

行政院國家科學委員會補助專題研究計畫期中報告

單晶片無線多媒體資訊家電之 設計與製作(3/3)

子計劃四：單晶片無線多媒體通訊系統

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 90-2218-E 009-014

執行期間： 90年8月1日至91年7月31日

計畫主持人：蔣迪豪 交通大學電子工程系所 副教授

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學電子工程系所

中 華 民 國 九 十 一 年 十 月 二 十 二 日

行政院國家科學委員會專題研究計畫期中報告
單晶片無線多媒體資訊家電之設計與製作(2/3)
子計劃四：單晶片無線多媒體通訊系統

Improved Error Resilient Encoding and Decoding Using MPEG-4

計畫編號：NSC 90-2218-E 009-014

執行期限：90年8月1日至91年7月31日

主持人：蔣迪豪 交通大學電子工程系所 副教授

計畫參與人員：王士豪、蘇子良、黃名彥、王俊能 交通大學電子工程系所 研究生

一、摘要

本計畫從事基於 MPEG-4 視訊壓縮標準之壓縮與解壓縮器在抗噪性與錯誤回復力方面和即時性之研究。在壓縮器方面，我們提出一個視訊內容自動 Intra 更新的方法。藉由傳輸頻道的錯誤對於視訊位元資料的影響之統計特性，我們可以在解壓縮端有效地控制錯誤資料在可回復的視訊畫面之間傳遞間隔。在解壓縮器方面，我們考慮 MPEG-4 視訊壓縮標準在語法與語意上的特徵，提出防止解壓縮器遇到錯誤位元資料時當機的方法，並以計算機與特定的頻道錯誤模型來模擬檢視其效能。除此，為了改善影像品質，我們也對回復的影像給予簡單的錯誤補償。同時，為了完成在 ARM 平台上即時性的壓縮與解壓縮器，我們也採用一些快速的演算法與高效率的記憶體存取方法來改善壓縮與解壓縮器執行效能。結合以上技術以及效能最佳化之壓縮與解壓縮器，將促使這 MPEG-4 壓縮與解壓縮器可被應用於單晶片無線多媒體資訊家電等應用。

關鍵詞：

MPEG-4, 視訊, 自動 Intra 更新, 錯誤回復力, 抗噪性, 錯誤補償, 快速(反)餘弦轉換, 快速動態偵測。

Abstract

MPEG-4 video coding standard provides applications for both Internet and mobile links. This article describes several error resilient techniques for MPEG-4 video coding under communications disruptions and quality degradation. At the encoding side, we utilize the adaptive intra refreshment (AIR) technique to make bitstream robust. The adaptive intra refreshment periodically refreshes part of reference pictures by selecting intra mode for certain macroblocks (MB). This will break the correlations to the previous reference picture

that may propagate errors that is often referred to as “drift”. At the decoder side, the first thing we do is to make the decoder resilient to any error and can finish decoding although the incoming bitstream is erroneous. We utilize error resilience tools provided by the MPEG-4 standard including resynchronize marker, data partitioning, reverse variable length code. Additionally, simple error concealment algorithms are applied to each recovered frames of improving the picture quality for playback. For SoC project, our MPEG-4 encoder and decoder are ported onto Linux platform that can be run on ARM-9 device.

Keywords:

MPEG-4, Video, MPEG-4 Encoder/Decoder, Multimedia, Error Resilience, Error Concealment, Auto Intra Refreshment (AIR), Robustness, Crash-proof, fast DCT/IDCT, fast motion estimation.

1. Introduction

MPEG-4 video coding standard is developed to provide users a new level of performance for video transmission in various applications such as Internet streaming and mobile multimedia applications. For these applications, both device complexity and channel bandwidth are limited. For hand-held devices, the encoding process of video should be as simple as possible to minimize power consumption. For the bandwidth constrained or mobile channels, the decoding process should be reliable and robust enough to deal with random error, burst error or packet losses. This report describes the improvement on both sides of encoding and decoding processes. For the encoding side, selective prediction of motion vector field and integer fast DCT/IDCT are implemented to reduce the complexity of motion estimation. On the other hand, a trustworthy decoder with error resilience ability is necessary for recovering

erroneous or lost data during transmission. In addition, the error resilient decoder has been sped up with fast IDCT and improved bitstream access.

2. Optimized MPEG-4 Simple Profile

Encoder

In [9], we speed up the MPEG-4 Simple Profile encoder based on the underlying optimization approaches, which can be divided into the following three categories.

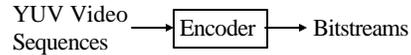
- a. Remove the unused procedures, parameters, and data structures from the code bases for the reduction of code size. For example, some of code bases, which provide identical functionalities, can be merged and some of the allocated buffers are not used for a Simple Profile decoder. Consequently, we can just retain the least needed modules and code bases of the Simple Profile by the MPEG-4 visual specification.
- b. Rewrite the code bases for saving the execution time and code sizes. For example, we remove the unnecessary data movement and conditional jumps at the reference code base. In addition, we avoid from using the arithmetic operations including division (/) and modulation (%) in the code bases. In addition, we rewrite the procedure for motion compensation by reducing the data movement and reuse of the data in the register.
- c. Use existing fast algorithms for the computational burden modules. Accounting for the computational complexity found by profiling, we replace DCT and IDCT with fast algorithms with negligible degradation of picture quality. Also, the fast MVFAST in MPEG document N4554 is used to reduce the computational load of motion estimation.

3. Optimized MPEG-4 Decoder with

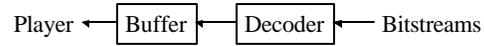
Error Resilience and Concealment

Similar to the speedup approaches used in the encoder, for example, the same fast IDCT is employed with the negligible degradation of picture quality. In addition, we improve the bitstream access method by adding a buffer of size 4 kilobytes for intermediately storing the input bitstream, and by rewriting the bit fetching from the buffer.

2. Porting onto Linux and ARM-9 device



(a) Encoder



(b) Decoder

Figure 1. Block diagram of our demo system.

Table I. The test conditions used for simulations on the X-Pilot MA-1000 system.

Resolution	QCIF (176x144)	CIF (352x288)
	Akiyo, Foreman	Akiyo, Foreman
Frame rate	10 Hz	
Sample bit rates	64kbps	256kbps
Objective measure of results	PSNR vs. bit rate	
Period of I frames	One I frame at beginning	
QP values	Decided by VM5+ rate control	
MV range	+/-32	

To realize the real-time MPEG-4 software encoder and decoder on the ARM-9 device that employs the Linux kernel, the MPEG-4 reference software is used as the basis. This is because the MPEG-4 reference software that can work on the Unix environment also works well within Linux environment.

To demonstrate the performance of our codec under Linux environment, we also we also build a player that can directly play the YUV video sequences on Linux. The basic block diagram is shown as in Figure 1. Our goals are to create a MPEG-4 encoder that can encode the input YUV video sequences into MPEG-4 Simple Profile bitstream. And to create a real-time decoder that can receive MPEG-4 video bitstreams from the files. With the bitstreams, our proposed decoder reconstructs the vide sequence and outputs the result on a device monitor.

For SoC project, since the demo board with ARM-9 processor and wireless communication tools is still in development. Before the integration into the real demo board, we are developing the realtime MPEG-4 encoder and decoder in parallel. The development plat for both the MPEG-4 encoder and decoder are StrongARM 1110 device [10], which supports the same environment such as the Linux-based

Table II. The simulation results for various testing conditions on the X-Pilot MA-1000 system. The overall decoder execution time includes the time to display all reconstructed frame on the monitor.

Sequence name	Akiyo	Akiyo	Foreman	Foreman
Resolution	CIF (352x288)	QCIF (176x144)	CIF (352x288)	QCIF (176x144)
Bistream Size (bytes)	279161	79791	320738	80281
Encoder Execution Time (sec)	551.5	138.4	676.9	166.8
Encoding Rate (fps)	0.18	0.72	0.15	0.6
Decoder Execution Time (sec)	32	9	94	16
Decoding Rate (fps)	3.125	11.11	1.06	6.25
Average PSNR (Y)	42.90	41.29	33.06	31.54
Average PSNR (U)	45.29	43.06	38.51	37.45
Average PSNR (V)	46.47	44.14	39.53	37.49

O.S. and I/O as that is used on ARM-9 demo board. In our decoder, the constructed image can be also shown in the monitor via the player under Linux environment. Our next steps are to improve the encoding and decoding rates of our MPEG-4 software encoder and decoder based on the optimized tools and special instruction sets provided on ARM Linux platform, to simulate the error robustness of the MPEG-4 software encoder and decoder online under the real transmission environment, and to enhance the visual quality of the reconstructed video.

4. Experiment Results

We simulate the improved MPEG-4 Simple Profile video encoder [9] and decoder with error resilience capabilities, which are still in development. Both encoder and decoder are running on ARM Linux platform, where is now built on the X-Pilot MA-1000 Table-PC by AboCom system, Inc [10]. The X-Pilot MA-1000 Table-PC uses Intel StrongARM 1110 CPU (the maximum working frequency is 206 MHz), 32/64/128 MB Memory SDRAM on-board, 16/32MB Flash ROM, and Crystal Clear 10.4" TFT (SVGA) with build-in Touch Screen. The test sequences and bitstreams are stored at IBM Microdrive. To execute the encoder and decoder, the compiler and linker on ARM Linux platform are used.

The testing conditions are summarized at the Table I. To demonstrate the influence of the motion estimation on the overall encoding rates, the large search range of 32 and test sequences with various object moving are used. In these simulations for showing the overall execution performance, the bitstream corruption is not considered.

The Table II shows the simulation results

using the testing conditions in Table I on the X-Pilot MA-1000 system. The results can be analyzed with the following four factors including the sequence, picture resolution, encoding/decoding rate, and the picture quality.

For the various sequences, the fast motion sequence like Foreman is slower than the slow motion sequence like Akiyo in both encoding and decoding rates. This is due to the fast motion sequences need to take more computation in motion compensation module.

For various picture resolutions of the same sequence, the sequences in QCIF format have almost four times of encoding rate than those in CIF format. As to the decoding rate, the speedup is larger than 4 times as the resolution of the sequence is decreased from CIF to QCIF. It's interesting that for the Foreman sequence, the speedup of the QCIF sequence over the CIF sequence is larger than 4 times, which may be caused by less memory movement and bitstream fetching in small resolution sequence.

As to the encoding and decoding rates, all simulations show that both the preliminary encoder and decoder, which do not afford the real-time processing, require advanced improvement on the speed. According to the profiling of function timing for the individual module, the most computational burden modules in the encoder cover motion estimation and bitstream I/O. The motion estimation module MVFAST in the MPEG-4 reference are not as fast as it's expected. For the decoder, the most time-consuming modules include bitstream I/O and YUV display. Especially, the YUV player on the ARM Linux platform occupies lots of the CPU time, which takes 20 milliseconds to display each frame in QCIF resolution. To realize a real-time encoder and decoder, we're searching for another fast algorithms to reduce

complexity of these time burden modules. In addition, we will employ the optimized tools and special instruction sets that are provided for the ARM device and Linux platform for speeding up the current encoder and decoder.

In the optimized decoder, the error resilience capabilities and crash proof implementation are completed. After the error resilient techniques were accomplished into the MPEG-4 reference software, as shown in Figure 2, we can decode all the IVOPs and PVOPs even under BER of value 10^{-5} and complete decoding successfully. Where the testing sequences, Foreman and Coastguard of CIF format, are encoded at 512 or 768 kilobits/sec. The frame rate is 30 frames per second (fps) and each GOV has 60 frames where an I-picture is followed by 59 Ppictures. For each bitstream, each VP has 500 bits. In this report, we take care of all PVOPs, which can improve the decoder based on the approaches similar to those adopted for IVOPs [8]. Especially, we will take care of the error drifting problem when the inter prediction is used for PVOPs. Thus, the AIR technique will cope with the drifting problem by interrupting the error propagation using Intra coding mode for PVOPS automatically. Additionally, a simple scheme of error concealment is adopted for the better visual quality.

5. Conclusion

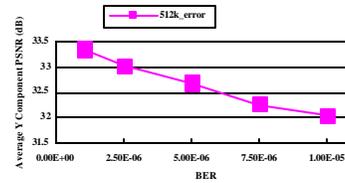
We have proposed the optimized MPEG-4 video encoder and decoder. Both the encoder and decoder that could not reach the real-time stage will be further improved. In addition, the video decoder of crash-proof and error resilience can successfully recover most of the corrupted VOPs under error-prone conditions. The next steps for the error resilient decoder are to strengthen error robustness, recovery and concealment.

It is the most important that combines the adaptive intra refreshment encoder with error resilience decoder, we believe the new results will be more interesting.

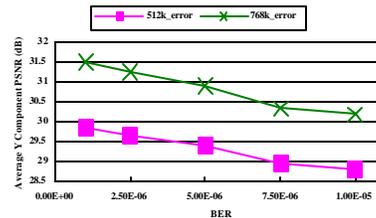
For the demo system on Linux platform, the future work is to optimize the elementary functions in speed and to integrate the real-time encoder and decoder into Linux-based platform the ARM-9 device.

6. References

[1] A. Dagiuklas and M.Ghanbari "Packet video transmission in an ATM network using forced frame refreshment," 1996



(a)Foreman_CIF



(b) Coastguard_CIF

Figure 2. Picture quality of the Y component under various BERs for the recovered video sequences.

[2] L. Favalli, C. Frascini, A. Mecocci, "A low refresh-rate video sequences compression technique using quadrees and adaptive spatial sampling," 1997

[3] E. Steinbach, N. Farber, B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environment," 1997

[4] J. Y. Liao and J. Vilasenor, "Adaptive intra block update for robust transmission of H.263", 2000.

[5] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal processing Magazine*, 2000

[6] R. Talluri, "Error resilient video coding in the ISO MPEG-4 standard," *IEEE Communications magazine*, June, 1998.

[7] D.-S. Luis, P. Fernando, "Error resilience and concealment performance for MPEG-4 frame-based video coding," *Signal Processing: Image Communication 14*, 1999

[8] Y.-L. Lee, C.-N. Wang, and Tihao Chiang, "An MPEG-4 error resilient decoder," in Proc. of WCE 2001.

[9] S-H Wang, C.-N. Wang, Tihao Chiang, and H. Sun, "AHG report on editorial convergence of MPEG-4 reference software" in ISO/IEC JTC1/SC29/WG11, m8884, 2002

[10] AboCom Systems, Inc.,

les@ms3.abocom.com.tw