Local Filters of B-spline Wavelets

Faramarz F. Samavati
Department of Computer Science
University of Calgary
samavati@cpsc.ucalgary.ca

Richard H. Bartels
Department of Computer Science
University of waterloo
rhbartel@cgl.uwaterloo.ca

Abstract

Haar wavelets have been widely used in Biometrics. One advantage of Haar wavelets is the simplicity and the locality of their decomposition and reconstruction filters. However, Haar wavelets are not satisfactory for some applications due to their non-continuous behaviour. Having a particular level of smoothness is important for many applications. B-spline wavelets are capable of being applied to signals and functions of any smoothness. However, the conventional B-spline wavelets results ”non-local” decomposition filters and consequently, they are not efficient as are the Haar wavelets.

We present our recently developed local filters of B-spline wavelets. Here, we focus on quadratic case that guarantees once-differentiable smoothness. Practical issues for the efficient implementation are discussed. We show that how the resulting filters can be applied to curves, images and surfaces.

Key words: subdivision, reverse subdivision, wavelets, multiresolution, B-splines, Haar.

1 Introduction

Wavelets and multiresolution have many applications related to the Biometric area such as data compression and noise removal. However, they are specifically appealed for extraction of high detailed features in fingerprint and iris recognition[3], [7] and [8]. To date, mostly Haar wavelets have been considered. We introduce an alternative multiresolution approach based on quadratic B-spline wavelets.

1.1 Wavelets and Multiresolution

Multiresolution operations are specified in terms of the filter matrices $\mathbf{A}^n$, $\mathbf{B}^n$, $\mathbf{P}^n$ and $\mathbf{Q}^n$. Consider a given discrete signal C^n, expressed as a column vector. A lower-resolution sample C^{n-1} is created by a downsampling filter on C^n. This process can be expressed as a matrix equation:

$$C^{n-1} = \mathbf{A}^n C^n. \tag{1}$$

The *details* D^{n-1} lost through the down-sampling are captured using $\mathbf{B}^n$:

$$D^{n-1} = \mathbf{B}^n C^n. \tag{2}$$

The pair of matrices $\mathbf{A}^n$ and $\mathbf{B}^n$ are called *analysis filters* and the process of splitting a signal C^n into C^{n-1} and D^{n-1} is called *decomposition*.

Recovering the original signal C^n is called *reconstruction*. It involves refinement of the low-resolution sample C^{n-1} and details D^{n-1} using the *synthesis filters* $\mathbf{P}^n$ and $\mathbf{Q}^n$, which basically reverse the operations of $\mathbf{A}^n$ and $\mathbf{B}^n$:

$$C^n = \mathbf{P}^n C^{n-1} + \mathbf{Q}^n D^{n-1}. \tag{3}$$

Such a decomposition and reconstruction corresponds to some underlying function spaces $\mathcal{V}^{n-1} \subset \mathcal{V}^n$ wherein C^n defines some function $c^n = \sum_i c_i^n \phi_i^n$ in the large space, C^{n-1} defines an approximation $c^{n-1} = \sum_j c_j^{n-1} \phi_j^{n-1}$ to that function in the smaller space, and D^{n-1} defines the difference $d^{n-1} = \sum_k d_k^{n-1} \psi_k^{n-1}$ in the complement space $\mathcal{V}^n \setminus \mathcal{V}^{n-1}$. The basis functions ψ^{n-1} are conventionally called *wavelets* and the ϕ are called *scale functions*, since the ϕ^n are frequently scaled and shifted versions of the ϕ^{n-1}.

For an efficient representation, the following properties are desired:

- C^{n-1} is a good approximation for C^n.
- The storage requirement for storing C^{n-1} and D^{n-1} is not more than that of C^n.
- The time required to decompose and reconstruct the signal is linearly dependent on n.

We can recursively decompose the original signal to $C^l, C^{l+1}, \ldots, C^{n-1}$ and details $D^l, D^{l+1}, \ldots, D^{n-1}$ where $l < n$. The original signal C^n can be recovered from the sequence $C^l, D^l, D^{l+1}, \ldots, D^{n-1}$, known as a *wavelet transform*. Based on the properties mentioned above the total size of the transform $C^l, D^l, D^{l+1}, \ldots, D^{n-1}$ is the same as that of the original signal C^n. In addition the required time to transform C^n to $C^l, D^l, D^{l+1}, \ldots, D^{n-1}$ and vice versa is is a linear function of n.

If C^n represents a high-resolution approximation of a curve, then C^l is a very coarse approximation of the curve showing the main outline, and D^i consist of vectors which perturb the curve into its original path. As Figure1 demonstrates, if we eliminate D^i, the reconstructed curve becomes much smoother but without any style. In fact, D^i can be considered as *characteristic* of the curves. It is possible to apply D^i to a new coarse curve to obtain a new curve but with the same character(See Figure 1). Consequently, D^i at different levels are important features for applications such as iris recognition.

The matrices $\mathbf{A}^n$, $\mathbf{B}^n$, $\mathbf{P}^n$ and $\mathbf{Q}^n$ form the core of the multiresolution approach, and the efficiency of the wavelet transform depends on the structure of these matrices. Specifically: banded matrices with repeated columns result in more efficient decomposition and reconstruction operations.

B-splines are often chosen as scaling functions [5]. The first order B-splines form a set of step functions and Haar functions are their associated wavelets [12, 13]. The resulting matrix filters are very simple and efficient. However, these scaling functions and wavelets are non-continuous. This is a problem when we have discrete data that is a sample of smooth signals and objects. Higher order B-splines and their wavelets can be considered for smooth signals [6, 5, 10]. A common knot arrangement for B-splines of order k is to have knots of single multiplicity uniformly spaced everywhere except at the ends of the domain, where knots have multiplicity k, [9, 1], an arrangement used for endpoint-interpolating curves and surfaces. The conventional definition of B-spline wavelets results in analysis whose bands are much wider than those for Haar wavelets.

We have introduced a new approach to construct multiresolution filters based on *reverse subdivision* as described in [2] and [11]. Based on this approach, it is possible to obtain banded matrices for B-spline wavelets whose bands are narrower than the ones conventionally produced. In this paper we present filter matrices for one important case, quadratic B-splines with the conventional knot sequence.The local filters of quadratic B-spline has been constructed based on Chaikin subdivision[4], for which the underlying scale functions are the quadratic B-splines. These filters are obtained from the approach of [2] and [11]. However, in this work, we emphasize on practical issues for efficient implementation.

Our local filters are described by matrix notation in Section 2. Efficient algorithms for the resulting filters are described in Section 3. We show that how the resulting filters can be applied to curves, images and surfaces in Section 4.

2 Filters by Matrix Notation

The pair of matrices $\mathbf{A}^n$ and $\mathbf{B}^n$, analysis filters, for decomposing the signal C^n into C^{n-1} and D^{n-1} are:

$$\mathbf{A}^n = \begin{bmatrix} \mathbf{A}^n_s \\ \mathbf{A}^n_r \\ \mathbf{A}^n_e \end{bmatrix} \tag{4}$$

where

$$\mathbf{A}^n_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ -\frac{1}{2} & 1 & \frac{3}{4} & -\frac{1}{4} & 0 & 0 & \cdots \end{bmatrix} \tag{5}$$

$$\mathbf{A}^n_r = \begin{bmatrix} 0 & 0 & -\frac{1}{4} & \frac{3}{4} & \frac{3}{4} & -\frac{1}{4} & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & -\frac{1}{4} & \frac{3}{4} & \frac{3}{4} & -\frac{1}{4} & \cdots \\ & & & & & & & & \ddots \end{bmatrix} \tag{6}$$

$$\mathbf{A}^n_e = \begin{bmatrix} \cdots & 0 & 0 & -\frac{1}{4} & \frac{3}{4} & 1 & -\frac{1}{2} \\ \cdots & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

And the same block notation for $\mathbf{B}^n$:

$$\mathbf{B}^n = \begin{bmatrix} \mathbf{B}^n_s \\ \mathbf{B}^n_r \\ \mathbf{B}^n_e \end{bmatrix} \tag{8}$$

where

$$\mathbf{B}^n_s = \begin{bmatrix} -\frac{1}{2} & 1 & -\frac{3}{4} & \frac{1}{4} & 0 & 0 & 0\ldots \\ 0 & 0 & -\frac{1}{4} & \frac{3}{4} & -\frac{3}{4} & \frac{1}{4} & 0\ldots \end{bmatrix} \tag{9}$$

$$\mathbf{B}^n_r = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{4} & -\frac{3}{4} & \frac{3}{4} & -\frac{1}{4} & 0 & 0\ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & -\frac{3}{4} & \frac{3}{4} & -\frac{1}{4}\ldots \\ & & & & & & & & \ddots & \end{bmatrix} \tag{10}$$

$$\mathbf{B}^n_e = \begin{bmatrix} \cdots & 0 & 0 & \frac{1}{4} & -\frac{3}{4} & 1 & -\frac{1}{2} \end{bmatrix} \tag{11}$$

$\mathbf{A}^n_s$ and $\mathbf{A}^n_e$ always have two rows for any n. $\mathbf{A}^n_r$ has different size based on the value of n however it always has a regular structure. Elements of each row are obtained by shifting right by two positions of the elements of the previous row. Therefore applying $\mathbf{A}^n$ to the high-resolution data C^n reduces the size of the data to $\frac{n+2}{2}$. In addition, the regularity of $\mathbf{A}^n_r$ allows us to obtain a linear time algorithm instead of matrix multiplication. The similar properties are true for $\mathbf{B}^n_r$. The size of D^n becomes $\frac{n-2}{2}$ and again a linear time algorithm can be employed instead of the operation $\mathbf{B}^n C^n$.

Using the blocked matrix notation for the synthesis filters $\mathbf{P}$, we obtain:

$$\mathbf{P}^n = \begin{bmatrix} \mathbf{P}^n_s \\ \mathbf{P}^n_r \\ \mathbf{P}^n_e \end{bmatrix} \tag{12}$$

where

$$\mathbf{P}_s^n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & \cdots \end{bmatrix} \tag{13}$$

$$\mathbf{P}_r^n = \begin{bmatrix} 0 & \frac{3}{4} & \frac{1}{4} & 0 & 0 & \dots \\ 0 & \frac{1}{4} & \frac{3}{4} & 0 & 0 & \dots \\ 0 & 0 & \frac{3}{4} & \frac{1}{4} & 0 & \dots \\ 0 & 0 & \frac{1}{4} & \frac{3}{4} & 0 & \dots \\ 0 & 0 & 0 & \frac{3}{4} & \frac{1}{4} & \dots \\ 0 & 0 & 0 & \frac{1}{4} & \frac{3}{4} & \dots \\ & & & & & \ddots \end{bmatrix} \tag{14}$$

$$\mathbf{P}_e^n = \begin{bmatrix} \dots & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \dots & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

And similarly for $\mathbf{Q}^n$:

$$\mathbf{Q}^n = \begin{bmatrix} \mathbf{Q}_s^n \\ \mathbf{Q}_r^n \\ \mathbf{Q}_e^n \end{bmatrix} \tag{16}$$

where

$$\mathbf{Q}_s^n = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots \\ \frac{1}{2} & 0 & 0 & 0 & \dots \\ -\frac{3}{4} & \frac{1}{4} & 0 & 0 & \dots \\ -\frac{1}{4} & \frac{3}{4} & 0 & 0 & \dots \\ 0 & -\frac{3}{4} & -\frac{1}{4} & 0 & \dots \\ 0 & -\frac{1}{4} & -\frac{3}{4} & 0 & \dots \end{bmatrix} \tag{17}$$

$$\mathbf{Q}_r^n = \begin{bmatrix} 0 & 0 & \frac{3}{4} & -\frac{1}{4} & 0 & 0 & \dots \\ 0 & 0 & \frac{1}{4} & -\frac{3}{4} & 0 & 0 & \dots \\ 0 & 0 & 0 & \frac{3}{4} & -\frac{1}{4} & 0 & \dots \\ 0 & 0 & 0 & \frac{1}{4} & -\frac{3}{4} & 0 & \dots \\ 0 & 0 & 0 & 0 & \frac{3}{4} & -\frac{1}{4} & \dots \\ 0 & 0 & 0 & 0 & \frac{1}{4} & -\frac{3}{4} & \dots \\ & & & & & & \ddots \end{bmatrix} \tag{18}$$

$$\mathbf{Q}_e^n = \begin{bmatrix} \dots & 0 & 0 & 0 & \frac{1}{2} \\ \dots & 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

Again we have a column regularity structure for $\mathbf{P}_r^n$ and $\mathbf{Q}_r^n$.

3 Filters by an Efficient Algorithm

We show how an efficient algorithm can be made based on the multiresolution filters in 2. For all algorithms we have focused on doing just one step of decomposition or reconstruction. Each algorithm can be used multiple times to construct general wavelet transform. In all cases F represents the vector of high-resolution data, C represent low-resolution data and D represents the detail vector.

The first algorithm is REDUCE-RESOLUTION. In this algorithm $F[1..m]$ is the input and the vector C is the output. The index i traverses the F and j traverses the C.

```
REDUCE-RESOLUTION($F[1..m]$)
 1  $C_1 = F_1$
 2  $C_2 = -\frac{1}{2}F_1 + F_2 + \frac{3}{4}F_3 - \frac{1}{4}F_4$
 3  $j = 3$
 4  for $i = 3$ to $m - 5$   step $2$
 5      $C_j = -\frac{1}{4}F_i + \frac{3}{4}F_{i+1} + \frac{3}{4}F_{i+2} - \frac{1}{4}F_{i+3}$
 6      $j = j + 1$
 7    endfor
 8  $C_j = -\frac{1}{4}F_{m-3} + \frac{3}{4}F_{m-2} + F_{m-1} - \frac{1}{2}F_m$
 9  $C_{j+1} = F_m$
10  return $C[1..j+1]$
```

The lines 1 and 2 in the REDUCE-RESOLUTION correspond to $\mathbf{A}_s^n$ matrix and the lines 8 and 9 correspond to $\mathbf{A}_e^n$ matrix. The for loop represents the act of the regular block $\mathbf{A}_r^n$.

The second algorithm is FIND-DETAILS.

```
FIND-DETAILS($F[1..m]$)
$D_1 = -\frac{1}{2}F_1 + F_2 - \frac{3}{4}F_3 + \frac{1}{4}F_4$
$D_2 = -\frac{1}{4}F_3 + \frac{3}{4}F_4 - \frac{3}{4}F_5 + \frac{1}{4}F_6$
$j = 3$
for $i = 5$ to $m - 5$ step $2$
    $D_j = \frac{1}{4}F_i - \frac{3}{4}F_{i+1} + \frac{3}{4}F_{i+2} - \frac{1}{4}F_{i+3}$
    $j = j + 1$
  endfor
$D_j = \frac{1}{4}F_{m-3} - \frac{3}{4}F_{m-2} + F_{m-1} - \frac{1}{2}F_m$
return $D[1..j]$
```

For the reconstruction we employ the following algorithm to evaluate $\mathbf{P}^nC + \mathbf{Q}^nD$.

```
RECONSTRUCTION($C[1..r], D[1..s]$)
$E_1 = 0D_1$
$E_2 = \frac{1}{2}D_1$
$E_3 = -\frac{3}{4}D_1 + \frac{1}{4}D_2$
$E_4 = -\frac{1}{4}D_1 + \frac{3}{4}D_2$
$E_5 = -\frac{3}{4}D_2 - \frac{1}{4}D_3$
$E_6 = -\frac{1}{4}D_2 - \frac{3}{4}D_3$
$j = 7$
for $i = 3$ to $s - 1$
    $E_j = \frac{3}{4}D_i - \frac{1}{4}D_{i+1}$
    $E_{j+1} = \frac{1}{4}D_i - \frac{3}{4}D_{i+1}$
    $j = j + 2$
  endfor
```

```
13 $E_j = \frac{1}{2}D_s$
14 $E_{j+1} = 0D_s$
15
16 $F_1 = C_1 + E_1$
17 $F_2 = \frac{1}{2}C_1 + \frac{1}{2}C_2 + E_2$
18 $j = 3$
19 **for** $i = 2$ **to** $r - 2$
20     $F_j = \frac{3}{4}C_i + \frac{1}{4}C_{i+1} + E_j$
21     $F_{j+1} = \frac{1}{4}C_i + \frac{3}{4}C_{i+1} + E_{j+1}$
22     $j = j + 2$
23   **endfor**
24 $F_j = \frac{1}{2}C_{r-1} + \frac{1}{2}C_r + E_j$
25 $F_{j+1} = C_r + E_{j+1}$
26 **return** $F[1..j + 1]$
```

Lines 1 through 14 in RECONSTRUCTION make the $E = \mathbf{Q}^n D$ term. Lines 1 through 6 correspond to $\mathbf{Q}_s^n$ and lines 13 and 14 are for the act of $\mathbf{Q}_e^n$. The for loop at the line 8 is for the regular block $\mathbf{Q}_r^n$. In the lines 1, E_1 has been set to $0D_1$ instead of 0 to have general algorithm that can work for the data with any dimension induced by D. Lines 16 trough 25 make $F = \mathbf{P}^n C + E$. Again the terms $\mathbf{P}_e^n, \mathbf{P}_r^n$ and $\mathbf{P}_e^n$ are distinguishable in the algorithm.

Note that m the size of the high-resolution data is equal to $r + s$. It is clear that the running time of all three algorithm are linear function of m the size of F.

4 Results

4.1 Curves

All filters and given algorithms are capable to apply directly to the curves. For example for a 3D curve F_i can be viewed as a coordinate vector in the three dimensional Euclidean space. The filters form valid operations in this Affine space. Figure1 shows an application of using quadratic B-spline wavelets.

4.2 Images

Similar to Haar Wavelets, we need to apply simultaneously one particular filter to all rows(or columns) of the image. Figure 2 shows an example of the filters on an image.

4.3 Surfaces

The filters can apply just only for tensor-product surfaces that can be parameterized with a rectangular domain. For these kind of surfaces two sets of embedded curves,u-curves and v-curves, exist on the surface. The filters can be used for all the u-curves (or v-curves) simultaneously. Figure 3 shows an example of a tensor-product surface. Each vertical curve on the surface is a v-curve and each circular curve on the surface is u-curve.

4.4 Closed data

A closed circular data such a circle-like curve doesn't need extra-ordinary operations near to the boundaries. For this kind of data that may appears in curves and surfaces, we only need to use regular filters (6),(10),(14) and (18).

4.5 Re-Sampling

A constraint on the number of input data is enforced by the B-spline scaling functions. This is similar to that imposed by the Fast Fourier Transform. In general, to achieve a decomposition without any excess points, we require the size of high-resolution data to be a value $m = 2^k + N$, where N is a small constant associated with the multiresolution scheme used. For the quadratic B-spline $N = 2$. An observation on the recursive structure of the filtering leads us to a less rigid constraint: for a decomposition of l levels, m may be of the form $2^l k + N$. Once the ideal number of points is known, the input data should be uniformly re-sampled to that number.

5 Conclusion

We described a set of new local filters of quadratic B-spline wavelets. An efficient algorithm for each filter operation was presented. We would like to explore the impact of these filters on iris recognition as a future work.

6 Acknowledgement

This work has been supported by the National Science and Engineering Research Council of Canada.

References

[1] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, 1987.

[2] R. H. Bartels and F. F. Samavati. Reversing subdivision rules: Local linear conditions and observations on inner products. *Journal of Computational and Applied Mathematics*, 119(1–2):29–67, July 2000.

[3] W.W. Boles, and B. Boashash.*A Human Identification Technique Using Images of the Iris and Wavelet Transform*. IEEE Trans. on Signal Processing, 46(4), 1998, pp.1185-1188.

[4] G. Chaikin. An algorithm for high speed curve generation. Comp. Graph. and Im. Proc. 3 (1974) 346–349.

[5] C. Chui. *An Introduction to Wavelets*. Academic Press, 1992.

[6] A. Finkelstein and D. H. Salesin. Multiresolution curves, in A. Glassner, editor, Proceedings of SIGGRAPH '94 (ACM Press, New York, 1994) 261–268.

[7] A. K. Jain, A. Prabhakar, L. Hong, and S. Pankanti .*Filterbank-based Fingerprint Matching*.IEEE Transactions on Image Processing, Vol. 9, No.5, pp. 846-859, May 2000.

[8] S. Lim , K. Lee, O. Byeon, and T. Kim.*Recognition through Improvement of Feature Vector and Classifier*. ETRI Journal, Volume 23, Number 2, June 2001.

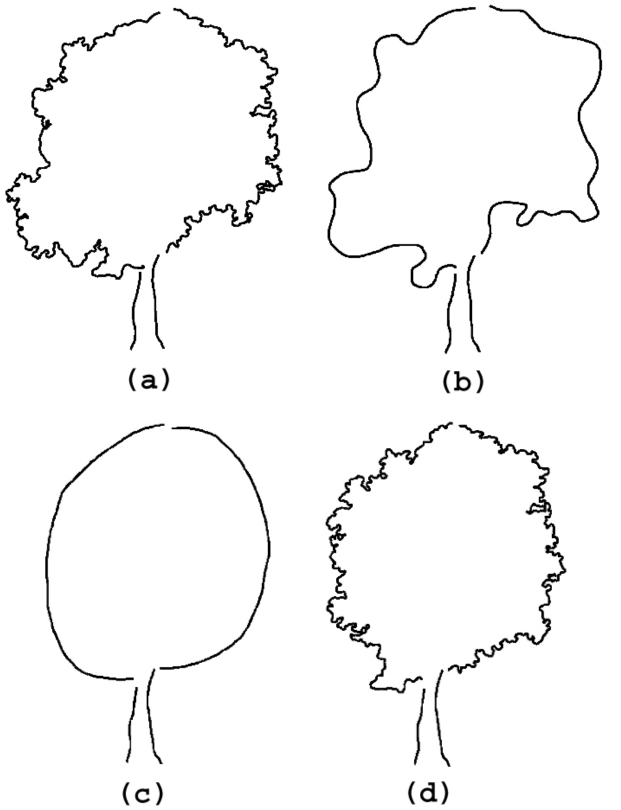

Figure 1: Tree Curve: (a)The original curve. (b) The reconstructed without D^i. (c) A new base curve. (d) The reconstructed curve with the (c) as the base curve and D^i.

[9] L. Piegl and W. Tiller. *The NURBS Book*. 2nd Edition. Springer, 1997.

[10] F. F. Samavati and R. H. Bartels.*Multiresolution curve and surface representation by reversing subdivision rules*. Computer Graphics Forum, 18(2), 97-119, June 1999.

[11] F. F. Samavati and R. H. Bartels. *Diagrammatic Tools for Generating Biorthogonal Multiresolutions*. Technical Report 2003-728-31, Computer Science Department, University of Calgary, 2003.

[12] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann Publishers, 1996.

[13] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

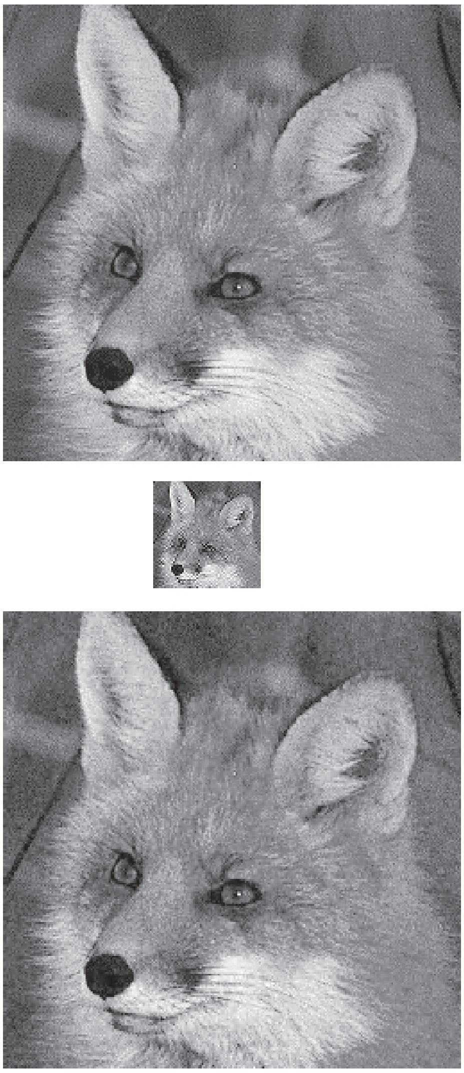

Figure 2: Fox Image: (Top) The original image. (Middle) The low-resolution image after two levels of decomposition. (Bottom) The reconstructed image with Smallest 60% D removed.

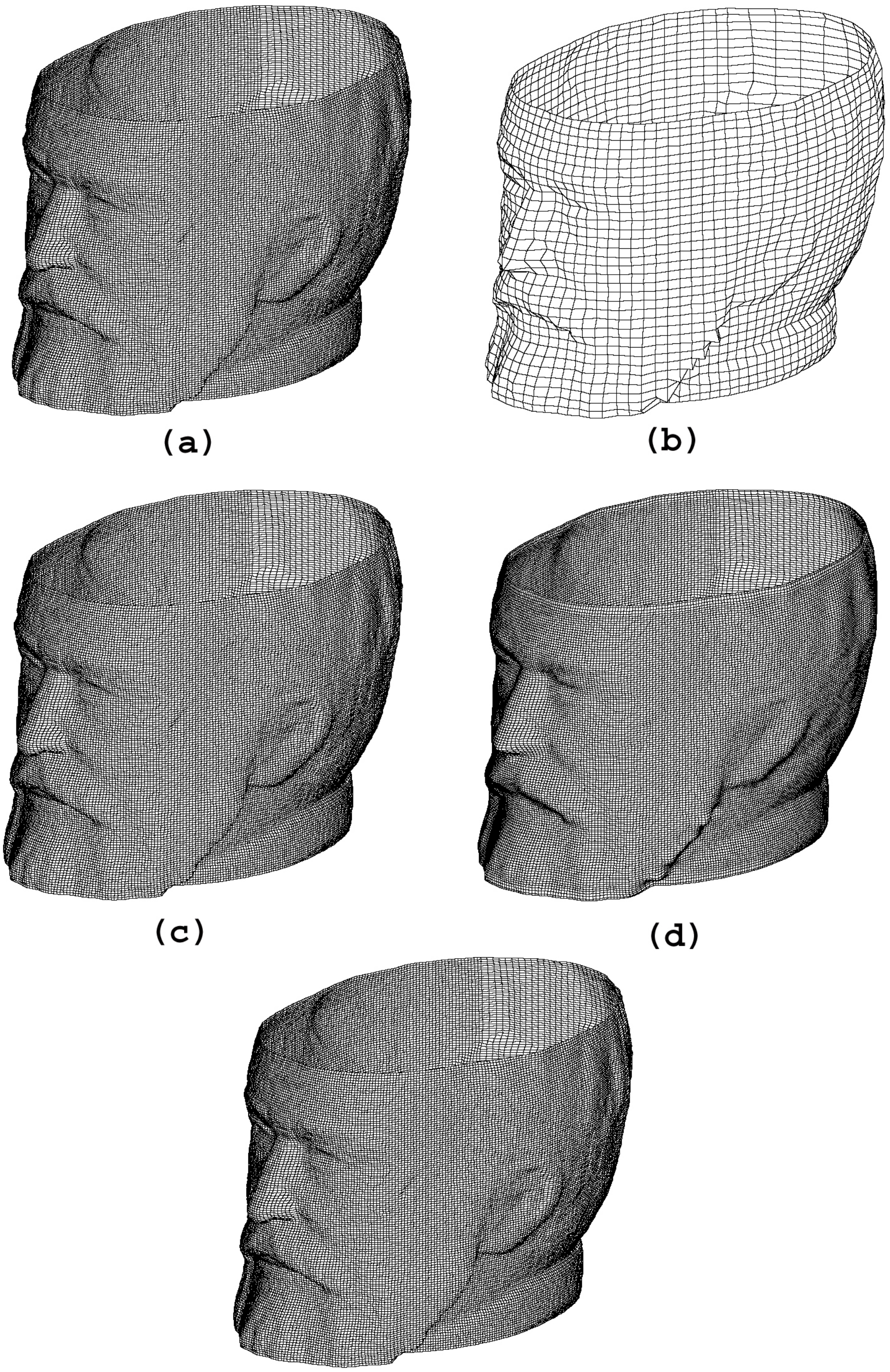

Figure 3: Victor Hugo Surface: (a) The original surface. (b) The coarse surface after two levels of decomposition in each direction, (c) The reconstructed with D^i. (d) The reconstructed without D^i. (e) The reconstructed with the smallest 60% of D^i.