A Content-Based Optimization of Data Stream Television Multiplex

Jaroslav Polec, Martin Šimek, Michal Martinovič, Elena Šikudová

Abstract—The television multiplex has reserved capacity and therefore we can use only limited numbers of videos for propagation of it. Appropriate composition of the multiplex has a major impact on how many videos is spread by multiplex. Therefore in this paper is designed a simple algorithm to optimize capacity utilization multiplex. Significant impact on the number of programs in the multiplex has also the fact from which programs is composed. Content of multiplex can be movies, news, sport, animated stories, documentaries, etc. These types have their own specific characteristics that affect their resulting data stream. In this paper is also done an impact analysis of the composition of the multiplex to use its capacity by video content.

Keywords—Multiplex, content, group of pictures, frame, capacity.

I. INTRODUCTION

Dynamically allocating transmission bandwidth between multiple video streams is one of the major challenges of broadcast systems such as digital video broadcasting (DVB) statistical multiplexing [8] based on joint control of multiple video coders offers a good-quality smoothing solution in this situation. Additionally, an attempt is made to maximize the number of video streams sharing the same transmission channel while facing multiple simultaneous constraints. These are related to bandwidth usage, minimum required video quality, smooth quality transitions over time, and balanced quality between streams [9].

The rate and distortion (R-D) characteristics of the transmitted streams are known as useful tools for solving the resulting constrained optimization problem. Efficient R-D characteristics represent the real behavior of the encoder and so provide a robust control [10].

In [6] the encoder rates are continually adapted. A rate controller determines the channel rate for each encoder as well as the target rate allocated to individual pictures encoded by the encoders. The target rates for individual pictures are computed in order to maintain constant quality across encoders while the encoder channel rates are derived so as to avoid encoder and decoder buffer underflow and overflow.

A lexicographic framework for bit allocation to the multiplexing of multiple VBR video streams onto a single CBR channel is realized in [7]. In the lexicographic framework, the maximum distortion is minimized, then the second highest distortion, and so on, resulting in nearly constant quality. With a suitably constructed multiplexing model, we show that the multiplexing problem reduces to a single-stream CBR bit allocation problem, to which we apply the lexicographic framework. This method has applications for video servers, especially for near-video-on-demand.

In [11] is presented a bandwidth allocation scheme based on content complexity to equalize the overall video quality, in effect a form of statistical multiplexing. Complexity metrics serve to estimate the appropriate bandwidth share for each stream, prior to distribution over a wireless access network. These metrics are derived after entropy decoding of the input compressed bit-streams, without the delay resulting from a full decode. Fuzzy logic control serves to adjust the balance between temporal and complexity metrics. The paper examines constant and varying bandwidth scenarios. Experimental results show a significant overall gain in video quality in comparison to a fixed bandwidth allocation.

In [12] is used Wavelet transform for characterizing measured Ethernet traffic. Research is concentrated on self-similarity presence and show that this phenomenon can be found in measured data.

Those contributions do not deal with position of I, P, B frames in the video. It also does not deal with the mutual position of I frames between videos in multiplex. If we consider the optimization of videos bit rates in multiplex using VBR is strongly dependent on the desired video quality, it is necessary to optimize the mutual position of I frames in a multiplex. Based on the assumption that for the television is in the most cases prescribed uniform length GOP, it is not possible to use any of optimizations that determine the location of frame into the place of sharp cut. I frames are therefore placed at specified times. And these positions have to be optimized in order to avoid either the large fluctuations in the resulting bit rate of multiplexor to fluctuations in the quality of videos of I frames and shortly thereafter.

The optimization of the total flow is also dependent on the movement and activities of spectral images in a multiplex.

A. Extracting Semantics from Video

The semantic information is an important clue in content-based video summarization and subsequent retrieval. With the semantic gap still being unsolved, the research in this area is...
very vivid. In videos different modalities can help to bridge the semantic gap. These modalities include visual, audio or textual clues, motion. A survey of different fusion technologies of these modalities can be found in [14]. In [15] is created a system that explores complementary modalities of Visual-Color, Visual-Region, Visual-Texture, Audio-Perceptual, Audio-Cepstral and Textual. Low-level features used to extract semantic information include average brightness, dominant color, Local Binary Patterns (LBP), color layout, FFT, shape recognition, color histograms, optical flow, etc. Medium-level features utilized in determination of the semantic information are face detection, motion detection, human detection, salient region detection, speech recognition, speaker identification and gesture detection. Among the best performing detection/recognition methods are SVM, HMM, bag of (visual) words and Bayesian inference.

Results of semantic categories detection are still not satisfactory. In TRECVid 2006 the highest ranking semantic concepts were Weather (~0.65), Sports and Maps (~0.5) [16]. On the other hand the shot boundary detection task in TRECVid 2005 resulted in Precision/Recall of 0.936/0.949 for cuts and 0.788/0.791 for gradual transitions [17].

II. PROPOSED ALGORITHM

Decoder has to get frame I to starting of video play. Frame I is many times bigger than P and B frames and if we sent at the same time frame I, we would unnecessarily waste bandwidth and in worse case we would not have even enough bandwidth. This can be easily avoided by individual video delay of several frames and in order to prevent of frames I overlapping in multiplex.

Algorithm in Fig. 1 works on a very simple principle: At the beginning of algorithm we take the first video without shift. Also we take another video with initial shift set to 0 (i.e. without shift). The following procedure is counting of bit rates in each time and finding maximum of them and this maximum is saved. Next step is increasing the shift by 1 and repeating the previous process (finding a maximum and then comparing with previous one). If current maximum is smaller than previous one, we will record only current maximum as reference value for the next steps and also in which shift is achieved this maximum. Shift is increased again, until the set value of maximum shift is achieved. In our case it is 30. We chose this threshold because the GOP size for DVB-T with H.264 encoder is also a size 30 and shifts are searched like this for its entire length. If GOP takes just over one second it run 25 frames per second and contains 30 frames. If we increase this value, we will achieve a lower bit rate, but at the expense of delay (it could be a very unpleasant especially for sport events: our neighbor watch non delayed video, but our multiplex is delayed for example about three second, so our neighbor would have shouted goal and we would still watch only beginning of goal situation). After achieving the last shift is remembered last maximum and shift in which is occurred this maximum and the next video is taken. This process is repeated until the entire multiplex is filled by the number of streams.

III. EXPERIMENT

The experiment is realized on a series of videos from Technical University of Munich [1], which are freely available via ftp server for non-commercial purposes. The database contains movies in standard and high definition. Videos are downloadable in the form images (TIFF format). It allows lossless storage. Specifically, these following video sequences:

- carving3 – approximating shot of shop with two people inside
- cc1-cc4 – file contains a video with mass start of cross country skiers captured by lateral camera, process of cross-country skiing captured by back camera and with zoomed camera, camera is static in all three cases, camera of the last video captures skiers from moving car
- forest1 – forest in winter, captured from a moving car, the rising sun from trees
- soccer1 – ball on the field, camera moves gradually up with move of player, in the beginning is the detailed
shot and gradually to the end at length

- soccer2 – the camera is followed the ball, which is kicking by goalkeeper
- soccer3 – camera is placed behind football goal and it shoots the attack
- soccer4 – tilted camera follows the training of football players
- turn1-3 – slowly approaching and isolate shot on the Technical University in Munich and its surrounding captured by above camera
- okt1 – Shots from Oktoberfest which capture carousels at night, lots of lights and colors
- okt3 – shot on balloons and crowd of people
- okt4 – passing people are captured by static camera
- okt5 – shot of carousel over the day

All of these video sequences are in full HD (1080 lines and color subsampled by 3:1:1 system). We converted them by x.264, using color subsampling 4:2:0 and 4:2:2. We recorded output values of PSNR for each components Y, U, V from both types and the resulting bit stream of each video. Brightness values Y were with these ways almost the same. It results from way of subsampling, because only color components are reduced (not brightness). So we display only PSNR for U, V components and bit complexity. All of these values are shown in Table I.

### Table I

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4:2:0</td>
<td></td>
<td></td>
<td>4:2:2</td>
<td></td>
</tr>
<tr>
<td>cc1</td>
<td>45,875</td>
<td>22,261</td>
<td>43,485</td>
<td>38,801</td>
</tr>
<tr>
<td>cc2</td>
<td>44,343</td>
<td>23,315</td>
<td>42,701</td>
<td>44,087</td>
</tr>
<tr>
<td>cc3</td>
<td>43,399</td>
<td>27,412</td>
<td>42,064</td>
<td>44,407</td>
</tr>
<tr>
<td>cc4</td>
<td>47,508</td>
<td>24,390</td>
<td>44,015</td>
<td>44,627</td>
</tr>
<tr>
<td>forest</td>
<td>45,595</td>
<td>46,173</td>
<td>42,995</td>
<td>44,305</td>
</tr>
<tr>
<td>soccer1</td>
<td>44,644</td>
<td>26,741</td>
<td>43,174</td>
<td>46,116</td>
</tr>
<tr>
<td>soccer2</td>
<td>46,680</td>
<td>48,602</td>
<td>44,052</td>
<td>46,517</td>
</tr>
<tr>
<td>soccer3</td>
<td>47,127</td>
<td>48,140</td>
<td>44,751</td>
<td>46,365</td>
</tr>
<tr>
<td>soccer4</td>
<td>46,557</td>
<td>48,035</td>
<td>44,593</td>
<td>47,862</td>
</tr>
<tr>
<td>okt1</td>
<td>45,076</td>
<td>47,163</td>
<td>45,411</td>
<td>47,541</td>
</tr>
<tr>
<td>okt3</td>
<td>48,257</td>
<td>48,441</td>
<td>48,871</td>
<td>48,919</td>
</tr>
<tr>
<td>okt5</td>
<td>45,481</td>
<td>46,417</td>
<td>46,141</td>
<td>47,203</td>
</tr>
</tbody>
</table>

As we can see in Table I, the difference of values for PSNR of the color components is in the range from -0.786 dB to 3.493 dB and the average percentage increase in the bit stream is only from 6% to 14%. Due to this fact in the next part of paper, we decided to create multiplexes only of video sequences that have better quality, because they are subsampling by system 4:2:2. Besides these videos we used further videos:

- ControlledBurn – captures firefighters who sprayed water on the tree and subsequent there is controlled fire of house (burned to the ground) [2]

### A. Multiplex Capacity

According to [5] we can transfer 38.47 Mbit/s payload data stream in one 8 MHz original analogue TV channel. The question is: how many programs can fit into such a capacity. Therefore, we first randomly selected different video sequences. Then we created multiplex. As we can see in Table II, the maximum value of the bit stream using an optimization algorithm for the 8 programs in a multiplex is about 36.5 Mbps. It is only slightly less than the total capacity of the multiplex. This value is however only a clear video stream without sound or teletext. After adding these components, which is 0.5 Mbps on one program, we get the value 40.5 Mbps and it is inconvenient for our multiplex.

Based on Table II, when we gradually reduce the number of programs in the multiplex, we can reach the maximum number of videos equal to 5 for multiplex without shift and the maximum number of videos is equal to 7 for multiplex with optimum shift.

### Table II

<table>
<thead>
<tr>
<th>Number of videos in multiplex</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum [Mbps]</td>
<td>11,4</td>
<td>18,6</td>
<td>22,9</td>
<td>26,9</td>
<td>28,6</td>
<td>36,5</td>
</tr>
</tbody>
</table>

| without shifts/   | 22,9 | 28,3 | 34,1 | 42,1 | 45,0 | 62,0 |
| optimal shifts/   | 152,4 | 148,5 | 156,1 | 157,1 | 169,9 |

### B. Multiplex Analysis

In this section we analyze multiplexes by the content of programs in a multiplex. We created programatically same multiplexes: with slow movement, animated and sports. We created multiple-optimized data flow in the end.

Multiplex of 7 slow, sometimes almost static images, is composed of videos: SnowMnt, ControlledBurn, WestWindEasy, carving3, Aspen okt4. Videos are described above.

The resulting data stream is shown in Fig. 2 and statistics are given in Table III.
The multiplex is made up of multiple slow scenes or scenes where the only movement is caused by movement of the camera, one would expect that this type will be the least difficult to bit stream. But the opposite is true. The maximum bit rate is 30.45 Mbps compared to 24.49 Mbps for animations (Table IV). Also the average value is 15.29 Mbps to 9.2 Mbps, which is quite a significant difference. It is caused by wide of spectrum of animated videos. Even very slow scenes are more difficult to bandwidth to comparison with very fast animated scenes.

In the next section, we used the before mentioned two animated movies. We cut them to sequences of size 20 seconds. Selected sequences are calmer but also with very demanding movement.

The resulting data stream is shown in Fig. 3 and statistics are given in Table IV.

Animated images are significantly different from the real images. They are unique, because their color range is not as wide as in the real shots. It is the reason why the bit complexity is much smaller too. It is the true also in slow scenes, as well as very fast demanding scenes and also the fluctuations in bit rate are lower. It can be seen in Table IV. Maximum bit complexity of multiplex with 7 videos without shift is only 37.9 Mbps. It is significantly less than 52.25 Mbps for sports multiplex. In the case of optimal shift, the maximum bit complexity is only 24.49 Mbps for animated programs. It is the lowest value of all studied types of videos. Composition of multiplex with 7 animated programs therefore very inefficient uses available bandwidth.

Multiplex for 7 sports programs consists of the videos: TouchdownPass, RedKayak, soccer2, RushFieldCuts, cc4, soccer3 and cc1 (each one is described above). These are records of American football match, from whitewater, football and the race of cross country skiers. The resulting data stream is shown in Fig. 4 and statistics are given in Table V.
Sports programs are characterized by particularly fast movement and dynamic nature of changes in the image. Of course it does not apply for all kinds of sports, such as pool, chess or darts, but mostly it is true. Similarly, it also applies to all sports videos, we worked with. Therefore, the maximum bit rate is greater than for other programs. In this case value for optimal shift is 37.75 Mbps, which exceeded our capacity. If we formed multiplex of 7 sports videos, it would often be at the expense of quality.

By a suitable composition of multiplex by content can be achieved maximum efficiency while maintaining bandwidth utilization claims to the video quality in all programs of multiplex.

Multiplex of 7 different videos consists of videos: tum1, cc1, okt1, aspen, forest1, RushFieldCuts a BigBuckBunny. Every mentioned video is described above.

An important think is, that it contains videos with different complexity on the data flow:
- With minimal movement (tum1 and aspen)
- With a lot of small movement (okt1 and RushFieldCuts)
- With a big changes in movement (cc1 and forest)
- Animated movie in form of BigBuckBunny.

The resulting data stream is shown in Fig. 5 and statistics are given in Table VI.

As we can see, we get the maximum bit rate 45 Mbps without shift with the number 7 videos and it greatly exceeds our capacity. But when we use the optimal shift of frames, we can get only 28.6 Mbps, which is saving. After adding remaining necessary items to multiplex we get value 32.14 Mbps.

### IV. Conclusion

In this paper we presented a simple algorithm to optimize capacity utilization multiplex. We also made analysis about impact of the multiplex composition according to video content to use its capacity.

Firstly, we created multiplex with a variable number of channels (from 3 to 8). From selected parameters of multiplex: bandwidth 8 MHz and 64 state quadrature modulation, we found that to multiplex we fit 7 videos with optimum shift and only 5 videos without shift. Using optimization algorithm, we saved a significant amount of multiplex capacity and we managed to still get into it by 2 – 3 videos more in full HD.

In next section we passed multiplexes by content: various sports, animated and slow video sequences. All of them consisted of 7 videos. In terms of the multiplex maximum bit rate sports events reached the highest value because only the video stream reached 31.75 Mbps. Second toughest was video completed of slow video sequences with 30.45 Mbps, followed by a variety of content with 28.65 Mbps and...
animated video sequences were last with only less than 24.45 Mbps. It should be noticed that all multiplexes are fitted in the desired capacity. A better view of the complexity with different types gave us average bit rate values. Sports multiplex is the most challenging again with 20.64 Mbps, but change is in other rankings: the second in order with 20.64 Mbps are video sequences, followed by slow video sequences with 15.29 Mbps and a significant difference is the smallest average bit rate for animated movies with average value only 9.2 Mbps. Sports videos were a very dynamic with great movement. It showed that they are clearly the most demanding from the given videos. Followed by videos of different content (each of video types contained it). But the most essential difference is between animated and slow movies, where it would seem that slow videos are not difficult, as there is minimum movement. Neither animations is very complex on data stream, because they contain a much smaller range of colors than any other and in addition they have a much smoother edges than real filmed videos.

We could maximize efficiency of our algorithm by supplementing mentioned algorithm for optimizing data flow by moving the GOP in the beginning of each video about

ACKNOWLEDGMENT

Research described in the paper was financially supported by the Slovak Research Grant Agency VEGA under grant No. 1/0602/11.

REFERENCES
