A Framework for Video Streaming to Resource-Constrained Terminals

Dmitri Jarnikov[1], Johan Lukkien[1], Peter van der Stok[2]

[1] Dept. of Mathematics and Computer Science, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
{d.s.jarnikov, j.j.lukkien}@tue.nl
[2] Philips Research
Prof Holstlaan 4, 5656 AA Eindhoven, the Netherlands
peter.van.der.stok@philips.com

Abstract. A large range of devices (from PDA to high-end TV) can receive and decode digital video. However, the capacity of one device is very different from the capacity of another device. This paper discusses how the same video but differently encoded can be efficiently distributed to a set of receiving devices. The key technology is scalable video coding. The paper shows how a framework assists in adapting the digital code to the changing transmission conditions to optimize the quality rendered at the different devices. The paper concludes with a validation based on a real-time streaming application.

1 Introduction

Delivering a high quality video over a network and seamless processing of the video on various devices depend for a larger part on how the network and device resources are handled. The paper addresses a concurrent video content distribution to multiple resource-constrained devices, focusing on wireless in-home networks. Fig. 1 shows a schematic view of a simple video transmission system that consists of a set of terminals wirelessly connected to a sender. A sender is a powerful device that stores videos and provides media access to video data for all receivers in the net.

Terminals, being mostly CE devices are resource-constrained. Moreover, the devices have different processor/memory capabilities, thus not every terminal may be capable of processing all video data that is streamed by the sender. To ensure that video data is processed by all terminals, a sender should send to each terminal the amount of data that can be successfully processed by the terminal. This is usually done by performing a content adaptation at the sender.

There are several strategies for content adaptation. The three foremost of those strategies are the simulcast model, transcoding and scalable video coding model. With the simulcast approach the sender produces several independently encoded copies of the same video content at varying features, such as different bit/frame rates, and spatial resolutions. The sender delivers these copies to the terminals, in agreement with the specifications coming from the terminal.

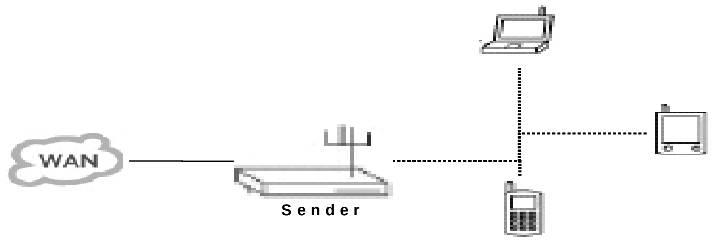

Fig. 1. General view of the system

Transcoding of a video stream is converting the video stream from one format to another (e.g. from MPEG-2 [3] to MPEG-4 [4]) or transformation of the video stream within the same media format (e.g. change of frame rate, bit-rate or image resolution).

A scalable video coding scheme describes an encoding of video into multiple layers, including a Base Layer (BL) of basic quality and several Enhancement Layers (EL) containing increasingly more video data to enhance the quality of the base layer [9]. Scalable video coding is represented by variety of methods that could be applied to many existing video coding standards [3,4,5]. These methods are based on principles of temporal, signal-to-noise ratio (SNR) and spatial scalability [7]. Combinations of the scalability techniques produce hybrid scalabilities (e.g. spatial-temporal).

In a heterogeneous environment there are terminals of different types, so multiple possibilities for content adaptation are possible. We propose a framework that enables video streaming to terminals with different resource constraints (the rest of the paper addresses only processing resources) based on a uniform content adaptation for all terminals.

2 Analysis

We distinguish three types of terminals. The first type has a fixed decoding process, based on a standard legacy decoder. The second type has a decoding process that is organized as a scalable video algorithm (SVA) [8], which allows changing of the quality of the processing to trade-off resource usage for output quality of the algorithm. The third type of terminals is capable of processing data that is created with scalable video coding techniques.

Type 1. Most terminals are equipped with a standard software or hardware decoder. In this case simulcast is the simplest content adaptation method. Every terminal in the system requests a stream that fits closely to the current resource limitations of the terminal. The overall bandwidth consumption of the simulcast approach is a sum of the bit-rates of all streams sent to the terminals. So, the simulcast strategy fills up the network bandwidth with a number of variations of the same content, which results in over utilization of the bandwidth. Also, if the bandwidth has frequent fluctuations, which is often the case in wireless networks, the simulcast approach is not the best technique because when available network bandwidth is lower than overall bit-rate of all streams, some of the streams need to be cut off.

Transcoding is another content adaptation technique for this type of terminals. The size of the video stream *BT* that can be processed by all terminals is chosen such that it satisfies the requirements of the weakest terminal

$$BT = \min(B_i) \text{ for } i = 1..N \, , \tag{1}$$

where B_i is the highest bit-rate that can be processed by receiver i.

Network bandwidth fluctuations could be handled by lowering/raising video stream bit-rate. Evidently, the highest possible bit-rate, *BA*, for the stream is defined as a minimum between maximum bit-rate allowed by network, *BN*, and suitable for all terminals

$$BA = \min(BN, BT) \, , \tag{2}$$

where BN is the currently available network bandwidth. A disadvantage of the approach is that receiver i in the system has unused resources

$$U_i = B_i - \min(B_j) \text{ for all } j = 1..N \, . \tag{3}$$

Type 2. If receivers are capable of adjusting the quality of decoding process, i.e. change usage of resources for decoding, the utilization of terminal and network resources can be improved. Each terminal can be characterized by the highest stream bit-rate (B) that can be processed. If resources are not sufficient for the complete processing of the input data with bit-rate B, the processing is performed with a lower quality. So, the output quality of the decoder lowers to a given level Q, which is the maximal quality that can be achieved under current resource limitations. We define BM as a bit-rate that can be fully processed by the terminal with the highest quality of the processing and produces the same quality Q with the current resources allowance.

In the system, a video stream is transcoded to the bit-rate that satisfies the requirements of the strongest terminal, but still can be processed by all others

$$BT = \min\left[\max(BM_i), \min(B_i)\right] \text{ for } i = 1..N \, . \tag{4}$$

The highest possible bit-rate is calculated as in (2).

The transcoding approach has difficulties in handling fast bandwidth changes. Whenever a change occurs, the transcoder adapts the bit-rate of the video stream. However, some time is necessary to detect the change and communicate it to the transcoder. Additional time is needed for the transcoder to adapt to new settings.

Type 3. If terminals are able to process scalable video coded streams, a scalable video coding adaptation technique is an effective solution to the drawbacks mentioned above. The advantage of scalable video is the easy adaptation to varying channel conditions. The content adaptation can be done on a per frame basis. This is very important for wireless video transmission, where a bandwidth change is fast and unpredictable. Also, usage of scalable video coding gives a terminal the possibility to subscribe to as many layers as it can handle.

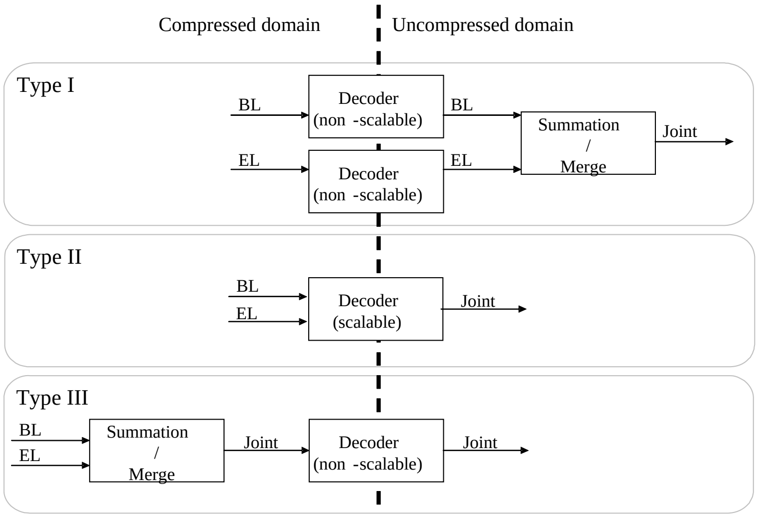

Fig. 2. Three types of organizing decoding process for scalable video coding

Possible types of organization of the decoding process for scalable video are depicted in Fig. 2. Type I assumes a video where BL and EL are encoded as a standard non-scalable video streams. The streams are decoded by different decoders and then merged together. It is possible to decide upfront, how many layers will be processed. Type II takes scalable video that complies with a standard scalable technique. The output of the decoder is a joint decoded video. Type III operates with video streams that can be merged together before being decoded.

The schemes shown in Fig. 2 provide a higher quality at the cost of increased resource consumption. The whole decoding process can be seen as a SVA. Choosing the number of processed layers is the key to trade-off quality for resource consumption. A decoding process may drop processing an enhancement layer at any moment thus providing complete input of a frame at the best quality possible given the processing limitations. Processor consumption can be changed on a per-frame basis in real-time.

An important requirement for a scalable video coding technique is that successful transmission and decoding of the BL should be guaranteed. This is not possible if the available bandwidth drops below BL bit-rate or there are not enough resources to decode BL. To handle this situation, a reconfiguration of layers (i.e. changing of number of layers and their bit-rates) at run-time is proposed.

The characteristics of the terminal, which processes scalable video, are the maximal number of layers (NL_i) and the maximal bit-rate of the BL (B_i). The bottleneck of the approach is the complexity of choosing a layers configuration that suits both network and terminals resources. Although it is easy to calculate the highest possible bit-rate for BL as in (2), the number and bit-rates of EL are usually chosen manually.

Conclusion. In general, transcoding into multiple layers is suitable for use with all types of terminals, where a one layered transcoding is a particular case of scalable video coding with *NL* equal to 1. We suggest a scalable video coding as a uniform content adaptation technique. In this paper we propose a general framework for video streaming to multiple terminals that is based on scalable video coding.

3 Framework description

The design of our framework is depicted on Fig. 3. A receiver (terminal wirelessly connected to a sender) processes incoming video data in accordance with local resource limitations. The sender makes an adaptation of input video data in conformity with requirements of the terminals and the network conditions. The intent of the system is that scalable-enabled receivers and non-scalable receivers use the same basic functionality and a single adaptation technique is performed based on requirements of all terminals.

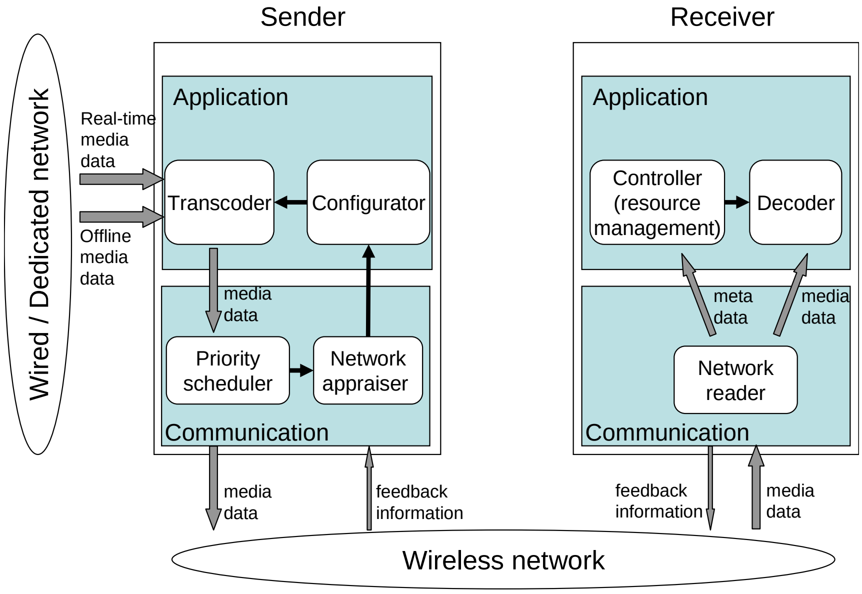

Fig. 3. Design of the system

3.1 Receiver

The receiver is responsible for receiving the required number of layers from the offered ones, decoding, and displaying video data. Also, it reports to the sender the maximal number of layers (NL), maximal bit-rate that can be processed without lowering processing quality (BM) and maximal bit-rate B . Additionally, the receiver provides a feedback to the sender in relation to the network statistics (how many layers/frames where successfully received). The three components shown in the receiver in Fig. 3 are discussed in detail.

Network reader. The main task of the network reader is to pre-process video data. In the case of scalable video coding, transmission delays may lead to loss of synchronization between the layers. Frames from both base and enhancement layers should be merged during decoding process. If a frame from enhancement layer comes too late, this frame is discarded before offering it to the decoder. The network reader informs the local controller about the amount of received video data. Network reader communicates network statistics back to the sender.

Decoder. Unless the terminal is equipped with a standard decoder, the decoding process is implemented as a scalable video algorithm. The amount of processed data is a parameter that changes the output quality as well as the resource consumption.

A decoding process for scalable video coding drops an enhancement layer to fit into the given processing resources. The control over resource usage can be changed accordingly.

Controller. A decoder implemented as a SVA needs a controller to assure that resource consumption of the decoder is scaled to meet the current limitations. However, changes in the amount of processed data result in a picture quality change. Frequent changes of picture quality are not appreciated by user. A control strategy, as proposed in [6,1], minimizes the number of quality changes while maximizing the average output quality.

3.2 Sender

The sender takes online or offline content and transcodes it into a scalable video. This video is sent over a wireless network to multiple receivers. A feedback from the receivers is used for making a decision about configurations of the scalable video, i.e. choosing number and bit-rates of layers.

Scalable video transcoder. The transcoder converts non-scalable video into multi-layered scalable video in accordance to the current layer configuration. The configuration may be changed at run-time. The input to the transcoder is provided via a reliable channel, thus assuming that there are no losses or delays in the incoming stream.

An important requirement for the transcoder is the ability to incorporate information about currently chosen configuration into data streams. Writing configuration settings into the base layer allows natural propagation of changes through the system, as all involved parties may be informed on the complete change history.

The encoded streams must have two properties to allow dropping of frames of a layer without consequences for other frames: 1) no dependencies of base layer frames on enhancement layer frames [2], 2) frames in an enhancement layer should have no relation to each other.

Priority scheduler. If transcoder outputs ELs along with BL, the sender uses a priority scheduler, which ensures that layers of a scalable video are sent in accordance to their priority. Since BL information is absolutely necessary to enable scalable video usage, this layer has the highest priority. The priority of EL decreases with increasing layer number.

When a channel degrades, the buffers of the sender get full. This affects the low priority streams on the first place and propagates towards the higher priority streams. The information regarding the fullness of the sender buffers is communicated to the network appraiser as an indication of the network status.

Network appraiser. The information from the scheduler is a rough but fast indication of a network condition change. In turn, a feedback channel delivers more precise information expressed in terms of error rate and burstiness. The network appraiser collects information from the priority scheduler and from receivers and communicates changes to the layer configurator.

Layer configurator. The layer configurator chooses number and bit-rates of layers based on the acquired information about network conditions and terminals status. The network information is used to estimate BN (currently available network bandwidth), worst-case error rate and burtiness of the errors. The terminal information is used to define the maximal number of layers that could be produced

$$L = \max(NL_i) \text{ for } i = 1..N \tag{5}$$

and the required BL bit-rate that is calculated based on (4). The layer configurator uses a pre-calculated strategy to choose a suitable configuration based on the above-mentioned values. A strategy is created offline by a network simulation environment.

Strategies are created with knowledge of maximal number of layers and the currently available network bandwidth. The maximal bit-rate for BL is not taken into account, as it again increases the number of needed strategies. If a lower BL bit-rate should be chosen at run-time due to terminals requirements, the bit-rate of BL is redistributed to the first EL.

5 Evaluation

A prototype has been constructed to validate the approach described in this paper. Our implementation of the scalable video transcoder is based on a modified MPEG-2 SNR video coding and corresponds to the decoding type I (Fig. 2), which suggests standard non-scalable decoders with an external summation of streams after the decoders. The implementation of our coding overcomes an important weakness of a standardized approach [3]: the dependency of BL on EL. Moreover, our scheme allows any number of ELs, which provides greater flexibility for accommodation to various network conditions. The details of the implementation are presented in [2].

The resource management is done based on the network aware controller [6]. The approach is based on a Markov Decision Process (MDP). The MDP is solved offline and the calculated optimal control strategy is applied online to control the decoding algorithm [6].

For the evaluation we used a system with three different types of terminals. Terminal 1 is equipped with standard MPEG-2 decoder and is capable of processing at run-time standard definition streams of not more than 3.5 Mbps. Terminal 2 is capable of adjusting the quality of decoding process. It handles standard definition streams of up to 6 Mbps, however only 4 Mbps could be efficiently decoded at run-time. Terminal 3 handles scalable video consisting of at most 3 layers and BL bit-rate should not exceed 8 Mbps.

Fig. 4 shows an example of layer configurations for different system setups. If Terminal 1 is the only receiver in the system, the sender streams only one video stream with a bit-rate not higher than 3.5 Mbps. If network conditions do not allow successful transmission of video, the bit-rate is decreased to fit into present network limitations. A similar behavior is observed for Terminal 2 as a receiver, the only difference is that bit-rate increases up to 4 Mbps (maximal efficiently decoded bit-rate), whenever network conditions allow it. For the system with Terminal 3, the configurations with BL and one EL are chosen as most optimal. New configuration of BL and EL is chosen when the network conditions change.

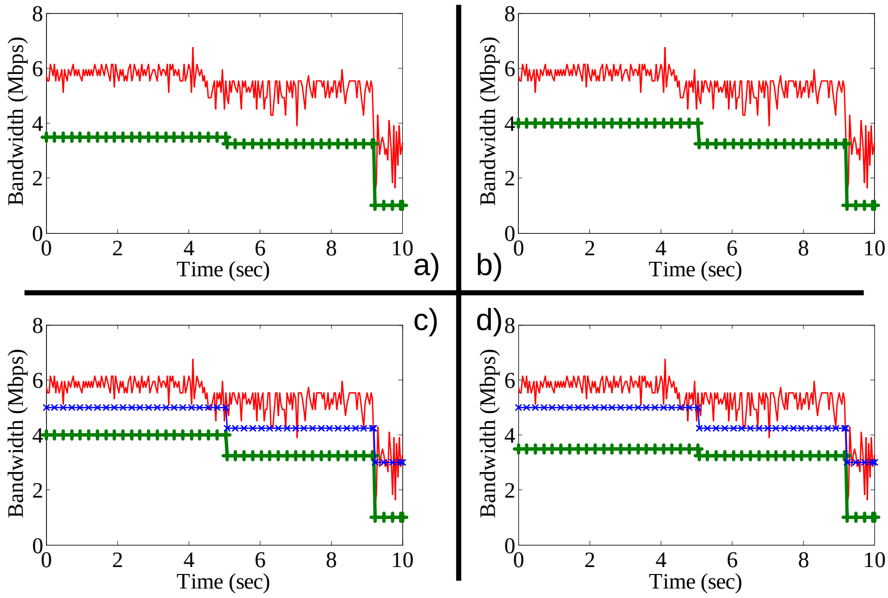

Fig. 4. Configuration of layers for different terminals: a) Terminal 1; b) Terminal 2; c) Terminal 3; d) Terminals 1, 2 and 3. Top line is available bandwidth, crossed (+) bottom line is BL bit-rate, crossed (x) line is bit-rate of BL and EL.

Bringing all three terminals together changes the configuration of layers (Fig. 4, d). The layer configuration delivers a BL and EL, with Terminals 1 and 2 subscribing only to BL and Terminal 3 receiving both layers. However, the BL bit-rate is chosen based on the requirements of the weakest terminal, which is Terminal 1. During time interval [0, 4.5] the BL bit-rate is lower than the optimal for Terminal 3 (Fig. 4, c). Because the optimal BL bit-rate of 4 Mbps is higher than the maximal bit-rate that Terminal 1 can handle (3.5 Mbps). The difference of 0.5 Mbps is reassigned to EL, resulting in the same overall bit-rate for layers.

As the second step we looked at the setup with all three terminals. We simulate starting of another application on a terminal by lowering its processing capabilities by half. Fig. 5 shows an example of changed layer configurations because of resource availability change on one of the terminals. The change occurs 5 seconds after start.

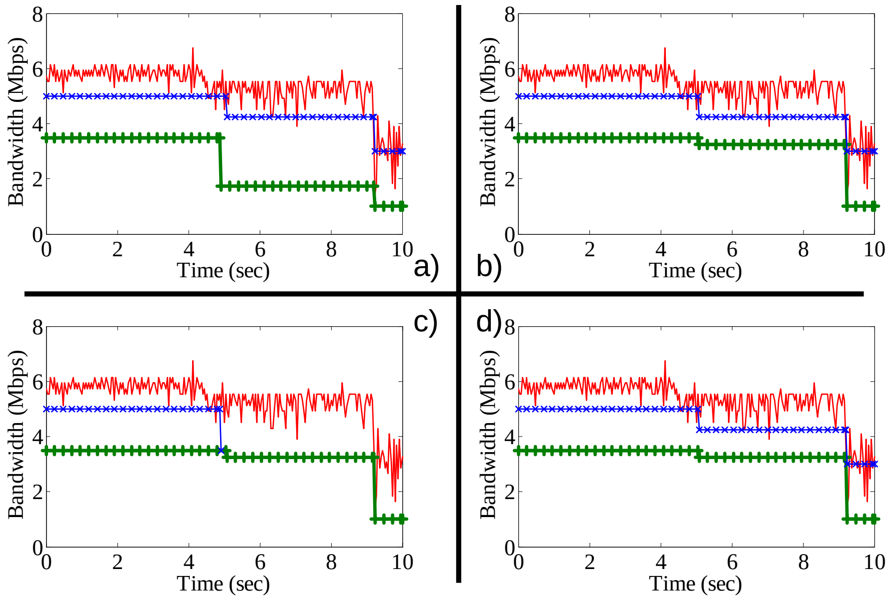

Fig. 5. Configuration of layers under changing resources availability: a) change on Terminal 1; b) change on Terminal 2; c) change on Terminal 3; d) no changes. Top line is available bandwidth, crossed (+) bottom line is BL bit-rate, crossed (x) line is bit-rate of BL and EL.

If Terminal 1 has a drop in available resources, it results in lower BL (Fig. 5, a). For scalable video the drop in BL bit-rate is compensated by increasing the bit-rate of EL such that the total bit-rate of the layers is the same as for the optimal configuration (Fig. 5, d). If a resource drop occurs in Terminal 2, it has no influence on layer configuration. The BL size stays the same and Terminal 2 lowers quality of processing to fit into current resource allowance. Finally, if Terminal 3 experiences a shortage of resources, it drops processing of EL.

6 Conclusions

In this paper we presented a framework for achieving high quality video transmission over wireless networks based on scalable video coding. Important quantities are the number of layers and the size of the layers. A change of the network conditions forces a change to the number or size of one or all layers. The paper shows at which moments such changes need to be initiated and what their value should be.

The paper also shows that the same scheme still applies when the receiving decoder can only handle one layer. Consequently, the proposed scalable video based content adaptation technique can be applied generally for all types of terminals.

References

1. C.C.Wust, L.Steffens,R.J.Bril, and W.F.J.Verhaegh, “QoS Control Strategies for High Quality”, Video Processing. In Proc. 16th Euromicro Conference on Real-Time Systems (ECRTS), Catania, Italy, 2004.
2. Dmitri Jarnikov, Peter van der Stok, Johan Lukkien, “Wireless streaming based on a scalability scheme using legacy MPEG-2 decoders”, Proceedings of The 9th IASTED International Conference on Internet and Multimedia Systems and Applications, ACTA Press, 2005.
3. ISO/IEC International Standard 13818-2, “Generic Coding of Moving Pictures and Associated Audio Information: Video”, Nov., 1994.
4. ISO/IEC International Standard 14496-2, “Information Technology – Generic Coding of Audio-Visual Objects, Part 2: Visual”, MPEG98/N2502a, Oct., 1998.
5. ITU-T International Telecommunication Union, “Draft ITU-T Recommendation H.263 (Video Coding for Low Bit Rate Communication)”, KPN Research, The Netherlands, Jan., 1995.
6. Jarnikov, D.; van der Stok, P.; Wust, C.C., “Predictive Control of Video Quality under Fluctuating Bandwidth Conditions”. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on, Volume: 2 , pp. 1051 – 1054, June 27-30, 2004
7. McCanne, S., Vetterli, M., Jacobson, V., “Low-complexity video coding for receiver-driven layered multicast”, IEEE journal on selected areas in communications, vol. 16, no 6, p.983-1001, 1997.
8. R.J. Bril, C. Hentschel, E.F.M. Steffens, M. Gabrani, G.C. van Loo and J.H.A. Gelissen, “Multimedia QoS in consumer terminals”, Proc. IEEE Workshop on Signal Processing Systems (SIPS), pp. 332-343, Sep. 2001.
9. Yao Wang, Joern Ostermann, and Ya-Qin Zhang, “Video Processing and Communications”, Prentice Hall, 2002