HAL archives-ouvertes.fr

H.261 software Codec for videoconferencing over the internet

Thierry Turletti

To cite this version:

Thierry Turletti. H.261 software Codec for videoconferencing over the internet. [Research Report] RR-1834, INRIA. 1993. <inria-00074837>

HAL Id: inria-00074837
https://hal.inria.fr/inria-00074837

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Rapports de Recherche

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

N°1834

Programme 1

Architectures parallèles, Bases de données, Réseaux et Systèmes distribués

H.261 SOFTWARE CODEC FOR VIDEOCONFERENCING OVER THE INTERNET

Thierry TURLETTI

January 1993

Institut National
de Recherche
en Informatique
et en Automatique

2004 route des Lucioles
B.P. 93
06902 Sophia-Antipolis
France

H.261 software codec for videoconferencing over the Internet.

Logiciel de vidéoconférence implémentant un codec H.261 au dessus du réseau Internet.

Thierry Turletti

I.N.R.I.A. Route des Lucioles
B.P. 93
06902 Sophia-Antipolis
France

turletti@sophia.inria.fr

Résumé Ce rapport décrit une application de vidéoconférence à bas débit sur le réseau Internet qui utilise le protocole de transport User Datagram Protocol (UDP) ainsi que les extensions multicast de Internet Protocol (IP). Le codeur-décodeur vidéo est réalisé en logiciel et suit la recommendation H.261 du CCITT prévue à l'origine pour le Réseau Numérique à Intégration de Services (RNIS). La compression obtenue est telle que ce genre d'application actuellement en pleine expansion convient pour des réseaux à bas débit; une vidéoconférence nécessitant moins de 30 kb/s.

Après une vue d'ensemble des différentes méthodes de compression de données et une présentation de la recommendation H.261, nous décrivons en détails notre implémentation IVS qui est librement disponible dans le domaine public.

Abstract This report describes a low-bandwidth videoconferencing application on the Internet using the IP multicast extensions and the User Datagram Protocol (UDP) transport protocol. The video coder-decoder is a software implementation of the CCITT recommendation H.261 originally developped for the Integrated Services Digital Network (ISDN). Until now, H.261 codecs have been implemented in hardware. We find that the mean output rate of the coder is less than 30 kb/s, thus making videoconferencing applications possible over low-speed networks such as the Internet.

After a brief overview of the different data compression techniques and a description of the recommendation H.261, we describe in more details IVS, our videoconferencing application which is freely available in the public domain.

Keywords codec, CCITT recommendation H.261, data compression, Internet, IP multicast extensions, UDP, videoconferencing.

1 Introduction

Videoconference applications have often been described as requiring very high bandwidth, efficient hardware and specific performance requirements such as throughput or delay guaranties. Such stringent requirements are difficult to support with current technology, especially since current networks such as the Internet only provide low-to-medium speed transmission. In reality, high bandwith is required only when images are sent over the network without any compression. Thus, various compression algorithms are routinely applied to video data to reduce the transmission bandwidth. Let us consider the easiest way to send images over the network. An image can be represented by a 2D-array of pixels or coefficients, each pixel being characterized by a shade of gray. The obvious way to send an image consists of scanning coefficients line by line and sending them over the network. But the total necessary bandwith for that approach is very large. For example, a $256*256$ image with each pixel encoded with 8 bits (256 shades of gray) requires about 500 kbits. Since video applications such as videoconferencing need several images per second, compression is indispensable to fit in with low-speed networks like the Internet.

Efficient compression algorithms require significant CPU time. This has tended to result in an unsatisfactory trade-off between video requirements for bandwidth and efficiency. Fortunately, the growing computing power of workstations has allowed us to implement a software version of a video coder-decoder (codec) with good performance characteristics. This approach provides numerous advantages. Minimal hardware requirements, such as cheap cameras and video frame grabbers, result in low cost. The software framework is easy to parameterize and modify and thus provides a simple way to respond dynamically to network congestion and to control the output rate. The codec follows the state of the art compression algorithm defined in the H.261 specification and uses User Datagram Protocol (UDP) to transmit data. It is freely available in the public domain and has been tested among several countries.

It is reasonable to have videoconferences over the Internet right now with about 30 kbs/s. Experiments show that the problem of guaranties such as jitter and losses does not prevent videoconferencing applications over the Internet. This paper shows that with our approach such applications are feasible with less than 30 kb/s.

The paper is organized as follows. In section 2.1, we briefly review different data compression techniques. In section 2.2, we describe the CCITT recommendation H.261 which is at the state of the art. In section 4, we describe the INRIA Videoconferencing system (IVS), which integrates a software codec implementation of this standard.

2 Data compression techniques

Data compression techniques fall into two categories, namely *entropy reduction* and *redundancy reduction* [9]. The main difference is that the redundancy reduction operation is a reversible process whereas the entropy reduction operation introduces the concept of compression with distortion, or irreversible compression.

2.1 Entropy reduction coding

Entropy is a concept used in the information theory to characterize the amount of information contained in a message. Reducing entropy amounts to losing some information. Thus, it is clear that entropy reduction is an irreversible operation.

One example of entropy reduction is *quantization*. Quantization is the name given to the process of approximating the individual signal samples to the nearest permitted digital level. A quantizer is usually characterized by its step size and its quantizing error which is the difference between the input analog value and the corresponding discrete output level.

Three types of quantization exist : *scalar*, *block* and *sequential*. In the scalar type, each sample is quantized independently of all others with the same quantizer. Better performances are obtained for block and sequential quantization which benefit from dependancy between samples. Block quantization techniques may be divided in *Vector quantization* and *Transform coding*; the most common form of sequential quantization is *Predictive coding*.

- Vector quantization:

  Instead of quantizing each sample individually with the same quantizing function, an approximation is made between each block and a codebook of several sequences that most closely matches the input sequence. This optimum match is based on the minimization of the distortion between the two blocks [18]. The codebook must be processed from the encoder using training pictures, which is the most important processing part of this technique. Then a copy of this codebook must be transmitted to the decoding process. After that, for each block a matching is done and only the label of the chosen vector is transmitted. Decoding part is easiest : looking up the label in the codebook and using the corresponding vector to represent the block in the image.

- Transform coding:

  In this technique, the image is transformed in a different domain space, in which each sample is then quantized. The goal is to quantize the entire block with fewer bits than would be needed in the original domain thanks to almost no correlation between samples in this new domain. The transformation that is usually used is a linear transformation such the Hadamar, Haar, Fourier, Cosine or Wavelet transformation. The optimum is the Karhunen Loeve Transform. The Discrete Cosine Transform (DCT) approaches the optimum performance for video bandwith compression [13]. Transform coding usually works on small blocks of pixels, such as $(8 * 8)$ or $(16 * 16)$.

- Predictive coding:

  Here, the next sample is predicted and only the difference between the predicted value and the actual value is quantized. This approach is worthwhile because the variance of the difference obtained in this way is less than the variance of the original signal, so, greater compression is possible. Generally, predictors are designed to use previous samples to predict the next sample, rather than a model of the data source which may be time-varying which in most cases it would be very difficult to implement. The best known techniques are *delta modulation* and *differential pulse code modulation* (DPCM).

2.2 Redundancy reduction coding

The second class of compression technique, *redundancy reduction* allows the exact recovery of the original data. Only redundant data is removed. Redundancy reduction techniques may be divided into several classes: *Optimum source coding*, *nonredundant sample coding* and *binary source coding*.

- Optimum source coding:

  The optimum coding techniques such as the Shannon-Fano code and the Huffman code are designed for statistically independant sources. Huffman compression gives a reduction in the average code length used to represent the symbols of an alphabet, but can be used only if the probabilities of all source outputs are known in advance.

- Nonredundant sample coding:

  Nonredundant sample coding looks like predictive coding. New values that do not match the predicted values are called nonredundant. Rather than transmit the difference between the new and predicted values, values are transmitted only if different than predicted. Since new samples which are equal to predicted samples are not transmitted, the output of this kind of predictor is asynchronous. So, to allow a complete reconstruction of the original sample sequence at the receiving end, a special code must be sent to encode timing information.

- Binary source coding:

  A binary source is a source which produces only two levels or levels. The goal of binary source coding is to make the number of bits per source block close to, or equal to the entropy of that block. The Run-Length Codes are the most known.

2.3 Hybrid coding

Methods which use combinations of the previous data compression techniques are called *hybrid coding*. The H.261 specification is a hybrid coding using both entropy and redundancy reduction. Utilization of transform coding, predictive coding and optimum source coding bring it to the state of the art of video compression techniques.

3 The CCITT Recommendation H.261

The CCITT recommendation H.261 describes the video coding and decoding methods for the moving picture component of audiovisual services at rates of $p * 64$ kbit/s, where p is in the range 1 to 30 [7], [8], [16]. The compression techniques used are at the state of the art with regard to video compression encoding methods. The H.261 includes descriptions of a coding mechanism and a scheme to organize video data in a hierarchical fashion. The compression techniques used by the coding mechanism include transform coding, quantization, Huffman encoding and optionally vector motion compensation.

3.1 Source coding

In order to allow a single recommendation to cover use between regions using different television standards, the CCITT has adopted the Common Intermediate Format (CIF) and the Quarter-CIF (QCIF). Pictures are encoded as *luminance* (Y) and two *color-difference components* (CB and CR). Y, CB and CR components are each functions of the standard chrominance components (red, green and blue) and are defined in CCIR Recommendation 601.

The first format (CIF) has 352 pixels per line and 288 lines per picture. Since each block of four pixels is encoded with four Y, one CB and one CR components, sampling of the two color-difference component is 176 pixels per line, 144 lines per image all in an orthogonal arrangement. The QCIF has half as many pixels and half as many lines as CIF. All codecs must be able to handle QCIF whereas use of CIF is optional. As a rule, QCIF is mainly used for desktop videophone applications and CIF is more suitable for videoconferencing applications due to its better resolution.

Coding is done either on the input pictures, or on the difference between successive images, i.e. the prediction error. The first case is referred as *intraframe* coding (INTRA mode), the second case to *interframe* coding (INTER mode). Intraframe coding means that the image is encoded without any relation to the older sequences. This kind of encoding removes only the spatial redundancy in a picture whereas interframe coding also removes