Scalable Video Coding Using Wavelets

Sai Karthik Vuppalapati (09d07050) Satya Naren (09d07051)
Swrangsar Basumatary (09d07040)
Group 21

April 11, 2013

Abstract

This is a report for the Scalable Video Coding Using Wavelets application assignment in the course EE 678 Wavelets. Our goal in this assignment is to demonstrate the spatial aspect of video scalability by encoding different spatial video resolutions into a single bitstream in a scalable manner. We also compare the performance achieved by the scalable approach to that of single-layer approach and simulcast.

1 Introduction

A scalable extension to the H.264/AVC video coding standard has been developed within the Joint Video Team (JVT), a joint organization of the ITU-T Video Coding Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG)[5]. Wavelets have the beautiful property of being inherently scalable[2, 3]. Currently there is a lot of research in the area of wavelet-based approach to scalable video coding[1]. Though the inherently scalable property of wavelets looks promising, no one has been able to come up with a wavelet-based approach to SVC that is superior in performance to the standard SVC. Here we demonstrate spatial scalability of videos using a wavelet-based approach.

In this experiment, we have decided to use *three* different spatial resolutions for demonstration. At the trasmitting end, we compress the three different resolutions and combine them into a single bitstream in a scalable way and transmit it. At the receiver, we uncompress the bitstream to get the resolution required.

The YUV format is known to be more efficient than the RGB. We use YUV format for our video frames here. Every frame has a Y, U and V component. The Y stands for the *luminance* (perceptual brightness) component and U and V are the *chrominance* (color) components. The human eye is more sensitive to the luminance component Y than the chrominance components U and V. It cannot perceive the difference between an image formed by less densely sampled U and V components and the original one as long as UV sampling rate is above some threshold. So we can sample the U and V components at a relatively lower rate without degrading the perceptual quality of the video. Here, we choose the size of U and V components to be half the size of the Y component.

We have three different components Y, U and V in each frame. But from hereon, we shall treat them as one. Because we shall be performing the same operations on all the three components.

2 Encoding the video

At first, we reshape every frame of the video into a $2^N * 2^N$ square frame where N is chosen such that 2^N lies somewhere close to the greater of the two dimensions of the original frame.

Then we do the following to each frame:

1. We do a 2-level 2D wavelet decomposition to get the sub-bands $A2$, $H2$, $V2$, $D2$, $H1$, $V1$ and $D1$.

2. The approximation sub-band, $A2$, is our base layer. $H2$, $V2$ and $D2$ form our first enhancement layer. The second enhancement layer is made up of $H1$, $V1$ and $D1$.

3. We compress the base layer using the Set Partitioning in Hierarchical Trees (SPIHT) algorithm[4].

4. We compress the two enhancement layers by quantizing and then run length encoding them.

Quantizing the sub-bands

We quantize the enhancement layers to achieve a peak Signal-to-Noise ratio (PSNR) that is above a user-specified minimum value. The number of quantization levels is chosen such that the PSNR requirement is satisfied. Starting from an initial number of two quantization levels we keep doubling the number of levels as long as PSNR is below a user-specified value. We double the number of levels instead of multiplying by some other arbitrary factor because we want the quantized values to be in the binary format.

3 Transmission

We arrange the compressed layers of each frame in the following order: the SPIHT compressed base layer, the compressed enhancement layer and then the compressed second enhancement layer. Then we combine all the frames of the video into a single bitstream and transmit them.

For demonstration purposes, we write the compressed layers into binary files and read them back at the receiver.

4 At the receiving end

At the receiving end, we choose our output video resolution according to the capacity of the channel. The logic used in determining the output resolution is as follows:

- If the channel can accommodate the bit rate required for the largest resolution, then we go for it. We uncompress the base layer using SPIHT[4] and run length decode the two compressed enhancement layers. And we use 2D wavelet reconstruction to get the frames of the maximum resolution video,

- or else if the channel can accommodate the bit rate required for the intermediate resolution, then we SPIHT uncompress the base layer and run length decode the compressed first enhancement layer. The 2D wavelet reconstruction gives the intermediate resolution video,

- or else if the channel can accommodate the bit rate required for the base resolution, then we uncompress the SPIHT compressed base layer to get the base resolution video,

- otherwise we can choose to quit or use buffering or some other techniques depending on our application.

In the experiment, we reconstruct all the three different video resolutions.

5 Comparing scalable approach to single-layer approach and simulcast

In single layer approach only one layer,the highest resolution layer, is SPIHT compressed and transmitted. Therefore there is less information to transmit compared to the scalable case. Thus the bit rate required is relatively low.

In simulcast all the three different resolutions are SPIHT compressed and transmitted together in a single bitstream without any scalability among the different resolutions. In simulcast there are no enhancement layers or base layers, all the three different resolutions are independent. Therefore there is more data to transmit compared to the scalable case and thus more bit rate is required than in the scalable case.

Therefore single layer approach is supposed to be faster than the scalable approach whereas the simulcast is supposed to be slowest of them all[5].

To show that the speed of the scalable approach lies in between that of single layer approach and simulcast, we demonstrate the single layer approach and the simulcast also.

Single layer approach

We compress all the frames using SPIHT algorithm and transmit them. At the receiving end, we uncompress the frames using the same algorithm to get the output video. Only one resolution, the highest resolution is used in this approach. There are no lower or higher resolution layers.

Simulcast

Here we have three layers of different resolutions for each frame. The layers are independent. They are not connected to each other in a scalable way. The layers have all the information in themselves and you need not get information from the immediate lower resolution layer. There is no concept of enhancing the immediate lower layer to get the current layer. We compress all the three different layers of each frame using SPIHT algorithm and transmit them.

At the receiving end, we uncompress the compressed layers of each frame using SPIHT to get three video sequences of different resolutions.

Observation

It turns out that the scalable approach required a bit rate higher than that of single-layer approach and lower than that of simulcast. In other words, the performance observed for the scalable approach is higher than that of simulcast but lower than that of the single-layer solution.

6 Summary

Though we have demonstrated the spatial scalability of video for only three different resolutions. This concept can be expanded to any number of resolutions. It is easier to implement dyadic scaling because 2D wavelet decomposition directly gives you a dyadic scaled down version of the original frame.

Furthermore, the performance achieved can be improved by using better compression and quantization techniques.

References

[1] N. Adami, A. Signoroni, and R. Leonardi. State-of-the-art and trends in scalable video compression with wavelet-based approaches. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9), September 2007.

[2] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5), September 1990.

[3] Ingrid Daubechies. Where do wavelets come from? a personal point of view. *Proceedings of the IEEE*, 84(4), April 1996.

[4] Amir Said and William A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3), 1996.

[5] C. Andrew Segall and Gary J. Sullivan. Spatial scalability within the h.264/avc scalable video coding extension. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9), September 2007.