MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Scalable Video Streaming Based on JPEG2000 Transcoding with Adaptive Rate Control

Anthony Vetro, Derek Schwenke, toshihiko Hata, Naoki Kuwahara

TR2007-016 December 2006

Abstract

This paper describes a scalable video streaming system based on JPEG2000 with various modes of streaming. A core function of the proposed system is a low-complexity transcoding technique that adapts the quality and resolution of the scene based on available bandwidth. One key feature of this technique that is important for surveillance applications is that interesting regions of the scene are assigned higher quality than background regions. To cope with varying network conditions, we also present a rate control algorithm that adaptively transcodes stored JPEG2000 frames. The proposed algorithm is designed to improve overall quality over a uniform rate control method by increasing bandwidth utilization, while satisfying buffer constraints and maintaining consistent quality over time. Simulation results confirm the effectiveness of the proposed system and rate control algorithm in terms of both objective measures and subjective evaluation. Complexity is also evaluated.

Advances in Multimedia

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 62094, 7 pages
doi:10.1155/2007/62094

Research Article

Scalable Video Streaming Based on JPEG2000 Transcoding with Adaptive Rate Control

Anthony Vetro,[1] Derek Schwenke,[1] Toshihiko Hata,[2] and Naoki Kuwahara[2]

[1] *Mitsubishi Electric Research Labs, Cambridge, MA 02139, USA*
[2] *Mitsubishi Electric Corporation, Amagasaki, Hyogo 661-8661, Japan*

Received 1 November 2006; Accepted 19 December 2006

Recommended by Guobin (Jacky) Shen

This paper describes a scalable video streaming system based on JPEG2000 with various modes of streaming. A core function of the proposed system is a low-complexity transcoding technique that adapts the quality and resolution of the scene based on available bandwidth. One key feature of this technique that is important for surveillance applications is that interesting regions of the scene are assigned higher quality than background regions. To cope with varying network conditions, we also present a rate control algorithm that adaptively transcodes stored JPEG2000 frames. The proposed algorithm is designed to improve overall quality over a uniform rate control method by increasing bandwidth utilization, while satisfying buffer constraints and maintaining consistent quality over time. Simulation results confirm the effectiveness of the proposed system and rate control algorithm in terms of both objective measures and subjective evaluation. Complexity is also evaluated.

1. INTRODUCTION

The JPEG2000 standard [1] is becoming an increasing popular coding format for a variety of applications that require efficient storage and scalable transmission of images and video. One application that is beginning to make use of this coding format is surveillance, where such features are particularly attractive for networked cameras and digital video recorders. Besides enabling access to different quality layers and resolution levels, JPEG2000 is also capable of providing access to regions of interest, which could help to significantly alleviate bandwidth requirements while still providing high quality to important regions of the scene.

Several ROI coding techniques for JPEG2000 images have been proposed in the recent years. These methods can be classified into two categories: static and dynamic ROI coding. In static ROI coding, the ROI is selected and defined during the encoding process. Once the ROI is encoded, it can no longer be changed. Such methods include a general wavelet coefficient scale up scheme [2], the max-shift method [3], bitplane-by-bitplane shift [4], and partial significant bitplane shift [5]. The main problem with such static schemes is that they are not suitable for interactive environments in which the ROI is defined after encoding. On the other hand, in dynamic ROI coding, the ROI is defined during the decoding process or during progressive transmission. One dynamic ROI coding method is described in [6]. This method allows for the definition of ROI in an interactive environment and handles the ROI by dynamically inserting layers. However, the dynamic layer insertion in this scheme reencodes the packet header, which requires rate-distortion recalculation and is an undesirable for real-time applications.

In this paper, we describe a video surveillance system based on JPEG2000 that allows for transmission of the scene over limited bandwidth networks. In our system, an image sequence is encoded and stored as a JPEG2000 bitstream, and then the stored images are efficiently transcoded in the compressed domain using a novel low-complexity adaptation technique that replaces data packets corresponding to higher quality layers with empty packets. This technique supports ROI-based transcoding, where the ROIs are transmitted with higher quality than the background. The proposed technique overcomes the drawbacks of prior dynamic ROI methods in that it does not require the packet headers to be reencoded.

This paper also addresses the problem of rate allocation to each frame. One straightforward method, which will be referred to as uniform rate control and is used as a reference in this work, is to allocate an equal amount of rate to each frame based on available channel bandwidth. The obvious

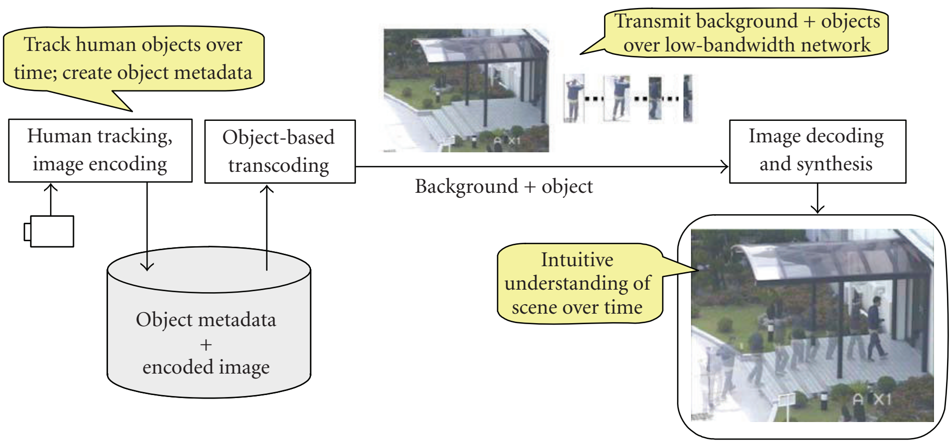


Figure 1: Object-aware streaming system for surveillance.

drawback of this method is that it is not adaptive to the scene contents. Also, since there is a fixed set of rate points that could be achieved by the transcoder, which depends on the rate allocated to each quality layer and other transcoding parameters such as output resolution level and ROI, it is very likely that the available bandwidth is not fully utilized. We propose a rate control technique that is adaptive to scene contents. The proposed rate allocation is designed to improve overall quality over the uniform rate control method by utilizing more of the available bandwidth, while satisfying buffer constraints and maintaining consistent quality over time.

The rest of this paper is organized as follows. In the next section, we provide an overview of our system including a description of the core transcoding techniques. In Section 3, the proposed rate control algorithm is presented. Experimental results are described in Section 4, including both objective and subjective evaluation, as well as complexity evaluation. Concluding remarks are given in Section 5.

2. TRANSCODING SYSTEM

In the following, we provide an overview of the key components of our object-aware surveillance system, including the main functionality of the transcoder.

2.1. Object-aware surveillance system

A diagram of the object-aware surveillance system is shown in Figure 1. The input video is analyzed and encoded. In this system, the analysis performs object detection, for example, for faces or humans. A bounding box with coordinates relative to the image coordinates is stored as the object metadata. The encoding performed using JPEG2000 and the corresponding image files are also stored. In order to enable streaming over low-bandwidth networks, object-based transcoding is employed to reduce the rate required for transmission. The operation of the transcoder will be elaborated on below. At the receiving end, the image data, which may be in the form of background and object data, is decoded and displayed.

There are various methods that the transcoded data could be streaming. In one method, which we refer to as frame-by-frame (FF), the spatial quality of ROIs and background are controlled in a frame-by-frame manner. For each region, a part of the encoded data with higher quality than a specified value is removed. FF transmits a background every frame, so the spatial quality is lower than that of the other methods described below, but changes in the background can be seen dimly. This method does not require synthesis of the decoded images for display and its implementation is simple.

A second method of streaming is to transmit the ROI successively with an occasional background refresh (BR). With this method, the transcoder mainly controls the temporal quality of ROIs considering a small occasional overhead for the background. First, ROI images and a background image with high spatial quality are transmitted. After that, only ROI images with high spatial quality are transmitted. At the receiver, the background image is decoded and held in a working memory. The successive ROI images are decoded and superimposed on the background. The background is renewed with lower frame rate according to its importance and available network bandwidth. BR does not transmit a background every frame, therefore the subjective quality is fine in low-bit rate. The overall bit rate can be significantly lowered with a low-quality background.

A third method that we introduce is mosaic streaming (MS), which is similar to BR mode, but it superimposes successive ROI images on a background in a mosaic style. Human behavior can be understood immediately and intuitively in a mosaic image, and it is very effective for behavior analysis and scene browsing as well as efficient to transmit the surveillance video over a narrow band network.

2.2. JPEG2000 transcoder

A more detailed look into the operations of the transcoder is given in Figure 2. The transcoding is invoked to satisfy

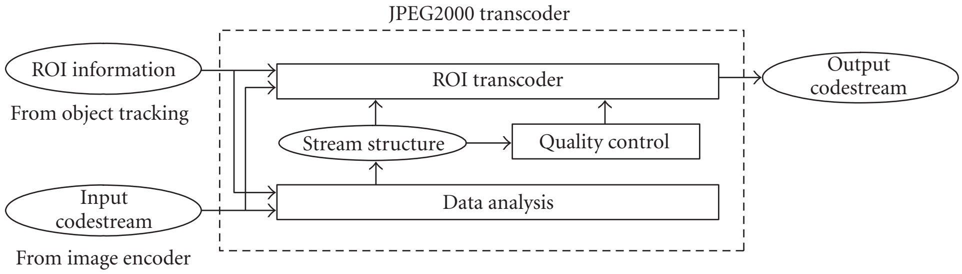


FIGURE 2: JPEG2000 transcoder.

network and display constraints, yielding an output code stream based on ROI information, and configuration settings. In the following, we describe three main components of our JPEG2000 transcoder including data analysis, ROI transcoding and quality control.

The data analysis module is responsible for extracting indexing information about the structure of the code stream. It is essentially a low-complexity parser that analyzes the packet header for each quality layer, resolution level, and component. A multiple-dimensional array is used to store the packet information, which indicates the byte position, header length and body length for each packet. Since this partial decoding operates on the packet header only without performing entropy arithmetic decoding for code blocks, the computational complexity is very low.

Our transcoder supports reduction of spatial resolution and quality layers. We focus mainly on quality layer reduction considering the ROI information. Given a set of ROI coordinates, we perform ROI transcoding by replacing packets at high-quality layers that are associated with the background of the scene with empty packets as defined by the JPEG2000 standard. This is an effective method for reducing the rate of the overall code stream while retaining the quality of important objects and keeping the complexity low.

The number of quality layers for the background and ROI are determined by the quality control module. In our previous work [7], the quality layers were set manually. In the next section, we describe an adaptive rate control algorithm that determines the quality layers based on target rate, buffer occupancy, and ROI information.

3. RATE CONTROL

The proposed rate control algorithm determines the rate allocation for the current frame based on the target rate, buffer occupancy, and ROI information. Given the bytes allocated to a frame, the transcoder determines quality layers for background and ROI. In the following, we describe the variable rate allocation, a frame skipping technique, as well as a quality stabilization algorithm.

3.1. Variable rate allocation

In the uniform rate control method, a fixed rate, $T_f = R/F$, is allocated to each frame, where R is the target rate and F is the output frame rate. To avoid overshooting the target rate, the quality layers in the transcoded output are chosen so as not to exceed the given budget. In our current system, we choose the quality layers for background and ROI in a systematic manner based on byte counts from the data analysis. We first assign the minimum quality to the background and ROI. The ROI quality is then successively increased. Finally, additional quality layers are added to the background. The main drawback of this uniform rate allocation approach is that it will typically underutilize the available bandwidth for a given stream because the quality layers can only provide a discrete set of rate points.

In the proposed rate control algorithm, we allocate rate nonuniformly to each frame and introduce a buffer to absorb the variations in allocated rate to each frame. The rate allocation to each frame is determined according to the following:

$$T_v = T_{\max} \cdot \max\left[0, \min\left(1, 1 - \alpha^2\right)\right], \tag{1}$$

where $T_{\max}$ sets the upper limit on the variable rate assigned to any frame and is $2T_f$ in our current system, and α is a buffer occupancy parameter that is a function of the buffer occupancy, B, the buffer size, B_s, and a safety margin, γ, with typical values in the range [0.05, 0.25]. The buffer occupancy parameter is given by

$$\alpha = \frac{B}{B_s \cdot (1 - \gamma)}. \tag{2}$$

When the buffer occupancy is near the upper margin, α tends towards unity, and a lower rate will be allocated to the current frame. Higher rate is allocated to the current frame when the buffer occupancy is near empty. In the next subsection, we will see how this behavior plays an important role in balancing the spatiotemporal quality tradeoff when frame skipping is employed.

3.2. Frame skipping

When frame skipping is enabled, periodic frames with no ROI defined may be skipped. The rationale behind this strategy is twofold. First, we aim to empty the buffer when there is no ROI to allow greater bandwidth for future frames that contain ROI. Second, we aim to improve the quality of the non-ROI image, which is possible since we could assign more bytes to an image sequence with a reduced frame rate.

With this strategy, frames are skipped to drive the buffer level towards its lower margin. When the buffer reaches this level, frames will no longer be skipped, and since the buffer is nearly empty, these frames are allocated a rate close to $T_{\max}$.

To state the skip condition more precisely, a frame is skipped when the following condition is true:

$$(\alpha > \gamma) \quad \& \quad (\tau < \tau_{\max}), \tag{3}$$

where τ is the interval of successive non-ROI frame skips, and $\tau_{\max}$ is the maximum frame skip interval.

3.3. Quality stabilization

The key objective of the quality stabilization is to establish a period in which the quality layers will be held stable, thereby avoiding unnecessary oscillation or frequent changes in quality. Depending on the available buffer size, the typical window period, ω, will be several frames.

Let Q_p denote the set of determined quality layers for the previous frame, Q_i the set of quality layers for the current frame i with rate allocated according to (1), and ω_c be a window counter that is reset when either the counter reaches the window period or a new set of quality layers for the current frame are determined. With quality stabilization enabled, the set of quality layers would be assigned according to

$$Q = \begin{cases} Q_p, & B_s \cdot \gamma < B < B_s \cdot (1-\gamma) \ \& \ (\omega_c < \omega), \\ Q_i, & \text{otherwise.} \end{cases} \tag{4}$$

With the above, the previous set of quality layers will be used for the current frame when the buffer is not in danger or overflow or underflow and the window counter is less than the window period.

4. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of our transcoding system and the corresponding techniques that have been proposed, we first evaluate the various streaming methods that have been introduced and discuss their pros and cons. We then describe the performance of the proposed rate control algorithm compared to uniform rate allocation. Finally, we provide an analysis of the complexity.

As input data for all experiments, we use an image sequence with 1467 frames and at a frame rate of 7.5 fps. Each frame of the image sequence is a full color image $(4:4:4)$ that is JPEG2000 encoded with 4 quality layers and 3 resolution levels with LRCP progression. The precinct sizes are set as $\{64\times64, 32\times32, 16\times16\}$, and the rate for each quality layer is set as $\{1.0, 0.5, 0.25, 0.125\}$. Each compressed frame is approximately 38 KB, yielding an overall bit-rate of 2.3 Mbps.

4.1. Evaluation of streaming methods

Various methods of streaming are illustrated in Figure 3. The original image, which is approximately 38 KB, is shown in Figure 3(a). In the following, we evaluate the quality of each setting and the required bandwidth.

Figures 3(c)–3(e) show examples of transcoded images with Qroi = 3 and Qbg = 0, 1, 2, respectively. The image with Qbg = 0 has very noticeable degradation in the background including degradation around the precinct boundaries. The data size is 9.4 KB or 25% of the input image size. The quality with Qbg = 1 is not so good, but better than that of Qbg = 0. The data size of this result is 13.0 KB or 34% of the original size. In the image with Qbg = 2, the background looks a little less sharp than the original and it is hardly noticeable. The data size is 20.9 KB, which corresponds to 55% of the original size. Observing that the transcoded image sequences are moving pictures, visual changes over time in the background with Qbg = 0 and 1 are very noticeable.

Figure 3(b) shows an example of ROI image with Qroi = 3 that is used in BR and MS streaming modes. The data size is only 6.0 KB or 16% of the original. It is very useful for a mobile phone because of its narrowband transmission and low-resolution display. The frame rate of the background depends on its importance and the available bandwidth. For example, when the frame rate of the ROI is 7.5 fps and that of background is 1 fps, the total bit rate is 644 Kbps or 29% of the original image sequence.

Figure 3(f) shows an example of a mosaic image in which ten ROIs with 1.5 fps are superimposed on the background. The walking trajectory is understood intuitively and immediately.

4.2. Rate control simulations

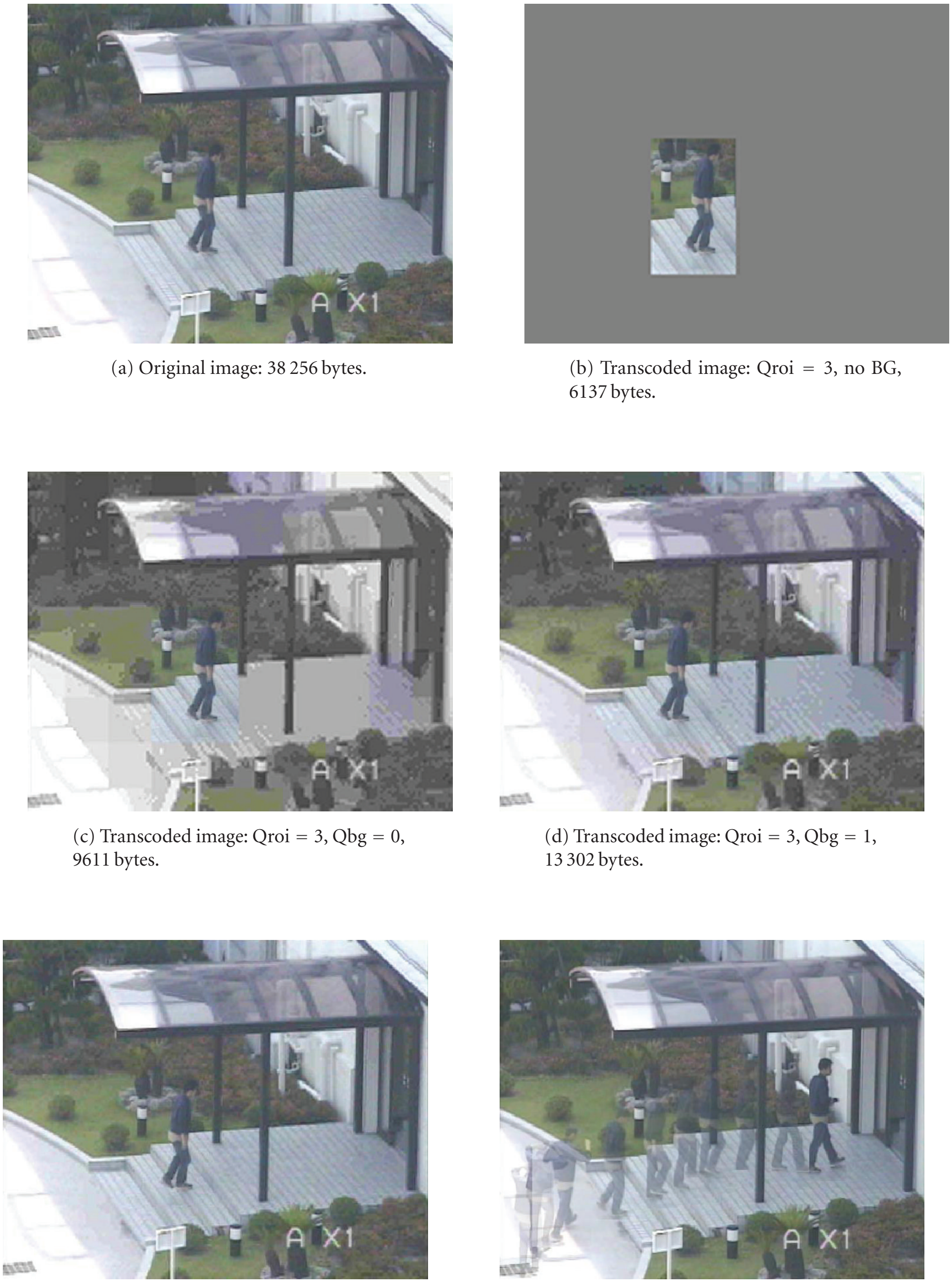
To test the performance of the proposed adaptive rate control algorithm, we perform a number of experiments with varying configurations and buffer size. We evaluate both objective and subjective quality and use the uniform rate control method as a benchmark.

For the purpose of this study, we define the following objective measures.

(i) BWU: bandwidth utilization defined as the ratio of transcoded output bits to the target rate.
(ii) ΔP0: number of changes in background quality.
(iii) Avg0: average background quality.
(iv) Avg1: average ROI quality.

As one would expect, achieving higher bandwidth utilization will generally increase overall quality. Also, minimizing the fluctuation in quality layers over time also has a positive impact on perceptual quality. It is noted that the average ROI quality is not as relevant as the average background quality since the ROI typically receives high quality regardless of the rate control method or algorithms used. In addition to the above metrics, we also report the number of frames skipped and MSE. It is noted that the MSE for skipped frames is computed assuming a zero-order hold, that is, based on the previously coded frame.

In our first experiment, we test the effectiveness of the proposed rate control components. The input code stream is transcoded to a target bit-rate of 800 kbps using the following transcoding methods: uniform rate control (URC), adaptive rate control (ARC) with variable rate allocation, ARC with



(a) Original image: 38 256 bytes.

(b) Transcoded image: Qroi = 3, no BG, 6137 bytes.

(c) Transcoded image: Qroi = 3, Qbg = 0, 9611 bytes.

(d) Transcoded image: Qroi = 3, Qbg = 1, 13 302 bytes.

(e) Transcoded image: Qroi = 3, Qbg = 2, 21 432 bytes.

(f) Mosaic image: 1.5 fps, object number = 10.

Figure 3: Example images corresponding to various ROI-based streaming methods and parameter settings.

$\omega = 4$ for quality stabilization, ARC with $\tau_{\max} = 7$ for frame skipping, and ARC with both quality stabilization and frame skipping enabled. In all simulations for ARC in this experiment, the buffer size is set to 1 MB.

The results of this first experiment are summarized in Table 1. From the table, we observe that the bandwidth utilization and quality of both background and ROI using URC is relatively low compared with ARC. While the overall quality of ARC is clearly higher than that of the URC method in terms of quality layers and MSE, the quality of the background using ARC fluctuates significantly. Such oscillations in quality have a notable impact on quality for certain segments of the video. With the proposed quality stabilization algorithm, these fluctuations can be controlled with minimal

Table 1: Experimental results comparing URC with various ARC configurations.

Configuration	Skip	BWU	ΔP0	Avg0	Avg1	MSE
URC	N/A	0.75	140	1.7	4.0	83.4
ARC	N/A	1.00	600	2.3	4.0	41.0
ARC + W (**4**)	N/A	1.00	128	2.3	4.0	40.8
ARC + Skip (**7**)	491	0.84	47	2.8	4.0	28.4
ARC + Skip (**7**) + W (**4**)	491	0.84	17	2.8	4.0	28.1

Table 2: Experimental results comparing ARC + W(4) + Skip(7) with various buffer sizes.

Buffer	Skip	BWU	ΔP0	Avg0	Avg1	MSE
1 MB	491	0.84	17	2.8	4.0	28.2
512 KB	472	0.77	39	2.6	4.0	32.7
256 KB	448	0.70	65	2.5	4.0	38.7
128 KB	429	0.64	110	2.3	3.6	50.8
64 KB	417	0.61	170	2.2	3.2	59.9

change to the overall average quality. Finally, with the frame skipping enabled, we see another moderate increase in quality and fewer fluctuations in quality.

From the data, we find that frame skipping accounts for the majority of gains observed for this particular sequence. This is likely due to the relatively high percentage of non-ROI frames in the test sequence, which is a typical surveillance video. Larger differences between the skip-only and skip-with-quality stabilizations could be expected for sequences with a higher percentage of ROI frames.

In our second set of experiments, we investigate the impact of buffer size on the efficiency of the adaptive rate control algorithm. Generally speaking, larger buffers not only require more memory in a device but also increase delay. Depending on the application, limited buffers or strict requirements on the delay may be imposed. Using the same image data and target bit-rates, the ARC algorithm with quality stabilization and frame skipping is simulated with varying buffer sizes from 1 MB to 64 KB.

The results of the second experiment are summarized in Table 2. As expected, we see a slight decline in performance with reduced buffer sizes. With smaller buffer sizes, we observe that the bandwidth utilization is decreased and hence the average quality becomes lower. Reduced buffer sizes also constrain the effect of frame skipping, that is, reducing the average number of bits for a coded frame and lowering overall quality. It is noted that even with reduced buffer sizes, the ARC method still outperforms URC in terms of average overall quality. The most significant gains will be obtained with larger buffer sizes though.

Extensive subjective evaluation has been carried out. The results reveal that the proposed ARC algorithm offers substantial improvement over URC when either quality stabilization and/or frame skipping are enabled. Without at least one of these options, frequent fluctuations in quality occur contributing to an overall decrease in subjective quality. Furthermore, for this particular sequence tested, it has been found that ARC with quality stabilization and a large buffer size is subjectively similar to ARC with frame skipping and small buffer size. Therefore, the skip only option is preferred for low-delay applications.

4.3. Complexity analysis

In order to demonstrate the computational efficiency of the proposed transcoder technique, we provide a run-time analysis of the processes including a breakdown of major components. The simulations are performed using a notebook PC with Mobile Pentium 1.6 GHz, 1 GB memory and Windows XP. The software is written in C and not optimized for performance or to a particular platform.

Table 3: Processing time (Mobile Pentium 1.6 GHz, 1 GB memory, WindowsXP).

Methods	Transcoding (ms)		Decoding (ms)	Display (ms)
	Data anal.	*ROI trans.*		
FF	9.5	0.2	39.5	7.9
BR	9.5	0.1	33.2	7.9

Table 3 shows average time per frame for the data analysis, ROI transcoding, decoding and display, respectively. In our simulations, we used several different ROI settings and different parameter configurations. However, since the vast majority of the total processing is due to the data analysis function, which is independent of the configuration settings, we report a single set of results. From these results, we observe that it takes only 9.7 milliseconds to transcode an image on average. It is clear that the majority of processing is due to the data analysis operation, which includes tag-tree decoding and calculation of the packet body lengths. A significant amount of processing is also due to memory allocation and free operations, which could easily be brought outside the main library routines to improve overall efficiency. Finally, we note that the ROI transcode operation is a very small portion of the total processing time.

It should be clear from these results that the proposed transcoding techniques are suitable for implementation on very low-cost processors and that further optimization of the memory handling and bit I/O is possible.

5. CONCLUDING REMARKS

This paper presented a scalable video streaming system based on JPEG2000 that is oriented towards surveillance applications. Several streaming methods were introduced and an adaptive rate control algorithm for JPEG2000 transcoding was presented. The algorithm allocates rate to each frame in an image sequence based on target rate, buffer occupancy, and ROI information. The key components of the proposed rate control algorithm include variable rate allocation, frame skipping, and quality stabilization. The benefits of these components have been studied and it has been shown that the proposed algorithm significantly outperforms the reference uniform rate control method. It has also been shown that the complexity of the proposed transcoding technique is very low and suitable for implementation on embedded processors.

In terms of future work, we believe there is still opportunity to improve these results of the rate control further, especially for applications that require a limited buffer size. Another interesting topic to explore is to maximize the perceptual quality considering quality fluctuation and frame skipping in nonbackground frames. Finally, for the BR streaming mode in which a refresh of the background is sent occasionally, an automated method of determining the need for a background update would be desirable, for example, based on scene information such as long term changes in illumination or objects in the scene.

REFERENCES

[1] ISO/IEC 15444-1, "Information technology—JPEG 2000 image coding system—part 1: core coding system," 2000.

[2] E. Atsumi and N. Farvardin, "Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees," in *Proceedings of International Conference on Image Processing (ICIP '98)*, vol. 1, pp. 87–91, Chicago, Ill, USA, October 1998.

[3] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[4] Z. Wang and A. C. Bovik, "Bitplane-by-bitplane shift (BbBShift)—a suggestion for JPEG 2000 region of interest coding," *IEEE Signal Processing Letters*, vol. 9, no. 5, pp. 160–162, 2002.

[5] L. Liu and G. Fan, "A new JPEG 2000 region-of-interest image coding method: partial significant bitplanes shift," *IEEE Signal Processing Letters*, vol. 10, no. 2, pp. 35–38, 2003.

[6] R. Rosenbaum and H. Schumann, "Flexible, dynamic and compliant region of interest coding in JPEG 2000," in *Proceedings of International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 101–104, Rochester, NY, USA, September 2002.

[7] T. Hata, N. Kuwahara, T. Nozawa, D. L. Schwenke, and A. Vetro, "Surveillance system with object-aware video transcoder," in *Proceedings of 7th IEEE Workshop on Multimedia Signal Processing (MMSP '05)*, pp. 1–4, Shanghai, China, November 2005.