A Wavelet Transform Applet for Interactive Learning

Claudia Schremmer, Christoph Esser, Christoph Kuhmünch and Wolfgang Effelsberg

Praktische Informatik IV
Universität Mannheim, Germany

schremmer@informatik.uni-mannheim.de

ABSTRACT

In recent years, new forms and techniques of teaching have appeared, based on the Internet and on multimedia applications. In the teleteaching Project *Virtual University of the Upper Rhine Valley* (VIROR), multimedia simulations and animations complement traditional teaching material. Lecturers use Java applets in their courses to explain complex structures. These are then stored in a multimedia database to enable asynchronous learning.

The wavelet transform has become the most interesting new algorithm for still image compression. Yet, there are many parameters within a wavelet analysis and synthesis: choice of the wavelet filter bank, decomposition strategy, image boundary policy, quantization threshold, etc.

We consider the wavelet transform to be a typical example of a complex, hard–to–understand algorithm that needs illustration by interactive multimedia. In this article, we present the didactic background and the implementation of a sample applet on the discrete wavelet transform, as taught in our multimedia course.

Keywords

Wavelet Transform, Image Processing, Interactive Learning, Didactic Aspects

1. INTRODUCTION

Multimedia applications offer new facilities that are conquering the world of distance education. The University of Mannheim launched a pilot teleteaching project with the University of Heidelberg in 1995, and since 1998, the four universities in the Upper Rhine Valley: Mannheim, Karlsruhe, Heidelberg and Freiburg, are affiliated in the much larger project VIROR [1]. Their engagement is twofold: (1) synchronous teleteaching between remote lecture rooms and (2) production of interactive material for both a 'live' lecture and for asynchronous learning, i.e. learning material that is stored on a multimedia database.

Factors determining the success of a lecture in this teleteaching scenario are the modularity of the lecture and the didactic concept of the modules. In traditional teaching lecturers often employ a series of still images of a time–dependent topic to visualize a concept. Their presentation then resembles a flip–book, whereby the more complex a topic is, the more pages of still images it will involve, causing students to lose sight of the general idea.

Java–based interactive demonstrations and simulations are a state–of–the–art technology which is helpful to overcome this didactic problem. But many applets just emulate a video or present a simple animation, where 'start', 'pause' and 'stop' are the only user interactions. Our experience is that an applet will successfully support the learning process only if it implements two major didactic postulates: (1) let the user play an *active* role, and (2) take the user by the hand, provide a concrete goal and *guidance*.

The wavelet transform is the most important new algorithm in image compression. Its compression performance is superior to that of the discrete cosine transform used in JPEG. Nevertheless, in the implementation of a wavelet transform, practical aspects call for attention that are not present in theoretical discussions: *choice of the wavelet filter*, *boundary extension policy*, *decomposition policy*, *decomposition depth*, and *decoding policy* for lossy coding. Theory says that longer filters possess better approximation qualities, but on the other hand, shorter filters allow a deeper descent into the iteration process, and thus achieve better concentration of the energy of an image.

This article exemplifies the advantages of interactive demonstrations, using the wavelet transform as an example. We also discuss the difficulties encountered in designing a didactically valid graphical user interface (GUI). The GUI allows the interactive experimentation with the above parameters. In order to motivate the learners, didactic design issues have been considered in the implementation. We have created a strong intuitive teaching tool to point out the concepts of the wavelet transform in image coding, along with its weaknesses and its strengths.

The article is organized as follows. In Section 2 we cite related work in the field of interactive teaching material that is realized as Java applets. Section 3 reviews the wavelet transform and details the aspects that are important for our simulation. Section 4 presents our applet in detail. It is subdivided into practical considerations and didactic considerations. Section 5 describes an empirical evaluation of the wavelet transform. As the parameter space is highly complex, and the visual implications of a parameter setting are by no means obvious to the non–expert, students were asked to judge the visual effects in different test series. The article concludes in Section 6 and gives an outlook on future work.

2. RELATED WORK

A large number of institutes are developing educational Java applets: The journal JERIC [2] is tightly related to a database [3] where teaching applets are stored. In computer science, [4][5] represent good databases for teaching applets. Within the VIROR project, a Java–based module on accounting is being developed [6], and our own courses on 'multimedia technology' and 'computer networks' are supplemented with other Java simulations [7][8].

The Web contains numerous tutorials on and simulations or demonstrations of the wavelet transform. Without giving a complete list, we cite the twin dragon applet by J. Kovačević [9], the wavelet applet by W. Sweldens [10], the wavelet tutorial by R. Polikar [11], and Armara'a wavelet page [12]. The former two applets deal with two very specific questions on the design of wavelet filters and are not suited for introductory courses. The latter tutorials provide a lot of background information on theory and application of the wavelet transform in image processing, but they are not interactive. The important step of actively involving a learner in the new topic thus remains unconsidered.

The wavelet tutorial by Matlab [13] is very powerful and allows more parameter settings than the presented simulation. However, Matlab is a commercial product which must be licensed before use. Students also have to learn the Matlab programming language in order to be able to handle its features. Our simulation pursues an alltogether different goal: to provide a powerful, easy–to–use tool free of charge.

3. THE WAVELET TRANSFORM

A wavelet is an (ideally) compact function, i.e., outside a certain interval it vanishes. Implementations are based on the fast wavelet transform (WT), where a given wavelet ('mother wavelet') is shifted and dilated so as to provide a base in the function space. The family of the shifted and dilated wavelets thus approximates an arbitrary function. In other words, a one–dimensional function is transformed into a two–dimensional space, where it is approximated by coefficients that depend on the *time* (determined by the translation parameter) and on the *scale*, i.e., frequency (determined by the dilation parameter). — By convention, the notion of time is used even for signals that depend on *location* rather than on time. Thus, a wavelet–transformed image is also located in the *time*–scale domain. — The localization of a wavelet in time spread (σ_t) and frequency spread (σ_ω) has the property $\sigma_t\sigma_\omega = \text{const}$. However, the resolution in time and frequency depends on the frequency. This is the so–called 'zoom'–phenomenon of the WT: it offers high temporal localization for high frequencies while offering good frequency resolution for low frequencies. Consequently, the WT is especially well suited to analyze local variations such as those in still images: 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.

3.1 Wavelet Transform and Filter Banks

By introducing multiresolution, Mallat [14][15] made an important stride in the application of wavelet theory in multimedia, the transition from mathematical theory to filters. A multiresolution analysis is implemented via high–pass filters, resp. band–pass filters (i.e., wavelets) and low–pass filters (i.e., scaling functions). Low–pass filters let all frequencies pass that are below a certain cut–off frequency, while removing the remaining frequency components 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 application of a 2–channel filter bank consisting of high–pass and low–pass filters: the detail coefficients (resulting from the application of the high–pass resp. band–pass filter) of every iteration step are kept apart, and the iteration starts again with the remaining approximation coefficients (from application of the low–pass filter) of the transform.

This matching of a discrete signal with a wavelet filter is realized via the mathematical notion of *convolution*. In practical regards, this means that the filter is 'laid over the signal', filter coefficients and signal coefficients that lay one upon the other are multiplied, and the filter is shifted to the next location.

Mathematically speaking, let V_i denote an approximation space, and W_i denote a detail space. Let V_0 denote our starting space, i.e., the space where the original signal 'lives'. What has been described above can then be written as

$$\begin{aligned} V_0 &= V_{-1} \oplus W_{-1} \\ &= V_{-2} \oplus W_{-2} \oplus W_{-1} \\ &= V_{-3} \oplus W_{-3} \oplus W_{-2} \oplus W_{-1} \\ &= \dots, \end{aligned} \tag{1}$$

where $\oplus$ denotes the direct sum of two spaces.

This multiresolution theory is 'per se' defined only for one–dimensional wavelets on one–dimensional signals. Application of the WT on still images requires an extension into two dimensions.

3.2 Wavelets in two dimensions

Still images are two–dimensional discrete signals. Thus, a multiresolution analysis requires two–dimensional wavelet filters. As two–dimensional wavelet filter design remains an active field of research [16][17][18], current implementations are restricted to *separable* filters. Separability denotes the fact that the successive application of a one–dimensional filter into one dimension and afterwards into the second dimension, is mathematically identical to a two–dimensional wavelet transform from the outset.

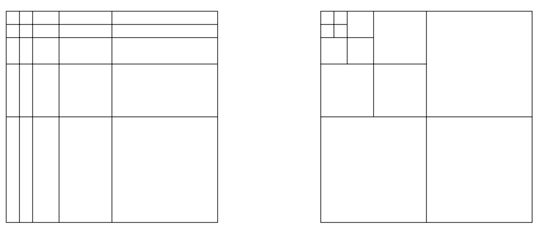
The successive convolution of filter and signal in both dimensions opens two potential iterations: non–standard and standard decompositions. When formula (1) is extended to two dimensions via the tensor product, i.e., $V_0^{(2)} = V_0 \times V_0$, the decomposition into approximation and detail starts identically in the first decomposition step:

$$\begin{aligned}
V_0^{(2)} &= V_0 \times V_0 \\
&= (V_{-1} \oplus W_{-1}) \times (V_{-1} \oplus W_{-1}) \\
&= V_{-1} \times V_{-1} \oplus V_{-1} \times W_{-1} \oplus W_{-1} \times V_{-1} \oplus W_{-1} \times W_{-1} \\
&=: (\square) \qquad (2)
\end{aligned}$$

Here, the *standard decomposition* iterates on *all* approximation spaces V_{-1}, resulting in

$$\begin{aligned}
(\square) &= (V_{-2} \oplus W_{-2}) \times (V_{-2} \oplus W_{-2}) \oplus (V_{-2} \oplus W_{-2}) \times W_{-1} \\
&\quad \oplus W_{-1} \times (V_{-2} \oplus W_{-2}) \oplus W_{-1} \times W_{-1} \\
&= V_{-2} \times V_{-2} \oplus V_{-2} \times W_{-2} \oplus W_{-2} \times V_{-2} \oplus W_{-2} \times W_{-2} \\
&\quad \oplus V_{-2} \times W_{-1} \oplus W_{-2} \times W_{-1} \oplus W_{-1} \times V_{-2} \\
&\quad \oplus W_{-1} \times W_{-2} \oplus \; W_{-1} \times W_{-1}
\end{aligned}$$

after the second iteration step, thus in 9 summands. In this sum, the only remnants of the first iteration (cf. (2)) are the details of step 1, i.e., W_{-1}. The approximations V_{-1} of the first iteration (cf. (2)) are dissected into approximations and details of the next level, i.e., V_{-2} and W_{-2}.



(a) Standard decomposition.

(b) Non–standard decomposition.

Figure 1: Two methods of decomposition in two dimensions. Here, decomposition depth=4.

The *non–standard decomposition*, however, only iterates the purely low–pass filtered approximations $V_{-1} \times V_{-1}$ and leaves the mixed terms unchanged. This results in

$$\begin{aligned}
(\square) &= (V_{-2} \oplus W_{-2}) \times (V_{-2} \oplus W_{-2}) \oplus V_{-1} \times W_{-1} \\
&\quad \oplus W_{-1} \times V_{-1} \oplus W_{-1} \times W_{-1} \\
&= V_{-2} \times V_{-2} \oplus V_{-2} \times W_{-2} \oplus W_{-2} \times V_{-2} \oplus W_{-2} \times W_{-2} \\
&\quad \oplus V_{-1} \times W_{-1} \oplus W_{-1} \times V_{-1} \oplus W_{-1} \times W_{-1},
\end{aligned}$$

thus in 7 summands. In this non–standard decomposition, the mixed terms $V_{-1} \times W_{-1}$ and $W_{-1} \times V_{-1}$ of the first iteration remain unchanged.

The difference between both decompositions is thus that the standard decomposition iterates also the parts of the approximations that are located within mixed terms, while the non–standard decomposition iterates only purely low–pass filtered components. Consequently, the standard decomposition results in many more summands in the time–scale domain. Figure 1 demonstrates the two policies in graphical form.

3.3 Image Boundary

A digital filter is applied to a signal by *convolution*. Convolution, however, is defined only *within* a signal. In order to result in a mathematically correct, reversible wavelet transform, *each* signal coefficient must enter into `filter length/2` calculations of convolution (here, the subsampling process by factor 2 is already incorporated). Consequently, every filter longer than 2 entries, i.e., every filter except *Haar* requires a solution for the boundary coefficients of the signal. Furthermore, images are typical signals with a relatively short signal length (in rows and columns), thus the boundary treatment is even more important than e.g. in audio coding. Two common boundary policies are circular convolution and padding.

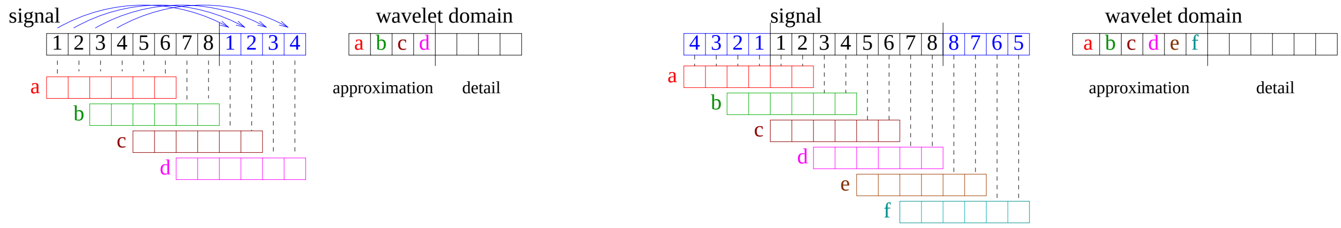
3.3.1 *Circular Convolution*

The idea of circular convolution is to 'wrap' the signal around at the boundary, i.e., wrap the end of a signal to the beginning (or vice versa). Figure 2 (a) illustrates this approach. In so doing, circular convolution is the only boundary treatment that maintains the number of coefficients for a WT, simplifying storage handling. However, the time information contained in the time–scale domain of the wavelet transformed coefficients 'blurs': the coefficients in the time–scale domain that are next to the right border (resp. left border) also affect signal coefficients that are located left (resp. right). In the example of Figure 2 (a), this means that information on pixels 1 and 2 of the left border of the original signal is contained in the entries a, c and d of the time–scale domain. c and d are the coefficients that due to circular convolution contain information on the 'other' side of the signal.

3.3.2 *Padding Policies*

Padding policies have in common that they add coefficients to the signal on either border. The border pixels of the signal are padded with `filter length-2` coefficients. Consequently, each signal coefficient enters into `filter length/2` calculations of convolution, and the transform is reversible. Many padding policies exist: *zero padding* where 0's are added, *constant padding* where the signal's boundary coefficient is padded, *mirror padding* where the signal is mirrored at the boundary, *spline padding* where the last n border coefficients are extended by spline interpolation, etc. All padding policies have in common that storage space in the wavelet domain is physically enlarged at each iteration step (cf. Figure 2 (b)). A strength of all padding approaches, however, is that the time information is preserved.

A comparison of the iteration behavior between both boundary policies states the following. Convolving the signal with a filter is only reasonable for a signal length greater than the filter length, and each iteration step reduces the size of the approximating signal by factor 2. Both policies stop iteration when the signal length has shrunk to filter length. Consequently, the decomposition depth for circular convolution varies with the filter length: the longer the filter, the fewer decomposition iterations are possible. With padding, however, the iteration depth is independent of the selected wavelet filter bank; only the size of the approximation in each iteration level varies with the filter bank (see Section 4.1.1 and Table 1 for more details).



(a) Circular Convolution.

(b) Mirror Padding.

Figure 2: Circular convolution versus mirror padding for a signal of length 8 and a filter of length 6. Here, the filter is a low–pass filter, thus the coefficients resulting from the convolution form the approximation entries. In (a), the approximation contains half as many entries as the original signal. Together with the details, the entries of the wavelet domain require the same storage space as the original signal. In (b), the padding results in inflated storage space in the wavelet domain.

3.4 Synthesis Strategies

Until now, we have discussed some problems and solutions with wavelet analysis, i.e., forward transform. The decoding process that transform coefficients of the time–scale domain back into signal coefficients is called *synthesis*. A wavelet transform is mathematically reversible. For compression reasons though, one is interested in a well–directed discarding of information. Of course, this discarding should result in minimum visual deterioration.

One of the reasons for the success of the wavelet transform in image processing is the WT's quality to decompose an image into multiresolution levels of the same image, which reflects the human visual perception: when an image is being presented to a person, he/she first resolves the greater context of the situation: a car, a donkey, a crowd of people. Subsequently, more and more details enter the perception: the model and color of the car, the individuals in the crowd. Finally, details might be resolved: scratches in the paint, expression of joy in a face [19].

In this section, we discuss two different discarding policies for data compression, and two policies to represent a signal in coarser resolution.

3.4.1 Compression

In transform–based image compression, data reduction is a three–step process: discarding information in the time–scale domain, run–length encoding and Huffman encoding. In this article we concentrate on information discarding only. Two major discarding policies prevail: block–wise discarding and quantization.

The *block–wise discarding* policy takes advantage of the idea that a wavelet transform has been *constructed* such that the least important information, the details, are separated out first. Inversely, the approximations are supposed to contain the most important information. They are decoded first. If capacity allows, time–scale blocks containing detail information are subsequently added in the decoded image. If not, whole blocks of detail coefficients are discarded. A major drawback to this scaling approach is the very coarse granularity. In non–standard decomposition, the first iteration results in 4 blocks, thus it would allow to synthesize either 25% of the information, or 50%, 75%, or 100%. Even though the blocks shrink with increasing steps of the iteration, this policy is not arbitrarily scalable.

Quantization is a better approach. Here, a certain threshold is set. All coefficients of the time–scale domain that are below this threshold are discarded. This policy implements the hypothesis that large absolute coefficients contain the visually important information. The approximation coefficients normally surpass the threshold and are maintained. The difference to the above discarding policy is that large coefficients of the detail blocks are now maintained. Furthermore, this discarding approach is arbitrarily scalable by adjusting the value of the threshold.

4. THE WAVELET TRANSFORM APPLET

For teaching purposes the wavelet transform for still images, as described above, was implemented as a Java applet. We used Java 1.3, the swing GUI classes and the Java Advanced Imaging (JAI) package. The WT demonstration enables the user to experiment with all the different aspects of the discrete wavelet transform on still images described in Section 3. In this section, we focus on practical considerations as well as on didactic issues in the design of the applet.

Since the theoretical presentation of a demonstration (like in this article) cannot substitute practical experience, we highly recommend that the reader access the applet [20] and play around with its features.

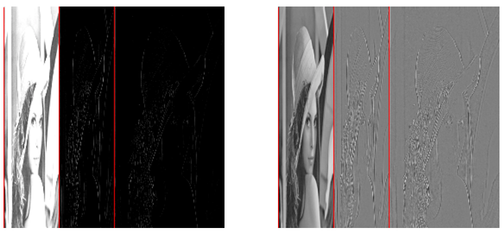
4.1 Practical Considerations

The practical considerations illuminate two important factors of the applet: visualization of the time–scale domain and representation of synthesis–in–progress. Finally, the implemented heuristic for quantization is discussed.

4.1.1 'Painting the Time–Scale Domain'

The applet visualizes the coefficients in the time–scale domain. As the wavelet–transformed coefficients are *not* pixel values, different aspects need to be considered: normalization and range.

Normalization. The orthogonal Daubechies–n filters discussed in this article have the property that the sum of the low–pass filter coefficients is $\sqrt{2} > 1$. Application of this filter on a signal thus raises the average luminance by $\sqrt{2}$. Pixel values though can be painted only in the range 0 (black) to 255 (white). One way out is to set all pixel values in the time–scale domain brighter than 255 to 255 (cf. Figure 3 (a)). Similarly, the high–pass filter results in detail information towards the approximation. In other words, the details specify the variation of a specific pixel towards an average. This variation can be positive or negative. One could draw these coefficients by cutting off the negative parts and considering only the positive values (Figure 3 (a)). With



(a) No normalization.

(b) Normalization.

Figure 3: The two possible realizations of 'painting the time–scale coefficients' (Daubechies–2 wavelet filter, standard decomposition).

normalization, we denote the effect that the coefficients in the time–scale domain are 'edited' before they are visualized. Therefore, the coefficients in the low–pass filtered regions are divided by powers of $\sqrt{2}$. This makes the total luminance of the approximation remain constant. The high–pass filtered coefficients are elevated by 128, so that former negative variations appear darker and former positive variations appear brighter (cf. Figure 3 (b)).

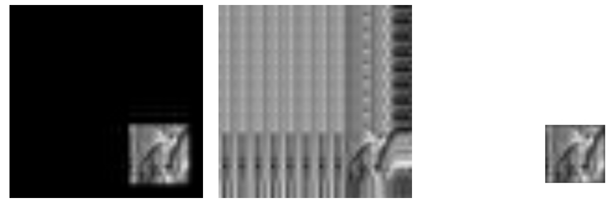
Growing Spatial Rage with Padding. As we have discussed in Section 3.3, boundary padding policies result in an 'enlarged' time–scale domain. This enlargement increases with every iteration. Moreover, only the iterated (low–pass filtered) parts are inflated, thus the time–scale domain does not grow symmetrically.

We illustrate the problem with an example. We analyze an image of size 256 × 256 pixels with the Haar filter and the Daubechies–20 filter. The decomposition policy is non–standard, the boundary is treated with zero padding. Table 1 shows the size of the purely low–pass filtered part (i.e., left upper corner) in each iteration step.

Consequently, the coefficients in the time–scale domain in the example with the Daubechies–20 filter contain many 'padded' coefficients, and only a minor number of 'real' approximation coefficients. When the time–scale domain of a wavelet–transformed image with padding policy is visu-

Level of iteration	**Size of 'upper left corner'**	
	Haar	**Daub.–20**
1	128 × 128	147 × 147
2	64 × 64	93 × 93
3	32 × 32	66 × 66
4	16 × 16	52 × 52
5	8 × 8	45 × 45
6	4 × 4	42 × 42
7	2 × 2	40 × 40
8	1 × 1	39 × 39

Table 1: The size of the time–scale domain with padding depends on the wavelet filter.



(a) All coefficients in the time–scale domain with zero padding.

(b) All coefficients in the time–scale domain with mirror padding.

(c) ... and 'what we would prefer for painting'.

Figure 4: 'Trimming' the approximation with zero padding and mirror padding. The parameters have been set to non–standard decomposition, Daubechies–20 wavelet filter bank, and iteration level 4.

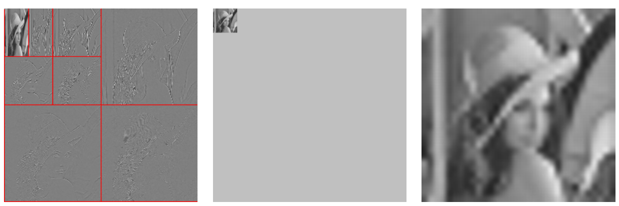
alized, we actually 'cheat' a bit as we cut off the padded coefficients from visualization. Figure 4 illustrates the problem. This raises a new question: how can we distinguish the 'real', i.e., approximating coefficients in the time–scale domain from the padding coefficients? The size of the 'real' approximation coefficients at each level is known. The method of finding them has been realized differently for zero padding and mirror padding.

With zero padding, the implementation supposes that the original image is not all black. An iteration on the rows and columns of the image then searches for non–black boundary pixels (cf. Figure 4 (a)). As the target size of the 'real' approximation is known, this approach is stable even given some black border pixels.

Mirror padding does not allow the same easy approach. Figure 4 (b) illustrates that the low–pass filtered coefficients in the time–scale domain with mirror padding extend in each iteration with mirrors of the image's borders. These have the same gray values as the original image, however; thus detection of the approximation signal by comparison of the gray values to the 'padded' coefficients would not work. Our solution was to cut out a piece of the target size from the middle of the corresponding time–scale domain. As the 'real' approximations are not necessarily in the middle (cf. Figure 4), this approach is unstable, i.e., the deep iteration steps might draw coefficients in the low–pass filtered parts of the image that signify padding rather than 'real approximation'.

4.1.2 Representation

The synthesis reverses the analysis, thus the synthesis starts with the low–pass filtered part and subsequently adds information contained in the band–pass and high–pass filtered regions of the time–scale domain, which increases spatial resolution. Independently of whether information in the time–scale domain has been discarded or not, there are three ways to represent the subsequent resolution of an encoded image: a synthesis–in–progress can be represented by reversal of the analysis, by growing spatial resolution or by interpolation. Figure 5 demonstrates the three representation policies.



(a) Analysis reversal. (b) Growing spatial resolution. (c) Interpolation.

Figure 5: Representation of synthesis–in–progress of an 256×256 gray–level image 'Lena'. The image is analyzed using the Daubechies–2 wavelet filter, non–standard decomposition of depth 7, and circular convolution. (a) Analysis reversal at level '3.5'. (b) and (c) The low–pass size of the image is 32×32.

Analysis Reversal is the canonical way. The details are painted, and the synthesized image 'grows blockwise'. The screenshot shows the process when the vertical details of level 4 have already been added, but the horizontal details have not (thus level '3.5').

Growing spatial resolution 'draws' only the purely low–pass filtered approximation. When the synthesis starts, the approximation is a very small image (in the extreme, 1×1 pixel, depending on the parameters). Subsequently, as more and more information is added, the spatial size of this approximation continues to grow until it has reached the size of the original. This approach implements growth in the form of the Laplacian pyramid [21].

Interpolation always inflates the current approximation to the original size of the image and adds missing pixels by interpolation. The question remains which interpolation strategy shall be implemented: simple 'cloning', linear interpolation, bilinear, cubic, or bicubic — there are many options. In general, visual results are acceptable with cubic interpolation.

4.1.3 Quantization

The applet implements a heuristic for quantization. Figure 6 (a), 'step 3: Synthesis parameter selection' contains a selection box for an integer quantization threshold. The coefficients in the time–scale domain that are below the selected threshold are discarded. Since the approximation coefficients approximate the original signal, they constitute the most important part of the time–scale domain. Thus, the purely low–pass filtered coefficients shall not be diminished in quality, and we have implemented the quantization such that it operates on (at least once) high–pass filtered coefficients only.

4.2 Didactic Considerations

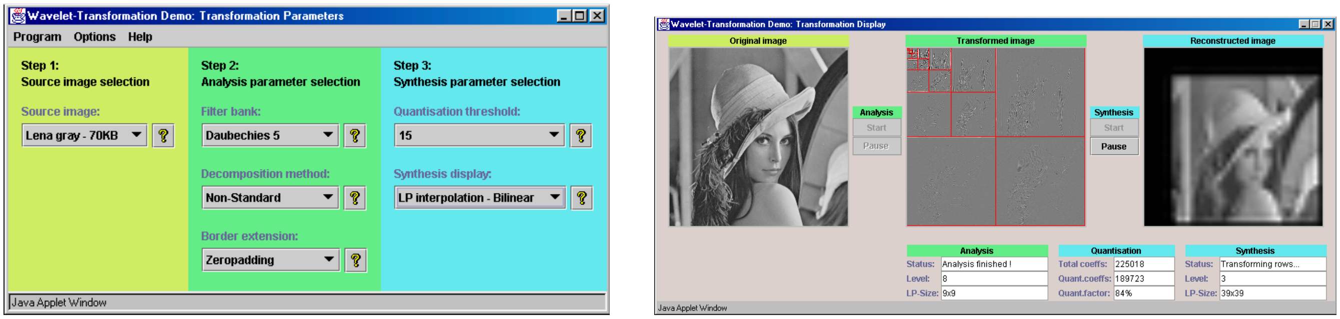
Pedagogic evaluations have proven that a learner's capability of imagination decreases with an increasing level of abstraction [22][23][24]. Thus, a topic can be imagined and reproduced by a student only as long as its complexity does not exceed a certain level. Highly abstract themes, though, will never be totally understood as long as there are no means of visualization. The better this visualization, the greater the learning success.

As far as the wavelet transform is concerned, the learning target of a student is to fully understand the concept of this transformation. At the end, he/she should be able to answer questions such as:

- How do frequency transforms work?
- What is the nature of a time–scale domain?
- What is the conceptual difference between standard decomposition and non–standard decomposition?
- What is the conceptual difference between the different boundary treatment policies?
- Why does the achievable iteration depth depend on the boundary policy?
- What are the most relevant differences between the different Daubechies–n wavelet filters?
- What is quantization? / How is it used in the context of the WT?
- What kind of synthesis strategies exist? / What strengths and weaknesses do they have?
- What influence do the parameter settings have on the decomposition process and image quality?

The GUI of the demonstration is divided into two parts, the *parameter definition* window (see Figure 6 (a)) and the *transform visualization* window (see Figure 6 (b)). Each window is structured from left to right. This is the normal direction of reading and viewing for Westerners. Three different background colors in the parameter definition window can be found again in the transform visualization window. They indicate the subdivision of parameters and visualization into the following fields: source image, analysis, and synthesis. This simple color analogy makes it intuitively clear that the parameters set in one window influence the specified part of the transformation in the other window.

The use of color, however, is only a first step towards a user–friendly interface. Many Java applets are designed by people with a good technical knowledge of a certain topic.



(a) Parameter definition.

(b) Transform visualization.

Figure 6: The two windows of the wavelet transform applet on still images. The structure is organized from left to right. The color analogy between both windows clarifies the correlation.

Nonetheless, the challenge remains how to make a simulation intuitive for those who work with it for the first time. Our experience indicates that a detailed explanation of the background, the purpose of the demonstration, and the interaction possibilities must be provided. In addition, in order to motivate students, a good applet has to provide an incentive, a stimulus to work with it. In traditional teaching, this stimulus is often a test or an exam. In self–paced learning, this incentive is necessarily of a different nature. In order to motivate the students, our applet implements three major didactic concepts:

- The concept of *user guidance* takes the user by the hand. For example, subtitles within the action panel are provided that indicate 'step 1', 'step 2' and 'step 3' (cf. Figure 6 (a)).
- *Deactivation of buttons*: The visualization window depends on the parameters set in the parameter definition window, and on the current phase of the program. With the specified parameters, the analysis and synthesis are calculated and visualized. While a calculation is in progress, the only action possible is to 'pause'. Only after calculation is complete, are all other actions enabled again. This is made obvious by the deactivation of certain buttons during the analysis or synthesis phase.
- *Context–sensitive help*: An extensive help menu has been implemented. The menu button 'help' in the parameter definition window opens the index page from where every topic can be accessed. Moreover, small '?'–icons open the same help window, but jump this time in a context–sensitive fashion to the entry explaining the parameter in question.

5. EMPIRICAL EVALUATION

With our wavelet transform applet, we have performed subjective tests with our students on the technical aspects of the WT. They were asked:

- What does horizontal and vertical filtering mean? / Where is the specific horizontal (resp. vertical) information to be found?
- What influence does the filter length have on the decomposition depth, i.e., the number of possible iteration steps?
- What is the role of quantization, and how should quantization parameters be chosen?

The parameters of the wavelet transform form such a complex scheme that the implication of a parameter setting is not obvious to the non–expert. Table 2 gives empirical results in terms of *heuristic for compression ratio*, i.e., percentage of discarded information in the time–scale domain, and *subjective visual quality* for different settings of the parameters *wavelet filter*, *decomposition policy*, *boundary policy* and *quantization factor*. The test was performed in various test series. In series (a), the role of the quantization factor on the quality of the decoded image was analyzed. Test series (b) aimed at illuminating the role of the boundary policy, while all other parameters were kept unchanged. In (c), the boundary policies and their impact on the quantization threshold were examined. Especially the visual difference between the boundary policies when all but the approximation coefficients were discarded was evaluated. Series (d) finally combined two comparisons: influence of the filter length, and influence of the decomposition policy.

With the help of the applet on the wavelet transform, our students were able to categorize visual phenomena and to express parameter setting recommendations. Concerning e.g. the choice of the wavelet filter, the following statements were made: short wavelet filters (e.g., Haar, Daubechies–2) produce strong block artifacts as the time–influence of the filters is very limited, and thus the transition between different color regions can get very harsh. Very long filters (e.g., Daubechies–10 and longer) result in strong flickers of the decoded image as the time–influence of distorted coefficients in the time–scale domain is immense. Best visual results were obtained with medium–length wavelet filters like Daubechies–4, and Daubechies–5.

Our experience shows that students value this applet very highly since the WT is a very complex and abstract function, not easy to understand from text books.

	Parameter				Result	
	Wavelet Filter	**Decomposit. policy**	**Boundary policy**	**Quant. threshold**	**Discarded information**	**Subjective visual quality**
series (a)	Haar	non–standard	all	1	15%	no visual losses
	Haar	non–standard	all	10	76%	slight artifacts, particularly in smooth regions
	Haar	non–standard	all	45	95%	stronger artifacts; color gradients are represented by only one value
	Haar	non–standard	all	max	100%	only the mean gray value of the image remains
series (b)	Daubechies–20	non–standard	zero padding	10	83%	better quality than same parameters with Haar filter; horizontal and vertical strip–artifacts at the borders
	Daubechies–20	non–standard	mirror padding	10	64%	same quality as zero padding, but no strip–artifacts at the image boundary
	Daubechies–20	non–standard	circular convolution	10	77%	similar to mirror padding; slightly blurry image
series (c)	Daubechies–20	non–standard	all	45	$83 - 96\%$	strong blurs
	Daubechies–20	non–standard	all	max	100%	strong differences between the boundary policies; mirror/zero padding: decomposition until level 8, $\rightarrow$ synthesis as mean value; circular conv.: synthesized image is the approximation at level 3
series (d)	Daubechies–20	non–standard	circular convolution	25	91%	strong flickers
	Daubechies–5	non–standard	circular convolution	25	91%	better quality than same parameters with Daubechies–20; good compromise between quality and compression factor
	Daubechies–5	standard	circular convolution	25	92%	worse quality than non–standard decomposition; blurry

Table 2: The parameter space of the wavelet transform allows the setting of the many parameters. The effects on visual quality as well as on compression ratio are not evident. This table shows the percentage of discarded information in the time–scale domain and the subjective visual quality of the gray–level image 'Lena' (256×256 pixels) with different parameters.

6. CONCLUSION AND OUTLOOK

We have presented a highly interactive Java applet illustrating the wavelet transform for still image coding. In interactive learning, not only do the algorithms have to be implemented carefully, but also the didactic issue of motivation becomes important. A GUI steers all user interactions. The applet allows many parameter settings and combinations. It displays not only the original and the decoded image, but also the coefficients in the time–scale domain, as the wavelet transform allows an easy interpretation of these coefficients.

Our future work will focus on the (technical) improvement of the representation of the 'real' approximation in the time–scale domain for mirror padding, and we will investigate better padding policies for the borders.

Concerning the underlying didactic assumptions of the presented applet, we will perform a thorough didactic evaluation on *New media in teaching versus traditional teaching* together with the Department of Education of the University of Mannheim on 2×60 students in the upcoming summer semester (SS2001). Early results are expected by mid–2001.

7. REFERENCES

[1] VIROR. Cooperation Project 'Virtuelle Hochschule Oberrhein', "Universities Freiburg, Heidelberg, Karlsruhe, and Mannheim," http://www.viror.de/en/, 1998.

[2] ACM, "Journal on Educational Resources in Computing (JERIC)," http://fox.cs.vt.edu/JERIC/, 1999.

[3] Edward A. Fox, "Computer Science Teaching Center," http://www.cstc.org/, 1999.

[4] John Morris, "Lecture on Data Structures and Algorithms (including interactive media)," http://ciips.ee.uwa.edu.au/˜morris/Year2/PLDS210/, 1998.

[5] Abdulmotaleb El Saddik, "The itBeanKit Site," http://www.kom.e-technik.tu-darmstadt.de/projects/-iteach/itbeankit/html/, 1999.

[6] VIROR. Verbundprojekt Virtuelle Hochschule Oberrhein, "Java–basierte Kosten– und Erlösrechnung (JOKER!)," http://viror.wifo.uni-mannheim.de/downld/joker/, 2000.

[7] Claudia Schremmer, Christoph Kuhmünch, and Holger Wons, "Simulations in Interactive Distance Learning," [submitted for publication], 2001.

[8] Claudia Schremmer and Christoph Kuhmünch, "Simulation applets in *Multimedia Technology* and *Computer Networks*," http://www.informatik.uni-mannheim.de/informatik/pi4/stud/animationen/, 1998–2001.

[9] Jelena Kovačević, "Twin dragon applet," http://cm.bell-labs.com/who/jelena/TwinDragon/, 1997.

[10] Wim Sweldens, "Cascade applet," http://cm.bell-labs.com/who/wim/cascade/, 1997.

[11] Robi Polikar, "The Wavelet Tutorial," http://www.-public.iastate.edu/˜rpolikar/WAVELETS/wavelet, 1996.

[12] Armara Graps, "Armara's Wavelet Page," http://www.armara.com/current/wavelet.html, 1995.

[13] The MathWorks, "Matlab," http://www.mathworks.com, 1994–2001.

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

[15] Stéphane Mallat, "A Compact Multiresolution Representation: The Wavelet Model," *IEEE Computer Society Workshop on Computer Vision (WCV)*, vol. 87, pp. 2–7, 1987.

[16] Jelena Kovačević and Wim Sweldens, "Wavelet Families of Increasing Order in Arbitrary Dimensions," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 480–496, March 2000.

[17] Jelena Kovačević and Martin Vetterli, "Nonseparable Two– and Three–Dimensional Wavelets," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1269–1273, May 1995.

[18] David B. H. Tay and N. G. Kingsbury, "Flexible Design of Multidimensional Perfect Reconstruction FIR 2–Band Filters using Transformations of Variables," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 466–480, October 1993.

[19] John P. Frisby, *Seeing — Illusion, Brain and Mind*, Oxford University Press, Walton Street, Oxford, 1979, ISBN 0-19-217672-2.

[20] Claudia Schremmer and Christoph Esser, "Simulation of the Wavelet Transform on Still Images," http://www-mm.informatik.uni-mannheim.de/veranstaltungen/animation/multimedia/wavelet/-WaveletDemo.html, 2001.

[21] Peter Burt and Edward Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, vol. COM-31, no. 4, pp. 532–541, April 1983.

[22] Tia Hansen, Lone Dirckinck-Holmfeld, Robert Lewis, and Jože Rugelj, "Using Telematics for Collaborative Learning," in *Collaborative Learning: Cognitive and Computational Approaches*, Pierre Dillenbourg, Ed. Elsevier Science, Oxford, 1999.

[23] Holger Horz, Andrea Buchholz, and Manfred Hofer, "Neue Lehr-/Lernformen durch Teleteaching?," *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, vol. 23, no. 3, pp. 129–136, 2000.

[24] Michael Kerres, *Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung*, Oldenbourg, München, Germany, 1998, ISBN 3-486-24539-2.