17th International Conference on the Application of Computer
Science and Mathematics in Architecture and Civil Engineering
K. Gürlebeck and C. Könke (eds.)
Weimar, Germany, 12–14 July 2006

SUBDIVIDE AND CONQUER RESOLUTION

A. Bultheel*, **M. Jansen, J. Maes, W. Van Aerschot, and E. Vanraes**

**Dept. Computer Science, K.U.Leuven*
Celestijnenlaan 200A, B-3001 Leuven (Belgium)
E-mail: adhemar.bultheel@cs.kuleuven.be

Keywords: subdivision, wavelets, lifting, signal processing, image processing, geometric modeling, visualisation, CAGD, normal offset, Powell–Sabin spline

Abstract. *This contribution will be freewheeling in the domain of signal, image and surface processing and touch briefly upon some topics that have been close to the heart of people in our research group. A lot of the research of the last 20 years in this domain that has been carried out world wide is dealing with multiresolution.*

Multiresolution allows to represent a function (in the broadest sense) at different levels of detail. This was not only applied in signals and images but also when solving all kinds of complex numerical problems. Since wavelets came into play in the 1980's, this idea was applied and generalized by many researchers. Therefore we use this as the central idea throughout this text.

Wavelets, subdivision and hierarchical bases are the appropriate tools to obtain these multiresolution effects. We shall introduce some of the concepts in a rather informal way and show that the same concepts will work in one, two and three dimensions. The applications in the three cases are however quite different, and thus one wants to achieve very different goals when dealing with signals, images or surfaces.

Because completeness in our treatment is impossible, we have chosen to describe two case studies after introducing some concepts in signal processing. These case studies are still the subject of current research. The first one attempts to solve a problem in image processing: how to approximate an edge in an image efficiently by subdivision. The method is based on normal offsets. The second case is the use of Powell–Sabin splines to give a smooth multiresolution representation of a surface. In this context we also illustrate the general method of construction of a spline wavelet basis using a lifting scheme.

1 INTRODUCTION

The birth of wavelets in the 1980's was not a coincidence. The wavelet concept is the crystalization of many ideas that were floating around in those days: time-frequency analysis, short-time of windowed Fourier transform, filter banks, multiresolution, etc.

At the same time, numerical analysts were using techniques whereby a solution to a numerical problem (e.g. the solution of a partial differential equation) was computed at several scales or resolution levels. First a rough approximation of the solution is computed. Then gradually more and more detail is added in an adaptive way. This means that the solution is only refined —and thus the extra computations are only performed— in those regions where more detail is actually needed.

In theoretical physics several researchers were playing with the idea of a multiresolution analysis. That is a mathematical construct that formalizes the idea that we can represent a function at different scales.

The Internet also stimulated the idea of progressively transmitting an image. That is again by first giving a rough idea of the image, and gradually adding details.

This is in a sense the opposite of image compression. To compress an image we have to throw away some of the information. The first things to be removed are the (less important) details on a fine scale. These are usually represented by the high frequency components. With the remaining information, one will still have a good idea of what the image looks like. Of course, depending on the application, there is a limit to what can be removed without harming the essential information. Therefore several layers of detail will be considered.

Just passing the signal through a low-pass filter, removing the high frequencies is however somewhat simplistic. In an image, high frequency components are important in some regions, while they are irrelevant at other places. Thus the idea is rather to throw away the (small) components that do not contribute to the essential information. These components can be high or low frequency. So we have to know which frequencies (i.e., which level of detail) is important or unimportant at what place. It is like a music score where you have to know not only what tones you have to play, but also at what time each tone should appear.

The same idea is of course important in computer aided geometric design. While designing a part of a car, you want to shape it by modifying the surface only locally, but still keep a smooth surface. This is obtained by moving certain so called control points. The surface is like a rubber fleece hanging on strings attached to these control points. Somehow one wants to have control over the effect that results from moving a control point. Sometimes the effect is desired to be global, sometimes is should be very local.

Assume for example that a scan has been made of some character that is used in a computer animated game. The shape of the body of the character is described by such control points. If you want to make for example the person larger or smaller, you may do that by moving such a control point. This should have an effect over the whole body. But if one wants to change the shape of the nose, then this should also be done by moving a control point, but the effect should be on the nose, and the immediate neighborhood, but certainly not on the whole body. Thus one should be able to control the shape at several resolution levels. Again the change is a result of moving a control point at a certain place and at a certain level of detail. Here, as in the other applications, this multiresolution representation can be obtained by subdivision.

Starting from a rough representation of our shape at a low resolution level by using a net of only few control points, we can subdivide this net to form a finer net with more control points, giving a representation at a higher resolution level. And this subdivision step can be repeated to whatever level of resolution one wants to go.

This brings us to more modern applications, but in other, more technical situations, these ideas were also ripening back in the 1980's.

So, along with wavelets came the idea of multiresolution, or vice versa. These tools existed back in the early days, but now, about twenty years later, they came to a much more mature age. Generalizations in all possible directions in theory and applications came about, and it is far beyond the scope of this brief survey to include all possible aspects involved.

Therefore we had to make a choice of what to include, without being too technical. We have chosen to introduce several of the previous ideas in a rather informal way, using a level of mathematics that is within the reach of most readers. This starts with the introduction of wavelets. The construction of wavelets is placed within the framework of the lifting scheme, which allows for far going generalizations. To keep the concepts as simple as possible, we shall introduce them using 1D signals.

The generalization to 2D signals (images) is straightforward. Although image processing is one of the areas where wavelets have been most successful, they do not solve everything. As an example we give a brief introduction to normal offsets. This is a technique where subdivision is used that is adapting to (smooth) edges in the image. This is still ongoing research.

Subdivision is again the keyword in the sections where we illustrate another example of current research. The representation of 3D objects with a certain degree of smoothness can be obtained using smooth components (splines) that allow for a multiresolution structure (Powell–Sabin splines). By designing a strategic subdivision scheme, these components do maintain a smooth surface, yet can be decomposed into several levels of detail.

2 THE GENESIS OF WAVELETS

Fourier analysis has been *the* (and still is *an*) essential tool for the analysis of signals and images. The fundamental result of Joseph Fourier (1768-1830) was that every sufficiently smooth function can be written as a linear combination of sines and cosines. Each of these sine/cosine terms corresponds to a specific frequency.

Because $e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$, we can, instead of using sines and cosines as basis functions, also use $e^{\pm i\omega t}$ as basis functions. Thus, when f belongs to a moderately restricted class, we can represent it as a linear combination of these basis functions. I.e.

$$f(t) = \int_{-\infty}^{\infty} c(\omega)e^{i\omega t}d\omega, \quad t \in \mathbb{R}.$$

For 2π-periodic functions, we can write this in a discrete form because only the 2π periodic versions of $e^{i\omega t}$ are needed, i.e., for $\omega \in \mathbb{Z}$:

$$f(t) = \sum_{\omega=-\infty}^{\infty} c(\omega)e^{i\omega t}, \quad t \in [0, 2\pi].$$

Thus we write our function as a linear combination (continuous or discrete) of basis functions. Let us use the discrete notation for simplicity:

$$f(t) = \sum_k c_k \psi_k(t), \quad \text{with } \psi_k(t) = e^{ikt}.$$

This choice of the basis has some advantages.

- All basis functions are generated by some “mother” function e^{it} which is dilated to generate all the (relevant) frequencies $\omega = k \in \mathbb{Z}$.
- The basis functions form an orthonormal basis $\langle \psi_k, \psi_\ell \rangle = \delta_{k-l}$ where

  $$\langle f(t), g(t) \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t)\overline{g(t)}dt.$$

  This is important because the best (with respect to the norm in this inner product space) low frequency approximation f_n to f is obtained by simply truncating the series:

  $$f_n(t) = \sum_{|k| \le n} c_k \psi_k(t).$$

The biggest disadvantage is that each basis function ψ_k is supported on the whole real line. This is not what we want because changing one of the coefficients c_k, will influence the value of $f(t)$ for all t. The change does not have a local effect. The fact that we have infinite support for the ψ_k is directly connected to the fact that each of these basis functions contains exactly one of the frequencies ω.

The interconnection is a consequence of the *Heisenberg (1901-1976) uncertainty principle* [32, 26, 59]. It says that the smaller the frequency content of a function, the larger its support will be and vice versa. More precisely, consider the function f with finite norm as being a probability density function (up to normalization) of a stochastic variable and denote by σ^2 the variance of that variable, then σ will give an indication of how much the function is “spread out” about its average. Let $\hat{f}$ be the Fourier transform of f to which we associate a variance $\hat{\sigma}^2$ in a similar way, then the uncertainty principle states that $\sigma^2\hat{\sigma}^2 \ge 1/4$. Thus the smaller the support in the t-domain, the wider the support in the frequency domain.

The Fourier basis is an extreme situation of wat is possible within the uncertainty principle. The support of the Fourier basis in the t-domain is infinite, while it contains exactly one frequency, so that the support in the frequency domain is just one point (a Dirac δ distribution). The other extreme is to have a “basis” of Dirac δ distributions in the t-domain,

$$f(t) = \int f(\tau)\delta(t-\tau)d\tau,$$

but their Fourier transforms are complex exponentials implying an infinite support in the frequency domain. This just inverts the Fourier situation.

To represent the function, we need the Dirac δ distribution and also all its (real) *translates* because $f(t)$ can have different values for every real t. Digital signals however consist of samples and the function will only be defined for discrete values of t. Suppose these discrete values are the integers $\mathbb{Z}$. In that case we can use discrete δ functions, i.e., the Kronecker delta with values δ_k being 1 for $k = 0$ and zero for all other integer values of k, and of course all its integer translates.

$$f_j = \sum_{k \in \mathbb{Z}} f_k \delta_{k-j}.$$

Obviously what we want as a wavelet basis is something in between the two extreme cases of the Fourier basis or the δ basis. A wavelet basis should preferably be local in the t as well as in the frequency domain. Because of the Heisenberg uncertainty principle, we have to give up all hope to have perfect localization in both domains, but we may hope for basis functions that have a finite σ and a finite $\hat{\sigma}$.

Let us embed sequences (functions of t defined in the integer values of t) into the set of functions depending on a continuous variable t. We may identify a sequence with a piecewise constant function, say $f(t) = f_k$, $t \in [k, k+1)$, $k \in \mathbb{Z}$. The Kronecker delta, then corresponds to a block function $\varphi(t)$ that is equal to 1 in the interval $[0, 1)$ and zero everywhere else. These block functions played an essential role in the work of Haar (1885-1933) [31]. Supposing $f(t)$ is only defined on a finite interval, we can have a very rough approximation by replacing it by its average value. That is φ dilated and translated to cover exactly the whole interval, multiplied by a constant (the average value). A better approximation can be obtained by subdividing the interval in two equal parts and approximating the function by its average in each of the subintervals. So we write it as a linear combination of two block functions, stretched and translated to cover each half of the interval. This finer approximation can be written as the first rough (constant) approximation plus some correction. Given the fact that the heights of the blocks were averages, it should be clear that the correction will be a constant multiple of a function $\psi(t)$ which takes the value 1 for $0 \leq t < 1/2$ and the value -1 for $1/2 \leq t < 1$, which has to be stretched to the whole interval. See Figure 1. This correction is just adding detail information to the first coarse approximation. This idea can be repeated recursively, in practical situations up to the desired level of detail, but in an abstract setting it can be repeated infinitely many times, i.e., to an infinite level of detail which brings us back to arbitrary real functions.

So Haar was the one who defined in 1909 the simplest example of a wavelet. The function ψ that we defined above is the "mother" wavelet function (the Haar wavelet) that generates all the basis functions $\psi_n(t)$ by translation and dilation. The block function φ is called the scaling function or "father" function. Translations of ψ are needed to cover the whole t-domain, while the successive dilations correspond to the successive levels of detail (resolution levels) corresponding to translations in the frequency domain. At the same time, it is obvious how subdivision can give us the desired basis functions by construction.

After Haar, we have to make a leap to the second half of the twentieth century. Several people realized the drawback of the classical Fourier basis (their infinite support). One possibility to avoid this was to introduce a window function. This gives rise to the short time Fourier transform. The basic idea is that the basis functions that are used are still the complex exponentials, but now multiplied by a window function that has a compact support, that is a positive function that is zero (or sufficiently small) outside an finite interval. This idea is the semen of what has now become known as Gabor analysis [23, 24]. It was however in 1984 that Morlet who came up with the idea that the necessary basis functions can be obtained, not by moving the window, but taking one windowed oscillator and translating and dilating it [29]. The Morlet wavelets were born. See Figure 2. Morlet is also the one who coined the term wavelet as a translation

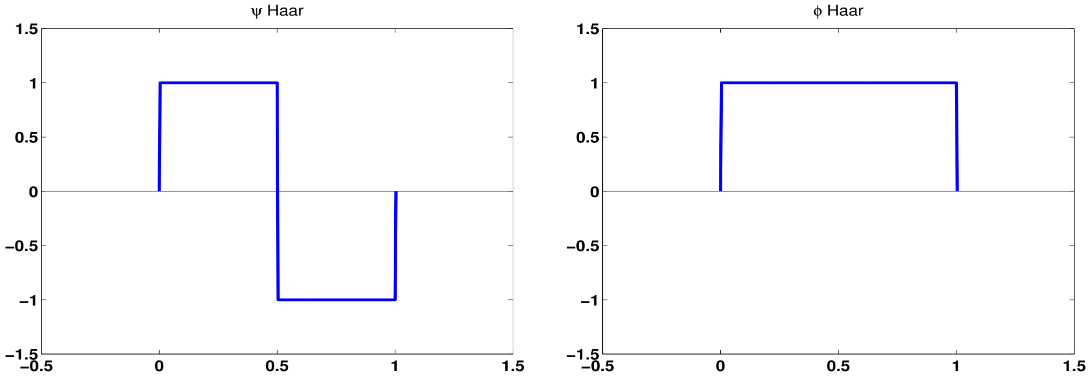


Figure 1: The Haar wavelet. Left: wavelet (mother) function; Right: scaling (father) function

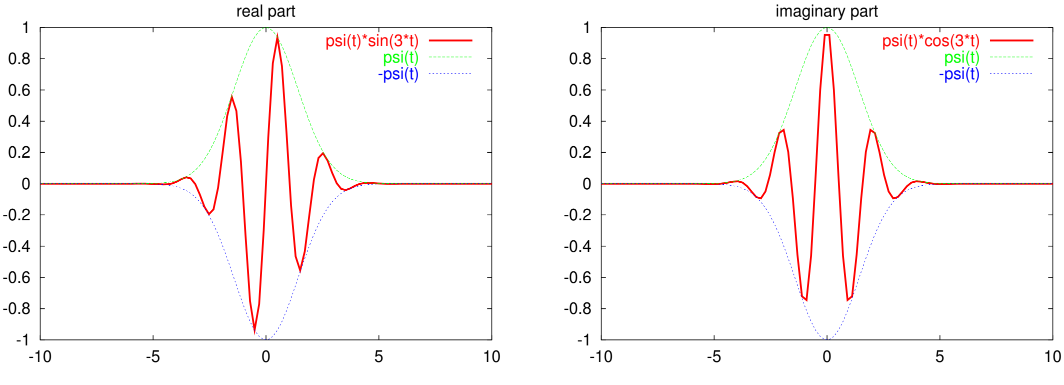


Figure 2: A Morlet wavelet, real and imaginary part. The windowing function is Gaussian that squizes the support of the sine and cosine practically to a finite interval. The oscillations are negligible outside a finite interval.

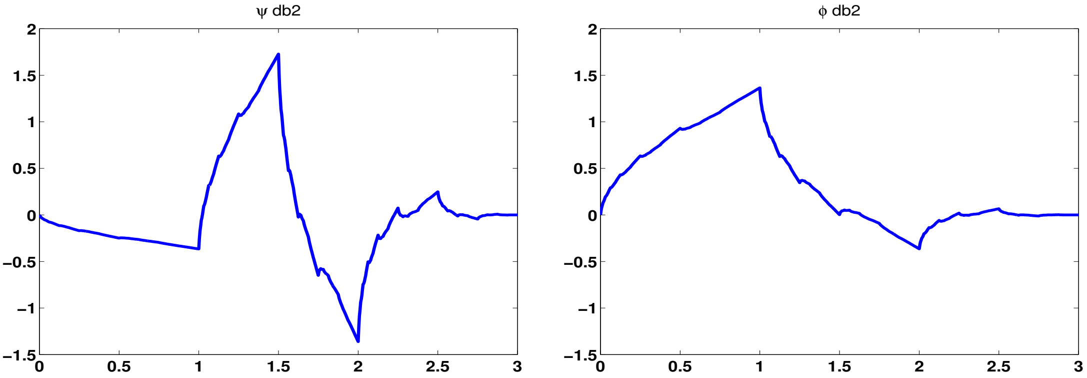


Figure 3: A Daubechies wavelet. Left: wavelet (mother) function; Right: scaling (father) function

of the French "ondelette" which means "small wave". Morlet, together with Grossman, had arrived at the concept of the wavelet transform and its inverse.

In the 1980's everything was set up for wavelets to burst the dike. Mallat and Meyer [50, 51, 53] designed orthogonal wavelets and came up with the mathematical concept of a *multiresolution analysis* (MRA) in 1986. In a MRA both the scaling function φ and the wavelet function ψ play an essential role. So far the wavelets did not have a compact support, i.e., they were small outside a finite interval, but not really zero. Daubechies discovered wavelets that were actually compactly supported, and that were orthogonal, and she obtained regularity conditions [11, 12]. These functions were far from obvious. They turn out to be continuous, but nowhere differentiable. See Figure 3. They are fractals. Nevertheless the functions shown in this example satisfy some regularity condition that seems hard to believe at first sight: using this functions and all its translates, it is possible to represent exactly any polynomial of degree 1 (a straight line!).

3 MULTIRESOLUTION ANALYSIS AND RECURSIVE FILTER BANKS

By setting up the concept of a multiresolution analysis (MRA), Morlet really gave an enormous impetus to the development of wavelets, since this mathematical construction is the natural breeding ground in which wavelets as they were originally conceived could flourish. Let us try to introduce this MRA idea, avoiding the technical mathematics as much as possible.

Consider Figure 4. In this figure you see in the top graph a short piece of a speech signal. In the bottom graph you see five different signals that are plotted at an appropriate scale and shifted vertically to be able to distinguish them. The speech signal on top is the sum of the five bottom signals. Let us call the signal component with the slowest oscillations, (lowest frequency content) an approximation at (resolution) level 0, and the most nervous component at the bottom (highest frequency content) is then at level 4. The whole speech signal is also at level 4. It contains all the frequencies from the lowest to the highest. The subdivision of the bandwidth of the speech signal into 5 parts can be obtained by passing the signal through a set of filters (a *filter bank*) where each filter is a bandpass filter killing all the components of the signal whose frequencies do not fall within a certain frequency subband.

Now suppose we realize this filter bank recursively in a dyadic fashion. This means the

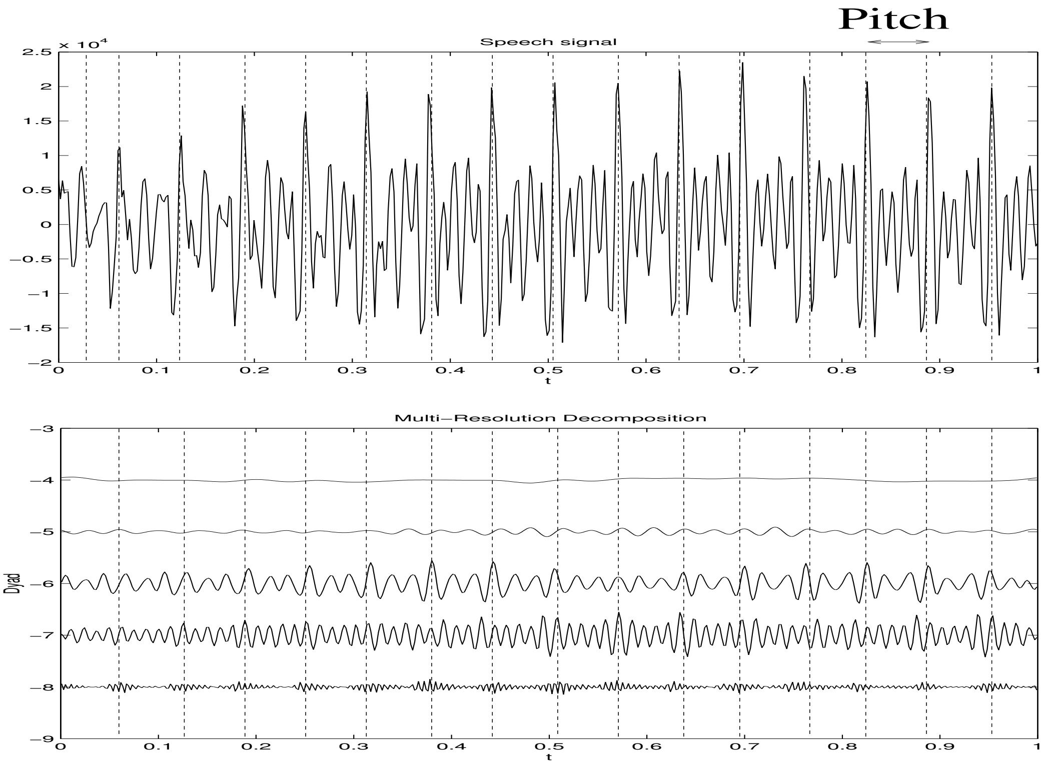


Figure 4: On top: a short slice of a speech signal and below its MRA decomposition. The top signal is the sum of the five signals at the bottom.

following. Given a signal $x_n(t)$ at level n say.

- Split the signal in two parts by passing it through a low-pass filter and a high-pass filter. The low-pass selects the part of the signal with the lower half of the frequencies and the high-pass selects the component that contains the upper half of the frequencies.
- Do the same splitting operation on the low frequency component, and repeat this until one arrives at level 0.

Now suppose that the space of signals at level n is denoted by V_n. The first splitting step decomposes the signal in a coarse approximation (low frequency component) which is at a lower level in a space V_{n-1}, and a detail component (high frequency part), which gives in fact the approximation error, and which belongs to a detail space, which we denote by W_{n-1}. Note that $V_{n-1} \subset V_n$ and $V_n = V_{n-1} + W_{n-1}$. Thus we have written x_n as $x_n = x_{n-1} + w_{n-1}$ with $x_{n-1} \in V_{n-1}$ and $w_{n-1} \in W_{n-1}$. Then x_{n-1} is in turn written as $x_{n-1} = x_{n-2} + w_{n-2}$ etc. This leads to the decomposition

$$x_n(t) = \underbrace{\underbrace{\underbrace{x_0(t)}_{x_0 \in V_0} + \underbrace{w_0(t)}_{\in W_0}}_{x_1 \in V_1} + \underbrace{w_1(t)}_{\in W_1}}_{x_2 \in V_2} + \ldots + w_{n-1}(t) \in V_n.$$

In mathematical terms, this corresponds to a change of basis in the representation of the signal. Suppose we have a set of basis functions at level n, namely $\varphi_{n,k}$. The signal x_n can be represented as a linear combination of these basis functions:

$$x_n(t) = \sum_k c_{nk}\varphi_{nk}(t)$$

and suppose as in the Haar case, that these φ_{nk} are all integer translates of φ_{n0}. Furthermore, it is clear that by scaling the t-axis by a factor of 2, we reduce the frequencies by a factor of 2. This implies that φ_{n0} is a dilated version of the generating father function $\varphi \in V_0$. Thus V_n will be spanned by the scaling functions φ_{nk} at level n

$$\varphi_{nk}(t) = \phi(2^n t - k).$$

Likewise we shall assume that the spaces W_n are spanned by wavelet functions $\psi_{nk}(t) = \psi(2^n t - k)$. The first step rewrites $x_n(t) = \sum_k c_{nk}\varphi_{nk}(t)$ as

$$x_n(t) = \sum_k c_{n-1,k}\varphi_{n-1,k}(t) + \sum_k d_{n-1,k}\psi_{n-1,k}(t),$$

and the recursive filter bank operation rewrites the signal as

$$x_n(t) = \underbrace{\sum_k c_{0k}\varphi_{0k}(t)}_{x_0(t)} + \sum_{l=0}^{n-1} \underbrace{\sum_k d_{lk}\psi_{lk}(t)}_{w_l(t)}\,.$$

Thus we have replaced the original basis φ_{nk} for V_n by a new set of basis functions

$$[\cdots \varphi_{0,k} \cdots | \cdots \psi_{0,k} \cdots | \cdots \psi_{1,k} \cdots | \cdots | \cdots \psi_{n-1,k} \cdots | \cdots \psi_{n,k} \cdots]$$

and the set of coefficients c_{nk} is transformed into the coefficients

$$[\cdots c_{0,k} \cdots | \cdots d_{0,k} \cdots | \cdots d_{1,k} \cdots | \cdots | \cdots d_{n-1,k} \cdots | \cdots d_{n,k} \cdots] \tag{1}$$

This transformation is called the (discrete) wavelet transform. In digital signal processing, the signals are not functions of a continuously varying t, but the signals are given as a sequence of samples. However, if n is large enough, the support of the basis functions φ_{nk} will be so small, that they can be considered to be almost like Dirac delta functions, which justifies to approximate the c_{nk} by the samples of the signal.

A digital filter corresponds to a convolution operation applied to the signal. Ideal low-pass and high-pass filters are not practical because they can only be realized with digital filters that are infinitely long. Thus all kinds of approximations are used in practice. For different choices of these two filters (low-pass-like and high-pass-like), one shall end up with different MRA's since these filters define the scaling function φ and the wavelet function ψ.

Now, if the operation does indeed correspond to a basis transformation, then we should end up with as many basis functions as we started out with. This can indeed be achieved as follows from the Shannon *sampling theorem* ([60], also attributed to Whittaker [80], Nyquist [55] and Kotelnikov [38]).

If a signal is given by a number of its samples, then it is clearly impossible to fill in the missing function values in between, unless we know something more about the signal. For example

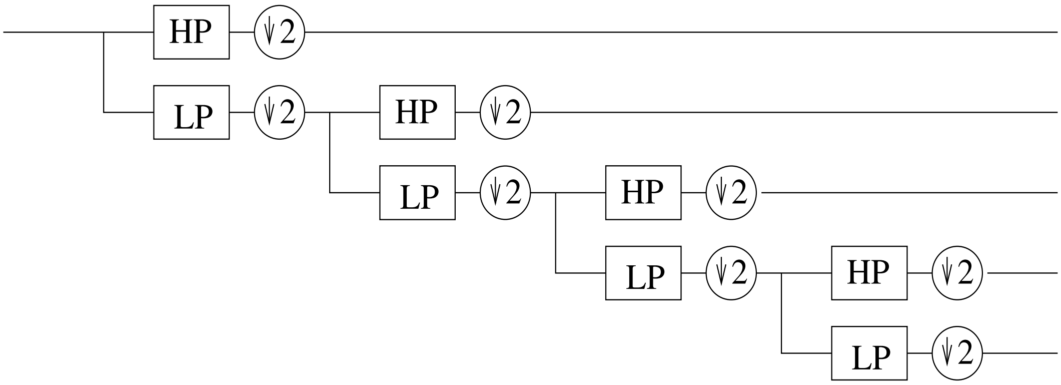


Figure 5: A recursive filter bank. HP represents the high-pass and LP the low-pass filter. The ↓2 operation denotes subsampling.

if the signal is known to be of the form $A\cos(\omega t + \phi)$, with only one given frequency ω, then we need only two function values to fix the amplitude A and the phase ϕ. More generally the sampling theorem says that if we know that the signal is band limited, say its Fourier transform vanishes outside the interval $|\omega| < \omega_m$, then it is sufficient to sample the signal at a frequency $\omega_s > 2\omega_m$.

This sampling theorem and all possible generalizations thereof are still the subject of intensive research. The problem is of great practical importance of course: how much information does one need, hence how much can be thrown away, to be able to recover a function that belongs to a certain class of functions. This is not a topic we shall elaborate in this paper.

For our situation of the recursive filter bank, this means that in the spaces V_{n-1} and W_{n-1} we need only half the information of what was needed in V_n since the bandwidths are halved. This means in practice that we can subsample the sample values after they are filtered. We need only one out of two samples for each of the two filtered signals.

So the recursive filter bank can be represented as in Figure 5 where the circled operations represent the downsampling.

One may wonder why we subdivide only the low-pass part in each step. This is because it is what corresponds most to the sensitivity of our visual and auditive senses. Mathematically, however, there is no reason to restrict ourselves to this choice and one could do otherwise. And this is in fact used in wavelet packets techniques [7] where one does a search in the binary tree that splits every subband in two, to find the representation that is best for the problem at hand.

An example of a wavelet transform is given in Figure 6.

To conclude, we have with our fuzzy explanation of what a wavelet is, reached the properties we desired at the outset and yet we have maintained as much as possible, all the nice properties of the Fourier basis viz.

- The wavelet basis $\psi_{n,k}$ is generated by one mother function ψ by dilates and translates.
- These basis functions have their main "energy" in a limited region of the (t, ω)-plane. The support in the t-domain and in its Fourier domain is essentially compact.
- The transform is fast in the sense that it requires only $O(N)$ operations when a signal of N samples is transformed.

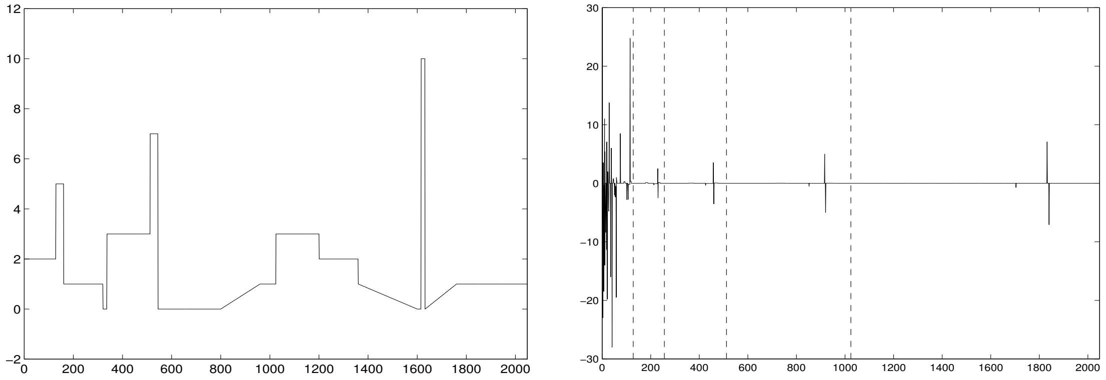


Figure 6: An example of a (discrete) wavelet transform. On the left a skyline signal. On the right its wavelet transform. The vertical dotted lines separate the coefficients as in (1).

There is one important property still missing which is quite important for ease of computation and for stability reasons: can we make the wavelet basis orthogonal? The answer is yes, but that comes at a cost. The Daubechies wavelet that we saw before is a typical example. It is a fractal function, which does not have any symmetry. Moreover, one is limited in imposing regularity conditions. A minimal regularity requirement is for example that the integral of the ψ function is zero. In a classical orthogonal setting, this implies that all (piecewise) constant functions can be represented in the space V_0 and hence also in any V_n. This also justifies the *wave*let character: a wave suggests oscillation. An additional regularity condition would for example infer that also polynomials of the first degree can be represented like for the Daubechies wavelets etc. Note however, that the more conditions that are imposed, the more parameters we need. The filters of the filterbank will be longer, and thus require more computation, but also the support of the basis functions will become larger, so that we loose on the locality, the first commandment of wavelets.

Relaxing the idea of an orthogonal basis, one may think of a biorthogonal basis. This means that we have two MRA's. One is as we have described, and is visualized by the filter bank as in Figure 5 which describes the analysis phase, i.e., it describes how to decompose a signal in its wavelet components. The other is associated in a similar way with the mirror image of this analysis bank on the reconstruction side. It represents the inverse wavelet transform, and describes how the signal can be reconstructed from its wavelet decomposition. The two MRA's are connected through the *perfect reconstruction* condition, which means that the synthesis step should be able to recover the original signal from is wavelet transform.

Biorthogonality here means that the basis functions of an MRA and its dual need not be orthogonal systems on themselves but the basis functions of the first MRA are orthogonal to the basis functions of its dual MRA. This gives more freedom in the design of the wavelets.

Of course the idea can be generalized even more and one may give up independency of the basis functions, which results in redundancy. We get a *frame*, in a sense this is an "overcomplete basis" (which is actually a contradictio in terminis). One should still be careful and require that a small change in the coefficients does not result in an unbounded change in the signal and conversely. This is of course directly related to the numerical stability of the computations.

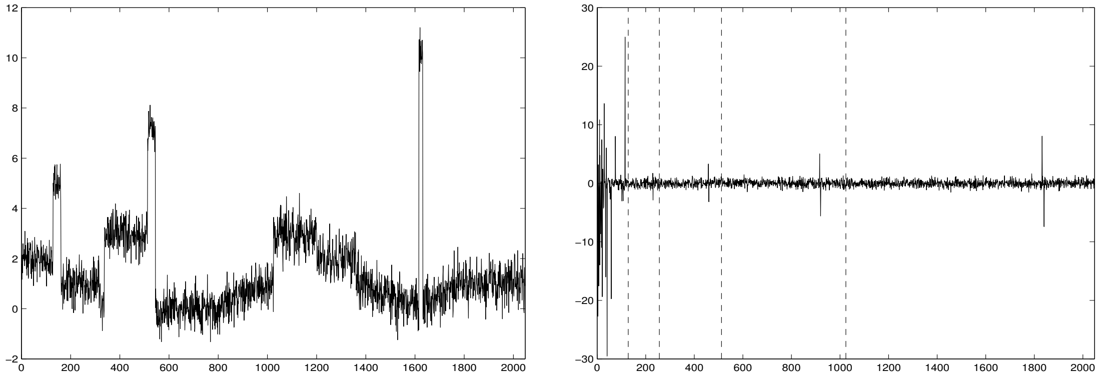


Figure 7: The skyline signal of Figure 6 with white noise (left) and its discrete wavelet transform (right).

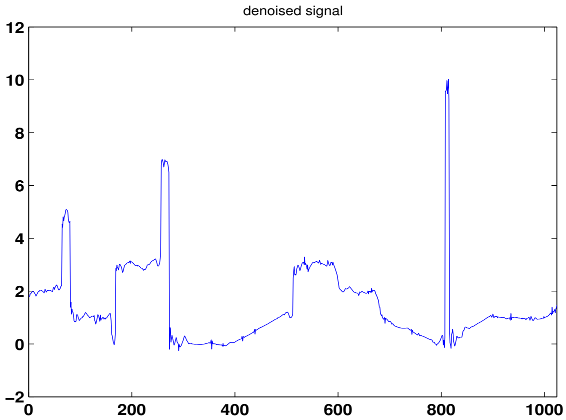


Figure 8: The wavelet transform of Figure 7 is thresholded and its inverse transform gives the denoised signal shown above.

4 AN APPLICATION: NOISE REMOVAL AND COMPRESSION

Suppose we add to the signal of Figure 6 some stationary Gaussian distributed random signal (white noise) with average zero and some standard deviation σ. This results in the signal shown in Figure 7.

If we compute its wavelet transform, then it is seen that the stationary noise (i.e., noise with uniform energy) on the signal results in stationary noise on its wavelet transform. This is not true in general but it is guaranteed by the noise being white and stationary and the orthogonality of the filter bank used. So it is clear that if we remove (replace by zero) all the coefficients in the wavelet transform that are below a certain magnitude, we remove all the disinformation introduced by the noise and possibly some information that is relevant, but because the coefficients are small, the contribution of the latter will only be small (given the stability of the basis). The main problem here is to choose the level of the threshold. There are some known rules of thumb that assume that the standard deviation of the noise is given. However in most practical applications, all we know is the noisy signal, in which case, σ is unknown.

There is a lot of literature on how to compute a threshold that is optimal in some sense [19, 20, 35]. It would bring us too far away from this introduction to go into this topic. Moreover,

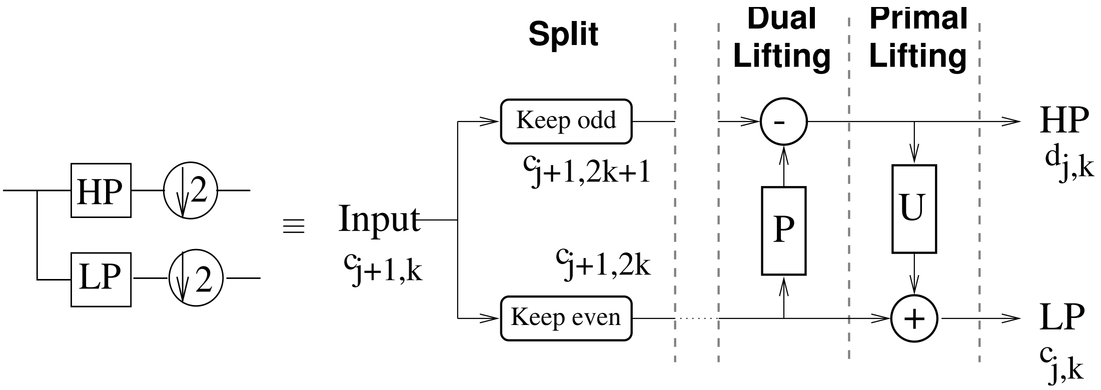


Figure 9: The building block of the filter bank can be decomposed in elementary lifting steps. On the left the classical filter bank realization, of the right, the lifting realization.

the example given here is only the simplest possible situation. There are still unsolved problems to deal with non-additive and non-white noise, and there can be severe stability problems when the sample points are not equidistant [75, 70].

What has been said in the context of denoising, can also be used in the context of compression. Because of the local support of the energy of the wavelet basis functions in the time-frequency plane, and because the wavelet coefficients represent only (small) corrections to previous approximations, the wavelet transform consists of many small coefficients, while the dominant features are covered by few large coefficients. Thus a certain amount of compression can be obtained by keeping only the large coefficients and removing the small ones. Since the bulk of the coefficients is small, this may lead to a considerable reduction in the storage without loosing the essential features of the signal.

5 THE LIFTING SCHEME AND SUBDIVISION

Before we move on to 2D images, we want to introduce the notion of lifting and subdivision. Let us start from the filter bank. As we have explained before, a wavelet transform can be realized by recursively applying some elementary splitting of a signal in its low frequency part and its high frequency part. Another way of describing this operation is the lifting scheme. This scheme is not just another approach to arrive at the same result, but it is also a formulation that can be easily generalized, so that classical wavelets can be seen as a very special case of a much more general setting. The term was invented by W. Sweldens [67, 65, 64, 66, 68]. See also [5, 42] for other related approaches.

By splitting the signal in a low and a high frequency part, we have a coarse approximation of the signal (the low frequency part) and the difference with the true signal (i.e., the high frequency part) is then the approximation error, which will in general be much smaller than the signal. Taking also the subsampling into account, we can describe the step as follows. See Figure 9. We first split the signal into two parts, say E and O (e.g., the even and the odd samples). Then approximate or ‘predict’ the O part (e.g., the odd samples) by information drawn from the E part (we can for example take the average of the two even neighbors). We then only need to keep the E part as a rough approximation and instead of O the (presumably much smaller) prediction error. This is one part of the lifting (the dual lifting step). Leaving

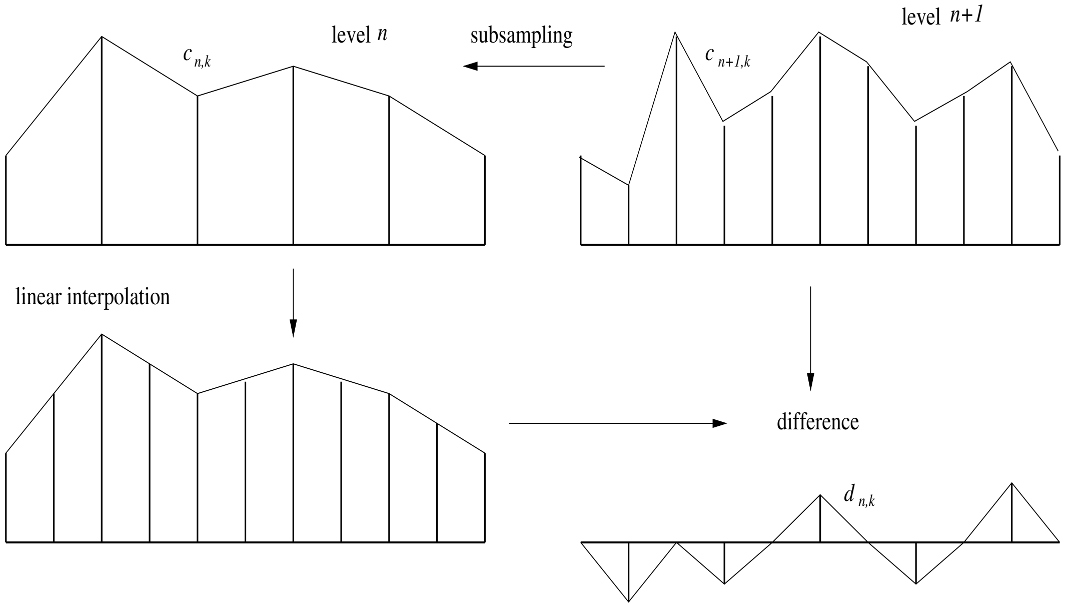


Figure 10: Subdivision by linear interpolation.

the E part unchanged is not a good option though. If we want to keep other properties of wavelets like smoothness conditions, an 'update' step is necessary which will update E with the prediction errors that were just obtained. This is the primal lifting step. In general there may be several of these primal-dual lifting steps to realize the high-pass-low-pass splitting of the filter bank. In fact one may show that it is even possible to decompose every such high-pass-low-pass couple of filters in a sequence of elementary primal-dual lifting steps, a decomposition that is in fact easily computable by a slight generalization of the Euclidean algorithm to compute a greatest common devisor of two Laurent polynomials [13]. Euclid would be surprised to see this application of his algorithm.

This lifting scheme brought so-called *second generation wavelets* into the picture. Sampling points need not be equidistant in principle, wavelets can be designed on finite intervals, etc. It also allows non-linearities. For example it may be most desirable to work with integers to be efficient in storage and computing time. So one may assume that all the samples are integers. The filter operations may however destroy this, and so the result of a prediction or updating filter operation will be rounded to an integer again. This is a non-linear operation and one might think that in this case it is impossible to recuperate the original signal from its decomposed (and rounded) representation. However, as long as the rounding rules are fixed, there is no problem because to invert an operation "add a rounded value" we just have to "subtract (the same) rounded value". Thus even with integer arithmetic, it is possible to have perfect reconstruction with the lifting scheme [2, 69].

The subdivision schemes by Deslauriers and Dubuc immediately fit into this framework [14, 15]. To decompose a signal in different resolution levels, one may for example use linear interpolation as a predictor in the previous lifting scheme. See Figure 10. Suppose we have a coarse approximation in only a few sample points. To find a better approximation, one may predict by linear interpolation the value in the middle between two of the coarse sample points. Only the difference between the true value in that midpoint and the prediction has to be kept. This process can be repeated to add information for computing the values in the points lying in

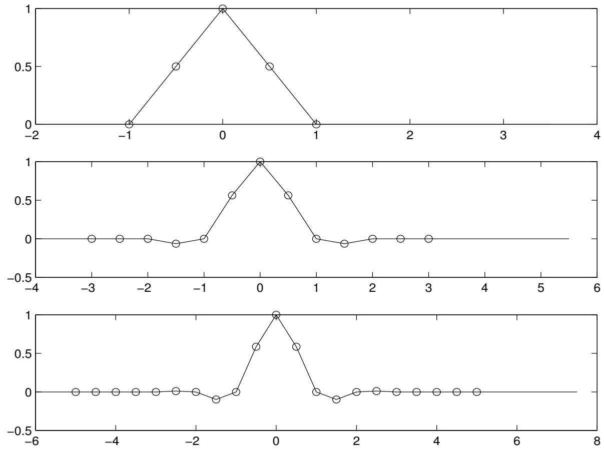


Figure 11: Subdivision on a unit pulse at level 0 results in a function φ. From top to bottom: interpolation of degree 1, 3, and 5.

the middle between the previous sample points. But linear interpolation is of course only the simplest possible example. Higher order interpolation of odd degree is possible, except at the boundary of the interval where some modifications have to be taken care of.

If we number the levels as follows: the coarsest level is level 0, after one subdivision, we get an approximation at level 1, after two subdivisions, at level 2 etc. Denote by V_n the space of all possible functions that can be represented at resolution level n. An approximation at level n can be considered as a “projection” of the signal onto this space V_n, formally: $f_n = P_n f$. The approximation error $f - f_n$ is in a complementary space W_n, and it can also be represented as an operator Q_n acting on f.

If one does this carefully, there will also be a MRA associated to these schemes. Some preliminary φ function can be obtained by applying the subdivision scheme to a coarse grid which starts from all samples equal to zero, except one sample where we assume the value one. The subdivision scheme then results in a function φ. For degree 1 interpolation, this results in a “hat function”. See Figure 11. Of course, this φ has not taken the primal (i.e., updating) step into account. Also here we have primal and dual bases corresponding to analysis and synthesis duality in the original filter bank formulation. This can be placed into an abstract mathematical setting, which we do not want to discuss here. For example the dual bases for the interpolating subdivision will be Dirac delta functions. We shall come back to this idea later in this paper in the context of Powell–Sabin subdivision.

6 IMAGES AND WAVELETS

In the previous sections we have tried to introduce some concepts that we want to use in more general situations: multiresolution, wavelets, lifting scheme (via filter banks), and finally subdivision.

Now let us move to two dimensional signals, i.e., images. Although 1D signals are important, our modern age of multimedia is much more depending on images: Internet, digital photography, digital television, medical imaging, virtual museums, GIS (geographic informa-

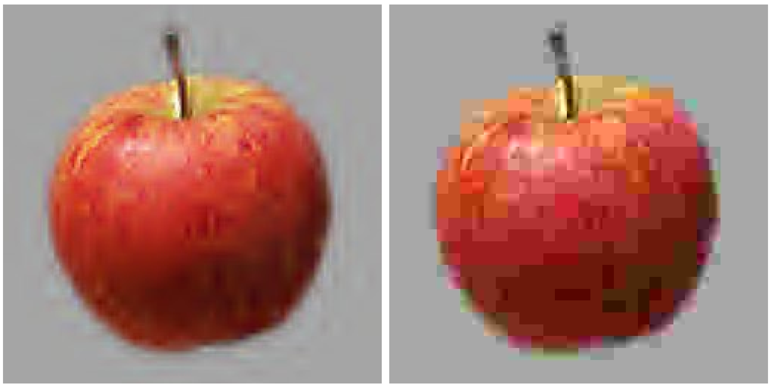


Figure 12: Over-compression: on the left: wavelet compression, on the right JPEG compression. The compression rate is approximtely 105:1.

tion systems), databases of images, etc.

Image processing, in particular compression and noise reduction, are the applications where initially wavelets have been most successful. The (true) story where the FBI database of fingerprints has been compressed using wavelets is well known [1]. The FBI had collected 200 million fingerprints, worth 2,000 terabytes of images. And they accumulate 30,000-50,000 new cards per day! The target was to compress the images to 0.75 bpp (bits per pixel) which corresponds to a compression ratio of 15:1. Classical JPEG ('Joint Photographic Experts Group) compression at this rate starts showing blocking effects. This is because classical JPEG uses a discrete cosine transform on 8×8 blocks, and these blocks become visible when compression is too high. See Figure 12. Wavelets were very successful in this application, which is partly explaining the wavelet hype in the 1990's. JPEG has formulated a new standard (JPEG2000) incorporating wavelet techniques for compression [34].

So how can the wavelet techniques of the 1D case be generalized to the 2D case? By far the simplest way is to use a tensor product approach. An image is considered to be a collection of rows (1D signals) and a 1D wavelet transform is applied to each row. The result is a new image which is now considered to be a collection of columns, and a 1D wavelet transform is applied to each column. To repeat the process we have to decide on what is to be considered the low resolution part in the transformed image, which has to be further analyzed. Most often, the quadrant that corresponds to the low resolution part in both directions is the one that is further split up. So we arrive at a subdivision as depicted in Figure 13. A wavelet transformed fingerprint then looks like in Figure 14. The smallest top left square is a negative for illustration purposes. It gives the coarsest idea of the general picture. The gray scale corresponds to the magnitude of the wavelet coefficients. The lighter the pixels, the larger the coefficients. It is clear that most of the coefficients are rather small, which allows for a high compression rate.

In Figure 15 we illustrate a 2-level transform of the picture of a house. This illustrates some other properties of the wavelet transform. Again for reasons of visibility, we have now shown the negative of the whole transform. In this image, the larger coefficients are darker. What is striking here is that vertical edges are best seen in the LH parts, that is low resolution for the columns and high resolution for the rows. This is to be expected since, vertical edges are high resolution components if the image is scanned along rows. On the other hand, the horizontal edges are best seen in the HL parts for similar reasons.

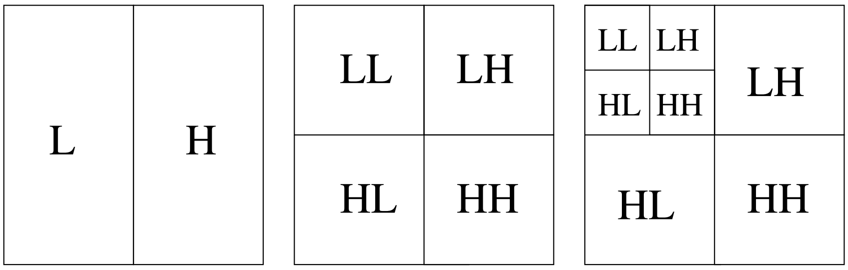


Figure 13: Tensor product wavelet transform. First a 1D wavelet transform is applied to each row. This splits each row in a low resolution part (L) and a high resolution part (H). Then a 1D wavelet transform is applied to each of the resulting columns. The result is splitting the image in four quadrants (LL, LH, HL, HH). The LL part is then treated in the same way etc.

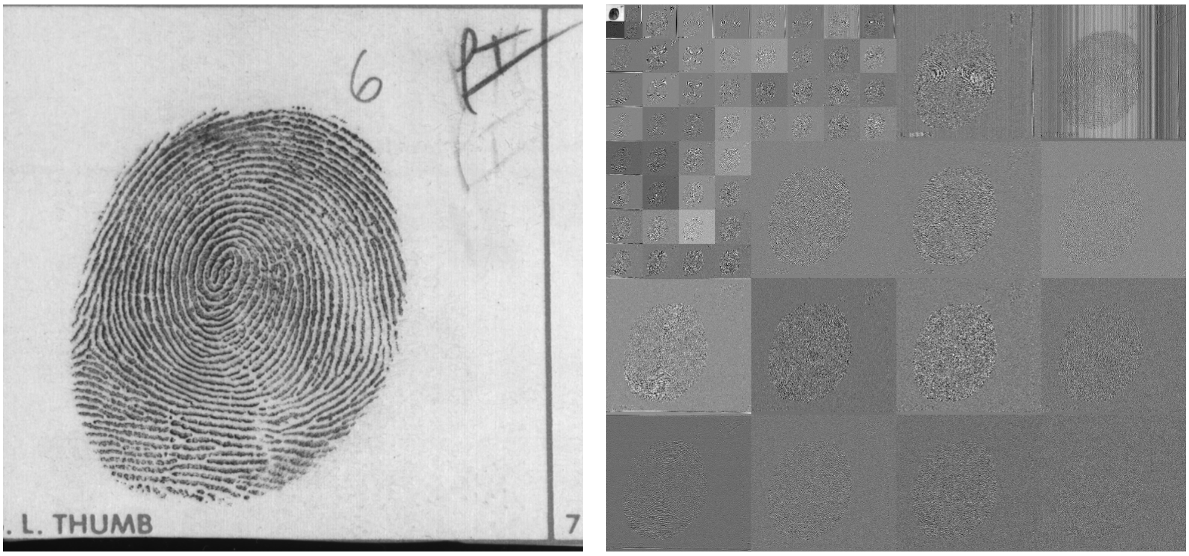


Figure 14: On the left the original image. On the right its wavelet transform. There are 5 resolution levels shown.

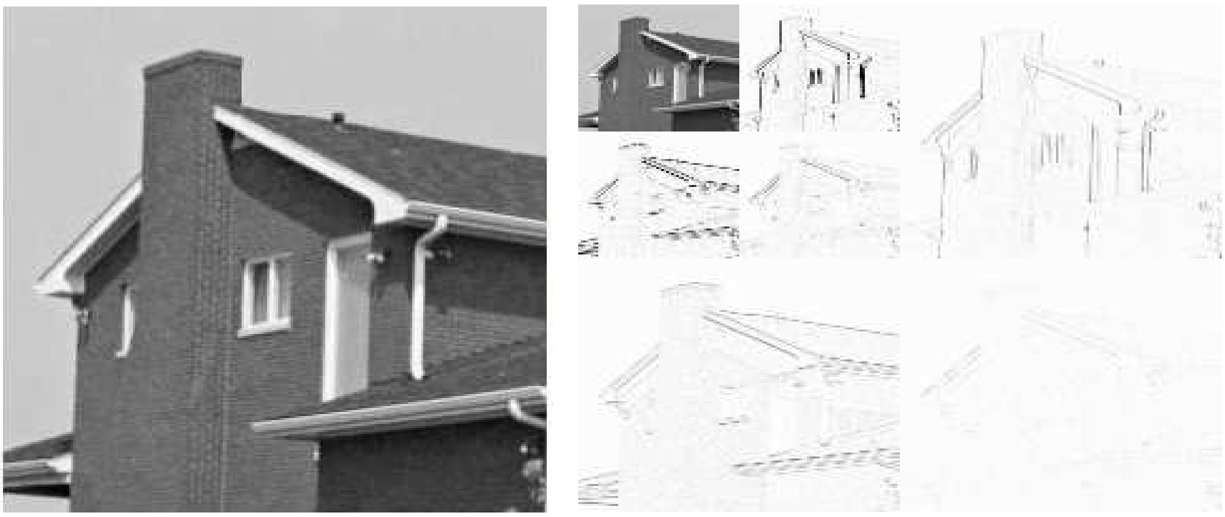


Figure 15: On the left the original image, and on the right two levels of the Haar wavelet transform. Note how horizontal and vertical edges are visible in the different subimages.

There exist many discussions on wavelets in 2D that are not based on a tensor product approach. In that case the wavelets are not separable in the x and y direction. One reason not to use the tensor product approach could be to keep properties of circular symmetry in the image, but in many cases it is just the feature that we illustrated in the image of the house that gave rise to a zoological garden of more or less exotic -lets like curvelets [4], contourlets [18], ridgelets [3], chirplets [52], etc.

The 1D wavelet does an excellent job in detecting discontinuities in a 1D signal. Splines are piecewise polynomials that are fitted smoothly together by keeping continuity of the function value, and possibly in derivatives up to a certain order. With wavelets, it is still possible to detect the discontinuity in the nth derivative of a univariate spline which is not visible to the human eye.

The tensor product wavelet is therefore very good in detecting discontinuities (edges) in horizontal and vertical direction. However the edges are not always aligned along these special directions. See for example the inclined edge of the roof, which is less prominently visible in the wavelet transform of Figure 15. However detecting edges in an images is very important in many applications and not only for plain edge enhancement. Edges form the basis for applications such as segmentation and contrast enhancement. Also for the consulting of an image database, one has to isolate the object that one wants to identify, which has then to be transformed (translated, rotated, morphed) to match with the image in the database on the basis of some pattern recognition software, etc. Thus also here segmentation, and detection of edges play an essential role.

7 HORIZON IMAGES AND NORMAL OFFSETS

Rather than giving a survey of all possible 2D multiresolution techniques, let us consider an example of an image processing technique based on normal offsets. It is certainly not the only technique used for dealing with edges. It is not a wavelet based technique, but it does rely on multiresolution obtained by subdivision. Research on normal offsets is still in progress.

Normal offsets is a non-linear subdivision technique that is especially designed for detecting edges in an image [71]. Let us simplify the situation and assume that we want to obtain a multiresolution decomposition of an horizon image. The class of horizon images can be described as the class of images consisting of large smooth parts (that are of a more or less constant texture) separated by smooth curves, which we shall call *horizons* or *horizon lines*. See Figure 16.

The idea is to represent the image first by a coarse triangularization, keeping only the pixel values at the vertices of the triangles. Then the triangles are subdivided into smaller triangles, introducing an extra number of pixel values etc. The clutch of normal offsets is that they allow to arrange the subdivision of the triangles such that their edges align along the horizons in the image.

Let us illustrate the idea of what a normal offset is in one dimension. See Figure 17. In the figure, the curved line with the jump discontinuity is approximated by a straight line. In the case of a linear interpolating subdivision as we have illustrated before, one would approximate the odd sample in the middle between two even samples by the value predicted by linear interpolant

Figure 16: Example of an horizon image: smooth parts separated by smooth curves. The latter are called horizons.

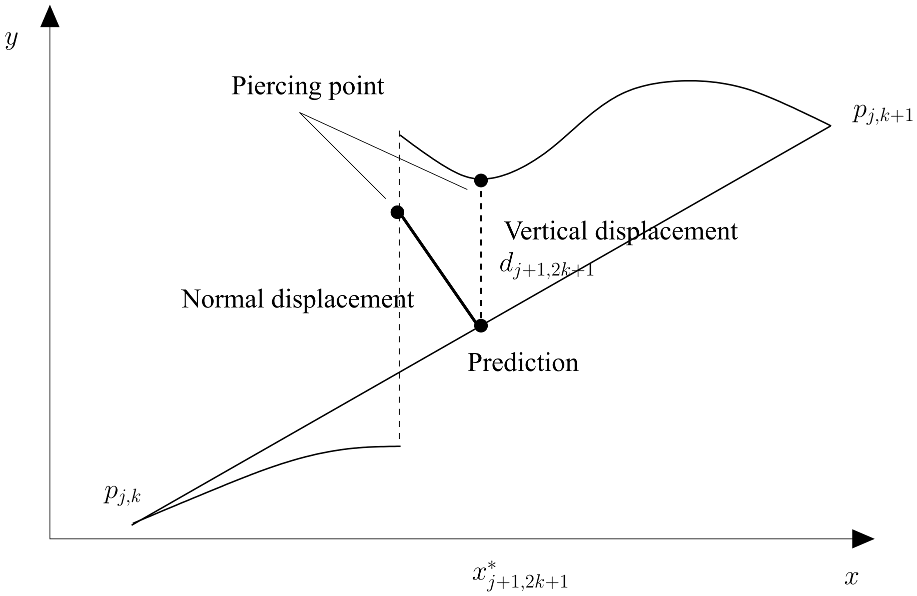


Figure 17: Normal offset for a 1D signal.

that passes through the two neighboring even samples, and we would keep the prediction error, that is the (vertical) distance between the the actual sample value and the predicted value in the odd sample point. We could call this a *vertical offset*. In this way we would never (unless accidentally) find the exact location of the discontinuity. We could only approximate it to some accuracy by going to a high resolution level.

The normal offset however draws a line in the predicted point $p^*_{j+1,2k+1}$ that is normal to the line that represents the linear interpolation. Instead of the vertical offset, we now keep the distance from the predicted point to the *piercing point* (where the normal "pierces" the graph of the function): the *normal offsets*. If the predicted point is near a singularity, we know the actual distance to the singularity exactly. In 2D this is important because in this way we collect information about the geometry of the image in the values of the normal offsets. And we get this information at a coarse level already, we do not need to go to a fine level to find a good approximation of the location of the singularity (i.e., the horizons of the images).

The application of this normal offset idea for images is not so difficult. The (x, y)-region of the image is divided in triangles. Pixel values at each vertex define a surface in 3D which represents the image. At first sight one would expect normal lines to the triangles in 3D. However, this is not a practical idea because the projection of the piercing point on the (x, y)-plane may not be inside the corresponding 2D triangle, and is therefore not suited to subdivide that triangle.

Instead we apply the 1D offset idea to the edges of the triangles. Consider the plane vertical to the (x, y)-plane that contains the edge of a triangle. Consider the two vertices of the triangle as being even sample points, and we want to predict the value in the middle of the edge by linear interpolation, then, in that vertical plane, we get exactly the same situation as depicted in the previous graph 17.

Now we have for each triangle the values of the vertices and the values in the midpoints of the edges. This gives information about the location of the horizon line intersecting the triangle (if there is one). We want to subdivide each triangle into 4 smaller subtriangles. As you can see in Figure 18 (where it was supposed that there is indeed an intersection with an horizon), there are several ways to do this. Now assume that every triangle is cut by at most one of the horizon lines in the image. If there is an intersection then one of the corners of the triangle is on one side of the horizon and the two others are on the other side as seen in the Figure 18. Because we want the edges of the triangles to align along the horizon, we choose the subdivision that makes the least possible cuts with the horizon line. If there is more than 1 candidate as in the first two cases of Figure 18 or if there is no intersection with an horizon, then we choose the most "uniform" subdivision connecting the three midpoints.

In Figure 19 you may appreciate the difference between a "uniform" subdivision where in each step the most regular subdivision is used (left) and the one that is proposed above (middle). Clearly the latter gives a better approximation of the horizon in this image.

But adaptivity can go much further. Assuming that the smooth regions on both sides of a horizon are indeed smooth and hence do not need a high resolution approximation, it is a natural choice to subdivide only the triangles that are intersected by a horizon. Triangles that are totally in one homogeneous region do not need splitting. This is illustrated in the rightmost illustration of Figure 19. The result depends highly on the initial triangulation one starts from.

The results for 7 subdivisions of a gray scale version of the horizon image 16 is given in

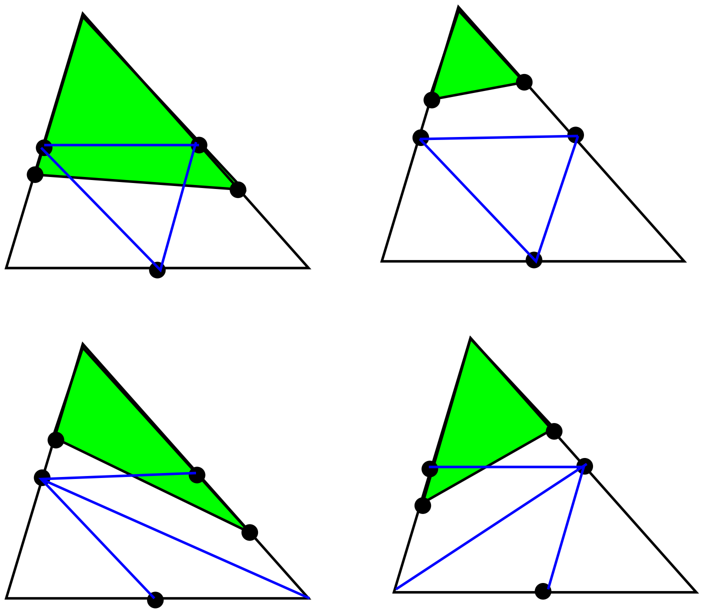


Figure 18: Normal offset split. The green shaded part belongs to one smooth area in the picture, the other area of the triangle belongs to another part. The blue lines indicate which subdivision is most appropriate.

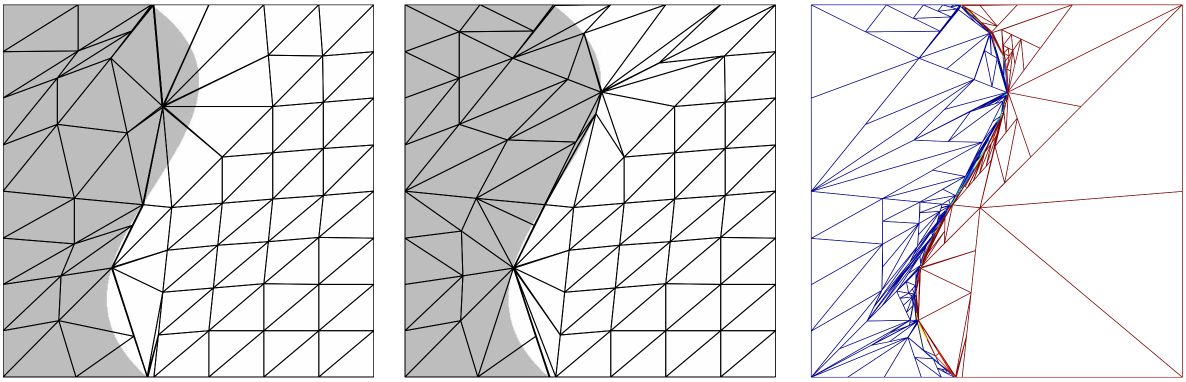


Figure 19: Normal offset split. On the left the uniform splitting, and in the middle the adaptive effect of triangular subdivision aligning along a horizon. On the right only the triangles that intersect an horizon are further subdivided.

Figure 20.

You may notice (although it is not quite visible at the size of the images that are included) that some of the triangles are not aligned with the horizon lines. This is because the algorithm is still very naive and it is assumed that a triangle has either no intersection or has one vertex on one side of the horizon and the two other vertices at the other side. If, as in this picture two horizon lines are close together, two horizons may cut a coarse triangle and then the 3 vertices will have similar values from which the algorithm concludes that the whole triangle is in one homogeneous region and it is not subdivided anymore, which is a wrong decision in this case. This is illustrated in Figure 21 which shows an enlarged detail of the last image. You can clearly see the artifacts that are a consequence of coarse triangles that are not properly subdivided.

That the images need not be images of landscapes is illustrated on a classical pepper image in Figure 22. It illustrates how a non-optimized subdivision gives a compression ratio of 12:1. This compression is obtained just on the basis of the number of coefficients to be kept in a normal offset representation. The encoding may give an even better compression. Again it is illustrated that when horizon lines come close together, some artifacts appear. The detail image shows some ectoplasm being exchanged between two peppers as a consequence of non-aligned triangles.

Normal offsets are based on subdivision and they implement some ideas of multiresolution, but as such, they are certainly not corresponding to wavelets. It is a nonlinear transformation and smoothness or regularity are not guaranteed. That would require some extra conditions but it is hoped that normal offset techniques with such properties can be designed in the near future.

By construction, the normal offset approach is especially suited to treat images in the horizon class. Wavelets do much better in characterizing texture. A possible track of research could be to combine the two as suggested in Figure 23.

8 FROM IMAGES TO SURFACES AND 3D OBJECTS

To give a survey of image processing techniques would be too ambitious an objective for this paper. Most of them are dealing one way or another with multiresolution. So we have confined ourselves to introducing the tensor product wavelet transforms and, because the edges in an image are very important in many applications, we described as a case study an elementary introduction to normal offsets, a nonlinear subdivision technique that is still developing.

To make the step from 2D images to surfaces in 3D is not so difficult. In the beginning of the previous section we described already an image as a surface where the pixel values are set out in the z-direction. Figure 24 shows an unconventional way of viewing in perspective the surface representing the Lena image. Of course 3D objects can not always be described as a surface given in the form $z = f(x, y)$. In some cases, it may be better to use a parametric representation. The three coordinates are functions of two parameters, say u and v. Thus $(x(u, v), y(u, v), z(u, v))$ is a representation of a surface. In this way it is for example possible to describe a sphere, although spherical coordinates would be much better of course.

Color images also have three components, for example the three color components RGB (Red, Green, Blue), or HSV (Hue, Saturation, value), YUV (luminance and two chromatic components), etc. So in a sense images are a lot like 3D objects or surfaces. The problems to be

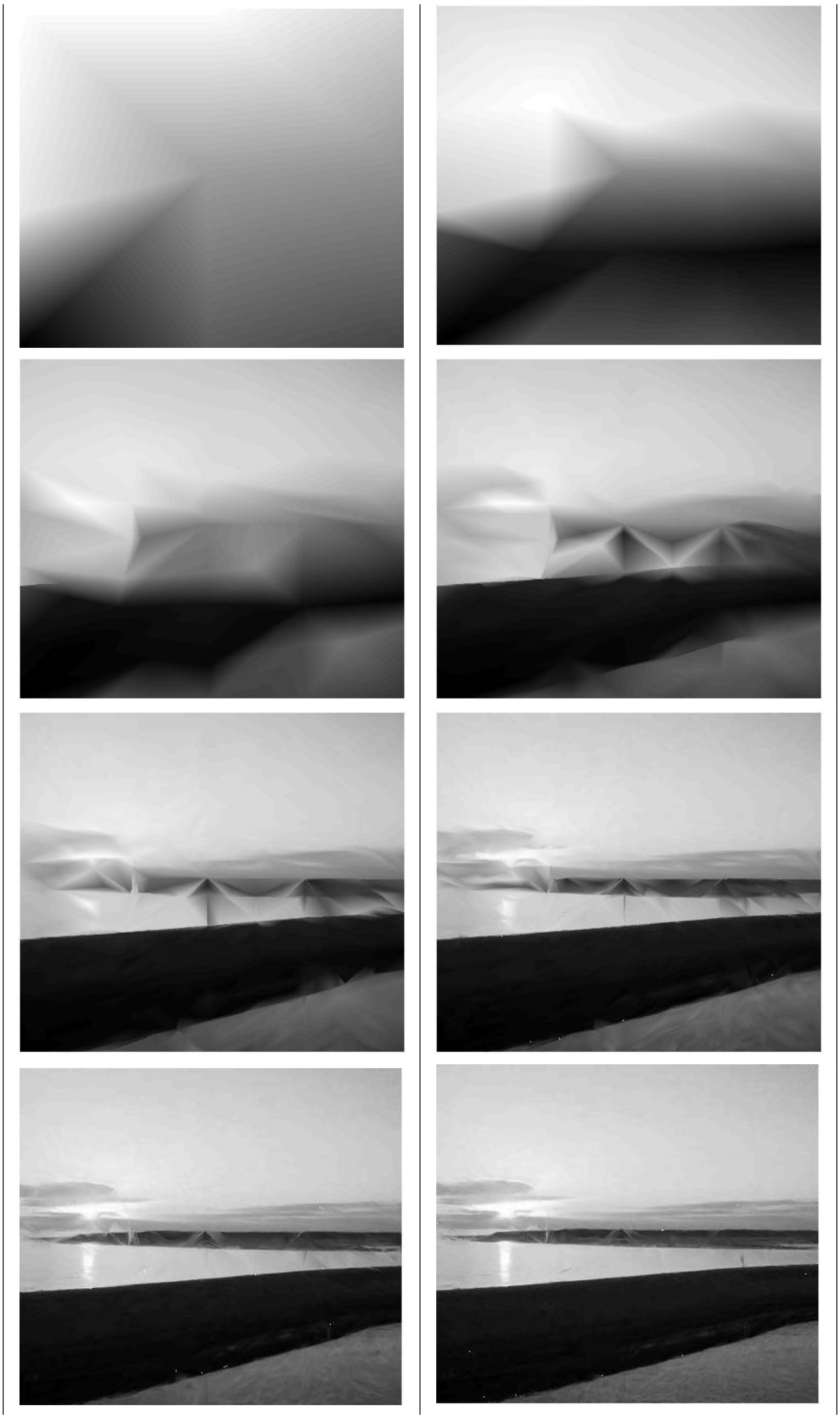


Figure 20: Successive subdivisions of a gray scale version of the image 16.

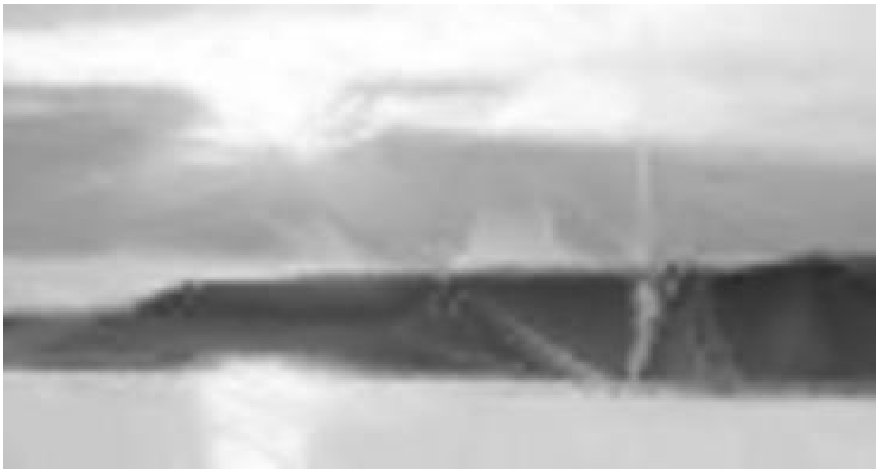


Figure 21: Enlarged detail of the last picture in 20. It illustrates that some of the triangles are not properly aligned, which results in artifacts.

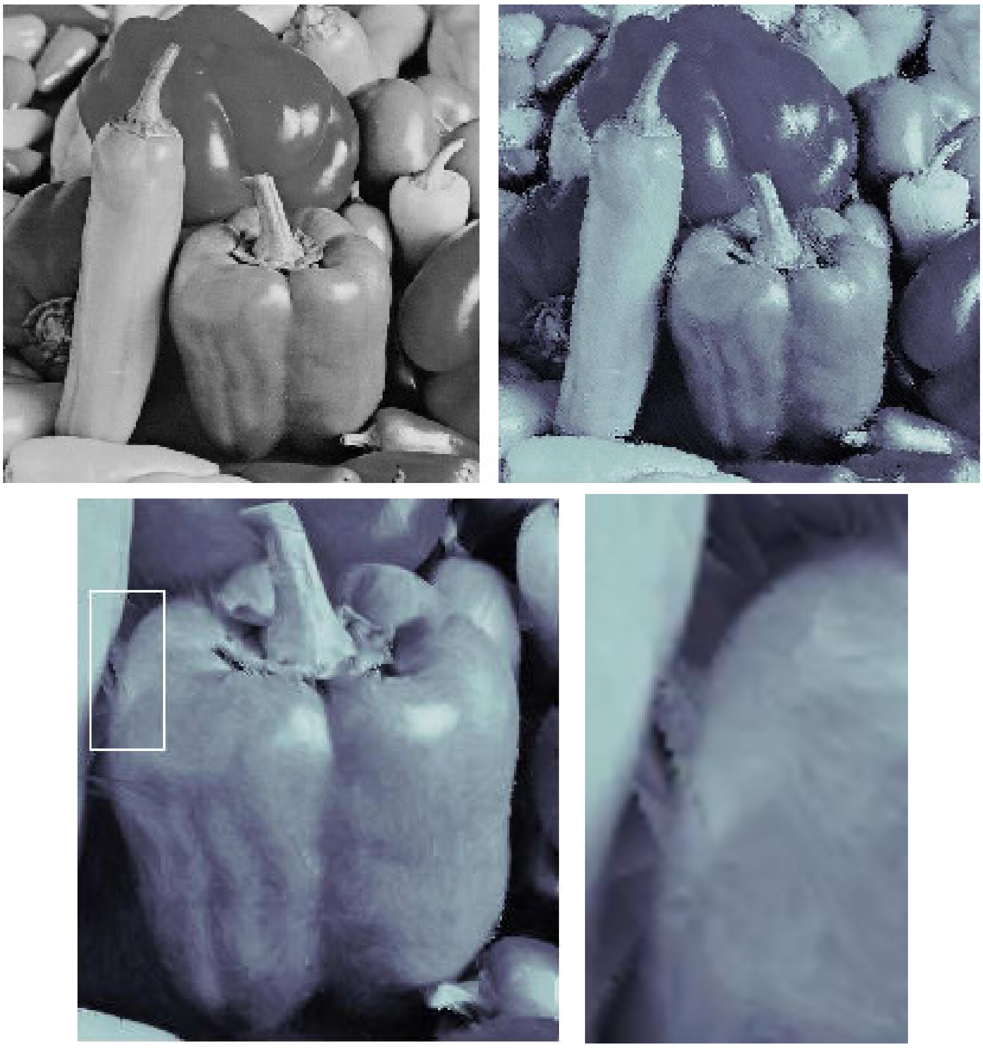


Figure 22: Top left original image. Top right the same image compressed using normal offsets. The compression rate is 12:1. Bottom line gives details of the compressed image showing some artifacts.

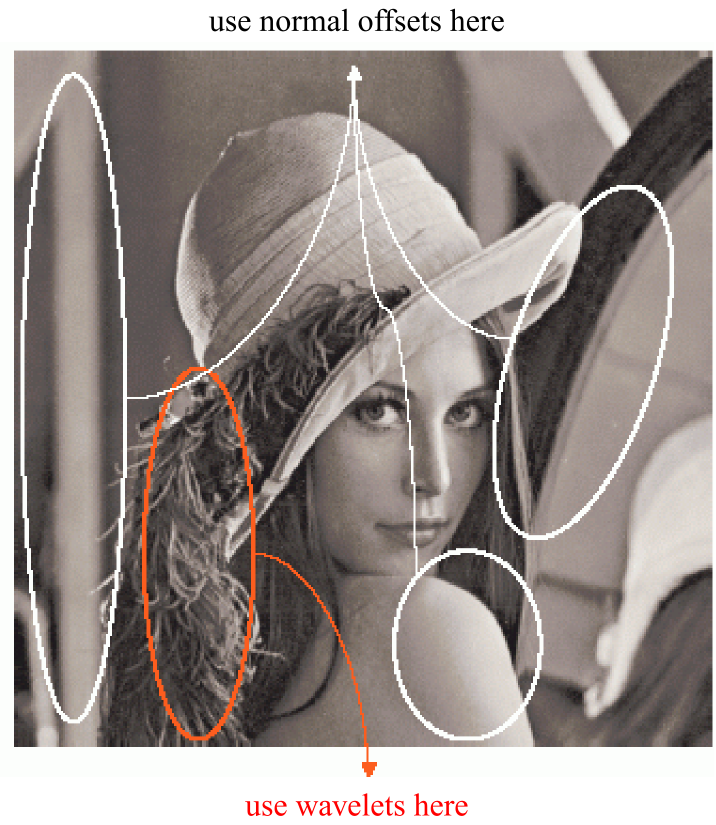


Figure 23: Wavelets are better to represent texture, normal offsets are better in representing edges of an image.

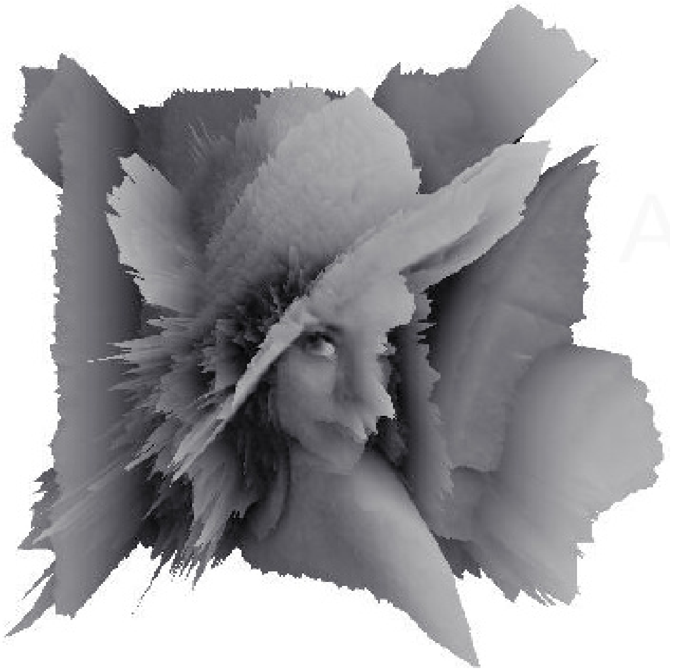


Figure 24: A view in perspective of the 3D surface representing the Lena image.

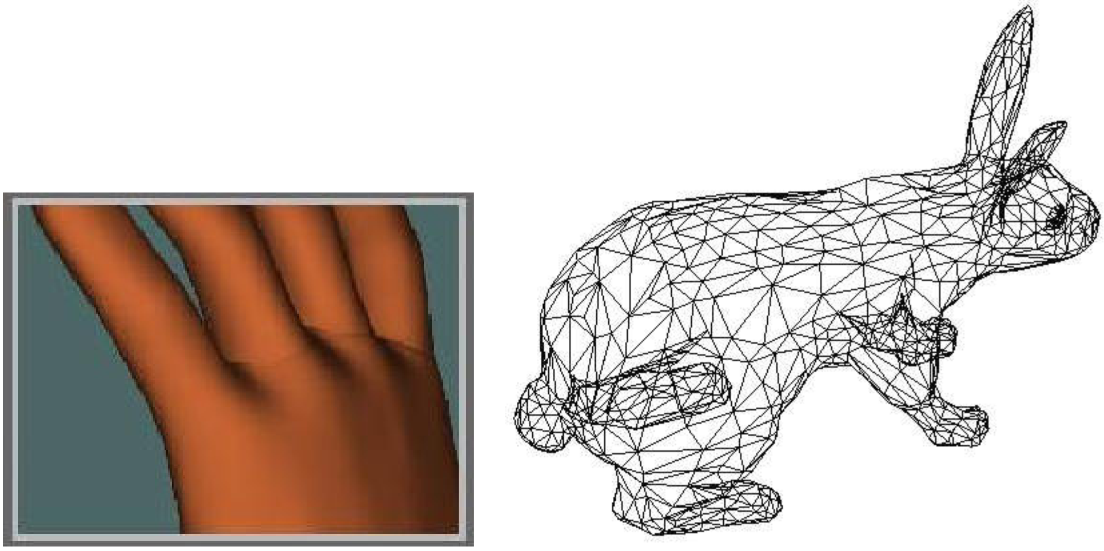


Figure 25: On the left: It is difficult to avoid seams in mathematical description of different patches. Note the seams where the fingers join the palm of the hand. On the right: a polygonal model of a bunny.

solved for images are usually quite different from the problems for surfaces that are encountered in, e.g., CAGD (computer aided geometric design).

The parametric representation is one possible *mathematical functional description* of a surface. In CAGD where smoothness is an important issue, often splines or NURBS (nonuniform, rational B-spline) are used [22]. By this description, surfaces are homeomorphic to pieces of the plane. Complex surfaces will have to be cut up and each piece will have its own description. For example a hand can be described by several pieces: the palm of the hand and the fingers have a separate description. See Figure 25 (left). An extra difficulty is that these patches need to be stitched together preferably keeping some continuity in values and in the tangent plane, which is often quite difficult to obtain. Seams will be visible where patches fit together. Moreover, most often we do not get an easy decomposition into the subsequent resolution levels.

Another way of representing surfaces is by *polyhedral wire frames*. The geometry of the object is often obtained by scanning. The object is then described by a *point cloud*. Certain algorithms have to be applied to find some triangulation in that point cloud, where it is often difficult to maintain the appropriate convexity [25, 33, 79]. See Figure 25 (right). There is no problem of stitching patches together in this case. It works for arbitrary topologies and by removing some of the points one may obtain a representation of the image at a lower resolution. For complex surfaces, one needs a lot of triangles though, which are not easy to manipulate, and it is not simple to maintain smoothness over larger regions.

A *procedural description* based on *subdivision* is a rather flexible way of describing a surface. Such a subdivision is much like the description we gave before. One starts with a coarse control polyhedron. Each of its faces is a control triangle that "controls" the shape of a spline surface in its immediate neighborhood. By successive refinement, the control triangles are reduced in size and their range of influence is reduced too, giving a finer, more localized control over the detail features of the surface. See Figure 26. These control triangles will give the local characterization of the shape of a spline function, which is by construction continuous. So we get a mathematical description (via the spline), without having the stitching problem of

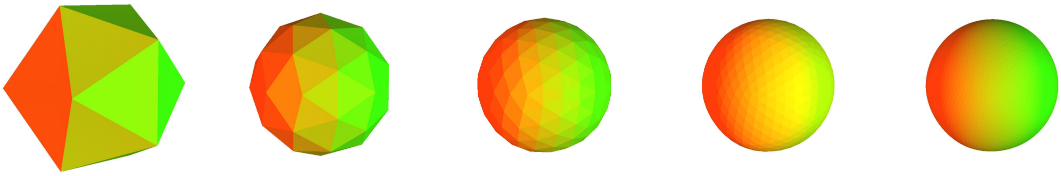


Figure 26: The control triangles in the control polyhedron are tangent to the surface. The refinement procedure will give a representation of the surface at different resolution levels.

different patches, and we keep the possibility of multiresolution. Moreover we have algorithms of polynomial complexity that work on arbitrary topologies. So we have tried to keep the best from all possible worlds.

There is a multitude of subdivision schemes [41] (Loop), [6] (Catmull-Clark), [21] (Butterfly), and there are several ways to classify them.

- Stationary or non stationary. Each element can be subdivided in the same way or not
- Linear or nonlinear. The function values at the finer scale can be a linear combination of function values at a coarser scale or not.
- The mesh can be of several types: triangular, quadrilateral,…
- The subdivision rule can be such that the function value at the finer level is interpolating the exact function values or it can be only an approximation.

In the next section we shall go into some detail concerning Powell–Sabin spline subdivision.

9 POWELL-SABIN SPLINE SUBDIVISION

Like normal offsets for horizon images, Powell–Sabin splines to represent surfaces is only one possible example of a vast literature on this subject [40, 54]. So consider it as a case study which is related to current research.

We start from the definition of a Powell–Sabin (PS) refinement, more precisely, a PS 6-split of a triangulation [56].

Given a triangulation Δ which is supposed to be conformal. This means that edges belong as a whole to one or two of the triangles, i.e, there can not be a vertex of one triangle on an edge of another triangle without being a vertex of that other triangle. We then split each triangle into 6 subtriangles giving a finer triangulation Δ^*. This finer triangulation is obtained by taking a point inside every triangle of Δ. They are the *new vertices* while the vertices of Δ are the *old vertices*. The new vertices are connected (using straight lines) to the old vertices and also to all the new vertices in the neighboring triangles of Δ that have an edge in common. For the triangles with a boundary edge, the new vertex is also connected with a point on the boundary edge (e.g., the midpoint). This is illustrated in Figure 27. The finer triangulation that has thus been constructed is called the *PS refinement* Δ^* of the triangulation Δ. The result is that each triangle of Δ is split into 6 smaller triangles from Δ^*.

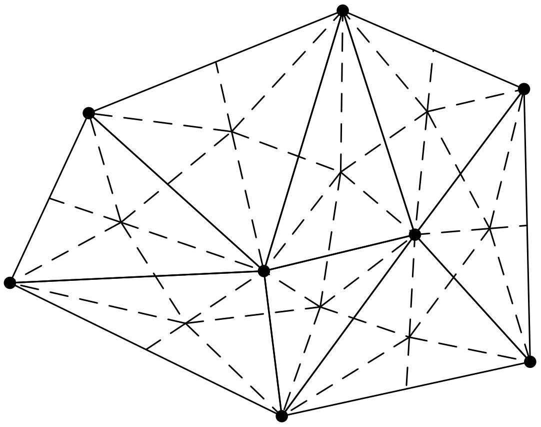


Figure 27: PS refinement. The original triangulation Δ (solid lines) is refined by choosing a new vertex in each triangle and connecting the new vertex with the old vertices, with the new vertices of neighboring triangles, and in the case of a boundary triangle also with the (mid)point of the boundary edges. The new edges are plotted with dashed lines. The finer triangulation Δ^* is a PS refinement of Δ.

The space of PS splines for the triangulation Δ is denoted as $S_2^1(\Delta^*)$ and consists of all piecewise quadratic polynomials in two variables that are C^1 continuous, which means that its values and the first derivatives (hence the normal to the surface and thus also the tangent plane) are continuous. Piecewise means that we have one polynomial per triangle, but the polynomials may differ for different triangles.

It can be shown that the dimension of the space $S_2^1(\Delta^*)$ is $3N$ where N is the number of vertices in Δ. This follows from the fact that there is exactly one $s \in S_2^1(\Delta^*)$ that satisfies the following Hermite interpolating conditions in the N vertices of Δ:

- $s(V_i) = f_i$ (it interpolates function values)
- $\nabla s(V_i) = \nabla f_i$ (the normal, hence the tangent plane to the surface, is matched)

Thus, denoting the derivatives with respect to x and y as s'_x and s'_y, the PS spline will be completely characterized by the data

$$(s(V_i), \nabla s(V_i)) = (s(V_i), s'_x(V_i), s'_y(V_i)) = (\alpha_i, \beta_i, \gamma_i), \quad i = 1, 2, \ldots, N.$$

Since the dimension of $S_2^1(\Delta^*)$ is $3N$, there must be $3N$ basis functions B_{ij}, that can be used to write the PS spline as

$$s(x, y) = \sum_{i=1}^{N} \sum_{j=1}^{3} c_{ij} B_{ij}(x, y).$$

We choose this basis by fixing per vertex V_i three independent triplets $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$, $j = 1, 2, 3$ and defining B_{ij} as the solution in $s \in S_2^1(\Delta^*)$ that satisfies the interpolation conditions

$$(s(V_k), s'_x(V_k), s'_y(V_k)) = \delta_{i-k}(\alpha_{ij}, \beta_{ij}, \gamma_{ij}), \quad j = 1, 2, 3; \;\; k = 1, \ldots, N.$$

Here δ_{i-k} is the Kronecker delta which is 1 for $i = k$ and zero otherwise. Thus the B_{ij} are defined by giving 3 independent triples of interpolation data at vertex V_i and using interpolation data that are zero at all vertices V_k.

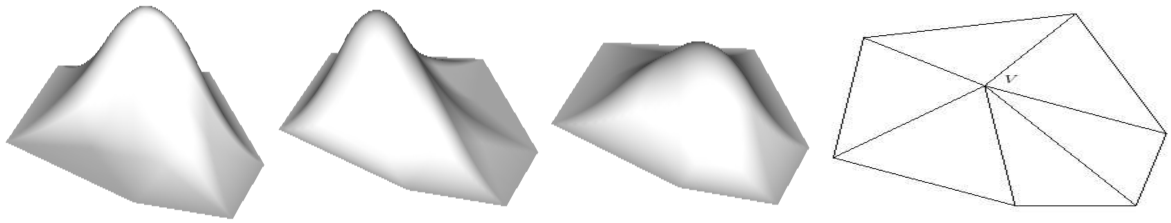


Figure 28: Example of three B-splines associated with the same vertex V. On the right the molecule that is associated with V. It is the support of the three B-splines.

We can choose the 3 nonzero triplets at each vertex in such a way that the basis functions B_{ij} form a *partition of unity*. This means that

- $\sum_{i=1}^{N}\sum_{j=1}^{3} B_{ij}(x,y) = 1$
- $B_{ij}(x,y) \geq 0$.

The basis functions B_{ij} constructed in this way are called B-splines (B stands for basis). The term was coined by Schoenberg [58], the one we use here was given by Dierckx [17]. Three B-splines associated with the vertex V are plotted in Figure 28. Note that in this example they are only different from zero on a hexagonal domain. Their support is the *molecule* of the vertex V, that is the union of all the triangles in Δ that have V as a vertex. This shows that the B-splines constitute a "local basis". The coefficients c_{ij} of B_{ij} in the B-spline decomposition of the PS spline will only have a local effect in a molecule, and the finer the triangulation, the smaller the molecules will be, thus the more localized the domain of influence will be.

The B-spline basis has also some smoothness property that was desirable for wavelets too: it can represent exactly any linear (i.e., flat) surface. This means that we can represent x and y, both polynomials of degree 1, exactly as a linear combination of B-splines. Thus there exist $Q_{ij} = (X_{ij}, Y_{ij}, c_{ij})$, $i = 1, 2, \dots, N$, $j = 1, 2, 3$ such that

$$\begin{aligned} x &= \sum_{i=1}^{N}\sum_{j=1}^{3} X_{ij}B_{ij}(x,y) \\ y &= \sum_{i=1}^{N}\sum_{j=1}^{3} Y_{ij}B_{ij}(x,y) \\ s(x,y) &= \sum_{i=1}^{N}\sum_{j=1}^{3} c_{ij}B_{ij}(x,y) \end{aligned}$$

Thus with each vertex V_i, we can associate three points Q_{ij}, $j = 1, 2, 3$ in 3D that form a triangle. This is the *control triangle*. The control triangle is like the wooden frame pulling the strings that are manipulated by the puppeteer to control his marionette. The control triangle is tangent to the spline surface in the vertex V_i. So its orientation defines the tangent plane in V_i and its three vertices fix the (local) influence of the three B-splines on the surface. In Figure 29 several examples are given of PS surfaces and their control triangles.

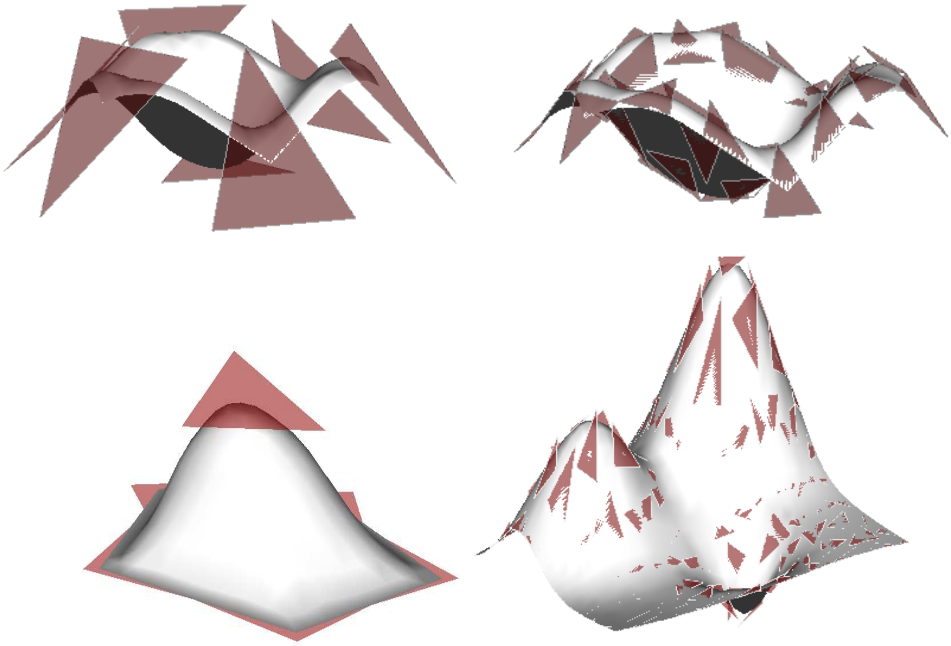


Figure 29: Several examples of PS splines and their control triangles. The top row shows the same surface represented at different resolution levels.

It will not be used in the sequel of this paper, but we just mention here that the PS-spline is a mathematical description of an approximating surface, it is easily transformed into a Bézier control net, and there exists a fast algorithm (the de Casteljau algorithm), which allows to compute its function value and its derivatives (i.e., its normal and hence the tangent plane). This is important because that is exactly what is needed to render nice plots of the surface. The function values describe the shape and the normal defines the reflection of light on the surface.

Note however that on practice (using openGL) the triangular mesh is plotted which gives the shape of the object, and the control triangles define the normals used to control the light scattering. Using a resolution that is high enough will give a result that is hardly distinguishable from the exact surface.

10 SUBDIVISION WITH PS SPLINES

Now we have to describe how a coarse triangulation can be refined. The transition from a PS spline at one level to the PS spline at the next should be easy and efficient.

There are several possibilities. The most natural seems to use the 6-split that was already used in the definition of the PS spline. Figure 30 shows the 6-split, also called the dyadic subdivision because every edge of the triangle is divided in two parts. The figure explains it all: starting from a triangle in Δ (left graph in solid line) and its PS 6-split (dashed lines), the triangle can be subdivided into 4 smaller triangles (middle graph in solid lines) and for these 4 new triangles, a PS 6-split can be constructed efficiently because all of the existing 6-split can be re-used. Re-use of the PS refinement is not only efficient, it is a must if we want the associated PS spline spaces to be nested.

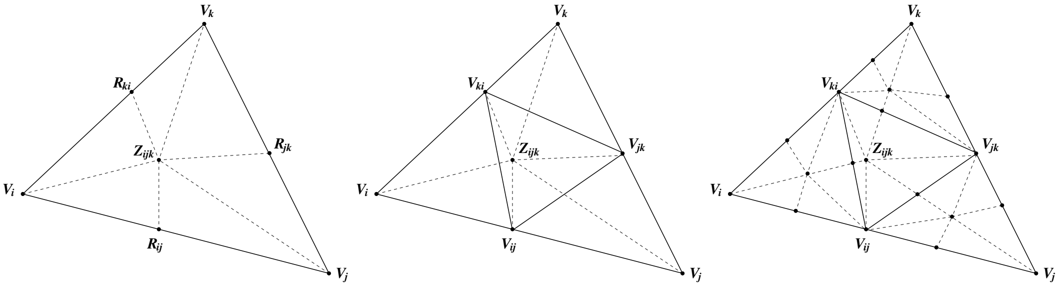


Figure 30: Dyadic subdivision copies the PS 6-split. Every triangles is split into 4 smaller triangles.

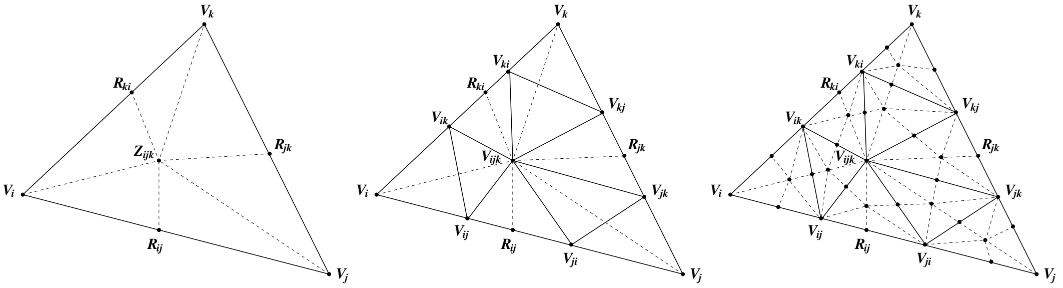


Figure 31: Triadic subdivision: every triangle is split into 9 smaller triangles.

This dyadic subdivision is certainly the simplest possible choice, but there is one severe problem: the central point in the original triangle has to be inside the central one of the smaller triangles. That will be guaranteed if the original triangulation is a uniform one: all the triangles are the same, but this can not be guaranteed for an arbitrary nonuniform triangulation.

Fortunately, there is an alternative: the triadic subdivision. In this case every edge of the original triangle is split into 3 parts. Thus we have to select 2 points on every edge: one on either side of the PS 6-split points that were used in the dyadic scheme. See Figure 31. The original triangle is split into 9 smaller triangles. As can be seen in this figure, the PS 6-split of the original triangle can be completely re-used, but the additional work to move from one level to the next is much more complex. The computational effort needed to gain one level of resolution will grow very fast. The price to pay for one extra step will be quickly too large. We shall gain a lot of resolution, but maybe more than we want. The main advantage over the dyadic subdivision is however that it is always possible, however irregular the triangulation may be.

11 THE HIERARCHICAL BASIS

Now we are all set to describe a hierarchical basis that is the essential element for the multiresolution setting.

We start from a coarse triangulation Δ_0 and its PS refinement Δ_0^*, and associated with this some PS spline space $S_0 = S_2^1(\Delta_0^*)$. By the triadic subdivision (or possibly the dyadic subdivision for a uniform grid), we obtain a finer triangulation Δ_1, its PS refinement Δ_1^*, and a PS spline space $S_1 = S_2^1(\Delta_1^*)$. The important thing is that these objects are nested, and the same holds for alle subsequent steps. Thus we have

$$\begin{aligned}\Delta_0 \subset \Delta_1 \subset \cdots \subset \Delta_\ell \subset \cdots \\ \Delta_0^* \subset \Delta_1^* \subset \cdots \subset \Delta_\ell^* \subset \cdots \\ S_0 \subset S_1 \subset \cdots \subset S_\ell \subset \cdots\end{aligned}$$

If the surface is approximated by a PS spline in S_ℓ and we want to move to a finer approximation in $S_{\ell+1}$ we have to add some detail. So we may write $S_{\ell+1} = S_\ell \oplus W_\ell$ where S_ℓ represents the course approximation space and W_ℓ is the space of the details to be added.

We shall now mimic the interpolating subdivision scheme that was explained in Section 5. We shall even go one step further and construct wavelets. So we have to describe the basis functions that span the spaces S_ℓ and W_ℓ.

If Δ_ℓ has N_ℓ vertices, then the space S_ℓ is spanned by $3N_\ell$ basis functions (B-splines $B_{ij\ell}$) which we arrange in a row vector ϕ_ℓ of length $3N_\ell$. The corresponding coefficients $c_{ij\ell}$ in the corresponding decomposition of the PS spline $s_\ell \in S_\ell$ are arranged in a column vector c_ℓ so that

$$s_\ell(x, y) = \phi_\ell \cdot c_\ell = \sum_{i=1}^{N\ell} \sum_{j=1}^{3} B_{ij\ell}(x, y) \cdot c_{ij\ell}.$$

As we have explained before, when going from level ℓ to level $\ell + 1$, we have to introduce new vertices in $\Delta_{\ell+1}$ (one on every edge of Δ_ℓ in the dyadic case and two per edge in the triadic case), and we re-use the vertices of Δ_ℓ, which we shall call the old vertices of $\Delta_{\ell+1}$. Since we have 3 B-splines associated with every vertex, we can divide the basis functions of $S_{\ell+1}$ into the ones associated with the old vertices (collected in a row vector $\phi_{\ell+1}^o$ of size $3N_\ell$) and the ones associated with the new vertices (collected in a row vector $\phi_{\ell+1}^n$ of size $3(N_{\ell+1} - N_\ell)$). Note however that the $\phi_{\ell+1}^o$ differ from the ϕ_ℓ. The former are associated with a finer triangulation and therefore have a smaller support. But, since $S_\ell \subset S_{\ell+1}$, it should be possible to express the ϕ_ℓ functions as linear combinations of the $\phi_{\ell+1}$. Thus there exists a matrix A_ℓ such that $\phi_\ell = \phi_{\ell+1} A_\ell$.

Now, note that like in the interpolating subdivision scheme of Section 5, the ϕ_ℓ are giving the coarse information, while the $\phi_{\ell+1}^n$ describe the new details that have to be added to get the information at level $\ell + 1$. Thus the ϕ_ℓ are like scaling functions and the $\phi_{\ell+1}^n$ are like wavelet functions in an MRA setting. Therefore we try to describe the basis $\phi_{\ell+1}$ for $S_{\ell+1}$ in terms of ϕ_ℓ and $\phi_{\ell+1}^n$.

This is not difficult. Recall from the previous relation $\phi_\ell = \phi_{\ell+1} A_\ell$. Then, splitting the $\phi_{\ell+1}$ functions into $\phi_{\ell+1}^o$ and $\phi_{\ell+1}^n$ as explained above, we can write for appropriate matrices O_ℓ and N_ℓ

$$[\phi_\ell \;\; \phi_{\ell+1}^n] = [\phi_{\ell+1}^o \;\; \phi_{\ell+1}^n] \begin{bmatrix} O_\ell & 0 \\ N_\ell & I \end{bmatrix}.$$

Since the rightmost matrix is nonsingular, this describes a change of basis for $S_{\ell+1}$. Every $s_{\ell+1} \in S_{\ell+1}$ can be written as an element in S_ℓ plus an element in the space spanned by the $\phi_{\ell+1}^n$.

The term "scaling functions" for ϕ_ℓ and "wavelet functions" for $\phi_{\ell+1}^n$ is somewhat premature though. We do not have all the necessary properties of a wavelet. For example, a wavelet is supposed to oscillate, while the $\phi_{\ell+1}^n$ functions are nonnegative by definition. In fact we have just described the prediction step in a lifting scheme. The update step is still missing.

Nonetheless, there are some interesting problems that have been, (or should be) investigated for these or related hierarchical basis functions.

- What kind of functions can be represented in this way? In other words, if the resolution level ℓ tends to infinity, what is the class of functions of which we may expect convergence? [44, 43, 72, 5, 8, 9, 16].
- What can be proved about stability? This means the following: if we perturb the coefficients c_ℓ slightly (for example when removing the coefficients below a certain threshold), then we want this to result in a small perturbation of the PS spline. This is equivalent with the stability of the basis that is used. Of course this is also linked to the stability of the triangulation. The minimum angle in the triangulation should not become arbitrary small. [43, 72, 81, 49, 44, 45, 9].
- We do not have orthogonality or biorthogonality of the basis functions. This would not only be highly desirable for the efficiency of the computations but it would also solve the stability problem. This problem can be solved for uniform grids and for non-uniform grids in the case of linear splines [10, 62, 63].
- The splines guarantee smoothness. That is why splines were designed in the first place. However, that will make it more difficult to represent (sharp) edges. Special techniques should be, and have been designed to deal with them. [73, 74].
- For a nonuniform grid, the triadic subdivision has to be used instead of the dyadic subdivision. For practical applications, the growth in complexity for the triadic scheme may be too steep, so that it becomes unfeasible to go to even moderately deep resolution levels. It has been shown that the triadic subdivision can be split into two consecutive steps doing half of the work. This is called $\sqrt{3}$ subdivision [36, 39, 77, 78]. It should be noted though that we do not have the nesting $\Delta_0 \subset \Delta_1 \subset \cdots$ anymore but the set of nodes are nested, and that is enough for the machinery to work.

12 POWELL-SABIN SPLINE WAVELETS

Let us now see how we can turn the basis $\phi^n_{\ell+1}$ into a wavelet basis. This requires an update step which corresponds to yet another change of basis in the space W_ℓ [43, 72, 76, 47].

$$[\phi_\ell \ \ \psi_\ell] = [\phi_\ell \ \ \phi^n_{\ell+1}] \begin{bmatrix} I & -C_\ell \\ 0 & I \end{bmatrix}.$$

The update can for example make the integrals of the ψ_ℓ functions vanish, by which the oscillating character is ensured. But one may try to satisfy other criteria as well. We come back to the question of fixing C_ℓ after we have analyzed what properties should hold for the wavelet basis. So, for the moment, assume that we have chosen a C_ℓ.

Putting together our previous transformations of the basis, we get

$$[\phi_\ell \ \ \psi_\ell] = [\phi^o_{\ell+1} \ \ \phi^n_{\ell+1}][A_\ell \ \ B_\ell], \quad \text{with } [A_\ell \ \ B_\ell] = \begin{bmatrix} O_\ell & 0 \\ N_\ell & I \end{bmatrix} \begin{bmatrix} I & -C_\ell \\ 0 & I \end{bmatrix}.$$

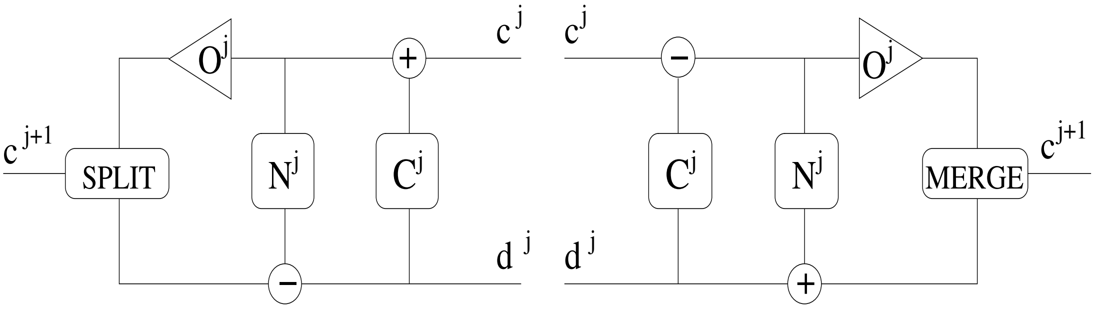


Figure 32: The analysis and synthesis filters in the form of a lifting scheme.

The A_ℓ and B_ℓ are like the low-pass and the high-pass filters used in the analysis phase as we explained in Section 3 for a 1D signal in a filter bank context. Indeed assume the surface is given at a very high resolution level $V_{\ell+1}$, so that the support of the basis functions $\phi_{\ell+1}$ is so small that they may be considered as good approximations to Dirac delta functions, and therefore, we may consider the corresponding coefficients $c_{\ell+1}$ as approximations for the sample values of the surface. By changing the basis $\phi_{\ell+1}$ to the basis consisting of the ϕ_ℓ and the ψ_ℓ functions, we in fact split the surface in a coarse approximation (the ϕ_ℓ components) and the missing detail (the ψ_ℓ components). The two steps in the transformation of the basis that we described above correspond to the two lifting steps in the lifting scheme and they tell us how to transform the $c_{\ell+1}$ coefficients into c_ℓ and d_ℓ coefficients such that

$$s_{\ell+1} = \underbrace{\phi_{\ell+1} \cdot c_{\ell+1}}_{\in S_{\ell+1}} = \underbrace{\phi_\ell \cdot c_\ell}_{\in S_\ell} + \underbrace{\psi_\ell \cdot d_\ell}_{\in W_\ell}.$$

On the synthesis side, we also have two filters that should somehow undo this analysis and reconstruct the surface from its two components. Let us call these filters $\tilde{A}_\ell$ and $\tilde{B}_\ell$. So they recombine the basis functions ϕ_ℓ, ψ_ℓ to get the basis $\phi_{\ell+1}$ back. So they satisfy

$$\phi_{\ell+1} = [\phi_\ell \;\; \psi_\ell] \begin{bmatrix} \tilde{A}_\ell \\ \tilde{B}_\ell \end{bmatrix},$$

and because they invert the decomposition, we also have

$$[A_\ell \;\; B_\ell] \begin{bmatrix} \tilde{A}_\ell \\ \tilde{B}_\ell \end{bmatrix} = I.$$

This is shown in a lifting scheme in Figure 32.

With the dual filters $\tilde{A}_\ell$ and $\tilde{B}_\ell$, we may associate dual sets of basis functions (denoted with a tilde). We summarize everything we have in the following scheme.

Primal		Dual	
$[\phi_\ell \;\; \psi_\ell] = \phi_{\ell+1} M_\ell$	$[\phi_\ell \;\; \psi_\ell]\tilde{M}_\ell = \phi_{\ell+1}$	$[\tilde{\phi}_\ell \;\; \tilde{\psi}_\ell] = \tilde{\phi}_{\ell+1}\tilde{M}_\ell$	$[\tilde{\phi}_\ell \;\; \tilde{\psi}_\ell] M_\ell = \tilde{\phi}_{\ell+1}$
$\tilde{M}_\ell c_{\ell+1} = \begin{bmatrix} c_\ell \\ d_\ell \end{bmatrix}$	$c_{\ell+1} = M_\ell \begin{bmatrix} c_\ell \\ d_\ell \end{bmatrix}$	$M_\ell \tilde{c}_{\ell+1} = \begin{bmatrix} \tilde{c}_\ell \\ \tilde{d}_\ell \end{bmatrix}$	$\tilde{c}_{\ell+1} = \tilde{M}_\ell \begin{bmatrix} \tilde{c}_\ell \\ \tilde{d}_\ell \end{bmatrix}$

where

$$M_\ell = [A_\ell \;\; B_\ell] \;\text{ and }\; \tilde{M}_\ell = \begin{bmatrix} \tilde{A}_\ell \\ \tilde{B}_\ell \end{bmatrix}.$$

Recall that we want to achieve all this by an appropriate choice of the updating filter C_ℓ which appeared in

$$\psi_\ell = \phi^n_{\ell+1} - \phi_\ell C_\ell.$$

If we want to satisfy the orthogonality condition $M_\ell \tilde{M}_\ell = I$, then C_ℓ will not be a sparse matrix. This is bad news for the complexity of the computations, but it is intolerable because it implies that the ψ_ℓ will depend on not just a few local ϕ_ℓ functions, but on all of them and hence it will inherit the support of all the ϕ_ℓ , which is the whole domain, so they are not at all local anymore. Giving up on orthogonality is also not an option because that would result in possibly severe instability of the subdivision process. So we shall have to live with a deal between conflicting requirements. Let us give up on the property that the (dual) basis should be able to reproduce polynomials of the first degree exactly. Instead we relax this and require only the weaker condition that constant functions should be recoverable, which is equivalent to requiring that the wavelet functions oscillate. This will solve already the stability problem to a large extend. So we translate this in an extra linear constraint, expressing that we can write the constant 1 as a linear combination of the $\tilde{\phi}_\ell$ functions. This leaves us with some degrees of freedom to satisfy the orthogonality condition, but not enough to satisfy it exactly. We shall use the remaining free parameters to satisfy orthogonality conditions in a least squares sense. This idea is explained in [42].

This principle of constructing wavelets can be applied in all kinds of situations. We can use other splines (for example of lower degree, giving up on smoothness). This can be generalized to higher dimensions or to the univariate situation as a special case. Of course, then the computations will be much simpler and the orthogonality conditions can be satisfied exactly (if we work on uniform grids). In Figure 33, we give an example of a univariate spline wavelet that is piecewise linear. We show the scaling function ϕ (which is the hat function because of linear interpolation), the corresponding wavelet ψ, and the dual scaling function $\tilde{\phi}$. All functions are zero outside the interval that is shown. The same type of splines in 2D are shown on the second line.

Figure 34 shows typical scaling and wavelet functions for the bivariate quadratic PS splines. Note that near the boundary of the domain of the surface, some adaptations are needed in the construction which introduces extra complications in the computation.

Now we are all set to describe all possible surfaces that can be represented in the form $z = s(x, y)$. Clearly not all 3D objects can be described in that way. Much more general is to use the parametric description in which x, y, and z are described as a function of two parameters. Our previous technique can be used for each of the coordinate functions. In this way, we may obtain a functional approximate description (in the form of a PS spline for example) with the possibility of a multiresolution decomposition.

We take as an example the 3D object of a gargoyle as in Figure 35. This object is given as a point cloud of 10,002 points in 3D. The image is consists of 20,000 triangles. However, using this information, the same object can be described by a much smoother PS spline. This gives a much smoother surface, and it is possible to render it with arbitrary resolution since we have a functional description.

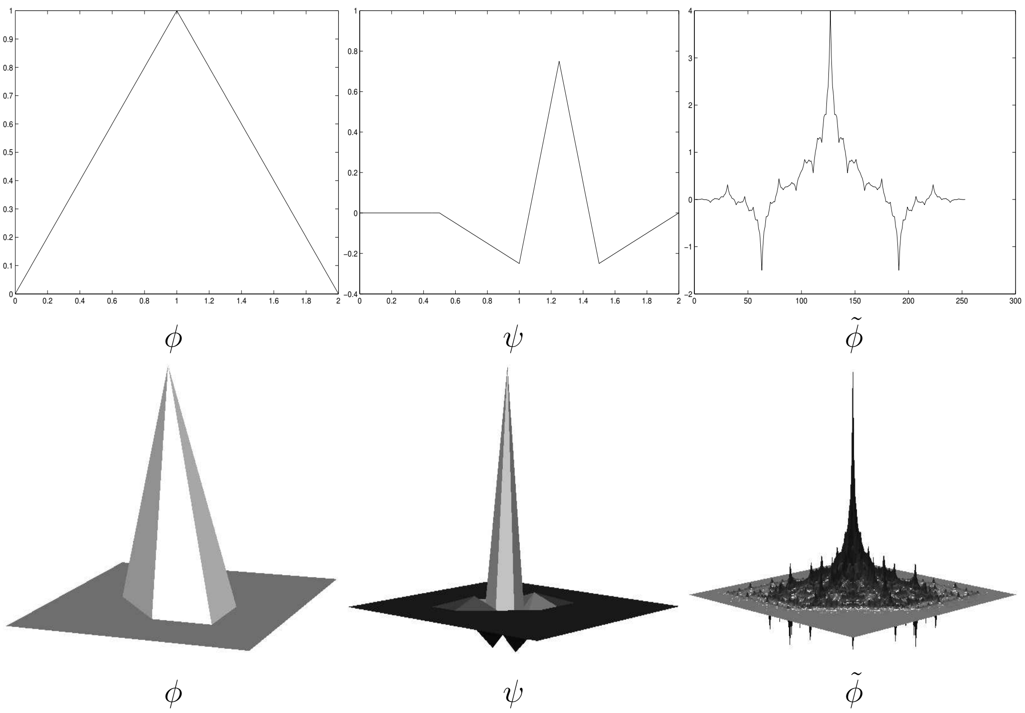


Figure 33: Univariate and bivariate spline wavelets based on splines of degree one (piecewise linear polynomials).

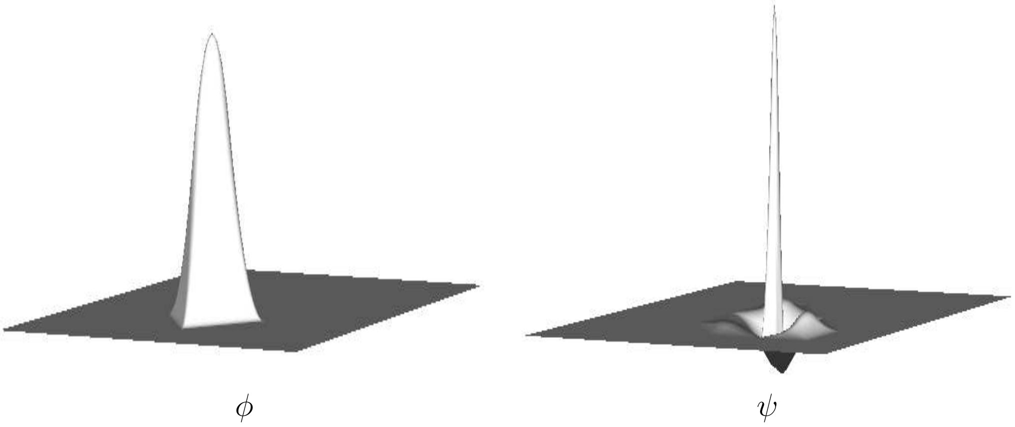
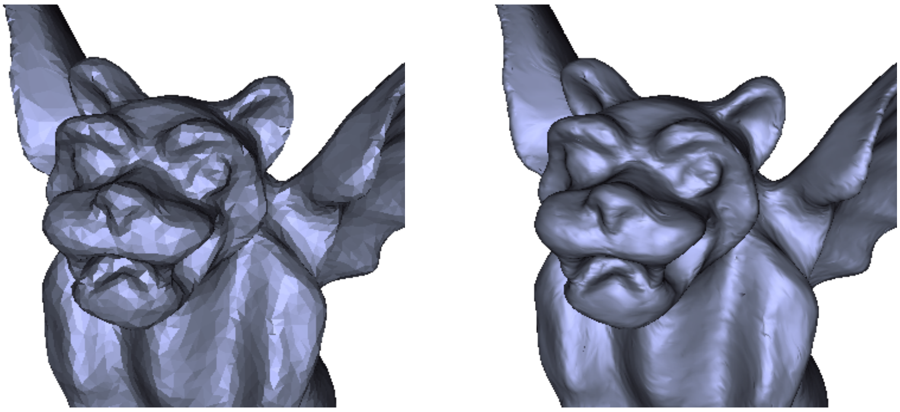


Figure 34: Bivariate quadratic PS spline wavelets. Scaling function and wavelet.



point cloud PS spline

Figure 35: Gargoyle. On the left it is rendered from a point cloud of 10,002 points in 3D. On the right the same object described by a PS spline.

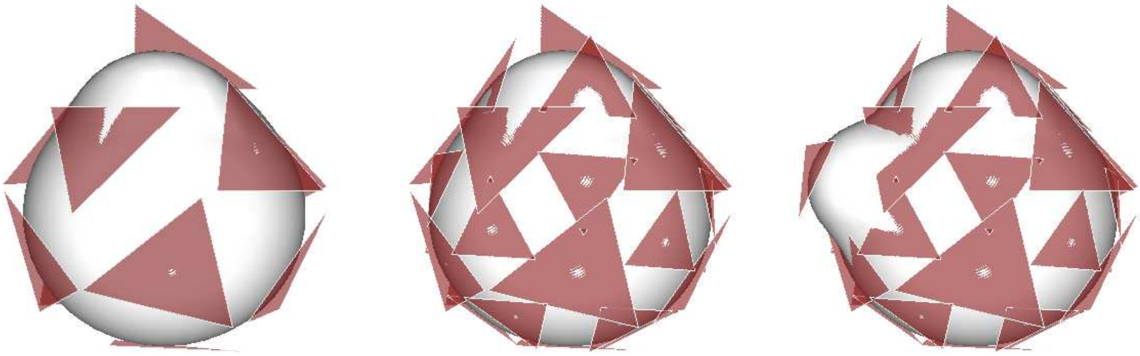


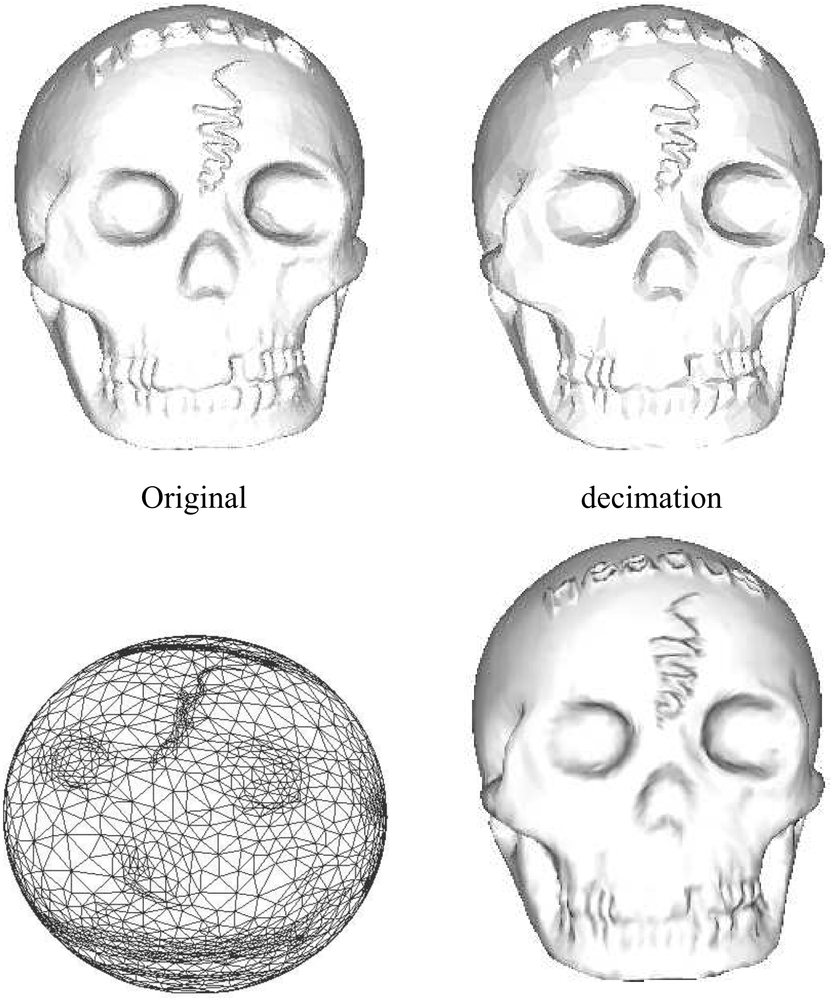
Figure 36: The control trangles shape the spherical PS spline surface locally.

Because PS splines are smooth functions, we do not need to keep all the data points. With only a decimated version it is possible to obtain a result that looks as nice as with the point cloud.

We illustrate this with an even more advanced example. For an object described by a mesh of genus zero that can be mapped onto a sphere, it is natural to develop the whole theory of PS spline surfaces on a sphere. This is not so difficult, the main observation is that the triangulation of the sphere can be locally mapped onto the tangent plane and the construction in the tangent plane which corresponds to what has been discussed before is then radially projected back onto the sphere [46, 48, 43]. All the properties concerning control triangles like being tangent to the spline surface are essentially kept. The control triangles give local control over the spline. See Figure 36.

Consider the skull that is rendered by 40,000 triangles as in the top left image of Figure 37.

A first step is to reduce the number of points. This is called *decimation* and there exist efficient algorithms to do this in an intelligent way [27, 37]. The result of rendering the same skull with only 5,000 triangles is on the top right in Figure 37. The next step is to map the reduced mesh to the sphere, a process called *parametrization* of the mesh [28, 30, 57, 61]. The result is shown in the bottom left image of Figure 37. Each triangle on the sphere corresponds



Original

decimation

parameterization

spherical PS spline

Figure 37: Skull, an example of a mesh with genus zero. The original contains 40,000 triangles, the decimated one 5,000 triangles.

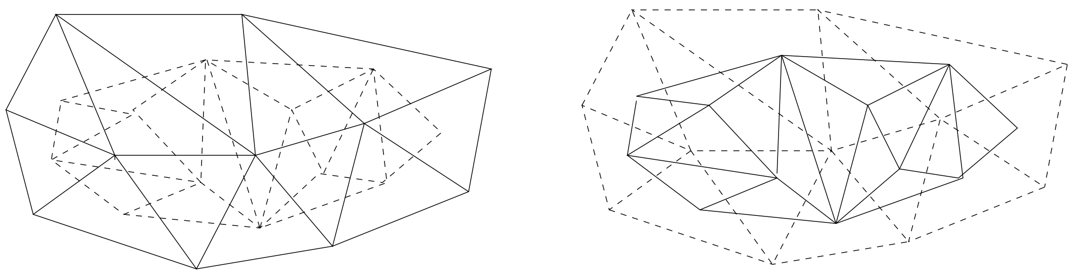


Figure 38: Left: A triangularization (solid lines) and its dual (dashed lines). Right: The original triangles are considered to be the projections of the control triangles of the vertices of the dual triangulation.

to a triangle in the reduced mesh representation and each vertex on the sphere to a point in the reduced mesh.

Now we need to construct control triangles for all the vertices on the sphere. They should be in a tangent plane that goes through the points in the mesh. Note that since we only know a point cloud, a tangent plane is not really defined. So, we construct a dual mesh that results from taking a point inside every triangle on the sphere.

All these points form a new (dual) triangulation. We can consider the triangles of the original triangulation to be the projections of the control triangles for the vertices of the dual mesh. See Figure 38. With the parameters used in defining the B-splines, they can be mapped to control triangles in a plane through the (dual) mesh point. This plane is then by definition considered to be the tangent plane in that point. Note that in the case of a sphere, there is no boundary.

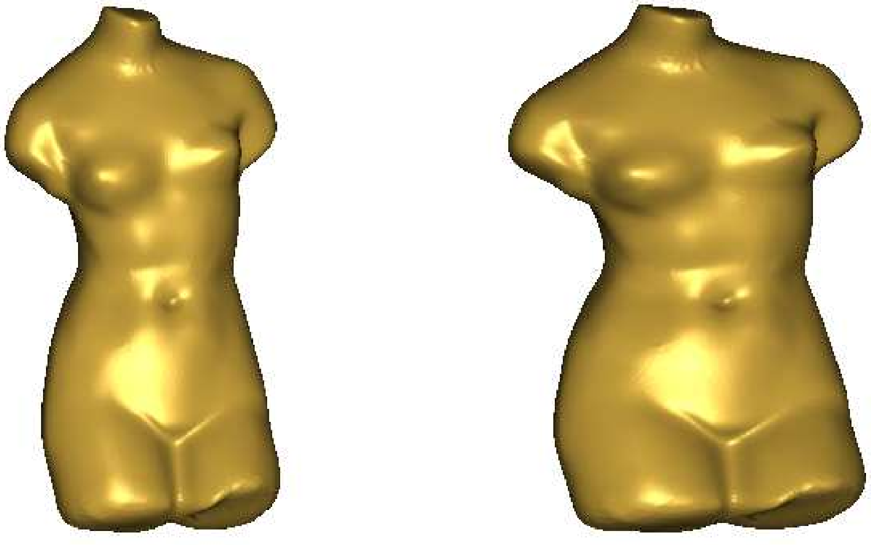
Also in this spherical case we still need a parametric description to characterise a general surface because a radial ray may intersect the surface in more than one point while spherical PS splines have only one value along such a ray.

All this results in a spherical PS spline rendering of the skull as shown in the bottom right picture. One can observe that the text on top of the skull is sharper in the last picture than in the original and it is certainly sharper than for the reduced mesh. This is a consequence of the fact that the functional description of the surface (in this case its three coordinates) with a spline, makes it possible to render at any scale we want, while for the mesh, one is restricted to the number of points in the mesh.

The manipulation of the control triangles will allow the desired reshaping of the object as is illustrated in Figure 39.

13 CONCLUSION

In this brief walk through the jungle of problems and techniques in signal, image and surface processing, we illustrated the meaning and advantages of a multiresolution representation of a function. We have explained what wavelets are supposed to be and how they fit into the multiresolution picture. Subdivision is the way in which a coarse mesh in the support of the function is refined. Associated with this procedure, is a natural splitting of the function values into a coarse predicted value and the corresponding prediction error that corresponds to the detail. This is describing predictive part of the lifting scheme and gives rise to a natural construction of some hierarchical bases. The second part of the lifting scheme is an update



Original Reshaped

Figure 39: The surface can be reshaped by manipulating the control triangles.

step which is desirable to impose certain stability properties of the basis. This step can force the basis to be a wavelet basis, or at least come close to being one. We have introduced these concepts in the one dimensional case and in the case of modeling a surface with Powell–Sabin splines. In the context of image processing, we have opted for a nonlinear subdivision technique based on normal offsets. This is a promising method to deal with horizon-type images, which can be extremely well compressed if the horizons, i.e, the smooth edges in the image are well approximated.

14 ACKNOWLEGEMENT

The work is partially supported by the Fund for Scientific Research (FWO) projects "MISS: Multiresolution on arbitrary grids via subdivision and Powell-Sabin splines", grant #G.0211.02 and "SMID: Stability of Multiscale Transforms on Irregular Data", grant #G.0431.05 and the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with the authors.

REFERENCES

[1] C.M. Brislawn. Fingerprints go digital. *Notices AMS*, 42(11):1278–1283, 1995. Available from: `http://www.ams.org/notices/199511/brislawn.pdf`.

[2] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Comput. Harmonic Anal.*, 5(3):332–369, 1998. Avail-

able from: http://cm.bell-labs.com/cm/ms/who/wim/papers/papers.html#integer.

[3] E.J. Candès. *Ridgelets: Theory and applications*. PhD thesis, Dept. of Statistics, Stanford Univ., 1998. Available from: http://www.acm.caltech.edu/~emmanuel/papers/Thesis.ps.gz.

[4] E.J. Candès and D.L. Donoho. Curvelets- a surprisingly effective nonadaptive representation for objects with edges. In L.L. Schumaker et al., editors, *Curves and surfaces*, Nashville, TN, 1999. Vanderbilt University Press. Available from: http://www-stat.stanford.edu/~donoho/Reports/1999/curveletsurprise.pdf.

[5] J.M. Carnicer, W. Dahmen, and J.M. Peña. Local decomposition of refinable spaces. *Appl. Comput. Harmonic Anal.*, 3(2):127–153, 1996. Available from: ftp://www.igpm.rwth-aachen.de/pub/dahmen/ld.ps.Z.

[6] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological mesh es. *Computer Aided Design*, 10(6):350–355, 1978. Republished in Rosalee Wolfe, Ed, *Seminal Graphics*, ACM Press, 183–188, 1998. Available from: http://dx.doi.org/10.1016/0010-4485(78)90110-0.

[7] R.R. Coifman, Y. Meyer, S. Quake, and M.V. Wickerhauser. Signal processing and compression with wave packets. In Y. Meyer and S. Roques, editors, *Progress in Wavelet Analysis and Applications*, pages 77–93. Editions Frontieres, Gif-sur-Yvette, France, 1993. Proceedings of the International Conference “Wavelets and Applications,” Toulouse, France, 8–13 June 1992. Available from: http://www.math.wustl.edu/~victor/papers/cmqw.pdf.

[8] W. Dahmen. Some remarks on multiscale transformations, stability and biorthogonality. In P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, editors, *Curves and surfaces*. Academic Press, 1994. Available from: ftp://www.igpm.rwth-aachen.de/pub/dahmen/remarks.ps.Z.

[9] W. Dahmen. Stability of multiscale transformations. *J. Fourier Anal. Appl.*, 4:341–362, 1996. Available from: ftp://www.igpm.rwth-aachen.de/pub/dahmen/smt.ps.Z.

[10] W. Dahmen and R. Stevenson. Element-by-element construction of wavelets satisfying stability and moment conditions. *SIAM J. Numer. Anal.*, 37(1):319–352, 2000. Available from: ftp://www.igpm.rwth-aachen.de/pub/dahmen/dst_rev3.ps.Z.

[11] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:909–996, 1988.

[12] I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conf. Ser. in Appl. Math.* SIAM, 1992. Available from: http://www.ec-securehost.com/SIAM/CB61.html.

[13] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4:245–267, 1998. Available from: `http://cm.bell-labs.com/who/wim/papers/papers.html#factor`.

[14] G. Deslauriers and S. Dubuc. Interpolation dyadique. In G. Cherbit, editor, *Fractals, Dimensions non-entières et applications*, pages 44–55, Paris, 1987. Masson. English translation: Fractals: Non-Integral Dimensions and Applications, J. Wiley & Sons, 1991.

[15] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constr. Approx.*, 5(1):49–68, 1989. Available from: `http://dx.doi.org/10.1007/BF01889598`.

[16] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:1–99, 1998. Available from: `http://www.wisc.cs.wisc.edu/~deboor/887_98/nonlinear.ps`.

[17] P. Dierckx. On calculating normalized Powell-Sabin B-splines. *Comput. Aided Geom. Design*, 15(1):61–78, 1997. Available from: `http://dx.doi.org/10.1016/S0167-8396(97)81785-2`.

[18] M. N. Do and M. Vetterli. Contourlets: A directional multiresolution image representation. In *Proc. of IEEE International Conference on Image Processing (ICIP)*. IEEE, 2002. Available from: `http://www.ifp.uiuc.edu/~minhdo/publications/icip_countourlet.pdf`.

[19] D.L. Donoho. De-noising by soft thresholding. *IEEE Trans. Inf. Th.*, 41(3):613–627, 1995. Available from: `http://www-stat.stanford.edu/~donoho/Reports/1992/denoiserelease3.pdf`.

[20] D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994. Available from: `http://www-stat.stanford.edu/~donoho/Reports/1992/mews.pdf`.

[21] N. Dyn, D. Levin, and J. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graphics*, 9(2):160–169, April 1990. Available from: `http://doi.acm.org/10.1145/78956.78958`.

[22] G. Farin. *Curves and surfaces for computer aided geometric design, a practical guide*. Academic Press, 1988.

[23] H. Feichtinger and T. Strohmer, editors. *Gabor analysis and algorithms*. Applied and Numerical Harmonic Analysis. Birkhäuser Verlag, 1998.

[24] H. Feichtinger and T. Strohmer, editors. *Advances in Gabor analysis*. Applied and Numerical Harmonic Analysis. Birkhäuser Verlag, 2003.

[25] M.S. Floater and M. Reimers. Meshless parameterization and surface reconstruction. *Comput. Aided Geom. Design*, 18(2):77–92, 2001. Available from: `http://heim.ifi.uio.no/~michaelf/papers/triangulate.ps.gz`.

[26] G. Folland and A. Sitaram. The uncertainty principle: A mathematical survey. *J. Fourier Anal. Appl.*, 3(3):207–238, 1997.

[27] M. Garland and P.S. Heckbert. Simplifyig surfaces with color and texture using quadratic error metrics. In D. Ebert, H. Hagen, and H. Rushmeier, editors, *IEEE Visualisation '98*, pages 263–270. IEEE, 1998. Available from: `http://graphics.cs.uiuc.edu/~garland/papers/quadratics.pdf`.

[28] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3D meshes. *ACM Trans. Graphics*, 22(3):358–363, 2003. Available from: `http://www.cs.technion.ac.il/~gotsman/AmendedPubl/SphereCDV/SphereCDV.pdf`.

[29] P. Goupillaud, A. Grossmann, and J. Morlet. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration*, 23:85–102, 1984-1985. Available from: `http://dx.doi.org/10.1016/0016-7142(84)90025-5`.

[30] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau. Genus zero surface conformal mapping and its applications to brain surface mapping. In *Proceedings of Infrmation Processing in Medical Imaging*, pages 172–184, 2003. Available from: `http://www.cise.ufl.edu/~gu/papers/ipmi.pdf`.

[31] A. Haar. Zur Theorie der orthogonalen Functionen-Systeme. *Math. Ann.*, 69:331–371, 1910. Available from: `http://dx.doi.org/10.1007/BF0145632`.

[32] W. Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Z. für Physik*, 43:172–198, 1927.

[33] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992. Available from: `http://research.microsoft.com/~hoppe/recon.pdf`.

[34] ISO/IEC. Information technology – JPEG 2000 image coding system: Core coding system, iso/iec standard 15444-1:2004(e), 2004. Second edition. Available from: `http://www.jpeg.org/jpeg2000/`.

[35] M. Jansen, M. Malfait, and A. Bultheel. Generalized cross validation for wavelet thresholding. *Signal Processing*, 56:33–44, 1997. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/gcv/`.

[36] L. Kobbelt. $\sqrt{3}$-subdivision. In *Computer Graphics Proceedings*, Annual Conference Series. ACM SIGGRAPH, 2000. Available from: `http://www-i8.informatik.rwth-aachen.de/publications/downloads/sqrt3.pdf`.

[37] L. Kobbelt, S. Campagna, and H.-P. Seidel. A general framework for mesh decimation. In *Graphics interface '98 Proceedings*, pages 43–50, 1998. Available from: `http://www-i8.informatik.rwth-aachen.de/publications/downloads/mesh.ps.gz`.

[38] V.A. Kotelnikov. On the carrying capacity of the ether and wire in telecommunication. In *First All-Union Conference on Questions of Communicatio*, 1933. (in Russian).

[39] U. Labsik and G. Greiner. Interpolatory $\sqrt{3}$-subdivision. *Comput. Graph. Forum*, 19(3):131–138, 2000. Available from: `http://www.uni-weimar.de/~caw/papers/greiner.pdf`.

[40] M. J. Lai and L. L. Schumaker. On the approximation power of bivariate splines. *Adv. Comput. Math.*, 9:251–279, 1998. Available from: http://atlas.math.vanderbilt.edu/~schumake/approx.pdf.

[41] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Departement of Mathematics, 1987.

[42] M. Lounsbery, T.D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graphics*, 16(1):34–73, 1997. Available from: http://www.cs.rice.edu/~jwarren/papers/wavelet.ps.

[43] J. Maes. *Powell-Sabin spline based multilevel preconditioners for elliptic partial differential equations*. PhD thesis, Dept. Computer Science, K.U.Leuven, 2006. Available from: http://www.cs.kuleuven.be/publicaties/doctoraten/tw/TW2006_02.abs.html.

[44] J. Maes and A. Bultheel. Stable multiresolution analysis on triangles for surface compression. *Electron. Trans. Numer. Anal.*, 2004. Accepted. Available from: http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/smra/.

[45] J. Maes and A. Bultheel. C^1 hierarchical Riesz bases of Lagrange type on Powell-Sabin triangulations. *J. Comput. Appl. Math.*, 2005. Accepted. Available from: http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/lagr/.

[46] J. Maes and A. Bultheel. A hierarchical basis preconditioner for the biharmonic equation on a sphere. *IMA J. Numer. Anal.*, 2005. Published on-line. Available from: http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/hbbh/.

[47] J. Maes and A. Bultheel. Stable lifting construction of non-uniform biorthogonal spline wavelets with compact support. *J. Fourier Anal. Appl.*, 2005. Submitted. Available from: http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/jfa05/.

[48] J. Maes and A. Bultheel. Modeling genus zero closed manifolds with spherical Powell-Sabin B-splines. *Comput. Aided Geom. Design*, 2006. Submitted. Available from: http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/genus/.

[49] J. Maes, E. Vanraes, P.Dierckx, and A. Bultheel. On the stability of normalized Powell-Sabin B-splines. *J. Comput. Appl. Math.*, 170(1):181–196, 2004. Available from: http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/stable/.

[50] S. Mallat. *Multiresolution representation and wavelets*. Phd thesis, Univ. of Pennsylvenia, PA, 1988.

[51] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Anal. Machine Intell.*, 11(7):674–693, 1989. Available from: http://www.cmap.polytechnique.fr/~mallat/papiers/MallatTheory89.pdf.

[52] S. Mann and S. Haykin. The chirplet transform: A generalization of Gabor's logon transform. *Vision Interface '91*, pages 205–212, June 3-7 1991. ISSN 0843-803X. Available from: http://wearcam.org/chirplet/vi91/pork.pdf.

[53] Y. Meyer. Ondelettes et fonctions splines. Séminaire sur les équations aus dérivées partielles, École Polytechnique, Palaisseau, 1987.

[54] G. Nürnberger, V. Rayevskaya, L. L. Schumaker, and F. Zeilfelder. Local Lagrange interpolation with bivariate splines of arbitrary smoothness. *Constr. Approx.*, 23(1):33–59, 2005. Available from: `http://atlas.math.vanderbilt.edu/~schumake/srd.pdf`.

[55] H. Nyquist. Certain topics in telegraph transmission theory. *Trans. AIEE*, 47:617–644, 1928.

[56] M.J.D. Powell and M.A. Sabin. Piecewise quadratic approximations on triangles. *ACM Trans. Math. Software*, 3:316–325, 1977. Available from: `http://portal.acm.org/citation.cfm?doid=355759.355761`.

[57] S. Saba, I. Yavneh, C. Gotsman, and A. Sheffer. Practical spherical embedding of manifold triangle meshes. In *International Conference on Shape Modeling and Applications 2005 (SMI' 05)*, pages 258–267, 2005. Available from: `http://www.cs.technion.ac.il/~gotsman/AmendedPubl/Shadi/spherical-SMI.pdf`.

[58] I.J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Q. Appl. Math.*, 4:45–99, 113–141, 1946.

[59] K.K. Selig. Uncertainty principles revisited. *Electron. Trans. Numer. Anal.*, 14:165–177, 2002. Available from: `http://etna.mcs.kent.edu/vol.14.2002/pp165-177.dir/pp165-177.pdf`.

[60] C.E. Shannon. Communication in the presence of noise. *Proc. Institute of Radio Engeneers*, 37(1):10–21, 1949.

[61] A. Sheffer, C. Gotsman, and N. Dyn. Robust spherical parameterization of triangular meshes. *Computing*, 72(1-2):185–193, 2004. Available from: `http://www.cs.technion.ac.il/~gotsman/AmendedPubl/sphere/sphere.pdf`.

[62] R. Stevenson. Piecewise linear (pre-)wavelets on non-uniform meshes. In W. Hackbusch and G. Wittum, editors, *Multigrid Methods V*, pages 306–319. Springer-Verlag, Heidelberg, 1996. Available from: `http://www.math.ru.nl/onderzoek/reports/rep1997/rep9701.ps.gz`.

[63] R. Stevenson. Locally supported, piecewise polynomial biorthogonal wavelets on non-uniform meshes. *Constr. Approx.*, 19(4):477–508, 2003. Available from: `http://www.math.uu.nl/people/stevenso/expldulsweb.ps.gz`.

[64] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A.F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79, 68-79, 1995. Proc. SPIE 2569. Available from: `http://cm.bell-labs.com/who/wim/papers/papers.html#spie95`.

[65] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmonic Anal.*, 3(2):186–200, 1996. Available from: `http://cm.bell-labs.com/who/wim/papers/papers.html`.

[66] W. Sweldens. The lifting scheme: A custom design construction of biorthogonal wavelets. *Appl. Comput. Harmonic Anal.*, 3(2):186–200, 1996. Available from: `http://cm.bell-labs.com/who/wim/papers/papers.html#lift1`.

[67] W. Sweldens. Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech.*, 76 (Suppl. 2):41–44, 1996. Available from: `http.cm.bell-labs.com/who/wim/papers/papers.html#iciam95`.

[68] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, ACM SIGGRAPH Course Notes. ACM, 1996. Available from: `http://cm.bell-labs.com/who/wim/papers/papers.html#athome`.

[69] G. Uytterhoeven, F. Van Wulpen, M. Jansen, D. Roose, and A. Bultheel. WAILI: A software library for image processing using integer wavelet transforms. In K.M. Hanson, editor, *Medical Imaging 1998: Image Processing*, volume 3338 of *Proc. SPIE Int. Soc. Opt. Eng.*, pages 1490–1501. International Society for Optical Engineering, 1998. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/geertSPIE/`.

[70] W. Van Aerschot, M. Jansen, and A. Bultheel. Adaptive splitting for stabilizing 1-D wavelet decomposition on irregular grids. *Signal Processing*, 2005. Accepted. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/asplit/`.

[71] W. Van Aerschot, M. Jansen, and A. Bultheel. Normal offsets for digital image compression. In F. Truchetet and O. Laligant, editors, *Optics East, Wavelet Applications in Industrial Processing III*, volume 6001 of *SPIE Proceedings*, pages 148–159. SPIE, October 2005. SPIE Optics East conference, Boston, MA, October 23-26, 2005. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/spie05/`.

[72] E. Vanraes. *Powell-Sabin splines and multiresolution techniques*. PhD thesis, Dept. Computer Science, K.U.Leuven, 2004. Available from: `http://www.cs.kuleuven.be/publicaties/doctoraten/tw/TW2004_01.abs.html`.

[73] E. Vanraes and A. Bultheel. Tangent subdivision scheme. *ACM Trans. Graphics*, 2003. Submitted. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/tss/`.

[74] E. Vanraes and A. Bultheel. Modelling sharp features with tangent subdivision. In M. Dæhlen, K. Marken, and L. Schumaker, editors, *Mathematical methods for curves and surfaces, Tromsø 2004*, pages 362–372. Nashboro Press, 2005. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/tromso/`.

[75] E. Vanraes, M. Jansen, and A. Bultheel. Stabilised wavelet transforms for nonequispaced data smoothing. *Signal Processing*, 82(12):1979–1990, 2002. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/stabwt/`.

[76] E. Vanraes, J. Maes, and A. Bultheel. Powell-Sabin spline wavelets. *Internat. J. Wavelets, Multiresolution and Information Processing*, 2(1):23–42, 2003. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/psw/`.

[77] E. Vanraes, J. Windmolders, A. Bultheel, and P. Dierckx. Dyadic and $\sqrt{3}$– subdivision for uniform Powell–Sabin splines. In D. Williams, editor, *Proceedings of the International Symposium of Computer Aided Geometric Design, IV02-CAGD, London, England July 2002*, pages 639–643. IEEE Computer Science Society, 2002. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/iv2002/`.

[78] E. Vanraes, J. Windmolders, A. Bultheel, and P. Dierckx. Automatic construction of control triangles for subdivided Powell-Sabin splines. *Comput. Aided Geom. Design*, 21(7):671–682, May 2004. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/pss/`.

[79] T. Volodine, D. Roose, and D. Vanderstraeten. Efficient triangulation of point clouds using floater parameterization. In M. Lucian and M. Neamtu, editors, *Proceedings of the Eighth SIAM Conference on Geometric Design and Computing*, pages 1–14, 2005. Available from: `http://www.cs.kuleuven.ac.be/publicaties/rapporten/tw/TW385.abs.html`.

[80] E.T. Whittaker. On the functions which are represented by the expansions of the interpolation theory. *Proc. Royal Soc. Edinburgh, Sec. A*, 35:181–194, 1915.

[81] J. Windmolders, E. Vanraes, P. Dierckx, and A. Bultheel. Uniform Powell-Sabin spline wavelets. *J. Comput. Appl. Math.*, 154:125–142, 2003. Available from: `http://www.cs.kuleuven.be/cwis/research/nalag/papers/ade/powsab/`.