Image/Video Transcoding with HPL Technology

Bo Shen, Susie Wee
Mobile and Media Systems Laboratory
HP Laboratories Palo Alto

HPL-2007-145
August 27, 2007*

video transcoding, MPEG, H.26x, JPEG

It is evident that the variety of communication channels and terminals is rapidly increasing. With more emphasis on media-rich applications in mobile environments, content adaptation is becoming a key technology enabler. This paper briefly evaluates different alternatives for content adaptation and then focuses the discussion on video transcoding, the more practical solution for content adaptation. The primary goal of media transcoding is twofold, reducing the complexity of the task and maintaining the visual quality of the transcoded result as much as possible. HP offers state-of-the-art best solutions in this space. This white paper briefly introduces the solutions and key technologies that HP offers.

IMAGE/VIDEO TRANSCODING WITH HPL TECHNOLOGY
A Technical White Paper*

Bo Shen and Susie Wee
Mobile and Media Systems Lab.
Hewlett-Packard Laboratories
{boshen,swee@hpl.hp.com}

ABSTRACT

It is evident that the variety of communication channels and terminals is rapidly increasing. With more emphasis on media-rich applications in mobile environments, content adaptation is becoming a key technology enabler. This paper briefly evaluates different alternatives for content adaptation and then focuses the discussion on video transcoding, the more practical solution for content adaptation. The primary goal of media transcoding is twofold, reducing the complexity of the task and maintaining the visual quality of the transcoded result as much as possible. HP offers state-of-the-art best solutions in this space. This white paper briefly introduces the solutions and key technologies that HP offers.

1. INTRODUCTION

In multimedia communication, four factors play important roles in deciding the variety of the terminals: connection bandwidth, processing power, display screen size, and codec support. The communication networks are moving from homogenous wired networks to a variety of wireless networks, each has its own networking characteristics including link capacity. This requires the creation of media content at different rates fitting pipes of different bandwidth capacity. On the other hand, the use of portable devices is increasingly becoming part of everyday life. These portable devices vary in their processing power, display screen size, sometimes drastically. Moreover, there are multiple standards in video coding community for codec support, which results in different handsets may support mutually incompatible video formats. Table 1 gives some perspectives on these varieties. It is evident that technologies of different generations often coexist for a long period of time, the list such as that in Table 1 can only get longer not shorter.

The number of varieties in each dimension may be limited; however, the combination of them can be multiplicative, and hence prohibitive for the source (content creator or host) to anticipate and *precode* content in different varieties. In addition, the precode approach does not solve the dynamic content distribution case such as live broadcasting. On the other hand, since the source previsions the heterogeneity, content adaptation is carried out by selecting or switching to different precoded versions. Therefore there are no computing, quality and security issues during adaptation.

Factors	Varieties	
Link capacity	T1	10 ~ 100 Mbps
	DSL	128 Kbps ~ 1.5 Mbps
	Modem	28 ~ 56 Kbps
	802.11	1 ~ 11 Mbps
	3G	144 Kbps ~ 2 Mbps
	2.5G	48 ~ 64 Kbps
	…	
Screen size	HDTV	1920x1080
	TV	720x480
	PC	800x600 above
	PDA	240x320, 160x160, …
	cell phone	176x144, 128x96, …
	…	
Codec support[1]	PC	MPEG-1, -2, -4
	DVD	MPEG-2
	PDA	MPEG-1, -4
	cell phone	H.263, MPEG-4, H.264
	…	
Processing power[2]	PC PDA	30+ fps 15~30 fps
	cell phone	5~15 fps

Table 1 Variety of communication terminals

Instead of precoding independent versions of content, another option is to code content in a *scalable* fashion (e.g., multi-layer or multiple-description coding); so that when adaptation is needed, simple dropping of parts of the

[1] Proprietary codecs are ignored.
[2] In terms of frame per second (fps) the device can decode and display.

content achieves some quality of service tradeoff. Scalable coding offers a nice solution for content adaptation and when most content becomes coded in scalable format, it will be the best solution. Without focusing on them, this white paper also discusses some of HP's technology offerings in this aspect.

For current predominate video production which compresses content into non-scalable, single layer bit streams, the transcode approach is becoming the preferred solution by providing real-time adaptation. Evaluating the pros and cons of each option, a summary of the *relative* scores in different aspects of these options is given in Table 2. Transcode has clear advantages in the first three aspects, indicating no overhead in creating and storing multiple versions for adaptation. Transcode is also the best solution in matching the dynamics of the varieties. This can be crucial since it can be sometimes impossible to prevision the varieties of different kind. On the other hand, the transcode approach trades off the performance in the last three aspects (shaded columns), that is, it requires computing overhead and the quality of transcoding may not be as good as precoding and scalable-coding. In addition, the transcode approach may incur security concerns. The goal of transcoding technologies development is exactly to improving the performance in these three aspects.

	Source creation complexity	Source storage overhead	Flexibility	Security	Adaptation computing load	Adaptation quality
precode	High	High	L	H	None	High
scalable	Medium	Medium	M	H	Low	High
transcode	Low	Low	H	L	High	Low

Table 2 Content adaptation choices.

A typical use case of transcoding is depicted in Figure 1, whereas media content is delivered to various client devices with various types of connections through a network intermediary with transcoding capability (transcoding proxy/gateway/surrogate). The transcoding proxy may or may not be co-located with the content server, but most likely, it is separated from the clients by at least the last hub. The content is transcoded based on device and connection profiles of the clients (registered or communicated in real time) before delivering.

Since computing load and quality issues are the key concerns in adopting the transcoding approach, the key enabler is then the technologies that reducing this load while producing reasonable quality. We will look further into this problem and discuss HP's methodology towards solving this problem in the next section. An overview of solutions to four basic types of transcoding is presented in Section 3. Section 4 and 5 will then focus more on the key technologies that we have developed for image and video transcoding respectively. Feature set of our prototype implementations of transcoding technologies and their preliminary performances are presented in Section 6. In Section 7, we will discuss other associated HP technologies that are related to how the transcoding technologies can be used in distributed systems. The summary is provided in Section 8.

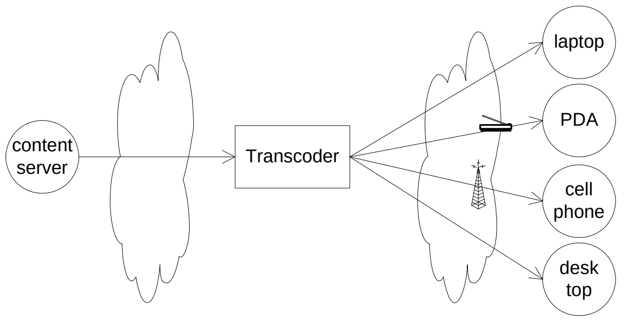


Figure 1. Example: Transcoding proxy system.

2. DESIGN PRINCIPLE

Media transcoding is inherently computing intensive simply because the creation of compressed video is always considered an offline, heavy-duty task. A transcoding task can be viewed as a decoding process followed by an encoding process, with a content-adapting operation (e.g., reducing screen-size by a factor of 2) in the middle as shown in Figure 2(a). This *recoding* process should not be considered as a transcoding simply because it is too heavy-duty and does not explore the correlation between the input and output at all.

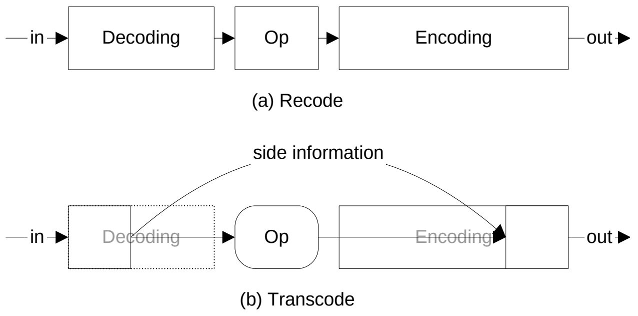


Figure 2. Recode vs. Transcode.

A transcoding approach, on the other hand, attempts to explore the correlation. For example, if some side information such as motion or macroblock coding type information from the input can be reused, it directly implies that less decoding and/or less encoding would be required. A *partial decode* process decodes the input data to a transform domain version which is the frequency representation of the frames most of the compression methods use for further compression. This is the compressed domain in which our algorithms typically operate. The content-adapting operations can be carried out in such kind of compressed domain so as to achieve the maximal amount of computing reduction. Note that we want to avoid, whenever possible, a full decode process which produces the final pixel-domain frames since the decoding from the transform-domain representation to the final pixel-domain frames is often the more computing intensive part. The same applies to the encoding part, that is, it is often the most computing intensive part in an encoding process to obtain the transform-domain data. Therefore, when the transform-domain data is already available (as is in the case of Figure 2(b)), the encoding process can be much faster.

We outline two key principles in developing the transcoding technology:

- Maximize the amount of side information that can be reused. This is directly derived from the goal that we want to reduce the computing load as much as possible. Reusing side information leads to skipped operations in both decoding and encoding.
- Minimize the quality difference between the transcoding results and the recoding results. Note from Table 2 that the transcoder operates on already compressed content; the quality of the output is inevitably lower than precode or scalable-coding methods. Hence, we want the quality to be at least comparable to that of the recode method. This indicates that any quality assurance components in the recoding process should also be in the transcoding process.

It is easily seen that the two principles can lead to conflicting interests. For example, reusing macroblock coding type saves on the re-computing of it, but it may not be the optimal selection that renders the best quality. In some scenarios, these two principles can also work *with* each other. For example, reusing side information not only leads to less computing load, but also renders possibly better quality. This is because the motion information from the input is created with high-quality original frames. In the case of recoding, only lower-quality reproduction frames are available for the encoder for determination of motion.

The principles outlined here are used in our design of transcoding systems for both image and video. Rigorously conforming to, and carefully balancing the conflicts of the two principles separate our transcoding technologies from others'. For example in video transcoding, we do not use approximate transform domain motion compensation to gain on speedup since it sacrifices the quality. In stead, we rely on superior architecture design with advanced component technologies to achieve overall performance improvement. HP offers a complete solution in this space.

3. OVERVIEW OF SOLUTION

We first independently consider the transcoding needs due to the four factors listed in Table 1. For adapting to codec support, screen size, processing power and link capacity, we outline four independent solutions namely transFormat, transScale, transFrameRat and transBitRate, respectively. For each solution, we provide an example problem, a brief description of the solution with a simplified list of algorithm steps and an analysis on its pros and cons.

Converting video's coding format (codec) is called *transFormat*. This is required, for example, when a user wants to playback a video mail in MPEG-4, but her video phone only supports H.263 codec. The solution is to map the syntax and data from the input format to the output format. And it can be carried out in the following simplified steps:

1. Decode the side information from the input;
2. Partially decode the input data to the level of transform domain;
3. Encode by mapping the results from Step 1 and 2 to the format (syntax) specified by the output.

This method is extremely fast by reusing everything from the input and there is no quality degradation if there is a perfect mapping between the input format and output format (as is with the case of MPEG4 Simple Profile and H.263 Profile 0 Level 10). Note that if there is not a perfect mapping between the input format and output format, more processing is needed.

Converting (mostly downscaling) the spatial resolution (screen size, frame size) of video is called *transScale*. This is often required in scenarios such as displaying a DVD movie on the screen of a cell phone. Downscaling the spatial resolution of each frame in the input video achieves it. And it can be carried out in the following simplified steps:

1. Decode the side information from the input;
2. Partially decode the input data to the level of transform domain (or pixel domain if necessary);
3. Downscale the transform-domain (or pixel-domain if necessary) data from Step 2;
4. Encode based on the downscaled transform-domain data (or pixel-domain data if necessary) from Step 3 by reusing as much side information (e.g., motion vector and its associated differential data) as possible from Step 1.

This method can be very fast by reusing as much side information and data as possible from the input. Motion drift is efficiently compensated (that is, the transcoding quality is preserved without too much computing overhead.). Quantization drift correction loop can be switched on and off.

Converting (mostly reducing) the video frame rate is called *transFrameRate*. Consider a source video encoded at a high frame rate for Internet distribution, but a mobile user wants to view it on a CPU-challenged video phone which can not decode and/or display at the high frame rate, the solution is to provide a transcoded video with some of the intermediate frames skipped. Of course, frame rate will not be a problem when the CPU power of users' devices catches up. However, transFrameRate can be a viable vehicle for general rate adapting by trading off motion smoothness. It can be carried out in the following simplified steps:

1. Fully decode every frame (side-information and data) from the input (e.g., at 30 fps);
2. Encode a subset (e.g., 1 in every 5) of the decoded frames from Step 1 to achieve the target frame rate (e.g., 6 fps). Depending on how the input video is coded, some special target frame rate can be achieved by simply dropping compressed frames without decoding them at all.

This method is faster when the output frame rate is lower. Non compressed-domain algorithms are used here since no input information is easily reused. Motion information can potentially be reused by motion-tracking across frames, but the precision may not be good. For low frame rate video, compression achieved based on motion may not be dominant because motion will be less continuous anyway. In other words, it is not worth doing transFrameRate in the compressed domain.

Adapting video's coding bit rate (bandwidth, file size) is called *transBitRate*. This is required, for example, when streaming a high quality news clip at higher bit rate (e.g., 1 Mbps) in a 3G wireless channel with lower bandwidth (e.g., 128 Kbps). The solution is to reduce the bit rate by compressing the input more severely (so that it requires less number of bits per second). And it can be carried out in the following simplified steps:

1. Decode the side information (e.g., motion vectors) from the input;
2. Partially decode the input data to the level of transform domain;
3. Re-quantize (compress more) the transform domain data from Step 2
4. Encode based on the re-quantized data from Step 3 and reuse all side information from Step 1.

This process is very fast by reusing the side information and data from the input. Quantization drift correction loop can be switched on and off. The drawback is of course that there is quality degradation since the content is compressed more severely. This drawback is acceptable in some situations, for example, if the user is willing to tolerate quality degradation rather than no-service. The drawback can also be overcome by coupling it with tranScale and/or transFrameRate (since there are smaller and/or fewer frames to code).

In general, a transcoding task can be any combination of transFormat, transScale, transRate and transFrameRate. In the following sections, we will briefly discuss some specific technologies used in these solutions.

4. HP IMAGE TRANSCODING TECHNOLOGIES

HP has developed technologies for transcoding of images in both non-scalable and scalable formats. These technologies are either covered by HP's IP or IP in the application pipeline.

4.1. JPEG image transcoding

JPEG is the dominant image compression format. It is based on a transform-based technology. That is, pixel signal in images is first transformed to the frequency domain so that it can be better compressed by intelligently discarding some of the high frequency components that is not going to affect the reconstruction quality anyway. This transformation process is often the bottleneck of either the encoding or the decoding process. In order to transcode this type of compressed images, we want to find algorithms that can operate in the transform (frequency) domain. For example, JPEG images can be promptly downscaled since the downscaling can be performed in the transform domain. Several researches in HP Labs pioneered the research in this area and some key solutions were developed including a granted patent [1][2]. These methods can be grouped into three categories:

- Forward mapping which relays on the sparseness of the transform data.
- Optimized matrix multiplication which relays on the sparseness of the manipulating matrices.
- Transform-domain subsampling.

We have investigated the pros and cons of all of the above and can design systems that adapt in different situations. We also have full implementation of these technologies.

4.2 Secure transcoding for JPEG-2000

JPEG-2000 is a scalable image-coding standard that supports very low-complexity transcoding operations for image downscaling, bit-rate reduction, and cropping. JPSEC is a part of the standard that supports security services such as confidentiality and authentication for JPEG-2000 bit streams. When encrypting compressed bit streams, a problem that arises is maintaining the transcodability of the protected stream. A conventional approach to transcoding protected streams involves first decrypting the bit stream, then transcoding the unencrypted content, and finally re-encrypting the content. However, this solution threatens the end-to-end security of the content because the transcoding node needs the key to access the content. HP developed a technology called secure scalable streaming that simultaneously enables the two seemingly conflicting properties of end-to-end security and transcoding without decryption. This is achieved by co-designing the coding and encryption operations so that secure transcoding can be performed [3]. HP is actively incorporating this technology into the JPSEC standard.

5. HP VIDEO TRANSCODING TECHNOLOGIES

As mentioned before, most video content is coded in single layer, non-scalable format; the transcoding of it requires careful design of the architecture to reduce the computing load and quality degradation.

5.1. Optimal reuse of motion information

Motion estimation has been shown to be the most computing intensive component in video encoding. It is at least 60% of the whole encoding process. Given already compressed video as input to the transcoder, it is only natural to reuse the motion information already there. However, it can be very problematic, especially when screen-size adaptation is required. Using screen-size reduction as an example, it is obvious that the mapping of the motion information from input to output is not one to one, but many to one. It is very important to find optimal mapping not only for motion information but also coding components that are associated with motion information. HPL researchers are among the first to research in this area and have extensively investigated the pros and cons of various approaches [4]. Considering numerical operations only, the methods we proposed in reusing the motion information results in the transcoding cost to be only 2.67% that of recoding[3].

5.2. Transform domain downsampling

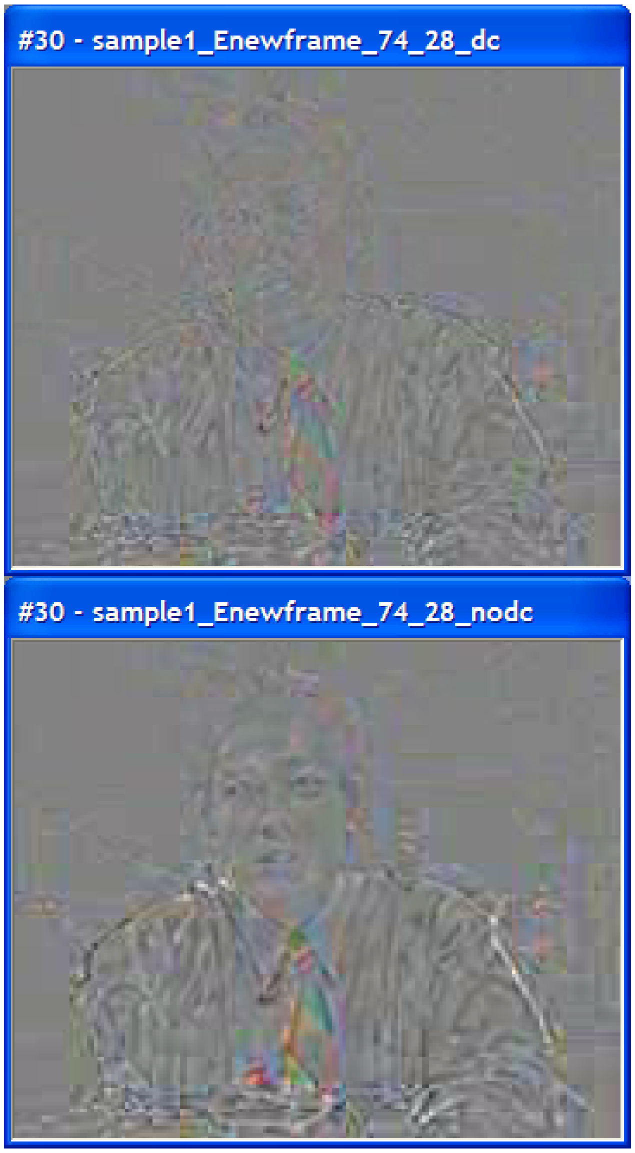
The technology designed for image transcoding (as discussed in Section 3.1) can also be used in video transcoding [5]. For example, transcoding of intra frames (frames that are independently coded) in compressed video are directly applicable. This directly results in a computational saving of 37% for scale factor 2. With careful designs in the transcoding architecture, it can also be applied to the transcoding of other types of frames in compressed video.

5.3. Reduced-resolution motion compensation

As mentioned above, transform domain downsampling can be more helpful if it can be applied to inter frames which are predicatively coded (i.e., based on prediction relative to reference frames). However, due to the double dependency track present in the downscale transcoding architecture, this is very difficult. We have developed an improved architecture that enables the use of transform domain downsampling for inter frames [6]. It is verified that the amount of required processor operations is reduced by 40% for inter pictures for the scale factor 2. In the mean time, transcoded frames maintain high quality.

This solution is especially well suited for transcoder designs on hybrid platforms with general-purpose processors and FPGA (or ASIC). The processor takes care of the complicated logics presented in video transcoding and the computing intensive (yet better hardware-optimizable) parts are offloaded to FPGA or ASIC chips or boards. Our optimized design based on reduced-resolution motion compensation also leads to minimum requirement in buffer memory, which helps driving down the cost of this kind of hybrid transcoding devices.

[3] Numerical operations considered here are add, shift and multiplication. The rest of the paper also follows the same rule in comparing the computing complexities.


#30 - sample1_Enewframe_74_28_dc
#30 - sample1_Enewframe_74_28_nodc
Figure 3. Quality degradation with (top) or without (bottom) drift correction.

5.4. Complete drift correction

Rate adaptation transcoding generates quantization drift and screen-size adaptation transcoding generates motion drift. Any architecture without considering both of these two factors will fail in maintaining transcoding quality. Our architecture design and implementation includes a complete drift correction loop. Moreover, the drift correction loop is algorithmically optimized to reduce its computing complexity. Figure 3 shows an example. The top picture shows the reconstruction error for transcoding with drift correction, bottom picture without drift correction. The transcoder reduces bit rate from 74 Kbps to 28 Kbps for a QCIF video clip. With zero error normalizing to gray, any part darker, whiter or more colorful indicates larger error. It is evident that, with drift accumulated through a sequence of 30 frames, drift correction greatly reduces the transcoding artifacts.

5.5. Field to frame conversion

Another form of transcoding involves converting MPEG-2 digital television content into formats more easily rendered on computers, laptops, cell phones, and PDAs, including MPEG-1, MPEG-4, H.261, and H.263. MPEG-2 supports field-based coding for compressing interlaced video content, however, the non-interlaced compression formats only support frame-based coding. Thus, a need arises for converting field-based coding formats to frame-based coding formats. HP developed field-to-frame transcoding algorithms to meet this need [7]. These algorithms have improved performance using the techniques described in this section including re-use of motion information, transform-domain downsampling, and drift correction.

5.6. Transform domain rate control

We consider rate control as another important component for maintaining or even improving quality of transcoded results. In our transcoder architecture, rate control is performed fully based on transform-domain information. This is very important in transcoding since we want to avoid going to the pixel domain whenever possible. We have designed full transform-domain rate control methods that are equivalent to TM5 (MPEG-1 or -2), TMN8 (H.263) or VM8 (MPEG-4). We have also designed other transform domain rate control algorithms that offer quality improvement.

Note that the effort in rate control algorithm design should be orthogonal to the other technologies. That is, the rate control should be designed in a way to bring quality gains without trading off the performance improvements achieved by the other technologies.

6. FEATURE AND PERFORMANCE

6.1 Image transcoding

The image transcoding technologies are implemented in software; here is a list of the features.

- JPEG
    - Screen-size reduction for factors of 2, 3, 4, and 8.
    - File size reduction (recompress with a higher compression ratio).
- JPEG-2000
    - Screen-size adaptation
    - Compression ratio adaptation
    - Secure adaptation (JPSEC)

6.2 Video transcoding

The video transcoding technologies are implemented in software; here is a list of the features.

- Support transcoding of MPEG-1, MPEG-2 (SP/MP), MPEG-4 (ASP) and H.263 (baseline level 10), same format or cross format.
- Support arbitrary bit rate reduction transcoding.
- Support screen size adaptation (factors of 2, 3 and 4).
- Support simple frame rate reduction.
- Support any combination of the above, i.e., format, bit rate, screen size and frame rate adaptation.
- Support file-based and network-based (RTP) transcoding interfaces.
- Code base is compatible with HPUX, Linux and Windows© platforms.

Even without additional platform-dependent code optimization (e.g., MMX), we can already achieve more than 10x speedup over real time decoding (playback speed). On HP Proliant DL 140 with dual 2.4 GHz Xeon processor, our transcoder can carry 15 concurrent sessions for screen-size reduction (CIF-to-QCIF) + rate reduction transcoding. For QCIF-to-QCIF rate-reduction-only transcoding, 28 concurrent sessions can be supported. Figure 4 shows the CPU time for one QCIF-to-QCIF session. Bottom (blue) curve represents the per frame CPU time for our transcoder. The curve is much lower than the top (red) curve which represents the results from a simplified recoding, i.e., a recoding with the encoding part not doing motion estimation (if using regular recoding with motion estimation, the red curve would be magnitude higher).

For all the cases, transcoding quality is maintained within 0.5 dB[4] PSNR (peak signal to noise ratio) difference comparing with recoding. Our transcoder can sometimes achieve better PSNR than recoding. This is because transcoder reuses motion information from the input which is based on raw frames *before* compression. The recoding approach, on the other hand, obtains the motion information from the decompressed frames *after* compression. Table 3 shows the PSNR differences between recoding and transcoding for 6 test sequences.

[4] In video coding, 1 dB difference starts to be visible.

Three technologies are compared. FresMC indicates the transcoder with technology described in Section 4.1, iRresMC with that in Section 4.1 and 4.2 and RresMC with that in Section 4.1 to 4.3. Technologies described in Section 4.4 and 4.5 are applied to all of the above. It is observed that the quality degradation decreases when more component technologies are used. For some test cases, we even observe improved quality (negative difference). Note that these quality numbers are achieved with significantly less computing load compared with recoding.

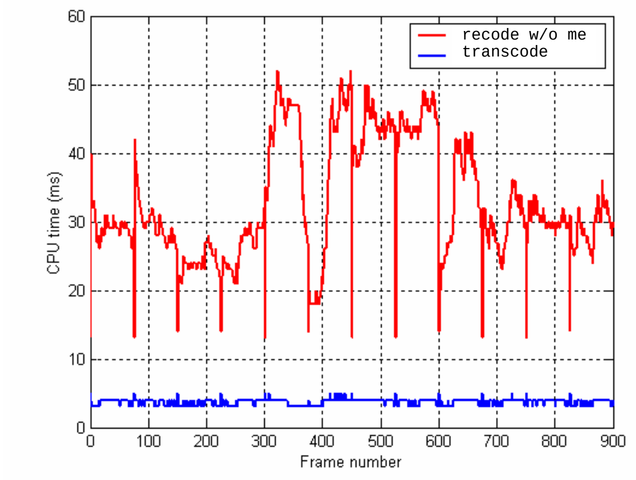


Figure 4. CPU time/frame for one rate-reduction session (QCIF, 15fps, 74kbps to 28kbps).

Sequence name	FresMC		iRresMC		RresMC	
	Mean	Std.	Mean	Std.	Mean	Std.
akiyo	0.13	0.08	0.03	0.14	-0.09	0.09
table-tennis	0.18	0.26	0.16	0.46	0.08	0.37
foreman	0.47	0.38	0.50	0.39	0.35	0.41
hallway	0.11	0.09	-0.17	0.09	-0.19	0.09
news	0.11	0.09	-0.11	0.10	-0.18	0.09
coastguard	0.19	0.18	0.32	0.30	0.20	0.26

Table 3 PSNR difference with Recode (dB). Input video is at CIF resolution with 12-frame GOP size and IPBBPBB structure. Output video is at QCIF resolution and one quarter the bit rate of input video.

7. OTHER ASSOCIATED HP TECHNOLOGIES

In addition to the core portfolio of transcoding technologies described above, we have also developed other associated technologies.

In bit rate reduction video transcoding, blockness in the transcoded result can become a major problem, especially for transcoding to low bit rate that is suitable for mobile device with wireless connection. Previous works have shown that deblocking can be achieved by offsetting some of the DCT coefficients. However, in video transcoding, these DCT coefficients are subject to a re-quantization process, which may render the deblocking ineffective. We propose to consider deblocking within the transcoding loop, i.e., in conjunction with the requantization process therefore accelerating the deblock-enabled transcoding processes. In addition, the method also motivates a design of an optimized quantizer selection algorithm that boosts the deblocking capability of the video transcoder [8].

An emerging video coding standard H.264/AVC offers superior compression performance. It is predicted to become the codec of choice in the foreseeable future. H.264 uses an integer transform to produce transform coefficients. This is different than the DCT (discrete cosine transform) that is used in most current video coding standards such as MPEG-x. As mentioned before, being able to reuse transform coefficients is the key to transcoding performance; we are the first to investigate and offer preliminary solutions to this problem [9].

Commoditization of transcoding capability also offers network intermediaries in content distribution networks the ability to better addressing the heterogeneous issues. One typical example is that an intermediary such as a cache proxy now has more options in trading off computation vs. storage so that overall performance can be increased significantly. We have investigated this problem and proposed transcoding-enabled caching methods in [10].

Further more, in the same transcoding-enabled caching scenario, since the transcoder sees a collective behavior of the client accesses, aggregated optimization is possible by sharing partial transcoding results among different sessions. We have proposed novel concepts of *meta-caching* and *meta-transcoding* which indicate that intermediate results (metadata) of transcoding sessions can be cached and subsequent identical transcoding requests are served through transcoding from the cached metadata. The proposed methods provide a foundation to achieving superior storage and computation resource balance at the proxies. Extensible to any kind of media processing services, we specifically investigated and identified the appropriate metadata and its characteristics for three types of video transcoding tasks. Simulations based on these characteristics validate that the proposed methods achieve superior computing load reduction on transcoding service proxies [11].

Coupling contend adaptation with the mobile environment, one important issue is to support seamless hand-off among transcoding sessions. We have explored different hand-off solutions, starting with the transfer of sufficient amount of state so as to produce byte identical streams when compared to the case when there is no hand-off. However it is found that such a scheme introduces a considerable amount of hand-off delay at the client, due to the large amount of data that needs to be transferred. We have proposed, analyzed and experimentally evaluated more efficient hand-off schemes that reduce delay, while introducing no noticeable degradation in the output quality. These hand-off schemes can also be used to provide load-balancing at the transcoder, or for fault tolerance in either cluster or distributed systems [12].

8. SUMMARY

The proliferation of heterogeneous communication networks and devices demands content adaptation. Solutions relaying on provisioning the variety and creating versions beforehand are not effective and sometimes can be impossible. Also, scalable coding solution has not become prime time, as is evident that most of the media content one can find today is not scalable-coded. To satisfy the current and imminent need of content adaptation, transcoding is the key solution that matches the dynamic nature of the ever-increasing variety of communication networks and devices.

HP offers a full set of solutions in this space, ranging from core technologies that reduce transcoding complexity to its full potential while maintaining the highest quality, to a full spectrum of related technologies that enables more effective and efficient use of transcoding in the distributed environment.

These technology offerings are coupled with prototyping implementations that have already demonstrated superior performances as well as abundant supported features. A collection of related documents, status reports on past and current HPL activities in this area is maintained in an internal website [13].

9. REFERENCES

[1] N. Merhav and V. Bhaskaran, “A fast algorithm of DCT-domain image downscaling,” *Proc. ICASSP’96*, vol. II, pp. 2307-2310, Atlanta GA, May 1996.

[2] N. Merhav and V. Bhaskaran, “Fast DCT domain downsampling and inverse motion compensation,” US Patent, US5708732. Issued 1998. (HP patent)

[3] Susie Wee and John Apostolopoulos, “Secure Scalable Streaming and Secure Transcoding with JPEG-2000,” IEEE International Conference on Image Processing (ICIP) 14-17 September 2003, Barcelona, Spain. (paten pending)

[4] Bo Shen, I.K.Sethi and V.Bhaskaran, "Adaptive motion vector resampling for compressed video down-scaling", IEEE Trans. Circuits and Systems for Video Technology, vol.9, no.6, pp.929-936, Sept. 1999.

[5] B. Shen, and S. Roy, “A Very Fast Video Special Resolution Reduction Transcoder,” *Proceedings of International Conf. On Acoustics Speech and Signal Processing (ICASSP),* Orlando, FL, May 2002. (patent pending)

[6] B. Shen, “Motion Drift Modeling and Correction for Fast Video Downscale Transcoding”, IEEE Trans. Circuits and Systems for Video Technology 2005. (patent pending).

[7] S.J. Wee, J.G. Apostolopoulos, and N. Feamster, “Field-to-Frame Transcoding with Temporal and Spatial Downsampling,” IEEE International Conference on Image Processing, Kobe, Japan, October 1999.

[8] Bo Shen, "Efficient deblocking and optimal quantizer selection for video transcoding," Proc. IEEE International Conf. on Image Processing (ICIP03), Sept. 2003, Barcelona, Spain. (patent pending)

[9] Bo Shen, "From 8-tap DCT to 4-tap integer-transform for MPEG to H.264 transcoding," Proc. IEEE International Conf. on Image Processing (ICIP04), Oct. 2004. (patent pending)

[10] Bo Shen and Sung-Ju Lee, Sujoy Basu, "Caching Strategies in Transcoding-enabled Proxy Systems for Streaming Media Distribution Networks," IEEE Trans. on Multimedia, vol.6, no. 2, pp.375-386, Apr. 2004. (patent pending)

[11] Bo Shen, "Meta-caching and Meta-transcoding for server side service proxy", Proc. of IEEE International Conf. on Multimedia and Expo (ICME03), vol. I, pp. 457-460, Baltimore, MD, July 2003. (patent pending)

[12] S. Roy, Bo Shen, V. Sundaram and Raj Kumar, "Application Level Hand-off Support for Mobile Media Transcoding Sessions", Proc. of The 12th International Conf. On Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'02), May 2002. (patent pending)

[13] HPL Streamteam, Media Transcoding Technologies (MT^2), http://galley.hpl.hp.com/~boshen/trans_home.html.

[*] First version was written in Sept. 2004 for a technology transfer to HP OpenCall Business Unit.