

Scalable Video Coding Using Wavelets

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

April 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.

1 Introduction

Spatial Scalability within the H.264/AVC Scalable Video Coding Extension has already been demonstrated[2] by well-known organizations like Motion Pictures Experts Group(MPEG). But here we are trying to demonstrate spatial scalability using wavelets. Wavelets have the beautiful property of being inherently scalable. Many researchers are trying to harness the scalable property of wavelets to improve the efficiency of scalable video coding techniques.

In this experiment, we decided to use *three* different spatial resolutions for demonstration. We encoded the three different resolutions at the transmitter side into a single bitstream and transmitted it. At the receiver, we decoded the bitstream to get the resolution required.

Also, since the YUV format is known to be more efficient than RGB, we used YUV format for our video frames. Every frame has a Y, U and V component. The size of U and V components were chosen to be half the size of the Y component. This is because the U and V components carry less information compared

to the Y component. So, they can be sampled at a relatively lower rate.

2 Encoding the video

At first, we resized every frame of the video into a $2^N * 2^N$ square frame where,

$$N = \text{floor}(\log_2(\max(\text{size}(\text{frame}))))$$

Then we did the following to each frame:

1. We did a 2-level 2-D wavelet decomposition to get the subbands $A2$, $H2$, $V2$, $D2$, $H1$, $V1$ and $D1$.
2. $A2$, the approximation subband, is our base layer. $H2$, $V2$ and $D2$ forms our first enhancement layer. The second enhancement layer is formed by $H1$, $V1$ and $D1$.
3. Then we compressed each layer using SPIHT[1] algorithm and quantized them.

Quantizing the subbands

We quantize the compressed 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 so that the PSNR requirement is satisfied. We start from two quantization levels and keep doubling the number of levels as long as PSNR is below user-specified value. This loop is finite because PSNR increases as we increase the number of quantization levels. Of course, the user-specified value

should be below the maximum possible PSNR value, or else we have to run into an infinite loop.

3 Transmission

We encoded all the quantized layers of each frame using Run Length Encoding.

We then arranged every encoded frame and so that for every frame the data was arranged in the following order: the base layer, the first enhancement layer and then the second enhancement layer. Then we combined all the frames of the video into a single bitstream.

For demonstration purposes, we wrote the encoded layers into binary files instead and read them back at the receiver.

4 At the receiving end

At the receiving end, if we want the smallest of the three pre-decided resolutions then we retain only the base layer data of each frame. And if we want the intermediate resolution then we retain the first enhancement layers also. For the maximum resolution, we retain all the layers.

We take the retained data and then decode them to get the output.

Decoding the video

First, we run length decode the retained data. Then we uncompress them using SPIHT[1] algorithm. The uncompressed layers are combined using 2-D wavelet reconstruction and then resized to a user-specified dimension.

These resized frames together form the output video.

5 Comparing scalable video to single-layer bitstream and simulcast video

In single layer approach only one layer most probably the highest resolution layer is transmitted without any scaling. And in simulcast all the three different resolutions are transmitted together without any scaling among them. In single layer approach and simulcast, we don't any sort of enhancement layers, all the layers are self-contained.

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

To show that the bitrate of the scalable approach lies in between that of single layer approach and simulcasting, we demonstrated the single layer approach and the simulcast approach also.

Single layer approach

We took the highest resolution layer of each frame and quantized them. Then we did a run length encoding of the frames and transmitted them.

At the receiving end, the incoming bitstream was run length decoded to get back the output video frames.

Simulcasting

Here we compressed all the three different resolution frames using SPIHT algorithm, and then quantized, run length encoded and transmitted them.

At the receiving end, we run length decoded and uncompressed the frames using SPIHT. The output was the video with the required resolution.

Observation

The scalable approach turned out to have a bitrate in between that of single-layer approach and simulcast.

References

- [1] 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.
- [2] 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.