A Study of Real-Time Packet Video Quality using Random Neural Networks

Samir Mohamed, Gerardo Rubino

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Study of Real-Time Packet Video Quality using Random Neural Networks

Samir Mohamed, Gerardo Rubino

N˚4525

Aout 2002

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--4525--FR+ENG

A Study of Real-Time Packet Video Quality using Random Neural Networks

Samir Mohamed, Gerardo Rubino

Thème 1 — Réseaux et systèmes
Projet ARMOR

Rapport de recherche n ° 4525 — Aout 2002 — 33 pages

Abstract: An important and unsolved problem today is the automatic quantification of the quality of video flows transmitted over packet networks. In particular, the ability to perform this task in real time (typically for streams sent themselves in real time) is specially interesting. The problem is still unsolved because there are many parameters affecting video quality and because their combined effect is not well identified and understood. Among these parameters we have the source bit rate, the encoded frame type, the frame rate at the source, the packet loss rate in the network, etc. Only subjective evaluations give good results but, by definition, they are not automatic. We previously explored the possibility of using Artificial Neural Networks to automatically quantify the quality of video flows and we showed that they can give results well correlated with human perception.

In this paper, our goal is twofold: First, we report on a significant enhancement of our method by means of a new neural approach, the Random Neural Network model. Second, we follow our approach to study and analyze the behavior of video quality for wide range variations of a set of selected parameters. This may help in developing control mechanisms in order to deliver the best possible video quality given the current network situation, and in better understanding of QoS aspects in multimedia engineering.

Key-words: Packet video, Random Neural Networks, Real-time video transmission, Video quality assessment.

(Résumé : tsvp)

This work was partially supported by the European ITEA Project 99011 "RTIPA" (Real-Time Internet Platform Architectures).

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Une étude de la qualité de la vidéo temps réel employant Random Neural Networks

Résumé : Un problème important et non résolu efficacement jusqu'aujourd'hui est l'évaluation automatique de la qualité de flux vidéo transmis sur des réseaux de paquet, telle que perçue par l'utilisateur. La capacité de faire cette tâche en temps réel (typiquement pour des flux de type temps réel) est particulièrement intéressante. Le problème est toujours non résolu parce qu'il y a beaucoup de paramètres affectant la qualité de la vidéo et parce que leurs effets combinés ne sont pas bien identifiés et compris. Parmi ces paramètres, nous avons le débit de la source, le type d'encodage d'images, le débit des trames, le taux de perte dans le réseau, etc. Seulement des évaluations subjectives donnent de bons résultats, mais, par définition, elles ne sont pas automatiques. Nous avons précédemment exploré la possibilité d'employer les réseaux de neurones artificiels pour automatiquement évaluer quantitativement la qualité de flux vidéo et nous avons montré qu'ils peuvent donner des résultats bien corrélés avec la perception humaine.

Dans ce papier, notre but est double : d'abord, nous présentons une amélioration significative de notre méthode en utilisant une nouvelle approche neuronale, le modèle de réseaux de neurones aléatoires. Ensuite, nous utilisons notre approche pour étudier et analyser le comportement de la qualité vidéo pour des variations importantes d'un jeu de paramètres préalablement choisis. Cela peut aider à développer des mécanismes de contrôle pour livrer la meilleure qualité vidéo possible étant donné la situation actuelle du réseau, et à une meilleure compréhension des aspects QoS dans l'ingénierie multimédia.

Mots-clé : Paquets vidéo, réseaux de neurones aléatoires, transmission de vidéo temps réel, évaluation de la qualité de la vidéo.

1 Introduction

Even though the main concern of multimedia QoS is to maximize the quality of the data delivered to the destination points, most of the current discussions are concentrated on finding a way to keep within certain limits some of the network parameters (e.g., packet loss rate and delay variation), and little attention has been paid to the quality perceived by the end-users of the applications running over the network.

There are several parameters that affect the quality (quality-affecting parameters) of video transmission over packet networks. We can classify them as follows:

- Coding and compression parameters: They control the amount of quality losses that take place during the encoding process; so they depend on the type of the encoding algorithm (MPEG, H26x, etc.), the output bit rate, the frame rate (the number of frames per sec.), the temporal relation among frame types, etc.;
- Network parameters: They are a result of packetization of the video stream and the transmission through the network, such as the packet loss rate, the loss distribution, the delay, the delay variation (jitter), etc.;
- Other parameters like the nature of the scene (e.g. amount of motion, color, contrast, image size, ...) can also have an impact on the human perception of the flow. Last, it is possible that some other characteristics of the user population itself such as the age or even social and economic factors have some impact on the perceived quality.

The analysis of the video quality can be done using either *objective* tests or *subjective* ones. Objective tests are usually explicit functions of measurable parameters related to the encoder or to the network [26]. Subjective tests are based on evaluations made by human subjects under well defined and controlled conditions [7], [27]. Obviously, the reference is the end-user's perception, which is directly captured by subjective tests. Concerning available objective tests, it is well known that they do not always correlate very well with human perception [35], [19], [32].

Some existing objective methods are MSE (Mean Square Error) or PSNR (Peak Signal to Noise Ratio) which measure the quality by simple pixel-to-pixel comparisons. There are other more complicated methods such as the moving picture quality metric (MPQM) and the normalized video fidelity metric (NVFM) [38], [39]. A state of the art for objective video quality assessment methods is [26].

Subjective quality assessment methods [8], [19], [7], [37], [27], measure the overall perceived video quality. They are carried out by human subjects. The most commonly used one for video quality evaluation is the Mean Opinion Score (MOS) [7], [27], recommended by the ITU. It consists of having a set of subjects view the distorted video sequences in order to rate their

quality, according to a predefined quality scale. That is, human subjects are trained to *build* a mapping between a set of processed video sequences and the quality scale. Although MOS studies have served as the basis for analyzing many aspects of signal processing, they present several limitations: a) very stringent environments are required; b) the process can not be automated; c) it is very costly and time consuming, making very difficult to repeat it frequently. Consequently, it is impossible to use it in real-time quality assessment. On the other hand, the disadvantages of objective methods are: a) they do not always correlate well with human visual perception[1]; b) they require high calculation power, and are time consuming (they usually operate at the pixel level); c) it is very hard to adapt them to real-time quality assessment, as they usually work on both the original video sequence and the transmitted/distorted one; d) as stated before, it is difficult to build a model that takes into account the effect of many quality-affecting parameters, specially network parameters.

Previous studies either concentrated on the effect of network parameters without paying attention to encoding parameters, or the contrary. Papers that consider network parameters and use subjective tests for the evaluation restrict the study to only one or two of the most important ones. For example, the Loss Rate and Consecutive Lost Packet metrics are studied in [17], while [38] studies mainly the effect of Bit Rate and [15] works only on the effect of Frame Rate, *etc.* Other examples are [9], [19], [32], [30] (see Section 2 for more details). The main reason for this is the fact that subjective quality tests are expensive to carry out. Moreover, to analyze the combined effect, for instance, of three or four parameters, we need to build a very large set of human evaluations in order to reach a minimal precision level. Concerning objective approaches, there is no previously published objective quality test that can take into account the direct influence on the quality of the whole set of parameters simultaneously.

In this paper, we propose an approach that, first of all, can evaluate the combined effect of an arbitrary number of parameters on the quality of a video sequence. Moreover, our method has two supplementary properties: (i) it correlates well with the results obtained from subjective tests (because, as we will see, it is in fact based on them) and (ii) it can work automatically and in real time. Specifically, we build a tool that takes as input the values of a set of parameters associated with the encoder and with the network used to transmit the video stream, and correspondingly quantifies the video quality. The tool is based on a neural network trained with the results of previously performed subjective tests, in which wide ranges of the selected parameters and real network conditions are considered. In [25], we proposed to use standard Artificial Neural Networks (ANN) to evaluate video quality as a function of certain quality-affecting parameters.

Here, we report on a significant enhancement of this method, obtained by using a different type of neural network called Random Neural Network (RNN). This recently invented tool [11], [13], [12] appears to capture with higher accuracy and in a more robust way the function mapping

[1] By the way, this claim comes from comparing the results to those obtained from subjective methods.

the various involved parameters to the quality metric. This is partly due to nice mathematical properties exhibited by RNN, which makes one of their main differences with ANN.

In addition, we use our tool to study and analyze the impact of certain quality-affecting parameters (the source bit rate, the encoded frame type, the frame rate, the packet loss rate, and the loss burst size) on real-time video quality. The possibility of performing this type of analysis is another contribution of the approach developped here.

The organization of the rest of our paper is as follows: in Section 2, we give a brief overview of related works. Section 3 summarizes our previously proposed model to evaluate real-time video quality, introduces the theory of RNN, compares both RNN and ANN, and mentions some benefits of our model. Section 4, presents the subjective quality tests. In Section 5, we describe the quality-affecting parameters and the MOS test we carried out. Finally, in Section 6, we study the impact of the quality-affecting parameters on video quality.

2 Related Works

In [9], the authors study the effect of both loss and jitter on the perceptual quality of video. They argue that, if there is no mechanism to mask the effect of jitter, the perceived quality degrades in the same way as it degrades with losses. In [34], [17], [19], [36], [37] and [32], the effect of audio synchronization on the perceived video quality is analyzed (for instance, by quantifying the benefits of audio synchronization on the overall quality of the flow). The main goal of [15] is to study the effect of the frame rate for different standard video sequences on the overall perceived quality. A related work is [30] where the effect of FEC (Forward Error Correction) and Frame Rate on the quality is the subject of the study. The work presented in [6] is a study of the packet loss effects on MPEG video streams. The authors consider the effect of loss rate on the different types of MPEG frames. In [38] and [26], a study of the effect of bit rate on the objective quality metrics (PSNR, NVFM, and MPQM) is presented. The effect of the number of consecutively lost packets on video quality is analyzed in [17].

The authors of [20] study the effect of packet size and the distribution of I-frames in the layered video transmission over IP networks. In [8] the analysis goes deeper: the authors present a study of the effect of motion on the perceived video quality. In [29], a dynamic bit-rate prediction method is proposed. It consists of predicting the bit-rate for future frames based on the past information and dynamically change the quantization parameter if the estimated bit rate exceeds certain threshold so that the encoder output's objective quality remains constant. In [18], another objective video quality assessment method is proposed, based on the evaluation of the quality of each frame separately. The idea is to compute a weighted mean of these individual frame values, taking into account the short-term characteristics of human memory.

The work in [28] presents a methodology for video quality assessment using objective parameters based on image segmentation. An image encoded by MPEG-2 is segmented into three regions: plane, edges, and texture; then, a set of objective parameters is assigned to each region. After that, a perceptual-based model is defined by computing the relationship between objective measures and results of subjective tests. In [40], a de-jittering scheme is proposed to compensate the effect of delay variation for the transport of MPEG-4/2 video streams. Furthermore, in [40], an algorithm to protect the packet-loss resilience is proposed. It is based on the switching between Inter/Intra macro-bloc encoding. Although the main goal of the works presented in [35], [19] and [32] is to develop objective methods that give good correlation with subjective evaluations, the results obtained clearly show that the two types of outputs do not always correlate well.

In previous work, we showed how to use ANN to measure audio quality in real time when this audio is transmitted over a packet network [23]. Based on this technique, we developed a new control mechanism that permits a better use of the given bandwidth and the delivery of the best possible audio quality given the current network situation [24]. Then, we proposed a tool to evaluate video quality in real time when this video is subjected to wide range variation of certain network and video parameters [25]. The present paper extends and improves some of these works, in the specific case of video streams.

Concerning the specific neural approach followed in the present paper, based on the RNN model, it is important to mention that RNN are used in several related problems. For example, they are used in video compression with compression ratios that range from 500:1 to 1000:1 for moving gray-scale images and full-color video sequences respectively [10]. They are also used as decoders for error correcting codes in noisy communication channels [2]. Furthermore, they are used in a variety of image processing techniques which go from image enlargement and fusion to image segmentation [14]. A survey of RNN applications is given in [4].

3 Automatic Measuring of Video Quality in Real Time

In this Section, we summarize our proposal to evaluate video quality in an automatic way, in real time if necessary, and with results close to those that can be obtained from subjective tests. Then we briefly describe the fundamentals of RNN. After that, we compare ANN and RNN for the sake of our study. Finally, possible applications for our method are proposed.

3.1 Summary of the Neural Network approach

In this Section, we describe the overall steps to be followed in order to build a tool to automatically assess (and in real time if necessary) the quality of video streams transmitted over packet networks. The automatic evaluation is performed by a suitable trained Neural Network (NN).

We must first choose the most *a priori* effective quality-affecting parameters corresponding to the type of video application and to the network that will support the transmission. Then, for each parameter we must select the most frequent occurrences of its values and identify their ranges. For example, if the percentage loss rate is expected to vary from 0 to 10%, then we may use 0, 1, 2, 5, and 10 % as typical values for this parameter. If we call *configuration* of the set of quality-affecting parameters, a set of values for each one, the total number of possible configurations is usually large. We must then select a part of this large cardinality set, which will be used as (part of) the input data of the NN in the learning phase.

To generate a video database composed of sequences corresponding to different configurations of the selected parameters (called "Distorted Database"), a simulation environment or a testbed must be implemented. This is used to send video sequences from the source to the destination and to control the underlying packet network. Every configuration in the defined input data must be mapped into the system composed of the network, the source and the receiver. For example, working with IP networks, the source controls the bit rate, the frame rate and the encoding algorithm, and it sends RTP video packets; the routers' behavior contribute to the loss rate and the loss distribution, together with the traffic conditions in the network. The destination stores the transmitted video sequence and collects the corresponding values of the parameters. Then, by running the testbed or using the simulations, we produce and store a set of distorted video sequences along with the corresponding parameters' values.

After completing the Distorted Database, a subjective quality test must be carried out. There are several subjective quality methods in the recommendations of the ITU-R [7, 27]. We selected to use Degradation Category Rating (DCR), discussed in Section 4. A group of human subjects is then invited to evaluate the quality of the video sequences (i.e. every subject gives each video sequence a score from a predefined quality scale). The subjects must not establish any relation between the sequences and the corresponding parameters' values.

The next step is to calculate the MOS values for all the video sequences. Based on the scores given by the human subjects, screening and statistical analysis should be carried out to remove the grading of the individuals suspected to give unreliable results [7]. After that, we store the MOS values and the corresponding parameters' values in a second database (which we call the "Quality Database").

In the third step, a suitable NN architecture and a training algorithm must be selected. The Quality Database is divided into two parts: one to train the NN and the other one to test its

accuracy. The trained NN will then be able to evaluate the quality measure for any given values of the parameters.

To put this more formally, we build a set $\mathcal{S} = \{\sigma_1, \sigma_2, \cdots, \sigma_S\}$ of video sequences that have encountered varied conditions when transmitted and that constitute the "training part" of the Quality Database. A second part of the data base, the set $\mathcal{S}' = \{\sigma'_1, \sigma'_2, \cdots, \sigma'_{S'}\}$, called the "validation part" of the Quality Database, is reserved. We also define a set $\mathcal{P} = \{\pi_1, \pi_2, \cdots, \pi_P\}$ of parameters such as the bit rate of the source, the packet loss rate in the network, etc. Then, we denote by v_{ps} the value of parameter π_p in sequence σ_s, and by V the matrix $V = (v_{ps})$. For $s = 1, 2, \cdots, S$, sequence σ_s receives the MOS evaluation $\mu_s \in [0, M]$ from the subjective test phase. The goal of the NN is to find a real function f having P real variables and with values in $[0, M]$, such that

(i) for any sequence s, $f(v_{1s}, \cdots, v_{P,s}) \approx \mu_s$,

(ii) and such that for *any other* vector of parameter values $(v_1, \cdots, v_P)$, $f(v_1, \cdots, v_P)$ is close to the MOS that would receive any video sequence for which the selected parameters would have those specific values $v_1, \cdots, v_P$.

When such a function f is built (f is actually the final NN obtained when the convergence criteria is satisfied), we test it using the validation part of the Quality Database. If for any sequence $\sigma'_s \in \mathcal{S}'$ we have $f(v_{1s}, \cdots, v_{P,s}) \approx \mu_s$, then the training process ends. Otherwise, we must go back to the first phase, probably to use more data or to change some parameters of the neural network and build another function f.

The final tool is then composed of two modules: the first one collects the values of the selected quality-affecting parameters. The second one is the trained NN that will take the given values of the quality-affecting parameters and correspondingly computes the MOS quality score.

3.2 Random Neural Networks

The method we propose uses a new family of neural networks, the so called Random Neural Networks, recently invented by Erol Gelenbe in [11], [13], [12]. This choice was suggested by the success of this approach in many different areas [10], [2], [14], [4], ...

Gelenbe's idea can be described as a merge between the classical Artificial Neural Networks (ANN) model and queuing networks. Since this tool is a novel one, let us describe here its main characteristics. RNN are, as ANN, composed of a set of interconnected neurons. These neurons exchange signals that travel instantaneously from neuron to neuron, and send and receive signals to and from the environment. Each neuron has a *potential* associated with, which is an integer (random) variable. The potential of neuron i at time t is denoted by $q_i(t)$. If the potential of neuron i is strictly positive, the neuron is *excited*; in that state, it randomly sends signals (to

other neurons or to the environment), according to a Poisson process with rate r_i. Signals can be positive or negative. The probability that a signal sent by neuron i goes to neuron j as a positive one, is denoted by $p^+_{i,j}$, and as a negative one, by $p^-_{i,j}$; the signal goes to the environment (that is, it leaves the network) with probability d_i. So, if N is the number of neurons, we must have for all $i = 1, \cdots, N$,

$$d_i + \sum_{j=1}^{N} \left(p^+_{i,j} + p^-_{i,j} \right) = 1.$$

When a neuron receives a positive signal, either from another neuron or from the environment, its potential is increased by 1; if it receives a negative one, its potential decreases by 1 if it was strictly positive and it does not change if its value was 0. In the same way, when a neuron sends a signal, positive or negative, its potential is decreased by one unit (it was necessarily strictly positive since only excited neurons send signals)[2]. The flow of positive (resp. negative) signals arriving from the environment to neuron i (if any) is a Poisson process which rate is denoted by λ^+_i (resp. λ^-_i). It is possible to have $\lambda^+_i = 0$ and/or $\lambda^-_i = 0$ for some neuron i, but to deal with an "alive" network, we need $\sum_{i=1}^{N} \lambda^+_i > 0$. Finally, we make the usual independence assumptions between these arrival processes, the processes composed of the signals sent by each neuron, etc.

The discovery of Gelenbe is that this model has a *product form* stationary solution. This is similar to the classical Jackson's result on open networks of queues. If process $\vec{q}(t) = (q_1(t), \cdots, q_N(t))$ is ergodic (we will say that the network is *stable*), Gelenbe proved that

$$\lim_{t \to \infty} \Pr(\vec{q}(t) = (n_1, \cdots, n_N)) = \prod_{i=1}^{N} (1 - \varrho_i) \varrho_i^{n_i} \tag{1}$$

where the ϱ_is satisfy the following non-linear system of equations:

$$\text{for each node } i, \quad \varrho_i = \frac{T^+_i}{r_i + T^-_i}, \tag{2}$$

$$\text{for each node } i, \quad T^+_i = \lambda^+_i + \sum_{j=1}^{N} \varrho_j r_j p^+_{j,i}, \tag{3}$$

and

$$\text{for each node } i, \quad T^-_i = \lambda^-_i + \sum_{j=1}^{N} \varrho_j r_j p^-_{j,i}. \tag{4}$$

[2]In the general mathematical model, loops are allowed, that is, it is possible to have $p^+_{ii} > 0$ or $p^-_{ii} > 0$. In our application, we set $p^+_{ii} = p^-_{ii} = 0$.

Relation (1) tells us that ϱ_i is the probability that, in equilibrium, neuron i is excited, that is,

$$\varrho_i = \lim_{t\to\infty} \Pr(q_i(t) > 0).$$

Observe that the non-linear system composed of equations (2), (3) and (4) has $3N$ equations and $3N$ unknowns (the ϱ_is, the T_i^+s and the T_i^-s). Relations (3) and (4) tell us that T_i^+ is the mean throughput of positive signals arriving to neuron i and that T_i^- is the corresponding mean throughput of negative signals (always in equilibrium). Finally, Gelenbe proved, first, that this non-linear system has a unique solution, and, second, that the stability condition of the network is equivalent to the fact that, for all node i, $\varrho_i < 1$.

Let us describe now the use of this model in statistical learning. Following previous applications of RNN, we fix the λ_i^-s to 0 (so, there is no negative signal arriving from outside). As a learning tool, the RNN will be seen as a black-box having N inputs and N outputs. The inputs are the rates of the incoming flows of positive signals arriving from outside, i.e. the λ_i^+s. The output values are the ϱ_is. In fact, in applications, most of the time some neurons do not receive signals from outside, which simply corresponds to fixing some λ_i^+s to 0; in the same way, users often use as output only a subset of ϱ_is.

At this point, let us assume that the number of neurons has been chosen, and that the topology of the network is selected; this means that we have selected the pairs of neurons that will exchange signals, without fixing the values of the rates r_i and the branching probabilities $p_{i,j}^+$ and $p_{i,j}^-$.

Our learning data is then composed of a set of K input-output pairs, which we will denote here by $\{(\vec{x}^{(k)}, \vec{y}^{\,(k)}),\ k = 1, \cdots, K\}$, where $\vec{x}^{(k)} = (x_1^{(k)}, \cdots, x_N^{(k)})$ and $\vec{y}^{\,(k)} = (y_1^{(k)}, \cdots, y_N^{(k)})$. The goal of the learning process is to obtain values for the remaining parameters of the RNN (the rates r_i and the branching probabilities $p_{i,j}^+$ and $p_{i,j}^-$) such that if, in the resulting RNN, we set $\lambda_i^+ = x_i^{(k)}$ for all i (and $\lambda_i^- = 0$), then, for all i, the steady-state occupation probability ϱ_i is close to $y_i^{(k)}$. This must hold for any value of $k \in \{1, \cdots, K\}$.

To obtain this result, first of all, instead of working with rates and branching probabilities, the following variables are used:

$$w_{i,j}^+ = r_i p_{i,j}^+ \quad \text{and} \quad w_{i,j}^- = r_i p_{i,j}^-.$$

This means that $w_{i,j}^+$ (resp. $w_{i,j}^-$) is the mean throughput in equilibrium of positive (resp. negative) signals going from neuron i to neuron j. They are called *weights* by analogy to standard ANN. The learning algorithm proceeds then formally as follows. The set of weights in the network's topology is initialized to some arbitrary positive value, and then K iterations are performed which modify them. Let us call $w_{i,j}^{+\,(0)}$ and $w_{i,j}^{-\,(0)}$ the initial weights for the connection between i and j. Then, for $k = 1, \cdots, K$, the set of weights at step k is computed from the set of

weights at step $k-1$ using a *learning scheme*, as usual with neural networks. More specifically, denote by $\mathcal{R}^{(k-1)}$ the network obtained after step $k-1$, defined by weights $w_{i,j}^{+\,(k-1)}$ and $w_{i,j}^{-\,(k-1)}$. When we set the input rates (external positive signals) in $\mathcal{R}^{(k-1)}$ to the $x_i^{(k)}$s values, we obtain the steady-state occupations $\varrho_i^{(k)}$s (assuming stability). The weights at step k are then defined by

$$w_{i,j}^{+\,(k)} = w_{i,j}^{+\,(k-1)} - \eta \sum_{l=1}^{N} c_l (\varrho_l^{(k)} - y_l^{(k)}) \frac{\partial \varrho_l}{\partial w_{i,j}^{+}}, \tag{5}$$

$$w_{i,j}^{-\,(k)} = w_{i,j}^{-\,(k-1)} - \eta \sum_{l=1}^{N} c_l (\varrho_l^{(k)} - y_l^{(k)}) \frac{\partial \varrho_l}{\partial w_{i,j}^{-}}, \tag{6}$$

where the partial derivatives $\partial \varrho_h / \partial w_{m,n}^{*}$ ('$*$' being '+' or '$-$') are evaluated at $\varrho_h = \varrho_h^{(k)}$ and $w_{m,n}^{*} = w_{m,n}^{*\,(k-1)}$. Factor c_l is a cost positive term allowing to give different importance to different ouput neurons. If some neuron l must not be considered in the output, we simply set $c_l = 0$. In our cases, we have only one output neuron, so, this factor is not relevant; we put it in (5) and (6) just to give the general form of the equation. This is a gradient descent algorithm, corresponding to the minimization of the cost function

$$\frac{1}{2} \sum_{l=1}^{N} c_l (\varrho_l^{(k)} - y_l^{(k)}) 2.$$

Once again, the relations between the output and the input parameters in the product form result allows to explicitly derive a calculation scheme for the partial derivatives. Instead of solving a non-linear system as (2), (3) and (4), it is shown in [13] that here we just have a linear system to be solved. When relations (5) and (6) are applied, it may happen that some value $w_{i,j}^{+\,(k)}$ or $w_{i,j}^{-\,(k)}$ is negative. Since this is not allowed in the model, the weight is set to 0 and it is no more concerned by the updating process (another possibility is to modify the η coefficient and apply the relation again; previous studies have been done using the first discussed solution, which we also adopt).

Once the K learning values have been used, the whole process is repeated several times, until some convergence conditions are satisfied. Remember that, ideally, we want to obtain a network able to give output $\vec{y}^{\,(k)}$ when the input is $\vec{x}^{(k)}$, for $k = 1, \cdots, K$. The link with the notation in previous section is simply the following: K is the size of the training part of the Quality Database; for $k = 1, 2, \cdots, K$, $x_i^{(k)} = v_i$ and for the (scalar) output, $y^{(k)} = \mu_k$.

As in most applications of ANN for learning purposes, we use a 3-level network structure: the set of neurons $\{1, \cdots, N\}$ is partitioned into 3 subsets: the set of *input* nodes, the set of intermediate or *hidden* nodes and the set of *output* nodes. The input nodes receive (positive)

signals from outside and don't send signals outside (that is, for each input node i, $\lambda_i^+ > 0$ and $d_i = 0$). For output nodes, the situation is the opposite: $\lambda_i^+ = 0$ and $d_i > 0$. The intermediate nodes are not directly connected to the environment; that is, for any hidden neuron i, we have $\lambda_i^+ = \lambda_i^- = d_i = 0$. Moreover, between the nodes inside each level there are no transitions. Last, input neurons are only connected to hidden ones, and hidden neurons are only connected to output ones.

This is a typical structure for neural networks used as a learning tool. Moreover, RNN mathematical analysis (that is, solving the non-linear and linear systems) is considerably simplified in this case. In particular, it can easily be shown that the network is always stable. See again [13] or [12] for the details.

3.3 Comparison between ANN and RNN

In this subsection, we compare the two considered types of neural networks: Artificial Neural Networks (ANN) and Random Neural Networks (RNN), in the context of our specific problem.

We used the Neural Networks MATLAB Toolbox when working with ANN, and a MATLAB package [3] for RNN. We observed that the ANN training process was relatively faster than that of RNN. However, during the run-time phase, RNN outperformed ANN in the total calculation time. This is because in the RNN's three-level architecture, the computation of the output ϱ_o for the unique output neuron o is done extremely fast: the non-linear system of equations (2), (3) and (4) allows, in this topology, to compute the ϱ_is of the input layer directly, and of hidden layer neurons from the values for input layer ones. To be more specific, for each input neuron i we have

$$\varrho_i = \frac{\lambda_i^+}{r_i + \lambda_i^-},$$

(where, actually, we choose to set $\lambda_i^- = 0$), and for each hidden layer neuron h,

$$\varrho_h = \frac{\sum_{\text{input neuron } i} \varrho_i w_{i,h}^+}{r_h + \sum_{\text{input neuron } i} \varrho_i w_{i,h}^-}.$$

The output of the black-box is then ϱ_o, given by

$$\varrho_o = \frac{\sum_{\text{hidden neuron } h} \varrho_h w_{h,o}^+}{r_o + \sum_{\text{hidden neuron } h} \varrho_h w_{h,o}^-}.$$

(The cost of computing the output ϱ_o is exactly $2IH + 3H + 1$ products (or divisions) and $H + 1$ sums, where I is the number of input neurons and H is the number of hidden ones.)

ANN's computations are slower because they involve more calculations per neuron in the architecture. This makes RNN particularly attractive for using them in contexts with real-time constraints, or for lightweight applications. This can be important in some kind of network applications; for example, in [21], an ANN packet-loss predictor is proposed for real-time multimedia streams. The prediction precision is good, but the calculation time is much more than the next packet arrival time which makes the system useless unless for very powerful computers.

The most important feature of RNN we found for our problem is that it captures very well the mapping from parameters' values to the quality evaluation. This concerns also their ability to extrapolate in a coherent way for parameters' values out of the ranges used during the training phase. For instance, this led in [2] to build a zero-error channel decoder.

It is well known that the most common problems of ANN's learning are the overtraining and the sensitivity to the number of hidden neurons (the choice of the optimal number is difficult; one usually use heuristic methods for this). The overtraining problem makes the NN memorize the training patterns, but gives poor generalizations for new inputs. Moreover, if we can not identify some near-optimal number of hidden neurons, the performance may be bad for both the training set and the new inputs. There exist some heuristic methods aimed to find rough approximation of the optimal number of hidden neurons, but they work well for some problems and fail for others (see [5] for more details). Fig. 1 shows an example of an over-trained ANN network, where we can see irregularities, bad generalizations and bad capturing of the function mapping (see Subsection 6.3 for a comparison with RNN).

We trained different architectures (varying the number of hidden neurons) for both ANN and RNN, with the same data (described in Section 5) and the same *mean square error* threshold. Let us look, for instance, at the behavior of the quality as a function of the normalized bit rate BR (the 4 remaining variables were set to their most frequent observed values). In the database, BR varies between 0.15 to 0.7. In Fig. 2 and Fig. 3, we depict the ability of both neural networks (ANN and RNN) to interpolate and extrapolate the results when BR varies from zero to its maximum value 1. Both figures show that RNN captures the mapping between the input and output variables, and that RNN is not very sensitive to the number of hidden neurons. While ANN gives quite different approximations for small changes in the size of the hidden layer. Moreover, if the optimal ANN architecture can not be identified, its accuracy may be bad. Let us look now at the extreme values. If BR=0.0, the output should be around one, while for BR=1.0, the output should be between 8.5 and 9.0 on the y-axis. For the case of ANN, as shown in Fig. 3, when the number of hidden neurons changes from four (optimal experimentally) to five, the generalization is bad, specially when BR goes to zero, where the

output is 3.2 instead of 1. As said before, RNN is much less sensitive to this factor, at least in our applications.

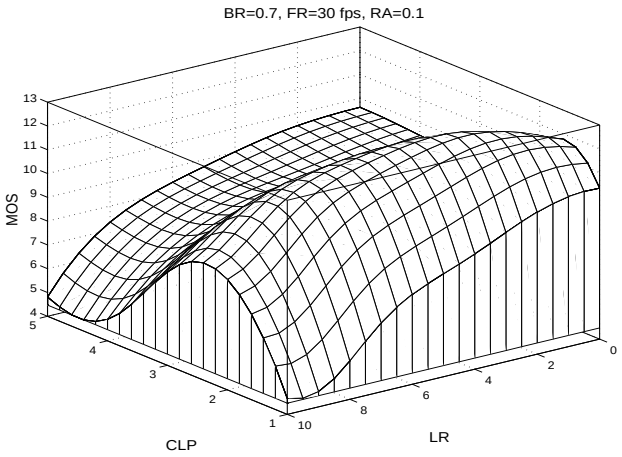

Figure 1: Example of an over-trained ANN

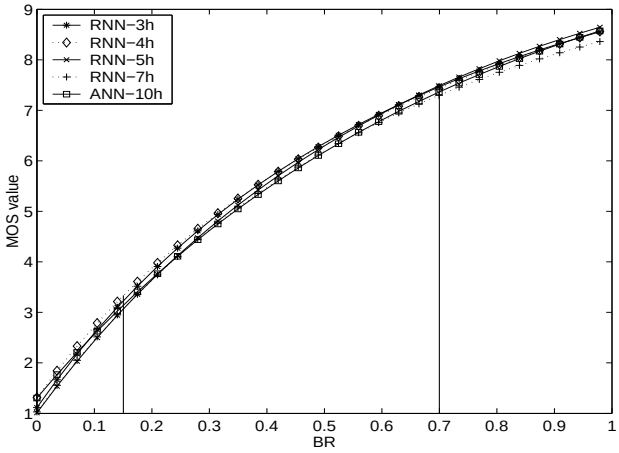

Figure 2: Performance of RNN to interpolate and extrapolate for different number of hidden neurons. The number of hidden neurons goes from 3 (code '3h') to 10 (code '10h').

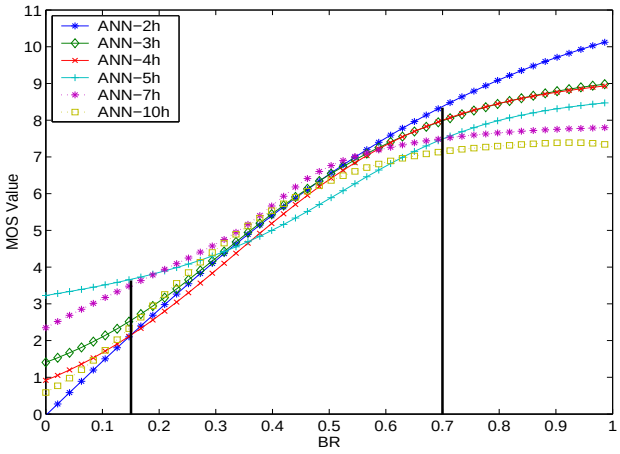

Figure 3: Performance of ANN to interpolate and extrapolate for different number of hidden neurons. The number of hidden neurons goes from 2 (code '2h') to 10 (code '10h').

3.4 Applications of Our Model

Let us briefly discuss here on the benefits from having a tool to automatically measure (and, if useful, in real time) video quality:

- In video conferencing and in the majority of video applications, at both end-user sides, such a tool can be used to monitor in real time the received video quality for control purposes. For example, changing the bit rate, using another codec, changing the frame rate, using some kind of FEC, changing the playback buffer size, etc. are possible decisions that can be taken to improve the quality or to maintain it at a certain level.
- Based on the ability of video quality measurement in real time, operators could use quality as a criteria for billing.
- The applications that transmit video over packet networks can use this tool to negotiate the best configuration to maximize the quality.
- It can help also in the encoding process, as quality in encoders is also a way to "fit" the stream into the available global channel bandwidth. In video codecs using temporal compression (ex: MPEG, H.261, H.263...), a quality factor parameter is usually used to reduce the output stream bandwidth and to reduce, at the same time, the assessed quality (yet before any transmission). It will be even more interesting to have the history of all

the important parameters (network and video) to compress the video signal rather than only the PSNR (Peak Signal to Noise Ratio) objective measure (which is poorer in quality and time consumption than our approach).

4 Subjective Quality and MOS Calculations

As mentioned in the introduction, the main drawback of objective quality tests is that they do not correlate well with human perception. To evaluate the quality of video systems (codecs, telecommunications, television pictures, etc.), a subjective quality test is generally used. In this kind of test, a group of human subjects is invited to judge the quality of the video sequence under different system conditions (distortions). There are several recommendations [7], [27] that specify strict conditions to be followed in order to carry out subjective tests. The main subjective quality methods are Degradation Category Rating (DCR), Pair Comparison (PC) and Absolute Category Rating (ACR). In our case, subjective tests were done using DCR.

In the DCR subjective quality test, a pair of video sequences is presented to each observer, one after the other. The observer must see the first one, which is not distorted by any impairment, and then the second one, which is the original signal distorted by some configuration of the set of chosen quality-affecting parameters. Fig. 4 shows the sequence and timing of presentations for this test. The time values come from ITU-R recommendation [7].

The observer is asked to assess the overall quality of the distorted sequence with respect to the non-distorted one (the reference sequence), using a grade from one to nine corresponding to his/her mental measure of the quality associated with. It should be noted that there exist several quality scales. We chose this nine-grade one as a tradeoff between precision and dispersion of the subjective evaluations. Fig. 5 depicts the ITU-R nine-grade scale.

Following the ITU-R recommendations, overall subjective tests are to be divided into multiple sessions; each session should not last more than 30 minutes. Several dummy sequences are to be added in every session (about four or five) for training purposes. These sequences should not be taken into account in the calculation. They are used to learn the observers how to give meaningful rates.

Let us denote by N the number of observers in the chosen subjective method and by u_{is} the evaluation of sequence σ_s made by user i. The set of values $(u_{is})_{i=1,\cdots,N}$ will probably present variations due to the differences in judgment between subjects. Moreover, it is possible that some observers do not pay attention enough during the experiment, or behave in some unusual way face to the sequences; this can lead to inconsistent data for the training phase. Some statistical filtering is thus necessary on the set of brute data. The most widely used reference to deal with this topic is the Recommendation of ITU-R BT.500-10, [7]. The described procedure allows to remove the ratings of those subjects who could not conduct consistent scores.

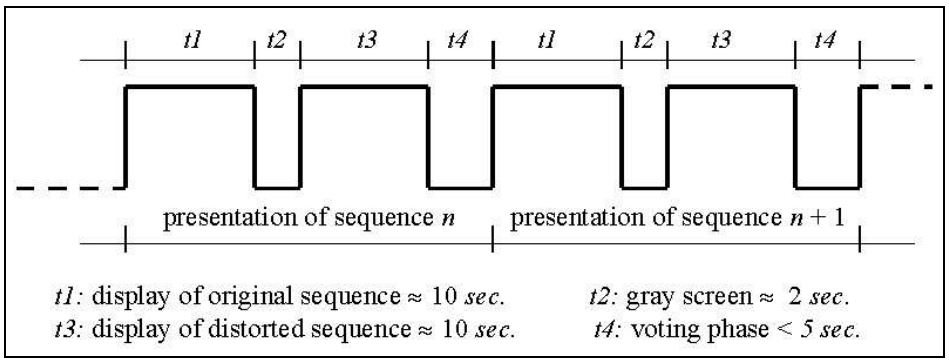

Figure 4: Presentation structure of video sequences to the set of human observers in a DCR subjective quality test experiment.

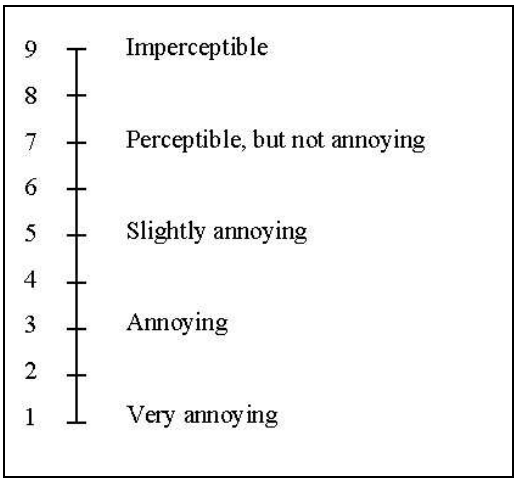

Figure 5: The ITU-R nine-point quality scale

First, denote by $\bar{u}_s$ the mean of the evaluations of sequence σ_s on the set of observers, that is,

$$\bar{u}_s = \frac{1}{N} \sum_{i=1}^{N} u_{is}. \tag{7}$$

Denote by $[\bar{u}_s - \Delta_s, \bar{u}_s + \Delta_s]$ the 95%-confidence interval obtained from the (u_{is}), that is, $\Delta_s = 1.96\delta_s/\sqrt{N}$, where

$$\delta_s = \sqrt{\sum_{i=1}^{N} \frac{(u_{is} - \bar{u}_s)2}{N-1}}.$$

As stated in [7], it must be ascertained whether this distribution of scores is normal or not using the β_2 test (by calculating the "kurtosis" coefficient of the function, i.e. the ratio of the fourth order moment to the square of the second order moment). If β_2 is between 2 and 4, the distribution may be taken to be normal. In symbols, denoting $\beta_{2s} = m_{4s}/m_{2s}^2$ where

$$m_{xs} = \frac{1}{N}\sum_{i=1}^{N}(u_{is} - \bar{u}_s)^x,$$

if $2 \leq \beta_{2s} \leq 4$ then the distribution $(u_{is})_{i=1,\cdots,N}$ can be assumed to be normal. For each subject i, we must compute two integer values L_i and R_i, following the following procedure:

set $L_i = 0$ and $R_i = 0$
for each sequence $\sigma_s \in \mathcal{S} = \{\sigma_1, \cdots, \sigma_S\}$
    if $2 \leq \beta_{2s} \leq 4$, then
        if $u_{is} \geq \bar{u}_s + 2\,\delta_s$ then $R_i = R_i + 1$
        if $u_{is} \leq \bar{u}_s - 2\,\delta_s$ then $L_i = L_i + 1$
    else
        if $u_{is} \geq \bar{u}_s + \sqrt{20}\,\delta_s$ then $R_i = R_i + 1$
        if $u_{is} \leq \bar{u}_s - \sqrt{20}\,\delta_s$ then $L_i = L_i + 1$

Finally, if $(L_i + R_i)/S > 0.05$ and $|(L_i - R_i)/(L_i + R_i)| < 0.3$ then the scores of subject i must be deleted. For more details about this topic and the other methods of subjective tests see [7, 27].

After eliminating the scores of those subjects who could not conduct coherent ratings using the above technique, the mean score should be recomputed using Eq. 7. This will constitute the MOS database that we will use to train and test the NN.

5 Parameters Description and MOS Tests

To generate the distorted video sequences, we used a tool that encodes a real-time video stream over IP networks into the H263 format [16], simulates the packetization of the video stream, decodes the received stream, and allows us to simulate the network transmission conditions (packet loss process, etc.). The encoder can also be parameterized, in order to control the bit rate, the frame rate, the intra macro blocs refresh rate (i.e. it encodes the given macro bloc into

intra mode rather than inter mode - this is done to make the stream more resistant to losses [22]), image format (QCIF, CIF, ...), etc. The packetization process is in conformance with RFC 2429 [1].

In the esperiments reported here, we employed a standard video sequence called `stefan` widely used to test the performance of H26x and MPEG4 codecs. It contains 300 frames encoded at 30 frames per sec., and lasts[3] 10 sec. The format of the encoded sequence is CIF (352 lines $\times$ 288 pixels). The maximum allowed packet length is 536 bytes, in order to avoid the fragmentation of packets between routers. For the presentation done in this paper, we used one single sequence for the tests to avoid any kind of semantic dependencies between the contents of the images and the evaluation made by the human subjects. The same procedure can be followed with any other sequence or set of sequences as well.

5.1 The Quality-Affecting Parameters

We present here the quality-affecting parameters that we consider having the highest impact on the quality:

- The Bit Rate (BR): this is the rate of the actual encoder's output. It is chosen to take four values (256, 512, 768 and 1024 KBps.). With respect to quality, not all the scenes behave the same face to compression, depending on the amount of redundancy in the scene (spatial and temporal), as well as the image dimensions. All video encoders use a mixture of lossless and lossy compression techniques. Lossless compression does not degrade the quality, as the process is reversible. Lossy compression degrades the quality depending on the compression ratio needed by changing the quantization parameter of the Discrete Cosine Transform (DCT). For details about video encoding, see [16]. If the video is encoded only by lossless compression, the decoded video will have the same quality as the original one, provided that there is no other quality degradation. In our method, we normalize the encoder's output in the following way. If BRmax denotes the bit rate after the lossless compression process and BRout is the final encoder's output bit rate, we select the scaled parameter BR = BRout/BRmax. In our environments, we had BRmax = 1430 KBps.

- The Frame Rate (FR): this is the number of frames per second. The original video sequence is encoded at 30 fps. The distorted sequence can be encoded at one of the following values: 6, 10, 15 and 30 fps. The encoder does this by dropping frames uniformly.

- The Loss Rate (LR): the simulator can drop packets randomly and uniformly to satisfy a given percentage loss rate. For this parameter, we selected five values: 0, 1, 2, 4 and 10

[3]This follows ITU recommendations, as usual in the area.

%. This is because loss rates higher than 10 % drastically reduce video quality. In the networks where the LR is expected to be higher than this value, some kind of FEC [33] should be used to reduce the effect of losses.

- The number of Consecutively Lost Packets (CLP): we chose to drop packets in bursts of 1 to 5. The choice of 5 as upper limit for this parameter comes from real measurements that we performed before [23]. Regarding the loss model (LR and CLP), we used the same one as [17].
- The ratio of the encoded intra macro-blocs to inter macro-blocs (RA): the encoder can change the refresh rate of the intra macro-blocs in order to make the encoded sequence more or less sensitive to packet losses [22]. This parameter takes values that vary between 0.05 and 0.50 depending on the BR and the FR. We selected five values for it.

The delay and the delay variation are indirectly considered: they are included in the LR parameter. Indeed, if a de-jittering mechanism with a strict playback buffer length is used, then all the packets arriving after a predefined threshold are considered as lost. Hence, in this way, all delays and delay variations are mapped into loss. This is confirmed in [40]. In addition, the authors argue that the effect of delay and delay jitter can be reduced by the MPEG decoder buffer at the receiver. This is also valid for H263 decoder buffer. A similar de-jittering scheme for the transport of video over ATM is given in [31]. As stated, the author argues that the scheme can be easily adapted to IP networks. Other study about jitter is given in [41] where it is stated that the jitter affect the decoder buffer size and the loss ratio in a significant way.

The whole set of configurations of these 5 parameters has 2000 elements. To build the Distorted Database, we first selected a default value for each parameter (the most frequent observed value). Then, for each possible selection of 3 parameters among the 5 (10 possibilities), we set the 3 parameters to their defaults values, and we built all the possible combinations of the remaining 2. In this way, a list of configurations was defined, containing many duplications. Once these duplications removed, we obtained a set of 94 different configurations.

5.2 Subjective Quality Test and the RNN Architecture

The subjective quality test is in conformance with the method Degradation Category Rating (DCR), with a quality scale consisting of 9 points (see Section 4). We divided the test into two sessions, and added 5 distorted sequences to the first session and 4 to the second one. These nine sequences were not considered in the MOS calculation since they are used as a training phase for the human subjects. At the same time, they are used to verify how reliable is the person carrying out the test, as they are replicated from the real 94 samples.

We invited 20 subjects to perform the subjective tests. After that, a statistical analysis of the results was performed; as a consequence, we discarded the notes of two subjects (refer to Section 4 for more detail). Then, we calculated the MOS scores and the corresponding parameters' values used to train and test the RNN.

As explained in Subsection 3.1, we divided the database into two parts: one to train the RNN, containing 80 samples, and the other to test its performance, containing 14 samples. After training the RNN using the first database and comparing the training set against the values predicted by the RNN, we got a correlation factor = 0.9801, and a mean square error = 0.108; this means that the NN model fits quite well the way in which humans rated the video quality. Then, we used the trained RNN to predict the scores of samples in the testing database (containing samples that have never been seen during the training phase). The results were correlation factor = 0.9821 and mean square error = 0.07. These values validate the obtained neural network.

We used as architecture of RNN, the three-layer feedforward consisting of five neurons in the input layer (which corresponds to the five chosen parameters), one output neuron (corresponding to the MOS score), and 5 neurons in the hidden layer. It should be noted that for the optimal number of hidden neurons for both ANN and RNN (4 and 5 neurons respectively), the obtained performances are almost the same. However, ANN can be overtrained easily if care is not taken, see Section 3.3. To explore this issue, we varied the number of hidden neurons for both ANN and RNN. We used them to evaluate the quality for each case when varying the Bit Rate and fixing the other parameters. As we can see from Fig. 2 and Fig. 3 that for the optimal number of hidden neurons, they give the same performance. But ANN cannot generalize for the other values. However, RNN almost give the same results for different values of hidden neurons.

6 Parameters Impact on Video Quality

In this section, we study the effects of the mentioned five parameters on video quality. As it is impossible to visualize the variation of the quality as a function of all the five parameters, we chose to visualize on a set of 3-D figures in which we varied two parameters and kept the other three fixed. The MOS value is computed as a function of the values of all the five parameters by using the RNN tool described in the previous Section. It should be noted that the axis of these figures are rotated in such a way to give the best visualization of each figure. This means that the orientation of some axis can vary from figure to figure. In addition, the scale of quality axis (or Z-axis) is variable from one figure to another to show some useful features of the given figure. In the following subsections, we analyze the effect of each parameter on the quality and we also discuss their main combined effects.

6.1 Bit Rate (BR)

As shown in Fig. 6 through Fig. 9, the parameter BR exhibits a great influence on the quality. From Fig. 6, MOS values, obtained by the trained RNN, vary from 3 to 6.7 for (FR=6 fps, LR=0 %, CLP=2, and RA=0.1) and from 3.7 to 7.7 when the FR goes up to 30 fps. This is for a variation of BR from 0.15 to 0.8. Its effect is comparable to the effect of LR as shown in Fig. 7. The improvement of the quality is significant for lower LR and decreases for higher LR. The quality goes from 2.8 to 7.8 for zero loss, but it only presents about 1.3 of absolute improvement for 10% LR. The fact that in Fig. 8 there is no effect of CLP on the quality comes simply from the setting LR = 0.0%. This shows how well the NN learned the problem of evaluating the quality. From Fig. 9, for higher values of BR, there is no effect of the variations of the RA parameter on the quality; however, its effect increases and becomes benefic for lower values of BR.

When the encoder has to decrease the BR of the given stream, it increases the quantization parameter in order to further compress the original image. This process increases the artifacts of the output stream, and hence increases the distortion which becomes more noticeable by humans as BR decreases. This parameter is studied in [38] and [26]; however, the interaction with other parameters is not shown.

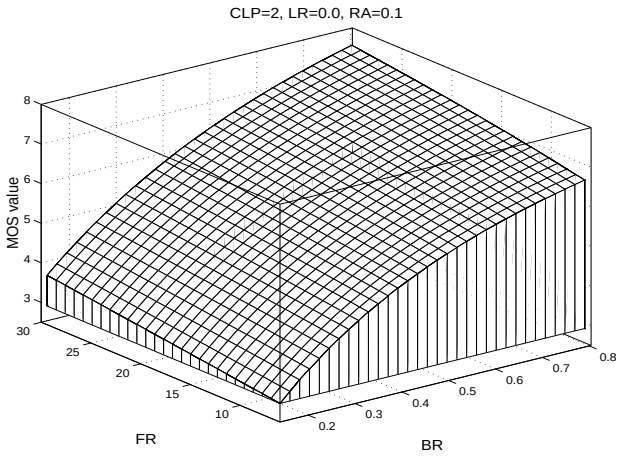

Figure 6: The impact of BR and FR on video quality.

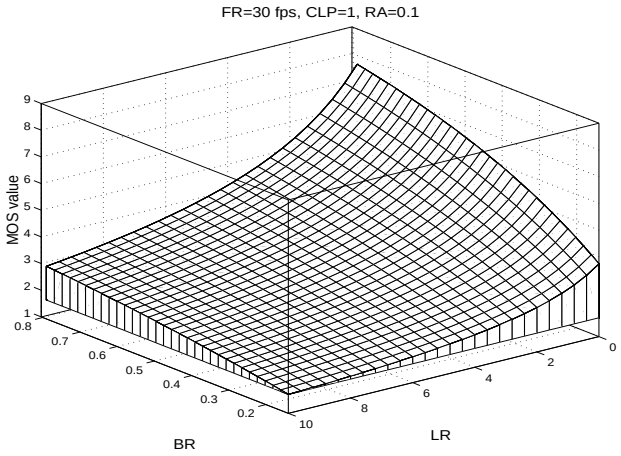

Figure 7: The impact of BR and LR on video quality.

6.2 Frame Rate (FR)

The effect of this parameter on the quality is not as significant as in the BR or LR (see below) cases. This is clearly shown in Figs. 6, 10, 11 and 12. For (BR=0.8, LR=0 %, CLP=2, and RA=0.1), an improvement from 6.6 up to 7.6 is achieved for a wide range variation going from 6 to 30 fps. Similarly, there is an absolute change of 0.8 for BR=0.15, as depicted in Fig. 6. We can also note that the enhancement of the quality for FR greater than 16 is negligible, as already observed in previous works [30]. As the BR decreases, the effect of FR becomes smaller, see Fig. 16.

The improvement of the quality when the LR changes from zero to 10% (MOS value varies from 6.5 up to 7.5 for FR=30 fps) decreases until it becomes negligible as shown in Fig. 10. From Fig. 12, we can see that we can get constant relative improvements of the quality whatever the value of RA (from 6.0 to 7.0 for RA=0.05 and from 6.4 to 7.3 for RA=0.5 – this is for BR=0.6, CLP=2 and LR=0 %).

These results may seem surprising, but they have been observed in different previous studies. For example, experimental results showed that for FR larger than 16 fps, the viewer is not so sensitive to changes in FR values [30]. The work done in [15] clearly shows that the enhancement of the quality for a wide range variation from 6 to 25 fps is really small. Both studies were based on subjective quality tests.

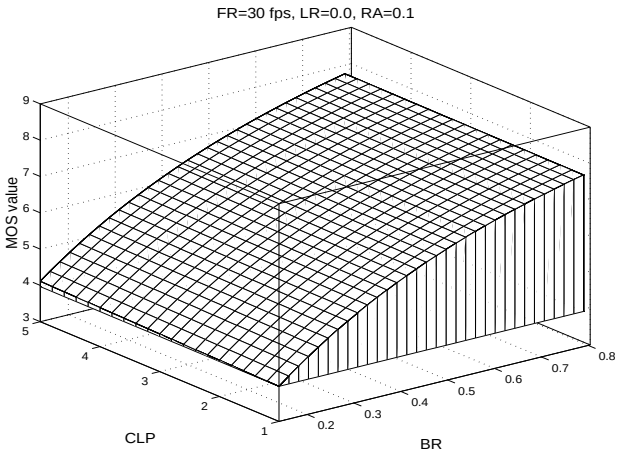

Figure 8: The impact of BR and CLP on video quality.

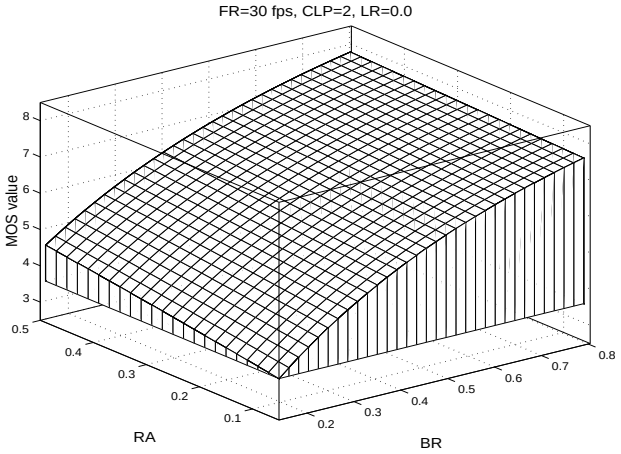

Figure 9: The impact of BR and RA on video quality.

6.3 Loss Rate (LR)

The effect of LR on video quality was the main goal of several previous studies as it is an important parameter [6], [15], [17]. As shown in Fig. 7, the quality drastically decreases when the LR increases from 0 up to 10 %. But the absolute decrease of the quality depends on the

other parameters' values: about 1.3 for BR=0.15, while 5.0 for BR=0.8. As depicted in Fig. 10, the absolute deterioration is about 3.8 for 6 fps, but 4.6 for 30 fps. The variation of the quality is about 4.4 for RA=0.5 and 3.8 for RA=0.05, when BR=0.35, FR=30 fps and CLP=2, as shown in Fig. 14.

Again, as said before, the fact that for LR set to zero there is no change on quality when CLP varies (Figs. 8, 11 and 13), shows that our RNN captures with great precision and reliability the characteristics of the quality as a function of the parameters. It may be thought that this is due to large values of BR and FR, but we varied all these parameters and we got the same behavior. In the case of using ANN, it is not easy to get this result (see Fig. 1).

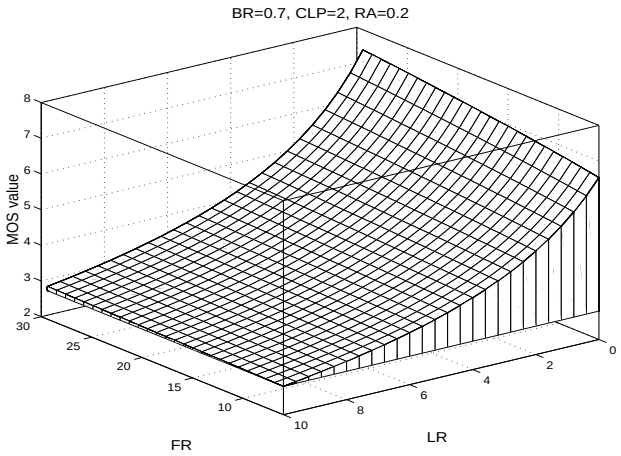

Figure 10: The impact of FR and LR on video quality.

6.4 The number of Consecutively Lost Packet (CLP)

As the result obtained for this parameter may seem strange, let us clarify its effects. When keeping all other parameters constant and increasing the value of CLP, the distance between any two consecutive loss occurrences increases. This has two consequences: loss occurrences decrease and consequently the deterioration of frames becomes smaller (i.e. smaller number of frames partially distorted by loss). As it is well known and also shown in our previous analysis of FR, the eye is less sensitive to higher values of FR. Moreover, for each lost packet the past macro-blocs (a portion of the image) are still shown on the screen as a kind of error resilience. Hence, larger values of CLP may introduce deterioration in smaller frames. This is equivalent to smaller LR values but slightly lower FR values. As it was previously shown that the effect of

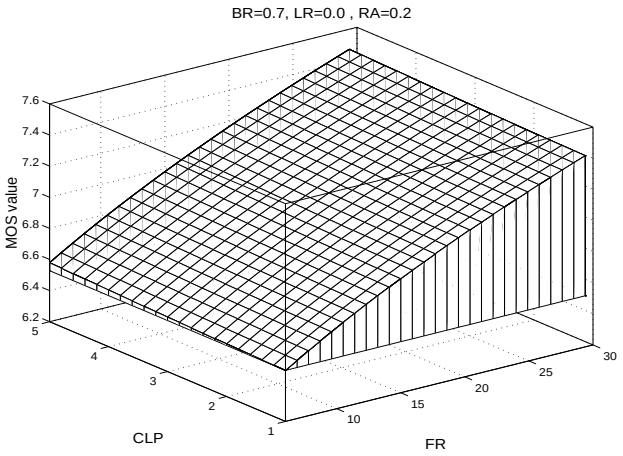

Figure 11: The impact of FR and CLP on video quality.

LR is much greater than that of FR, the effect of higher values of CLP is benefic to the quality. This result is in agreement with that obtained in [17], although the authors did not explain the reasons why this happens.

Starting by Fig. 13, and for zero LR, there is no visible effect of CLP on the quality, but as the LR increases the effect of CLP increases up to 0.5 for LR=0.0%. From Fig. 15, we can see that there is an improvement of 0.5 of the quality when CLP changes from 1 to 5 for BR=0.7, FR=30 fps and LR=1.0%. In conclusion, CLP effect is important and comparable to FR's. When CLP increases, quality increases too, particularly in the case of poor conditions (i.e. lower values of BR or higher values of LR), while increasing FR improves quality especially for good conditions (high BR and low LR).

6.5 Intra-to-Inter Ratio (RA)

RA is the ratio of the encoded intra macro-blocs to those encoded as inter, hence it is expected that this parameter has benefic effect when it increases. This is clearly shown in Figs. 9, 12, 14 and 15. Its effect is more important than FR's, and more interestingly, for lower values of the BR parameter. This is shown in Fig. 9, where for BR=0.15 the quality increased from 3.4 up to 4.6. This improvement is relatively important when the network bandwidth is small, but for larger BR, the effect is negligible.

The two parameters LR and RA are related, as shown in Fig. 14. For smaller LR, there is an improvement on the quality for the increase in RA. But this improvement vanishes when

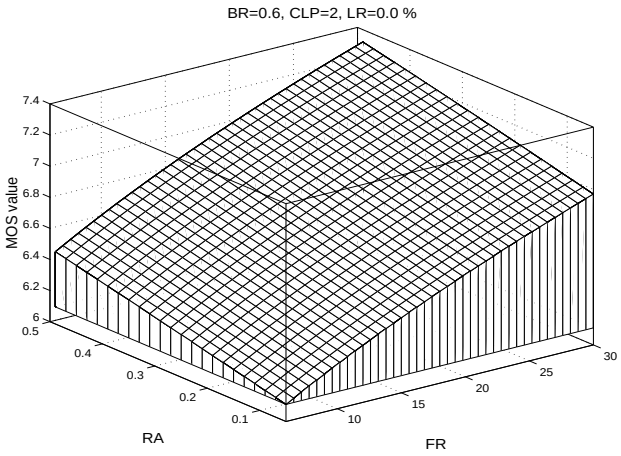

Figure 12: The impact of FR and RA on video quality.

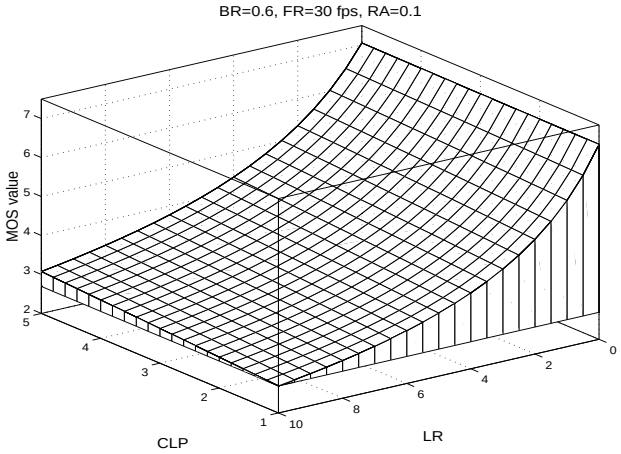

Figure 13: The impact of LR and CLP on video quality.

the LR increases. This means that we can get better video quality when the available channel bandwidth (BR) is small and for smaller values of LR by increasing the value of RA. In such a case, the effect of RA on the quality is more benefic than that of FR, as shown in Fig. 16.

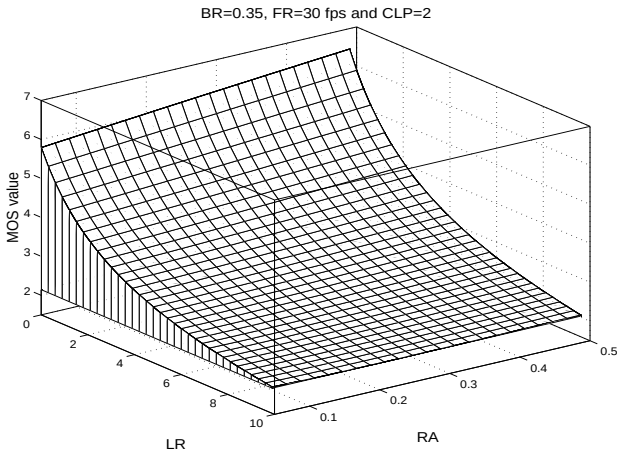

Figure 14: The impact of LR and RA on video quality.

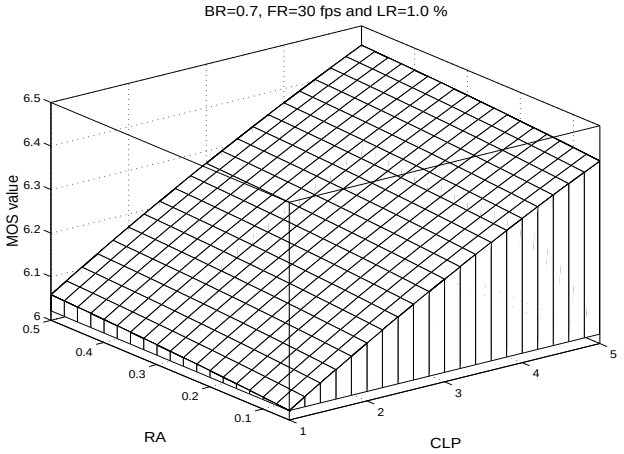

Figure 15: The impact of CLP and RA on video quality.

7 Conclusions

In this paper, we propose a methodology that is able to automatically quantify the quality of a video stream at the receiver side, after its transport over a packet network. The main properties of our technique are the following: (i) the evaluation can be done in real time, and (ii) the value

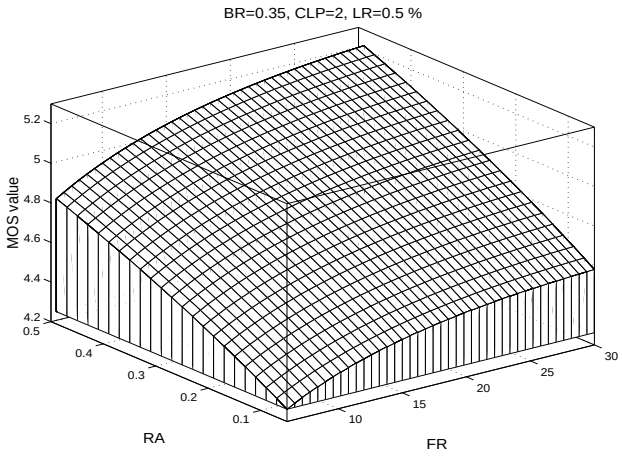

Figure 16: RA is more benefic than FR for lower values of BR.

assigned to the quality of the flow is close to the value that could be obtained from a subjective test, because it is done by a neural network that was trained to behave as the average of a set of human observers of the video stream. Once the neural network is trained, the tool continuously receives the values of the selected parameters (from measurements) to perform its evaluation.

We also used our technique to study the impact on video quality of some important parameters, namely, the stream bit rate, the scene frame rate, the network loss rate, the burst loss size and the ratio of the encoded intra to inter macro-blocs for H263 codecs. As far as we know, there is no previous study of the combined effect of several parameters on video quality. The goal of this analysis is to help in the understanding of the behavior of real-time video streams transmitted over best-effort networks. This may be used, for instance, in developing control mechanisms allowing the delivery of the best possible video quality given the current network state.

To achieve these goals, we improved our previous proposals (both in audio and video transmission) using a new and efficient neural model called Random Neural Network, which behaves significantly better than the standard one. Of course, it must be pointed out that these better performance was observed in our specific application. We do not claim that the same will hold for other applications.

Although we based our study on IP networks, our approach can be followed for ATM and wireless technologies as well (the specific type of packet technology has no effect on the relevance of the method we discuss here). Some future research directions for this work include the study of other codecs like MPEG2/4, the analysis of the effect of audio and video synchronization.

Concerning the set of selected parameters, we also intend to explore the usefulness of more sophisticated models on the different aspects that we analyze here.

References

[1] 2429, R. RTP payload format for the 1998 version of ITU-T Rec. H.263 video H.263+. In *IETF* (October 1998).

[2] Abdelbaki, A., and Gelenbe, E. Random neural network decoder for error correcting codes. In *International Joint Conference on Neural Networks* (1999), vol. 5.

[3] Abdelbaki, H. Random neural network simulator. `ftp://ftp.mathworks.com/pub/contrib/v5/nnet/rnnsimv2/`.

[4] Bakircioglu, H., and Kocak, T. Survey of random neural network applications. *European Journal of Operational Research 126* (2000).

[5] Bishop, C. *Neural Networks for Pattern Recognition.* Oxford, New York, 1995.

[6] Boyce, J., and Gaglianello, R. Packet loss effects on MPEG video sent over the public internet. In *Proc. ACM Multimedia'98* (1998).

[7] BT.500-10, R. I. Methodology for the subjective assessment of the quality of television pictures. In *International Telecommunication Union* (March 2000).

[8] Caramma, M., Lancini, R., and Marconi, M. Subjective quality evaluation of video sequences by using motion information. In *Proc. of International Conference on Image Processing, ICIP'99* (October 1999).

[9] Claypool, M., and Tanner, J. The effects of jitter on the perceptual quality of video. In *ACM Multimedia 99* (1999).

[10] Cramer, C., Gelenbe, E., and Gelenbe, P. Image and video compression. In *IEEE Potentials* (1998).

[11] Gelenbe, E. Random neural networks with negative and positive signals and product form solution. In *Neural Computation* (1989), vol. 1, pp. 502–511.

[12] Gelenbe, E. Stability of the random neural network model. In *Neural Computation* (1990), vol. 2, pp. 239–247.

[13] Gelenbe, E. Learning in the recurrent random neural network. In *Neural Computation* (1993), vol. 5, pp. 154–511.

[14] Gelenbe, E., Bakircioglu, H., and Kocak, T. Image processing with the random neural network. In *Proc. of the IEEE Digital Signal Processing Conference* (June 1997).

[15] Ghinea, G., and Thomas, J. QoS impact on user perception and understanding of multimedia video clips. In *Proc. ACM Multimedia 98* (1998).

[16] H.263, I. R. Video coding for low bit rate communication. In *International Telecommunication Union* (February 1998).

[17] Hands, D., and Wilkins, M. A study of the impact of network loss and burst size on video streaming quality and acceptability. In *Interactive Distributed Multimedia Systems and Telecommunication Services Workshop* (1999).

[18] Inazumi, Y., Horita, Y., Kotani, K., and Murai, T. Quality evaluation method considering time transition of coded video quality. In *Proc. of International Conference on Image Processing* (1999).

[19] Jones, C., and Atkinson, D. Development of opinion-based audiovisual quality models for desktop video-teleconferencing. In *International Workshop on Quality of Service* (1998).

[20] Lavington, S., Dewhurst, N., , and Ghanbari, M. The performance of layered video over an IP network. In *Packet Video Workshop* (May 2000).

[21] Lavington, S., Hagras, H., and Dewhurst, N. Using a MLP to predict packet loss during real-time video transmission. Tech. Rep. CSM329, Univ. Of Essexm, UK, July 1999.

[22] Le Leannec, F., and Guillemot, C. Packet loss resilient H.263+ compliant video coding. In *International Conference on Image Processing* (September 2000).

[23] Mohamed, S., Cervantes, F., and Afifi, H. Audio quality assessment in packet networks: an inter-subjective neural network model. In *Proc. of the 15th IEEE International Conference on Information Networking (ICOIN-15)* (January 2001).

[24] Mohamed, S., Cervantes, F., and Afifi, H. Integrating networks measurements and speech quality subjective scores for control purposes. In *Proc. of the IEEE INFOCOM'01* (April 2001).

[25] Mohamed, S., Rubino, G., Afifi, H., and Cervantes, F. Real-time video quality assessment in packet networks: A neural network model. In *Proc. International Conference on Parallel and Distibuted Processing Techniques and Applications (PDPTA'01)* (June 2001).

[26] Olsson, O., Stoppiana, M., and Baina, J. Objective methods for assessment of video quality: state of the art. *IEEE Transactions on Broadcasting 43*, 4 (December 1997).

[27] P.910, R. I. Subjective video quality assessment methods for multimedia applications. In *International Telecommunication Union* (September 1999).

[28] Pessoa, A., Falcao, A., Nishihara, R., Silva, A., and Lotufo, R. Video quality assessment using objective parameters based on image segmentation. *SMPTE Journal* (December 1999).

[29] Qiao, D., and Zheng, F. Dynamic bit-rate estimation and control for constant-quality communication of video. In *Proc. of the 3rd World of Intelligent Control and Automation* (2000).

[30] Shuaib, K., Saadawi, T., Lee, M., and Basch, B. A de-jittering scheme for the transport of MPEG-4 and MPEG-2 video over ATM. *IEEE Military Communications Conference Proc., MILCOM* (1999).

[31] Shuaib, K., Saadawi, T., Lee, M., and Basch, B. A de-jittering scheme for the transport of MPEG-4 and MPEG-2 video over ATM. In *Proceedings of IEEE Military Communications Conference - MILCOM'99* (vol. 2, pp. 1211-1215).

[32] Tan, K., and Ghanbari, M. A combinational automated MPEG video quality assessment model. In *Proc. of Image Processing and its Application Conference* (1999).

[33] Tan, W., and Zakhor, A. Multicast transmission of scalable video using receiver-driven hierarchical FEC. In *Packet Video Workshop 99* (April 1999).

[34] Vahedian, A., Frater, M., and Arnold, J. Impact of audio on subjective assessment of video quality. In *Proc. of International Conference on Image Processing* (1999).

[35] Voran, S., and Wolf, S. The development and evaluation of an objective quality assessment system that emulates human viewing panels. In *International Broadcasting Convention* (1992).

[36] Watson, A., and Sasse, M. Evaluating audio and video quality in low-cost multimedia conferencing systems. *ACM Interacting with Computers Journal 8*, 3 (1996), 255–275.

[37] Watson, A., and Sasse, M. Measuring perceived quality of speech and video in multimedia conferencing applications. In *Proc. of ACM Multimedia'98* (September 1998), pp. 55–60.

[38] Wu, H., Ferguson, T., and Qiu, B. Digital video quality evaluation using quantitative quality metrics. In *Proc. of the 4th International Conference on Signal Processing* (1998).

[39] Wu, H., Lambrecht, C., Yuen, M., and Qiu, B. Quantitative quality and impairment metrics for digitally coded images and image sequences. In *Proc. of Australian Telecommunication Networks and Applications Conference* (December 1996).

[40] Zhang, R., Regunathan, R., and Rose, K. Video coding with optimal Inter/Intra-mode switching for packet-loss resilience. *IEEE Journal on Selected Area in Communications 18*, 6 (June 2000).

[41] Zhu, W., Hou, Y., and Wang, Y. Modeling and simulation of MPEG-2 video transport over ATM networks considering the jitter effect. In *IEEE Workshop on Multimedia Signal Processing* (1997).

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399