# ADAPTIVE SPLINE-WAVELET IMAGE ENCODING AND REAL-TIME SYNTHESIS

*P. C. Marais and E. H. Blake and A. A. M. Kuijk*

CWI, Department of Interactive Systems, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
e-mail: {patrickm, edwin, fons}@cwi.nl

## ABSTRACT

A wavelet-based image-encoding is described which, when used in conjunction with the Difference Engine (a custom-designed VLSI display processor) allows us to reconstruct an image in real-time without the need to set each pixel explicitly. The image is compressed using a quadratic spline-wavelet transform; when reconstructing, an image-adaptive instruction generator attempts to produce the minimal instruction stream to give a good reproduction. The wavelet coefficients are used to decide which regions of the detail images should be retained in the multi-resolution analysis (MRA). A decision is made for each scan-line as to whether it is more economical, in terms of rendering time, to use the 'truncated MRA' or to set the pixels directly. The above approach provides a significant gain over standard image reconstruction/rendering schemes.

## 1. INTRODUCTION

Studies have shown that the *wavelet transform* [1] offers a better compression/fidelity tradeoff than the Discrete Cosine Transform (DCT)[2]. The complexity of the blocked DCT is of the same order as that of an (unblocked) fast wavelet transform — consequently, blocking is not required and blocking artifacts are no longer a problem. Furthermore, the *multi-resolution* structure of the transform allows for resolution-dependent coding techniques.

The 'standard' approach to image synthesis, after such transform coding, is to perform an inverse transform, thus producing the data required for each pixel. However, by requiring that our image be expressible on a suitably defined (quadratic) spline basis, and using the properties of the Difference Engine, it is possible to regenerate the image, progressively, if this is desired, from a *subset* of the full MRA, by examining the transform coefficients which underlie the analysis. This synthesis procedure allows one to reduce the number of instructions required to render an image, when compared with the direct approach.

## 2. THE WAVELET TRANSFORM

A *wavelet*, $\Psi(x,y)$, is an $L^2(\mathbb{R}^2)$ function which satisfies

$$\int\int \Psi(x,y)dxdy = 0 \tag{1}$$

This condition ensures that the wavelet is localized in both time and frequency and exhibits a measure of oscillation — hence the name. The *discrete (dyadic) wavelet transform*, $(W_\Psi I)(j;i,l)$ of an $L^2(\mathbb{R}^2)$ function, $I(x,y)$, with respect to the wavelet $\Psi$ is defined as

$$(W_\Psi I)(j;i,l) = \langle \Psi_{j;i,l}, I\rangle,\ i,j,l \in \mathbb{Z} \tag{2}$$

where $\langle , \rangle$ denotes the $L^2$ inner product and $\Psi_{j;i,l}(x,y) \equiv 2^j\Psi(2x-i, 2y-l)$. For non-orthogonal wavelets, there is a corresponding *dual wavelet*, $\tilde{\Psi}$, which satisfies the relationship

$$\langle \Psi_{k;i,p}, \tilde{\Psi}_{l;j,q}\rangle = \delta_{kl}\delta_{ij}\delta_{pq}. \tag{3}$$

It can be shown that the functions $\{\Psi_{j;i,l};\ j,k,l \in \mathbb{Z}\}$ span the space $L^2(\mathbb{R}^2)$ [1]. Hence, any function, $I(x,y)$, in this space can be written as a linear combination of such scaled and translated wavelets:

$$I(x,y) = \cdots + g_{-1}(x,y) + g_0(x,y) + g_1(x,y) + \cdots \tag{4}$$

where

$$g_j(x,y) = \sum_{i,l} d_{j;il}\Psi_{j;i,l}(x,y),\ j \in \mathbb{Z}. \tag{5}$$

Because of the *bi-orthogonality relation*, Equation (3), one may write $d_{j;il} = \langle I, \tilde{\Psi}_{j;i,l}\rangle,\ i,j,l \in \mathbb{Z}$.

## 3. MULTI-RESOLUTION ANALYSIS

A *multi-resolution analysis* (MRA) of $L^2(\mathbb{R})$ produces a sequence of spaces, $V_{j+1} \equiv V_j \oplus W_j$, with $V_j \subset V_{j+1}$. The space $W_j$ is the orthogonal complement of $V_j$ in $V_{j+1}$. See [3] for a description of MRA and its relation to wavelet transforms.

A common method used to generate a 2-D MRA, is to take the tensor product of the corresponding 1-D multi-resolution analysis with itself [1]. This provides one with *three* wavelets, $\Psi^{[p]}(x,y)$, $p = 1,2,3$ and a scaling function, $\Phi(x,y)$, all of which are separable 2-D functions:

$$\Psi^{[1]}(x,y) = \phi(x)\psi(y) \tag{6}$$
$$\Psi^{[2]}(x,y) = \psi(x)\phi(y) \tag{7}$$
$$\Psi^{[3]}(x,y) = \psi(x)\psi(y) \tag{8}$$
$$\Phi(x,y) = \phi(x)\phi(y) \tag{9}$$

These wavelets are essentially orientated, resolution-dependent band-pass filters; the scaling function may be viewed as a low-pass filter. The detail spaces, spanned by each wavelet type, thus contain difference information with a specific orientation only: vertical, horizontal and diagonal.

The multi-resolution pyramid goes off to infinity in both directions. However, realisable signals are band-limited. Thus, we truncate the representation, discarding all higher level information, by 'projecting' our input function into a space which has sufficient detail to represent the sampled signal — $V_0$ by convention. Similarly, since signals do not always contain arbitrarily low frequencies, it may be unnecessary to decompose one's signal beyond a certain level. Thus, one has a $J$th level multi-resolution decomposition

$$\begin{aligned} I(x,y) &\equiv I_0(x,y) \\ &= g_{-1}(x,y) + \cdots + g_{-J}(x,y) + I_{-J}(x,y) \\ &= \sum_{l=-1}^{-J} \sum_{i,j} \sum_{p=1}^{3} d_{[p]_{l;ij}} \Psi^{[p]}_{l;i,j}(x,y) + \\ & \quad \sum_{i,j} c_{-J;ij} \Phi_{-J;i,j}(x,y). \end{aligned} \tag{10}$$

The wavelet transform is also truncated; the $J$th level discrete wavelet transform provides the set of coefficients

$$\{\{d_{[p]_{l;ij}}\}, \{c_{-J;ij}\},\ i,j \in \mathbb{Z}, l = -1,-2,\ldots -J;\ \ p = 1,2,3\} \tag{11}$$

### 3.1. Semi-Orthogonal Cardinal Spline MRA

The space of cardinal splines of order $m$, $S_m$, contains all those functions expressible as a weighted sum of $m$th order *cardinal B-splines*, $N_m(x)$:

$$f(x) = \sum c_k N_m(x-k), \quad f \in S_m. \tag{12}$$

The values of $N_m(x)$ may be found using the following identity:

$$N_m(x) = \frac{x}{m-1} N_{m-1}(x) + \frac{m-x}{m-1} N_{m-1}(x-1). \tag{13}$$

where

$$N_1(x) = \chi_{[0,1)}(x) = \begin{cases} 1 & \text{if } x \in [0,1); \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

The spline-based MRA introduced in [3, 4] has $N_m(x)$ as its scaling function.

The corresponding $m$th order spline wavelet, $\psi_m(x)$, has support on the interval $[0, 2m-1]$. This wavelet is *semi-orthogonal*, meaning that it is orthogonal to scaled versions of itself, but not to translates on the same resolution level. These functions satisfy the following *two-scale* relationships

$$N_m(x) = \sum_{k=0}^{m} p_k N_m(2x-k), \tag{15}$$

$$\psi_m(x) = \sum_{k=0}^{3m-2} q_k N_m(2x-k) \tag{16}$$

The values of these sequences, for the quadratic case, can be found in [3].

## 4. CALCULATION OF THE WAVELET COEFFICIENTS

Before one can use the MRA, a means must be found to compute the coefficients of the wavelet transform. To this end we use the filtering scheme proposed in [5]. In the context of this work, this gives us the following set of separable convolutional equations for computing the detail and approximation coefficients (from the approximation coefficients of the previous level):

$$c_{j-1;kl} = \sum_m \sum_n a_{m-2k} a_{n-2l} c_{j;mn} \tag{17}$$

$$d_{[1]_{j-1;kl}} = \sum_m \sum_n a_{m-2k} b_{n-2l} c_{j;mn} \tag{18}$$

$$d_{[2]_{j-1;kl}} = \sum_m \sum_n b_{m-2k} a_{n-2l} c_{j;mn} \tag{19}$$

$$d_{[3]_{j-1;kl}} = \sum_m \sum_n b_{m-2k} b_{n-2l} c_{j;mn}. \tag{20}$$

To reconstruct the approximation coefficients (from those lower down in the analysis), one has the following *reconstruction relation*:

$$\begin{aligned} c_{j;km} = \ & \textstyle\sum_l \sum_t p_{k-2l} p_{m-2t} c_{j-1;lt} + \\ & \textstyle\sum_l \sum_t p_{k-2l} q_{m-2t} d_{[1]_{j-1;lt}} + \\ & \textstyle\sum_l \sum_t q_{k-2l} p_{m-2t} d_{[2]_{j-1;lt}} + \\ & \textstyle\sum_l \sum_t q_{k-2l} q_{m-2t} d_{[3]_{j-1;lt}}. \end{aligned} \tag{21}$$

The $\{a_k\}$ and $\{b_k\}$ sequences can be found in [6].

### 4.1. Calculation of $\{c_{0;i,j}\}$

In order that we can use the filtering scheme above, one must first generate the initial set of approximation coefficients, $\{c_{0;ij}\}$ — which are the basis coefficients of the B-spline representation of the input image. If one just wants to achieve compression, the image samples may be used as the initial coefficient values. If, however, one wishes to evaluate the MRA, then these values must be properly computed.

We use *quasi-interpolation* [7] to obtain these coefficients. Quasi-interpolation is a local interpolation scheme, in which the amount of data used to determine the approximating quasi-interpolant can be limited. In this work a 3x3 convolution mask ($k = 1$, below) was used to determine the required coefficients:

$$c_{0;ij} = (\lambda_k I)(i,j),\ i,j \in \mathbb{Z},\ I \in L^2(\mathbb{R}^2). \tag{22}$$

This sequence is computed as

$$\{(\lambda_k I)(i)\} = (\delta - m + \cdots + (-1)^k \underbrace{m * \cdots * m}_{k \text{ times}}) * I^0(i), i \in \mathbb{Z}^2, \tag{23}$$

where $\delta \equiv \delta_{i,j;0} = 1$ if i, j = 0, and 0 otherwise and

$$m_{i,j} = \begin{cases} \Phi(0,0) - 1 & \text{for i, j} = 0; \\ \Phi(i,j) & \text{for i, j} \neq 0. \end{cases} \tag{24}$$

Because the B-splines must be centred [7], $\Phi(x, y) = N_3(x+3/2)N_3(y+3/2)$, and the coefficient values actually represent the shifted image $I_0(x+3/2, y+3/2)$. It is important to remember this shift when evaluating image functions in the MRA.

## 5. QUANTIZATION

We used *vector quantization* to compress the wavelet encoded image. The approach of [8] was used: the various wavelet sub-bands were sub-divided into 2x2 or 4x4 blocks (as determined by the desired compression ratio) and these blocks were quantized with the previously trained codebooks to yield 8-bit indices (thus permitting 256 reproduction levels per sub-band). The LBG algorithm with a minimum mean-squared error measure was used [9]. The codebook was trained with a collection of disparate images, so as not to introduce any kind of image bias; the test images were not in the training sequence. As is done elsewhere, for example [8, 10], the entropy of the coefficient sequence is used as a measure of compression i.e., we assume that the quantization is followed by a perfect entropy coding.

## 6. THE DIFFERENCE ENGINE

The Difference Engine (DE) is the final component in the rendering pipeline of a new display architecture developed at CWI [11]. This display processor has the ability to interpolate an arbitrary length polynomial span with a single instruction, in time proportional to the degree of the polynomial . The forward difference interpolatory logic is implemented as a systolic array — each new cycle produces the complete set of difference values for the specified span. The 11ns cycle time of this processor means that one can perform these calculations with sufficient speed to ensure pixel production at the display refresh rate.

There are two important points which should be noted:

- the DE can interpolate arbitrary order polynomials, in time proportional to the degree (currently $n+2$ cycles for a polynomial of degree $n-1$).
- the DE provides a scan-line accumulator.

The existence of an intensity accumulator is essential if one wishes to use the DE for multi-resolution image synthesis, since one then needs to accumulate several levels of detail for each scan-line.

## 7. MULTI-RESOLUTION IMAGE SYNTHESIS

The various images in the quadratic cardinal spline MRA satisfy certain very stringent conditions:

- They are elements of $C^1(\mathbb{R}^2)$
- The approximation images consist of quadratic patches, with support on

$$[2^j k, 2^j(k+1)] \times [2^j k, 2^j(k+1)],\ k \in \mathbb{Z}$$

- The detail images also have this property, but over squares half the size on the resolution level $j$.

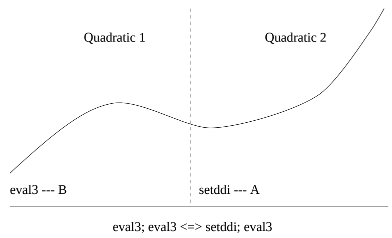


Figure 1: **The use of setddi.**

Thus, the image data along a scan-line (on each level) is composed of adjacent quadratic segments of the same length. It is a simple matter to compute the differences for any such polynomial (using the shifted image functions), and to compose the DE instructions which will interpolate the polynomial scan-line data.

If used without care, multi-resolution synthesis can be far more expensive (in terms of DE instruction cycles) than just setting each pixel directly, since many instructions must be issued to accumulate all the detail information for each scan-line. If however, only 'busy' regions of the detail images are added back to the approximation image, this 'truncated' MRA can provide significant gains over direct reconstruction (i.e., IWT and setting each pixel directly). Wavelet compression should maintain only the most important coefficients viz. those which will ensure good reconstruction fidelity. These retained coefficients can be used as an indication of 'busy' image areas, and the bases which they weight can be used to build the truncated MRA. We determine the extents of these bases which intersect the current scan-line — this information is recorded and used to determine whether it is more economical (in terms of DE instruction cycles required) to simply set the pixels in the current scan-line or to render the truncated MRA. If the latter option is selected, the function evaluations are done and the tiers of detail are accumulated on top of the approximation signal. If it is less economical (as will be the case in highly detailed regions), the scan-line pixels are set directly.

Due to the continuity constraints, and the architecture of the chip, we need only issue one quadratic interpolation instruction to interpolate the entire approximation scan-line: only the second order differences need be changed as we cross each new span boundary — Figure 1. These can be computed and set before the interpolation instruction is issued, by using a low cost set-difference instruction. A similar strategy can be used for detail scan-line segments consisting of several adjacent spans.

To improve performance, neighbouring quadratic spans are merged if their differences are the same; this reduces the number of instructions required to interpolate a multi-span segment. However, since this kind of redundancy is only likely to occur in the approximation image, merging is not applied to detail scan-line segments. Furthermore, for reasons of efficiency, the merging procedure is not applied prior to deciding what kind of synthesis method to employ. Doing so would require additional calculations which would be wasted if direct synthesis were used.

Figure 2: **Test Images.** The (8-bit grey-scale) images are, from top left, Lenna, House and Sugarbowl.

Figure 3: Typical VQ compression result — 0.82 bpp.

## 8. RESULTS

### 8.1. Wavelet Compression

It was apparent that the fidelity of the reconstructed images left something to be desired, even at modest bit-rates (around 1 bpp) — Figure 3. There are a number of reasons for this lack of performance, in particular, the use of a MMSE distortion metric, which takes no account of edge information and does not guarantee simultaneous minimization of reconstruction error and transform domain quantization error (since Parseval's identity does not hold in a semi-orthogonal framework). Simple thresholding tests revealed that MMSE VQ was not exploiting the redundancy provided by the wavelet transform effectively.

### 8.2. Image Synthesis

The results given below are based on a three level wavelet decomposition in which, rather than applying VQ, the wavelet coefficients were thresholded and those retained were used in the MR synthesis calculations. This was done to decouple the compression implementation from the synthesis algorithms, since the former retained too many (unrepresentative) coefficients to illustrate the concepts referred to earlier. The thresholding used is adapted to orientation and resolution level and forms part of the new compression scheme we are investigating. To enable us to quantify the gains produced by MR synthesis, we introduce the Gain Factor (GF) — the ratio of the instruction cycles required to render the image directly to the number of cycles required if adaptation is used. The GF is always $\geq 1.0$.

Table 1 summarizes the results of this preliminary work. Observe that two sets of data are given: the first uses the current cycle costs for the relevant instruction[1] while the second uses the cycle costs which will be used in subsequent implementations of the DE.

| | Cycles | Lenna | House | Sugarbowl |
|---|---|---|---|---|
| Setddi | 2 | 0 | 4 | 62 |
| Eval0 | 1 | 100 | 95 | 33 |
| Eval1 | 3 | 0 | 0.5 | 0.2 |
| Eval3 | 5 | 0 | 0.5 | 4.8 |
| GF | - | 1.23 | 1.63 | 3.43 |
| MR | - | No | Yes | Yes |
| | Cycles | Lenna | House | Sugarbowl |
| Setddi | 1 | 1.4 | 7.6 | 65.7 |
| Eval0 | 1 | 97.5 | 87.0 | 13.9 |
| Eval1 | 1 | 1.0 | 4.7 | 14.7 |
| Eval3 | 3 | 0.1 | 0.7 | 5.7 |
| GF | - | 1.22 | 1.57 | 3.86 |
| MR | - | Yes | Yes | Yes |

Table 1: **Synthesis Results.** The first four rows of each table give the percentages each of the instruction types contributed to the final rendering cost. The final row indicates whether multi-resolution synthesis was invoked or not. The same threshold was employed with all images. The second table gives the figures when the proposed lower cost instructions are used.

There are several things which were evident from our experiments. Firstly, the smoothness of an image is directly related to the gains obtainable when using MR synthesis: the more texture the image possess, the less likely MR synthesis is to yield any benefit, unless the texture is highly localised. In the latter case, the non-textured scan-lines can still be rendered more cheaply. Secondly, image detail is expensive to render, because a) it is present on multiple levels of the MRA and b) the quadratic spans are smaller and consequently more instructions are required to interpolate a scan-line. This is the motivation for truncating the MRA.

Images which are themselves composed of splines (such as the Phong shaded images in [11], of which 'Sugarbowl' is an example) will experience greater gains than other (smooth) images. However, the extent of this reduction will depend on the size of the spline patches of which the image

[1] The interpolation instructions are of the form 'eval$n$', where $n$ is the order of the polynomial to be interpolated; 'eval0' switches off accumulation of subsequent pixel values at the given location, otherwise acting like an 'eval1' — since it is cheaper, it is used for direct reconstruction. The 'setddi' instruction can be used to set the second difference at a specified point; subsequent interpolations, passing through this point, will uses this value rather than the one they had been propagating.

is composed and for most images these are fairly small. The DE is ideally suited to rendering such images.

The images ‘Sugarbowl’ and ‘House’ were both able to derive varying degrees of benefit from MR synthesis, since there were regions in which the intensity data varied slowly. ‘Lenna’ contains a lot of texture; but with lower instruction costs, it becomes economical to use the MRA on some scan-lines. In highly uniform or smooth images, span merging on the approximation level can become significant, boosting rendering efficiency substantially. An extreme example of this would be an object on a uniform background; the background would only be present in the approximation image and could be generated very quickly and efficiently.

For highly textured images, when we are forced to chose direct reconstruction, we can still gain by merging neighbouring pixels; this saves one having to set each pixel individually. Since pixels are usually correlated, even the most chaotic of images may benefit (albeit marginally) from such merging. In the examples given above, Lenna experienced a GF of 1.23 from such pixel merging: all neighbouring pixels along a scan-line which are within one gray scale of the first pixel considered are approximated by this initial value, and a zero-degree polynomial (eval0) of the appropriate length is emitted. When using MR synthesis, smooth images can yield very large gains (a GF of $> 3$ for non-trivial images like Sugarbowl). The *nature* of the smoothness plays an important role in determining the magnitude of these gains i.e., is the image actually a spline, or just smoothish? True spline images can be approximated with fewer resolution levels and coefficients.

Although not explicitly indicated in the tables above, the level of the decomposition has a very definite affect on the rendering gains one can achieve. If the number of levels is too low, then one gains nothing in rendering time, since short pixel spans (less than the order of the polynomial) must be set directly. If, on the other hand, the number of levels is too high, then too much information must be accumulated from the detail tiers and the rendering efficiency drops. A three level decomposition appears to be optimal.

The DE is able to produce low resolution approximation images very efficiently, since the spline patches are then quite large (the 3rd level approximation of Lenna can be rendered in a quarter of the time required to render the full image, using the old instruction costs). Progressive transmission is possible if the receiver is equipped with a screen buffer in which incoming information can be accumulated.

## 9. CONCLUSION

Although the implementation of the quantization algorithm was inadequate, the compression potential of the spline WT can be exploited by a better algorithm. Smooth images can be rendered more rapidly using MR synthesis than by direct reconstruction. Even heavily textured images can be rendered more efficiently if zero-degree pixel merging is applied to exploit pixel correlation.

## 10. FUTURE WORK

A better quantization system is currently under development. Work can be done to improve the usability of the DE w.r.t. MR synthesis — the DE was not specifically designed to render this kind of structure. One of the modifications that can be made, is the addition of a screen-wide accumulator which the DE can access to enable efficient rendering of progressively transmitted images. Work can also be done to improve the simple efficiency measures used — the emphasis here was on rendering performance, which assumes that the MR data can be produced at an adequate rate. All the required information can be computed using parallelised FFT hardware — so on the face of it, this assumption is a reasonable one. Nonetheless, one may desire a different measure of efficiency.

## Acknowledgment

We would like to thank CWI (Centre for Mathematics and Computer Science) for supporting this research as well as the South African Foundation for Research Development (FRD).

## 11. REFERENCES

[1] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, vol 61 in CBMS-NSF series in applied mathematics, 1992.

[2] P. Desarte, B. Macq, D. Slock, *Signal-adapted Multiresolution Transform for Image Coding*, IEEE Trans. on Info. Theory, vol 38, no 2, 1992, pp719–746.

[3] C.K. Chui, *An Introduction to Wavelets: Wavelet Analysis and its Applications*, Academic Press, 1992, Boston.

[4] M. Unser, A. Aldroubi, M. Eden, *A family of polynomial spline wavelet transforms*, Signal Processing, vol 30, 1993, pp 141–162.

[5] S. Mallat, *A Theory of Multiresolution Signal Decomposition: The Wavelet Representation*, IEEE Trans. on Patt. Ana. and Mach Intell, vol 11, 1989, pp 674–693.

[6] P. Marais, E. Blake, A. Kuijk, *A Cardinal-Spline Image Decomposition On A Systolic Array Display Processor*, CWI Quarterly, vol 4, 1993.

[7] C.K. Chui and H. Diamond, *A Natural Formulation of Quasi-Interpolation by Multi-Variate Splines*, Proceedings of the American Mathematical Society, vol 99, no 4, 1987.

[8] M. Antonini and M. Barlaud and P. Mathieu and I. Daubechies, *Image Coding using Wavelet Transforms*, IEEE trans. on image processing, vol 1, no 2, 1992, pp 205–220.

[9] Y. Linde, A. Buzo, R. M. Gray, *An algorithm for vector quantizer design*, IEEE trans. on comm, vol com-28, no 1, 1980, pp 84–95.

[10] P. Westerink, D. Boekee, J. Biemond, *Sub-band coding of images using vector quantization*, IEEE trans on comm, vol 36, no 6, 1988, pp 713–719.

[11] E. H. Blake and A.A.M. Kuijk, *A Difference Engine For Images With Applications To Wavelet Decomposition*, Proceedings of the Second International Conference on Image Communications (IMAGE'COM), 1993, pp 309–314.