Layered Wavelet Coding for Video

Claudia Schremmer, Christoph Kuhmünch, and Wolfgang Effelsberg
Department Praktische Informatik IV, University of Mannheim, Germany
{schremmer, kuhmuench, effelsberg}@informatik.uni-mannheim.de

Abstract

Video coding for Internet applications faces major challenges. Due to the heterogeneity of the network, users with very different access bandwidths to the Internet want to be satisfied by the highest possible quality of their real–time application. A good coding scheme for layered video asks for maximization of the subjective visual quality at a given bandwidth, for scalability and coding complexity. In this article, we discuss and evaluate different policies for layered wavelet video coding. A heuristic of the actual bit rate originating from our implementation, the degree of scalability, and the visual quality of the coded video lead us to pronounce parameter setting recommendations.

Keywords: *Layered video, Wavelet coding, Layering policy.*

1 Introduction

Streaming video is one of the most promising Internet applications of the future. Nevertheless, a major drawback for rapid deployment is the heterogeneity of the Internet. The available bandwidth is a major parameter for the quality of real–time streaming applications: the more bandwidth available, the better the quality of the video. But available bandwidth varies from user to user. Consequently, an encoded video stream should be scalable for different network capacities. This is realized through layered video streaming. The subdivision of an encoded data stream into different layers enables the user to receive (in the ideal case) exactly as much data as his/her individual facilities allow. The more layers received, the better the quality of the video will be.

The goal of good scalability leads us to search for a good layering technique: new algorithms focus on the Discrete Wavelet Transform (DWT). Its construction through multiresolution analysis reflects the frequency resolution of human visual perception: lower frequencies are resolved well, while high frequencies are only loosely resolved. Moreover, wavelet–transformed coefficients in the scale space contain not only frequency information, but also information about the region of the original image that they encode. This mixture of frequency–time information can be successfully exploited in layered wavelet coding. Moreover, the DWT is of complexity $O(n)$. Boundary problems require some clever dealing with the edges of a signal. Nevertheless, the DWT is fast enough to allow real–time applications.

In this article we discuss three different layering policies for a wavelet–encoded video. Based on the parameters information rate, bit rate, scalability, and human visual perception, we develop a recommendation according to which the different units of information are distributed over the different layers of the video stream.

This article is organized as follows. After the presentation of related work in Section 2, we briefly explain the philosophy of hierarchical video coding in Section 3. Section 4 introduces the theory of wavelet transforms and explains how wavelet analysis and synthesis are being implemented via filter banks. The discussion of video layering in Section 5

Submitted to Workshop on Packet Video 2001, Kyongju, Korea.

constitutes the kernel of this article. We describe the parameters of a wavelet video encoder, and we detail the different policies for wavelet–encoded video layering which we evaluate in the following. The experimental setup of our implementation is described in Section 6. Our results in Section 7 precede a heuristic for the actual bit rate of the different layering policies. Recommendations for the choice of parameters in layered wavelet video coding are presented in Section 7.3. The article concludes with an outlook in Section 8.

2 Related Work

This article deals with layered wavelet video encoding and thus exploits findings of earlier works in the area of layered video algorithms. In [9] a number of typical approaches for layered video encoding are described which – similar to our approach – carry out scaling in the spatial domain. They can be split into two categories: *layered DCT* and *pyramid coding*.

Layered DCT approaches transform the video signal into the frequency domain. Unlike conventional compression methods, the DCT coefficients are not entropy encoded in a single step, rather they are distributed over several layers. Again two typical methods can be mentioned: (1) In the *layered frequencies* approach each 8×8 block of each image of the video is transformed using the DCT. Afterwards, the coefficients are quantized but they are not entropy encoded in a single step, rather they are grouped into subsets each of which is then entropy encoded on a different layer. (2) In the *layered quantization* approach each block of each image is also transformed. Afterwards the DC coefficient is quantized in a single step and the resulting value is encoded in the base layer. But instead of quantizing the AC coefficients in a single step, the precision of the value is gradually progressed. For example in the “Progressive JPEG standard” [10], each AC coefficient is refined one bit at a time. Although this technique is widely used in the Web, the successive reconstruction is quite time consuming. Therefore, [1] defines a more general approach called *LDCT*. Here, AC coefficients are encoded with 9–bit accuracy and split into four groups. The first group contains the three most significant bits of each coefficient, while the remaining three groups encode two bits each. Each group is transmitted on a different layer. At the decoder side the AC coefficients are reassembled. If the decoder receives only the base layer a coarser image can still be reconstructed.

Pyramid encoding approaches are first mentioned in [4]. Here, a downscaled version of the original image is coded on the base layer. The downscaled image is then upscaled to its original size and compared to the original. The difference is coded on the enhancement layer, resulting in a two–layer hierarchy. In order to gain more than just two layers the algorithm is recursively repeated. The philosophy of pyramid encoding bears similarities to the separable discrete wavelet transform in two dimensions, which is deployed in this article.

The new image coding standard JPEG2000 [6] [11] is also based on the wavelet transform. It supports two different component transformations: the *irreversible component transformation* is implemented via the Daubechies 9–tap/7–tap filter [2], and the *reversible component transformation* uses the Daubechies 5–tap/3–tap filter. The standard works on image tiles, thus partitions the original image into blocks. It enables the definition of regions of interest (ROI), so that certain ROIs of the image can be coded with better quality. Quantization and arithmetic entropy encoding optimize the output bit stream.

In [12] a real–time software implementation of a scalable video codec is presented. It relies on fast subband filtering of the original image with linear phase filters. The focus of this article is on entropy encoding rather than on the transform: the authors find arithmetic coding to be a major bottleneck in real–time applications. Speed is improved by replacing the arithmetic coding part with hierarchically stored block decoding.

In [3] the authors describe a layered video transmission system based on IP multicast. Video encoding is done with the subband codec described in [12], and Active Network technology is used to signal the available link bandwidths and to perform packet filtering in the multicast distribution tree.

3 Motivation for Hierarchical Video Coding

Common video encoding and compression techniques allow the adaption of the compression rate and

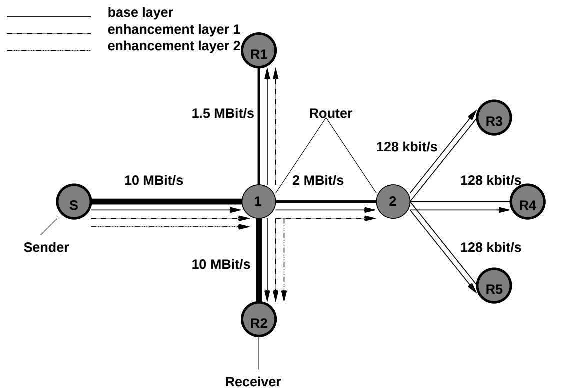


Figure 1: Layered data transmission in a heterogeneous network. The sender sends the base layer plus all enhancement layers. Each receiver chooses how many layers he/she can receive according to the bandwidth available.

thereby of the quality of the video to the (network) resources of a single receiver. These techniques fail if a video is transmitted simultaneously to several receivers with different network capacities. In order to solve this problem, *hierarchical* or *layered encoding* schemes have been developed. The idea of these schemes is to encode video signals not only into one but into several layers. Each layer l_i depends on all lower layers $l_0, ..., l_{i-1}$, thus it can only be decoded together with these lower layers. Each layer adds to the quality of the transmitted video, and each participant receives a certain number of layers, depending on his network resources. Figure 1 demonstrates an example scenario.

4 Wavelet Transform and Filter Banks

A wavelet is a compact function that vanishes outside a certain interval. The WT offers high temporal localization for high frequencies while offering good frequency resolution for low frequencies [5]. This is the reason why the WT is especially well suited to analyze local variations like in still images or in videos: a high–frequency part of an image (e.g., a transition from colored foreground to black background) will be analyzed by short, high–amplitude wavelets. Low variations (e.g., color within the same object) will be analyzed by long, low–amplitude wavelets.

An important step in the application of wavelet theory in multimedia applications, the transition from the mathematical theory to filters, has been presented by Mallat [8] through multiresolution. A multiresolution analysis is implemented via high–pass filters ($\sim$ wavelets) and low–pass filters ($\sim$ scaling functions). Low–pass filters let all frequencies pass that are below a cut–off frequency, whereas the remaining frequency components are removed from the signal. High–pass filters work vice versa. In this context, the wavelet transform of a signal can be realized with a filter bank via successive applications of a 2–channel filter bank consisting of high–pass and low–pass filters: the detail coefficients of every recursion step are kept apart, and the recursion starts again with the remaining approximation coefficients of the transform.

5 Video Layering

Scaling video decoders that are running at a variable bit rate all face the same problem: if the bandwidth at a specific moment is not sufficient for the reception of all layers, the data has to be scaled in either the spatial domain (i.e., discarding coefficients in spatial resolution), in the time domain (i.e., discarding frames), or in a hybrid spatial–temporal mixture [7]. In this paper we focus on spatially scaled video. The video sequence is thus regarded as a sequence of still images, and the DWT is performed on each single image.

In the video wavelet coding process, and the subsequent classification of the information into different layers, the following steps have to be performed: *analysis*, i.e., the calculation of the wavelet–transformed data, *layering*, i.e., a strategy to assess the wavelet data in decreasing order for human perception, and *synthesis*, i.e., the application of the inverse wavelet transform that results in the decoded video. These three steps are demonstrated in Figure 2.

5.1 Parameters

A wavelet–encoded video has numerous parameters: choice of the wavelet filters, decomposition depth of the analysis (i.e., number of recursions on the low–pass filtered part of the signal), recursion type (standard or non–standard), and layering policy. Obviously, the most important information of

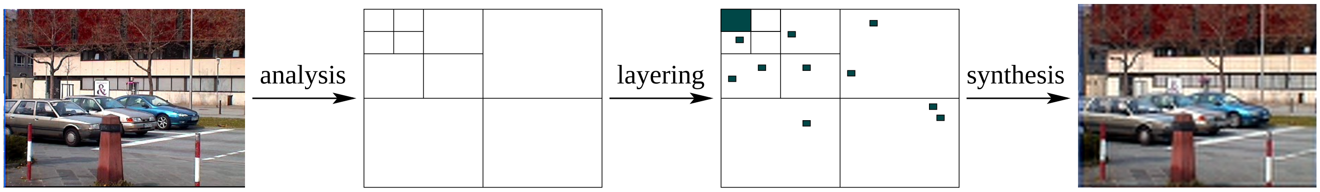


Figure 2: The process of layered video wavelet coding: the analysis (here: decomposition into 3 levels of recursion) is followed by a subdivision into different layers. The synthesis of the wavelet coefficients finally results in the decoded video sequence.

a video has to be stored in the base layer l_0, and less important information has to be stored stepwise in the enhancement layers l_i. But the *ranking* of the importance of the information depends on the layering policy.

5.2 Layering policies of DWT

The layering of wavelet–transformed data can be carried out according to the following policies (see also Figure 3).

Policy 1: Blockwise. In Section 4 we have mentioned that with wavelet decomposition, the low scales in the multiscale analysis approximate best the frequencies that are most important for human visual perception. Consequently, the layering and its respective synthesis work just the other way round: the low–pass filtered parts are synthesized first, and if there is still capacity for further synthesis, the high–pass filtered blocks are successively included in the decoding process.

Policy 2: Maximum coefficients. One could claim that the philosophy of a wavelet decomposition is to concentrate the energy of a signal (and thus the information most important to human perception of video) in those coefficients in the time–scale mixture of the wavelet domain that have maximal absolute value, no matter where these coefficients are located in the wavelet–transformed space. Consequently, the layering should look for the coefficients whose absolute value is above a certain threshold. Subsequent layers are filled with the difference data at increasing thresholds.

Policy 3: Mixture: low–pass plus maximum coefficients. A compromise would be to always synthesize the low–pass filtered part of a video (to be put in layer l_0), and if bandwidth allows, to add successively coefficients with high absolute values, no matter where they are situated in the scale space.

We have implemented the three policies and discuss the results in Section 7.

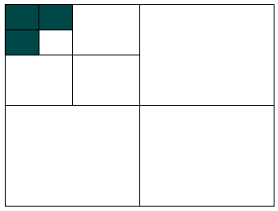
5.3 Scalability

The three layering policies presented above differ strongly in granularity. The *blockwise* policy is the coarsest one. One block of decomposition level 1 contains $1/4 = 25\%$ of information, a block of decomposition level 2 contains $1/16 = 6.25\%$, and a block of level 3 contains $1/64 = 1.5625\%$ of the information. Consequently, the granularity of the blockwise layering policy is restricted to the information levels 75%, 50 %, 25%, 18.75%, 12.5%, 6.25%, 4.6875%, 3.125%, 1.5626%,... (see Table 1).

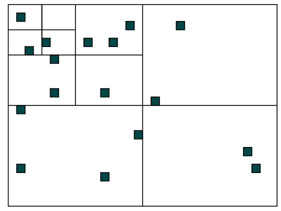
While policy 2 is almost infinitesimally scalable, the mixed policy requires that the percentage of information be at least as high as the size of the low–pass filtered part of the signal. When the percentage is exactly as high as the low–pass filtered part, the mixed policy is identical to the blockwise policy[1].

Another phenomenon shall shortly be mentioned: the visual quality of policies 2 and 3 depends highly on the decomposition depth of the image. This results from the fact that coefficients in a wavelet space where no decomposition has been executed, (or only a very rough one) still contain too much locality information. A low information percentage for synthesis might then result in many image pix-

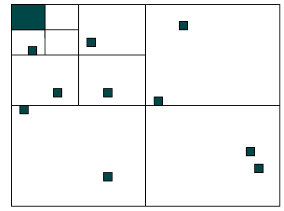
[1]Example: as 1.5625% is just the size of the low–pass filtered part of the image in three decomposition levels — and no additional information is allowed — the results of policy 1 are identical to the results of policy 3 according to the construction scheme (see Table 1, last column).



(a) Policy 1. Blockwise: Inverse order of the decomposition.



(b) Policy 2. Maximum coefficients, no matter where they are located.



(c) Policy 3. Mixture: low–pass filtered part plus maximum coefficients.

Figure 3: Layering policies of a wavelet–transformed image with decomposition depth 3. The information percentage is set to 4.6875% (i.e., 3 blocks).

(a) Decomposition depth = 1.

(b) Decomposition depth = 2.

Figure 4: Frame 21 of the test sequence "car", decoded with the layering policy 2 at 6.25% of the information. (a) was synthesized after *one* decomposition step, (b) was synthesized with the same amount of information, but after *two* decomposition steps. The original image is shown in Figure 6 (a).

els obtaining no information at all and thus staying gray instead (see Figure 4).

6 Experimental Setup

For our layered video codec, we have implemented a motion–wavelet encoder and decoder for images in CIF (*Common Intermediate Format*) format on a Linux machine. The coding language is C++.

We have implemented all orthogonal Daubechies wavelet filters from filter length 2 (Haar filter) to filter length 40 (Daubechies–20). For boundary treatment, we have implemented *circular padding*, i.e., the missing signal values on one edge of the signal are taken from the other edge. This is the only padding strategy that does not "blow up" the wavelet domain. Our test sequences contain 225 color frames. The two dimensional DWT has been realized via tensor product: all Daubechies wavelet filters are separable, so that the 2–d DWT is implemented by filtering the frames with the low–pass filter (resp. high–pass filter) first horizontally, and then vertically. The recursion has only been carried out on the purely low–pass filtered part (i.e., non–standard decomposition). As the recursion stops when the length of the wavelet–transformed signal reaches filter length, only 3 recursions have been possible for the longer Daubechies filters. To get comparable results, we have stopped recursion for *all* filters at this level.

7 Results

A good video coding scheme asks for best subjective quality at a given compression rate. Since subjective testing is very labor–intensive, time consuming, and hard to evaluate, earlier research has attempted to devise parameters to automatically evaluate the perceptual quality of a video. Human visual perception of a video sequence is still not totally understood. It proves nevertheless to be very sensible to edges and "objects". The video metric presented in [14] is used as a standard for many evaluations. Another sophisticated metric for automatic video assessment is presented in [13]. We have implemented and tested both metrics, but have not been convinced by either; for details please refer to [7]. The measure of distortion between the original sequence and a decoded sequence used in this article is thus the peak–signal–to–noise ratio[2] (PSNR), measured in decibels.

7.1 Visual quality

The evaluation of the three layering policies was carried out with comparability in mind: i.e., at the percentages of coefficient information that the blockwise layering policy meets (cf. Section 5.3). As the WT always produces many detail coefficients close to zero, visual quality for the upper percentages (75%, 50%, and 25%) was excellent for all policies and all filters. Thus, Table 1 shows the evaluation of visual perception only of the interesting lower information levels. Figure 5 is based on the values of Table 1. The distinction between different wavelet filters has been removed and has been replaced by the average PSNR value of the six wavelet filters at the given percentage.

Note that the perceived quality increases with increasing PSNR. In other words, higher values stand for higher quality. Table 1 clearly shows that visual perception (generally) increases with increasing filter length. This is known from theory and practice. A closer look at the values of the PSNR, however, shows that the PSNR sometimes decreases with increasing filter length. This phenomenon appears at a low information rate only. The explanation is that the synthesis of little information necessarily has to be erroneous, and longer filters broaden the influence of such erroneous wavelet coefficients.

Moreover, Table 1 and Figure 5 demonstrate that the visual perception of both policies 2 and 3 is very similar, and much better than the perception of the images synthesized blockwise.

Figure 6 shows the frame 21 of our test sequence "car". This test sequence contains a lot of sharp edges (lantern, pile, house in background, advertisement "&") while at the same time being composed of large uniform areas (house, cars, street, pavement). The frame has been decomposed into level 3. While images (d) and (e) do not show large differences, (c) is clearly blurred. As both layering policies 2 and 3 allow the synthesis of detail information in low layers, the reconstructed image contains parts with high spatial resolution (i.e., sharp edges) — note especially the "&" in the advertisement. In contrast, less important parts, such as the tree leaves, are resolved worse than in (c).

7.2 Bit rate

Besides the visual quality, the produced bit rate of a layering policy is an important factor. The bit rate depends heavily on the entropy encoding algorithm though. DCT–based compression algorithms like JPEG and MPEG usually use run length and Huffman encoding in order to compress the DCT coefficients. Since our layering policies lead to a huge number of zero–valued coefficients in the wavelet scale space, we assume that the same techniques (run length and Huffman) lead to good compression rates of the wavelet coefficients. Thus we suggest the following simple compression approach:

The coefficients of each recursion step in the wavelet analysis (see Figure 2) are handled separately. Starting with coefficients of the low–pass filtered part ("upper left corner"), the coefficients are processed from left to right and from top to

[2]The PSNR of each frame j has been calculated as follows:

$$10 \cdot \log \left(\frac{\sum_{xy} 255^2}{\sum_{xy} (\mathrm{org}_j(x,y) - \mathrm{dec}_j(x,y))^2} \right),$$

where $\mathrm{org}_j(x,y)$ depicts the pixel value of the original frame j at position (x,y), and $\mathrm{dec}_j(x,y)$ denotes the pixel value of the decoded frame j at position (x,y).

Quality of visual perception — PSNR									
	Percentage of maintained coefficients								
	18.75%			12.5%			6.25%		
Wavelet	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3
Haar	47.185	69.257	69.210	43.892	63.085	63.008	41.004	54.628	54.385
Daub–3	47.260	68.347	68.311	44.468	62.024	61.956	40.988	53.535	53.280
Daub–6	48.393	67.111	67.073	45.225	60.887	60.835	42.079	52.89	52.723
Daub–10	47.958	65.215	65.183	44.923	59.087	59.018	41.802	51.052	50.863
Daub–15	48.664	64.312	64.273	45.339	58.388	58.313	41.717	50.796	50.593
Daub–20	48.295	62.992	62.960	45.153	57.173	57.101	41.656	49.816	49.627
average	47.959	66.205	66.168	44.833	60.107	60.039	41.541	52.12	51.912
	4.6875%			3.125%			1.5625%		
Wavelet	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3[1]
Haar	40.57	51.505	51.088	39.047	47.341	46.435	35.210	40.882	35.210
Daub–3	40.609	50.596	50.190	39.214	46.685	45.899	37.235	40.757	37.235
Daub–6	41.64	49.969	49.599	40.077	46.275	45.602	37.041	41.253	37.041
Daub–10	41.372	48.428	48.133	39.701	45.272	44.743	36.734	40.441	36.734
Daub–15	41.291	48.176	47.850	39.644	44.951	44.370	36.817	40.136	36.817
Daub–20	41.237	47.371	47.096	39.610	44.371	43.880	36.882	40.038	36.882
average	41.120	49.341	48.993	39.549	45.816	45.155	36.653	40.585	36.651

Table 1: The PSNR of frame 21 of the test sequence "car" for different decoding policies and different percentages of restored information. Policy 1 denotes the policy of reversing the analysis and taking entire decomposition blocks in reverse construction order into consideration. Policy 2 only regards the highest absolute values in the wavelet space. Policy 3 requires the low–pass filtered part of the decomposition and fills the remaining information with the highest absolute values. See also Figure 5 for a better visualisation.

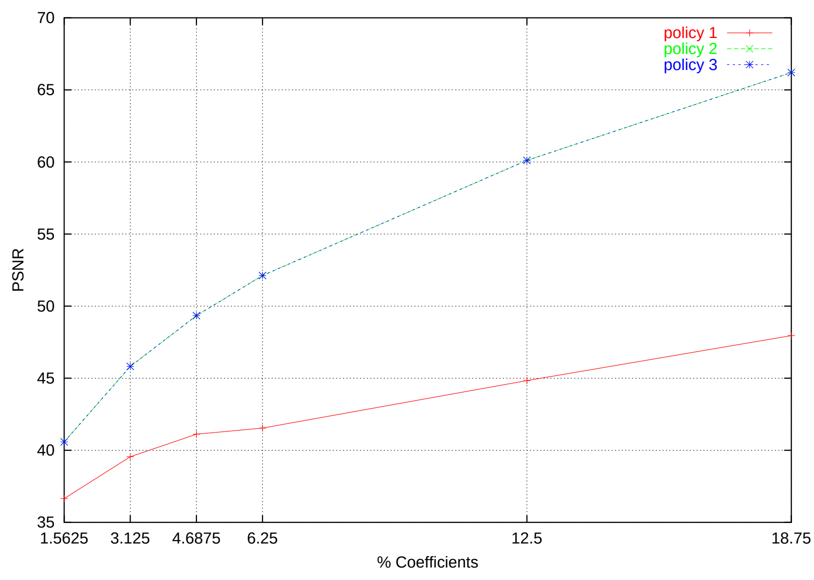
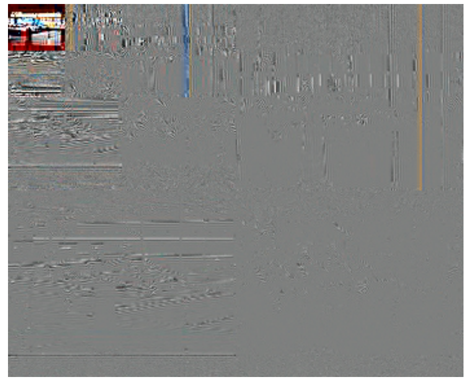


Figure 5: Run of the curve of the average PSNR value of Table 1 for different percentages of maintained wavelet coefficients. While the perceived qualities of policies 2 and 3 are so close that both curves appear identical, policy 1 produces by far the lower quality.

(a) Original frame.



(b) Wavelet–transformed.

(c) Policy 1: blockwise synthesis.

(d) Policy 2: maximum absolute coefficients.

(e) Policy 3: mixture of both.

Figure 6: Frame 21 of the test sequence "car". (a) shows the original frame. (b) visualizes the wavelet transform with a Daubechies–6 filter and decomposition depth 3. Images (c) to (e) show the syntheses of the transform with 6.25% of coefficient information.

Number of Runs (16 bit)						
	Percentage of maintained coefficients					
	18.75%		12.5%		6.25%	
Wavelet	policy 2	policy 3	policy 2	policy 3	policy 2	policy 3
Haar	24443	24388	16885	16807	8511	8323
Daub–3	23588	23557	16095	16042	7945	7821
Daub–6	23178	23137	15747	15687	7821	7654
Daub–10	23006	22972	15521	15462	7619	7484
Daub–15	23258	23214	15736	15663	7742	7605
Daub–20	23359	23312	15804	15736	7887	7711

Table 2: Heuristics for the bit rate of a wavelet encoder for frame 21 of the test sequence "car" with different wavelet filters. After Huffman coding, the bit rate might further shrink by factor 2.

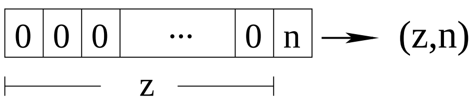


Figure 7: Definition of a *run*.

bottom into *runs*. A run (z, n) stands for an arbitrary number of $z + 1$ succeeding coefficients (see Figure 7). The first z of those coefficients are zero–valued, while the $z + 1^{\text{st}}$ coefficient has a value of $n \neq 0$. The process ends with the block of high–pass filtered coefficients ("lower right corner").

Table 2 represents a heuristic for the de–facto bit rate of a layered coder: we use 6 bits in order to encode the run length z and 10 bits to encode the non–zero value n. Thus we need 16 bits in order to encode a single run. Table 2 shows the bit rate that we have gained for each policy[3]. It can be seen that the bit rate for the two best–quality layering policies, i.e. policies 2 and 3, is close together. Policy 3 wins the competition tightly. Concerning the choice of wavelet filters, the Haar wavelet produces considerably shorter runs and thus higher bit rates. Yet the Daubechies–3 filter with filter length 6 is sufficiently regular to result in a bit rate comparable to our longest test filter, Daubechies–20 with filter length 40.

[3]Due to the enormous effort necessary to calculate symbol tables for our coefficients, we dispensed with Huffman encoding, although it can be assumed that this would lead to a further reduction of the data rate.

7.3 Recommendation

In the above discussion, we have analyzed layered wavelet coding with regard to *layering policy*, *scalability*, *visual quality*, *choice of orthogonal filter* and *expected bit rate*. In Sections 5.3 and 7.1 we have detailed why we would not consider blockwise synthesis further. Section 5.3 has also revealed that a filter length of 6 to 12 coefficients is advisable, as shorter filters produce strong artifacts (cf. Table 1, Haar wavelet) and longer filters broaden the influence of erroneous synthesis at high compression rates (cf. Table 1, Daubechies–15 and Daubechies–20 wavelets). Finally, we have analyzed the expected bit rate for a single frame of a video sequence in Section 7.2. Our tests state that the two layering policies 2 and 3 produce comparable bit rate, but policy 3 is expected to perform even better. Taking into consideration that the scalability of policy 2 is finer, we recommend to implement both layering policies, and choose one depending on the context.

8 Outlook

In this article, we have proposed a novel approach to layered video coding based on wavelets. We have analyzed the affected parameters and given a recommendation for layered wavelet video coding. As we have stated in the introduction, the wavelet transform and its synthesis are fast enough for real–time applications. Our implementation though is not yet optimized. In further steps, we will strive for a real–time implementation. We will also add Huffman encoding of the coefficients.

References

[1] Elan Amir, Steven McCanne, and Martin Vetterli. A Layered DCT Coder for Internet Video. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, pages 13–16, Lausanne Switzerland, September 1996. IEEE.

[2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image Coding Using the Wavelet Transform. *IEEE Trans. Image Proc.*, pages 205–220, April 1992.

[3] Albert Banchs, Wolfgang Effelsberg, Christian Tschudin, and Volker Turau. Multicasting Multimedia Streams with Active Networks. In *Local Computer Networks (LCN)*, pages 150–159, 1998.

[4] Peter Burt and Edward Adelson. The Laplacian Pyramid as A Compact Image Code. *IEEE Transactions on Communications*, 1983.

[5] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61. SIAM. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992. ISBN 0-89871-274-2.

[6] ITU. *JPEG 2000 Image Coding System*. International Telecommunication Union, Final Committee Draft Version 1.0 – FCD15444-1 edition, March 2000.

[7] Christoph Kuhmünch, Gerald Kühne, Claudia Schremmer, and Thomas Haenselmann. A Video-Scaling Algorithm Based on Human Perception for Spatio-Temporal Stimuli. In *Multimedia Computing and Networking (MMCN)*, http://spie.org/web/meetings/-calls/pw01/confs/EI25.html, January 2001. SPIE.

[8] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, USA, 1998. ISBN 0-12-466605-1.

[9] Steven McCanne. *Scalable Compression and Transmission of Internet Multicast Video*. Ph.D. Thesis, University of California, Berkeley, CA, USA, 1996.

[10] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard*. ISBN 0-442-01272-1. Van Nostrand Reinhold, New York, 1993.

[11] A. N. Skodras, C. A. Christopoulos, and T. Ebrahimi. JPEG2000: The Upcoming Still Image Compression Standard. In *11th Portuguese Conference on Pattern Recognition (RECPA00D)*, pages 359–366, Porto, Portugal, May 2000.

[12] W. Tan, E. Cheng, and Avideh Zakhor. Real–time Software Implementation of Scalable Video Codec. In *International Conference on Image Processing (ICIP)*, pages 17–20, Lausanne, Switzerland, September 1996.

[13] Christian van den Branden Lambrecht. *Perceptual Models and Architectures for Video Coding Applications*. Ph.D. Thesis, École Polytechnique Fédérale de Lausanne, 1996.

[14] Arthur A. Webster, Coleen T. Jones, Margaret H. Pinson, Stephen D. Vorna, and Stephen Wolf. An objective video quality assessment system based on human perception. In *SPIE Human Vision, Visual Processing, and Digital Display IV*, volume 1913, pages 15–26, San Jose, CA, February 1993. Society of Photo–Optical Instrumentation Engineers (SPIE).