Video Pre- and Post-Processing Algorithms for Break through Cost-Effective Video Compression

Angel DeCegama, Ph.D.
Wentworth Institute of Technology

Introduction

The volumes and costs of video storage and transmission are soaring. This situation can only be ameliorated by massive investments in infrastructure or by technological breakthroughs or both.

This paper presents one such technological breakthrough that can reduce the size of any video file compressed by any existing video codec, e.g., MPEG-4, H.264, DivX, VC-1, etc., to between 25% and 10% of such compressed size without loss of the video quality resulting from the decompression and display by the codec of its compressed file and without any changes to the codec.

The processing presented in this paper that achieves such results involves the preparation of the frames of the original video file before passing them on to a given codec. The codec then processes the received frames in its usual way to produce a much smaller compressed video file than without the initial pre-processing. The compressed video can then be stored and/or transmitted. For decompression and playback, the codec decompresses the compressed video frames in its usual way and then passes them onto the algorithm presented in this paper for final post-processing before display with a quality indistinguishable from that produced by the codec alone without using such pre- and post-processing.

Mathematical Basis

The mathematical principles behind the algorithms presented here are those of the Wavelet Transform (WT). This is important because it has been demonstrated that human beings use the basic concepts of the WT to process in their brains all sensory information specially visual information that requires enormous amounts of compression. Such compresssion involves the discarding of all data that is irrelevant from the standpoint of human perception.

There have been many attempts to apply the WT to the problem of video compression (4,5). The results have been encouraging showing better results than other existing codecs but not good enough to motivate the people working in this area to change their already set up procedures because of he computational complexity of WT methods needed to achieve high quality video with WT-based codecs.

The approach presented in this paper is different. Do not change the codec. Pre-process and post-process the video.

Detailed Description

In this approach, a crucial feature is the ability to recreate a given image or video frame from the low-frequency component of its WT which is ¼ the size of hte original image or video frame. This can be done precisely by applying the math of the direct WT and the IWT (inverse WT).

In order to minimize computational complexity, the Haar WT can be used. The direct Haar WT low-frequency coefficients are $a_2 = 0.5$ and a_1 - 0.5 and the high-frequency coefficients are $b_2 = 0.5$ and $b_1 = -0.5$. The IWT low-frequency coefficients are $aa_2 = 1.0$ and $aa_1 = 1.0$ and the IWT high-frequency coefficients are $bb_2 = -1.0$ and $bb_1 = 1.0$. The WT is applied to the individual pixel rows and columns of a given image or video frame. This is done separately for the luminance (Y) and chrominance (U,V) components of the different pixels of each row and column. It can also be done for the R, G and B planes.

Let's define a set of y_is to constitute the different values of one such component of a given row or column of an image or video frame. Let's also define a set of x_is to be the corresponding WT low-frequency values and a set of z_is to be the corresponding WT high-frequency values.

We can then write for the Haar WT (with decimation and now wraparound)

$$\begin{array}{ll}
X_0 = a_2y_0 + a_1y_1 & Z_0 = b_2y_0 + b_1y_1 \\
X_1 = a_2y_2 + a_1y_3 & Z_1 = b_2y_2 + b_1y3 \\
. & . \\
. & . \\
. & . \\
X_n = a_2y_{2n} + a_1y_{2n+1} & Z_n = b_2y_{2n} + b_1y_{2n+1}
\end{array}$$

The procedure for these calculations is shown in Fig. 1

Knowing both the x_is and zis we can reconstruct <u>exactly</u> the y_is by calculating the corresponding IWT.

$$\begin{array}{l}
y_0 = aa_2x_0 + bb_2z_0 = x_0 - z_0 \\
y_1 = aa_1x_1 + bb_1z_1 = x_1 + z1 \\
y_2 = aa_2x_1 + bb_2z_1 = x_1 - z_1 \\
y_3 = aa_1x_2 + bb_1z_2 = x_2 + z_2
\end{array}$$

...

$$y_{2n-1} = x_n + Z_n$$

$$y_{zn} = aa_2x_n + bb_2z_n = x_n - z_n$$

The above equations represent the IWT process shown in Figure 2.

Assuming that y_{2n+1} is known, we can write

$$y_{2n} = x_n - b_2y_{2n} - b_1y_{2n+1} = x_n - 0.5y_{2n} + 0.5y_{2n+1}$$

And

$$y2n = \frac{x_n}{1.5} + \frac{0.5*y_{2n+1}}{1.5} = \frac{2x_n + y_{2n+1}}{3}$$

Similarly, we can keep moving back towards y0.

$$\begin{aligned}
y_{2n-1} &= x_n + z_n = x_n + b_2y_{2n} + b_1y_{2n+1} \\
&= x_n + 0.5y_{2n} - 0.5y_{2n+1} \\
&= \frac{4x_n - y_{2n+1}}{3}
\end{aligned}$$

.

.

.

$$y_2 = \frac{2x_1 + y_3}{3}$$

$$y_1 = \frac{4x_1 - y_3}{3}$$

$$y_0 = \frac{2x_0 + y_1}{3}$$

Similar equations can be obtained by moving from top to bottom and from left to right.

In other words, given the xis that are the low-frequency values of the decimated Haar WT of the y_is, and given the very last value of the y_is of a row or column of the original image or video frame, the y_is values for the entire row or column can be found. Therefore, besides the xis values, one more value, the very last value of the y_is, can also be stored to the be able to recreate precisely the entire original row or column. This is a negligible overhead when one considers that we are dealing with hundreds or even thousands of pixels for each row and column of images or video frames of typical applications.

By applying such a procedure to every row and column of an image or video frame, the size is reduced to approximately ¼ of the original that can be reproduced exactly from its reduced version. This process can be repeated on the reduced images or video frames for further size reductions of 1/16, 1/64, etc., of the original.

Of course, this cannot be done indefinitely because the precision of the calculations must be limited in order to avoid increasing the required number of bits instead of reducing it and information is being diluted at each ¼ reduction. However, extensive tests show that the quality of image reproduction is maintained up to 2 or 3 reduction levels with size reduction of up to 16 or 64 times before compression by the codec which is very significant in terms of video storage and transmission costs.

The reproduction by any standard codec of the reduced size frame is precise enough to be able to apply the above calculations for recovery of the original full size frames with similar quality to that of the frames recovered by the codec without the initial size reduction step.

In addition, the process can be further extended by using, after more than 2 or 3 reduction levels, any of the expansion filters disclosed in US Patent No. 7,317,840 (reference 2) to enlarge very small images with high quality.

Such procedures can be extended to other wavelets beyond the Haar Wavelet although the calculations are more complicated and time consuming. In this case, the corresponding equations for the WT and IWT lead to a sparse system of linear equations in which only a small number of its matrix elements are non-zero, resulting in a band diagonal matrix in which the width of the band depends on the number of Wavelet coefficients. There are software packages applicable to such systems, e.g., Yale Sparse Matrix Package, but the Haar method above provides the quality and speed that make such more complicated approaches unnecessary for situations in which real-time processing is an important requirement.

Description of Frame Reduction and Expansion Techniques

As above indicated, the application of the decimated Haar WT to a given video frame results in a frame that is ¼ the original size because only the low-frequency Haar wavelet filter is applied. It has been proven above that the high-frequency Haar wavelet filter need not be applied if just the last original value before wavelet transformation of a row or column pixel is saved. With this information, all the preceding original pixels of a row or column can be calculated exactly.

This process can be repeated again on the resulting reduced frame for additional sized reduction to 1/16 of the original and so on. This process is described in detail below.

One-Level Frame Size Reduction and Expansion Back to the Original Size

Figure 3 shows the process of reducing the size of a frame to about ¼ of its original size.

"A" is the original frame with dimensions x and y. The decimated low-pass Haar WT is applied to A horizontally resulting in frame "B' of dimension (x/2) +1 and y. The last column of A, i.e., "LC", is copied to the last column of B.

Next, the decimated low-pass Haar WT is applied to the (x/2)+1 columns of B resulting in frame "C" of dimensions (x/2)+1 and (y/2)+1. Notice that pixel X (R, G, B, or Y, U, V component) is kept through this process. LR/2 is the decimated WT of LR and LC/2 is the decimated WT of LC.

The process of recovering the original frame from C of Figure 3 is shown in Figure 4. First the last row of C is used to precisely recover the columns of B using the reconstruction algorithmic calculations disclosed above in this patent application. Finally, the reconstruction algorithm is applied to B horizontally starting with the values of LC reconstructed from the value of X using the reconstruction algorithm from right to left to recover A exactly.

The procedure can be interfaced to any exiting codec, eg., MPEG-4, H.264, VC-1, DivX, etc, to improve its compression performance significantly (60% to 80% reduction in storage and transmission costs for all extensively tested video files) with no loss of video quality compared to that produced by the original codec after decompression. First, the size reduction process is applied to the original frames of a given video file. Then the codec is applied to such smaller frames to produce a much smaller file than without the size reduction step. The resulting compressed video file can then be stored and/or transmitted at a greatly reduced cost.

For decompression and display, the codec is applied for decompression and then the above frame expansion procedure is used prior to displaying high-quality video in its original full-size

Because of the lossy compression of existing standard video codecs, there is some minor loss of video quality compared to the original before compression by the codec but the algorithm described here does not result in any perceived degradation of quality when compared to that produced by the codec on its own from a much larger file.

Multiple-Level Frame Size Reduction and Expansion

The process described in Figures 3 and 4 can be continued one or more levels starting with the C frame instead of the A frame. There are additional right columns and bottom rows to be saved but they are one half the sizes of the previous level and, consequently, they don't appreciably detract from the saving in storage and transmission bandwidth.

Figure 5 shows the process of reducing the frame size one more level.

Figure 6 shows the expansion by one-level of a two-level sized reduction. The original full size can then be recovered by the process of Figure 4

Figure 7 shows the process of going from 2-Level size reduction to 3-Level size reduction and Figure 8 shows the profess of expansion from 3-Level size reduction to 2-Level size reduction. Additional levels of reduction can be handled similarly.

Modes of Operation

The above ideas and algorithms can be implemented in a number of different ways.

For example,

1. Frame size reduction of only one level but codec compression at different levels of bit assignment per compressed frame.

2. Frame size reduction of multiple levels and codec bit assignment as a function of the reduced frame size of the different levels.

Performance

Such new algorithms (named ADC2 for Advanced Digital Compression Squared) for video data pre- and post-processing to be used with any given video codec without modifying the codec in any way are the basis for the extraordinary typical results shown in the following table.

Typical Results for H-264 (Without loss of video quality)

HD 1080P Videos	Original Size	Codec-Compressed at 6 Mbps	Codec+ADC2 at 1.0 Mbps	Codec+ADC2 at 0.5 Mbps
Living Landscapes	36.6 GB	226 MB	40 MB	21 MB
Casino Royale	73 GB	451 MB	80 MB	42 MB
Planet Earth	76 GB	460 MB	83 MB	44 MB
Yellowstone	38 GB	238 MB	43 MB	23 MB
Over California	82 GB	508 MB	93 MB	49 MB
Steelers	79 GB	490 MB	88 MB	46 MB

Note that the x_i values are approximate because the actual original values have been processed by the codec being used to compress/decompress. Therefore, the y_is calculated with the preceding algorithm are also approximations. The question is how good the approximations are. This can be determined by the perceived visual quality of the output video and by calculating the PSNR values for every output video frame for different bit rates and comparing such values for the codec alone and for the values obtained using the methods of this paper.

In terms of the PSNR values, the following table shows the comparison in PSNR values (in decibels (db)) between the codec alone and the codec with the ADC2 algorithms for different video examples.

PSNR Values (in db)

HD Videos (1920x1080)	Casino Royale	Baseball	King Kong	Terminator	Yellowstone
H264 – 3 Mbps	127.05	125.32	126.91	126.82	110.66
ADC2 – 3 Mbps	128.14	128.677394	128.132767	128.118862	114.999
H264 – 2 Mbps	102.69	107.22	105.49	106.18	108.76
ADC2 – 2 Mbps	128.1179233	128.6735	128.127	128.1181	114.987
H264 – 1 Mbps	81.29	86.13	81.23	82.15	76.82
ADC2 – 1 Mbps	128.117233	128.6731	128.121	128.11879	114.966
H264 – 0.5 Mbps	69.19	72.67	70.83	72.21	65.31
ADC2 – 0.5 Mbps	128.117224	128.6728	128.119	128.11876	114.953
H264 – 0.25 Mbps	54.33	57.92	56.11	56.62	48.12
ADC2 – 0.25 Mbps	128.117223	128.6724	128.117432	128.118744	114.952515

This table clearly shows the high quality of the video reconstructed by the ADC2 algorithms even at very high levels of compression, whereas the quality of the video decompressed by any standard codec (H264 in this example) quickly decays with lower bit-rates which makes such standard codecs unsuitable for practical applications requiring good visual quality at very low bit-rate. These mathematical results confirm the visual perception evaluations of the results of the invention.

It can bee seen from the preceding tables that the size reductions of video files compressed using the techniques of this invention are about 1/5 to 1/10 (depending on the level of size reduction) of the compressed files using the codecs alone. The perceived qualities of the decompressed videos for the different reduction levels are indistinguishable from that of the decompressed videos produced by the codec alone for all the codecs.

The implications of these results for the reduction of the current staggering costs of video storage an transmission are extremely significant.

Summary

Embodiments of this technology use mathematical methods to develop software that uses wavelet transforms (WT) to process video frames that can then be compressed using a variety of codecs. Such compressed video frames can then be transmitted, decompressed, and displayed in their original size and quality, thereby producing real-time high-quality reproduction of video sequences.

Such performance can be verified by accessing the website at www.ADC2Video.com. By following the User Instructions, the ADC2 Video Player is downloaded and installed automatically first and then any of the examples in the menu can be downloaded, decompressed and played with unbeatable quality obtained from compressed video files of very small size that can be read out of the ADC2 Video Files Directory where the compressed files are automatically downloaded. The videos can be seen during streaming download and repeatedly after downloading by double clicking on the ADC2 Video icon to launch the player created by the ADC2 Video Player installation process and selecting the video from the ADC2VideoFiles directory.

This technology is applicable in any system that will benefit from any significant improvement in image and video compression in terms of cost and storage and transmission bandwidth without disrupting or interfering with coexisting codecs, such as MPEG-4, H.264 or VC-1 and without loss of quality of the decompressed video. The technology is capable of improving the video compression of any codec by a factor of between 5 and 10, as demonstrated in extensive tests, without having to make any changes to the currently used codecs and without any appreciable loss of visual quality as indicated above. Therefore, all video coummunication systems, e.g., video conferencing, distant learning, entertainment, medical care, and real-time news can greatly benefit from using embodiments of the technology.

References

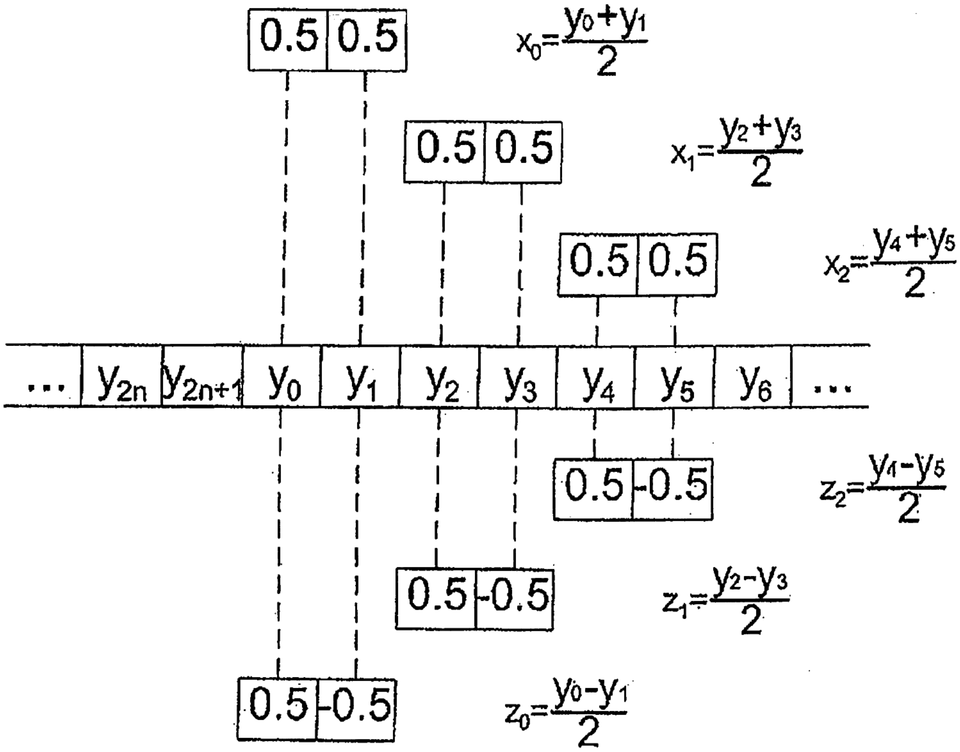
1.) Ten Lectures on Wavelets by Ingrid Daubechies, Society for Industrial and Applied Mathematics.

2.) Methods for Real-Time Software Video/Audio Compression, Transmission, Decompression, and Display, US Patent #7,317,840, January 8, 2008, Angel DeCegama

3.) Systems and Methods for Compression, Transmission and Decompression of Video Codecs, US Patent #8,031,782, October 4, 2011, Angel DeCegama

4.) A new Wavelet Transform Video Compression Algorithm, IEEE 3rd International Conference on Communication Software and Networks, 2011, Zhang Shu

5.) Three Dimensional Wavelet Transform Video Compression, 1999 IEEE International Conference on Multimedia Computing and Systems, July 1999, I.K. Levy



$$x_0=\frac{y_0+y_1}{2} \qquad z_0=\frac{y_0-y_1}{2}$$

$$x_1=\frac{y_2+y_3}{2} \qquad z_1=\frac{y_2-y_3}{2}$$

$$x_2=\frac{y_4+y_5}{2} \qquad z_2=\frac{y_4-y_5}{2}$$

$$\vdots$$

$$x_{n-1}=\frac{y_{2n-2}+y_{2n-1}}{2} \qquad z_{n-1}=\frac{y_{2n-2}-y_{2n-1}}{2}$$

$$x_n=\frac{y_{2n}+y_{2n+1}}{2} \qquad z_n=\frac{y_{2n}-y_{2n+1}}{2}$$

FIG. 1

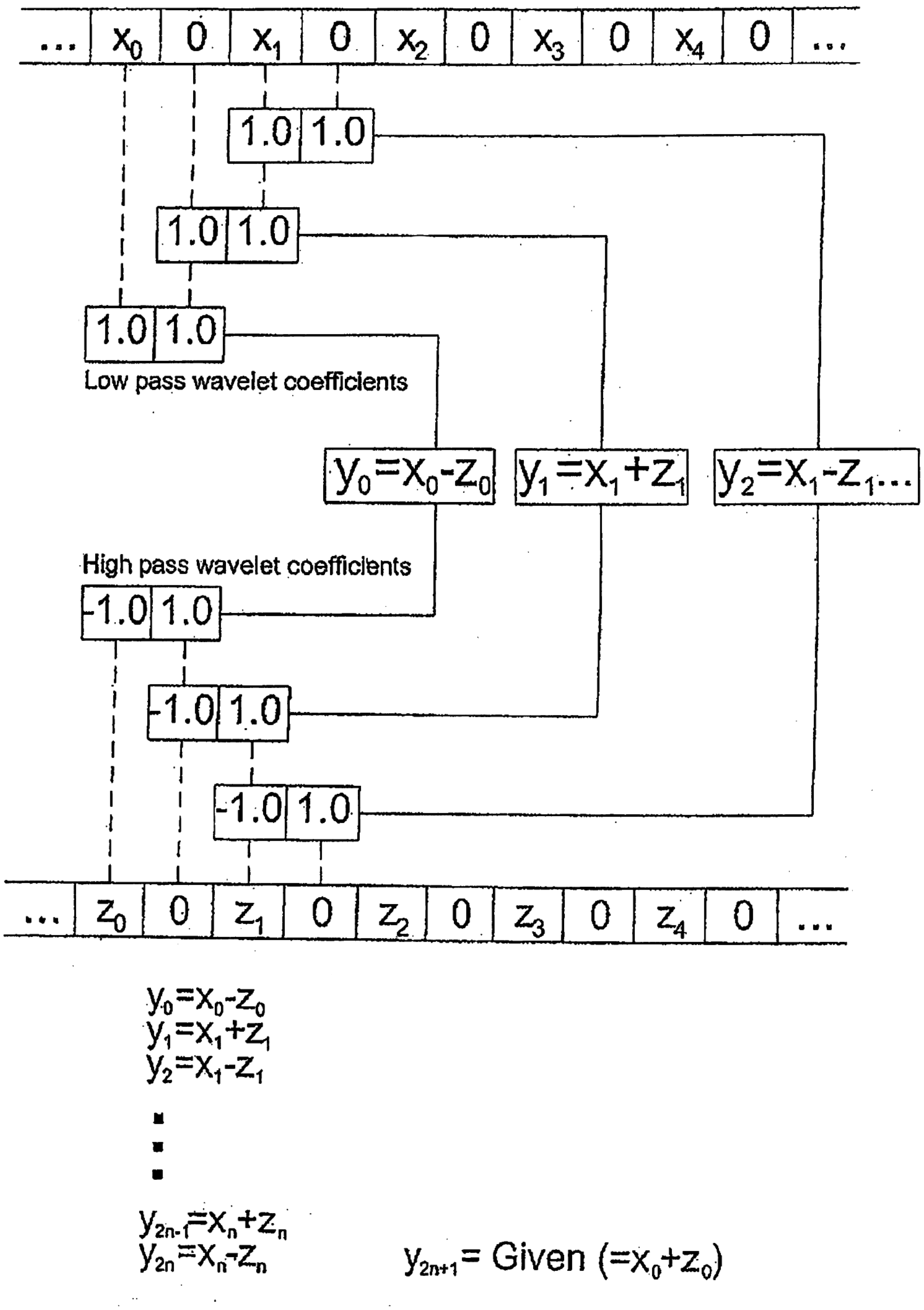

... x_0 0 x_1 0 x_2 0 x_3 0 x_4 0 ...
1.0 1.0
1.0 1.0
1.0 1.0
Low pass wavelet coefficients
$y_0=x_0-z_0$
$y_1=x_1+z_1$
$y_2=x_1-z_1$...
High pass wavelet coefficients
-1.0 1.0
-1.0 1.0
-1.0 1.0
... z_0 0 z_1 0 z_2 0 z_3 0 z_4 0 ...
$y_0=x_0-z_0$
$y_1=x_1+z_1$
$y_2=x_1-z_1$
$y_{2n-1}=x_n+z_n$
$y_{2n}=x_n-z_n$
y_{2n+1}= Given ($=x_0+z_0$)

FIG. 2

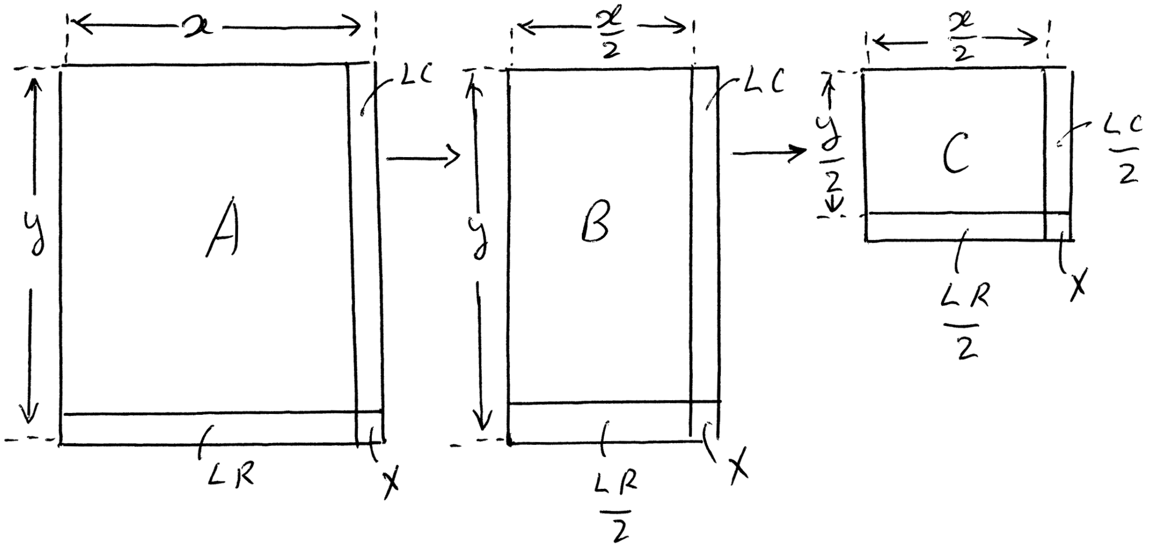


FIG. 3: One-Level Size Reduction

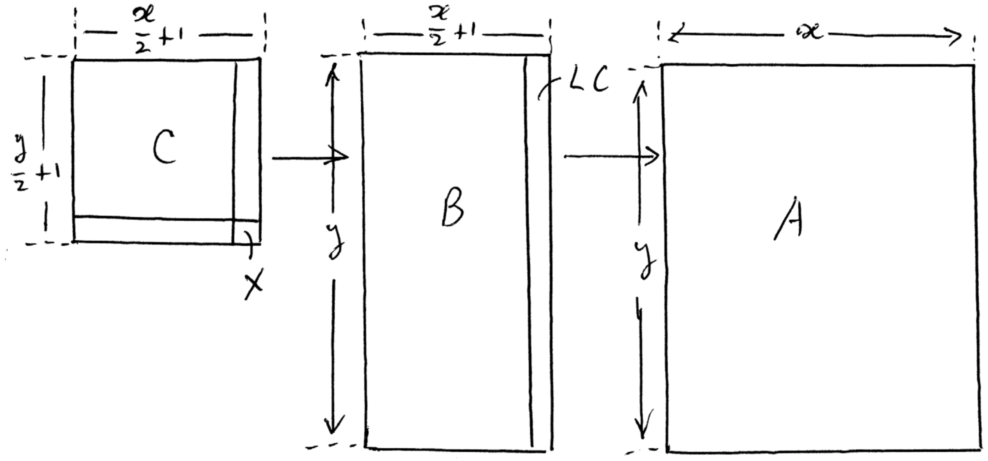


FIG. 4: One-Level Size Expansion

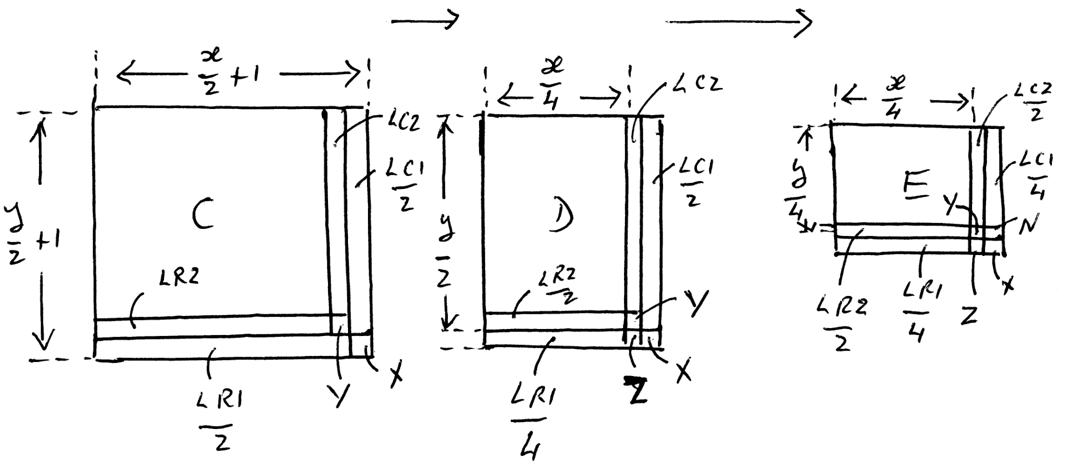


FIG. 5: Two-Level Size Reductino

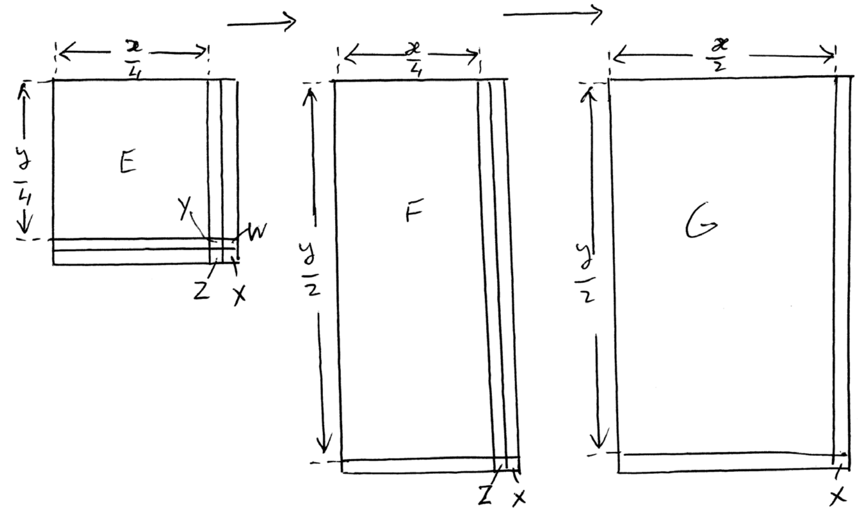


FIG. 6: One-Level Expansion of Two-Level Reduction

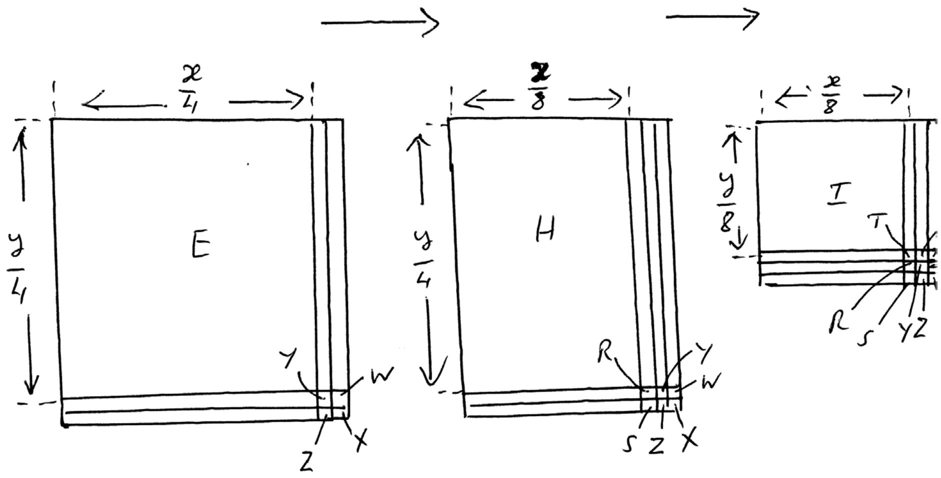


FIG. 7: Three Level Size Reduction

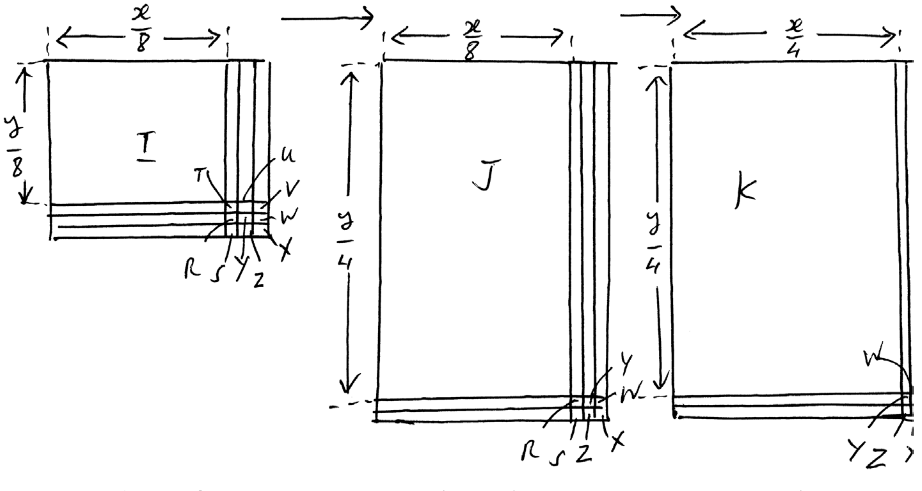


FIG. 8: One-Level Expansion of Three-Level Reduction