Wavelets on Planar Tesselations

Martin Bertram[1,2] Mark A. Duchaineau[1] Bernd Hamann[2] Kenneth I. Joy[2]

Abstract *We present a new technique for progressive C^0-continuous approximation and compression of polygonal objects in images. Our technique uses local parametrizations defined by meshes of convex polygons in the plane. We generalize a lifted biorthogonal wavelet transform to polygonal domains to perform multiresolution analysis and compression of image regions. The advantage of our technique over conventional wavelet methods is that the domain is an arbitrary tesselation rather than a rectilinear grid. We expect that this technique has many applications image compression, progressive transmission, radiosity, virtual reality, and image morphing.*

Keywords: Image Compression, Multiresolution Methods, Subdivision Surfaces, Tesselations, Wavelets

1 Introduction

Wavelet techniques [6, 20] are used for compression of images, progressive transmission, morphing, and solving complex mathematical problems like the integration of radiosity kernels [10]. The advantage of the discrete wavelet transform over many other techniques is linear computation time and sparse representation of highly detailed functions on rectilinear domains. In this paper we define wavelets on planar tesselations to allow for more flexible domains that can be deformed for morphing images or video compression.

Wavelets defined on arbitrary topology (two-manifolds with arbitrary genus) have been introduced by Lounsbery [12, 13]. These constructions are defined on subdivision surface schemes [5, 8, 14] that refine an arbitrary polygonal or triangular mesh in a regular way and thus converge rapidly to a limit surface. In a related paper [1] we presented a highly efficient approach generalizing tensor-product B-spline wavelets to arbitrary topology. We used a generalized bicubic wavelet transform to represent and compress isosurfaces [2]. Constructions on spherical domains were provided by Schröder/Sweldens [17], Nielson *et al.* [16], and Bonneau [3]. Signal processing algorithms for mesh hierarchies with completely irregular subdivision were presented by Daubechies *et al.* [7] and Guskov *et al.* [11].

It has been shown by Stam that subdivision surfaces often can be evaluated analytically at arbitrary parameter values [19]. Thus, they are an ideal tool for constructing continuous basis functions on irregular domains like tesselations. Most subdivision schemes that offer tangent plane continuity, however, assume that the parametric domain is deformed by the same subdivision rules. For evaluation of the corresponding basis functions at global parameter values, this deformation needs to be inverted, which cannot be done in a closed form in general. For simpler subdivision schemes that generate C^0-continuous surfaces like linear or bilinear splines, it is straight-forward to construct global closed-form parametrizations.

We use the generalized bilinear B-spline wavelets constructed in [1] as basis functions on tesselations. We present a complete signal processing algorithm based on this wavelet transform and outline some applications.

2 Parametrization

Before we describe our wavelet transform, we need to define a parametrization for bilinear subdivision generalized to tesselations composed of convex polygonal regions. Such a tesselation is defined by sets of vertices V, edges E, and convex polygons (faces) F that completely cover an *image domain* I, for example, the unit square $I = [0,1] \times [0,1]$, without overlapping each other. It is assumed that the *valence* of every vertex, *i.e.*, the number of in-

[1] Center for Applied Scientific Computing (CASC), Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, U.S.A.

[2] Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California at Davis, Davis, CA 95616-8562, U.S.A.

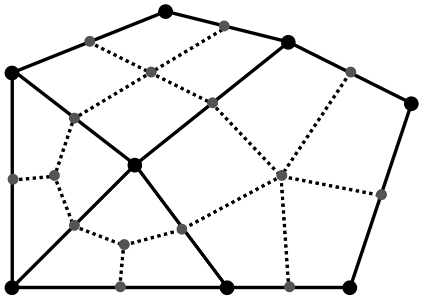

Figure 1: Subdivision process.

cident edges, is at least three, except for boundary vertices. The edges are line segments between two vertices that encompass all convex combinations of these, except for the vertices themselves. The faces are the remaining open regions enclosed by vertices and edges. It is assumed that every angle between two incident edges enclosing a face is strictly less than 180 degrees, *i.e.*, all faces must be convex and *T-nodes* are not allowed.

Consider an initial tesselation

$$T_0 \ = \ \{V_0, E_0, F_0\}. \tag{2.1}$$

We now define recursive refinement rules for a sequence of tesselations $T_1, T_2, \ldots$ The set V_{i+1} is composed of V_i, the midpoints of all edges in E_i, and the centroids of all faces in F_i. The set E_{i+1} is composed of four edges incident to every midpoint of an edge in E_i connecting this midpoint to the two adjacent vertices in V_i and the centroids of the two adjacent Faces in F_i. The midpoints of boundary edges have only three incident edges, since they have only one adjacent face in F_i. The set F_{i+1} contains the remaining open surface regions in the image plane. We note that all polygons in F_{i+1} are convex quadrilaterals, provided that all polygons in F_i are convex, see Figure 1.

We define a local parametrization for the quadrilaterals in F_1. This parametrization can be used for all subsequent levels of subdivision, since the quadrilaterals are uniformly subdivided. For every face $f_k \in F_1$, defined by points $\mathbf{p}_{k,00}, \mathbf{p}_{k,10}, \mathbf{p}_{k,11}, \mathbf{p}_{k,01} \in V_1$ in counterclockwise order, we define local coordinates $u, v \in [0,1]$ so that every point $\mathbf{p}$ in the closure of f_k has the representation

$$\begin{aligned} \mathbf{p} \ &= (1-u)(1-v)\mathbf{p}_{k,00} \ + \ u(1-v)\mathbf{p}_{k,10} \\ &\quad + \ (1-u)v\mathbf{p}_{k,01} \ + \ uv\mathbf{p}_{k,11}. \end{aligned} \tag{2.2}$$

It is more difficult to compute the local coordinates u and v for a point $\mathbf{p} = (x,y)$ in the global image domain I. We compute these local coordinates in this way: first, the face index k needs to be determined (k is not uniquely defined if $\mathbf{p}$ lies on an edge or vertex); second, we compute u from the z-component of

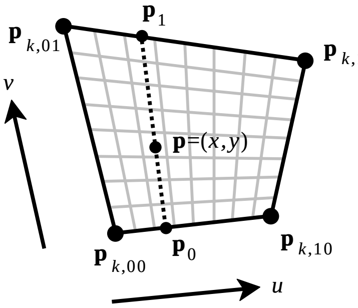

Figure 2: Computing local parameters u and v for point $\mathbf{p}$.

$$\begin{aligned} (\mathbf{p}-\mathbf{p}_0) \times (\mathbf{p}_1 - \mathbf{p}_0) \ &= \ 0, \quad \text{where} \\ \mathbf{p}_0 \ &= \ (1-u)\mathbf{p}_{k,00} \ + \ u\mathbf{p}_{k,10} \quad \text{and} \\ \mathbf{p}_1 \ &= \ (1-u)\mathbf{p}_{k,01} \ + \ u\mathbf{p}_{k,11}. \end{aligned} \tag{2.3}$$

Equation (2.3) is a quadratic equation that has a unique solution $u \in [0,1]$, provided that the quadrilateral f_k is convex. Once u is determined, v can be computed from

$$\mathbf{p} \ = \ (1-v)\mathbf{p}_0 + v\mathbf{p}_1. \tag{2.4}$$

This process is illustrated in Figure 2. We note that computing the local parameters u and v from the global ones is expensive, due to the evaluation of a square root for solving the quadratic equation. We therefore avoid using global parameters in our signal processing algorithm. For resampling of an image represented by a fine tesselation T_j, we suggest to use graphics hardware to render the quadrilaterals of T_1, with texture representing all finer subdivision levels.

3 Wavelet Construction

In the previous section, we have defined a parametrization of the image plane based on recursive subdivision of a tesselation into quadrilaterals. The quadrilateral regions resulting from the first subdivision are refined by recursively cutting the parameter intervals for u and v into halfs. In the following, we construct a lifted bilinear wavelet transform based on this parametrization. A more detailed description of the bilinear wavelet construction, and also of bicubic and biquintic equivalents, can be found in [1].

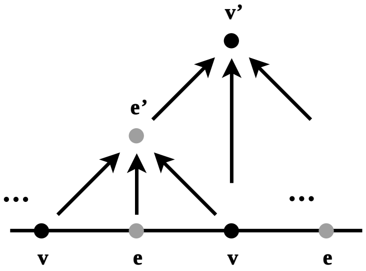

Figure 3: Lifting scheme for one-dimensional decomposition.

The images that we intend to represent in a wavelet basis can be considered as continuous, bilinearly blended functions $f(I)$, mapping the image domain into a vector space containing all possible colors. The dimension of this vector space depends on the application and the color model used, *e.g.*, one dimension for grey-scale images, three for RGB images. When using the wavelet transform in rendering applications, additional coordinates for opacity and z-buffering can be added. In radiosity applications, the domain I represents all surface components in a scene and f describes the local radiance emanating from them. Since our wavelet transform is applied independently yet in the same way to all coordinates, the number of dimensions does not have an impact.

3.1 One-Dimensional Wavelets

Given a piecewise linear function defined by a list of control points, the one-dimensional wavelet transform eliminates every second point and thus provides a coarser representation of this function. The eliminated points are replaced by accumulated differences from which the function at its original resolution can be reconstructed without loss. The coarsening can be interpreted as low-pass filtering and computing the accumulated differences is equivalent to band-pass filtering, since details of a certain frequency band are separated. The entire process is called *decomposition* and is recursively applied to the coarsest approximation of the function until a base resolution is reached. Assuming that the computation time for every decomposition step is linear in the number of transformed control points, the computation time of the entire transform is $O(n + \frac{1}{2}n + \frac{1}{4}n + \cdots) = O(n)$.

Our wavelet transform is based on the lifting scheme [21]. Lifting operations are used to design wavelets with certain properties, like vanishing moments, and to subdivide the computation into small local steps, called lifting operations. Our wavelet transform is based on linear B-spline subdivision [9]. Every decomposition step is computed by two *lifting* operations, implemented as follows: we label every second control point with $\mathbf{e}$ and the remaining ones with $\mathbf{v}$. The $\mathbf{v}$ points correspond to vertices in the next coarser level, and the $\mathbf{e}$ points, located on edges, are replaced by accumulated differences. The computation of the new values is defined by the two lifting operations

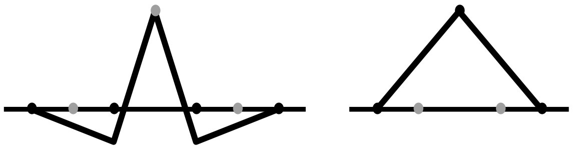

Figure 4: Wavelet and B-spline scaling function.

$$\begin{aligned} \mathbf{e}' &= \mathbf{e} - \overline{\mathbf{v}}_{\mathbf{e}} \quad \text{and} \\ \mathbf{v}' &= \mathbf{v} + \tfrac{1}{2}\overline{\mathbf{e}'}_{\mathbf{v}}, \end{aligned} \tag{3.1}$$

where the operator $\overline{x}_y$ returns, for every point of type y, the arithmetic average of all adjacent points of type x. The decomposition is thus implemented by subtracting from every $\mathbf{e}$ point the average of its two $\mathbf{v}$ neighbors and then adding to every $\mathbf{v}$ point one half of the average of its (modified) $\mathbf{e}'$ neighbors. This decomposition step is illustrated in Figure 3. Decomposition is recursively applied using the $\mathbf{v}'$ points from the previous step as input.

The inverse of a decomposition step is called *reconstruction*, and is defined by the lifting operations

$$\begin{aligned} \mathbf{v} &= \mathbf{v}' - \tfrac{1}{2}\overline{\mathbf{e}'}_{\mathbf{v}'} \quad \text{and} \\ \mathbf{e} &= \mathbf{e}' + \overline{\mathbf{v}}_{\mathbf{e}'}. \end{aligned} \tag{3.2}$$

Reconstruction is recursively applied starting with the coarsest representation (base level) and recursively reproducing the finer approximation levels. Assuming zero $\mathbf{e}'$ points on every level, the reconstruction formula becomes a subdivision scheme that converges to a continuous curve when applied *ad infinitum.* For our construction, the subdivision rules reproduce linear B-splines [9]. The basis functions of the wavelet transform can be visualized by pulling a control point and by recursively applying the reconstruction formula. The $\mathbf{e}'$ points correspond to wavelets and $\mathbf{v}'$ points to *scaling functions* (linear B-splines for our construction), see Figure 4. The wavelet has two vanishing moments, since it has zero *direct current* and is symmetric [1].

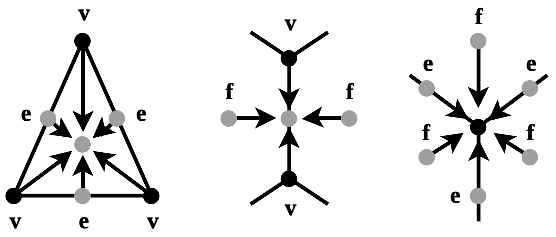

Figure 5: Lifting operations for two-dimensional decomposition.

3.2 Wavelets on Tesselations

Given a hierarchy of tesselations, as defined in Section 1, a wavelet transform can be defined based on the mesh hierarchy. Initially, a function is represented by a fine tesselation T_j and by a sample of the image $f(I)$ at every vertex in V_j. These samples are denoted as $\mathbf{v}$, $\mathbf{e}$, and $\mathbf{f}$ points, depending on the types of their associated vertices in V_j that correspond to vertices, edges and faces, respectively, in the next coarser tesselation T_{j-1}. A decomposition step transforms $\mathbf{v}$ points into control points $\mathbf{v}'$ on the next coarser level, $j-1$, and the $\mathbf{v}$ and $\mathbf{e}$ points become wavelet coefficients representing the missing detail. Using only the $\mathbf{v}'$ points as samples associated with the vertices in V_{j-1}, decomposition is recursively applied to all levels until base level $j=0$ is reached.

The decomposition formula is defined by the three lifting steps

$$\begin{aligned} \mathbf{f}' &= \mathbf{f} + \overline{\mathbf{v}}_{\mathbf{f}} - 2\overline{\mathbf{e}}_{\mathbf{f}}, \\ \mathbf{e}' &= \mathbf{e} - \overline{\mathbf{v}}_{\mathbf{e}} + \tfrac{1}{2}\overline{\mathbf{f}'}_{\mathbf{e}}, \quad \text{and} \\ \mathbf{v}' &= \mathbf{v} - \tfrac{1}{4}\overline{\mathbf{f}'}_{\mathbf{v}} + \overline{\mathbf{e}'}_{\mathbf{v}}. \end{aligned} \tag{3.3}$$

These operations are illustrated in Figure 5. It can be shown that this decomposition formula is identical to a tensor-product construction of equation (3.1) on a uniform rectilinear grid [1]. The scaling functions for this transform are thus bilinear B-splines in the regular regions of the tesselation. Some basis functions obtained from this transform are visualized in Figure 6.

The reconstruction formula is the inverse of every lifting operation, applied in reverse order. It is defined by

$$\begin{aligned} \mathbf{v} &= \mathbf{v}' + \tfrac{1}{4}\overline{\mathbf{f}'}_{\mathbf{v}'} - \overline{\mathbf{e}'}_{\mathbf{v}'}, \\ \mathbf{e} &= \mathbf{e}' + \overline{\mathbf{v}}_{\mathbf{e}'} - \tfrac{1}{2}\overline{\mathbf{f}'}_{\mathbf{e}'}, \quad \text{and} \\ \mathbf{f} &= \mathbf{f}' - \overline{\mathbf{v}}_{\mathbf{f}'} + 2\overline{\mathbf{e}}_{\mathbf{f}'}. \end{aligned} \tag{3.4}$$

To represent boundary edges properly and to model discontinuities along certain edges within a tesselation (which requires a double set of coefficients along these discontinuities), all vertices located on such boundary edges are transformed by the one-dimensional decomposition and reconstruction rules, instead. Corner vertices of the tesselation are not modified by any rule and thus contain the same color information through all levels of resolution.

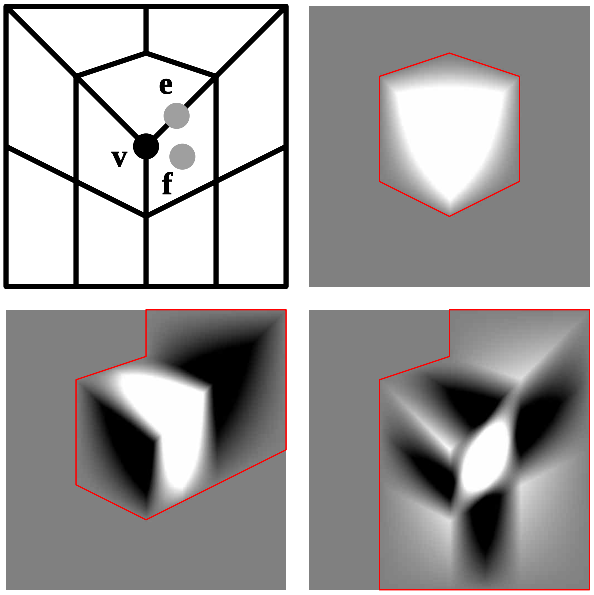

Figure 6: Basis functions near an *extraordinary vertex* of valence three. Scaling function (top right), $\mathbf{e}$ wavelet (bottom left), and $\mathbf{f}$ wavelet (bottom right). Dark regions correspond to negative and bright regions to positive function values.

The computation time for one decomposition step is linear in the number of transformed vertices. Since only one quarter of these vertices is transformed again on the next coarser level, the computation time for the wavelet transform, starting with n samples, is $O(n+\frac{1}{4}n+\frac{1}{16}n+\cdots) = O(n)$. Applications for our wavelet transform are outlined in the next section.

4 Applications

To demonstrate that our wavelet transform works correctly, we transformed the "Cygnus Loop" Hubble telescope image, courtesy of NASA. The image was resampled using a tesselation T_0 composed of 20 vertices and 14 faces. The finest tesselation T_7 has 217921 vertices, which is about 3.5 times the number of pixels of the original image (249 × 251). Figure 7 shows the resampled image, the tesselation, our wavelet transform and reconstructions using 10, 1, and 0.1 percent of the wavelet coefficients. Since the coefficients are vector-valued (RGB values), we use the Euclidean length for tresholding. For progressive transmission the coefficients can be sorted by decreasing absolute values in expected linear time by using a hash table. For lossless compression, our wavelet transform can be implemented in integer arithmetic [1, 4]. The integer-valued coefficients have low expected modulus and can be compressed by arithmetic coding [15].

Some of the most important applications of our wavelet transform are:

- Video compression. Video data can be compressed in a straight-forward way by applying a trivariate tensor product wavelet transform. Correlation in time, however, can be much better exploited when applying a wavelet transform defined on a grid that is deformed or moves with the objects. Our wavelet transform has therefore a great potential to improve compression rates for moving images.

- Image Morphing. Morphing algorithms typically deform objects and compute the intermediate image regions by blending two images. This blending operation can be improved by performing it in the range of the wavelet transform. This treatment would blend the individual frequency bands rather than pixel values resulting in a much more realistic image. Our wavelet transform can be applied to polygonal regions that are deformed simultaneoulsy with the blending process.

- Radiosity. Occlusion of objects causes radiance functions to be discontinuous on smooth surfaces. These discontinuities define tesselations on the surface regions that provide a natural parametrization for the radiance function. Since our wavelets have two vanishing moments (in local parameters, except at extraordinary points), they form an ideal basis for efficient and stable integration of radiosity kernels.

- Scattered Data Approximation. Starting with a tesselation of selected scattered data points, our efficient wavelet transform can be used for further regular refinement and higher detailed approximation. Our wavelet basis functions are piecewise bilinear in local parameters and can be computed analytically for fitting purposes. To construct initial tesselations at certain levels of resolution, we can use *hierarchical Voronoi diagrams* by Schussman *et al.* [18].

5 Conclusions

We have presented a technique for multiresolution analysis of functions defined on planar tesselations. Our wavelet transform has a wide range of applications that still need to be explored. Future work will be directed at constructing wavelets and subdivision surfaces of higher degree of continuity, *e.g.*, tangent-plane continuous representations, with local parametrizations that can be evaluated in a closed form.

6 Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. This work was also supported by the National Science Foundation under contracts ACI 9624034 and ACI 9983641 (CAREER Awards), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the Office of Naval Research under contract N00014-97-1-0222; the Army Research Office under contract ARO 36598-MA-RIP; the NASA Ames Research Center through an NRA award under contract NAG2-1216; the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628 awarded to the University of California, Davis. We also acknowledge the support of ALSTOM Schilling Robotics, Chevron, Silicon Graphics, Inc. and ST Microelectronics, Inc. We thank the members of the Visualization Thrust at CIPIC at the University of California, Davis and the members of the Data Exploration Thrust at CASC at the Lawrence Livermore National Laboratory.

References

[1] M. Bertram, M.A. Duchaineau, B. Hamann, and K.I. Joy, *Generalized B-spline subdivision surface wavelets and lossless compression*, submitted to Computer Aided Geometric Design, 2000.

[2] M. Bertram, M.A. Duchaineau, B. Hamann, and K.I. Joy, *Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization*, submitted to IEEE Visualization, Oct. 2000.

[3] G.-P. Bonneau, *Optimal triangular Haar bases for spherical data*, Proceedings of Visualization '99, IEEE Computer Society Press, Los Alamitos, California, 1999, pp. 279–284 & 534.

[4] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, *Wavelet transforms that map integers to integers*, Applied and Computational Harmonic Analysis, Vol. 5, No. 3, Academic Press, July 1998, pp. 332–369.

[5] E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, Computer-Aided Design, Vol. 10, No. 6, Nov. 1978, pp. 350–355.

[6] C.K. Chui, *An introduction to wavelets*, Academic Press, San Diego, California, 1992.

[7] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens *Wavelets on irregular point sets*, Phil. Trans. R. Soc. Lond. A, to appear.

[8] D. Doo and M. Sabin, *Behaviour of recursive division surfaces near extraordinary points*, Computer-Aided Design, Vol. 10, No. 6, Nov. 1978, pp. 356–360.

[9] M.A. Duchaineau, *Dyadic splines*, Ph.D. thesis, Department of Computer Science, University of California, Davis, 1996. `http://graphics.cs.ucdavis.edu/~duchaine/dyadic.html`

[10] S.J. Gortler, P. Schröder, M.F. Cohen, P. Hanrahan, *Wavelet radiosity* Computer Graphics (Proceedings of SIGGRAPH '93), Association for Computing Machinery (ACM), 1993, pp. 221–230.

[11] I. Guskov, W. Sweldens, and P. Schröder, *Multiresolution signal processing for meshes*, Computer Graphics (Proceedings of SIGGRAPH '99), Association for Computing Machinery (ACM), 1999, pp. 325–334.

[12] J.M. Lounsbery, *Multiresolution analysis for surfaces of arbitrary topological type*, Ph.D. thesis, Department of Mathematics, University of Washington, 1994.

[13] M. Lounsbery, T.D. DeRose, and J. Warren, *Multiresolution analysis for surfaces of arbitrary topological type*, ACM Transactions on Graphics, Vol. 16, No. 1, ACM, Jan. 1997, pp. 34–73.

[14] C.T. Loop, *Smooth subdivision surfaces based on triangles*, M.S. thesis, Department of Mathematics, University of Utah, 1987.

[15] A. Moffat, R.M. Neal, and I.H. Witten, *Arithmetic coding revisited*, Association for Computing Machinery (ACM) Transactions on Information Systems, Vol. 16, No. 3, July 1998, pp. 256–294.

[16] G.M. Nielson, I.-H. Jung, and J. Sung, *Haar wavelets over triangular domains with applications to multiresolution models for flow over a sphere*, Proceedings of Visualization '97, IEEE Computer Society Press, Los Alamitos, California, 1997, pp. 143–149 & 536.

[17] P. Schröder and W. Sweldens, *Spherical wavelets: efficiently representing functions on the sphere*, Computer Graphics (Proceedings of SIGGRAPH '95 Proc), Association for Computing Machinery (ACM), 1995, pp. 161–172.

[18] S.E. Schussman, M. Bertram, B. Hamann, and K.I. Joy, *Hierarchical data representations based on planar Voronoi diagrams*, in R. van Liere, I. Hermann, and W. Ribarsky, eds., Proceedings of VisSym '00, Joint Eurographics and IEEE TCVG Conference on Visualization, Amsterdam, The Netherlands, May 2000.

[19] J. Stam, *Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values*, Computer Graphics (Proceedings of SIGGRAPH '98 Proc), Association for Computing Machinery (ACM), 1998, pp. 395–404.

[20] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin, *Wavelets for computer graphics–theory and applications*, Morgan Kaufmann Publishers, Inc., San Francisco, California, 1996.

[21] W. Sweldens, *The lifting scheme: a custom-design construction of biorthogonal wavelets*, Applied and Computational Harmonic Analysis, Vol. 3, No. 2, pp. 186–200, 1996.

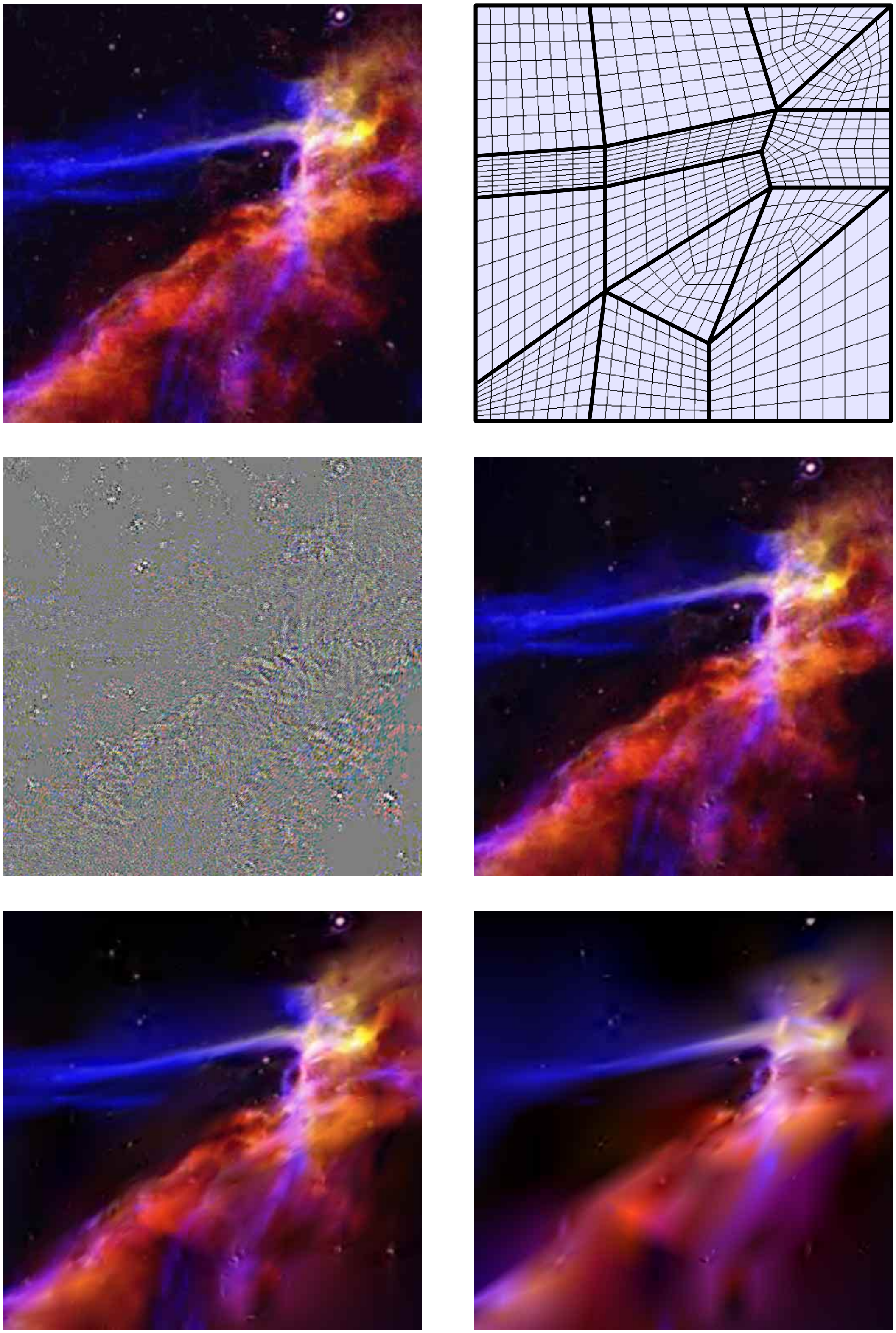

Figure 7: "Cygnus Loop" Hubble image. Top left: original, resampled image on tesselation T_7 (217921 samples); top right: tesselation T_0 and three levels of subdivision; middle left: our wavelet transform (the coefficients are scaled by 10 and a grey level is added); middle right: reconstruction from 10 percent of coefficients; bottom left: 1 percent of coefficients; bottom right: 0.1 percent of coefficients.