresholds to the lower pyramid levels, which is a desirable property. The quantized coefficients that are not inferred to be zero from the position bit information are then transmitted using PCM. The results are listed below:

SNR	PSNR	d/p PCM	cmp. fcr PCM	b/p ART	cmp. fcr ART
19.930002	29.303882	0.430352	18.589434	0.265558	30.125195
21.277158	32.694994	0.842121	9.499822	0.548143	14.594723
21.783139	33.630145	0.850170	9.409883	0.557579	14.347746
22.603016	34.020853	0.885782	9.031568	0.586856	13.631959
26.547373	39.924536	1.329506	6.017272	0.877409	9.117752
27.659691	39.077528	1.406331	5.688561	0.903741	8.852096
28.394373	39.812209	1.441138	5.551168	0.947087	8.446952
37.184176	50.561339	2.221246	3.601582	1.526050	5.242293

It can be easily seen that the last method is superior than the Laplacian pyramid. As an alternative the 2D algorithm can be applied to every slice separately, instead of applying the 3D algorithm (without exploiting the correlation in the third dimension). In this case we also include the bit rates achieved when the encoding of the quantized coefficients is done using arithmetic encoding instead of PCM. The results are:

SNR	PSNR	d/p PCM	cmp. fcr PCM	b/p ART	cmp. fcr ART
17.265432	31.679636	1.410650	5.671145	0.472861	16.918289
17.598590	32.01279	1.416403	5.648109	0.482551	16.578564
21.367675	35.804284	1.850989	4.322014	0.73400	10.899175
27.850795	41.787024	2.255356	3.547112	1.073385	7.453058
33.979381	47.915610	2.597102	3.080357	1.402353	5.704699
37.006432	50.942662	2.808339	2.848659	1.615891	4.950828

It can be seen that significant improvements can be achieved if we switch to entropy coding of the coefficients.

In addition the method can be extended to 4D signals, as discussed in [8], making possible the encoding of moving MRI images. Because of the amount of data involved compression would be even more necessary there.

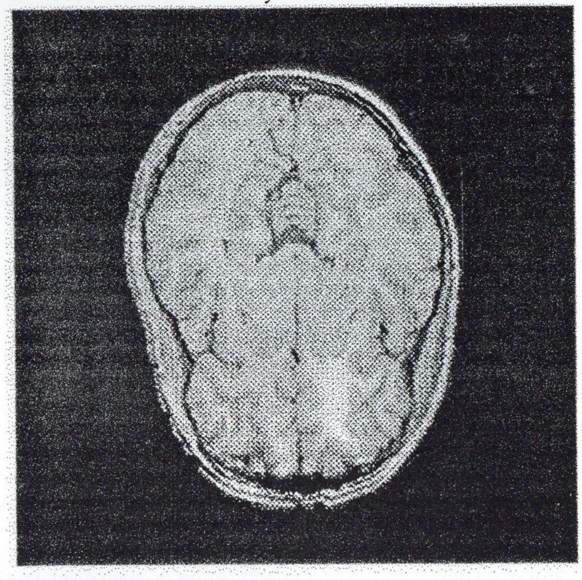


Fig. 3. One of the slices used in the simulations

REFERENCES

[1] M. A. Viergever, P. Roos: "Hierarchical interpolation", *IEEE Engineering in Medicine and Biology*, pp. 48-55, March, 1993.

[2] A. Papoulis: *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New Jersey, 3nd edition, 1991.

[3] P. Ross, M. A. Viergever, M. C. A. Van Dijke, J. H. Peters: "Reversible intraframe compression of medical images", *IEEE Transactions on Medical Imaging*, (7): 328-336, 1988.

[4] P. Ross, M. A. Viergever: "Reversible 3D decorrelation of medical images", *IEEE Transactions on Medical Imaging*, 12(3): 413-420, September, 1993.

[5] Jones PW Rabani M.: *Digital Image Compression Techniques*, SPIE Press, Bellingham WA, 1991.

[6] R. E. , L. R. Rabiner: *Multirate Digital Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1983.

[7] T. Chen, P. P. Vaidyanathan: "Recent developments in multidimensional multirate systems", *IEEE Trans. Circuits and Systems for Video Technology*, 3(2): 116-137, April, 1993.

[8] H. Sahinoglou: *Multiresolutional Image Encoding Methods Leading to Efficient Algorithms and Implementations.* PhD thesis, Pennsylvania State University, 1992.

[9] N. K. Bose: *Applied Multidimensional Systems Theory.* Van Nostrand Reinhold, 1982.

[10] A. V. Oppenheim, R. W. Schafer: *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1975.

[11] A. V. Oppenheim, R. W. Schafer: *Discrete-time Signal Processing*, Printece Hall, Englewood Cliffs, New Jersey, 1989.

[12] C. Harley: *Wavelets and Filter Banks*, PhD thesis, Columbia University, April, 1993.

[13] J. W. Woods, S. D. O'Neil: "Subband Coding of Images", *IEEE Trans. Acoust., Speech and Signal Processing*, 34(5): 1278-1288, October, 1986.

[14] J. W. Woods, editor: *Subband Image Coding.* Kluwer Academic Publisher, New York, 1991.

[15] J. S. Lim: *Two-Dimensional Signal and Image Processing*, Prentice Hall, 1990.

[16] I. Daubechies: "Orthonormal Bases of Compactly Supported Wavelets", *Comm. Pure Appl. Math.*, 41: 909-996, November, 1988.

[17] S. G. Mallat: "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. Pattern Anal. and Mach. Intell.*, 1(7): 674-693, October, 1989.

[18] P. P. Vaidyanathan: *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[19] Shigeru Muraki: "Volume Data and Wavelet Transforms", *IEEE Computer Graphics & Applications*, pp. 50-56, July, 1993.

[20] H. Sahinoglou, S. D. Cabrera: "A high-speed pyramid image coding algorithm for a VLSI implementation", *IEEE Trans. Circuits and Systems for Video Technology*, 1(4): 369-374, December, 1991.

[21] P. J. Burt, E. H. Adelson: "The Laplacian Pyramid as a Compact Image Code", *IEEE Trans. Computers*, 31(4): 332-540, April, 1983.

[22] K. M. Uz, M. Vetterli, D. J. LeGall: "Interpolative Multiresolution Coding of Advanced Television with Compatible Subchannels", *IEEE Trans. Circuits and Systems for Video Technology*, 1(1): 86-99, March, 1991.

CODING FOR MONOSCOPIC AND STEREOSCOPIC 3D MEDICAL DATA VISUALIZATION*

D. TZOVARAS, N. GRAMMALIDIS, S. MALASSIOTIS and M. G. STRINTZIS

INFORMATION PROCESSING LABORATORY, ARISTOTLE UNIVERSITY OF THESSALONIKI
THESSALONIKI 54006, GREECE

1. INTRODUCTION

Data coding and compression is an essential processing task in medical image transmission and storage. The transmission of the enormous information contained in medical data through limited capacity channels is still an open problem. In this paper we investigate various approaches for efficient coding of medical data. Alternative techniques are proposed, that do not require the transmission of the whole 3D data set, and depend on the desired complexity of the decoder.

In medical applications such as surgery planning and tumor detection, surfaces are extracted from the 3D data set, and then rendered to provide an easy to understand presentation of the information contained in the data set.

In surface rendering our goal is the creation of realistic pictures of 3D objects. This is done by simulating the lighting of the scene, the surface properties (e.g. surface color), the camera-eye system, the position of the viewer, the object motion etc. Rendering can take place at either the encoder or the decoder sites. In the former case, a low cost decoder-display system, provides a simple user interface for low level operations on the visualized surface. The user-specified parameters are transmitted over the channel to the coder, that transforms, renders and compresses the required information. Then, the decoder decompresses and displays the data sequence transmitted.

As graphics hardware systems are becoming less expensive, online rendering can be performed on low cost graphics workstations. In this case, higher level operations may be performed on the decoder, and the coder is decoupled from the user interaction, which is performed through a local software interface.

In the current work, we examine techniques for the coding of the whole 3D surface information or a number of 2D views of the surface. When these views are presented consecutively, object based motion compensation may be used to compress the sequence. All proposed methods were tested using head MRI data forming 256 slices with a 1 mm distance between slices.

2. 3D SURFACE COMPRESSION

If the decoder is a graphics workstation, interaction with the user can be performed through a local software interface. The task of the coder is to extract the 3D surface from the source data set (MRI, PET, etc.) and then transmit this information in an efficient way. In the decoder the information is uncompressed and visualized according to the user specifications (viewpoint, lighting, animation etc.).

Since the original 3D data usually consist of several slices of 2D data, the 3D surface may be equivalently represented by stacking several 2D contours. We assume that the coder is able to extract such contours with methods such as recently discussed in [1], and cope with noise that produces erroneous edges. The contours are constrained to be closed, and we only handle the case of a single contour per slice. In this work we investigate, lossless compression of each contour using contour following and subsequent entropy coding as well as spline modeling of contours.

2.1. Contour Following

In this approach [2], we simply choose an initial point on the contour and then trace the contour in a clockwise manner labeling the direction as we shift from one contour element to the next. The resulting data stream is then entropy coded using arithmetic coding techniques [3].

A disadvantage of this technique is that the entropy of the directional information does not decrease with the complexity of the object shape, as was concluded by experimental results. It has the advantage of simplicity allowing for a very fast implementation.

2.2. B-Spline Representation

Higher compression rates may be achieved by approximating the discrete data (x_i, y_i) with a continuous parametric curve $(x(t), y(t))$ [4] [5]. The shape of the curve depends on a set of parameters p which is the one to be transmitted. The approximation error decreases as the number of those parameters increases.

B-Splines are piecewise polynomial functions that can provide local approximations of contours of shapes using a small number of parameters, the control points. They are widely used for representation and smoothing of coarsely digitized contours.

The B-Spline representation is written as :

$$x(t) = \sum_{i=0}^{n} B_i^k(t)\mathbf{p}_i,$$

where $\mathbf{p}_i = 0, 1, \ldots, n$ is a set of control points, and $B_i^k(t)$ are the *normalized B-splines of order* k defined recursively as.

$$B_i^1(t) = \begin{cases} 1, & t_i \leq t \leq t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$B_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} B_i^{k-1}(t) + \frac{t_{i+k-1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1}^{k-1}(t).$$

The parameters *ti*, the *knots*, are locations where the spline functions are tied together and are usually uniformly defined over the range of t.

The contour approximation problem requires estimation of the control points $\mathbf{p}_i$ that give rise to minimization of the distance between the B-spline and the original contour, $\hat{\mathbf{x}}_i, \hat{t}_i, = 0, \ldots, m$. Since $\hat{\mathbf{x}}_i$ are defined on a Cartesian grid, we do not have the corresponding parametric values $\hat{t}_i$. Since t represents distance across the curve, for the discrete contour, this distance between two successive points is approximately 1 for horizontal and vertical directions and $\sqrt{2}$ for diagonal directions.

The estimation problem can be written as:

$$\min_{\mathbf{p}_i} \sum_i (\mathbf{x}(\hat{t}_i) - \hat{\mathbf{x}}_i)^2. \qquad (1)$$

Since $\mathbf{x}(t)$ is a linear combination of the control points Eq. (1) is equivalent to

$$\min_P \|AP - C\|^2, \qquad (2)$$

* This material is based on work supported by the RACE 2045-DISTIMA project.

where $P = \{\mathbf{p}_i\}_{i=0}^{n}$, $C = \{\hat{\mathbf{x}}_i\}_{i=0}^{m}$ and A is a k-band matrix with elements of the form $B_i^k(\hat{t}_i)$. Using linear least squares theory, the solution of Eq. (2) is

$$P = (A^T A)^{-1} A^T C.$$

The computational complexity of the algorithm is due to the inversion of a $m \times m$ matrix. Faster implementations can be based on iterative solution of Eq. (2). The number of control points m is selected according to the length of the contour and the complexity of its shape. This allows for adaptive contour compression.

The B-spline curve is invariant under rotation, translation and scaling transformation of its control points, therefore such operations can be performed very efficiently on the decoder. A parametric surface can be easily derived from two successive contours using linear interpolation

$$S(u, v) = vC_1(u) + (1 - v)C_2(u).$$

Hardware rendering of B-spline surfaces is a common feature of many graphics workstations [6].

2.3. Experimental Results

Experiments were performed using head MRI data. Segmentation of the head boundary was performed manually by thresholding and subsequent morphological filtering for noise elimination and shape smoothing.

The extracted contours were coded with the previously described techniques. In the case of B-spline modeling every second slice is skipped. The midpoint contours are interpolated at the decoder. Table 1 presents the total bytes required to code the 3D surface. Fig. 1b shows the reconstruction of the "Head" image when using spline modeling of the contour information. The interframe difference is 26 dB. As can be seen the spline surface filters noise on flat face regions but it fails to accurately represent highly detailed regions (e.g the ears). Such problems can be reduced by decreasing the compression ratio.

Table 1. Total bytes transmitted

Contour following	B-spline approximation
44 661 bytes	24 831 bytes (error=0.6)

3. DEPTH MAP CODING

In several important medical applications, the method of 3D surface description using contour following is not easily applicable. In such cases a preferable description of the depth maps may be afforded directly by the ray-tracing used.

A commonly met instance where the transmission of the 3D surface description is not necessary, is when a small number of snapshots of the depth maps, corresponding to certain angles of viewing the object, is sufficient for the depiction of the medical information that must be stored or transmitted. This series of snapshots may then be coded as though they were consecutive in time and their aggregate may be treated as a "moving" sequence. In fact, since motion parallax is a powerful aide to image understanding, their actual positioning in a moving sequence may be precisely what the medical observer requires. The rotation back and forth for example of a 3D MRI image gives a very good understanding of medical detail.

Efficient techniques must then be found for the coding and compression of the depth maps corresponding to multiple views of a particular 3D data set. Let (X_t, Y_t, Z_t) represent the depth map at time t. If $\vec{V}_t$ is the 3D vector denoting the position of a point in the 3D space at time instant t, the position of the same point at time $t + 1$ will be:

$$\overrightarrow{V_{t+1}} = \mathbf{R}\vec{V}_t + \mathbf{T}, \tag{3}$$

where **R** is the rotation matrix and **T** is the translation matrix. The general form of Eq. (3) is:

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{bmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}. \tag{4}$$

Thus, compensation is performed using the following estimation of the depth at $t + 1$:

$$\hat{Z}_{t+1}[X_{t+1}, Y_{t+1}] = R_{31}X_t + R_{32}Y_t + R_{33}Z_t + T_z, \tag{5}$$

where X_{t+1}, Y_{t+1} are determined from eq. (4). In the specific case where the object rotates around the Y axis with angle of rotation $\emptyset$ then Eq. (4) will be:

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{bmatrix} = \begin{pmatrix} \cos\emptyset & 0 & -\sin\emptyset \\ 0 & 1 & 0 \\ \sin\emptyset & 0 & \cos\emptyset \end{pmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix}. \tag{6}$$

As seen, for the computation of X_{t+1}, Y_{t+1}, the depth map at time $t(Z_t)$, has to be known at both the encoder and decoder sites. Some problems arise in the motion compensation procedure due to the floating point form of the 2D motion vectors. These motion vector values point to positions, outside the sampling grid of the subsequent frame. Therefore an interpolation procedure has to be adopted in order to assign values at integer pixel locations.

An efficient approach based on the upsampling of the depth map Z_t at time instant t was implemented. Linear interpolation was used for the calculation of depth values (intensities) between consequent points in the sampling grid at time instant t. The corresponding position $(X_{t+1}, Y_{t+1}, Z_{t+1})$ of a point (X_t, Y_t, Z_t) on the new sampling grid, is then computed using Eq. (4). Using motion compensation, an extended sampling grid estimate of the depth map at time $t+1$ is produced. Then the depth map is propagated to the desirable resolution using lowpass filtering followed by appropriate downsampling. This technique led to satisfactory results that improve as the upsampling rate increases.

The important feature of the motion compensation procedure is that only the 3D motion parameters of the objects and the object boundaries have to be transmitted. The proposed scheme can also work in lossless mode, with the lossless transmission of the information contained in the prediction error images.

3.1. Experimental Results

A sequence of depth maps corresponding to 2D views of the 3D data set were generated from the original MRI slices. In our test sequences each frame of the depth map sequence corresponds to an object rotation by 2 degrees around the Y axis. In order to test the efficiency of the motion compensation method we coded frames 1,2,3,4 and 5 using motion compensation based on frame 0 (which we assume that is intra coded). These frames are coded using the 2D contour of each object in the scene — a single object exists in our test sequence "Head" — as well as three rotational and three translational motion model parameters.

The performance in bit rate of transmission for each depth map is presented in Tab. 2. Results are presented in Tab. 3 for the sequence "Head". As a comparison the same table shows the motion compensated results that are obtained if an exhaustive search block-matching method is used for the same purpose. Furthermore as a lower bound for the performance of the motion compensated methods that may be implemented, the results provided by simple frame repetition are also presented.

The results show that the proposed motion compensation method generally outperforms block matching. In fact the block-matching method generally fails to detect correct correspondences between depth maps. The reason is that the latter method yields correspondences based on the similarity of the depths of a moving point between the two time instances. Such correspondences are valid only in the case of a translational movement parallel to the image plane (in this case the depth of the moving point remains almost constant). Results shown in Tab. 3, demonstrate the degradation of efficiency caused by increasing the angle of rotation.

Lossless coding of the depth map information is achieved if we adopt a lossless scheme for the coding of the prediction error images. An alternative would be to transmit the contour information, whenever this is feasible (see Section 3).

Table 2. Bit rate required for lossy and lossless coding of the Depth Map information ("Head").
Lossy Case: Bit rate for the transmission of the boundary information.
Lossless case: Bit rate for the transmission of the boundary information + Bit rate for lossless transmission of the prediction error image.

Frame Number	Lossless (Bits/pixel)	Lossy (Bits/pixel)
1	0.3117+0.0406	0.0406
2	0.5570+0.0408	0.0408
3	0.6240+0.0395	0.0395
4	0.8017+0.0397	0.0397
5	0.9075+0.0402	0.0402

Table 3. Depth map Motion Compensation compared with Exhaustive Block Matching ("Head").

Frame Number	3D Motion Compensation	Block-matching 2D Motion Comp.	Frame difference
1	45.59	40.42	31.01
2	43.08	37.85	28.11
3	40.19	36.32	28.11
4	38.26	34.91	26.02
5	35.83	33.99	24.67

4. VISUALIZATION AT THE DECODER SITE

If a full-capacity workstation is available at the decoder site, it will be sufficient to transmit the 3D information (a) using a lossless or lossy 3D data form, (b) in 3D surface form (Sec. 2), and (c) in the form of 2D snapshots of the depth maps (Sec. 3). The decoder will then visualize the data. Full interactivity is possible with this approach. Moreover, since the raw data coding to be done will probably be identical to that needed for data storage, transcoding will be minimal.

If the decoder hardware is able to render a given depth map then motion compensation may be performed in the depth map sequence (see Sec. 4). The decoder produces a reconstruction of the original depth map and then uses its hardware in order to produce a visualized image that is an approximation of the optimal visualized image (corresponding to the correct depth map). It must be noted that the visualization process is extremely sensitive to errors in depth map transmission.

Experiments in coding visualized images produced by the sequences "Head" and "Brain" have shown that this method produces very good results for small angles of rotation. This is due to the fact that the error in depth map approximation increases with the angle of rotation. As noted above the rendering procedure is very sensitive to depth map errors, and the error in reproducing the visualized images becomes significant. Therefore it is strongly recommended that depth maps transmission for visualization at the receiver site be as near to lossless as possible.

Results are presented in Table 6, for the case of lossy (Motion Compensated) transmission of the depth map information. Lossless transmission of the depth maps would yield perfect reconstruction at the decoder site. The corresponding bit rates required for lossy or lossless reconstruction are presented in Table 2. As a conclusion we may note that the performance of such a method is satisfactory for small angles of rotation, but the fact that a very sophisticated hardware is prerequired is a significant disadvantage.

Table 6. Results for visualization at the decoder, using motion compensated depth maps.

Frame Number	"Head" sequence	"Brain" sequence
1	33.97	30.01
2	31.58	28.95
3	29.51	28.19
4	27.13	26.79
6	24.80	24.54

5. VISUALIZATION AT THE ENCODER SITE

It may be very important in practice to ensure that simple, output-only monitors are capable of interactive display of 2D and 3D information. Such monitors may be simple personal computers or memoryless workstations or TV monitors. The coding scheme to be implemented must ensure that physicians will be able to view 3D data much as X-ray film is viewed today, using simple monitors and a mouse (or a joystick) or other very simple signalling device for the remote interactive manipulation (rotation, translation, zooming, cut) of the 2D and 3D data. The sequence available at the encoder site will contain all images required by the corresponding medical protocol.

The image formation process of the visualized images is a perspective projection of a 3D object onto the image plane. The center of projection is located at the origin of the 3D world coordinate system, at a distance f from the image plane. If $(X_t, Y_t, Z_t)^T$ is the position vector of a point in the 3D space and $(x_t, y_t, f)^T$ is the position vector of its projection at the image plane then the following relation exists between the image and the world coordinates at two consequent time instants:

$$x_{t+1} = f\frac{X_{t+1}}{Z_{t+1}} \; x_t = f\frac{X_t}{Z_t} \; y_{t+1} = f\frac{Y_{t+1}}{Z_{t+1}} \; y_t = f\frac{Y_t}{Z_t}. \qquad (7)$$

We now assume that the moving point belongs to a rigid moving object and that its instantaneous displacement (between two consecutive views) can be modeled using three translational parameters (T_x, T_y, T_z) and three rotational parameters that describe rotations around the x, y, z axes. These rotational parameters generate on orthonormal matrix R having the form of Eq. (4).

However it is very difficult (practically impossible) to use the model that yields the luminance of a 3D point at each time instant (and thus relates the luminances of two corresponding 3D points at two consecutive time instants). This model uses knowledge about the lighting conditions of the scene, the surface reflectability of the objects in the scene as well as knowledge about the gradient of the surface at each point (that is calculated from the available depth map for each object).

Therefore, since we cannot use the underlying luminance model, we shall make a simplification common in most motion estimation methods (including block matching): we shall assume that the projections of a 3D point at two consecutive time instants yield equal corresponding luminances. In other words, we assume that the pixel intensity is the same in any two consequent frames:

$$I_2(x_{t+1}, y_{t+1}) = I_1(x_t, y_t). \qquad (8)$$

In fact, Eq. (8) is incorrect, since the intensity of each pixel will change according to its position and the parameters (lighting model, outer surface gradient etc.) of the rendering procedure. However, the precise form of the intensity transformation is very complicated. The above assumption Eq. (8) simplifies very much the compensation procedure.

We propose an object-based 3D motion compensation scheme which makes motion compensation possible for every point of an object in the scene if the following information is available to the decoder:

- A reconstruction of the previous frame in the visualized image sequence.
- The 3D motion information (6 parameters).
- The depth map corresponding to the current visualized image (or a reconstruction of it — e.g. produced by the proposed depth map motion compensation method).

6. STEREOSCOPIC VIEWING AND TRANSMISSION

Depth perception, hence stereo viewing, is very useful in most common medical applications. If visualization is performed at the decoder site, generation of a second (stereo) view will necessitate the completion of two separate rendering procedures. If visualization is done at the encoder site, a second (right) channel image may be coded precisely as the first (left) sequence. Note however, that the methods found to be efficient in Section 6 of this report, for the coding of visualizations done at the encoder site depend upon the transmission of accurate depth maps. Thus, transmission of a right-channel sequence may be replaced by simple disparity compensation of the left-channel image using the depth maps to estimate disparity. In such a case each right view image is formed based on the corresponding left view image, under the assumption that corresponding pixels have the same intensity. We propose to use this method to generate the right channel images, whether visualization is done at the encoder or the decoder.

6.1. *Experimental Results*

The process of stereo image generation depends on the display configuration and human factors that determine the viewing accomodation. In our simulations a set up of virtual cameras with converging optical axes was adopted. The converging angle was assumed to be 4o, and the convergence point was on the center of the object viewed.

We compare the performance of the method proposed in the previous section with the 1st generation DISTIMA coder [7]. The proposed method performs equally well with the block based disparity compensation approach, as demonstrated in Table 13.

Table 13. Comparison of the performance of the 1st generation DISTIMA coder versus the proposed direct disparity compensation method for the coding of the stereoscopic sequence "Head".

Frame Number	Object Based Coding	Distima Coder	Frame difference
1	30.10	30.06	22.09
2	29.53	29.95	21.59
3	29.91	30.11	21.89
4	29.64	30.10	22.31
5	29.95	29.82	22.16
6	30.09	30.30	22.20

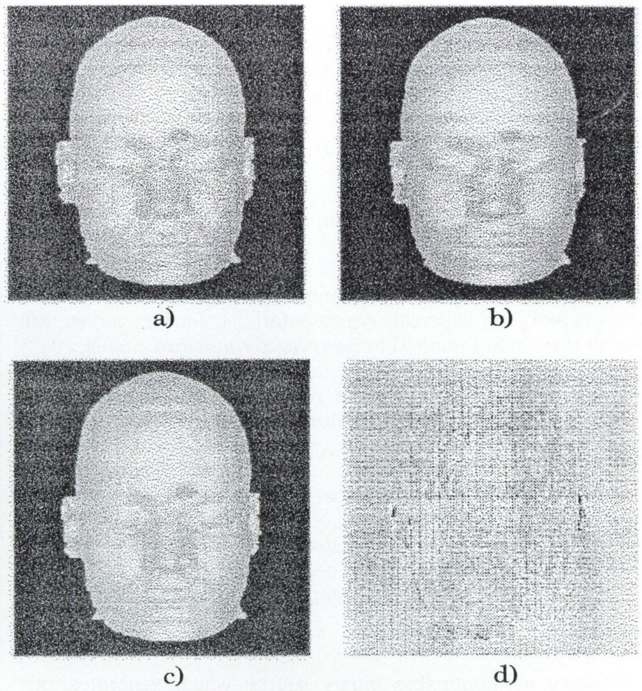


Fig. 1. (a) Original "Head" visualized image, (b) Visualized image after B-spline modeling of the contour information, (c) Reconstructed image after object based coding, (d) Prediction error image

REFERENCES

[1] R. L. Eisner, M. E. Brummer, R. M. Mersereau, R. R. J. Lewine: "Automatic Detection of Brain Contours in MRI Data Sets ", *IEEE Trans. Medical Imaging*, 12(2):153-166, 1993.

[2] J. A. Saghri, H. Freeman: "Computer Processing of Line Drawing Images". *IEEE Trans. PAMI*, pp. 533-539, 1981.

[3] R. M. Witten, I. H. Neal, J. G. Cleary: "Arithmetic Coding for Data Compression", *Communications of the ACM*, 30(6):520-540, June 1987.

[4] W. J. Gordon and R. F. Riesenfeld: "B-spline Curves arzd Surfaces", pp. 95-126. *Academic Press*, New York, 1974.

[5] D. Paglieroni, A. K. Jain: "A Control Point Theory for Boundary Representation and Matching". *In Proc. ICAASP*, Vol. 4. pp. 1851-1854, Tampa, 1985.

[6] S. K. Feiner, J. D. Foley, A. van Dam, J. F. Hughes: "Computer Graphics: Principle and Practice", *Second Edition*, Addison-Wesley, 1990.

[7] R. Lagendijk, R. Franich, R. Ter Horst: "Codec Specification for Hardware Implementation", Technical report, RACE II DISTIMA # R2045: Digital Stereoscopic Imaging and Applications (DISTIMA), Dec. 1992.

PROGRESSIVE IMAGE TRANSMISSION WITH RUN-LENGTH CODING

S. SIMON and J. DE VRIENDT

LABORATORY FOR COMMUNICATIONS AND IMAGE PROCESSING,
UNIVERSITY OF GHENT
SINT-PIETERSNIEUWSTRAAT 41, B-9000 GHENT

1. THEORY

In general, there are two ways of representing pixels in images. With the first and most straightforward method, the pixel values are represented explicitly. Each pixel is represented by a Huffman codeword. Codewords are of variable length, according to the probability of occurrence of the pixel.

A disadvantage of this method is that the mean code length per pixel cannot be less than one bit. A trick to overcome this, is to take blocks of symbols together. Another solution is using *run-length* codes, where a symbol is not coded directly, but the number of times the symbol occurs in one consecutive "run" is assigned a codeword to. Further methods use two-dimensional codes where one codeword is assigned to pairs of run-lengths and pixel values.

A common disadvantage to most coding systems is that there is no provision for progressive transmission of images. The source is reconstructed at the accuracy it was coded at, from the beginning to the end. If half of the amount of codewords is transmitted, only half of the image can be reconstructed at the decoder, and it is impossible to have an idea of the rest of the image.

We present another approach to run-length coding which solves the aforementioned problems. Run-length coding is applied not only for just one symbol, but for all of them. The proposed coder proceeds as follows: we have an alphabet of m symbols. Firstly, symbol S_0 is tackled. The positions of all occurrences of symbol

S_0 are coded by counting the run-lengths of non-symbol S_0's between the occurrences of symbol S_0. This coding process goes on for the next $m-2$ symbols in the same way. The occurrences of the symbols which already have been coded, are not counted in the run-lengths (skip coding).

Note that the last symbol S_{m-1} doesn't have to be coded, as it occurs in the gaps which have been left open.

We will prove theoretically that in the case of the method just described, the zero-th order entropy of the run-lengths is exactly equal to the entropy of the original pixels. S_0, in final codelength, there is no difference between direct coding of the pixels (with implicit positioning) and the dual approach of just coding the pixel positions (with implicit value representation). Further on, we will refer to the second method as *generalized run-length coding.*

In the following paragraph, the outline of the proof is sketched. First, the proof is given for the rather trivial case of a binary source. The result is then generalized to alphabet-m sources.

1.1. Proofs

We have a memoryless source with an alphabet with m symbols $\{S_i | i = 0, \ldots, m-1\}$. Each symbol S_i has a probability of occuring p_i. The source generates N symbols. To be proven is that the total code length A needed for representing the N symbols is the same for direct coding of the pixel values as for the generalized run-length coding.

1.1.1. Binary source

We have a memoryless binary source which generates two symbols: 0 and 1. The minimal total code length for N symbols in the direct case is equal to N times the entropy of the symbols.

$$A_1 = N \cdot H(p_0, p_2, \ldots p_{m-1}) = -N \sum_{i=1}^{1} p_i \log p_i .$$

The probability of having an uninterrupted run of r ones is $P(r) = p_i^r p_0$. The entropy of the run-lengths of ones, or, in other words, of the relative positions of the symbol $S_0 = 0$ is then

$$\begin{aligned} H_N &= -\sum_{r=0}^{\infty} P(r) \log P(r) = \\ &= -\sum_{r=0}^{\infty} p_1^r p_0 \log(p_1^r p_0) \\ &= -\sum_{r=0}^{\infty} p_1^r p_0 r \log p_1 - \sum_{r=0}^{\infty} p_1^r p_0 \log p_0 . \end{aligned} \tag{1}$$

Now, we make use of some well-known series sums:

$$\sum_{r=0}^{\infty} a^r = \frac{1}{1-a}$$

$$\sum_{r=0}^{\infty} r a^r = \frac{1}{1-a^2}$$

so that the Eq. (1) can be rewritten as

$$H_N = -\frac{p_1}{(1-p_1)^2} p_0 \log p_1 - \frac{1}{1-p_1} p_0 \log p_0 .$$

With $p_0 = 1 - p_1$ we get $H_N = -\frac{1}{p_0}(p_0 \log p_0 + p_1 \log p_1) = = \frac{1}{p_0}\mathcal{H}(p_0)$.

This entropy represents the minimum number of bits needed to represent the relative position of a zero with respect to the former zero (the run-lengths of non-S_0's before each S_0). In a stream of N symbols, and a probability of p_0 for a S_0, there are $p_0 N$ zeroes on average in the stream, so the code with length given in the formula above has to be applied $p_0 N$ times.

The total code length then becomes

$$A_2 = p_0 N H_N = -N(p_0 \log p_0 + p_1 \log p_1),$$

which is equal to A_1.

The two representations are exactly equivalent in terms of entropy. Note that for the run-length code, only the positions of the symbols S_0 have been coded. The symbols S_1 fit in where no zeroes occur. This result is of course already well-known. However, the calculations will serve as the basis for the next proof.

1.1.2. Source with alphabet of m symbols

These results can now be generalized to memoryless sources with an alphabet of m symbols.

The probability of occurring of symbol S_i is denoted as p_i, the probability that another symbol than S_i occurs is called $\overline{p_i}$. The total number of symbols to code is N. Symbols are coded starting with symbol S_0.

Direct coding of the pixel values leads to a minimum code length of

$$A_1(m) = -N \sum_{i=0}^{m-1} p_i \log p_i . \tag{2}$$

It is remarked that the probability of the last symbol S_{m-1} can be written as

$$p_{m-1} = 1 - \sum_{k=0}^{m-2} p_k . \tag{3}$$

For the generalized run-length coding, all symbols which already have been coded at a certain moment, are skipped when the next symbol is coded. p_i' is the probability of occurring of a symbol S_i when all symbols $\{S_i, j = 0..i\}$ have been coded already, and $\overline{p_i}'$ is its complement. To calculate p_i', the original probability distribution has to be renormalized. p_i and p_i' are related as follows :

$$p_i' = \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}$$

$$\overline{p_i}' = 1 - \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}$$

In analogy with the binary case, the probability of having a relative position of r is $P(r) = \overline{p_i}'^r p_i'$. The entropy of the run-lengths (or relative positions) of symbol S_i now becomes :

$$\begin{aligned} H_{N_i'} &= \sum_{r=0}^{\infty} \overline{p_i}'^r p_i' \log(\overline{p_i}'^r p_i') = \\ &= \frac{1}{p_i'}(\overline{p_i}' \log \overline{p_i}' + p_i' \log p_i') = \\ &= \frac{1}{p_i'}\mathcal{H}(p_i'). \end{aligned}$$

The number of symbols left over after coding symbols $\{S_k | k = = 0..i-1\}$ is

$$N_i' = N \cdot (1 - \sum_{k=0}^{i-1} p_k), \tag{4}$$

of which, again in analogy with the binary case, an expected number of $p_i' N_i'$ will be equal to S_i. We now will prove that

$$A_2(m) = \sum_{i=0}^{m-2} p_i' N_i' \cdot H_{N_i'} = A_1(m).$$

We remark that the A_2-sum runs only over $m-2$ symbols. It is not needed to code the locations of the last symbol explicitly, because it can be filled in in the places not taken by the other symbols.

When all symbols but the last one have been coded, the total code length becomes:

$$A_2(m) = \sum_{i=0}^{m-2} p_i' N_i' \cdot H_{N_i'} =$$

$$= - \sum_{i=0}^{m-2} N(1 - \sum_{k=0}^{i-1} p_k)$$

$$\left[(1 - \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}) \log \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}) + \right.$$

$$\left. + \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}) \log \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}) \right] =$$

$$= -N \sum_{i=0}^{m-2} \left[(1 - \sum_{k=0}^{i-1} p_k - p_i) \log (1 - \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}) + \right.$$

$$\left. + p_i \log (\frac{p_i}{1 - \sum_{k=0}^{i-1} p_k}) \right] =$$

$$= -N \sum_{i=0}^{m-2} \left[(1 - \sum_{k=0}^{i-1} p_k - p_i) \right.$$

$$\left(\log (1 - \sum_{k=0}^{i-1} p_k - p_i) - \log (1 - \sum_{k=0}^{i-1} - p_k) \right)$$

$$\left. + p_i \log p_i - p_i \log (1 - \sum_{k=0}^{i-1} p_k) \right] =$$

$$= -N \sum_{i=0}^{m-2} \left[(1 - \sum_{k=0}^{i} p_k) \log (1 - \sum_{k=0}^{i} p_k) - \right.$$

$$- (1 - \sum_{k=0}^{i-1} p_k) \log (1 - \sum_{k=0}^{i-1} p_k) + \tag{5}$$

$$\left. + p_i \log (1 - \sum_{k=0}^{i-1} p_k) + p_i \log p_i - p_i \log (1 - \sum_{k=0}^{i-1} p_k) \right] =$$

$$= -N \sum_{i=0}^{m-2} p_i \log p_i + (1 - \sum_{i=0}^{m-2} p_k) \log (1 - \sum_{i=0}^{m-2'} p_k). \tag{6}$$

With the formula $1 - \sum_{k=0}^{m-2} p_k = p_{m-1}$ we get finally :

$$A_2(m) = -N \sum_{i=0}^{m-1} p_i \log p_i$$

which is the same as the entropy $A_i(m)$ of the source. In other words, it is proved that the generalized run-length representation is fully equivalent in terms of entropy to the direct pixel representation.

More elegantly, the proof can be reformulated as follows. We make use of the *grouping axioma* from the information theory. Let G and $\overline{G}$ be groups of symbols, with p_G and $p_{\overline{G}}$ their associated probabilities, and $G \cap \overline{G} = \emptyset$, then: $H(G \cup \overline{G}) = \mathcal{H}(p_G) +$ $+ p_G H(G) + p_{\overline{G}} H(\overline{G})$, where $H(G) = H(p_0, p_1, p_2, \ldots) =$ $= - \sum_{i=0}^{m-1} p_i \log p_i$ the entropy of the symbols within a group, and $\sum_{i=0}^{m-1} p_i = 0$ within each group. Here, we regroup by setting apart a single symbol from the rest of the group. The entropy of a single symbol is zero.

$$H(p_0, p_2, \ldots p_{m-1}) = \mathcal{H}(p_0) + p_0 \cdot 0 +$$
$$+ (1 - p_0) H \left(\frac{p_1}{1 - p_0}, \frac{p_2}{1 - p_0}, \frac{p_{m-1}}{1 - p_0} \right)$$
$$= \mathcal{H}(p_0) + (1 - p_0)$$
$$\left[\mathcal{H} \left(\frac{p_1}{1 - p_0} \right) \left(1 - \frac{p_1}{1 - p_0} \right) H \left(\frac{p_1}{1 - p_0 - p_1}, .. \right) \right] =$$
$$= \mathcal{H}(p_0) + (1 - p_0) \mathcal{H} \left(\frac{p_1}{1 - p_0} \right) +$$
$$+ (1 - p_0 - p_1) \mathcal{H} \left(\frac{p_2}{1 - p_0 - p_1} \right) + \ldots$$
$$+ (1 - p_0 - \ldots - p_j) \mathcal{H} \left(\frac{p_{j+1}}{1 - p_0 - \ldots - p_j} \right) + \ldots$$
$$+ (1 - p_0 - \ldots - p_{m-3}) \mathcal{H} \left(\frac{p_{m-2}}{1 - \sum_{i=0}^{m-3} p_i} \right) =$$
$$= \sum_{i=0}^{m-2} \left(1 - \sum_{k=0}^{i-1} p_k \right) \mathcal{H} \left(\frac{p_i}{1 - \sum_{k=0}^{i-1} p_k} \right). \tag{7}$$

So, finally we get:

$$H(p_0, p_2, \ldots p_{m-1} = \sum_{i=0}^{m-2} \frac{p_i}{p_i'} \mathcal{H}(p_i') \tag{8}$$

Proof of $A_2(m) = A_1(m)$ (taking into account Eq. (2) and Eq. (4)):

$$A_1(m) = N H(p_0, p_1, p_2 \ldots p_{m-1}) \tag{9}$$

$$A_2(m) = \sum_{i=0}^{m-2} p_i' N_i' \cdot H_{N_i'}$$
$$= \sum_{i=0}^{m-2} \frac{p_i}{1 - \sum_{k=0}^{i-1} p_k} N \left(1 - \sum_{k=0}^{i-1} p_k \right) \frac{1}{p_i'} \mathcal{H}(p_i')$$
$$= \sum_{i=0}^{m-2} \frac{p_i}{p_i'} \mathcal{H}(p_i'). \tag{10}$$

Substituting Eq. (8) in Eq. (9), and comparing with Eq. (10), we conclude that $A_1(m) = A_2(m)$, q.e.d. The general version of Eq. (8) can be written as:

$$H(p_0, p_2, \ldots p_{m-1}) = \sum_{i=0}^{n-1} \frac{p_i}{p_i'} \mathcal{H}(p_i') +$$
$$+ \frac{p_n}{p_n'} H \left(p_n', \frac{p_{n+1}}{1 - \sum_{k=0}^{k-1} p_k}, \ldots, \frac{p_{m-1}}{1 - \sum_{k=0}^{k-1} p_k} \right). \tag{11}$$

A very important conclusion from Eq. (11) and Eq. (6) is that the codelength generated up to any symbol S is proportional to the entropy of just those symbols $\{S_i | i = 0..n-1\}$ which have been coded. This entropy does not take into account the individual probabilities of the symbols $\{S_j | j \geq n\}$ to come, but only their summed probabilities.

2. PRACTICAL IMPLEMENTATIONS AND APPLICATIONS

From these theoretical results, we have derived a family of coding systems to be used for digital images. The symbols are image pixels or transform coefficients. The methods share a common feature : symbols are not coded by their value, but by their relative positions. Every method allows for easy simple SNR-scalability. The lower-SNR code stream makes part of the higher-SNR code stream. Extra picture quality can be added without any overhead. This makes this coding approach very suitable for progressive transmission of images, especially in a wavelet or sub-band environment. A well-presented overview of other progressive transmission schemes can be found in [1].

The relative positions or run-lengths are coded with the simple, however efficient limited word-length run-length coding scheme of Tanaka and Garcia, as described in [2]: the ATRL-code or adaptive run-length code. In the Tanaka/Garcia run-length coding scheme, only codewords of length 1 (for run-lengths M) or of length $m + 1$, where $m = \log_2 M$, (run-lengths 0 to $M - 1$) can appear. Such a code is called a mode-m truncated run-length (TRL) code. m is optimized using a statistic of the run-lengths. If the probability of a pixel is p_i' given, and with $p = \overline{p_i}' = 1 - p_i'$, then

$$m = \lceil \log_2 \left[\frac{\mathrm{In}\gamma}{\mathrm{In}p} \right] \rceil,$$

where $\gamma = \frac{\sqrt{5}-1}{2}$.

The expected codelength per symbol

$$l(p,m) = (1-p) \left[\frac{p^{2^m}}{1-p^{2^m}} + (m+1) \right].$$

The efficiency of this code is never less than 95.94, for $p = \gamma$. Higher values of p increase the efficiency. The optimal m can be made adaptive by calculating experimental statistics over a block of L source symbols. For each coded block, a prefix with the value of m is then included in the codestream. In our implementation, the optimal m for each pixel value is calculated from the expected distribution of the pixel values. For sub-band or transform coefficients, this distribution is approximately Laplacian ($L(x) = \frac{1}{\sqrt{2\sigma}} \exp\left(-\frac{\sqrt{2}|x|}{\sigma}\right)$ [3], which is symmetrical and highly peaked around zero. In the explanation which follows, a uniformly quantized Laplacian distribution with an uneven number of levels is assumed for the pixel values. For each sub-band in a sub-band coding scheme, or each order of the coefficients in a transform coding scheme, the variance of its distribution has to be transmitted to be able to determine the optimal m for all symbols at the decoder.

2.1. GRLC

We call our first own method generalized run-length coding. In fact, this technique, as it is described in the theoretical part before, is equivalent to variable thresholding: first, the most important (the greatest) pixels are transmitted. Gradually, the smaller pixels get their turn. The transmission can be stopped at any moment. The coding method guarantees that all pixels transmitted up to that moment, are represented in the most efficient way possible (neglecting possible inter-pixel correlations). The pixels in the high bands are ordered in such a way that the less probable, but important pixels with large absolute values are coded first. Alternatingly, positive and negative values are coded. The first pixels which are coded are relatively sparse, but when the absolute values decrease, positions get more dense, and relative run-lengths become smaller.

A disadvantage of this method is the fact that, to code and decode all pixels, a number of passes over the image has to be made which is equal to the number of possible pixel values, e.g. 256 for byte images. A parallel approach to decoding is not possible, as all positions of already decoded pixels have to be known in order to determine new pixel positions.

2.2. HRLC

This problem is solved with the — more elaborate — second method, which we call *hierarchical run-length coding* (HRLC). In this method, the pixels with values in an interval $[a,\ b]$ are recursively divided into two classes, nl. a class with all pixels smaller than a threshold t, nl. $[a,\ t]$, and a class with pixels larger or equal to t, nl. $[t,\ b]$. First, the positions of all pixels belonging to one class are determined and coded. Then, for each class, a new threshold is chosen, and each of the two partitions is divided into two new subpartitions. The positions of all pixels of one subpartition within a partition are coded. Only the positions of the pixels of *one* partition have to be coded, of course, as the pixels of the other partition occur at the positions which are left over. The same procedure is carried out again on the subpartitions. If a subpartition which is formed in this way contains only one sort of pixels, no partitioning is carried out anymore. This process goes on until, finally, all pixels have been coded. Once a partitioning is made, the two partitions can be processed in parallel, as the positioning within each partition can be done independently.

A trivial choice for the thresholds is e.g. the powers of two. For high-pass images with mean zero, a first partitioning can be done with a threshold of 0. Then, thresholds of -64 resp. $+64$, to be used within the first two partitions, are chosen.

Bit rate regulation is now carried out by changing the partitioning depth, which is equivalent to applying a uniform quantizer with a changing step size. This method resembles somewhat bitplane coding, where for each bitplane run-length coding is used. The difference is that, with our method lower-order bits belonging to different higher-order bit values are coded separately. With simple bit plane coding, multistage quantization or embedded quantization [1], lower order bits are not coded conditionally on the values of higher-order bits, so a lot of potential for compression is lost in comparison to our method.

Other choices for the thresholds could be equally viable. Some interesting optimizations can be done for a practical coder: after a histogram analysis, the thresholds could be chosen in order to allow better performance of the actual implementation of the TRL coding for the relative positions, by matching the probabilities of the partition runs to the optimal probabilities of the TRL coder (nl. $p = (1/2)^{2^{-m}}$, see [2]).

2.3. GHRLC

GRLC and HRLC are the extremes of a whole spectrum of different coding possibilities. Hybrid systems could use a division into more than 2 partitions. We call these methods *generalized hierarchical run-length coding* GHRLC).

In an implementation of ours, the pixels belonging to an interval $[-a, +a]$ are divided into three partitions: $[-a, -t]$, $[-t, t]$ and $[t, a]$. This partitioning into three classes needs two passes over the data: one to code pixels belonging to the first class, and a second for the pixels belonging to class 3. The rest of the pixels belongs — of course — to the middle class. The middle class is coded again with the GHRLC method, while both other classes are further coded with the HRLC code (with partitioning in two classes). This goes on recursively, until the class size becomes less than two bins of the pixel distribution. The advantage of this method over HRLC is that it is centered around zero, which is favorable for pixel distributions peaked around zero. Bit rate regulation is done by varying the partitioning depth, which is equivalent to using a zero-centered uniform threshold quantizer with variable step size.

2.4. UVLC and MUVLC

A code which resembles very much the codes we use, has been developed by Macq [4]. This method is commonly referred to as the UVLC, which stands for *universal variable length code*, in spite of the fact that this code is not really universal*. The UVLC is very suited for progressive or compatible coding [5]. With this code, the bitplanes are scanned. First, the positions of all pixels with their most significant bit set are coded. The less significant

* A code is universal if its efficiency converges to one when the length of the source increases, regardless of the source's statistics.

bits of those pixels are transmitted literally, i.e. uncoded. Then, a second pass over the data is done, and the pixels with the second most significant bit set, are coded (while skipping those pixels already coded). This goes on until the least significant bit is reached. Bit rate regulation is here equivalent to variable thresholding.

It is assumed that the distribution of the lower order bits when the higher order bit is one, is more or less uniform, so that additional variable length coding does not reduce the bit rate anymore. This assumption is influenced by implementation complexity considerations, as no Huffman coder with its look-up tables is needed in this way. However, we found that the assumption does not hold for Laplacian sources. The distribution of the lower-order bits is significantly different from the uniform distribution, so that using a Huffman code is useful. We call the code where this is done the *modified universal variable length code*.

3. CODER PERFORMANCE COMPARISON

We compared the resulting bit rates when compressing a source with Laplacian distribution with the different coders. The source is pre-quantized with a zero-level uniform quantizer, with 257 levels. The step size of the quantizer is always 1, and the performance is evaluated for distributions with variances ranging from 1 to 25. For the coders where bit rate regulation is equivalent to thresholding (UVLC, MUVLC and GRLC), the reconstruction level the thresholded region is zero. In the case of coders where the bit rate is regulated through changing the step size (HRLC and GHRLC), the reconstruction level is taken equal to the mean of the distribution in the interval which is represented by the corresponding reconstruction level. The distortion is calculated with respect to the pre-quantized Laplacian source. So, when all pixels are coded, the distortion is zero.

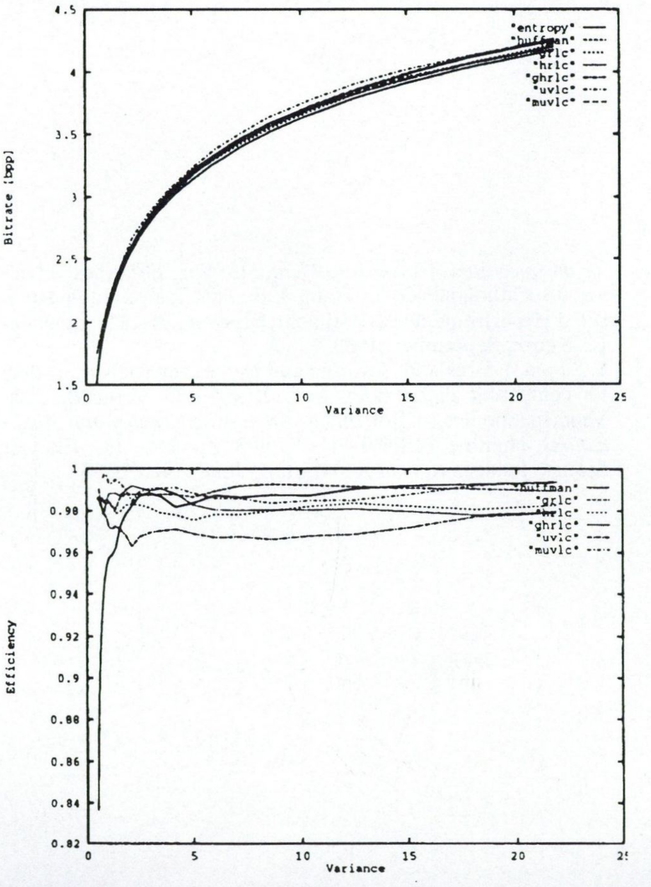


Fig. 1. Bit rate (up) and efficiency (down) for coding Laplacian sources with different variance

In Fig. 1, the bit rate after coding with zero distortion is set out versus the original signal variance. The entropy poses a minimum to the position of the curves. Signals with small variance have low entropy, higher variance leads to higher entropy and higher bit rates. As can be seen, the bit rate B of all coding systems follows very closely the entropy H. The highest bit rate is produced by the UVLC coder. This is due to the rather inefficient representation of the lower-order bits. The difference between the methods can better be seen on the right graph, where the efficiency $\eta = 1 - (B - H)/H$ is set out vs. the variance. For low bit rates (around 1 bpp) the Huffman code performs not well. At higher bit rates, however, the Huffman coder's performance leads the pack. The efficiency of all the other methods, except the UVLC code, is always higher than 98%. For intermediate variances (5 to 15) the GRLC is best.

The results show that the methods we propose, nl. GRLC, HRLC and GHRLC perform comparably to existing methods (Huffman and MUVLC) in terms of efficiency, even when their special features, nl. combined compression and bit rate regulation, are not exploited.

Figs. 2-3 show the distortion-rate curves for the different coders, in the case of a Laplacian source with variance $\sigma^2 = = 1, 4, 16$ and 64. The distortion (the mean square quantization error) is normalized on the signal variance σ^2. For each coder, we regulated the bit rate by the quantization features of the coders. We remark again that the code-stream for lower distortion is simply a superset of the higher-distortion code-stream, so that SNR-scaling is very easily implemented.

The UVLC, MUVLC and our GRLC perform comparably. The MUVLC performs consistently a bit better than UVLC, which is simpler to implement. The bit rate of the GRLC method can be regulated more finely in comparison to MUVLC and UVLC: this is due to the fact that with GRLC, thresholding can occur at any value, while MUVLC and UVLC only allow thresholding at values which are powers of two.

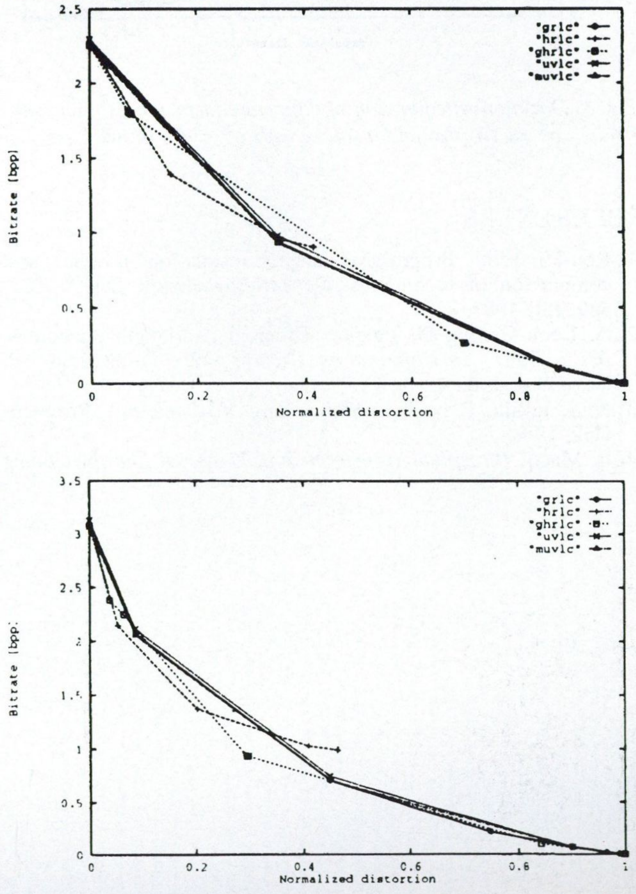


Fig. 2. Disortion-rate function of different coders, for a source with $\sigma^2 = 1$ (up) and a source with $\sigma^2 = 4$ (down)

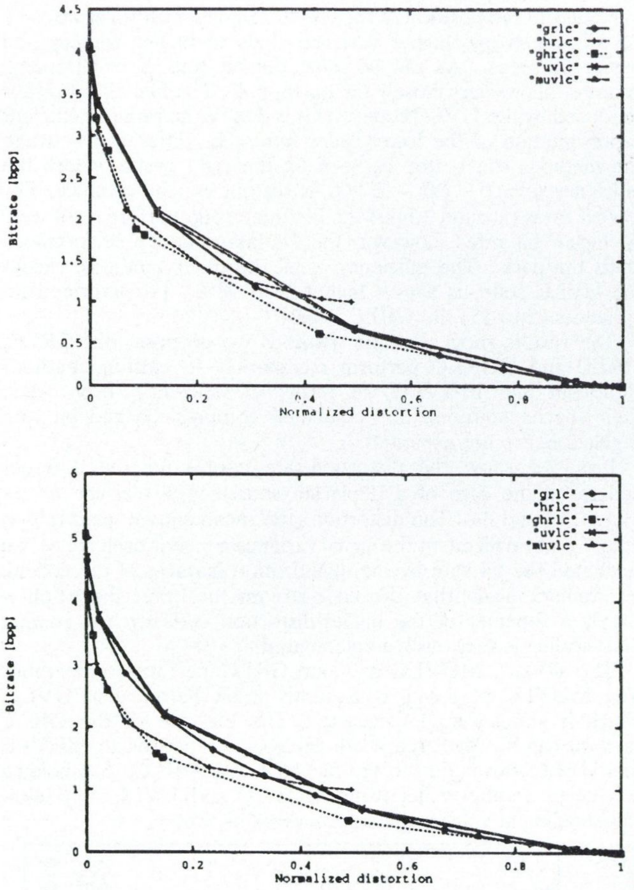


Fig. 3. Disortion-rate function of different coders, for a source with $\sigma^2 = 16$ (up) and a source with $\sigma^2 = 64$ (down)

The HRLC method performs best for higher bit rates, but its performance degrades for the lower rates, where it gets stuck around bit rate=1 bpp. This is because in the case of lower bit rates, only two reconstruction levels remain, one for the positive and zero-valued pixels and one for the negative pixels, each of which occur with a probability of about 0.5. So, this method can only be used for relatively high-quality coding.

For sources with small variance ($\sigma^2 = 1$) the GHRLC method does not allow very fine control over the bit rate. However, when the source variance goes up, the GHRLC method with changing step size clearly gets ahead of the methods which simply perform a thresholding (UVLC, MUVLC, GRLC): the bit rate is always lower for a fixed distortion, and while the control over the bit rate is not as fine-grained as with GRLC, it is clearly better than with the UVLC/MUVLC.

So, generally the GHRLC method seems best.

4. CONCLUSION

In this paper, we have shown that variable length coding which is based on coding of the pixel positions is equivalent in terms of entropy to direct pixel value coding. Using this theoretical result, we designed a family of coding methods which are very suited for progressive transmission of pictures with a Laplacian distribution, as can be found in the case of sub-band coding, transform coding or predictive coding. We proposed a "generalized hierarchical run-length coder" which allows fine control over the bit rate while maintaining high efficiency and a very good distortion-rate characteristic. Furthermore, the method does not need any look-up tables, and it can be implemented in parallel.

REFERENCES

[1] Kou-Hu Tzou: Progressive image transmission: a review and comparison of techniques, *Optical Engineering*, 26(7): 581-589, July 1987.

[2] A. Leon-Garcia, H. Tanaka: Efficient run-length encodings, *IEEE Trans. on Information Theory*, Vol. IT-28(6):88-890, November 1982.

[3] N. S. Jayant, P. Noll: *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[4] B. Macq: *Perceptual Transforms and Universal Entropy Coding for an Integrated Approach to Picture Coding*, PhD thesis, Universitè Catholique de Louvain, Lab. de Telecommunication set d'Hyperfrèquences, Bâtiment Maxwell, B-1348 Louvain-La-Neuve, September, 1989.

[5] S. Simon, I. Bruyland: Layered and packetized subband coding for compatible digital hdtv. In C. Braccini G. Vernazza, A.N. Venetsanopoulos, editor, *Image Processing: Theory and Applications*, Number ISBN 0-444-89969-3, pp. 115-118. Elsevier Science Publishers, San Remo, Italy, June, 14-16 1993.

IMAGE COMPRESSION, CODING II.

BLOCK PREDICTION IN VECTOR QUANTIZATION

S. FIORAVANTI
DIBE,
UNIVERSITY OF GENOVA, ITALY

D. D. GIUSTO
INSTITUTE OF ELECTRICAL ENGINEERING,
UNIVERSITY OF CAGLIARI, ITALY

1. VECTOR QUANTIZATION AND VECTOR PREDICTION

Vector quantization is a well-known strategy [1,2], widely used in many application fields, as well as for image-data compression [3]; main reasons for its consensus lie in its simple implementation and good rate-distortion performances in many applications [4].

Briefly, it consists in the extension of scalar quantization to an ordered set of real numbers (a block in the original data set, that is, a vector) coming from monodimensional signals as well as from multidimensional ones. More precisely, it is defined as an operator T that maps a vector belonging to an n-dimensional Euclidean space $\mathbf{R}^n$ into a finite subset $\mathbf{S}$ of $\mathbf{R}^n$ made up of only N vectors (i.e., the codewords or codevectors, $\mathbf{S}$ being the codebook).

Two crucial points can be identified: the mapping operation, which aims at searching, for a given vector, the nearest codevector; and the identification of the S space, that is the optimal codebook generation. Two common strategies are usually employed. For the former problem, the Mean Square Error (MSE) is very popular, while for the latter, the classical suboptimal strategy propesed in [5] is adopted in many cases.

The simple vector quantizer described above operates on vectors as single entities, generating only a series of codevectors; therefore it is called memoryless VQ. Form the point of view of data compression, the resulting bitrate B needed to code a vector by a memoryless VQ equals to $B = (\log_2 N)/n$. However, in applications dealing with highly correlated sources (in particular, images), the rate-distortion theory states that high performances are reachable only by employing prohibitively large codevectors and codebooks. However, one can achieve lower bitrates also by exploiting the notable amount of redundancy between neighboring codevectors, which are usually highly correlated; it is sufficient to calculate the transition matrices to understand that memoryless VQ is not able to exploit this notable amount of redundancy. These matrices are defined as $[\mathbf{M}_d]_{p,q} = P(\mathbf{x}_p|\mathbf{x}_q)$ where the entries (i, j) are proportional to the probability of finding the vector $\mathbf{x}_p$, given the vector $\mathbf{x}_q$ in a previous position, in a raster scan, in one of the main directions d, that is 0, 45, 90 and 135 degrees; the inter-vector redundancies result in matrices very sparse.

So, in order to bypass this intrinsic limit of VQ and to improve its performances, considerable efforts have been devoted to exploiting this inter-block correlation. A widely known strategy is the so-called Finite-State Vector Quantization (FSVQ), which was first proposed in [6,7], and subsequently used as a basis for various algorithms (e.g., [8,9]).

In [10], an FSVQ-like strategy was implemented through a cache-based lossless prediction scheme (Cache VQ, CVQ); in short, it consists in the generation and exploitation of a reduced codebook, by taking advantage of the information coming from the neighboring-blocks transition matrices. Later on [11], the same strategy was implemented using, as predictor, a two-layers neural network.

A slightly different and more performing implementation of this strategy is presented in the paper, followed by the proposal of a novel and different predicted VQ scheme. Within this scheme, the prediction step, carried out by a neural network in the same way as before, is aimed at reordering the codebook dynamically (Dynamic Codebook Reordering, DCR), thus making possible an efficient Huffman encoding of codevector addresses, as the entropy of address data is notably reduced.

2. THE PROBLEM OF PREDICTION AND THE NEURAL APPROACH

The most powerful approach to the problem of prediction is to find a law underlying the given dynamic process of phenomenon [12]. If such a law can be discovered and analytically described (e.g., by a set of ordinary differential equations), then by solving them it is possible to predict the future, when the initial conditions are completely specified. Unfortunately, the information about a dynamic process is often only partial and incomplete; so the prediction cannot be based on a known analytical model. A less powerful approach attempts to discover some empirical regularities in the observation of the series. According to this, the unknown dynamic process is described by a generic (usually nonlinear) multivariable function $z_1 = F[z_{t-1}, z_{t-2}, \ldots, z_{t-K}]$ where the $\{z_i\}$'s are given samples of the series, and $F(\bullet)$ is an unknown function. In the simplest case, this function is linear and a standard auto regressive (AR) model can be used $\hat{z}_1 = \sum_{k=1}^{K} a_k z_{t-k}$ where the predicted value is given in terms of a linear combination of a fixed number K of past values of the series [13]. This model provides good results only if a dynamic process is, of course, linear or nearly linear. For highly nonlinear processes, the AR model-based prediction may be very poor or completely wrong. Therefore, a more flexible and universal approach is to employ a neural network able to approximate, with nonlinear processing units, any nonlinear continuos function on the basis of training examples [14]. As said, the advantage of the neural network model lies in its generality and flexibility. The training process produces multidimensional surface composed by a set of simple nonlinear functions that fit the training set in some best way. The neural network is trained on the basis of the available examples to find such a function.

The ability of neural networks to generalization and prediction of the future can be proved by the approximation theory [15].

3. STRUCTURES AND TRAINING OF THE NEURAL PREDICTOR

The proposed approach involves the implementation of a three-layers perception neural network [16], trained to reconstruct a codevector $\mathbf{x}_{i,j}$ on the basis of the 4 neighboring vectors previously decoded, that is $\mathbf{x}_{i,j} = F[\mathbf{x}_{i-1,j-1}, \mathbf{x}_{i-1,j}, \mathbf{x}_{i-1,j+1}, \mathbf{x}_{i,j-1}]$. The functional relationship $F'(\bullet)$ of the neural network approximates the unknown function $F(\bullet)$ on the basis of the examples given as a training set, so obtaining a vector $\mathbf{x}'_{i,j}$ prediction of the codevector $\mathbf{x}_{i,j}$, that is $\mathbf{x}'_{i,j} = F'[\mathbf{x}_{i-1,j-1}, \mathbf{x}_{i-1,j}, \mathbf{x}_{i-1,j+1}, \mathbf{x}_{i,j-1}]$. In particular, the adopted training set is the same set of images utilized to build up the VQ codebook. For a good generalization, this set has been selected with dimensions satisfying the constraints given in [17]. Such constraints relate the number of weights to the number of patterns to be learned.

The classical back-propagation training algorithm has been adopted [18]. This algorithm aims at minimizing the distance among the vector $\mathbf{x}'_{i,j}$ (generated by the network on the output layer) and the current codevector $\mathbf{x}_{i,j}$, after the input layer has received a configuration made up of the 4 neighboring codevectors preceding it. In other words, the input layer has the dimension that equals to four times the dimension of a vector, while the output one has simply the dimension of a vector. The dimension of the hidden layer has been experimentally chosen two times the dimension of a vector.

Each neuron (except for the input ones) is characterized by a nonlinear activation function $f(z)$ of the hyperbolic tangent type $f(z) = (e^z - e^{-z})/(e^z + e^{-z})$ $f : \mathbf{R} \rightarrow (-1, 1)$. To avoid a compression of the output dynamics, due to the saturation of $f(z)$ for the lower and upper values, the output activation functions are driven to operate in the linear region by subtracting, from the desired values, the mean gray value of the 4 nehgboring codevectors. The targets $\mathbf{t}_p = \mathbf{t}_{i,j}$ of the network are:

$$\mathbf{t}_p = \mathbf{x}_{i,j} - \left[\frac{1}{4 \cdot n} \sum_{m=1}^{n} x_{i-1,j-1}(m) + \right.$$
$$\left. + x_{i-1,j}(m) + x_{i-1,j+1}(m) + x_{i,j-1}(m) \right] \cdot \mathbf{1} \quad (1)$$

By defining $\mathbf{x}$ the input vector, $\mathbf{h}$ and $\mathbf{y}$ the hidden and output ones, respectively, the output depends on the inputs as $\mathbf{h} = f\{[W^1]\mathbf{x}\}$; $\mathbf{y} = f\{[W^2]\mathbf{h}\}$. The purpose of the network is to produce ouputs $\mathbf{y}$ that are as equal to the target $\mathbf{t}_p$ as possible; in other words, it aims to minimize the global error on all the targets $\mathbf{t}_p$, with $p = 1, \ldots, L$ (L being the number of training-set elements), associated with the inputs

$$E = \sum_{p=1}^{L} \parallel \mathbf{y}_p - \mathbf{t}_p \parallel^2 = f\{[W^1], [W^2]\} \quad (2)$$

If E is regarded as a surface that is a potential function of the weights, a steepest-descent minimization can be performed. Let us consider only one vector, for the sake of brevity; we have

$$E_p = \parallel \mathbf{y}_p - \mathbf{t}_p \parallel^2 = \Sigma_i (y_p(i) - t_p(i))^2 \quad E = \Sigma_p E_p \quad (3)$$

and then

$$\Delta[W^1]_{i,j} = -\varepsilon \Sigma_p \frac{\partial E_p}{\partial [W^1]_{i,j}} \quad (4)$$

To obtain the vector on which the minimization step is to be performed, the gradient is computed as follows

$$h_j = f(\Sigma_i [W^1]_{i,j} \cdot x_i) \quad y_i = f(\Sigma_j [W^2]_{i,j} \cdot h_j) \quad (5)$$

$$\frac{\partial E}{\partial [W^2]_{h,k}} = \frac{\partial}{\partial [W^2]_{h,k}} \Sigma_i (y_i - t_i)^2 = 2\Sigma_i (y_i - t_i) \frac{\partial y_i}{\partial [W^2]_{h,k}} =$$
$$= 2(y_h - t_h) \frac{\partial y_h}{\partial [W^2]_{h,k}} = 2(y_h - t_h) h_k \frac{\partial f(x)}{\partial x} |_{f=y_i} \quad (6)$$

As the function $f(x)$ is of the sigmoid type, then $\frac{\partial f(x)}{\partial x} |_{f(x)=y_i} = 1 - y_i^2$ and then

$$\frac{\partial E}{\partial [W^2]_{h,k}} = 2(y_h - t_h) h_k (1 - y_h^2) \quad (7)$$

The input weights vary according to

$$\Delta[W^1]_{n,m} = -\eta \frac{\partial E}{\partial [W^1]_{n,m}} = -\eta \frac{\partial E}{\partial h_n} \frac{\partial h_n}{\partial [W^1]_{n,m}} \quad (8)$$

$$\delta = \frac{\partial E_p}{\partial h_n} = 2\Sigma_k (y_k - t_k) \frac{\partial y_k}{\partial h_n} = 2\Sigma_k (y_k - t_k) [W^2]_{k,n} (1 - y_k^2) \quad (9)$$

$$\frac{\partial h_n}{\partial [W^1]_{n,m}} = x_m (1 - h_n^2) \quad (10)$$

The resulting weight updating rule is the following

$$\Delta[W^1]_{n,m} = 2\varepsilon \delta_n (1 - h_n^2); \; \Delta[W^2]_{h,k} = 2\varepsilon (y_n - t_h) h_k (1 - y_h^2) \quad (11)$$

This learning rule tends to converge to local minima of the energy function, if any. To bypass such minima, in the descent phase, a moment of inertia is included:

$$\Delta[W^1]_{i,j}(t) = -\varepsilon \Sigma_p \frac{\partial E_p}{\partial [W^1]_{i,j}} + \alpha \Delta[W^1]_{i,j}(t-1), \quad (12)$$

where t stands for the iteration index. Moreover, in order to speed up the convergence, some heuristic criteria are applied [19], concerning the variations in the parameters α and ε. Such criteria are:

- weights are updated according to (12);
- if a global error turns out to be smaller than the previous one, then $\varepsilon = \phi\varepsilon$, with $\phi >> 1$, and the weights are updated;
- otherwise, $\varepsilon = \beta\varepsilon$, with $\beta << 1$, $\alpha = 0$ and the current iteration is discarded.

In summary, at each iteration, when a set of codevectors (corresponding to the neighboring codevectors preceding the one to be predicted, in a raster scan) are presented to the network input, the algorithm applies the updating rule to the synaptic weights to minimize the MSE between the vector generated by the network and the target.

3. NEURAL PREDICTED CACHE VQ (CVQ)

In [10], the Cache VQ was proposed, that is a predicted VQ scheme (conceptually based on FSVQ) that allows one to obtain higher compression factors, without causing further losses in the signal-to-noise ratio (SNR). The basic idea starts from the observation of the transition matrices, and develops itself through a prediction of a codevector on the basis of these transition probabilities. In practice, on the basis of the previously decoded neighboring codevectors, a score function f_s is computed for each codevector $\mathbf{x}(h)$ in the codebook; such a function is equal to

$$f_s\{\mathbf{x}(h)\} = [\mathbf{M}_0]_{x(h), x_{i,j-1}} + [\mathbf{M}_{45}]_{x(h), x_{i-1,j-1}} +$$
$$+ [\mathbf{M}_{90}]_{x(h), x_{i-1,j}} + [\mathbf{M}_{135}]_{x(h), x_{i-1,j+1}} \quad (13)$$

The codevectors $\mathbf{x}(k)$ ($k = 0, \ldots, 2^b - 1$, $b < \log_2 N$, N being the size of the codebook) with higher f_s are stored in a codebook of reduced dimensions, that is a cache codebook. By addressing this smaller codebook, if the correct codevector is contained in it, one can achieve a bitrate reduction. In other words, if the current codevector $\mathbf{x}_{i,j}$ is one of the $\{\mathbf{x}(k)\}$'s, only b bits are required to transmit its address within the cache codebook, otherwise, a fault configuration (i.e. the "0" one) is transmitted, followed by the $\log_2 N$-bit address of the codevector $\mathbf{x}_{i,j}$ in the overall codebook.

In the case of a correct prediction, the bitrate is multiplied by a factor $F_{cp} = b/(\log_2 N)$ whereas, in the case of a wrong prediction, it is multiplied by a factor $F_{wp} = (b + \log_2 N)/(\log_2 N)$. Denoting by P_{cp} the percentage of correct predictions within an image, and by P_{wp} the percentage of wrong ones (that is, $P_{cp} = 1 - P_{wp}$), the resulting final bitrate B_f is equal to

$$B_f = B \cdot F_{cp} \cdot P_{cp} + B \cdot F_{wp} \cdot P_{wp} = B \frac{b}{\log_2 N} P_{cp} +$$
$$+ B \frac{b + \log_2 N}{\log_2 N} (1 - P_{cp}) \quad (14)$$

The threshold value of P_{cp} over which a gain is achieved can be computed by setting $B_f = B$ in (14). So $P_{cp}^T > b/(\log_2 N)$ gives the condition on the percentage of correct prediction to achieve a bitrate reduction.

Regarding the architecture, the decoder must be of course synchronous with the encoder, in the sense that it has to accomplish the same sequence of operation to find out the current codevector. For each codevector, the prediction is calculated at the decoder in the same way as at the encoder, and the same cache codebook is generated. If the b-bit configuration received is a valid address, the vector is selected from the reduced codebook; otherwise (fault condition), the next $\log_2 N$ bits are used to address the overall codebook directly.

However, such a scheme exhibits some drawbacks that make it not so attractive for software and hardware implementation. First of all, it requires the storage of the transition matrices, which are quite large. Moreover, the computation of the score function for all the codevectors poses severe problems; this task requires that one access the transition matrices too many times, and is therefore very time-consuming. Instead, the prediction carried out by a three-layer neural network features many advantages: an easy hardware realization; no requirements for additional storage (except for the synaptic weights of the neural network); and above all, the considerable generalization, as the performance are similar whether images belong or not to the training set (as reported in the results section).

The prediction process does not change so much; every time a codevector has to be transmitted, the network receives the four neighboring codevectors (i.e. the codevectors already coded) as inputs, and generates a predicted vector (not a codevector) on the output layer. The prediction result is then used to organize a cache codebook that contains the codevectors $\mathbf{x}(k)$ ($k = 0, \ldots, 2^b - 1$, $b < \log_2 N$) at the minimum distance from the predicted vector $\mathbf{x'}_{i,j}$. As in the previous case, if the current codevector $\mathbf{x}_{i,j}$ is one of the $\{\mathbf{x}(k)\}$'s, only b bits are required to transmit its address within the cache codebook; otherwise, a fault configuration (i.e., the "0" one) is transmitted, followed by the $\log_2 N$-bit address of the codevector $\mathbf{x}_{i,j}$ in the overall codebook.

The values of the synaptic weights computed during the training phase are stored and used by the encoder and the decoder to achieve the prediction in the same way. Of course, also in this case the decoder must be synchronous with the encoder. Concerning the architecture of the neural-CVQ encoding-decoding system, it is the same as in the case without neural network.

4. DYNAMIC CODEBOOK REORDERING VQ (DCRVQ)

The CVQ presented above shows the disadvantage to be not dynamic in the choice of the reduced-codebook dimension, even if the predictor succeeds, the bitrate is fixed (two choices), and the power of the predictor is not exploited at its best.

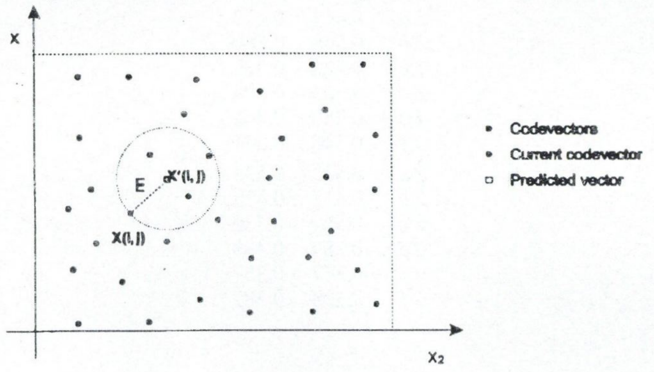


Fig. 1. The basic idea of DCR vector quantization

The optimal way to exploit the predictor is to code the difference between the predicted vector and the codevector to be transmitted. If one thinks in terms of vectors in a 2D space (as the Fig. 1), a codevector can be coded by the distance between it and the predicted vector, where distance stands for the number of codevectors with lower Euclidean distances from the prediction. At the best, if the predictor succeeds every time (the nearest codevector is the correct one), a sequence of 0 will be generated. However, as in the real cases, the predictor produces vectors near enough to the correct codevector, so maintaining lower distances (in terms of number of vectors in between).

The practical way to implement this idea is to reorder in a dynamic way (i.e., for each codevector to be transmitted) the codebook on the basis of the distances from the predicted vector (0 means the nearest, N the farthest); this strategy is called Dynamic Codebook Reordering VQ (DCRVQ). As a result, a sequence of addresses quite similar is produced, characterized by a low entropy. To enhance this fact, an example of an addresses' histogram for a memoryless VQ is presented in Fig. 2, together with an example of the histogram of addresses with the dynamic reordering procedure. In the last case, the energy of the codebook addresses is gathered near low values, so achieving a very low entropy. This allows for a very efficient entropy coding of the latter sequence, if compared to the former one.

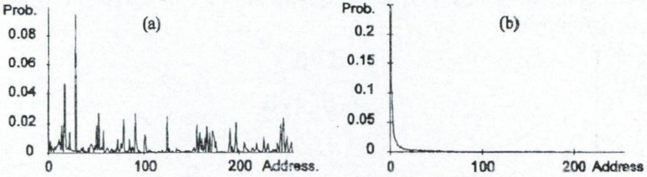


Fig. 2. (a) An example of the addresses' histogram for a memoryless vector quantizer; (b) the same example but for a dynamic-codebook-reordering vector quantizer

As a comment, this strategy aims to produce something correlated, while the standard VQ, as well as CVQ (even if with a minor grade), produces usually something uncorrelated. The global architecture of the encoding-decoding system is displayed in Fig. 3, and consists of a standard vector quantizer, the neural predictor, the codebook reordering module, and the entropy coder. The vector quantizer operates, at the encoder, by associating to a given vector a codevector $\mathbf{x}_{i,j}$ minimizing the MSE between them; at the decoder, it simply works as a look-up table. The entropy coder performs the compression of the codevector addresses (in the codebook) by the classical Huffman algorithm [20].

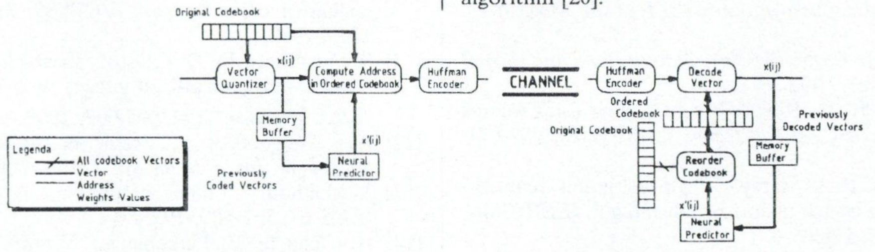


Fig. 3. The global architecture of a dynamic-codebook-reordering vector quantizer

The neural predictor is identical both at the encoder and at the decoder and consists, as said above, of a three-layer perception neural network, trained to reconstruct a codevector on the basis of the four neighboring codevectors previously decoded. The values of the synaptic weights computed during the training phase are stored and used by both encoder and decoder. In order to decrease the bitrate without further losses, the networks work in a synchronous way, in the sense that they receive the same data (i.e., the previously coded codevectors), so producing the same predicted vector. The predicted vector $\mathbf{x'}_{i,j}$ is used to dynamically organize an ordered codebook. First, the coder computes both the MSE between each codevector $\mathbf{x}(h)$ and $\mathbf{x'}_{i,j}$, that is $E_h = \|\mathbf{x}(h) - \mathbf{x'}_{i,j}\|^2$, and the MSE between $\mathbf{x}_{i,j}$ and $\mathbf{x'}_{i,j}$ that is $E = \|\mathbf{x}_{i,j} - \mathbf{x'}_{i,j}\|^2$. The address to be transmitted equals to $a = \dim\{E_l; l = 1, \ldots, N : E_l < E\}$. The decoder is synchronous with the encoder, so it has to compute the predicted vector, the MSEs $\{E_h\}$'s, and to reorder the codebook on the basis of these $\{E_h\}$'s (i.e., by assigning address 1 to the codevector corresponding to the smallest E_h, and so on). Then it has to extract, from the reordered codebook, the codevector corresponding to the transmitted address. The computational complexity of the coding phase is comparable with standard vector quantizers, whereas the decoder has to execute a larger number of operations.

5. RESULTS AND COMPARISONS

Results reported in the following were obtained on two sets of images (first set: a) aircraft; b) Carmen; c) jeans; d) masquerade; e) sweater; f) shoes; second set: a) agave; b) LENA; c) bike; d) peppers; e) fiji; f) Tiffany). No decoded images are shown as the prediction step is carried out in a lossless way; so the final quality is the same as in memoryless VQ. In particular, the used vector quantizer features vectors of dimensions 3 * 3 pels, 256 gray-levels per pel, and a codebook made up of 256 codevectors (same properties of the original vectors).

Several tests have been done to assess the performances of the two coding scheme. In particular, Table 1, which refers to the neural-predicted CVQ, shows the results when the cache codebook contains 8 codevectors. Bitrates are given in the two different cases: images within the training set (that is, the training set is made up by the same image set) and outside the training set (that is, the training set is made up by the other image set). Table 2, instead, furnishes the results of DCRVQ; legenda is the same as for Table 1. Finally in Fig. 4, a comparison among the two VQ schemes is presented. The most important aspect of the neural prediction, as pointed out by the results, is its flexibility: the neural-predictor performances are quite similar also for images outside the training set. Moreover, DCRVQ is more efficient than CVQ, as it exploits intervector correlations in a dynamic way.

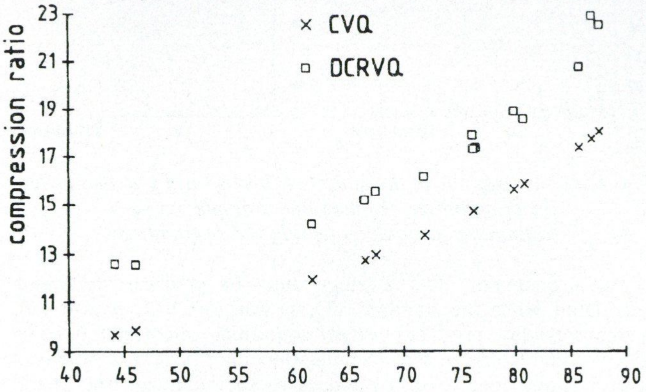


Fig. 4. Performance comparison for neural-predicted CVQ and DCRVQ

REFERENCES

[1] R. M. Gray: "Vector Quantization", *IEEE-ASSP Mag.*, 4-29, April 1984.

[2] A. Gersho, R. M. Gray: *Vector Quantization and Signal Compression*, Kluwer, 1992.

[3] N. M. Nasrabadi, R. A. King: "Image coding using vector quantization: A review", *IEEE Trans.* COM-36(8), 957-971, 1988.

[4] T. D. Lookabaugh, R. M. Gray: "High-resolution quantization theory and the vector quantizer advantage", *IEEE Trans.* IT-35(5),1020-1033, 1989.

[5] Y. Linde et al.: "An algorithm for vector quantizer design", *IEEE Trans.* COM-28(1), 84-95, 1980.

[6] J. Foster et al.: "Finite-state vector quantization for waveform coding", *IEEE Trans.* IT-31(3), 348- 359, 1985.

[7] M. O. Dunham, R. M. Gray: "An algorithm for design of label-transition finite-state vector quantization", *IEEE Trans.* COM-33(1), 83-89, 1985.

[8] A. Aravind, A. Gersho: "Image compression based on vector quantization with finite memory", *Opt. Eng.* 26(7), 570-580, 1987.

[9] T. Kim: "New finite state vector quantizers for images", *Proc. IEEE-ICASSP,* 1180-1183, 1988.

[10] F. G. B. DeNatale et al.: "A framework for high-compression coding of color images", VCIP'89, *Proc. SPIE* 1199,1430-1439, 1989.

[11] S. Fioravanti, D. D. Giusto: "Inter-block redundancy reduction in vector-quantized images by a neural predictor", *Eur. Trans. Telecom.* 3(6), 605-607, 1992.

[12] A. S. Weigend et al.: "Predicting the future: A connectionist approach", *Int. J. Neur. Sys.* 1, 193- 209, 1990.

[13] J. Makhoul: "Linear prediction: A tutorial review", *Proc. IEEE* 63, 561-580, 1975.

[14] A. Cichocki, R. Unbehauen: *Neural Networks for Optimization and Signal Processing*, Wiley, 1993.

[15] J. J. Hopfield: "Neurons with graded response have collective computational properties like those of two-state neurons, *Proc. USA Nat. Acad Sc.* 81, 3088-3092, 1984.

[16] M. Minsky, S. Papert: *Perceptrons,* MIT Press, 1972.

[17] E. B. Baum, D. Haussler: "What size gives valid generalization", *Neur. Computing* 1, 151-160, 1989.

[18] D. E. Rumelhart et al.: "Learning representations by backpropagating errors", *Nature* 323, 533- 536, 1986.

[19] T. P. Vogl et al.: "Accelerating the convergence of the backpropagation method", *Biol. Cyber.* 59, 257-263, 1988.

[20] D. A. Huffman: "A method for the construction of minimum redundancy codes", *Proc. IRE* 40, 1098-1101, 1952.

6. ACKNOWLEDGEMENTS

This work was funded by the National Research Council of Italy (CNR) within the framework of the National Target Project on Telecommunications.

Table 1. Results of neural predicted CVQ. The given bitrates refer to: BR (a), images are within the training set; BR (b), images are outside the training set (made up by the other image set).

Image	BR (a)	BR (b)
1.a	0.550	0.554
1.b	0.681	0.680
1.c	0.519	0.515
1.d	0.587	0.588
1.e	0.546	0.538
1.f	0.450	0.449
2.a	0.549	0.548
2.b	0.509	0.512
2.c	0.632	0.631
2.d	0.462	0.461
2.e	0.437	0.436
2.f	0.484	0.476

Table 2. Results of DCRVQ. The given bitrates refer to: BR (a) images are within the training set; BR (b), images are outside the training set (made up by the other image set).

Image	BR (a)	BR (b)
1.a	0.467	0.472
1.b	0.568	0.569
1.c	0.422	0.429
1.d	0.503	0.503
1.e	0.458	0.463
1.f	0.346	0.341
2.a	0.522	0.533
2.b	0.437	0.438
2.c	0.529	0.530
2.d	0.389	0.388
2.e	0.357	0.357
2.f	0.388	0.387

PARALLEL ARCHITECTURES FOR MULTIRESOLUTION IMAGE SEQUENCE CODING

K. FAZEKAS

DEPT. OF MICROWAVE TELECOMMUNICATION
TECHNICAL UNIVERSITY OF BUDAPEST
GOLDMAN TÉR 3, BUDAPEST, HUNGARY H-1111

1. INTRODUCTION

In recent years, a considerable amount of research has been focused on image data compression, which plays a significant role in image processing and transmission. Videotransmission will become a major service of future video communication networks. Improvement of network transmission capabilities — and particularly, introduction of the B-ISDN — will allow new services such as HDTV, TV to reach customers through optical fiber cables. The transmission of these digital image sequences requires that a very large amount of data be transmitted. Image compression techniques are used to reduce the redundancies in the sequences and make their transmission more economical.

In the last decades, several new requirements came into being: progressive image transmission, compatible coding, layered coding for networks. Progressive image transmission allows an approximative image to be built up quickly and the details to be transmitted progressively through several passes over the image. Compatibility between the various services which will reach customers (HDTV, TV, videophone, etc.) is clearly requested. In layered coding, the video information is divided into several layers with lower layers containing low resolution information and higher layers the fine information in descending order of importance. Such a model has the potential to enable integration of the several video services for packet switched networks. The solutions of these problems are related to splitting the image signal into required parts and to effectively coding these subimages.

Digital image processing operations — naturally involved image coding operations — are usually computationally intensive, because of the large amount of data that must be processed and of the complexity of the elementary operations. In addition, in the cases of video coding and transmission all calculations must be performed in real-time i.e., at a rate of 25 – 30 images per second. To achieve a throughput rate adequate for above mentioned applications, the processor arrays appear to be massively concurrent processing by special-purpose hardware. A processor array is a collection of many similar processing elements (PEs), which can be executed in both parallel and pipeline processing. Progress of VLSI technology, which allows hundreds of thousands of transistors to be manufactured in a single chip, has lowered implementation costs for large processor arrays to an acceptable level.

2. MULTIRESOLUTION IMAGE DECOMPOSITION

The Fourier transform and its extensions have historically been the prime vehicle for signal analysis and representation. Since the early 1970s, orthogonal block transforms with real basis functions, have been studied extensively for transform coding applications. The availability of simple fast transform algorithms and good signal coding performance made the DCT the standard signal decomposition technique, particularly for image and video. The international standard image/video coding algorithms, i.e., CCITT H.261, JPEG, MPEG I – IV, all employ DCT-based transform coding.

Since the recent research activities in signal decomposition are basically driven by visual signal processing and coding applications, the properties of the human visual system (HVS) are examined and incorporated in the signal decomposition step. The VHS inherently performs multiresolution signal processing.

The multiresolution signal analysis concept fits a wide spectrum of visual signal processing and visual communications applications. Lower, i.e., coarser resolution, version of an image frame or image sequence are often sufficient in many instances. Progressive improvement of the signal quality in visual applications, from coarse to finer resolution, has many uses in visual communications and related fields.

Subband coding (SBC) is well established technique used in a variety of image processing applications. SBC based techniques use splitting of the image into subimages called subbands, which are coded separately using individually optimized quantizers and coders. This provides the flexibility to dynamically allocate bits to different frequency bands, thereby shaping the signal noise spectrum according to perceptual or other criteria. The actual bit rate is determined by the importance of the particular band for the image at hand. Thus, by closely matching the individual band statistics, the coding of each subband separately can be done more accurately than the coding of the entire image.

Multiresolution transforms are also known as subband decomposition techniques or pyramid transforms. Each of them have been developed separately and is expressed with different mathematical tools. The wavelet theory as it has been introduced recently relate all these techniques by associating a mathematical modeling to get a better understanding.

Wavelet transform recently have been proposed as a new multiresolution decomposition tool for continuous time signals. The kernel of the wavelet transform is obtained by dilation and translation of a prototype bandpass function. The discrete wavelet transform employs discretized dilation and translation parameters. The wavelet transform permits a decomposition of a signal into the sum of a lower resolution (or coarser) signal plus a detail, much like the dyadic subband tree in the discrete-time case. Each coarse approximation in turn can be decomposed further into yet a coarser signal and a detail signal at that resolution.

3. PARALLEL ARCHITECTURES

The applicability of an algorithm and its adequate architecture to image sequence coding problems strongly depends on the speed of operation. There are a number of alternative approaches to the problem of speeding up an algorithm, namely

- hardware acceleration,
- design of fast sequential algorithms,
- design of parallel algorithms.

Image compression is an important technique in image processing. In order to implement this processing by hardware, one can use several array architectures. For real-time image sequence coding, general-purpose signal processors are often too expensive or too slow to the incurred supervisory overhead. To achieve a throughput rate adequate for real-time applications, an alternative appears to be massively concurrent processing by special-purpose hardware, e.g., processor arrays. A processor array is a collection of many similar processing elements (PEs), which can be executed in both parallel and pipeline processing. VLSI processor arrays dedicated for specific applications appear to be effective, feasible, and economic.

Parallel algorithm expression is a basic tool for a proper description of an algorithm for parallel and pipeline processing. There are a number of parallel algorithm expressions, such as snapshot, recursive equation, parallel code, single assignment code, dependence graph (DG), signal flow graph (SFG). etc. A good expression should express algorithms clearly and concisely so that the execution of the algorithm can be easily pursued.

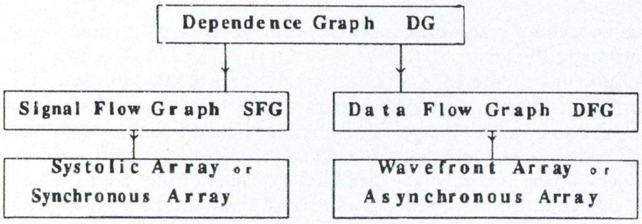
Here, we will use the following expressions:

- *dependence graph* **DG** is a directed graph, which is embedded in an index space and specifies the data dependences of an algorithm;
- *signal flow graph* **SFG** which explicitly express all time information in the form of delay edges is suitable to represent a synchronous array operations;

- *data flow graph* **DFG** is suitable for the expression of asynchronous array operations.

The mapping from an algorithm to processor arrays consists of three stages (Fig. 1):

- express an algorithm by the DG notation and, if necessary, modify the DG to achieve a better algorithm,
- map the DG to an SFG or a DFG array, and
- convert the SFG array to a synchronous array and the DFG to an asynchronous array.



Dependence Graph DG
Signal Flow Graph SFG
Data Flow Graph DFG
Systolic Array or Synchronous Array
Wavefront Array or Asynchronous Array

Fig. 1. Mapping

Two popular special-purpose processor array architectures are systolic and wavefront arrays. By Kung [15], "A systolic system is a network of processors which rhythmically compute and pass data through the system". The systolic array features the important properties of modularity, regularity, local interconnection, a high degree of pipelining, and highly synchronized multiprocessing. The data movements in the systolic arrays are controlled by global timing reference beats. A wavefront array is similar to the systolic array. However, the wavefront array does not employ global synchronization, instead, each PE has its own local clock (self-timed) and exchanges data with neighboring PEs by asynchronous handshaking.

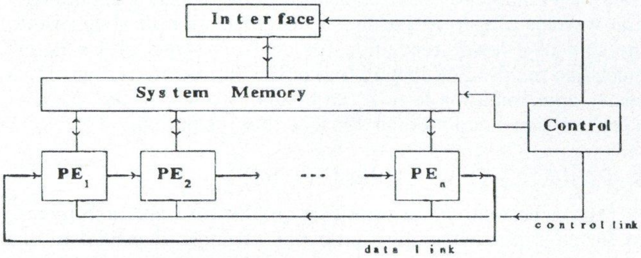


Fig. 2. The ring systolic system architecture

In an array, the processor elements must be able to communicate data effectively and the total system has to be controlled in an adequate manner. The communication structure, the structure of the processing elements themselves and the control structure together form the architecture [11]. In parallel architectures the execution of a program is carried out in parallel. The overall execution time is usually shortened by a factor approximately dependent on the number of processors participating in the parallel processing. One of the most promising version is the PASM (Partitionable SIMD MIMD) parallel processing system.

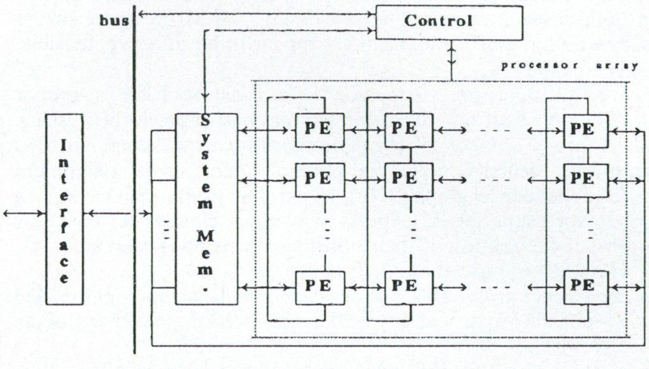


Fig. 3. The overall system of a mesh-connected systolic array

The PASM is a dynamically reconfigurable architecture designed to allow both SIMD (single instruction multiple data) and MIMD (multiple instruction multiple data) operation, and to provide the flexible computation and communications capability needed for the wide range of algorithms as found in image processing applications [12].

4. PARALLEL PROCESSING IN MULTIRESOLUTION ENCODERS

The hybrid coding structure is common for both the H.261 and the MPEG. The JPEG is involved in this structure. The main parts of a hybrid coder are the motion estimator, forward and inverse transform units, motion-compensated predictor, quantizer, run-length coding (Fig. 4 generic structure of hybrid encoder/decoder). The real-time operation of these units requires parallel processing in most cases.

It should also be mentioned that digital images are usually sampled on a grid and are stored on a 2D array. Therefore, they possess an inherent geometrical parallelism. This parallelism can be exploited by using a 2D array of processors, possibly one per pel. For practical reasons, the $N \times N$ size image is segmented in square $M \times M$ $(M < N)$ blocks or in strips and each block or strip is assigned to a specific processor array. The only problem to be encountered in such a solution is the blocking effect.

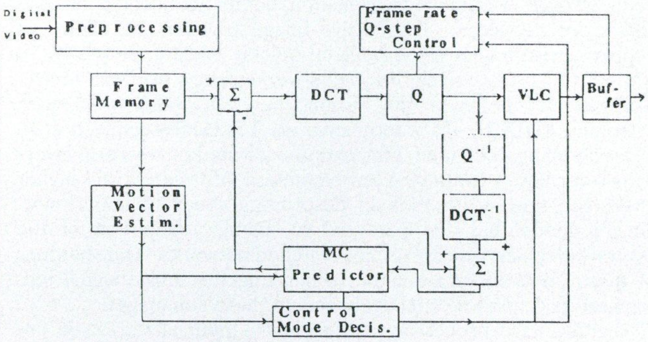


Fig. 4a. Generic block scheme of motion-compensated hybrid encoder

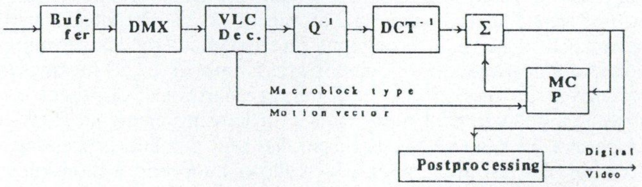


Fig. 4b. Generic block scheme of motion-compensated hybrid decoder

There are two ways for the implementation of hybrid codec:

- the overall implementation supposing massive parallel processing (MMP);
- the individual implementation of the several parts of the codec (DCT, IDCT, VLC, etc.) with parallel processing units containing smaller number of PEs.

In the next, the last one will be discussed presenting some typical examples, without the requirement of completeness.

One of the most popular techniques for motion estimation is the block-matching algorithm (BMA). Fig. 5 shows the block-matching process between two adjacent frames. BMA estimates the motion vector on a block-by-block basis. The reference block is compared with the corresponding blocks within a search area in the previous frame. The motion vector can be obtained when the best match candidate block is found.

The criterion of search is the MAD (mean absolute difference), which is given by

$$MAD(u,v)=\sum_{k=1}^{N}\sum_{l=1}^{N}\left|\,S(k+u,l+v)-R(k,l)\,\right|=AD_{uv}(k,l)$$

$$-p \leq k,l \leq p,$$

where $R(k,l)$ is he reference block and $S(k+u,l+v)$ is the candidate block within the search window in the previous frame. The u and v represent the components of motion vector. To speed up the BMA, the inherent spatial and temporal parallelism must be fully exploited. The criterion equation can be rewritten as

$$MAD(u,v) = \sum_{l=1}^{N} PS_{uv}^{l} = \sum_{k=1}^{N} \left[\sum_{l=1}^{N} AD_{uv}(k,l) \right].$$

The PS_{uv}^{l} represents the partial sum of the absolute differences for the jth column. Last equation implies that the computation of PS_{uv}^{m} and PS_{uv}^{n} for $m \neq n$, are mutually independent and can be performed simultaneously. Therefore the full search BMA can be operated in a parallel manner.

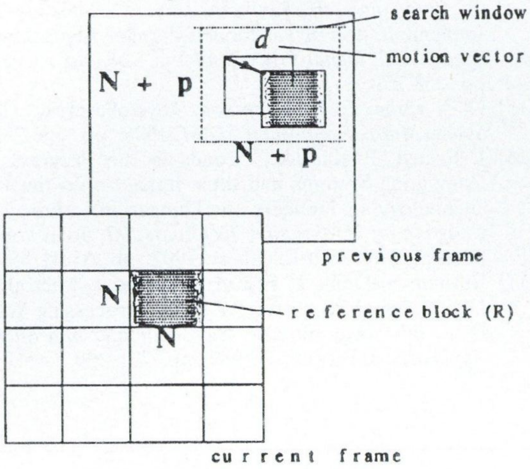


Fig. 5. Motion vector estimation with BMA

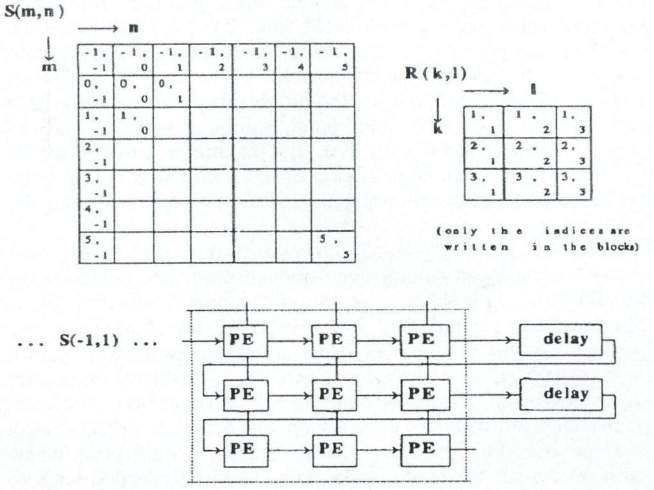
Let a simple example [13] be the following:

$$\{R(k,l) \mid 1 \leq k,l \leq 3\}, \quad p = 2,$$

the search area

$$\{S(m,n) \mid -1 \leq n,n \leq 5\}.$$

Fig. 6 shows the search area, the reference block and the simplified systolic array. The data of the search window are, from left to right, top to bottom, serially fed into the simplified array. In this case, the structure of PE is very simple, because it executes only the consecutive summation.



-1, -1	-1, 0	-1, 1	-1, 2	-1, 3	-1, 4	-1, 5
0, -1	0, 0	0, 1				
1, -1	1, 0					
2, -1						
3, -1						
4, -1						
5, -1						5, 5

1, 1	1, 2	1, 3
2, 1	2, 2	2, 3
3, 1	3, 2	3, 3

(only the indices are written in the blocks)

Fig. 6. The operation of a simplified BMA

In the case of a pel-recursive algorithm, one can determine the parallel architecture on similar manner. Because of the limited extent of this paper, here will be shown only some initial steps. For every pel at location (x,y,t), a real-valued motion vector $d_0(x,y,t)$ is predicted based on previously estimated motion vectors of certain neighboring pels. The actual motion vector $d(x,y,t)$ is obtained through an update scheme that computes the update $u(x,y,t) = d(x,y,t) - d_0(x,y,t)$, using a causal window around of data, around the pel in the current frame that is being processed. So, the basic formula is

$$\hat{d}(x,y,t) = d_0(x,y,t) + \hat{u}(x,y,t).$$

Fig. 7a shows the data dependency for a node, and on this base, one can form the dependency graph shown in Fig. 7b. Using the dependency graph, after some manipulation one can form the resulting SFG and finally the systolic array [13].

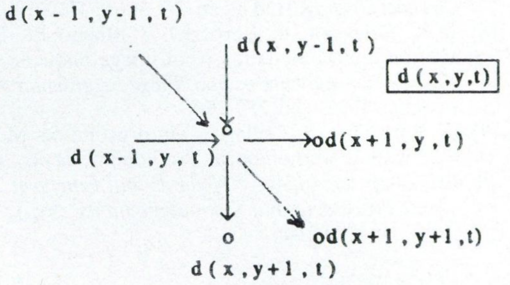


Fig. 7a.

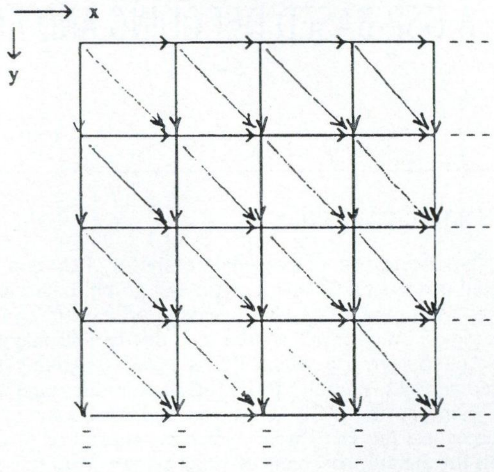


Fig. 7b.

Some further examples for mapping algorithm into arrays will be shown at the presentation on the workshop (e.g., the parallel DCT and IDCT, etc.).

5. CONCLUSION

The main purpose of our research work is the investigation of the several mapping methods for mapping parallel algorithms of hybrid encoding/decoding systems. The first results of this are shown here. This work is supported by COST-PECO (contract No. 12907) and will be executed in the framework of COST 229 project.

Future work is the simulation of the selected mapping methods and their evaluation taking into account the requirement of broadband communication.

REFERENCES

[1] H. Jeschke, K. Gaedke, P. Pirsch: "Multiprocessor Performance for Real-Time Processing of Video Coding Applications", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 2. No. 2. June, 1992. pp. 221-230.

[2] H. Yamauchi, Z. Tashiro, T. Minami, Y. Suzuki: *Architecture and Implementation of a Highly Parallel Single-Chip Video DSP*, ibid., pp. 207-220.

[3] C.-T. Chiu, K. J. Ray Liu: "Real-time Parallel and Fully Pipelined Two-Dimensional DCT Lattice Structures with Application to HDTV System", *IEEE Trans on Circuits and Systems for Video Technology*, Vol. 2. No. 1. March, 1992. pp. 25-36.

[4] J. H. Husoy, T. A. Ramstad: "Application of an Efficient Parallel IIR Filter Bank to Image Subband Coding", *Signal Processing*, Vol. 20. 1990. pp. 279-292.

[5] C. Hoek, R. Heiss, D. Müller: "An Array Processor Approach for Low Bit Rate Video Coding", *Signal Processing, Image Communication*, Vol. 1, 1989. pp. 213-223.

[6] K. Fazekas: "Parallel Processing Architecture for Subband Coding Using DSPs", *European Trans. on Telecommunications*, Vol. 3, No. 6. Nov-Dec. 1992. pp. 69-75.

[7] P. Csillag, L. Böröcky, K. Fazekas: "Comparative Study of Motion Estimation Algorithms for Subband Coding", *Proc. of COST 229 Workshop on Intelligent Terminals and Source and Channel Coding*, Budapest, 7-9 Sept., 1993. pp. 227-234.

[8] J. N. Driessen, L. Böröczky, J. Biemond: "Pel-Recursive Motion Field Estimation from Image Sequences", *Journal on Visual Communications and Image Representation*, Vol. 2, No. 3. Sept., 1991. pp. 259-280.

[9] L. Böröczky, P. Csillag: "Multiresolution Motion Estimation with Spatiotemporal Prediction", *Proc. of COST 229 Worhshop on Adatptive Methods and Emergent Techniques for Signal Processing and Communications*, Vigo, 16-18. June, 1993. pp. 290-294.

[10] I. Pitas: *Parallel Algorithms for Digital Image Processing and Computer Vision and Neural Networks*, John Wiley & Sons Ltd., 1993.

[11] A. Faruque, D. Y. Fong, D. W. Bray: "Parallel Processing Architecture for the Efficient Use of Memory in Image Processing Applications", *Optical Engineering*, July, 1991. Vol. 30, No. 7. pp. 994-1004.

[12] E. C. Bronson, T. L. Casavant, L. H. Jamieson: "Experimental Application-Driven Architecture Analysis of an SIMD/MIMD Parallel Processing System", *IEEE Int. Conf. on Parallel Processing*, 1989. pp. I-59 – I-67.

[13] E. D. Frimout, J. N. Driessen, E. F. Deprettere: "Parallel Architectures for a Pel-Recursive Motion Estimation Algorithm", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 2, No. 2. June, 1992. pp. 159-168.

[14] G. Carayannis, E. Koukoutsis, C. C. Halkias: "Hardware Implementation of Partitioned-Parallel Algorithms in Linear Prediction", *Signal Processing*, Vol. 24, No. 3. Sept., 1991. pp. 253-270.

[15] H. T. Kung, C. E. Leiserson: "Systolic arrays (for VLSI)", *Sparse Matrix Symposium SIAM*, 1978. pp. 256-282.

[16] I. Erenyi, P. Kacsuk: "Trends in the Progress of Parallel Computing Systems and their Impact onto the Information Technology in Hungary (in Hungarian)", *Proc. of the 5th Congress on Information Technology'92*, John von Neumann Society, Debrecen-Budapest, 1992. pp. 35-41.

[17] Burham al-Dein, I. Erényi, I. Szabó: "Microprogrammed Digital Signal Processor for Image Processing Workstation", *Proc. 6th Symposium on Microcomputer and Microprocessor Applications*, Budapest, 1989. pp. 791-799.

A DSP-BASED DECODING AND ENCODING OF COMPOSITE PAL/NTSC VIDEO SIGNALS

T. S. IBIYEMI*

DEPT. OF ELECTRICAL ENGINEERING,
UNIVERSITY OF ILORIN, ILORIN, NIGERIA

1. INTRODUCTION

The recent use of personal computer, PC, as a platform for digital processing of video, audio and graphics succinctly referred to as Multimedia has also necessitated careful consideration of the video/ audio input source in order to still maintain the low-cost philosophy on which PC is based. Among the available video sources, namely, RGB, off-the-air television video signals, TV camera, and VCR the cheap and commonly available video sources are the last three. However, there are two major problems for digital processing of video signals from these three video sources. Firstly, the video incompatibility among the three world broadcast video standards of NTSC, SECAM, PAL. Secondly, the composite nature of the video signals and the different implementation of frequency interleaving of luminance and chrominance signals within the bandwidth meant for luminance only. Hence, there is a problem of universal sampling frequency common to the three, compounded by the fact that SECAM uses two different subcarrier frequency modulated colour signals instead of quadratured subcarrier frequency modulated colour signals as utilized by NTSC and PAL. Towards finding common sampling frequency for the three standards, the CCIR/SMPTE recommend sampling of video signal in luminance, Y, and two colour difference $B - Y$, $R - Y$ component forms at basic rate of 13.5 MHz. [1], [2].

However, our interest in this paper is digital decoding and encoding of samples obtained directly from sampling of composite video signals. Hence, we only have to consider PAL and NTSC composite signals because of their similarity in colour modulation which is amenable to direct sampling. In view of the diverse applications of digital video, two sampling rates are considered. Sampling rate at 13.5 MHz for applications requiring CCIR/SMPTE component video; and sampling rate at four times the subcarrier frequency, $4f_{sc}$, of the composite video. Both have their advantages and disadvantages. The sampling of composite video at 13.5 MHz provides a common sampling frequency for both NTSC and PAL, few samples than sampling at $4f_{sc}$, orthogonal sample structure which facilitate ease of comb filtering implementation for separating luminance and chrominance data, but has the disadvantage of complex colour demodulation. On the other hand, sampling at $4f_{sc}$ leads to huge samples especially for PAL but facilitates ease of colour demodulation and implementation of filter with slow falling slope because of the large spectral gap between the baseband and the sidebands.

With the choice of sampling frequency now settled, the next task is decoding into luminance and chrominance components, and then the demodulating of the chrominance into two colour components. Decoding of PAL or NTSC into luminance and chrominance components could lead to two major artifacts known as cross-colour, and cross-luminance as a result of imperfect separation of interleaved luminance and chrominance. The level of the incurred artifacts depends on the activities in the image, and the decoding process. These artifacts could pose major problem if such decoded components would be encoded back to composite video signal as is the case in this paper [3], [5].

Considering the three dimensional, 3D, spectrum of digital NTSC and PAL composite videos, the frequency interleaving of luminance at integer multiple of line rate with chrominance at odd multiple of half-line (NTSC)/quarter-line (PAL) frequency is exploited in the decoding in order to reduce the effect of the artifacts. Both 2D and 3D filtering were experimented with,

* The author is spending his sabbatical leave at CSELT, TORINO, ITALY

for the purpose of performance evaluation and it was found out that 2D filter decoding, if carefully crafted, suffices for good quality and low-cost. The decoding and encoding algorithms are implemented around Analogue Device ADSP21O1 16 bit fixed point DSP. Experimentation has proved that it is viable to realize a cost-effective decoding and encoding of PAL and NTSC composite signals in real-time using fixed l6 bit DSP. This decoder can serve as a front-end, and the encoder as a back-end in a digital video processing environment such as a PC.

2. FREQUENCY SPECTRUM OF NTSC & PAL

Consider the composite NTSC and PAL signal equations:

$$S(t)_{ntsc} = Y(t) + Q(t)\sin(2n\pi f_{sc}t) + I(t)\cos(2n\pi f_{sc}t), \quad (1)$$

where

$$\begin{aligned} f_{sc} &= (455/2)f_h, \quad f_h = 15.7346 \text{ kHz} \\ &= (455/2)(525/2)f_v, \quad f_v = 59.94 \text{ kHz} \end{aligned}$$

$$S(t)_{pal} = Y(t) + U(t)\sin(2n\pi f_{sc}t) \pm V(t)\cos(2n\pi f_{sc}t), \quad (2)$$

where

$$\begin{aligned} f_{sc} &= (1135/4)f_h + 1/2f_v, \quad f_h = 15.625 \text{ kHz} \\ &= (455/2)(625/2)f_v + 1/2f_v, \quad f_v = 59.94 \text{ kHz} \end{aligned}$$

Sampling Eq. (1) at a rate f_s gives:

$$\begin{aligned} S(n/f_s)_{ntsc} &= Y(n/f_s) + Q(n/f_s)\sin(2n\pi f_{sc}/f_s) + \\ &\quad + I(n/f_s)\cos(2n\pi f_{sc}/f_s) \\ S(n/f_s)_{pal} &= Y(n/f_s) + U(n/f_s)\sin(2n\pi f_{sc}/f_s) \pm \\ &\quad \pm V(n/f_s)\cos(2n\pi f_{sc}/f_s) \end{aligned} \quad (3)$$

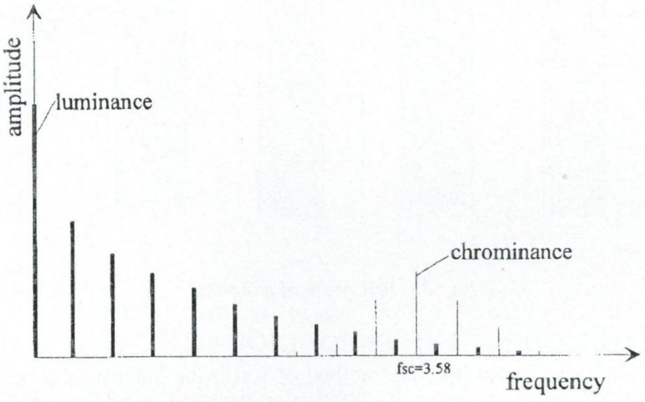


Fig. 1(a). NTSC horizontal frequency spectrum

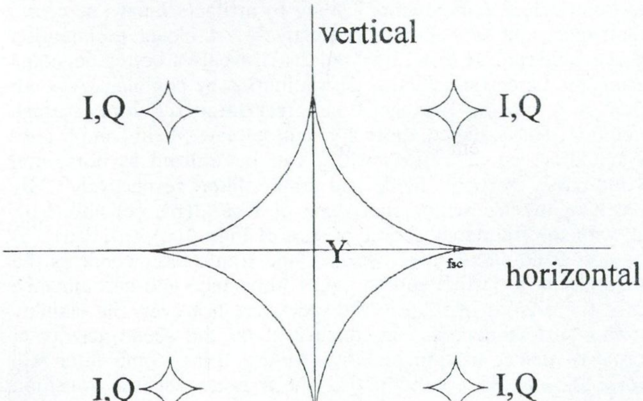


Fig. 1(b). NTSC vertical/horizontal frequency spectrum

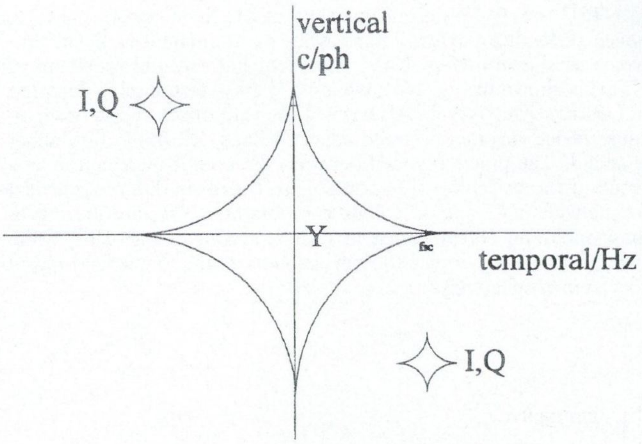


Fig. 1(c). NTSC vertical/temporal frequency spectrum

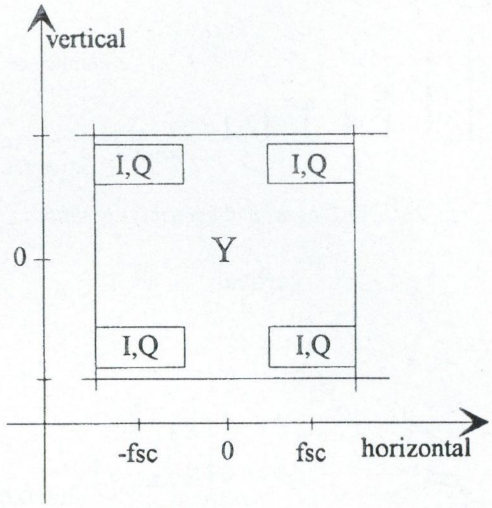


Fig. 1(d). NTSC vertical/horizontal

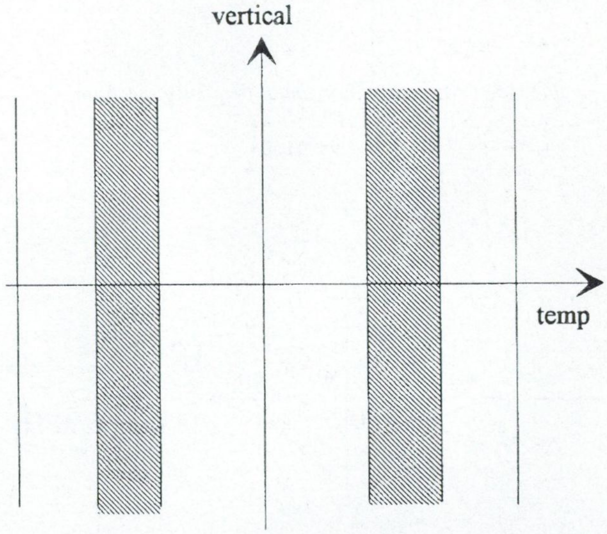


Fig. 1(e). NTSC vertical/temporal

A TV video exists in analogue form only in horizontal space, since the effect of scanning results in vertical and temporal sampling. This causes peaks at line rate with side lobes at frame-rate in its frequency spectrum. The broadness of the side lobes depends on the vertical details of the image. Colour subcarrier frequencies are chosen to exploit the empty space in between.

In NTSC the f_{sc} is an odd multiple of 1/2 line-, and frame-rate, hence, effecting perfect interleaving of luminance with chrominance as shown in Fig. 1(a). The vertical/horizontal spectrum of NTSC is shown in Fig. 1(b), while its vertical/temporal is depicted in Fig 1(c). The f_{sc} of PAL has 1/4 line rate offset and in order to cause phase reversal between adjacent lines, 1/2 frame-rate offset is added. The phase reversal between the colour information as a result of the switching of V component results in different vertical frequencies in U and V colour components. The interleaving of luminance and chrominance in PAL is shown in Fig. 2(a), while Fig. 2(b) and (c) depict the vertical/horizontal, vertical/temporal spectrum respectively.

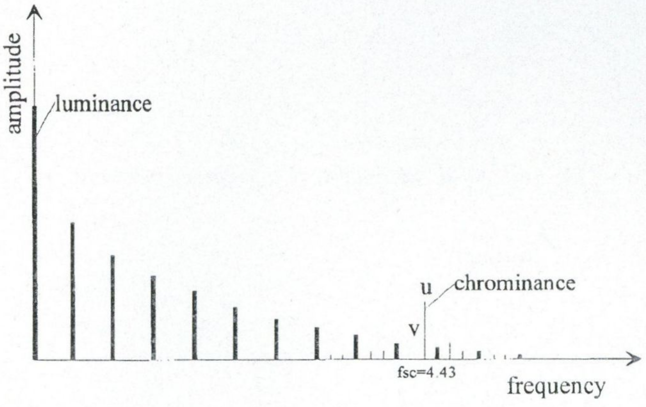


Fig. 2(a). PAL horizontal frequency spectrum

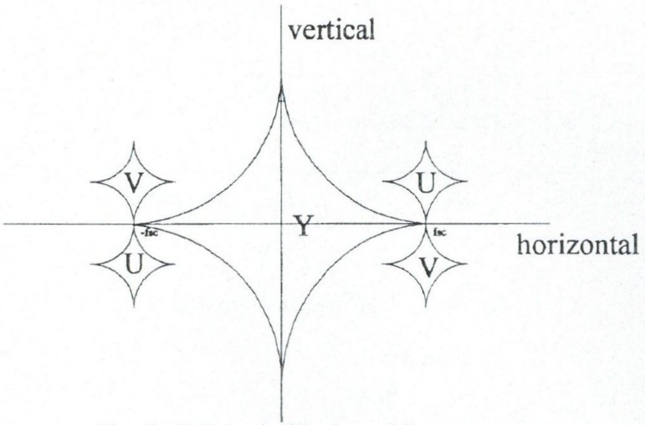


Fig. 2b. PAL vertical/horizontal frequency spectrum

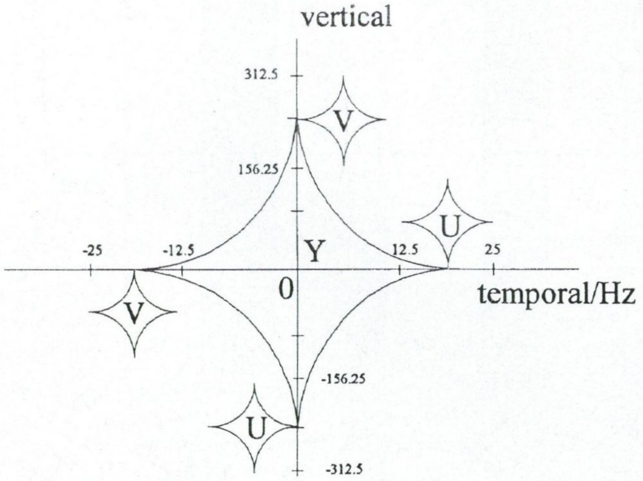


Fig. 2c. PAL vertical/temporal frequency spectrum

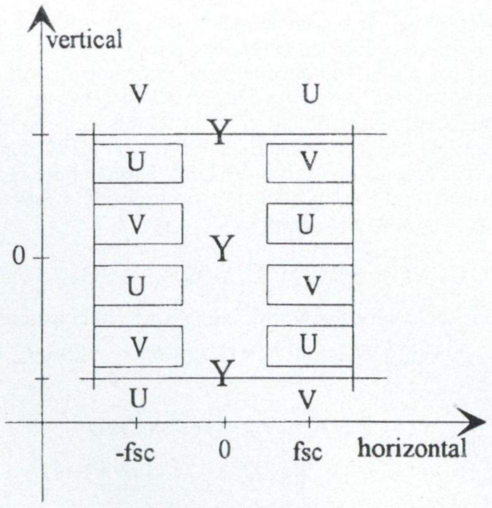


Fig. 2d. PAL vertical/horizontal

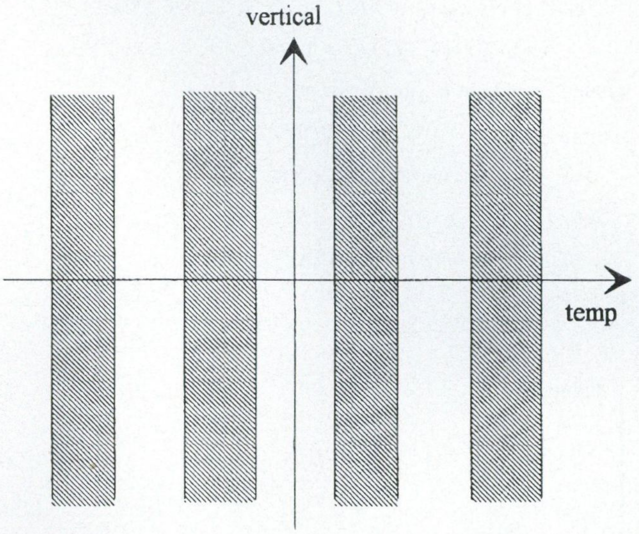


Fig. 2e. PAL vertical/temporal

3. DECODING AND ENCODING

In conventional TV, one method of retrieving luminance is by notch filtering rejection of all video information about the f_{sc}, and recovering of chrominance by a bandpass filter centred on the f_{sc}. But this block rejection decoding encompasses residual colour in luminance component and traps some luminance in the decoded colour components leading to artifacts known as cross-luminance, and cross-colour respectively. A closer examination of the spectrum of Figs. 1(a) and 2(a) reveals a better decoding technique based on selective discrimination by combing. A comb filter is a moving average transversal filter requiring storage memory (delay), hence, there are sample-, line-, field-, and frame-comb filters configurations which can be utilized as horizontal (band-pass), vertical, field, and frame filter respectively. By matching the frequency spectrums of Figs. 1(b), (c) and 2(b), (c) with the frequency characteristics of Figs. 1(d), (e), 2(d), (e) for post band-pass-filtered vertical, and frame filter confirms the feasibility of a perfect separation of luminance and chrominance since there is no overlap in the spectrum, however, the viability of this perfect decoding is conditional on the scene activity in terms of object motion and transitions. Line comb filter will work fine on the condition that there is no vertical transition over the corresponding sample points over the lines used; while frame comb filter gives satisfactory separation for still or slow moving picture but fails in rapid motion. Demodulation of the chrominance into two colour components, I, Q for NTSC; and U, V for PAL, involves multiply the chrominance samples by cosine/sine function as in Eq. 4, and then applying lowpass filter.

$$I_n, V_n = Cn\cos(2n\pi f_{sc}/f_s), \quad f_s = 4f_{sc}, \text{ or } 13.5 \text{ MHz}$$
$$Q_n, U_n = Cn\sin(2n\pi f_{sc}/f_s) \qquad (4)$$

At $f_s = 4f_{sc}$, the above equation involves only multiplication by 1,0, or -1 but for $f_s = 13.5$ MHz the operation is not as simple as that.

The process of encoding or recoding luminance and the two colour components into composite PAL or NTSC involves lowpass filtering to the stipulated bandwidth for that component, and to forestall cross effect artifacts, the bandlimited chrominance is comb filtered vertically.

4. DSP IMPLEMENTATION

The decoding and encoding processes are simulated on ADSP2101. The line comb filter algorithm is based on Eqs. (5) and (6); bandpass filter algorithm is shown in Eqs. 7 and 8. The frame filter algorithm is the same as the line algorithm except that a frame delay buffer is used instead of line buffer.

$$C_n = 1/2[L_{n-1} - 1/2(L_n + L_{n-2}]^* H_{bpf}, \text{ ntsc } 2 - \text{line}$$
$$Y_n = L_n - C_n \qquad (5)$$
$$C_n = 1/2[L_{n-2} - 1/2(L_n + L_{n-4}]^* H_{bpf}, \text{ pal } 4 - \text{line}$$
$$Y_n = L_n - C_n \qquad (6)$$
$$C_n = 1/2[S_{n-1} - 1/2(S_n + S_{n-2})], \text{ ntsc} - H_{bpf} \text{ bandpass filter} \qquad (7)$$
$$C_n = 1/2[S_{n-2} - 1/2(S_n + S_{n-4})], \text{ pal} - H_{bpf} \text{ bandpass filter} \qquad (8)$$

In view of the low cost, but nevertheless, effective philosophy of the project, the line comb filtering is considered optimum rather than the frame combing or the novel technique of circular filter [2], [4] which exploits the vertical/temporal frequency. However, an adaptive measure is incorporated in our algorithm to switch to horizontal filtering when there is a significant vertical transition.

5. CONCLUSIONS

The decoding and encoding of PAL and NTSC composite signals based on l6 bit fixed DSP in real-time was investigated. Experimentation has validated its possibility. An adaptive line comb filter was proved sufficient for maintaining low-cost philosophy without unduly sacrifying quality in terms of cross-effects artifacts. In the encoding process, vertical comb filtering was applied to the bandlimited chrominance in order to forestall subsequent cross effects. This system could serve as a front-end and back-end in a digital video processing environment.

6. ACKNOWLEDGEMENT

I am grateful, particularly to CSELT, TORINO ITALY for giving an opportunity to spend my sabbatical leave with them and to place at my disposal the equipment used for this project.

REFERENCES

[1] J. F. Blinn: "The world of digital video", *IEEE Computer Graphics & Applications*, pp. 106-112, Sept. 1992.

[2] J. Watkinson: The art of digital video, 2nd Ed., Focal Press, 1994.

[3] M. Tsinberg et al: "A system for artifact free NTSC encoding and decoding", *IEEE. Consumer Electronics*, Vol. CE-32, No. 5, pp. 228-236, August 1986.

[4] M. Weston: " Advanced colour decoding techniques ", *Proc. of l8th* ITS, Montreux, pp. 108-113.

[5] C. K. P. Clarke: High quality decoding for PAL inputs to digital YUV studios, Report No. BBC RD 1982/12, Research Dept., British Broadcasting Corporation, July 1982.

A REAL-TIME IMAGE-PROCESSING SYSTEM FOR VISUAL INSPECTION OF REAL ENVIRONMENTS

G. L. FORESTI, and V. MURINO

SIGNAL PROCESSING AND UNDERSTANDING GROUP
DEPARTMENT OF BIOPHYSICAL AND ELECTRONIC ENGINEERING (DIBE)
UNIVERSITY OF GENOA
I-16145 GENOVA, ITALY

1. INTRODUCTION

Visual inspection of real environment (e.g., airports, railway stations, underground stations, etc.) is a basic task of many surveillance systems [1], [2]. Traditionally, the most important works of surveillance and monitoring safety have been dependent on human visual observation. However, a system able to detect dangerous situations can be of help to an operator, even if the replacement of human surveillance is not meant. This paper describes the application of an image processing system to monitor the area of a railway level-crossing. The objective is to develop a surveillance system prototype for unattended level-crossings, aimed at giving a real-time alarm in dangerous situations [3]. The surveillance of unattended level-crossings, which can be often under the remote visual surveillance by an operator, is particularly important in the field of railway transport safety. In normal conditions, the presence of an operator is needed before and after the movement of the gate, especially when it is closed. The system described in this paper is inserted within the *Section III* of the framework *Progetto Finalizzato Trasporti II* of the National Research Council (CNR) of Italy.
The basic tasks to be performed by the system are:

- image acquisition;
- object detection;
- object localization.

Some words are needed to explain the meaning of the term "real-time" in the proposed application. The system must be able to send a danger signal to allow the level-crossing operator to stop a train, to raise the gate, or to make any other decision to face the situation. If one or more objects are present in the area of interest for a time exceeding a fixed time-out, the danger alarm must be activated.

Up to now, the tasks of monitoring safety and inspecting rails have depended on human visual observation [1], [2). It is however desirable that in the case of simple inspection and monitoring such tasks be executed by a machine vision system whereas, in the case of complex monitoring, decisions should be made by an human operator with the help of an advanced machine vision system. Many are the examples of monitoring facilities that are object of different applications in the railway field: inspection of wheel tread profile, of rail gap, of wiring abrasion, of sprinkler for snow melting and so on [2]. These applications of image processing in railway field have a static target: the system has to make a decision after checking the measures of an object made up of curved surfaces like a rail or a wheel. Every day the system must perform on-line inspections with high frequency, and manage statistical maintenance. This type of monitoring is not performed by a safety system like the proposed one, whose requirements are determined by the need to stop a train in an emergency. Similar problems can be encountered in other applications, like monitoring a platform for safety purposes (2), monitoring a train's front view [1], monitoring the crowd at a terminal or snow on tracks [4], which are under study in other research laboratories.

A fall detection system has been realized in France (Lyon) where a fully automatic metro line without onboard driver needs to secure safety of passengers standing on the platform versus the incoming train [3]. A crowd estimation system is currently being studied for a station of the Genoa metro rail system [4].

2. SYSTEM ARCHITECTURE

In this section, the architecture of the system (Fig. 1) is described and the reasons for the choices made are given. Image sequences of the surveilled scene are acquired by a CCD-TR105E Sony camera and digitized into 256x256 pixels by using the Imaging Technology ITEX (series 151) device.

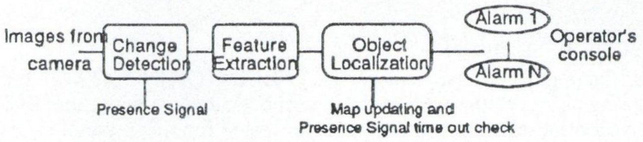


Fig. 1. Functional organization of the described modules

2.1. Object detection

The first step of the system consists into detect stationary or moving objects inside the area of interest of the level crossing. This task is performed by means of a new very fast change detection procedure able to identify on the image plane some areas characterized by significant modifications respect to a reference image (i.e., background image).

This procedure has been developed directly on the ITEX device that can be programmed in C language under UNIX using a Sun Workstation as host (Fig. 2). Real-time image acquisition and change detection operations can be obtained.

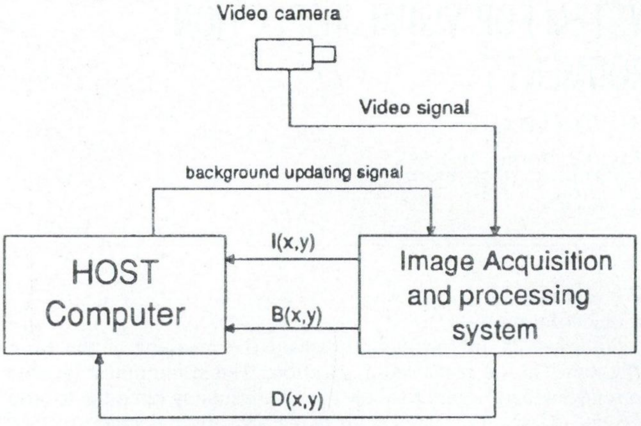

Video camera
Video signal
background updating signal
I(x,y)
B(x,y)
D(x,y)
HOST Computer
Image Acquisition and processing system

Fig. 2. The image acquisition module

A simple difference method with hysteresis [5] computes the absolute difference

$$D(x,y) = |I(x,y) - B(x,y)| \quad \forall (x,y) E \in NxN \qquad (1)$$

between the pixel intensities of the input image $I(x,y)$ and the background image $B(x,y)$, where N are the image dimensions. Two thresholds THR_{in}, THR_{out} and a state variable are used to establish if a point (x,y) is a background point or an object point. The output is represented by a binary image $D(x,y)$ whose pixels belonging to the changed parts are setted to high gray-level [6], [7]. The background image $B(x,y)$ is updated any time the illumination conditions of the scene are significantly changed.

The function used to decide the label of a point is a hysteresis function, as shown in Fig. 3. If the state is 'Background' and $D(x,y) > \mathrm{THR}_{in}$, the point (x,y) is marked as an 'object' point and the state is switched to 'Object'. Analogously, if the state is 'Object' and $D(x,y) < \mathrm{THR}_{out}$, the point (x,y) is marked as a background point and the state is switched to 'Background'.

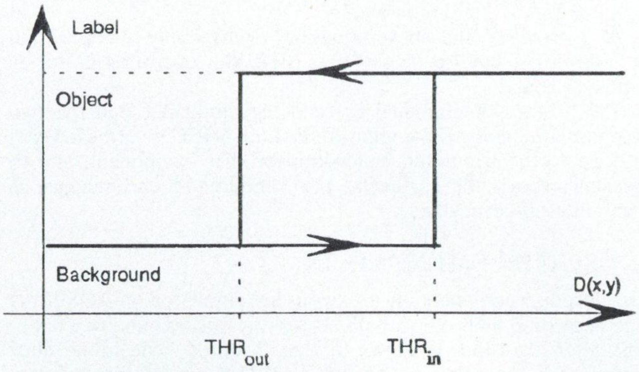


Fig. 3. Graphic representation of the hysteresis function used

The decision on how to mark a point is driven not only by the gray-level value of the point but also by those of neighbouring points. At this level of the processing chain, a presence signal is available, which is switched on if a significant image part has changed. Then, a focus of attention procedure is performed to reduce the areas to be processed, thus speeding up the whole system functioning [5]. The *minimum bounded rectangles (MBR)*, related to the detected changed areas on the image plane, are extracted and sent to the object localization module.

2.2. Object localization

The object localization module uses the position (c_x, c_y) of the left upper corner and the dimensions d_x and d_y of each MBR on the image plane to determine where the 3D object is placed. Even if an object can be seen at the same time in any zone of the interesting area of the level crossing by two or more cameras, stereo localization strategies are not used [8] in order to satisfy the real-time constraint of the application. The solution adopted is that to process separately the information coming from one camera at each time instant. The transformation of an image plane point into a 3D reference system is a well known ill-posed problem [9]. If the intrinsic parameters and the calibration matrix M of the used camera are known, only one constraint is necessary to regularize this problem. The following calibration matrix is used in the present application:

$$M = \begin{bmatrix} -0.019464 & 0.036133 & -0.279027 & 71.980393 \\ 0.158676 & -0.278248 & -0.170096 & 98.631627 \\ -0.000818 & -0.001059 & -0.000689 & 1.0 \end{bmatrix} \qquad (2)$$

A ground plane hypothesis [10] is assumed to regularize the localization problem: all objects (e.g., cars, lorries, etc.) moving toward the level-crossing are considered placed on the ground plane (i.e.,$Z = 0$).To this end, the image points belonging to the bottom line of the MBR correspond to 3D ground plane points. In particular, the point $P = (c_x, c_y - d_y/2)$ is used as reference point to localize the object. By considering the classical perspective equations and the calibration matrix, it is possible to write:

$$(m_1 - x_p \cdot m_3) \cdot X = 0 \qquad (3a)$$

$$(m_2 - y_p \cdot m_3) \cdot X = 0, \qquad (3b)$$

where m_i is the i-th row vector of the calibration matrix M, (x_p, y_p) are the coordinate of the point P on the image plane, and $X = [X_W, Y_W, Z_W, 1]$ is the 3D point to be determined. Eq. (3a) or (3b) represents a plane, while the system composed by both Eqs. (3a) and (3b) represents a 3D line in the 3D reference system. Such a 3D line is passing through both the pixel coordinates (u_i, v_i) and the origin of the 3D reference system (i.e., the system allows a solution for $s = 0$). Finally, by imposing the ground-plane hypothesis ($Z_w = 0$) and by eliminating for substitution a variable from Eqs. (3a) and (3b), it is possible to determine the component of the vector X which represent the 3D position of the image point P [9].

When an object was localized inside the interesting area, a check for the time-out is performed to decide if activate or not a danger alarm.

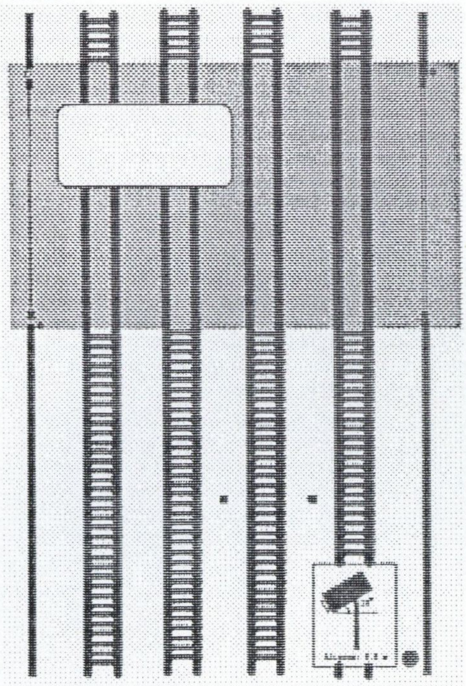


Fig. 4. Map of a level-crossing to visualize object positions

A 2D top-view map (the plane $X - Y$) of the level crossing is used to represent the object position and its approximated dimensions (Fig. 4). Such a map, visualized on the console screen, shows a rectangular box whose position and state (blinking and red for a danger alarm) are continuously updated.

3. RESULTS

In this section, the more significant results obtained by the proposed surveillance system are presented. The whole system has been installed at the level-crossing of Rivarolo located in the vicinity of Genoa. The processed images were chosen from real word sequences acquired with a CCD-TR105E Sony camera. Fig. 5 shows two time-contiguous photograms acquired and processed by the system, while Fig. 6 shows the background image.

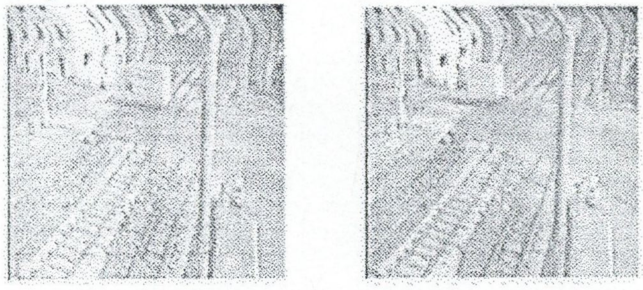


Fig. 5. Two time-contiguous photograms of the level-crossing acquired and processed by the system

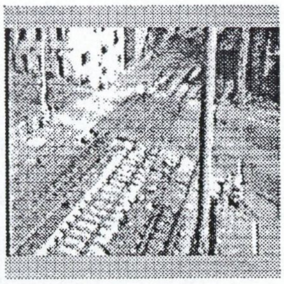


Fig. 6. The background image

Fig. 7 shows the final result of the change detection process. It is represented by a binary image obtained directly on the ITEX device by means of the simple difference with hysteresis technique. At the present, the system processes two images per second.

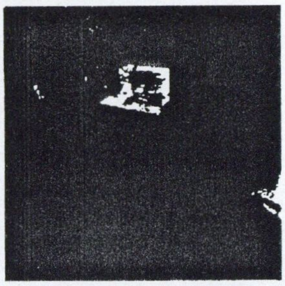


Fig. 7. Result obtained by the change detection module

Figs. 8a and 8b shows the results obtained in the case of very noisy input images. Such a result confirms the robustness of this method to noise. In these images it is also possible to see the results of the focus of attention module: each area containing groups of changed pixels is selected and identified by means of the minimum bounded rectangle (MBR). Then, each MBR is processed by the object localization module to determine the object position on the 2D top-view map (see Fig. 4).

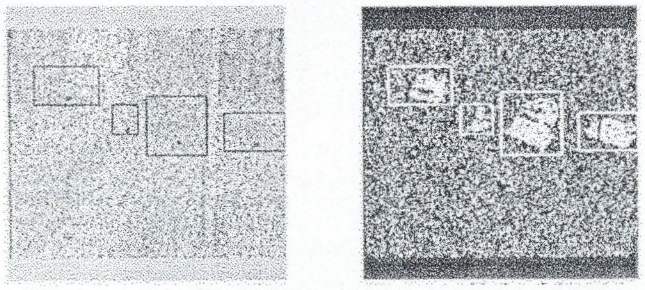


Fig. 8. (a) Original gray-level image with 42% noisy points added and (b) result of the hysteresis based change detection method

4. ACKNOWLEDGEMENTS

This paper has been partially supported by the *PROGETTO FINALIZZATO TRASPORTI II* (Italian Research Council) and by the *CEC ESPRIT*-project 6068 ATHENA. The authors wish to thank Mr. Federico Nebiacolombo for his valuable assistance during the implementation phase.

REFERENCES

[1] H. Susama, M. Ukay: "Application of Image Processing for Railways", *Q. Rep. of Railway Tech. Res. Inst. (RTRI),* Vol. 30, No. 2, pp. 74-81, May 1989.

[2] J.-C. Brachet: "Fall detection system in a metroline", Semaly Villeurbanne, France, 1990.

[3] Ente ferrovie dello Stato, "Istruzione per l'esercizio dei passaggi a livello", *Edizione*, 1989.

[4] G. L. Foresti, V. Murino, C. S. Regazzoni: "A real-time vision system for crowding monitoring", *19th International Conference of IEEE Industrial Electronics Society – IECON93*, Hyatt Regency-Maui, Lahana, Maui, Hawaii, November 15-19, 1993, (in press).

[5] G. Foresti, C. S. Regazzoni, G. Vernazza: "Sistema di Visione per la Sorveglianza di Passaggi a Livello Ferroviari non Presenziati", *First National Conference on Progetto Finalizzato Trasporti II*, Centro Nazionale delle Ricerche, Rome, Italy, October 19-21,1993, pp. 1995-2012.

[6] K. Skifstad, R. Jain: "Illumination independent change detection for real world sequences", *Computer Vision, Graphics, and Image Processing*, Vol. 46, pp. 387-399, 1989.

[7] A. Benveniste, M. Basseville, G. V. Moustakides: "The asymptotic local approach to change detection and model validation", *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 7, July 1987.

[8] X. Zhuangh, R. M. Haralick: "Two view motion analysis theory and algorithm", in *Proc. of International Conference on Acoustic, Speech and Signal Processing*, March 1985.

[9] D. D. Shew: "A generalized method for 3D object location from single 2D images", *Pattern recognition*, Vol. 25 No. 8, pp. 771-786, 1992.

[10] T. N. Tan, G. D. Sullivan, K. D. Baker: "Structure from motion using the ground plane constraint", in *Proc. of II. European Conference on Computer Vision*, Santa Margherita Ligure, Genova, Italy, 1992.

REMOTE SENSING

COMPUTER PROCESSING OF HARD-COPY MAPS AND THEIR APPLICATION IN SATELLITE POSITIONING

D. TORKAR, R. MURN, and A. DEDIĆ

JOZEF STEFAN INSTITUTE,
JAMOVA 39, 61111 LJUBLJANA, SLOVENIA

1. INTRODUCTION

The NAVSTAR/GPS is actually the most accurate positioning service compared to traditional radio navigation systems. It provides satisfactory accuracy for many civil applications. The accuracy of the Standard Positioning Service (C/A code), that is available to civil users, is under degradation of Selective Availability (SA) which contribution is the greatest among GPS errors (ionosphere, troposphere, satellite clock, ephemeris, multipath, receiver's noise, etc.). SA can be bypassed with the use of Differential GPS (DGPS) [1]. The information provided to the user by a GPS receiver contains many data that tell only a little to an inexperienced user. Therefore these data must be user-friendly represented and clearly visualized to be useful to anyone. Probably the most persuading way of GPS data visualization is visualization on computerized maps. To gain good results, maps must be prepared with special care [2]. Because the majority of maps exist as yet only as hard-copies, they must be first input into computer through the scanning device. The scanning process depends on the type of a certain map. Sometimes scanning errors have a great influence on the quality and geometry of digital maps. These errors may be reduced or neglected with some image processing techniques. These techniques are specific to a map type. On colour images some colour reduction and edge enhancement filters can be used to clear the picture and to reduce the noise. On gray-scale images some other filters should be used [6]. If black-and-white maps are concerned appropriate thresholding technique should be chosen to obtain the best result. Another significant error caused by the scanning device is a geometric distortion. This distortion is also a consequence of paper contraction and extension. Some mathematical transformation can be applied to compensate such a distortion. After image processing digital maps should be stored in a way to occupy as little space as possible and at the same time the decompression time should not exceed reasonable limits. The JPEG algorithm was found to have been very convenient, because of its ability to change the ratio between a compression factor and a restoration quality. Depending on what geographic or geodetic projection the hard-copy map was drawn in, the digitized map must be adjoined into the appropriate co-ordinate system in which the incoming GPS data must be transformed too. Maps prepared in such a way can be a good base for a CAD program that is used to visualize GPS data and can offer a wide range of applications [2].

2. MAP DIGITALIZATION

There exist many different types of maps. In general they can be classified by different criteria. One can be measuring scale (1:50000, 1:5000, etc.), the other can be purpose (tourist, geodetic, ...). From the digitalization and computerization point of view, maps can be divided by the technique they were drawn:

- black-and-white maps (cadastral maps, land register maps, simple line maps usually sparse);
- gray-level maps (geodetic, military and tourist maps, usually very dense with a lot of information);
- colour maps (geographic (atlas) or tourist maps with a lot of information).

The first decision that should be made when hard-copy maps are digitized is how they should appear in a computer. When simple black-and-white maps are concerned there is usually only one possibility. It should appear as a black-and- white digital map. Usually it does not make sense to produce coloured or gray-level map out of black-and-white one, even if it is quite possible to achieve. The same applies to colour and gray-level images. The informational contents should only stay the same or be reduced. If this is taken under consideration six possible transformations can be done:

hard-copy map		**digital map**
black-and-white		black-and-white
gray-level		black-and-white
gray-level	⟹	gray-level
colour		black-and-white
colour		gray-level
colour		colour

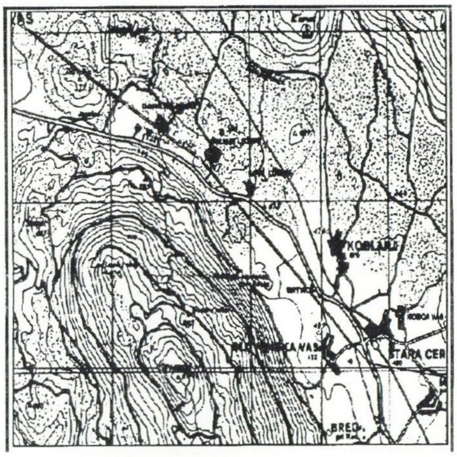


Fig. 1. Example of a gray-level tourist map (measuring scale 1:50000)

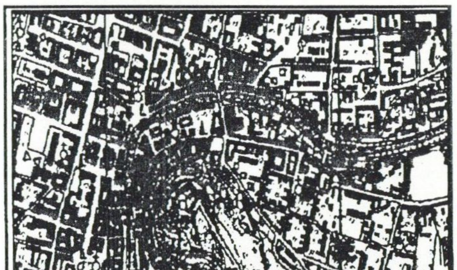


Fig. 2. Colour city map (measuring scale 1:50000)

There are two possible ways to achieve particular transformation. It can be done by

- scanning mode
- image processing

The first depends almost only on scanning device being used. After digitalization some image processing can also be done to improve image quality and to reduce noise. This way images are scanned in the mode as the final image should look like. If the hard-copy map is gray-level or colour and the result should be a binary (black-and-white) image, the scanning mode should be black-and-white. The binarization is done by a scanner and the user has only a small influence on quality.

The second way offers much more control over the process flow. Scanning is done in the same or one degree higher mode regarding informational rate as the final image should be. This results in these relations:

hard-copy map	**bitmap image**	**scanning mode**
black-and-white	black-and-white	gray-level
gray-level	black-and-white	gray-level
gray-level	gray-level	gray-level
colour	black-and-white	colour
colour	gray-level	colour
colour	colour	colour

The most important step here is pre-processing of scanned images. The user can choose various image processing techniques to gain the result that satisfies his demands [4].

In our case two particular cases from above were taken into investigation. The first was preparation of cadastral maps for a CAD program that can semi-automatically do raster-to-vector conversion [3] and can visualize GPS data in real time. The second was preparation of colour geographic maps for some geographic information system (GIS) applications, including GPS tracking.

3. CADASTRAL MAPS

Cadastral maps are black-and-white maps, containing usually only polygons and some text. The result of map pre-processing should be low-noise one bit raster image, that represents the original as accurately as possible. The distribution of black signal pixels and black noisy pixels on hard copy is very spatial dependant. The experiment with black-and-white scanning mode showed that the scanner made bad thresholding and the result was noisy bitmap image, which would be hardly improved by additional image processing. A good binary image was needed because of an easier implementation of semi-automatic raster-to-vector conversion algorithm. Therefore the maps were scanned with a dynamic range of 256 gray-levels and a resolution of 200 dpi. After that each pixel was classified in one of two classes using dynamic thresholding, which gave better results as local and much better results as global thresholding. Some methods of dynamic threshold were tested [5], resulting in a modified algorithm for dynamic threshold determination. The results were tested with calculating two coefficients: the shape measure and the uniformity.

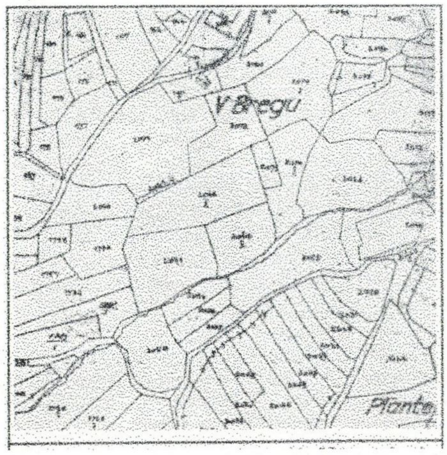


Fig. 3. The cadaster map scanned in 256 gray-levels

The threshold in every pixel of the gray-level image was determined using the formula:

$$T(x,y) = C(x,y)\sigma_{x,y} + \mu_{x,y},$$

where $C(x,y)$ is the so-called *threshold coefficient*, defined as $C(x,y) = \frac{kH_n}{N}$.

H_n is the number of pixels in the whole image having the same gray-level as the pixel itself; N is the number of all pixels in the image;

k is determined experimentally and lies between 0.65 and 1;

$\sigma_{x,y}$ is the standard deviation of the pixel (x,y) and eight adjacent pixels;

$\mu_{x,y}$ is the mean value of nine pixels.

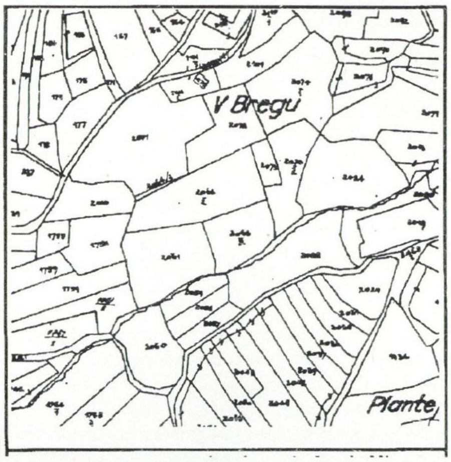


Fig. 4. Binary image after dynamic thresholding

4. COLOUR MAPS

Colour maps were digitized to be used in a GIS application, permitting real-time GPS positioning. Because of the amount of data and the nature of application, gray-level digital maps were chosen as satisfying. Scanning was made in a colour mode, using 256 colours.

Then colour was removed and a despecle filter was applied [4]. Resulting gray-level image was then processed by a sharpen filter twice. This results in a clear 8 bit gray-level image, suitable for zooming and unzooming with all text clearly visible and readable.

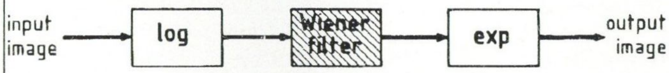


Fig. 5. Block diagram of a despecle filter

5. GEOMETRY CORRECTION

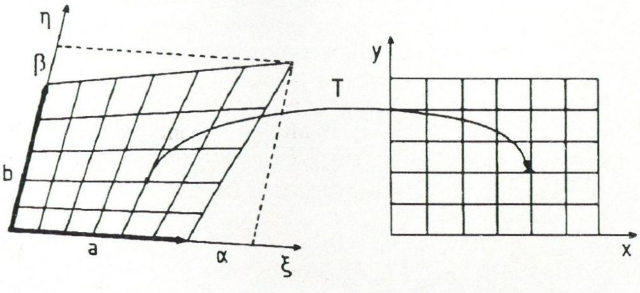


Fig. 6. Transforming of a general quadrangular to a regular rectangle

The scanner and paper errors result in a geometric distortion of a rectangle to a general quadrangular. This was compensated on a digital image using special geometric transformation that needs only four map's corner points determining general quadrangular as an input and four points of a resulting rectangle.

The transformation T from coordinate system (ξ, η), into coordinate system (x, y) is expressed:

$$x = \frac{1}{2\beta}(-1 + \beta\xi - \alpha\eta + |\sqrt{D}|)$$

$$y = \frac{1}{2\alpha}(-1 - \beta\xi + \alpha\eta + |\sqrt{D}|),$$

where D is a discriminant:

$$D = (1 + \beta x + \alpha y)^2.$$

6. IMAGE COMPRESSION

Digitized maps of large dimensions request also a lot of data space on hard disk. The need of data compression arises if many maps should be stored on a computer hard disk, being available to a certain CAD program to display them and use them in an application. On cadastral maps run length encoding is already good enough for compression. It was applied on image segments to make possible their fast redrawing and almost smooth moving across the image.

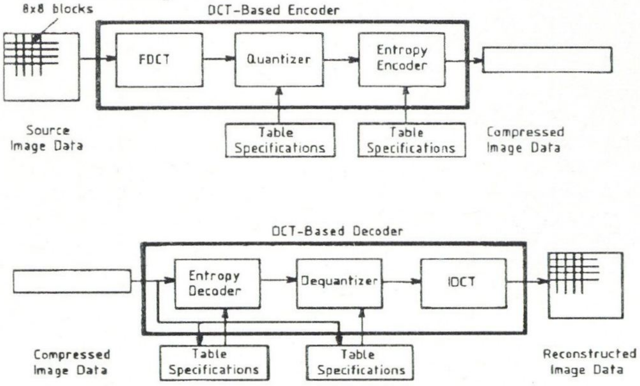


Fig. 7. JPEG compression and decompression

Grey level images demand much more storage space as black-and-white. The RLE (Runa Length Encoding) for every bitplane separately is not good enough. We found the JPEG algorithm [7] very suitable to compress 8 bit gray-level images. The compression ratio depends on a demanded image restoration quality. Because the input images were quite good, the quality of 40 % was selected and the compression ratio of approximate 10 was achieved. All the maps of Slovenia in measuring scale 1:50000, 256 gray-levels were compressed into 80 MB this way. The speed of JPEG software compression and decompression depends on a computer speed and available memory. This can be eliminated with hardware implementation of an algorithm, which is already commercially available.

7. SOME PROBLEMS WITH GPS DATA VISUALISATION

Both types of maps described so far were used in a GPS application where GPS data were visualized in real time. First maps were digitized and properly adjoined into coordinate system. Position from GPS receiver, which has a period of 1 second, was online transformed into the same coordinate system. GPS receiver provides position usually in geographic coordinates i.e., geographic latitude and longitude. Slovene cadastral maps are plotted in transverse mercator projection (TM) also called Gauss-Krüger projection. Transformation is done through mathematical queue, where the number of taken members, defines the precision. After the transformation a certain fixed error, that exceeded all known GPS errors together including SA, was observed. It was found out that this error was due to a digression of Slovenian geodetic net to world net. This fixed error was observed across the time and an average was estimated. Every incoming position was than first corrected by this average estimation. The same solution was applied to gray-level maps, plotted in geographic coordinates.

Some pre-processing of position data, including Kalman filtering was done to improve precision of position visualization in dynamic applications like vehicle tracking.

Besides the position it is necessary to visualize some other data from GPS (time, speed,...). Cadastral maps are usually sparse enough to permit writing some text near the position marks not covering important map data. Grey-level maps are much more sensitive to writing text over them. A layer based approach was implemented to overcome such troubles and to admit GPS data being incorporated in GIS environment.

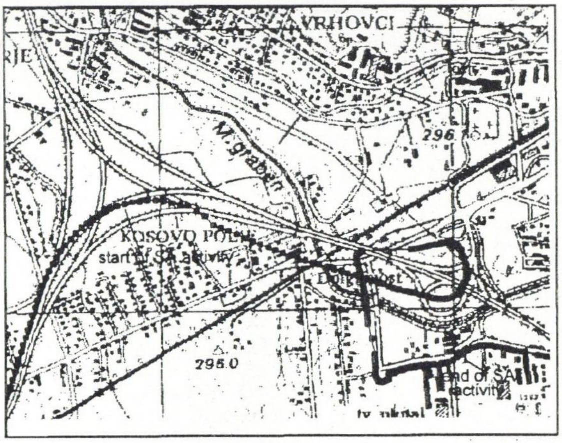


Fig. 8. Example of a real time GPS positioning (car tracking) using digital gray-level map

REFERENCES

[1] C. Feijóo, J. Ramos, F. Pérez: "A System for Fleet Management using Differential GPS and VHF Data Transmission Mobile Networks", VNIS, Ottawa, Oct. 12-15, 1993.

[2] M. Karima, K. S. Sadhal, T.O. McNeil: "From Paper Drawings to Computer-Aided Design", *IEEE Computer Graphics and Applications*, Vol. 5, No. 2, pp. 27-39. 1985.

[3] M. Ejiri et al.: "Automatic Recognition of Engineering Drawings and Map", Image Analysis Applications, R. Kasturi and M. M. Trivedi. eds., Marcel Dekker, New York, 1990.

[4] A. K. Jain: "Fundamentals of Digital Image Processing", Prentice Hall, Englewood Cliff, 1989.

[5] P. K. Sahoo, S. Soltani, A. K. C. Wong, Y. C. Chen: "A Survey of Thresholding Techniques", *Computer Vision, Graphics, and Image Processing*, No. 41, pp. 233-260. 1988.

[6] N. Pavešić: "Razpoznavanje vzorcev", ZAFER, Ljubljana, 1992.

[7] G. K. Wallace: "The JPEG Still Picture Compression Standard", *Communications*, April 1991.

[8] A. Dedić, R. Murn, D. Pećek: "Digitalization of Large Area Drawings and Maps", Proc. of Mediterranean Electrotechnical Conference – MELECOM, Ljubljana, 1991.

ON CLASSIFICATION OF AIRPLANE IMAGES FOR AUTOMATIC TARGET RECOGNITION*

C. TAŞKÍRAN, E. ANARÍM, B. SANKUR

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING,
BOGAZIÇI UNIVERSITY, BEBEK,
80815, İSTANBUL, TURKEY

1. INTRODUCTION

Automatic Target Recognition (ATR) is a very specific field of study within the general scope of pattern recognition (PR) and image processing. A typical problem of ATR may be to recognize, from a sequence of images (usually in the visual or infrared band), a possible target, such as a plane. ATR can be an extremely difficult task since the problem usually involves extraction of critical information from complex and uncertain data for which traditional approaches of signal processing, pattern recognition, and rule-based artificial intelligence have been unable to provide adequate solutions. However, despite the many difficulties inherent in ATR, it nevertheless will continue to be one of the key components of present and future defense weapon systems. This is due to the fact ATR removes man for the most part from the decision process and makes the system faster, reliable, and more invulnerable. The basic stages of an ATR system can be outlined as follows:

- *preprocessing* where noise in the incoming image is filtered,
- *image segmentation* or labelling where possible target or targets are spotted and extracted from other irrelevant objects and background,
- *feature selection* to obtain an optimal set of features,
- *classification* where the target is classified into one of the classes in the training set, and finally
- *tracking* for further action.

This paper is concerned with examination of two popular classification techniques used in PR, namely the k-nearest-neighbor (kNN) and linear discriminant function techniques, with special emphasis on the performance of the kNN algorithm. It is part of a project for the development of a complete ATR system, as described above. The outline of the paper is as follows: In Section II, an overview of different classifier types and training procedures is given. In Section III, the construction of the training set used in this study and dimensionality-problems, which are very important in designing classifiers are examined. The recursive method for estimating the L-class Bayes risk is introduced in Section IV. In Section V nearest neighbour techniques are described and various improvisations are discussed. In Section VI the linear discriminant classifier is examined. The noise performance of the kNN algorithm is examined in Section VII. Finally, in Section VIII conclusions are given.

2. CLASSIFIER TYPES AND ERROR ANALYSIS

Classifiers can be broadly categorized as statistical (decision-theoretic) and syntactic (structural). In the syntactic approach a complex pattern is represented by its simpler subpatterns. If the subpatterns are again complex, they may again be represented by even simpler subpatterns until one obtains the simplest subpatterns, called pattern primitives. Syntactic classification techniques are very powerful in cases where the patterns under study are very complex (for example, images of fingerprints) or when the number of classes is very large. The basic drawbacks for syntactic algorithms are their complexity and long processing times. On the other hand, in the statistical approach, a set of characteristic features is first extracted from the input pattern because usually the dimension of the pattern vector is too high for computation purposes (for example, for an image of 512 by 512 pixels, the resulting pattern vector is 262144 dimensional). Since the dimensionality must be substantially reduced, there is an inherent loss of information in the feature extraction process. Thus, the feature extraction algorithm should be chosen to make this loss as small as possible. Statistical algorithms are further divided into two subgroups as parametric where the form of the underlying probability function of the pattern vectors is assumed to be known, and nonparametric where such an assumption is not made. One other distinction between classifiers is their learning procedure. In supervised learning each feature vector in the training set has a class label and the classifier may improve its performance by checking its results with these known classifications. In unsupervised learning, however, such class labels are not given and in most cases it is not even known how many classes there are. Unfortunately, there is no general method to determine the best classifier in a given situation. The standard approach in PR literature is to try a number of classifiers and let the data dictate the best one.

The probability of misclassification (PMC) is the single most important property that characterizes the performance of the classifier, so effort should be spent to correctly estimate it. Small-sample effects are especially dominant in this estimation procedure and the estimate for the asymptotic PMC may be very biased, if not computed properly. To estimate the PMC, two well-known methods were used in this study: Resubstitution and leave-one-out methods. In the resubstitution method all observations are used in the design of the classifier and are also used in estimating its performance. This approach is particularly attractive when the number of observations is small. However, it suffers from too much optimistic bias. The leave-one-out method, on the other hand, produces an effectively unbiased estimate if the samples in the training set are independent. Since such a case is quite unlikely, in practical situations this estimate will also be biased. In [1] the symmetry properties of these two estimates are used to establish an estimate for the asymptotic error as

$$P_\infty = \frac{P_R + P_L}{2}, \tag{1}$$

where P_R and P_L are the resubstitution and leave-one-out estimates, respectively. This estimate is relatively unbiased and suitable for small-sample cases that may arise in practice. Another quantity developed in [1], which proves useful to determine if small-sample effects predominate, if the increase in classification error from the asymptotic value due to finiteness of the number of samples in the training set, $\Delta_N = E\{P_N\} - P_\infty$. This important quantity may be estimated using a nonparametric estimator as

$$\hat{\Delta}_N = \frac{P_N - P_L}{2}, \tag{2}$$

where P_N is the error estimate when N training samples are used in the estimation procedure, and P_∞ is the asymptotic error. If $\hat{\Delta}_N$ is "small" compared to the error estimate found from Eq. (1), then one may conclude that the number of samples in the training set is sufficient. If not, then new samples should be collected.

3. CONSTRUCTION OF THE TRAINING SET AND DIMENSIONALITY PROBLEMS

One of the most important points to consider in the design of classifiers is the problem of dimensionality. The relation between number of samples, N, number of features, n, and PMC was first investigated in [2]. It was theoretically shown that for a given sample size, the dimension of the feature vector has an optimal value, above which the PMC starts to increase again, a curious fact which has been called the "curse of dimensionality". A very rough rule of thumb is that the number of samples in the training set should be at least ten times the number of features used. However, in practice, it may often be quite hard to collect so many samples. Hence, small-sample size effects can easily degrade the design and evaluation of a proposed classifier.

* This work was supported by TÜBİTAK under contract EEEAG-83

The construction of the training set for this work was performed as follows: In our laboratory scaled down models for three different types of airplanes were built and their images were captured using an image grabber. The images were taken from the side, up and front for each plane model. Then, using these images a three dimensional tensor was formed for each plane class. By rotating this tensor by 14 degrees evenly throughout the space, 60 images per class were formed and an eleven dimensional feature vector was extracted from each image. These features consist of five of the traditional moment invariants calculated separately for the contour and silhouette of the airplane.

4. COMPUTATION OF THE BAYES RISK

The derivation of the Bayes risk for a given sample set should be the first thing done in classifier design because it gives a lower limit to the performance of any type classifier for the specific training set. If even the Bayes risk is unsuitable for one's purposes then designing a classifier is pointless and the sample set should be enlarged. However, computation of the Bayes risk for the L class case is difficult. Therefore, the iterative method given in [3] which is described below is particularly suited for the L class problem: The main idea of the method is to recursively reduce the problem of finding the Bayes risk for L classes to the well-known problem of two classes. If the *a priori* probabilities of the classes are equal, as usually assumed, then the bounds on the Bayes risk are

$$\Re \le \frac{L}{L(L-2)} \sum_{i=1}^{L} \Re_i^{L-1}$$

$$\Re \ge \frac{(L-1)^2}{L^2(L-2)} \sum_{i=1}^{L} \Re_i^{L-1}. \qquad (3)$$

We first compute the $\binom{L}{2}$ pairwise, binary Bayes risks. Then using the bounds given above the bounds on the ternary Bayes risks are found, and so forth. The pairwise Bayes risks may be computed once the probability density functions of the classes are estimated. This may be done using either the Parzen or the k-nearest-neighbour (kNN) method. In our work we have found that the Parzen estimate was better and gave lower variances than the kNN method, once a suitable smoothing parameter, α_n, is found. The Parzen estimate for the probability density function of $\mathbf{x}$ is given as

$$\hat{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{\alpha_n}\right)^{N} K'\left(\frac{\mathbf{x}-\mathbf{x}_i}{\alpha_n}\right). \qquad (4)$$

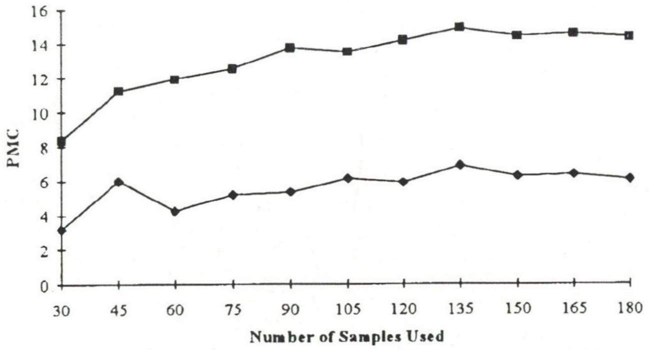


Fig. 1. Upper and lower bounds on the Bayes error for 3 features and $\alpha_n = 0.3$

The choice of the kernel $K'(\mathbf{u})$ does not affect the estimate very much and so we have chosen the simple hypercube function. The choice of α_n, however, greatly affects the estimate and care should be given when choosing it. Generally a range of values is tried and the best value is determined empirically. Using 3 features and 60 samples from each class we found that the optimal value of α_n is about 0.03. Such a small value of this parameter hints at the fact that the underlying density function is multimodal and non-Gaussian, a common phenomenon in ATR applications. The upper and lower bounds calculated using the above method are plotted in Fig. 1 for the case where the vectors in the training set are three dimensional (i.e. three features were used).

5. NEAREST NEIGHBOUR TYPE ALGORITHMS

The idea behind the NN rule is intuitively obvious as summarized by the saying "birds of a feather flock together". In recent years NN type classifiers have gained a tremendous popularity in PR applications. In the NN algorithm the distances between the feature vector to be classified and all the vectors in the training set are calculated. Then the given vector is classified to the class of the vector having the nearest distance. Despite its extreme simplicity, the NN algorithm may sometimes approach the PMC of a Bayes classifier and in the worst case it has twice this much error. In fact, the following well-known result may easily be proved [4]

$$\Re \le \Re_{NN} \le \Re\left(2 - \frac{L}{L-1}\Re\right), \qquad (5)$$

where L is the number of classes, $\mathcal{R}$ is the asymptotic Bayes risk, and $\Re_{NN}$ is the asymptotic risk of the NN classifier. It is seen that when L is large, as is the case usually, the NN risk is bounded above by twice the Bayes risk. One should be very careful in using Eq. (6), however, because that result was proved for the asymptotical case of infinite samples. When a finite number of samples is used the NN classifier is biased. In [5] the expected value of this bias is derived as:

$$E(b(\mathbf{x}_1\mathbf{x}_{\mathbf{NN}})) = \beta_1 E_{\mathbf{x}}(|\mathbf{A}|^{1/N} tr(\mathbf{A}^{-1}B_M(\mathbf{x}))), \qquad (6)$$

where $\mathbf{A}$ is the distance measure matrix used and $B_M(\mathbf{x})$ is the second order Taylor series expansion of the risk. The quantity β may be expressed as

$$\beta_1 = \frac{\Gamma^{2/N}(N/2+1)\Gamma(2/N+1)\Gamma(N+1)}{N\pi\Gamma(2/N+N+1)}, \qquad (7)$$

where Γ represents the gamma function and N is the number of samples used in classification. One thing to note from the above complicated expression is that the bias of the NN classifier drops off rather slowly with increasing number of samples.

5.1. *Effects of the Distance Metric*

The first step in using an NN type classifier algorithm is to decide on a suitable distance measure. The general quadratic distance function between a given feature vector, $\mathbf{x}$, and a feature vector of a sample belonging to class i, $\mathbf{y}_i$, may be written as

$$d^2 = (\mathbf{x}-\mathbf{y}_i)^T\mathbf{A}(\mathbf{x}-\mathbf{y}_i). \qquad (8)$$

Though there is quite a large number of such measures (for a summary refer to [6]), the most commonly used ones are the Euclidean distance for which $\mathbf{A} = \mathbf{I}$, and the Mahalanobis distance, for which $\mathbf{A} = \Sigma_i^{-1}$, where Σ_i is the covariance matrix of class i. The Mahalanobis distance has the useful property of taking into account the correlation between the feature elements. However, since in ATR applications computational time is a main consideration, the Mahalanobis distance is not used in this study because it involves a matrix inversion, which is computationally demanding.

The result given in Eq. (5) is independent of the particular metric used; however, when the number of samples is finite, the performance of the NN classifier depends a great deal on the distance metric. From Eq. (6) one may observe how the distance matrix $\mathbf{A}$ affects the bias of the NN classifier. Unfortunately, since $B_M(\mathbf{x})$ is a very complicated expression of $\mathbf{A}$, the selection of $\mathbf{A}$ to minimize the bias when the distribution of the data is nonGaussian is not clear. Trying to improve the efficiency of the NN algorithm by modifying the distance metric looks promising but very little work has been done in this area. Most studies generally assume either the Euclidean metric or a similar variant such as the city-block metric. Another popular approach, which is discussed below is to modify the metric by weighing the distances in the originally defined metric which cannot be considered too great a modification.

Quite a different approach is provided in [7] where an optimal distance measure is introduced. This measure uses a local approximation to the conditional density function in the neighbourhood of the given feature vector to be classified. One problem with this approach is that the metric should be evaluated for each different **x** from a small region surrounding **x**. Still another problem is that since the metric is local, information concerning important features of the distribution cannot be recovered from it easily. In order to remedy these problems, an optimal global NN metric was proposed in [8].

5.2. The kNN Rule

The k-nearest-neighbour (kNN) algorithm is a modification of the NN algorithm in which k feature vectors in the training set that have the least distance to the given feature vector are found and the vector is classified to the class which has the greatest number of representatives in these k vectors. If two classes have the same number of representatives, then either the given vector is rejected without classification or the tie is broken randomly by throwing an L-faced die. Unfortunately, the undesirable effect of the NN algorithm is also present in the kNN algorithm and the performance does not increase considerably even when the number of samples is doubled. The change of the performance of the kNN classifier with the number of samples and number of features used is shown in Fig. 2 for three neighbours. The slow decrease in PMC is evident. We also observe that the estimation of the Bayes risk found from Fig. 1 is quite correct. The change of performance with the number of neighbours used may be observed from Fig. 3 where 11 features were used. One may observe that for our training set between three or five neighbours gives good results whereas increasing the number of neighbours deteriorates the performance. This may be explained by the fact the feature vectors in our training set do not form neat hyperellipsoids in the feature space and the classes are quite mixed up with each other.

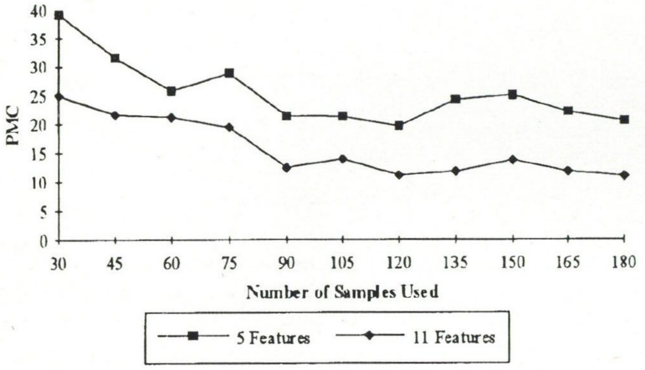


Fig. 2. The change of performances with different number of features of the kNN algorithm using 3 neighbours

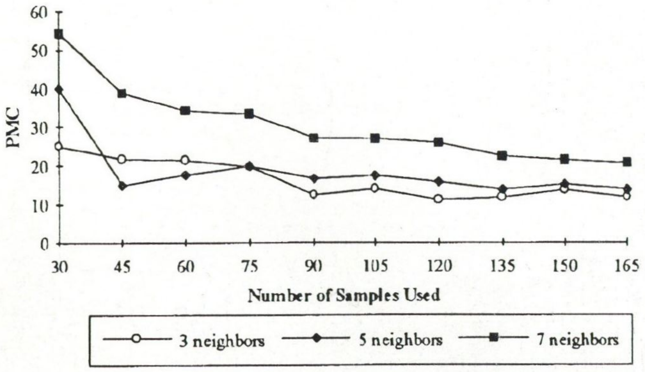


Fig. 3. The change of performances of the kNN algorithm with increasing number of neighbours using 11 features

Another property of the NN and kNN algorithms is that when the number of samples used is small, these type of classifiers run into small-sample size problems and result in very large values of PMC. This fact may be checked when the estimate given in Eq. (2) is used to test whether the number of samples is adequate. We have observed that for three classes when the total number of samples is less than 30, kNN has inadequate number of samples and should not be used. This limitation may not affect applications such as character recognition where hundreds of samples may easily be found; however in ATR application one rarely has this luxury and small sample size is the rule rather than the exception.

5.3. The Weighted kNN Rule

One of the main problems of the kNN algorithm is the occurance of ties. Although these may be resolved randomly, it is better to devise a method which avoids them completely. In order to solve this problem, while at the same time increasing the performance of the algorithm, one might propose the intuitively appealing idea of weighing the votes of the neighbours which are nearer to the vector to be classified more heavily instead of a plain majority-vote decision. Indeed this improvisation has been proposed quite early by Dudani in [9] as follows: Suppose for $j < k, d_j$ is the distance between the j'th neighbour and the given vector to be classified. Then we define the weight of this neighbour as

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & d_k \neq d_1 \\ 1 & d_k = d_1 \end{cases} .$$

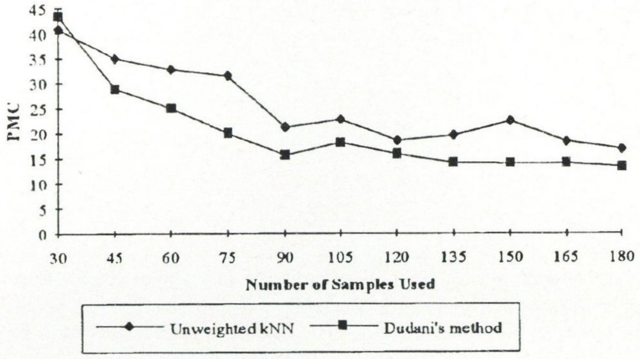


Fig. 4. Comparison of Dudani's method and unweighted kNN rule for 3 neighbours and 11 features

Then the given vector is assigned to the class that has the greatest total weight. It may easily be seen that the occurance of ties with this method is quite unlikely. This method has been criticised heavily by a number of researchers in that its performance is inferior to that of the unweighted kNN rule. The problem with the above rule is that when $d_j = d_k$ we set $w_j = 0$ and thus the k'th neighbour is effectively excluded from the decision process. Hence the weighted kNN rule, as described above, cannot perform as well as the unweighted one. We have implemented and compared Dudani's method as described above and compared it with the unweighted kNN classifier. The results are shown in Fig. 4 and Fig. 5 for different number of features used. It is observed that Dudani's method is worse when a small number of features is used and even when it is superior the difference and is too slight to warrant the use of the algorithm. Note that the pessimistic leave-one-out estimate only was plotted in these graphs and the averaging procedure given in Eq. (1) is not used. This is because it is impossible to use the resubstitution estimate when working with weighted kNN rules. This undesirable property may be corrected by redefining the weighing procedure as follows [10]:

$$w_j = \begin{cases} \frac{(d_s - d_j) + \alpha(d_s - d_1)}{(1+\alpha)(d_s - d_1)} & d_s \neq d_1 \\ 1 & d_s = d_1 \end{cases} ,$$

where d_s is the distance between the given vector and its s'th nearest neighbor and α is a positive constant. When $d_s = d_k$ and $\alpha = 0$ this rule boils down to the one given previously. It is shown in [11] that, given a training set containing infinitely many samples, the unweighted kNN rule always outperforms any type of weighted rule. However, in [10] it is argued that this may not always be the case when the number of training samples is finite. Hence with a good choice of α and d_s it may be possible to outperform the unweighted kNN rule. We have tried the weighted kNN rule with different values of these parameters together with the unweighted kNN rule. The results for this α-weighted kNN rule is shown in Fig. 6 for two values of α. It may be observed that the performance is bad compared with the unweighted kNN rule. One really has to play around with the variables in this model to get better results but in our case this rule has not produced superior results for a wide range of parameter values.

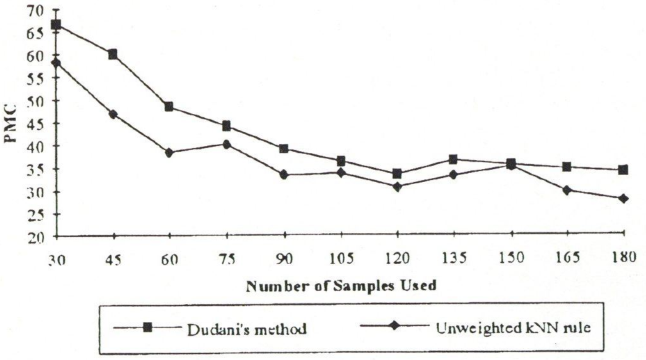


Fig. 5. Comparison of Dudani's method and unweighted kNN rule for 3 neighbours and 5 features

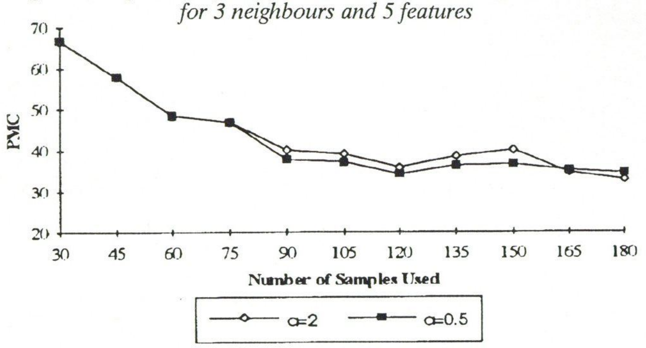


Fig. 6. Change of performance of the α-weighted kNN rule for different values of α

6. LINEAR DISCRIMINANT CLASSIFIERS

Linear discriminant classifiers (LDC's) represent quite a different approach from the distance-based algorithms in that they first transform the given sample space to a one having a lower dimension. In this way the number of distances to be computed from the given vector to the sample vectors is greatly reduced. The general LDC for the L classes may be written as L functions of the form

$$g_i(\mathbf{x}) = w_i^T \mathbf{x} + w_{i0}, \tag{9}$$

where w_i is an N dimensional vector, commonly referred to as the weight vector, for class i, and w_{i0} is called the threshold weight for the same class. The classification algorithm is then to calculate $g_i(\mathbf{x})$ for each class and assign $\mathbf{x}$ to class i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$. The question of where to assign x if the results for two classes are equal naturally arises and this is an open academic problem. Most researchers, [4] and [12], recommend the rejection of $\mathbf{x}$ when this occurs and we have followed this convention in this work. The LDC has its own drawbacks, however. The decision regions are convex which limit the applicability of the classifier a great deal. Due to this fact the LDC is better suited to problems where the *a priori* probabilities of the classes are unimodal, which is not generally expected in ATR applications. Moreover, the classes have to be linearly separable, which is an extremely hard condition to check for. One way to convert a linearly inseparable set into a separable one is to increase the number of features used. Unfortunately, as we have seen, this can not usually be achieved in practice because of dimensionality problems. To illustrate the method, we first define the vectors for each class i as

$$\mathbf{u}_j = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}.$$

These vectors are L dimensional and they have the special property that the distance between any two of them is identical and that their norms are identical. Then we let $\mathbf{v}_m = \mathbf{u}_i 2$ if the sample $\mathbf{x}_m$ belongs to class i. Thus, the set of vector pairs $(\mathbf{x}_m, \mathbf{v}{-}m)$ summarizes completely the information in the training set. Next we look for a linear transformation in the form:

$$\mathbf{v} = \mathbf{T}\mathbf{x} + \mathbf{b}. \tag{10}$$

We desire to find a T such that the interclass distance should be as large as possible while preserving the intraclass distances. We also desire T to be linear since deriving optimality conditions for a nonlinear transform is quite difficult and it also brings about excessive computational requirements. However, using a linear transformation is not as restrictive as it looks because if we want, say, a quadratic transform, then we first transform the sample space quadratically to a new one and seek a linear transformation there. The solution to this problem is the linear regression function as given by

$$v = \mathbf{R}_{\mathbf{VX}} \mathbf{R}_{\mathbf{X}}^{-1}(\mathbf{x} - \mathbf{m}_\mathbf{x}) + \mathbf{m}_\mathbf{v}, \tag{11}$$

where we define

$$\mathbf{m}_\mathbf{v} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}_\mathbf{i} \tag{12}$$

$$\mathbf{m}_\mathbf{x} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_\mathbf{i} \tag{13}$$

$$\mathbf{R}_{\mathbf{vx}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{v}_\mathbf{i} - \mathbf{m}_\mathbf{v})(\mathbf{x}_\mathbf{i} - \mathbf{m}_\mathbf{x})^\mathbf{T} \tag{14}$$

$$\mathbf{R}_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_\mathbf{i} - \mathbf{m}_\mathbf{x})(\mathbf{x}_\mathbf{i} - \mathbf{m}_\mathbf{x})^\mathbf{T}. \tag{15}$$

Note that, though a matrix inverse operation is called for, the number of distances to be computed has dropped from N to L, which brings about enormous savings in computation time (In our case, the maximum number of samples used was 180 while the number of classes were three!). The results of the LD function classifier is shown in Fig. 7. The performance much worse than the kNN rule as may be observed. This is due to the fact that our classes are probably not linearly separable.

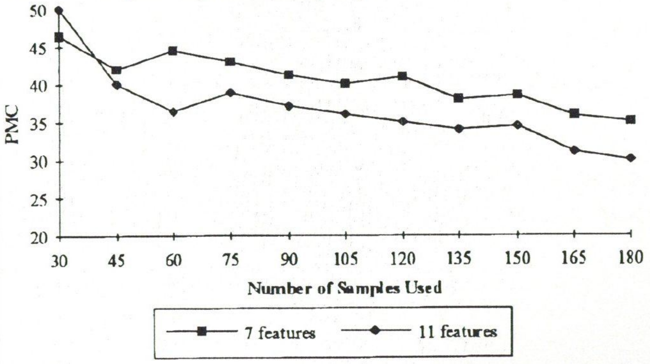


Fig. 7. The performance of the LDF classifier for different number of features

7. NOISE PERFORMANCE

Checking the performance of classification algorithm S against noise is important because even though the incoming image is filtered from noise, if the incoming image is too noisy, the filtering operations may not be able to filter all the noise present and some of it may reach the classification stage. The amount of this noise present would be very small if appropriate filtering operations are applied. Therefore, it is necessary to check the performance of classification algorithms for small values of noise present in the training set. The images in the training set were contaminated by 5 % impulsive noise. Then the feature vectors for all these contaminated images in the set were extracted and the classifier algorithms were tried on this noisy input. The performance of the kNN algorithm for the contaminated sample set is shown in Fig. 8 where 11 features were used. By comparing this figure with Fig. 3. we observe that the increase in PMC is between 10 %−15 % regardless of the number of features. When a high number of neighbours are used, the increase in PMC is greater. The reason for this difference is that the noise tends to be averaged out between the neighbours.

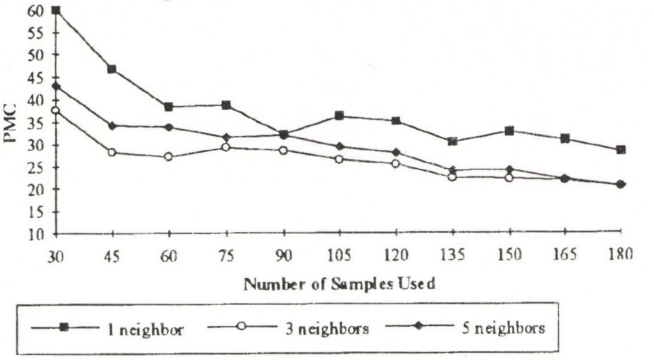


Fig. 8. The performance of the unweighted kNN rule against 5 % impulsive noise for different number of neighbours

8. CONCLUSIONS

From the results observed, we may conclude that the NN and kNN classifiers have small-sample size problems and should not be used unless the number of samples is proved to be adequate by the estimate in Eq. (2). Another drawback for the kNN and NN classifiers is that their bias does not drop considerably when the number of samples in the training set is increased. Hence, it may be quite difficult in practice to obtain the theoretical limits for the asymptotic PMC for these classifiers. One other factor that should be considered seriously is the amount of noise in the images constituting the training set of the classifier. Our results indicate that with as little as 5 % impulsive noise in the samples, the PMC tended to increase in the range of 10-15 %. Such an increase is not acceptable in most applications so noise reduction in the training images should be done very thoroughly. The weighted kNN algorithms did not perform as well as the unweighted ones for our training set. This may be due to the reason that our distribution is multimodal and non-Gaussian as witnessed by the value of a found from the Parzen estimate. The same argument holds for the inadequacy of the linear discriminant classifier. The classes are not quite linearly separable which leads to a high PMC of the LD classifier algorithm. One way to overcome these problems is to increase the number of features which in turn requires a great increase in the number of training samples. This may not always be possible practically.

REFERENCES

[1] S. J. Raudys, A. K. Jain: "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. PAMI-13, pp. 252-264, March 1991.

[2] G. F. Hughes: "On the Mean Accuracy of Statistical Pattern Recognizers," *IEEE Trans. Inform. Theory.*, Vol. IT-14, pp. 55-63.

[3] F. D. Garber, A. Djouadi: "Bounds on the Bayes Classification Error Based on Pairwise Risk Functions," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. PAMI-10, pp. 281-288, March 1988.

[4] R. O. Duda, P. E. Hart: *Pattern Classification and Scene Analysis*, Wiley Interscience.

[5] K. Fukunaga, D. M. Hummels: "Bias of Nearest Neighbor Estimates," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. PAMI-9, pp. 103-112, 1987.

[6] M. Baseville: "Distance Measures for Signal Processing and Pattern Recognition," *Signal Processing* No. 18, pp. 349-369, 1989.

[7] R. D. Short, K. Fukunaga: "The Optimal Distance Measure for Nearest Neighbor Classification," *IEEE Trans. on Inform. Theory*, Vol. IT-27, pp. 622-627, 1981.

[8] K. Fukunaga, T. E. Flick: "An Optimal Global Nearest Neighbor Metric," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. PAMI-6, pp. 314-318, 1984.

[9] S. A. Dudani: "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Transactions on Systems, Man., and Cybernetics*, Vol. SMC-6, pp. 325-327, 1976.

[10] J. E. S. Macleod, A. Luk, D. M. Titterington: "A Re-Examination of the Distance Weighted k-Nearest Neighbor Classification Rule," *IEEE Transactions on Systems, Man., and Cybernetics*, Vol. SMC-17, pp. 689-696, 1987.

[11] T. Bailey, A. K. Jain: "A Note on Distance-Weighted k-Nearest Neighbor Rules," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-8, pp. 311-313, 1978.

[12] T. Y. Young, T. W. Calvert: *Classification, Estimation, and Pattern Recognition*, Elsevier, 1974.

UNSUPERVISED STATISTICAL SEGMENTATION OF POLARIMETRIC SYNTHETIC APERTURE RADAR DATA USING THE K-DISTRIBUTION

A. SAAD and D. BARBA

LATI/SEI (SYSTÈMES ELECTRONIQUES ET INFORMATIQUES)
IRESTE, LA CHANTRERIE
CP 3003, 44087 NANTES CEDEX 03, FRANCE

S. EL ASSAD

S2HF/SEI (SYSTÈMES ELECTRONIQUES ET INFORMATIQUES)
IRESTE, LA CHANTRERIE
CP 3003, 44087 NANTES CEDEX 03, FRANCE

1. INTRODUCTION

Classification of earth terrain within an image is an important application of polarimetric data. The normalized magnitude and phase of the waves backscattering from terrain are reliable features for data classification purposes.

A number of polarimetric SAR analysis techniques have been reported in the literature to measure and characterize the polarization response of natural targets, to maximize the contrast between regions based on polarimetric filtering, or to classify data using Bayes classifier. Most of these classification techniques are supervized and then require the selection of training areas for each class of terrain cover. However, an accurate and detailed knowledge of the scene contents is required to select the appropriate classes, and training areas should be homogeneous and contain enough samples to estimate the polarimetric backscatter characteristics of each class with good accuracy. This is not always possible, and in addition, selecting training areas becomes time consuming as the number of classes, the data volume, and the data rate increase. Because of these limitations, there is a strong interest in developing unsupervized techniques for analyzing polarimetric SAR data.

The object of this paper is to present a method of unsupervized analysis for polarimetric SAR data. This method segments the data into classes of homogeneous microwave polarimetric backscatter characteristics.

Polarimetric backscatter classes selection is based on a multidimensional fuzzy clustering [1] of the logarithm of the parameters composing the polarimetric covariance matrix. The advantage of operating in the log domain are outlined in section 3.

Clustering is set up by only one parameter which is the number of classes. Given the classes of polarimetric backscatter, a maximum a posterior (MAP) polarimetric classifier is used to segment the entire polarimetric array. A recent study [4] has shown that the statistical properties of forest, using polarimetric SAR data, are well represented by the K-distribution. In this paper, the K-distribution in the MAP classifier is used to classify the forest areas and the results are compared with the ICM classification method using a Gaussian model.

The potential and usefulness of the unsupervized technique is illustrated using fully polarimetric SAR complex data acquired by the NASA/Jet propulsion Laboratory Airborne polarimetric radar (AIRSAR).

The paper is organized as follows. In section 2, Gaussian and the K-distribution models for the polarimetric SAR data are presented. In section 3 the polarimetric feature vector used for clustering of the polarimetric SAR data and the clustering technique are defined. Section 4 presents the MAP polarimetric classifier. In section 5 experimental results are analyzed. A conclusion summarizes the major drawbacks of the proposed technique.

2. STATISTICS MODELS OF POLARIMETRIC SAR DATA

In this section, two models for the conditional distribution of the polarimetric SAR complex data, given the region labels, are presented. The region label of pixel site s in the image plane is designated as $Ls = l$ with $l \in \{1 \dots k\}$, where k is the number of regions, Y_s denotes the single look polarimetric measurement vector at pixel site s, i.e., the vector of the three single look polarimetric complex amplitudes measured at site s by the polarimetric radar,

$$y_S = \begin{bmatrix} HH \\ HV \\ VV \end{bmatrix} = \begin{bmatrix} HH_i + iHH_q \\ HV_i + iHV_q \\ VV_i + iVV_q \end{bmatrix}, \tag{1}$$

with:
HH: the signal transmitted is H polarized and the return response is H polarized (copolarization).
HV: the signal transmitted is H polarized and the return response is V polarized (cross-polarization).
VV: the signal transmitted is V polarized and the return response is V polarized.

The VH return is not present in Eq. (1) since symmetrized with the *HV* return during compression and calibration of the data.

2.1. Gauss model

A Gaussian conditional distribution of the single look polarimetric measurement vector Ys given its region label Ls is assumed,

$$P(Y_s/l) = \frac{1}{\pi^{3/2}|C_l|^{1/2}} \exp -\frac{1}{2}(Y^{\gamma} C_l^{-1} Y_s). \tag{2}$$

The superscript γ denotes complex conjugation and the transpose of the vector. The 3 by 3 complex matrix $C_l =< YY^{\gamma} >_l$ where $< >$ denotes ensemble averaging, is the polarimetric covariance matrix of the data in region l. The Gaussian model assumes that each region has stationary backscatter statistics, i.e., the covariance matrix C_l is translated invariant. The only source of spatial variability of the SAR signal at different polarizations and within an homogeneous region is the image speckle. In the case of the real polarimetric SAR data used in this paper, measures of local statistical characteristics of the signal such as its variance and correlation coefficient reveal that the Gaussian assumption is not satisfactory [5][8]. The SAR signal is also significantly modulated by a spatial variability of the backscatter coefficient of the imaged surface, in which case multivariate K-distribution may better model the statistics of the polarimetric SAR data [5][7].

Direct evaluation of Eq. (2) at each pixel location involves prohibitive computation of the Hermitian form $(Y_s^{\gamma} C_l^{-1} Y_s)$. A simplification is possible in the presence of azimutally symmetric targets [3], since the *HV* amplitude is uncorrelated with the *HH* and *VV* complex amplitudes, and the covariance matrix is:

$$C_l = \sigma_l \begin{bmatrix} 1 & 0 & \rho_l\sqrt{\gamma_l} \\ 0 & \varepsilon_l & 0 \\ \rho_l^* \sqrt{\gamma_l} & 0 & \gamma_l \end{bmatrix} \tag{3}$$

with

$$\sigma_l = E[|HH|^2]_l \tag{4}$$

$$\epsilon = \frac{E[|HV|^2]_l}{E[|HH|^2]_l} \tag{5}$$

$$\gamma_l = \frac{E[|VV|^2]_l}{E[|HH|^2]_l} \tag{6}$$

$$\rho_l = \frac{E[HHVV^*]_l}{(E[|HH]^2]_l E[|VV|^2]_l)^{0.5}}, \tag{7}$$

where the superscript * denotes the complex conjugate. Replacing C_l in Eq. (2), leads to [3]:

$$P(Y_s/l) = \frac{1}{\pi^{3/2}} \exp \Big\{ -\frac{|HH|^2}{\sigma_l(1-|\rho_l|^2)} - \frac{|HV|^2}{\sigma_l \varepsilon_l} - \frac{|VV|^2}{\sigma_l \gamma_l (1-|\rho_l|^2} + 2\Re e \frac{(HH \cdot VV^* \cdot \rho_l)}{\sigma_l\sqrt{\gamma_l}(1-|\rho_l|^2)} -Ln(\rho_l^2 \varepsilon_l \gamma_l (1-|\rho_l|^2)) \Big\} . \tag{8}$$

To improve the labeling process to each pixel site s, the conditional distribution of a single polarimetric measurement vector Eq. (1) is replaced by the joint conditional distribution of a small set of polarimetric measurement vectors contained in a neighborhood *Vs* of site s. Then, region labeling will exploit the additional polarimetric information provided by the neighbors of s.

We will assume that the N polarimetric measurement vectors *Ys* contained in *Vs* are spatially uncorrelated, i.e., conditionally independent as they are Gaussian. The conditional distributions of the Y_i's becomes [3].

$$P(Y_s/l) = \frac{1}{\pi^{3N}} \exp\{-NU_1^s(Y_s/l)\}, \tag{9}$$

where the energy function $U_1^s(Ys/Ls = l)$ is:

$$U_1^s(Ys/Ls = l) = -\frac{|HH^N|^2}{\sigma_l(1-|\sigma_l|^2)} - \frac{|HV^N|^2}{\sigma_l\varepsilon_l} - \frac{|VV^N|^2}{\sigma_l\gamma_l(1-|\rho_l|^2)} + + 2\Re e \frac{(HH^N \cdot VV^{*N} \cdot \rho_l)}{\sigma_l\sqrt{\gamma_l}(1-|\rho_l|^2)} - Ln(\sigma_l^2 \varepsilon_l \gamma_l (1-|\rho_l|)) \tag{10}$$

with:

$$HH^N VV^{*N} = \frac{1}{N} \cdot \sum_{i=1}^{N} HH_i VV_i^*, \tag{11}$$

and

$$HH^N = \frac{1}{N} \cdot \sum_{i=1}^{N} |HH_i|^2. \tag{12}$$

Computing Eq. (12) is exactly equivalent to multi-look operation performed on N single look polarimetric SAR complex data samples. Hence, the segmentation technique described here is applicable in a straightforward manner to the case of multi-look polarimetric SAR complex data.

2.2. K-distribution model

In previous publications [2][3] on this subject, Gaussian statistics have been frequently assumed for the radar return signals to build the Bayes terrain classifier. However, abundant experimental evidence shows that terrain radar cluster is non Gaussian, i.e., non Rayleigh in amplitude distribution. Among many non Gaussian statistics, the K-distribution has proven to be useful in characterizing the amplitude distribution of electromagnetic echoes from various objects, including diverse ground surface and wave propagation through atmospheric turbulence.

By assuming a product model where the received polarimetric returns are the product of complex Gaussian random vector and a Gamma-distributed scaling factor, a K-distribution with two parameters μ and g was derived in Novak [11]. In this paper, another approach, a clustered terrain scatterer model, is used to derive the k distribution with one parameter α. By setting $\mu = \alpha$ and $g = 1/\alpha$; the result in [11] can be transformed into Eq. (13) in this paper,

$$P(Y/l) = \frac{(2\alpha)^{n/4+\alpha/2} \cdot (Y_s^{*T} C_l^{-1} Y_s)^{\alpha/2-n/4}}{(2\pi)^{n/2} |C_l|^{1/2}}$$

$$\times K_{n/2-\alpha}((2\alpha)^{1/2} \cdot (Y_s^{*T} C_l^{-1} Y_s)^{1/2}) \qquad (13),$$

where n is the vector dimension, a is the parameter of the model.

The normalized intensity moments of *HH*, *HV VV* are defined as follows

$$I^m HH = \frac{< |HH|^{2m} >}{< |HH|^2 >^m} \qquad (14)$$

$$I^m HV = \frac{< |HV|^{2m} >}{< |HV|^2 >^m} \qquad (15)$$

$$I^m VV = \frac{< |VV|^{2m} >}{< |VV|^2 >^m} \qquad (15)$$

In particular, the second-order intensity moment is given as:

$$I^{(2)} = 2(1 + 1/\alpha). \qquad (17)$$

In this paper α is estimated by simple averaging,

$$\alpha = \frac{(\alpha_{HH} + \alpha_{HV} + \alpha_{VV})}{3}. \qquad (18)$$

2.3. Markov random field

Markov random fields (MRF) are mathematically convenient for representing local interaction between neighboring pixel attributes. They can model the spatial extent of the interactions. They have an equivalent description in terms of Gibbs energy function which provides a more practical way of describing the state of organization of physical attributes within a system than local probabilities. The conditional distribution of the region label Ls, given the region labels elsewhere, is only dependent on the region labels of an immediate neighborhood (Markovian property) and is expressed as:

$$P(Ls = l/Lv) = \frac{1}{\mathcal{Z}} \exp\{-NU_2^s(Ls = l/Lv)\} \qquad (19)$$

with:

$$U_2^s(Ls = l/Lv) = -\frac{\beta}{N} \cdot \left[\sum_{r \in Vs} \delta(Ls - Lr)\right] \qquad (20)$$

and

$$\mathcal{Z} = \sum_i \exp\{-NU_2^s(Ls = l_i/Lv)\}. \qquad (21)$$

Lv is the set of labels neighbors, *Vs* is a neighborhood of N elements excluding its center s, $\mathcal{Z}$ is a positive normalizing constant independent of l, β is a positive constant and δ is the Kronecker symbol.

To be consistent with Eq. (10) the same neighborhood structure is selected for modeling polarimetric complex amplitudes and the region labels *Vs* is 3 by 3 square box, i.e., a second-order neighborhood window.

Higher order neighborhoods provide better smoothing of the data and segmentation accuracy within homogeneous areas but impair the detection of small structural details and also increase the classification error at region boundaries. The positive constant β encourages neighboring pixels to have the same region label and also determines the degree of clustering.

The value $\beta = 1, 4$ was adopted in all our experiments.

3. VECTOR CHARACTERISTIC AND MULTIDIMENSIONAL CLUSTERING USED TO PARAMETERS ESTIMATION

The polarimetric covariance matrix C_l fully characterizes the first order scatiscics of class l of polarimetric backscatter. Hence, a possible mode of unsupervized selection of the classes of polarimetric backscatter is to perform clustering on a feature vector composing the matrix C_l. We choose the feature vector Xc to be equal to the logarithm of the parameters composing the polarimetric covariance matrix as:

$$X_c = \begin{bmatrix} < Ln(|HH|^2) > \\ < Ln(|HV|^2) > \\ < Ln(|VV|^2) > \\ Ln(| < HHVV^* > |) \end{bmatrix}. \qquad (22)$$

The first three coefficients represent the backscatter cross section in dB of the surface element at three different linear polarizations. The fourth component measures the magnitude of the $HH - VV$ correlation function. The polarimetric feature vector *Xc* contains the complete polarimetric information in the case of azimutally symmetric targets.

The advantage of operating in the log domain instead of the linear domain are two-fold [2]:

- In the log domain, image speckle has the statistical characteristics of additive noise with a power level not varying much across the image. Therefore, operating in the log domain renders clustering robust to the presence of image speckle.
- In the linear domain, the cross-polarized terms (i.e. *HV*) are often several orders of magnitude smaller than the copolarized terms (i.e. *HH*, *VV*), and clustering is mainly driven by the copolarized terms unless an arbitrary weighting of the different channels is used.

The optimal weighting may vary with the type of targets or the environmental conditions.

To measure the separation between a polarimetric feature vector Xc_i of the $i^{'th}$ sample and cluster center O_l (written as a vector similar to Xc_i), an Euclidean distance may be used. The choice of this metric leads to iterated fuzzy c-means (FCM) clustering algorithm which minimize the following objective function EC with respect to fuzzy membership $P(Xc_i, l$ and cluster center O_l [1]:

$$EC = \sum_{l=1}^{k} \sum_{i=1}^{m} P(Xc_i,, l)^\alpha \cdot |Xc_l - O_l|^2, \qquad (23)$$

where m is the number of samples used in the clustering algorithm, α is the fuzziness index, $\alpha > 1$.

The FCM algorithm is executed in the following steps:

- Initialize memberships $P(Xc_i, l)$ of Xc_i belonging to cluster l such that
$$\sum_{i=1}^{k} P(Xc_i, l) = 1. \qquad (24)$$
- Compute the fuzzy centroid O_l for $i = 1, 2, \ldots, k$ using
$$O_j = \sum_{i=1}^{m} (P(Xc_i, j)^\alpha \cdot Xc_i) / \sum_{i=1}^{m} P(Xc_i, j)^\alpha. \qquad (25)$$
- Update the fuzzy membership $P(Xc_i, l)$ using
$$P(Xc_i, l)^\alpha = (1/|Xc_i - O_l|^2)^{\alpha-1} /$$
$$\sum_{j=1}^{k} (1/|Xc_i - O_j|^2)^{\alpha-1}. \qquad (26)$$

- Repeat steps 2) and 3) until the value of EC is no longer decreasing.

In our experiment, we have chosen $\alpha = 1.4$, the initial configuration of the cluster centers is the Isodata solution, and the FCM optimization process is iterated until the number of samples points changing their memberships $P(Xc_i, l)$ by more than $e = 1\ \%$ is less than $4\ \%$.

A known inconvenience of the FCM clustering algorithm is that it is not adapted to the situation where natural clusters are elongated in one direction. However, variability of the cluster elements can be modeled as resulting from three different sources:

- images speckle
- system noise
- texture.

In the log-domain, the variance of the signal due to image speckle and system noise is constant whatever the polarization. As a result, it is reasonable to assume that the clusters are nearly spherical in the polarimetric feature space.

4. THE MAP CLASSIFIER AND ITS IMPLEMENTATION

Using Bayes' theorem, the posterior distribution of the region label Ls given the single look polarimetric amplitudes Ys and the region label of a neighborhood Vs is:

$$P(Ls = l/Lv, Ys) = P(Y_s/l) \cdot P(Ls = l/Lv)/P(Y_s). \quad (27)$$

As Ys is known, $P(Ys)$ is just a positive constant independent of the region labels. Expressed in terms of an energy function, we have:

$$P(Ls = l/(Y_s, Lv)) \propto \exp[-NU_1^s(Ys/Ls = l) - NU_2^s(Ls = l/Lv)]. \quad (28)$$

The MAP estimate of L minimizes the global energy of the image expressed as:

$$E_{MAP} = \sum_s [U_1^s(Y_s/Ls = l + U_s^s(Ls = l/Lv)]. \quad (29)$$

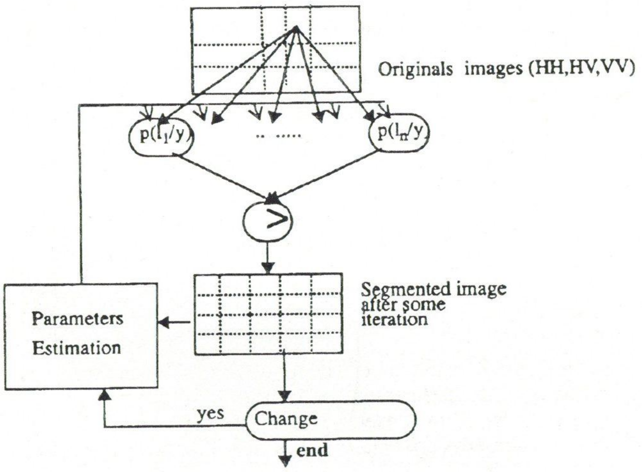

Originals images (HH,HV,VV)
$p(l_1/y)$
.. ..
$p(l_n/y)$
>
Segmented image after some iteration
Parameters Estimation
yes
Change
end

Fig. 1. ICM Implementation

Energy functions U_1^s and U_2^s only depend on local conditions, i.e., local polarimetric amplitudes and region labels. The necessary iterative stochastic relaxation algorithms are often implemented by simulated annealing, but the cost is intolerable for many applications. The MAP is thus implemented using a deterministic technique: the ICM (Iterative Conditional Modes) algorithm [10]. The initial configuration is taken as the result of the FCM algorithm. At each iteration through the image sites, the state of each site is either changed to the state that yields maximal decrease of energy, or is left unchanged when no energy reduction is possible. The process always stops at a local minimum when no more changes can be made. To accelerate the convergence of ICM we use a table of stability measured by Markov probability in the labeled image. A pixel is classified as unstable if there exists at least one pixel in the neighborhood with a different label. Fig. 1 shows the ICM method.

5. EXPERIMENTAL RESULTS

The results are obtained from data measurements of the MAESTRO-1 campaign made by the NASA/JPL polarimetric AIRSAR system which operates at P (0.44 GHz), L (1.225 GHz) and C (5.3 GHz) frequency band with an incident angle ranging approximately from 40° to 50°. The campaign measurements were made on the 16 of August 1989 when meteorological and environmental conditions were optimal in France. All data are in the form of one-look resolution complex scattering matrix and each pixel represents 6.66 m in slant range and 3 m in azimuth on the ground. The test-site selected is a Forest (Les Landes) in south-western France (near Bordeaux). It is almost totally formed of maritime pine (pinus pinaster) and consists of a section of forest where trees are of different ages and number.

For the experimental results, we have selected five regions of trees which differ by their ages (from one to 46 year old) and clear cut region.

Fig. 2 represents the HH image of the test-site at L-band with a look direction parallel to the row direction.

Fig. 3 shows an idealized configuration (there exists some trees in each area which does not meet the age range).

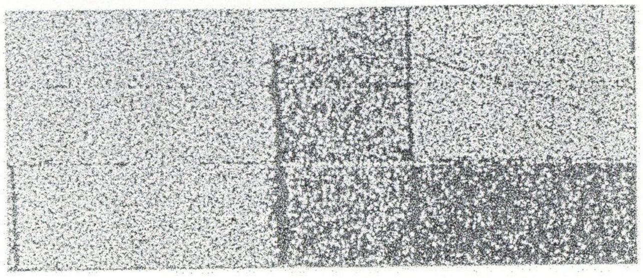


Fig. 2. Image HH at L-band

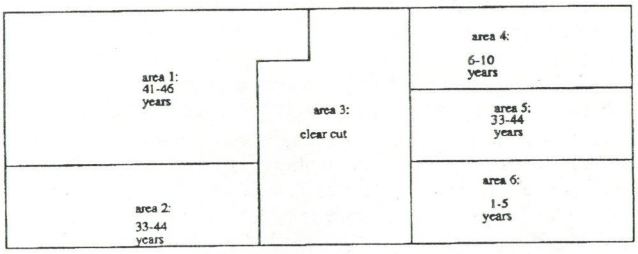


Fig. 3. Real configuration of the areas test

5.2. Segmentation using the Gauss model with one polarimetric component

To compute the characteristic vector Xc, we use a window of size 13 by 10 pixels (after some experimental trials). The segmentation result HH component (respectively VV) is shown in Fig. 4 (respectively Fig. 5) and Table 1 (respectively Table 2) summarizes the percentage of good classification.

Table 1. Numerical result of segmentation

	area 1: 41-46 years	area 2: 33-44 years	area3: clear cut	area 4: 6-10 years	area 5: 33-44 years	area 6: 1-5 years
VV	75%	39%	70%	29%	38%	86%

Table 2. Numerical result of segmentation using the HH component

	area 1: 41-46 years	area 2: 33-44 years	area3: clear cut	area 4: 6-10 years	area 5: 33-44 years	area 6: 1-5 years
HH	82%	38%	73%	6%	80%	84%

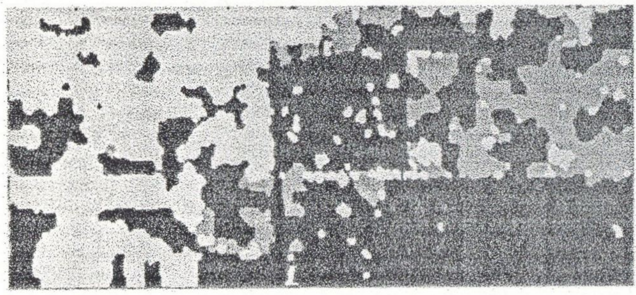


Fig. 4. Segmented image using the VV component

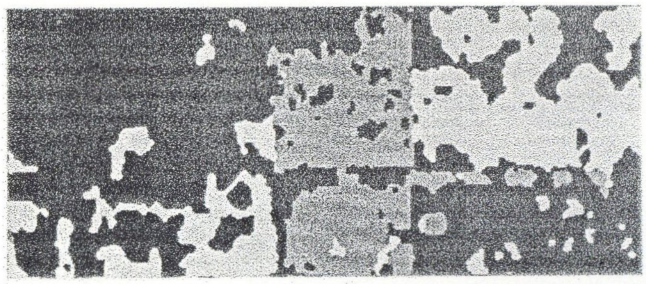


Fig. 5. Segmented image using the HH component

5.3. Segmentation using the three polarimetric components HH, HV, VV

Fig. 6 represents the polarimetric segmentation of image. Percentage of correct segmentation is presented in Table 3.

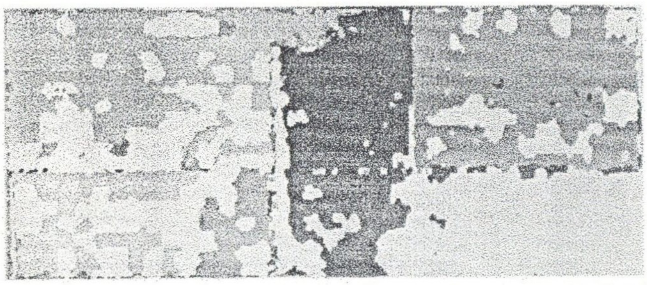


Fig. 6. Segmented image using the full polarimetric components

Table 3. Numerical result of segmentation using three polarimetric components

	area 1: 41-46 years	area 2: 33-44 years	area3: clear cut	area 4: 6-10 years	area 5: 33-44 years	area 6: 1-5 years
HH, HV, VV	75%	70%	82%	75%	62%	96%

For each area the latter classification exhibits good classification percentage which is always better than the worse result present in Table 1 or Table 2.

For area 2 and 4, the use of combined polarimetric data leads to considerable improvement.

No area present a misclassification percentage grater than 38 % when using the combined data.

The error dispersion matrix indicates that the misclassification rate increases when two conditions are met:

- neighboring areas
- overlap of age ranges as in area 1 and area 2.

Table 4. Errors disperation represented in the matrix (in percentage)

	area 1: 41-46 years	area 2: 33-44 years	area3: clear cut	area 4: 6-10 years	area 5: 33-44 years	area 6: 1-5 years
zone 1	0	30	0	0	0	0
zone 2	25	0	0	0	0	0
zone 3	0	0	0	0	0	18
zone 4	0	0	0	0	36	0
zone 5	0	0	0	25	0	0
zone 6	0	0	4	0	0	0

5.3. Segmentation using the K-distribution with three polarimetric components

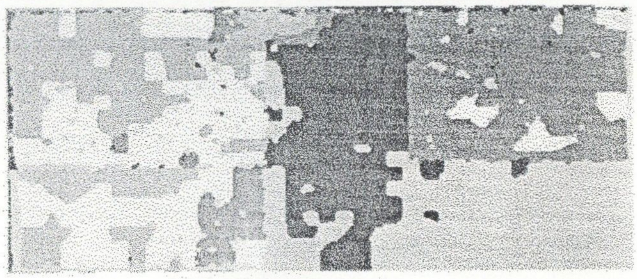


Fig. 7. Segmented image using the K-distribution.

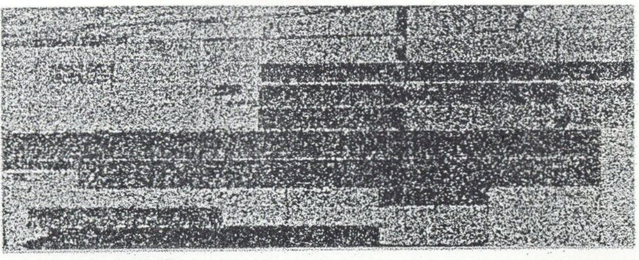


Fig. 8. Image HH in P-band

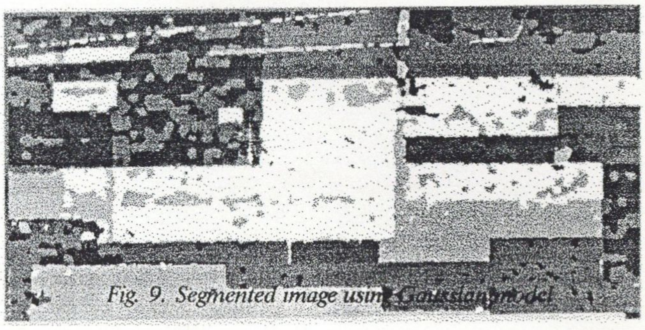


Fig. 9. Segmented image using Gaussian model

Results obtained with K-distribution are close to those given with the Gaussian distribution, but the segmentation with K-distribution gives better representation of real terrain configuration. For instance, little details (roads, rivers,. . .) are better preserved (see Figs. 8, 9, 10 for comparison).

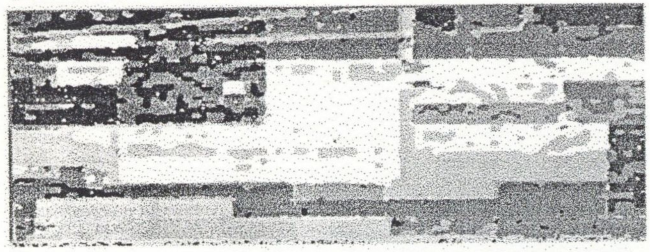


Fig. 10. Segmented image using a K-distribution model

6. CONCLUSION

The segmentation of forest data into areas of different ages is a difficult problem and the choice of the characteristic vector Xc and the statistical model for the data must be accurate.

Parameters estimation is based on FCM algorithm, which is an efficient and fast method.

Forest data are better represented by the K-distribution model than with a Gaussian model, even if increasing the computing time.

To improve the convergence rate of the ICM method, we use stability table calculated with Markov probability, in order to only treat boundary pixels.

We have shown that, the use of fully polarimetric SAR data increases the segmentation results.

Result obtained by ICM algorithm with a K-distribution model is quite satisfactory.

REFERENCES

[1] J. C. Bezdek, R. Ehrlich, W. Full: "FCM: the fuzzy *c* means clustering algorithm", *Comp. Geosci.*, Vol. 10, pp. 191-203, 1984.

[2] E. Rignot, R. Chellappa, P. Dubois: "Unsupervized Segmentation of Polarimetric SAR data Using the Covariance Matrix", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 30, No. 4, July 1992.

[3] E. Rignot, R. Chellappa: "Segmentation of Polarimetric Synthetic Aperture Radar Data", *IEEE Transactions On Image Processing*, Vol. 1, No. 3, July 1992.

[4] B. S. Manjunath et al.: "Unsupervized texture segmentation using Markov random field models", *IEEE Trans. PAMI*, Vol. 13, No. 5, 1991.

[5] J. K. Jao: "Amplitude Distribution of Composite Terrain Radar Clutter and the K-Distribution", *IEEE Transactions on Antennas and Propagation*, Vol. AP-32, No. 10. October 1980.

[6] F. Preteux: "L'approche Markovienne en analyse d'image et en reconnaissance de forme", *Document TELECOM Paris. Dept IMAGES*, Avril 1991.

[7] S. H. Yueh, J. A. Jao, R. T. Shin, J. A. Kong: "K-Distribution and Polarimetric Radar Clutter", *Journal of Electromagnetic Waves and Applications*, Vol. 3, No. 1, pp 73-89. 1987.

[8] Z. Belhadj, A. Saad, S. El Assad, J. Saillard, D. Barba: "Comparative Study Of Some Algorithms Of Terrain Classification Using SAR Images", *ICASSP'94: International Conference On Acoustics Speech & Signal Processing*

[9] Gath and Geva: "Unsupervized optimal fuzzy clustering", *IEEE Trans. PAMI*, Vol. 11, No. 7, July 1989.

[10] J. Besag: "One the statistical analysis of dirty pictures", *J. Royal Statist. Soc.*, Vol. 48, Serie B, No. 3, pp. 259-302. 1986.

[11] L. M. Novak, M. B. Sechtin, M. J. Cadulla: "Studies on targets detection algorithms which use polarimetric radar data", *IEEE Trans. Aerospace and Electronic Systems*, AES-25(2), 1989.

CONTEXTUAL ANALYSIS AND CLASSIFICATION OF DIGITAL IMAGES SENSED BY MICROWAVE REMOTE SENSORS (SLAR, SAR)

T. DÓKA and S. MIHÁLY

TECHNICAL UNIVERSITY OF BUDAPEST
DEPARTMENT OF MICROWAVE TELECOMMUNICATIONS
GOLDMANN TÉR 3. H-1111 BUDAPEST
HUNGARY

1. INTRODUCTION

Imaging radar is an active electromagnetic wave sensor which operates by radiating electromagnetic energy in the microwave part of the EM spectrum and sensing the presence and character of the returned signal from the earth surface. The echoes from a target surface detected by the receiver of the radar will undergo signal processing procedures and form a digital image in which the pixel values correspond to the backscattering properties of the earth surface.

In practice the well-known imaging radar is the Synthetic Aperture Radar (SAR) which was developed on the base of experiences of improvement the azimuth resolution of the Side Looking Airborne Radar (SLAR). Examples for these radar systems: The European Remote Sensing Satellite (ERS-1), the SLAR developed by the Remote Sensing Group at the Department of Microwave Telecomm. A radar imaging system compared with optical imaging sensors has the main advantage that the imaging capability is independent of time and weather conditions due to the use of microwave frequencies.

Although the radar images and the optical images may contain useful information the quality of a radar image should not be analysed by how it compares to an optical image. The main point is that how faithfully it represents the spatial distribution of microwave reflectivity of surface. Moreover, the radar images may contain unique information about target features revealed by image processing techniques and which are not obtainable otherwise.

The purpose of image processing techniques (i.e. noise filtering, edge enhancement, segmentation procedures), is to transform the digital radar image to a format which retains the main features and character of a target but it is readily understandable by human observers.

2. THE PROCESSING ENVIRONMENT

The image processing procedures were executed with the help of The Integrated Land and Water Information System (ILWIS) software package which has the facility using and running of user defined processing tools. The original raw SLAR images were created by The Remote Sensing Group at the Technical University of Budapest, the raw ERS-1 SAR image was available on CD by courtesy of European Space Agency.

3. SPECKLE IN RADAR IMAGES

The radar generates images by coherent processing of scattered signals from the tiny surface structure. The effect of the random fluctuations in the return signal observed from a surface area represented by one pixel produces a speckle on the image, i.e. there are variations in the gray-level between adjacent resolution cells that is the image has a specific granular appearance.

The presence of speckle largely reduces the detectibility of objects and observability of fine details. Fig. 1 shows a typical speckle pattern on ERS-1 SAR image.

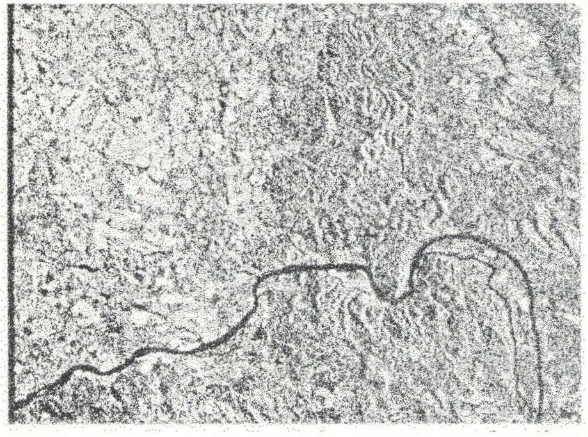


Fig. 1. ERS-1 SAR image over the river Danube

4. SPECKLE SMOOTHING TECHNIQUES

Numerous speckle suppression techniques have been proposed. These procedures basically fall into two categories. In the first category improving the appearance is performed by averaging several sample collected from the same part of the target area. This early method of speckle reduction can be executed on board of the aircraft. The disadvantage of this technique is the loss of spatial resolution and that the speckle noise is not completely suppressed. In Fig. 2 a SLAR image is shown. In this case eighth independent samples have been averaged for producing a pixel. The degradation resolution can be observed because of extension of composing of bright point like targets.

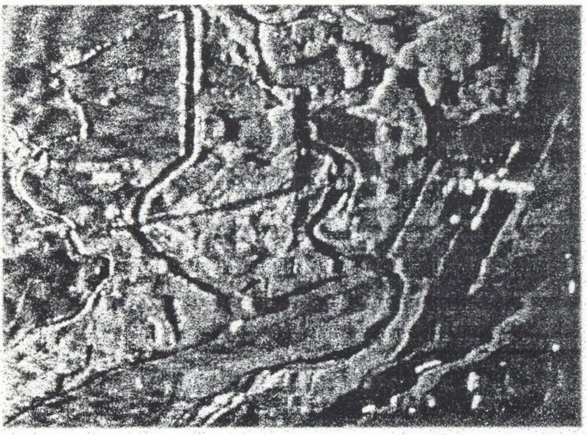


Fig. 2. SLAR image over the river Danube

The another procedure for smoothing the speckle noise is the filtering technique. Filtering is a process in which each pixel intensity is replaced with a new value with applying a certain function to the pixel and its neighbours. In Fig. 3 the result of median filtering is given which is a simple method but it has the advantage that it retains the character of step and slope transition between adjacent pixels and suppresses the spiky noise.

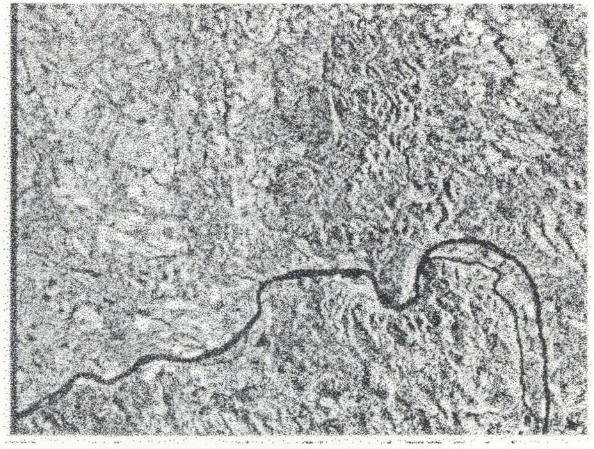


Fig. 3. A median filtered ERS-1 image

5. EDGE ENHANCEMENT

The significance of the edges, one of the simple features of an image, is that an image with over-enhanced edges produces better feeling to the human analyst, and the edges are widely used to outline the boundaries of objects for classification and analysis. For example in case of homogeneous areas such as agricultural fields it is necessary to preserve the average brightness values within, and maintain sharp edges between adjacent fields. Fig. 4 shows the effect of edge enhancement filter applied on the raw radar image.

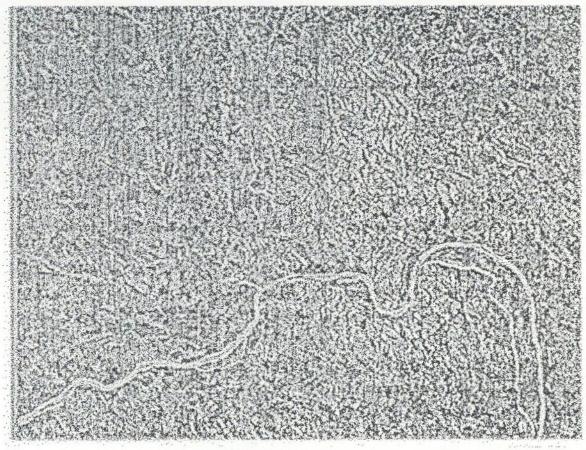


Fig. 4. Edge enhancement on ERS-1 image

6. ASPECTS OF IMAGE INTERPRETATION

It is desirable to reduce speckle noise in radar images to facilitate interpretation and to preprocess images for further classification. It has been accepted that some filters produce good performance in smoothing while others work well at the vicinity of edges. The outcomes of these procedures are shown in Fig. 5 and 6 successively. Fig. 5 is abundant in particulars while on the Fig. 6 the linear structures, like rivers, ridges, field boundaries can be observed better.

The classification and the clustering are parts of interpretation. Classification means the sorting of the image features into groups. The separation based on creation of a sample set on image with certain, well-recognizable image segments which have typical brightness distribution. In the case of Fig. 7 four samples have been chosen: river, forest, agricultural field, artifical object and the classification was a built-in ILWIS algorithm.

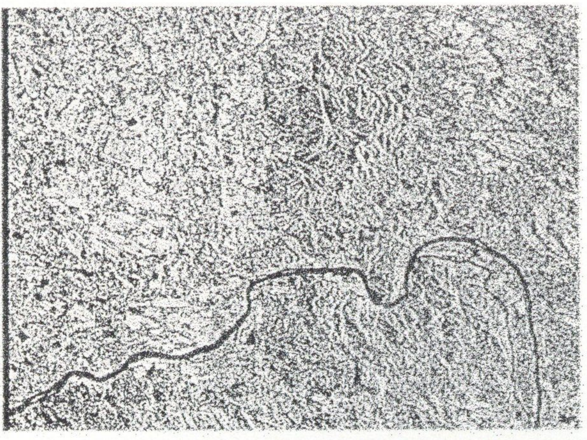


Fig. 5. Image with numerous well-seen particulars

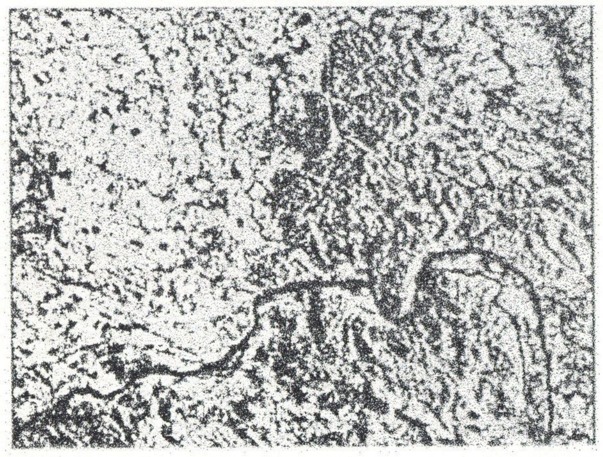


Fig. 6. ERS-1 image with enhanced rivers, ridges

The clustering is a kind of reverse classification. In this case the aim of the procedure is to find groups to the existed image elements. In clustering such segments belong to one group which are comparable with others well and there is significant difference to another ones. The effect of a built-in ILWIS clustering procedure is given on SLAR image in Fig. 8. It can be said that this result approaches better the original stage then the previous one.

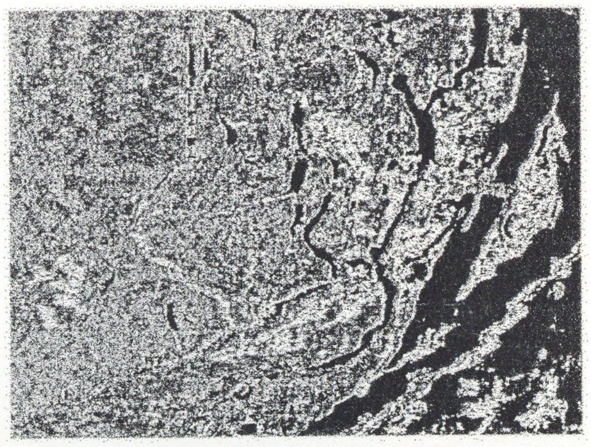


Fig. 7. Classification of SLAR image

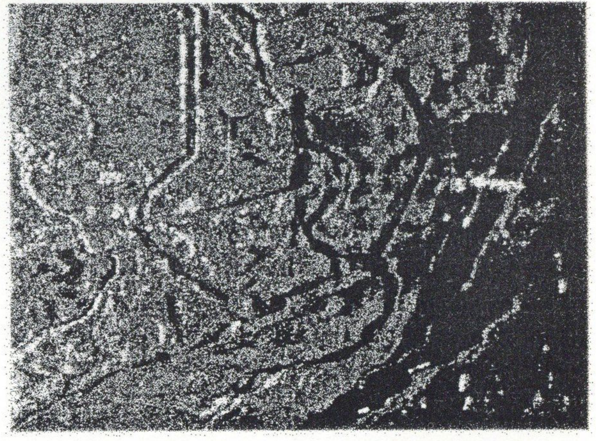


Fig. 8. Clustering of SLAR image

7. CONCLUSION

In this study some well-known image formation procedures have been applied to raw radar images. The outcomes of the analysis show that the different methods make the original (raw) image data readily understable by human observer. In case of after-image-formation procedures it has to make an effort to use algorithms (or series of algorithms) which produce a final result with suppressed noise effects and retained structural and textural features.

REFERENCES

[1] J. W. Goodman: "Some fundamental properties of speckle", *J. Opt. Soc. Am.*, Vol. 66, No. 11, pp. 1145-1150 (1976).

[2] J. S. Lee: "Speckle suppression and analysis for synthetic aperture radar images", *Optical Engineering*, Vol. 25, No. 5, pp. 643-645 (1986).

[3] F. T. Ulaby et al.: "Microwave Remote Sensing, Active and Passive", Vol. II, Addison-Wesley P. Co., 1982. pp. 583-595.

[4] ILWIS 1.3 User's Manual Volume 1-2 (1992).

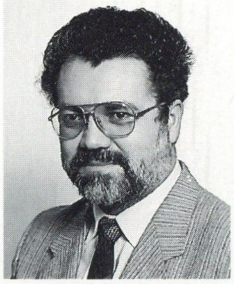
Antenna Hungária Hungarian Radiocommunications Corporation is the most significant company in the field of radio communication and broadcasting and among the first in telecommunication in Hungary.

It is engaged in the following principal activities and services:

- broadcasting of radio and television programmes
- relaying/transmission of programme signals both domestically and internationally
- conducting on-the-spot television broadcasts
- data transmission and radiogram services
- production, installation and maintenance of telecommunication equipment
- research, experimental development and consulting.

One of the aims of Antenna Hungária is to initiate the introduction of future oriented radio communication services in Hungary. It was the first in the region to join the satellite-based telecommunication and location system EUTELTRACS. Negotiations are under way to build and launch a Hungarian satellite programme called MAGYARSAT, which should be operational by 1996, the year of the Budapest World Exhibition. The company offers MMDS service in Budapest providing hundreds of thousands of Hungarian households with a rich choice of Hungarian and foreign made satellite programmes. TELEDATACAST transmissions provide widespread use of instant economic information wherever Hungarian Television broadcasts can be received. Antenna Hungária is a member of the international consortium Hungaro DigiTel. Based in Budapest, Hungaro DigiTel provides data and sound transmission services using the latest technologies. High quality service is guaranteed via the VSAT satellite communications network which can be hooked up with any public telecommunication network on the ground.

The chief advantages of the VSAT network are its great reliability, outstanding performance and ease of setting up the system. Presently under construction in Budapest is an uplink station with a dish of 8,1 meters diameter, which makes possible for the neighbouring countries' VSAT system to link up with the satellite.

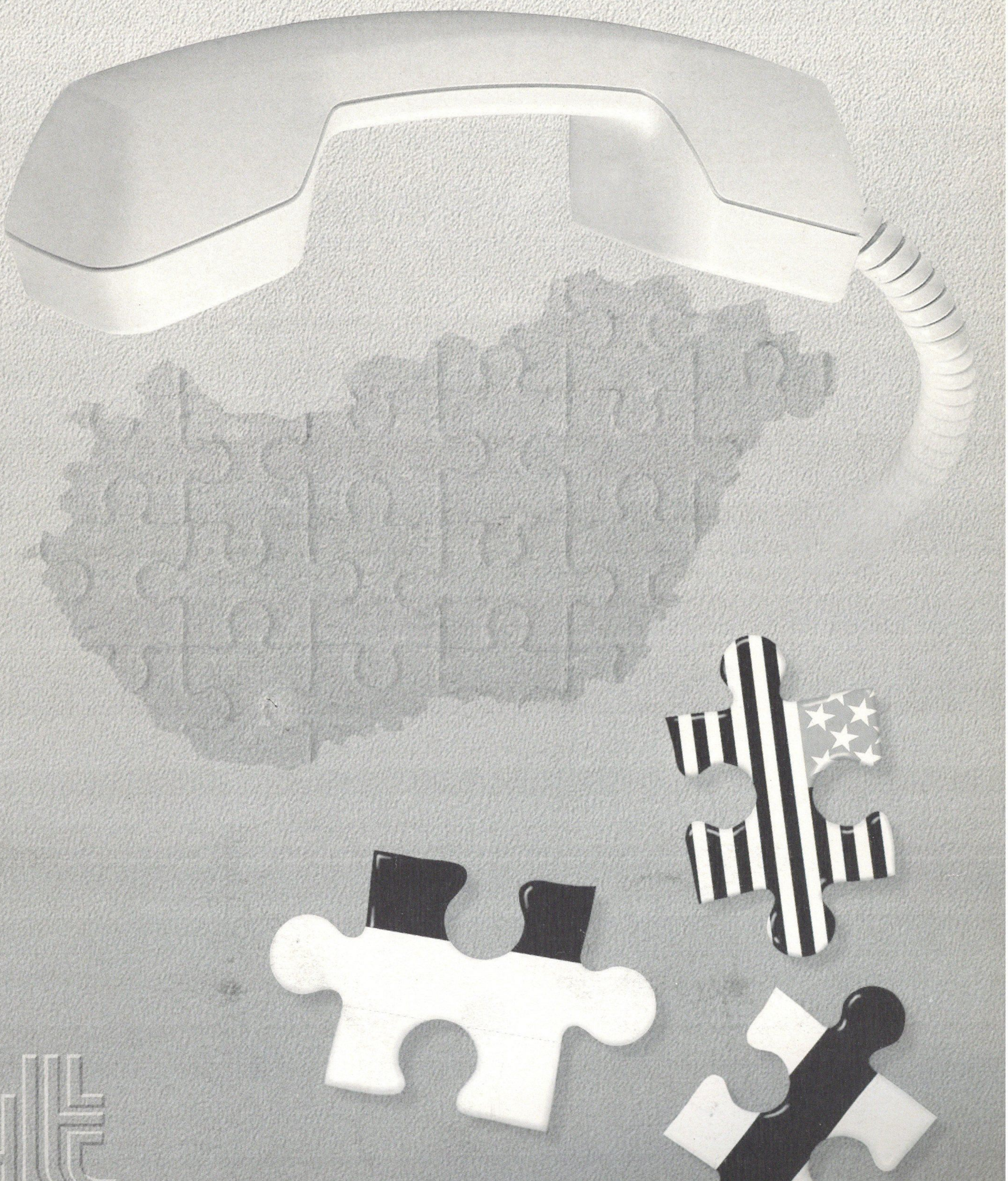


József Bartha
Director General

Should you require further information, please do not hesitate to contact us at:

Antenna Hungária
H-1115 Hungary, Budapest, Petzval J. u. 31-33
Phone: (361) 209-0379 Fax: (361) 209-0378

Shared Vision for Hungary...



MATÁV

Hungarian Telecommunications Company Limited

IMAGE COMPRESSION, CODING II.

REMOTE SENSING