Characterizing Video Coding Computing in Conference Systems

By G. Tuquerres (tuquerre@cs.utwente.nl)
Telematics Systems and Services – Twente University

Abstract
In this paper, a number of coding operations is provided for computing continuous data streams, in particular, video streams. A coding capability of the operations is expressed by a pyramidal structure in which coding processes and requirements of a distributed information system are represented. The application of the pyramidal structure is provided in terms of a sequential algorithm that analyze, convert, normalize and structure the encoded media message. A class of source coding techniques that is determined supports the sequential algorithm. The use of the class is provided in terms of supporting mechanisms and coding operations for a video conferencing system using multicasting communication.

1 Introduction

This paper concerns computing codes for data streams, in particular, codes for continuous data streams, which are important for implementing distributed multimedia systems. Needed are parameters of a coding format and the format at run time, which should not be static.

Our goal is to describe techniques that are able to specify decoder systems in the internetworking domain. Such a formulation allows a mapping solution for features of an application into network constraints. For instance, the proposed specification can support prototypes of heterogeneous video decoder systems for interactive video applications in wireless networks. However, a dynamical structure of the coding capability for Distributed Video systems is needed.

The large diversity of standardized media coding methods, in which some aspects of their functions often overlap, does not enhance an effective set of coding functions. A classification of coding methods has also to account for operations on continuous data stream like video streams. Difficulties that hamper the appropriate coding class are lack of standardization of encoding procedures and lack of a format for changing parameters during the transfer. This paper addresses the last problem.

The traditional classification considers only the aspects determined by the source coding of a data stream; the considered coding features are data compaction and data compression. The data compaction feature by its efficiency allows error-free data recovering. The data compression feature represents data even more concisely as compared to data compaction but it does so at the cost of clean recovery of the data. Data compression accepts some intentional distortion for coding. Data compression should discern the actual data and the environment of the data. It is the environmental data that are not included. Inclusion of environment at coding aspects urges for other classifications [1] and [2]. In [1], a set of coding techniques for storage and communication applications in media computing is described. Although this set supports design and implementation of applications, it is unable to specify parameters for systems to request symmetric processes such as a real-time decoding process and a real-time encoding process. In [2], an overview of source coding for the existing multimedia applications is presented, which identifies sets of parameters for symmetric processes. However these sets do not support computing coding functions.

A classification of coding methods and a set of coding processes that support distributed applications is missing in the field of computing codes for continuous media streams. The coding class contains levels of computing codes for entities of a distributed system. The class characterizes manipulation operations for the application context as defined by JPEG, M-JPEG, H.261 and MPEG-1 and –2 formats for video streaming and multicasting communication capability. This class is meant as a guideline to create order. A set of this class includes coding operations be applied the video streams sequentially. The application is meant as a guideline to use code for a continuous media stream. The listed coding operations do not include completeness nor includes all coding methods for converting video data stream. However a distinction can still be made between coding processes that characterize manipulation operations and coding processes that define the coding class. For the last coding processes, tables containing operations and parameters of generic supporting mechanisms are presented (tables in Appendix A). These supporting mechanisms are the basis for the design of video distributed systems. While for the transcoding capability, a table including operations for conversion processes are also presented (table 1.). These operations are essential for the design of video information systems.

The paper's organization is: Section 2 addresses the definition of a coding domain and its features, Section 3 contains the coding aspects for continuous data stream in such a way that they support the coding capability. Section 4 demonstrates how to identify the coding processes for a video conference system using the coding domain. Section 5 surveys on further research directions and open issues concerning computing codes for continuous data streams, especially for video streams.

2. Coding Domain

In this section, the introduction to a structure of coding capability for distributed video systems is considered. The structure determines a dynamical system as defined in Applied System Analysis [3]. The definition accounts for how signals will evolve over time. This definition is a triple -tuple: time set, signal space, and collection of signals. The last parameter determines the behavior of the system. The continuous data streams, especially video streams, are assumed to be the signals. In the paper, the time set and the behavior of the system are not considered.

The identification of space of the continuous data streams is presented in subsection 2.1. The support of the space in terms of the coding capability requested by distributed information systems is described in subsection 2.2. While the support due to aspects of the coding operations on the continuous data stream is explained in subsection 2.3. The effectiveness of the space is discussed in the subsection 2.4.

2.1. Pyramidal Structure for Coding Capability

The definition of the continuous data stream space used here is an orthogonal pyramid by which the features of the coding domain can be described. This definition is applicable for the design of systems in which continuous data streams are manipulated on the fly. For instance, the specification of a video message can be employed in multimedia systems by streaming the message using heterogeneous terminals. This specification accounts for different video formats along paths between the end and the intermediate points of the system.

This pyramidal structure allows an adaptive operation that converts a specific format, in which the media is encoded, into other format. The last format is a set of codes that represents only a part of the proposed structure. Concise functions, that yield a rich result of descriptions in the coding of continuous data streams, are made possible by such a set.

Figure 1 shows the pyramidal structure and its meaning. In the figure, the performed operation is represented on the lateral planes of the pyramid. The lateral planes are defined by coding levels and the effect of the restrictions inherent to distributed information systems. The coding levels represent the concise functions. The effect on the other hand represents the capability of the system for its presentation of the continuous data streams in the distributed system. For instance, dropping video frames can be present due to the streaming of video messages in a conferencing system.

In the planes of the pyramid, the features of the operation in terms of a set of coding functions, called a coding interface, are localized. The features define the coding methods for the data stream according to the required capability of the system. These planes can support system designers with a tool for the specification of internetworking applications. This tool specification can also be useful as a prototype system.

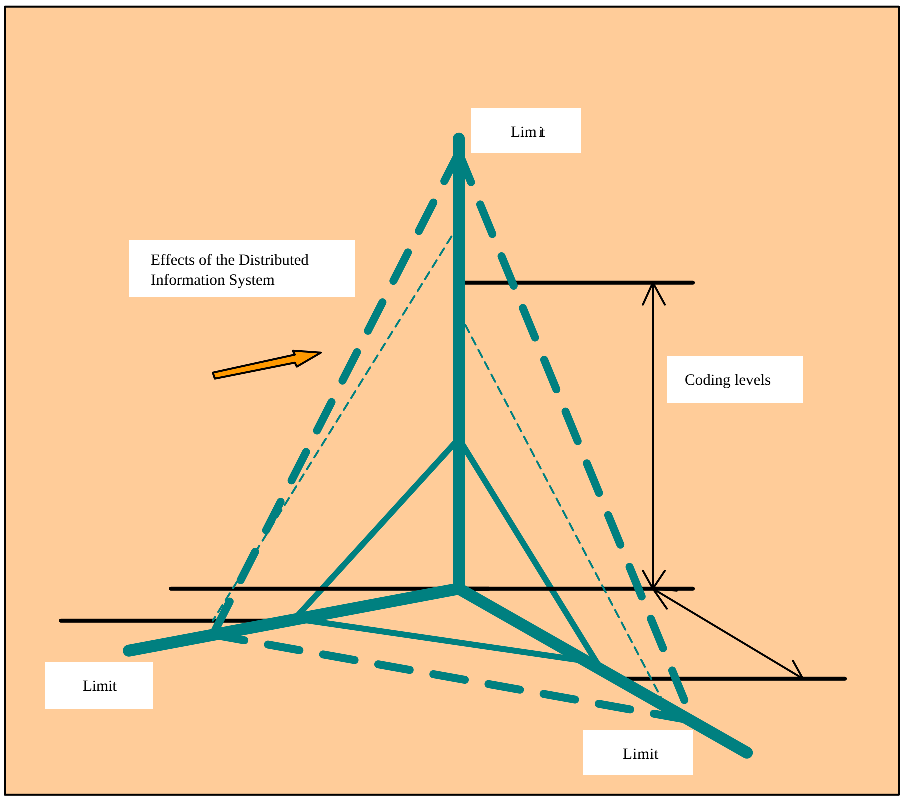

Figure 1. Dynamical Structure for supporting the Conversion of Video Streams

The axes of the pyramid are used for the quantitative levels of features of the coding operation. They will yield the results of a set of operations of a supporting mechanism involved. The supporting mechanism is always a coding system. The results are expressed in bits per sample since they are coding capability values.

The singularities of the operation are determined in the vertices of the pyramid. These vertices identify the limits the processing of continuous data streams imposed on the system by Information Theory and by the acceptance of the human perceptual system. For instance, a 4.5 quality at rates of 0.25 bit per sample is the constraint for processing of images and video signals reported in [4]. The quality is measured on a 5-point value scale, which is an accepted subjective measure for audio-visual communications [4].

2.2. Requirements of the Existing Distributed Video Systems

The communication requirements of the existing distributed video systems are given in measurable terms. The description is in terms of metrics to support the features of the coding domain. It presents subjective and objective metrics such a person-computer interaction or a frame rate.

A representation of the video message and the heterogeneity of the systems for transfer video data streams concern features of the existing distributed video systems. The features refer to requirements of network resources, definition of interfaces between the operating system and the existing network devices and profiles based in user preferences. For instance, in [11], the distributed multimedia system is presented as a set of enabling technologies that provides mainly communications and data compression processes to support Interactive TV, Telecooperation and Hypermedia applications. It is reported that these applications require very high transmission rate and stringent QoS guarantees, multicast communication, and support for multi-service. Requirements of friendly user interfaces and powerful presentation tools are also reported. In [12], an operating system for adaptive mobile data access that supports web browser, a video player and a speech recognition system is characterized in terms of the fidelity (a measure of data quality). This demonstrates that the collaboration between the operating system and the applications can accept other interface specifications, in addition to the currently inviolable interface between wireless devices and mobile software. In [5], a prototype system that includes mobile users in wireless internetworking environment was evaluated using video conference and remote surveillance functions. The performance of the prototype concerning video coding functions (H.263 coder) result in demands of user-selectable profiles for different network types. Therefore the features of the system distributed allow assume that objective metrics concern to the media message representation and the heterogeneity of the systems while subjective metrics only refer to heterogeneity of the systems, especially the graphic user interface systems.

The media message representations are expressed in values of spatio-temporal resolution or frame rate. These values are satisfied by operations that reconstruct and process the media. For instance, the coding algorithms of audio-video information are expressed by values such as 800 x 600 pixels and a color depth up to 16.8 mln at 15 frames/second for video in a conference, and 1.5 bits per sample and sampling rate of 44.1 kHz for CD-audio.

Furthermore requirements of the users and the underlying technology are expressed in different metrics. The user-oriented requirements are expressed in terms of a subjective metric quality. This identifies measurements concerning interactions between three actors: a user of a networked computer, the networked computer and a physical world. The interactions appear differently: person-to-person, person-to-computer, people-to-physical world. The system-oriented requirements on the other hand are expressed in terms of an objective metric quality. This defines a required capability of the application supported by the Distributed Video system. This capability can be streaming, unicasting, multicasting, storage, etc.

These objective and subjective metrics determine the coding domain to support an adaptive operation on the media message. The adaptive operation converts a media encoded in a specific format into another format. The adaptive operation modifies the requested spatio-temporal resolution or frame rate, the interaction of the system and the environment, and the capability that support the interaction. Moreover this characterization can allow designers of distributed systems to specify video systems more easily.

2.3. Coding Aspects of the Domain

This subsection introduces coding aspects of the continuous data streams that can be measured. These coding aspects are in terms of metrics such as JDN profile[1]. Furthermore an algorithm that identifies the coding domain is also introduced.

The algorithms of media processing identify coding aspects. The design of the processing algorithms uses metrics that the insensitivity property of the human visual and audio system has defined. This allows the coding domain be expressed in scales or profiles. Similarly the effective application of these algorithms depends on not only the nature of the media but also the system in which the media processed is transmitted. This identifies a generic coding process when the processing algorithms are applied in a networked computer system.

This generic coding process consists of:

A decomposition of the media,
A definition of the model of the decomposition
A comparison of the defined model and the decomposed media
An encoding of the comparison result

[1] Profile of signal levels that provide just noticeable distortion in the image and video signals

This structure allows identifying components of the coding domain.

Due to the insensitivity property of the human visual system, the processing algorithms identify coding aspects concerning the accepted lossy representation of the information. These coding aspects are expressed in values of metrics such as the Mean-opinion-score (MOS) scale[2], impairment scale[3] or the Just-noticeable-distortion (JND) profile. The algorithms based in JND profiles can represent the image with good quality by using less bits/pixel. In contrast, other image coding algorithms based only in MOS or impairment scale represent the image with the same quality by using more bits/pixel depending on the application that requests the image. For instance, the image quantization algorithms that use the JND profile to hide the quantization noise under the JND profile [2] improve the coding rate. For a black-and-white still image the use of the JDN-based algorithms allows coding the image with 2 bits/sample (or 2 bits/pixel). On the other hand, the same distortion in the image using noiseless coding methods such as Huffman coding is achieved with 5 bits/sample. Using only simple linear quantization, the same quality is achieved with more that 8 bits/pixel.

Due to the application of media processing algorithms in a transfer system, the coding aspects can be expressed in terms of the components of the generic process. Coding components includes decomposition of the media, definition parameters of the media modeling, and encoding the modeled media data.
For ISDN video conferencing, the coding components allow the video to be decomposed in set of frames by the filtering algorithms of the capture device. The first frame is encoded using a DCT on block of size 8 x 8 pixels. For the other frames the difference between the current frame and its previous frame is encoded by using motion compensation techniques. These techniques allow the displacement of groups of pixels from their position in the previous frame (as represented by so-called motion vectors) be transmitted together with the DCT coded difference between the predicted and original images.

The coding domain can be parameterized in terms that are measurable. Indeed this allows determination of features of the adaptive operation that converts a media encoded in a specific format into other format. The features are values of metrics due to processing of the media in transmission context as well as the application of the algorithm that decompose, define, compare and encode the media to be transfer sequentially.

2.4 Limitations of a Sequential Coding in Distributed Systems

This subsection introduces the effectiveness of encoding in the defined space. The effect of the requirements into operations of the encoding process is presented. This requests the redefinition of operations for encodin g.

The encoding process currently is the application of a set of sequential operations for adapting a continuous data stream. The operations satisfy only system-oriented or they have been adapted to support a specific application in distributed environment. In [1], for instance, techniques that applies source, entropy and hybrid coding principles are presented. They adapt a discrete or continuous media message in the scope of multimedia systems. Although the techniques are applied sequentially, they only satisfy the system-oriented requirements. The fulfillment of system-oriented requirements is also expressed in [11] in which the data compression techniques are tried as an enabling technology of distributed multimedia systems. We then consider that the adapting operations of a data stream must include the requirements identified in Section 2.2.

The coding operations into the characterized space permit to have real-time encoding processes. The current sequential coding applied for distributed system provides a no-real-time encoding process and supports a real-time decoding process. But the existing systems often request real-time encoders and decoders. For instance a video chat on the web requires encoding and decoding processes in real time to

[2] A subjective 5-point scale that is an average over signals inputs, subjects and other parameters of relevance
[3] A subjective 5-point scale that represents quality of the signal in terms of impairments

enable Internet sites to embed live, interactive audio, video and text chat. Moreover these requirements are requested to enable the video chat using wireless networks.

Furthermore the pyramidal structure allows the sequential encoding to convert homogeneous and heterogeneous media messages. The encoding process supports an operation that can convert the type of the media message not only into the same type but also into a different type. Conversion operations for inter-type messages are less complex and require less processing time to carry out operation decisions. Simulations have shown that conversion operations of MPEG-2 into H.261 base on modifications of GOP size require less processing time to carry out motion estimation and macroblock decisions [13]. The simulations also show that this way of conversion is much simpler that an off-the-shelf codec. The conversion includes dropping of frames, DCT-coefficients values or entropy codes. Therefore a control on the behavior of the coding domain is possible.

3. Characterizing the Coding Domain

In this section, the operations that support the coding capability are presented. The operations follows the generic algorithm introduced in Section 2.3. Moreover the coding mechanisms that support these operations are introduced.

3.1. Coding Capability

The coding domain supports the conversion of media messages in which MOS scale spatio-temporal resolution and MOS scale frame rate are modified, which also holds for Impairment scale or for JDN profile. The modification must comply with the generic algorithm of media processing. This conversion implies that the parameters of the media modeling and of the encoded media data are modified in order to satisfy the user-oriented and system-oriented requirements of the distributed system.

3.2. Providing Coding

The coding capability that allows obtaining a media message is provided by a set of operations. This set of operations is imposed onto the media message at end-terminals of the communication network. In the terminals, the transcoding capability represents executable operations in a sequential way to complain the dynamical structure of coding domain.

Due to the dynamical property of the coding capability, It is assumed that the operations are performed in four steps. These sequential steps form what is named set of coding interfaces: preparing, transforming, quantizing and coding. These coding interfaces transcode the media message so that the dynamical structure of the coding domain can be applied in a way that coding operations at:

The preparing interface analyzes (organizes) the message being transformed.
The transforming interface converts the type of the message.
The quantizing interface normalizes (maps) the transformed message into the coding levels of the domain.
The coding interface structures the message.

3.3. Supports of Coding

A coding interface refers to processes and parameters of coding source techniques. These techniques sequentially compressed the continuous media data at source. Using the generic algorithm, introduced in Section 2.3, the techniques encode continuous media data that is transferring. The encoding process involves the application of coding source process using different parameters.

The processes and parameters as a consequence support operations in the coding domain in a sequential way so transformation of media messages become possible. This allows the media messages to be analyzed, converted, mapped and structured. The manner in which a generic set of processes behaves will be explained below. This includes the supporting processes for each coding interface that was identified in the literature on data compression for audio and video [5, 6, 7 and 8]. The supporting processes determined during the survey of audio and video techniques are presented in Appendix A. They are expressed as coding mechanisms with operations and parameters.

The supporting encoding processes organize the message into two separate phases: a priori phase and a preparation phase. The a priori process identifies the coding functions of the media message to be transcoded. Subsequently the preparation process uses the identified functions to modify the media message. The supporting techniques of these processes are identified in the literature by: filters, analyzers, synthesizers and ordering (scanning) techniques. Analyzers and ordering techniques characterize the a priori phase while the filters and synthesizer identify the preparation phase. The operations and parameters of both types of processes are presented in table 1 of Appendix A. These processes are supportive for coding operations on audio and video streams.

Other processes for supporting encoding also convert the organized message into a message of the same type or into one for a different type. These processes are supported by transformer and predictor techniques. Transformer techniques allow a conversion of the coded message into another representation domain by transform operations without loss of its information. The conversion, in most of the cases, includes concentration of information into specific components of its domain. Predictor techniques, on the other hand, determine a new value for a data component of the coded message from values of its neighboring data components. The predicted values form a new representation of the media message that also concentrates information into specific components. The operation and parameters for the conversion processes are presented in table 2 of Appendix A.

Then other supporting processes normalize the converted message into a message determined by the coding levels of the coding domain. The supporting techniques of these processes are known in the literature as quantizers, samplers and compressors. Quantizers map the components of the converted message into components of the coding domain by scalar and vector operations. Samp lers also map the message components into coding domain components using decimation operations. On the other hand, the compressors normalize the components of the message to components of the coding domain by assigning codes to the components. The operations and parameters for the mapping processes are presented in table 3 of Appendix A.

Finally another supporting processes will structure the media message for transmission. The techniques that support these processes are well established like transmitters, translator and mixers. Transmitters determine the way the message is formed to send on to the network. They support the different ways of transmission but do not take in consideration any network bandwidth. Mixers and translators, however, determine the structure of the media message as constraint by the network bandwidth. They support only pre-established ways of transmission. The operations and parameters for the transmitters are presented in table 4 of Appendix A.

3.4. Effectiveness of the Coding Operations

The coding capability improves the processing power for coding of media messages into distributed applications. It can be supported either in the end terminals or in the networking terminals in which the average total processing can be reduced around 25 % respect to the processing using off-the-self encoders and decoders [13, 14]. The clear disadvantage is an increase of the average transmission time during a session. However this delay can only affect the real time requirement of some applications such as the interactive video applications.

4. Applying the Coding Domain

A simple example that describes the use of the coding capability is presented. The example is a conference system that demonstrates the application of the coding domain for transfer video stream in heterogeneous environment. For the conference system, the set of mechanisms supporting the coding interfaces is included

Depending on requirements of the conference system, a video message is converted. The system is able to deliver video for a group of end terminals such as desktops and laptops. These terminals request video messages in CIF and QCIF respectively. The system is also able to provide video streaming and multicasting communication for a video conference application. It is assumed that the users of the application request the video at different instance and they do not have random-access to the application.

The conference requirements establish the pyramidal structure of the coding domain. This pyramid supports the coding conversion of the video message by a set of mechanisms that supports the coding process for each coding interface. The coding process involves operations and parameters (tables in Appendix A). Moreover any point on the plane determined by the effects of the requirements represents a sub-pyramid. The coding capability of the sub-pyramid is expressed by the operations of the supporting mechanisms with different parameters.

The application of the generic coding process represents the pyramid for the conference system. Table 1 shows the set of coding processes. This table contains the supporting mechanisms and coding operations per each interface. The parameters of the operations can be determined from the tables of Appendix A.

INTERFACES	MECHANISMS	OPERATIONS
Preparing	*Raster scanning Shifting* *Sampling Filtering* *Analyzer*	Line-by-line, from left -to-right and from top-to-bottom Band-pass *Model-based Imaging*
Transforming	*Transformer* *Predictor*	*Discrete Cosine* *Motion compensation, Differential encoding, Loop filtering*
Quantizing	*Scalar Quantizers* *Adaptive quantizer*	*Uniform Forward, Uniform Backward ,Non -uniform midrise* *Vector Adaptive, Vector Multistage*
Coding	*Order* *Assignator* *Compressor* *Transmitter*	*Zig-zag ordering* *Run Length Coding* *Static Variable Length Coding, Dynamic Adaptive Coding* *Sequential Image, Progressive Image*

Table 1. Coding Processes for Video Conversion during a Conference System

The operations and parameters converted the scalable video message along the end-to-end path while the application demands are satisfied. At sending terminal, the video message that has scalable features is dispatched. At internetworking terminals, the video object is converted. The conversion follows the generic coding process introduced in section 2.3. Subsequently, the converted video object is rendered to the graphic user interface at the receiving terminal. Therefore during the coding process the sheer quality of the video data is conserved when the determined coding mechanisms are applied.

5. Conclusions and Further Research

The coding domain that accounts for the conversion of media messages is presented. It allows designing distributed video systems without a deep knowledge of video coding. It also supports implementing coding interfaces for distributed information systems using the established class of coding processes.

In the coding domain, a generic algorithm converts video streams. The algorithm dynamically allows coding operations to be applied a video stream in a sequential manner. The operations depend on the source coding techniques, features of the transport system and interactions of the application user in a distributed context.

Work in Progress

Behavior and a time set of the coding domain are determining to make complete the definition of the coding domain in distributed applications. For the time being, the characterization of the coding capability is expanded using mechanisms for channel coding such as error resilient based on feedback channel. Subsequently the requirements of the distributed Video systems are refined. The time set then will include how the continuous data streams, especially the video stream, evolve over time in the case the coding process is applied. Using source and channel coding mechanisms supports this coding process.

In Internet, applying the coding domain are optimized for delivery of video stream. This implies the identification of the mixer and translator mechanisms that support the coding interface. Constraints determined by the multicast communication supporting video delivery are also identified. The constraints are computing and communication resources in the Internet.

Future Work

The proposed solution against a coding method supported by MPEG-4 mechanisms will be compared. The comparison attempts to validate the efficiency of the supporting mechanisms of the determined coding interfaces. The operations for video transcoding are also validated in the comparison. The application context will be streaming of video using the Internet protocols.

A quantitative metric of the user-oriented requirements of video distributed systems will be assessed. It attempts to determine a model of random-requests toward video applications. The model will provide more solid values for the parameterization of the user-oriented requirements. This joins the required capability of the video distributed systems will improve the design of video applications in heterogeneous environments such as Internet, wireless or mobile environments.

References

1. Steinmetz, R. *"Data compression in multimedia computing –standards and systems"*. Multimedia Systems, Vol. 1, Pags: 187-204, 1994.
2. Cox, R., Haskell, B.G. Lecun, Y. Shahraray, B. and Rabiner, L. *"On the Applications of Multimedia Processing to Communications"*. Proceedings of the IEEE, Vol. 86, No.5, Pags: 755-824, May 1998.
3. Weiland, S. *"Applied System Analysis"*. Lecture Notes, Eindhoven University of Technology, September 1996.
4. Jayant, N.S. *"High Quality Networking of Audio-Visual Information"*. IEEE Communication Magazine, Pags: 84-95, September 1993.
5. Yeadon, N., Davies, N., and Blair, G., *"Supporting video in heterogeneous mobile environments"*. Lancaster University, www.lancs.ac.uk/computing/research/mobile/.
6. Jayant, N.S. and Noll, P. "Digital Coding of Waveforms: Principles and Applications to speech and video". Englewood Cliffs, NJ, Prentice-Hall, 1984.
7. Veldhuis, R. and Breeuwer M. *"An introduction to Source Coding"*. Prentice Hall (UK), 1993.
8. Sayood, K. *"Introduction to Data Compression"*. Morgan Kaufmann Publishers, Inc., 1996.
9. Haykin, S., *"Adaptive and learning systems for signal processing, communications, and Control"*, Wiley, 1995.
10. Dertouzos, M. L. *"The future of Computing"*. http://www.scientificamerican.com/1999/0899issue
11. Li, O.K.Victor, and Liao, Wanjiun. *"Distributed Multimedia Systems"*. Proceedings of the IEEE, Vol. 85, No.7, July 1997.
12. Noble B., *"System Support for Mobile, Adaptive Applications"*, IEEE Personal Communications, February 2000.
13. Shanableh, T., and Ghanbari, M. *"Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolution and Different Encoding Formats"*, IEEE Transactions on Multimedia, June 2000
14. Assuncao, P.A.A., Ghanbari, M., *"Transcoding of single-layer MPEG video into lower rates"*, IEE Proc, Vision, Image, and Signal Processing, vol 144, December 1997.

Appendix A: Supporting Coding Processes for Audio and Video Streams

Table 1. Supporting Coding Processes for organizing the streams of continuous data

MECHANISMS	FUNCTION	OPERATIONS		PARAMETERS
Filter	To separate symbol sequence into components according to a measure (frequency)	Low-pass High -pass		Cutoff Frequency
		Band-pass		Band-pass Frequencies
		Antialiasing		Half of the sampling frequency
		Finite Impulse Response		Number of taps
		Infinite Impulse Response		
Ordering (or scanning)	To make ordering of components	Line-by-line, from left -to-right and from top-to-bottom		Raster Scan
				Zig-Zag
Analyzer/Synthesizer	To model the media and send the model parameters/To synthesize the media based on the received model parameters	Speech	Channel Vocoder	Bank of band-pass filters (analysis filters)
			Linear Predictive Coder (LPC)	Linear Filter
			Code Excited Linear Prediction (CELP)	Linear Filter Codebook of excitation vectors Perceptual weighting filter
		Image	Fractal	Synthesizing model
			Model-based video	Global-local model

Table 2. Supporting Coding Processes for converting the organized streams of continuous data

MECHANISMS	FUNCTION	OPERATIONS			PARAMETERS
Predictor	To predict the value of the actual component from the value of the neighboring components	Linear Prediction (Differential Encoding)			Algorithm based on a linear combination of the neighboring components
		Linear Correlation			Algorithm based on a linear correlation of the neighboring components
		Error Square			Algorithm based on the squared prediction error of the neighboring components
		Motion Compensation	Block-Based		Size of block Displacement between blocks Prediction Vectors
			Motion - compensated Prediction	Frame	Region of searching Displacement between regions
				Field	
				Dual Prime	
				16x8 block	
Transformer	To transform lineally a symbol sequence into another sequence in which most of the information is contained in only a few elements	Discrete Cosine			Discrete Cosine Transform matrix
		Discrete Sine			Discrete Sine Transform matrix
		Karhunen-Loeve			"Eigenvectors" of an autocorrelation matrix
		Discrete Walsh -Hadamard			Arrangement of discrete Hadamard matrices

Table 3. Supporting Coding Processes for mapping the transformed streams of continuous data

MECHANISMS	FUNCTION	OPERATIONS			PARAMETERS
Quantizer	To make a sequence of continuous symbols into a sequence of discrete symbols	Scalar	Uniform	Midtread Midrise	Intervals Number, Set of codewords
				Forward Backward	Adaptation parameter (variance or pdf of sample values), Intervals Number, Set of codewords
			Nonuniform	Midrise	Distribution of Intervals, Set of codewords
				Companded	Pdf of the sample values, Set of codewords
		Vector	Structured	Pyramid	Vector Dimension, Set of code-vectors with binary index
				Polar and Spherical	
				Lattice	
			Tree-Structured Pruned Tree-Structured		
			Gain-Shape		Normalization Factor, Vector Dimension, Set of code-vectors with binary index
			Mean-Removed		Vector Dimension, Set of mean-removed code-vectors with binary index
			Classified		Vector Dimension, Classes of sets of code-vectors with binary index
			Multistage		Vector Dimension, Number of stages, Set of code-vectors with binary index for each stage
			Adaptive		Vector Dimension, Adaptive set of code-vectors with binary index
Compressor or Assignator	To assign binary sequences to symbol sequence	Static	Fixed-to-fixed-length coding		Codebook with fixed length codewords
			Fixed-to-variable-length coding		Codebook with variable length codewords
			Variable-to-variable-length coding		Length Indicators, Codebook with codewords of variable length
			Variable-to-fixed-length coding		Operation Flag, Codebook (or dictionary) with codewords of fixed length and their indexes
		Dynamics	Adaptive Coding		Symbol Initiator, Codebooks with codewords of fixed or variable length
			Bit allocation		Algorithm to bit allocation based on the estimation of the variance of the symbol sequence
			Threshold Coding		Algorithm to estimate bit allocation based on threshold value

Sampler	To make a sequence of analogous waveform into a sequence of continuous symbols	Down-sampling (Decimation)	Decimation Factor

Table 4. Supporting Coding Processes for structuring the requested streams of continuous data

MECHANISMS	FUNCTION	OPERATIONS		PARAMETERS
Transmitter	To define transmitting way	Images	Sequential Progressive	Descriptor of type of transmission
			Pyramid	
Compressor or Assignator	To assign binary sequences to symbol sequence	Static	Fixed-to-fixed-length coding	Codebook with fixed length codewords
			Fixed-to-variable-length coding	Codebook with variable length codewords
			Variable-to-variable-length coding	Length Indicators, Codebook with codewords of variable length
			Variable-to-fixed-length coding	Operation Flag, Codebook (or dictionary) with codewords of fixed length and their indexes
		Dynamics	Adaptive Coding	Symbol Initiator, Codebooks with codewords of fixed or variable length
			Bit allocation	Algorithm to bit allocation based on the estimation of the variance of the symbol sequence
			Threshold Coding	Algorithm to estimate bit allocation based on threshold value
Order (or scanner)	To make ordering of components	Line-by-line, from left -to-right and from top-to-bottom		Raster Scan
		Zig-Zag ordering		Limits for ordering