PERSONALIZING QUALITY ASPECTS IN SCALABLE VIDEO CODING

Sam Lerouge, Robbie De Sutter, and Rik Van de Walle

Department of Electronics and Information Systems, Multimedia Lab
Ghent University - IBBT, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium
sam.lerouge@ugent.be, robbie.desutter@ugent.be, rik.vandewalle@ugent.be

ABSTRACT

In video coding, certain limitations imposed by the environment, most typically the bit rate, need to be fulfilled. This is achieved by allowing the encoder to reduce the quality in one or several ways, such as the distortion, the resolution and the frame rate. The upcoming scalable video coding mechanisms allow this reduction to take place not exclusively during the encoding step, but at any time. This allows us to reduce the quality in a more personalized way, taking into consideration the preferences of the end user. This paper presents a framework that enables such user dependent quality reductions. We validated this framework by means of a test involving 19 test persons. The results of this mechanism are good, but up to now not sufficiently reliable to use it in commercial applications. At the same time, we still see some room for improvement.

1. INTRODUCTION

For the distribution of digital video content, compression is needed because uncompressed video needs bit rates far above what is feasible regarding the speed of networks, the speed of disks, storage capacity, etc. Because lossless compression is usually insufficient to meet those constraints, lossy digital video compression algorithms are developed so that a target bit rate can be achieved by reducing the quality of the video.

The most commonly used way of reducing the number of bits needed for representing video information, is by allowing errors by means of what is usually called adaptive quantization. Sometimes, this mechanism is not sufficient: the resulting quality is unacceptably low. In these cases, one can try other means of lowering the quality combined with adaptive quantization. The spatial resolution of the images can be reduced or a number of frames can be dropped (reduction of temporal resolution).

In [1], Reed and Lim present an effective approach in which all these types of quality reduction are dynamically combined in order to achieve a higher overall quality. In general, we can say that when we want to offer an optimal overall quality, there is a need for making trade-offs between different aspects of visual quality: the temporal resolution, the spatial resolution, and the distortion of the video sequence.

With the upcoming mechanisms for scalable video coding, we will soon be able to execute the reduction of different quality aspects in a more flexible way. By simply removing specific parts of the bitstream, we can generate a reduced version of this bitstream, imposing less requirements on the terminal and the network in comparison with the original bitstream. Instead of reducing the quality during the encoding step, we can now reduce quality in real time, during the transmission of the video sequence.

A consequence of this flexibility is that we no longer have to make trade-offs between the different quality aspects during the encoding. In fact, we can even try to personalize these decisions: we can take the preferences of the end user into consideration when making this trade-off.

More formally, what we try to do is maximize multiple criteria at the same time [2]. This kind of optimization problems is more complex than classical optimization problems where only one function needs to be maximized. Often, it is not possible to find one single optimal solution. What we can do, however, is construct a Pareto frontier, which contains the set of all candidate optimal solutions.

In multimedia content distribution in constrained environments where the maximum quality cannot be achieved (e.g. because of bandwidth limitations), we want to end up with one single version that offers the best quality towards the end user by making a trade-off between several quality aspects. However, users seem to have different preferences regarding the different quality aspects, and therefore prefer different versions. In this paper, we propose a framework for selecting one optimal version from the set of candidate versions, and this in a personalized way. This is achieved

The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), the Belgian Federal Science Policy Office (BFSPO), and the European Union.

0-7803-9332-5/05/$20.00 ©2005 IEEE

by means of a machine learning approach that tries to model the visual quality preferences of the end user.

We have performed a subjective test for validating this framework. In the next section, we give a description of the setup of this test. In Sect. 3, we describe how we can deduce a model of the preferences of one particular user, based on a limited number of example decisions. In Sect. 4, we go through the results of the test. Finally, we draw some conclusions and discuss some future work.

2. TEST SETUP

In our subjective test, we try to find out if it is possible to predict the overall preferences of a certain user, based on a limited number of example preferences the user has given. The basic idea was to present the end user two versions of the same original sequence but having different characteristics, and let him decide which version he prefers. We chose to use relative comparisons in our tests instead of absolute ranking values, as used in most tests related to subjective visual quality, because it is known that users often find it easier to take relative decisions.

We used 6 different test sequences, that were all encoded using the fully scalable wavelet-based MC-EZBC (Motion Compensated Embedded Zerotree Block Coding) codec [3]. From the original, nearly lossless encoded sequence, we generated 6 different versions, using 3 different frame rates (30, 15 and 7.5 frames per second), 2 different resolutions (CIF and QCIF), each one using the same bit rate.

In each step, the application that controls the subjective test randomly chooses one of the 6 different test sequences. Then, it selects the version from that group that is currently considered to be the best version of the group (initially, this is chosen randomly). This version will be compared with one of the versions that was not shown yet. If the user finds the latter version better than the former, it becomes the best version of the group. Because we use 6 different versions in each group, the user will have to take 5 decisions for each group, or 30 decisions in the entire test.

For completeness, we want to mention that users are allowed to skip a certain decision, when they find it too hard to take. They are also allowed to replay the sequences in case of doubts.

The way we evaluate the data collected by these subjective tests is as follows. The set of actual decisions (all decisions that where not skipped by the user) is called the *data set*. This data set is split up into a *training set* and a *test set*. From the training set, we try to construct a model of the preferences of the end user. How we actually do this, is presented in the next section. During validation, we use this model for predicting the decisions that are part of the test set. The results shown in Sect. 4 are average numbers taken from the results of 19 different users.

3. MODELLING USER PREFERENCES

3.1. Introduction

Before selecting a particular algorithm for modelling user preferences, we defined the following requirements. For the model that is presented in this section, all these requirements are fulfilled.

- As we cannot expect from a user that he is willing to go through a long test for training the model, a reasonable accuracy should be achieved when only a small number of comparisons is available.
- Constructing a model from a training set should be possible in a reasonable amount of time.
- Selecting one best version from a set of candidate versions should happen in real time: the system must be able to take such decisions immediately.

The basic assumption of the model that we present in this section, is that we can describe the overall quality of a video sequence as a weighted sum of the different aspects of the quality, also called features, such as the frame rate, resolution and PSNR[1] value of the sequence. These values only depend on the sequence, but the weights are considered user-dependent. To summarize, we can describe the quality of a sequence S as follows:

$$Q(S) = \sum_{f \in F} w_f S_f; \tag{1}$$

with F the set of features considered in the model, w_f the user-dependent value of the weight for feature f, and S_f the value of feature f for sequence S.

When a user states that version A is better than version B, this is reflected in our model by means of the following inequality: $Q(A) > Q(B)$, which is equivalent to:

$$\sum_{f \in F} w_f (A_f - B_f) > 0. \tag{2}$$

Thus, each statement that is part of the training set results in an inequality. The entire training set generates a system of linear inequalities. Note that this type of inequality is equivalent to the following inequality:

$$\sum_{f \in F} w'_f (A_f - B_f) > 0; \quad w'_f = \frac{w_f}{w_0}. \tag{3}$$

In other words, we are allowed to rescale the weights with some constant value. In particular, if we select one of the weights as this value, we can remove one unknown value. This corresponds with stating that $w'_0 = 1$.

[1] The Peak Signal-to-Noise Ratio is a commonly used measure for expressing the distortion in a video sequence or an image.

We have to draw the attention to some of the weak points in this way of modelling user preferences. The most important one is that our method relies on a linear behavior of different quality features in terms of subjective quality, which is certainly not always valid. As an example, the difference in frame rate between 30 fps and 15 fps is 15 fps, but in terms of subjective quality it is much less significant than the difference between 15 fps and 7.5 fps. Similarly, but less extreme, is the case of PSNR values: the difference between 46 and 48 dB will probably be less noticeable than the difference between 32 and 34 dB.

This problem of non-linear behavior can in some cases be solved by not directly using the actual feature itself, but rather a translation of it. This is what we have done in the case of the frame rate, in two different ways, as it is presented in Sect. 3.2 and 3.3.

An additional problem is that a system of inequalities may be inconsistent. In that case, it is not possible to generate a solution. In case the system is consistent, we need to be able to select one single solution from the region of valid solutions. We used the centroid of this region, because we expect that a solution in the center of this region is more likely. This way of selecting a solution has a downside however: when the region defined by the system of inequalities is unbounded, no centroid can be determined. This case is handled in different ways, as we discuss in the following subsections. In the first two versions (Sect. 3.2 and 3.3), we ignored such a situation and didn't generate a solution. In the third version (Sect. 3.4), we solved this problem in a straightforward way.

3.2. Initial version

In a first version of the implementation of our model, we used the PSNR value of the sequence, S_{psnr}, as one feature of F in equation 1. For the resolution, S_{res} equals to 1 in the case of a QCIF sequence, and 2 in the case of a CIF sequence. For the frame rate, S_{fr} has a value of 1 if it has a frame rate of 7.5 fps, 2 in case of 15 fps, and 3 in the case of 30 fps.

As we explained when introducing Eq. 3, we are allowed to remove one weight, in this case w_{psnr}. This is also the case in the other versions that will be explained in the following sections.

3.3. Better estimation of temporal quality

In a second version, we tried to use a more reliable way of estimating the temporal visual quality (the smoothness or jerkiness of a sequence). In the previous version, the quality loss between going from 30 fps to 15 fps was considered equally important as the quality loss when going from 15 fps to 7.5 fps. In practice however, the second case is much more severe than the first case. Therefore, in this version, we used respectively 1, 4 and 5 as values for S_{fr} in the case of 7.5, 15 or 30 fps. The consequences of this modification will be discussed in Sect. 4.

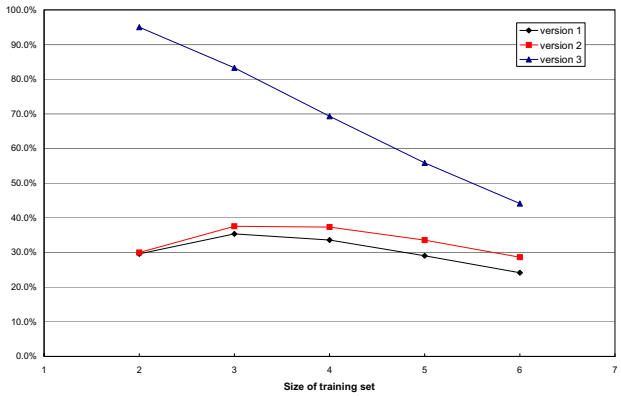

Fig. 1. Average amount of training sets generating a feasible and bounded solution region.

3.4. Avoiding unbounded regions

When analyzing the performance of the two versions of the model described in Sect. 3.2 and 3.3, we noticed that some users had unexpectedly bad results: not even half of the decisions in the test set were correctly predicted, which means that a random guess would be more reliable than using the model we just described. When investigating the numbers more closely, we noticed that these users had a high number of training sets that yielded an unbounded solution region, and therefore no weights could be deduced.

However, ending up with a system producing an unbounded region of solutions means something totally different than having an inconsistent system. In the case of an unbounded region, it means that at least one of the unknown weights (in our case, w_{res} and w_{fr}) is much more important than the weight that was assigned a value of 1 (in our case, w_{psnr}). Instead of ignoring this case, we should be capable of incorporating this information in our model.

The easiest way to do this, is to add upper bounds to the values of the unknown weights. A limited number of tests on a subset of 5 out of the 19 users indicated that limiting each weight to a value of 8 produced the best results.

4. RESULTS

First of all, we observe the influence of using a rather simple quality metric with one that has a better correlation with the subjective observations of the user. In particular, we look at the difference between a very simple way of describing the temporal quality and a more advanced estimation, as described in Sect. 3.2 (version 1) and 3.3 (version 2).

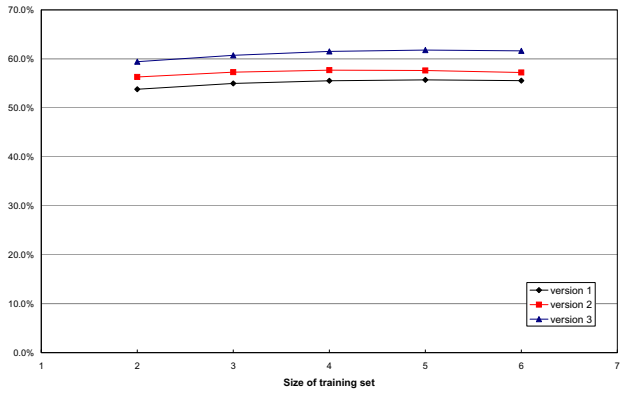

Fig. 2. Average amount of decisions in the test set that were correctly predicted from the training set.

When looking at Fig. 1, in which the average amount of feasible systems of inequalities (these are the systems that produce a bounded region of solutions) is shown in function of the size of the training set, we clearly see that the amount of feasible systems is higher when using a more accurate estimation of the temporal quality. Note that this number first increases, because the number of unbounded regions is decreasing, but then decreases, because there are more systems that are inconsistent.

When we can avoid the problem of unbounded regions that cannot produce a model, by introducing upper bounds on the possible weight values (version 3), a significantly larger amount of training sets will produce a solution, as can be seen in Fig. 1. This will also have a positive effect on the reliability of our model.

Figure 2 shows the average reliability of the different versions of the model, in function of the size of the training set. It is interesting to note that for all versions, from a certain point, increasing the training set no longer improves the reliability of the model, possibly because of the increasing number of inconsistent training sets.

Note that the results of Fig. 2 show the average reliability, this is what we can expect when we use a randomly selected training set. In the future, we will need to investigate if we can compose one particular training set that has a higher accuracy than average for the vast majority of the users.

We also see in Fig. 2 that using a more reliable estimation of temporal quality improves the accuracy of our model. Still, a maximum accuracy of 57.7% when using a training set of size 4 is not sufficient to make use of this model in real life applications. Fortunately, using the information of training sets producing unbounded regions by means of upper bounds, realizes an additional improvement in the reliability of our framework: now we achieve a maximum accuracy of 61.8%.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a framework for adapting the quality of a scalably coded video sequence in a user-centric approach, taking into account his preferences. These preferences are captured by means of pair-wise comparisons, and are used for building a model that can be used for future predictions of preferences.

This model is validated by means of a subjective test, in which 19 test persons had to express their preferences by means of 30 pair-wise comparisons that are used for building a model and for validating its reliability.

From this test, we can conclude that the reliability of our approach is acceptable, but needs further improvement before it is ready for use in real life applications. This improvement can be achieved by using more accurate estimations of the aspects of visual quality. We have shown that we can make our model more accurate by modifying the way temporal visual quality is expressed.

We are convinced that further improvements using this approach are still possible. In the first place, it is known that PSNR is not very reliable in predicting the visual quality of an image. More advanced methods taking into consideration the properties of the Human Visual System, have a better correlation with subjective user ratings, as reported by the Video Quality Experts Group [4].

Similarly, a more accurate estimation of the temporal visual quality should be possible. In high motion sequences, reducing the frame rate is more disturbing than in sequences with low motion. Unfortunately, to this day, no metric exists that can capture the temporal quality of a video sequence more accurately than the frame rate itself. We think that such a measure would improve the reliability of our model.

6. REFERENCES

[1] Eric C. Reed and Jae S. Lim, "Optimal multidimensional bit-rate control for video communication," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 873 – 885, Aug. 2002.

[2] Sam Lerouge, Peter Lambert, and Rik Van de Walle, "Multi-criteria optimization for scalable bitstreams," in *Visual Content Processing and Representation, 8th International Workshop VLBV 2003*, Sept. 2003, vol. 2849 of *Lecture Notes in Computer Science*.

[3] Shih-Ta Hsiang and John W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank," *Signal Processing: Image Communication*, vol. 16, no. 8, pp. 705–724, May 2001.

[4] "Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment, phase II," Tech. Rep., VQEG, Aug. 2003.