COMPLEXITY-QUALITY ANALYSIS OF MPEG-2 TO MPEG-4 TRANSCODING ARCHITECTURES

Anthony Vetro[1], Toshihiko Hata[2], Naoki Kuwahara[2], and Hari Kalva[1]

[1] Mitsubishi Electric Research Laboratories, Murray Hill, NJ, USA
[2] Mitsubishi Electric Corporation, Amagasaki City, Japan

ABSTRACT

This paper presents a detailed complexity-quality analysis of various MPEG-2 to MPEG-4 transcoding architectures that perform spatial resolution reduction. The optimization of several processing components is also presented.

1. INTRODUCTION

To enable broadcast-quality video streams to be decoded and displayed on mobile devices, transcoding from MPEG-2 MP@ML to MPEG-4 Simple Profile is needed. This conversion implies a reduction in bit-rate from approximately 6Mbps to 384kbps and lower, as well as a reduction in spatial resolution from 720x480 interlace to 352x240 progressive. In [1], several low-complexity video transcoding architectures that meet these requirements have been proposed. In this paper, a detailed analysis of the complexity and quality of select architectures is presented. The optimizations of several processing components, such as the DCT down-conversion, will also be presented.

The architectures under consideration are shown in Figs. 1-3. Fig. 1 illustrates the Reference architecture, which is simply a cascaded approach that decodes, down-samples and re-encodes the video. Fig. 2 shows the proposed Intra Refresh architecture (Pro1), which compensates for various errors by converting select macroblocks to intra-coded blocks. Fig. 3 shows the proposed Partial Encoder architecture (Pro2), which is similar to the Reference architecture, but simplifies the re-encoding process by not compensating for re-quantization errors. The details of these architectures can be found in [1].

2. COMPLEXITY ANALYSIS

In our simulations, the input video was encoded with GOP parameters N=15, M=3. The transcoder drops B-frames, hence the transcoded output has a frame-rate of 10 frames/sec. Table 1 provides a comparison of the execution time using non-optimized software for the three architectures under consideration. From this analysis, it is clear that the complexity of the Reference is higher than both of the proposed architecture and that the complexity of Pro2 is between that of the Reference and Pro1. Analyzing the breakdown of the transcoding time, we find that the down-conversion process occupies a significant portion of the time for all architectures. For the Pro1 architecture, the processing contributing to MB_Conv are also time-consuming, mainly due to the DCT process. Similarly, in both the Reference and Pro2 architectures, the processes contributing to Drift_Comp consume a significant percentage of time due to both DCT and MC processes.

Based on this data, we consider optimizing the most time-consuming processes. For the Pro1 architecture, the down-conversion is performed in the DCT-domain using the filters presented in [2]. We have observed that these filters exhibit favorable properties towards reducing the complexity. For even pixel outputs, the filtering can be reduced from a 16-tap filter to a simple averaging operation. For odd pixel outputs, we can exploit the symmetric properties of the filter taps to significantly reduce the number of multiplications and additions. To optimize the common decoding functions, such as FDCT, IDCT and MC, MMX-based implementations have been used.

The results of the optimized transcoding software are shown in Table 2. From the data, we can see that a speed-up of approximately 70% in the down conversion process for the Pro1 architecture was achieved. For the overall execution time, an improvement of 26% and 9% were calculated using the MMX-based FDCT and IDCT, respectively. Overall, the total complexity of all transcoding architectures have been reduced by approximately 61%. We note that further reduction in complexity is still possible in some parts, such as the rate control, which is currently implemented with division operations. Also, we have observed that in the optimized software, VLC/VLD operations are quite time-consuming and more challenging to optimize.

3. QUALITY ANALYSIS

Due to space limitations, a detailed analysis of the quality is not presented at this time. We maintain that the quality of the Reference is highest, followed by Pro2 and Pro1. However, all architectures offer acceptable quality. In the full paper, detailed objective measures will be provided, along with a subjective assessment of the results. Also, the impact of using MV refinement techniques will be discussed.

4. CONCLUDING REMARKS

This paper presents a complexity-quality analysis of various transcoding architectures for reduced spatial resolution conversion. We believe that this analysis provides useful information for others working in this area. The proposed architectures offer acceptable alternatives to the reference architecture. As an additional contribution we provide a simplified DCT-domain down-conversion method, which reduces the time for this process by more than 70%.

REFERENCES

[1] P. Yin, A. Vetro, H. Sun and B. Liu, "Drift compensation architectures for reduced resolution transcoding," *Proc. VCIP*, San Jose, CA, Jan. 2001.

[2] A. Vetro, H. Sun, P. DaGraca, and T. Poon, "Minimum drift architectures for three-layer scalable DTV decoding," *IEEE Trans. Consumer Electronics*, Aug 1998.

0-7803-7300-6/02 $17.00 © 2002 IEEE

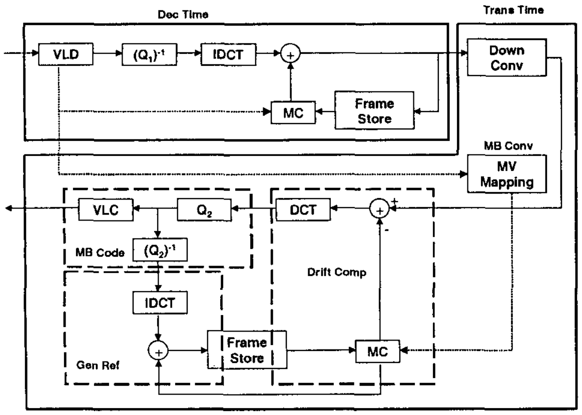

Figure 1. Reference Architecture

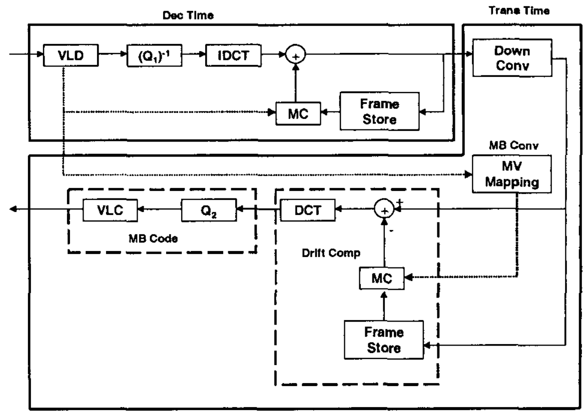

Figure 3. Partial Encode Architecture (Pro2)

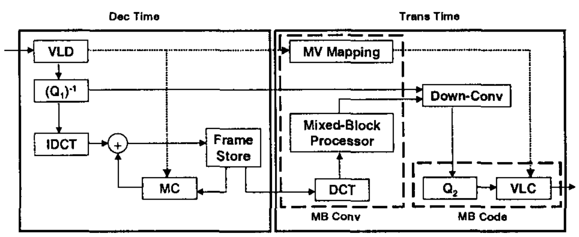

Figure 2. Intra Refresh Architecture (Pro1)

Table 1. Comparison of Complexity using Non-Optimized Code. Execution times are based on a 30s video clip. All times below are in seconds. [1.8GHz Pentium 4 Processor with 512MB Ram].

Arch	Frame Rate	Total Time	Dec Time	Trans Time	MB Conv	Down Conv	MB Code	Rate Control	Drift Comp	Gen Ref	Other
Ref	10fps	43.80	7.43	36.1	0.40 (1.1%)	10.00 (27.7%)	5.40 (14.9%)	0.88 (2.4%)	16.22 (44.9%)	1.88 (5.2%)	1.32 (3.7%)
Pro1	10fps	24.79	7.40	17.11	5.68 (33.2%)	5.38 (31.4%)	5.09 (20.5%)	0.88 (5.14%)	N/A	N/A	0.08 (0.46%)
Pro2	10fps	38.32	7.40	30.63	0.32 (0.01%)	8.90 (29.1%)	4.91 (16.0%)	0.89 (2.9%)	15.49 (50.6%)	N/A	0.11 (0.39%)

Table 2. Comparison of Complexity using Optimized Code. Execution times are based on a 30s video clip. All times below are in seconds. [1.8GHz Pentium 4 Processor with 512MB Ram].

Arch	Frame Rate	Total Time	Dec Time	Trans Time	MB Conv	Down Conv	MB Code	Rate Control	Drift Comp	Gen Ref	Other
Ref	10fps	16.78	3.68	12.85	0.02 (0.15%)	4.96 (38.6%)	3.06 (23.8%)	0.94 (7.3%)	2.24 (17.4%)	1.49 (11.6%)	0.14 (1.1%)
Pro1	10fps	9.69	3.66	5.78	0.51 (8.8%)	1.59 (27.5%)	2.68 (46.4%)	0.93 (16.1%)	N/A	N/A	0.02 (1.2%)
Pro2	10fps	14.05	3.67	10.12	0.02 (0.2%)	4.60 (45.5%)	2.69 (26.6%)	0.95 (9.4%)	1.81 (17.9%)	N/A	0.05 (0.49%)