Multiscale Implicit Models

LUIZ VELHO[†‡]
DEMETRI TERZOPOULOS[‡]
JONAS DE MIRANDA GOMES[†]

[†] IMPA – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110, Rio de Janeiro, Brazil, 22460

[‡] University of Toronto – Department of Computer Science
Toronto, Ontario, M5S-1A4

Abstract. In this paper we introduce a new method for the piecewise representation of implicit objects based on multiscale decomposition of the implicit function. We define a model that is adapted to the variations of the implicit function at multiple scales. This model is described in terms of B-spline functions which are suitable for geometric computations. We construct a representation that is structured and hierarchical. This description combines the best features of point skeletons and volume arrays. The multiscale decomposition model has several applications in computer graphics, such as variable level of detail models, the conversion from volumetric to implicit representations, rendering of implicit surfaces using ray tracing, and volume visualization.

1 Introduction

Implicit models have been used since the early days of computer graphics (Goldstein and Malin, 1979). Until recently, they were employed mostly as an auxiliary representation for classes of objects that allowed both parametric and implicit descriptions. Typical examples are the quadrics and superquadrics. More powerful models are piecewise descriptions developed as a generalization of implicit algebraic models. The implicit function is a density field constructed as a distance function from a skeleton.

Piecewise models are effective because they are based on simple elements and powerful combination rules. Complex objects can be described by their components in a natural way. They can be simple and yet retain descriptive power.

There are two types of piecewise models: constructive and decomposition. A constructive model is generated by assembling primitive blocks. A decomposition model is generated by subdividing composite structures. Constructive techniques are most appropriate for shape design while decomposition techniques are better suited to shape computation.

Most of the existing piecewise implicit models are constructive. Some examples are CSG models, blobby models, metaballs and soft objects.

In this paper we investigate a new implicit model based on functional decomposition. This type of model decomposes the implicit function into simpler functions, and has many advantages over previous models. It is general, computationally efficient and establishes a direct connection between implicit and volume representations.

Functional decomposition models should reveal relevant aspects of the implicit function, such as its variations. It is also desirable that such a functional decomposition result in a representation that exhibits spatial locatity (an exception to this rule is the case where certain operations are performed more efficiently using some special representation without this property (Totsuka and Levoy, 1993)).

The Fourier transform describes the spectral behavior of a function, discriminating its frequency content. The Fourier series of periodic functions gives the best localization in terms of frequency, but its basis functions have infinite support. Good localization in both space and frequency is achieved by the "short time Fourier transform" (Gabor, 1946), which employs a Gaussian windowed sinusoidal function or by the wavelet transform that uses scaled "small waves" (Daubechies, 1992). The Gabor and wavelet representations are localized and very effective in describing the variations of a function. This makes them attractive from the standpoint of image analysis tasks, such as edge detection.

While these decompositions are suitable representations for image processing, they are not so adequate for computer graphics. From the point of view of graphics applications a representation in terms of the wavelet coefficients presents some problems. The Gabor and wavelet functions must oscillate in order to capture frequency information and, for this reason, can be complicated. Moreover, in higher dimensions many wavelet basis functions are required.

For image synthesis, it is desirable to have a multiscale representation that is constructive, compact, and expressible in terms of a simple function. Intuitively, we want a description of a function as a summation of "blobs" of different sizes.

This paper introduces a functional decomposition based on the scaling function associated with a dyadic wavelet. Such a description fulfills all the above requirements and is adequate for modeling with implicit surfaces.

2 Previous Work

Previous piecewise implicit descriptions include point skeletons and wavelet based models. In this section we review previous related work and compare it with our results.

2.1 Point Skeletons

Primitive implicit objects can be defined by a distance function from lower dimensional geometric elements, such as a point or a line. These elements are called *skeletons*. The implicit function is expressed as $f(x, y, z) - c$, where f is some pseudo-distance function from the skeleton.

Blinn was the first to introduce skeleton based implicit models in computer graphics (Blinn, 1982). His model was inspired by electron density maps. It uses a point skeleton and the distance function is a Gaussian centered at each point of the skeleton. Blinn's idea was further developed by (Nishimura et al., 1985) and (Wyvill, McPheeters and Wyvill, 1986). One important aspect of these latter models is that they use functions that drop to zero at a certain distance from the skeleton.

The basic skeleton models consist of points with simple blending functions, making them attractive computationally. In an attempt to develop better mechanisms for shape design, the complexity of the skeleton models was increased, rendering them less efficient computationally.

The multiscale decomposition model developed in this paper can be considered a point skeleton model that is structured and hierarchical. These two characteristics make it computationally efficient. The decomposition mechanism makes it expressive.

2.2 Wavelet Based Models

The wavelet transform has been used mainly in image processing applications, such as coding and compression (DeVore, Jawerth and Lucier, 1992). We employ a multiscale representation which is essentially equivalent to the Laplacian pyramid (Burt, 1983), but we apply it in a very different context. In image processing, this type of description is used primarily for analysis, while in computer graphics this description is used for synthesis. Consequently, the problems which need to be solved are of a different nature.

Wavelet models of implicit surfaces and solids were proposed in (Muraki, 1993) and (Perlin and Zhu, 1990). Muraki's model uses a 3D orthogonal wavelet. Because it is based on a tensor product, the representation is given in terms of 7 different wavelets, which complicates the computations with the model. Perlin's model uses only one directional spline wavelet, avoiding this problem. His method employs an empirical procedure to construct the representation, and the model is restricted to a particular level surface.

Our multiscale implicit model is based on the scaling function associated with a wavelet decomposition. It gives a description of the implicit function as a whole in terms of a B-spline function. For this reason, it is more adequate for image synthesis computations than the wavelet models. Furthermore, this representation can be computed with a fast and exact algorithm developed from a solid theoretical foundation.

3 Multiscale Decompositions

The decomposition of a function in terms of elements at multiple scales provides a representation that reflects the function behavior over neighborhoods of variable size. It is a hierarchical structure which is adapted to the function variations.

3.1 Definition

A multiscale decomposition is a description based on a set of functions localized both in space and scale. Under this representation a function f is expressed as a linear expansion over this set.

A suitable family of functions for this purpose can be defined by scaling and translating a single function $\phi(x) \in L^2(\mathbb{R})$. A member of this family is denoted by ϕ_γ, where the index $\gamma = (s, t)$ specifies the scaling parameter s and the translation parameter t. Here, we assume that ϕ is a scaling function, i.e. it is a function whose shifts and dilations generate a nested sequence of approximating spaces V_j, such that their union is dense in $L^2(\mathbb{R})$ and their intersection is the null space.

The parameters s, t are restricted to dyadic scales $s = 2^{-j}$, $j \in \mathbb{Z}$ and integer translations

$t = 2^{-j}k$, $k \in \mathbb{Z}$. Thus, the family ϕ_γ is

$$\phi_\gamma(x) = \frac{1}{\sqrt{2^j}}\phi(\frac{x}{2^j} - k).$$

The multiscale decomposition of a function f is a linear expansion into a countable subset of elements $(\phi_{\gamma_i}(x))_{i\in\mathbb{N}}$, with $\gamma_i = (j,k)$, from the family ϕ_{γ_i}, such that

$$f(x) = \sum_{i=0}^{\infty} a_i \phi_{\gamma_i}(x),$$

where γ_i is an element of the index set $\Gamma = \mathbb{Z}^+ \times \mathbb{Z}$.

The family of functions ϕ_γ plays the role of a dictionary $\mathcal{D} = (\phi_\gamma)_{\gamma\in\Gamma}$, from which a subset of elements ϕ_{γ_i} is selected for a multiscale representation (Mallat and Zhang, 1993).

The coefficients a_i are computed by orthogonal projection on the duals of the selected vectors ϕ_{γ_i} of the dictionary $\mathcal{D}$.

$$a_i =< f, \tilde{\phi}_{\gamma_i} >$$

Intuitively, this decomposition indicates the features of f that "belong" to each scale 2^j.

A multiscale representation is given by the list of coefficients a_i, together with the corresponding indices $\gamma_i = (j_i, k_i)$ in the dictionary $\mathcal{D}$.

In practice, we use only a finite number of elements from $\mathcal{D}$. For this reason, it is important to obtain a sequence of decompositions with increasing number of elements which converges to the function f; i.e.,

$$||f - \sum_{i=0}^{m} a_i \phi_{\gamma_i}|| \leq \epsilon ||f||$$

This provides a mechanism to approximate f with the desired precision ϵ.

The definition of a multiscale decomposition for n-dimensional functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ in $L^2(\mathbb{R}^n)$ is a direct extension of the scheme above.

The scaling function ϕ is associated with a multiresolution analysis of $L^2(\mathbb{R})$ consisting of a sequence of nested spaces V_j

$$\cdots V_{-1} \subset V_0 \subset V_1 \cdots$$

The collection of functions $\{\phi_{0;k} : k \in \mathbb{Z}\}$ forms a Riesz basis of the approximation space V_0 and all spaces in the sequence are scaled versions of this reference space V_0 (i.e. each space V_j has a "natural" scale 2^j).

Note that, unlike the above scheme, in the framework of multiresolution analysis, a function f is represented by its approximations simultaneously at all these scales (Rosenfeld, 1984). Such a description is composed of several versions of the function f that are computed by projecting f onto the spaces V_j.

3.2 Existence and Analysis

The existence of a multiscale decomposition is simple to show. Denote by $P_j : L^2(\mathbb{R}) \rightarrow V_j$ the orthogonal projection onto V_j. The properties of the hierarchy of spaces V_j guarantee that:

1. $\lim_{j\to\infty} P_j f = f$ for all $f \in L^2(\mathbb{R})$, since $\overline{\cup V_j} = L^2(\mathbb{R})$
2. if $f \in V_l$ then f is also in V_j, for all $j > l$.

Condition (1) ensures that every function has a trivial decomposition. Condition (2) implies that a function $f \in V_l$ can be expressed in the basis of any one of the approximation spaces V_j, $j > l$, which contain V_l. It is just the projection on V_j, $f = P_j f$ (there is no loss of information in this projection if $j > l$).

The dictionary $\mathcal{D}$ is very redundant. This is evident from the nested structure of the spaces V_j. A direct implication of the redundancy of $\mathcal{D}$ is that the multiscale decomposition is not unique in general. It is easy to see that any function $f \in L^2(\mathbb{R})$ has infinitely many multiscale representations (by simply adding and subtracting different projections, $P_m g$ and $P_n g$ of the same function $g \in V_l$, with $l < m, n$).

If a function $f \in V_0$ has different components g_j in the spaces V_j, $j \geq 0$, then f can be written as as a linear combination of these functions at scales 2^j

$$f = \sum_j g_j = \sum_j \sum_i a_{i,j} \phi_{2^j,k_i}(x).$$

The problem now is: 1) to construct a multiscale decomposition such that $\sum_j g_j$ generates a suitable representation of f; and 2) to compute the coefficients of $g_j = \sum_i a_{i,j}\phi_{2^j,k_i}$.

In the next section we describe in detail a strategy for generating a multiscale decomposition, as well as a computational method based on the wavelet transform. We subdivide the approximation spaces V_j into wavelet spaces W_j creating a partition of $L^2(\mathbb{R})$. This scheme is very general and has the advantage of being independent of the function f.

4 A Wavelet Based Method

Orthogonal wavelets generate a direct sum decomposition of $L^2(\mathbb{R})$ and are intimately related to multiresolution analysis. They provide a rigorous mathematical framework that can be used to derive multiscale decomposition methods.

4.1 The Wavelet Transform

The wavelet spaces W_j can be defined as the difference spaces between two consecutive approximation spaces V_{j+1} and V_j of a multiresolution analysis. The W_j is the orthogonal complement of V_j in V_{j+1}

$$W_j \oplus V_j = V_{j+1}, \qquad W_j \perp V_j.$$

$L^2(\mathbb{R})$ is naturally decomposed into

$$\bigoplus_j W_j = L^2(\mathbb{R}).$$

The connection of wavelets with multiresolution analysis is the key to the efficient computation of the wavelet decomposition of a function (Mallat, 1989).

The wavelet transform decomposes a function $f \in V_j$ recursively through orthogonal projection onto the spaces V_{j-1} and W_{j-1}. In this algorithm, the spaces V_j are just intermediate elements that are used for the recursion step.

4.2 The Laplacian Decomposition

The wavelet transform gives a multiple scale representation of a function. Unfortunately, we cannot use the wavelet description directly as a multiscale decomposition because its coefficients are not relative to a scaling function.

A multiresolution analysis is the projection of a function onto the spaces V_j spawned by a scaling function. It represents approximations of the function f at each resolution 2^j. We also cannot use this description either because, it does not constitute a linear expansion of f.

Instead, we would like to obtain a representation that is multiscale, but given in terms of the scaling function ϕ. Such a representation can be constructed if we combine the wavelet description and the multiresolution analysis.

Observing that $V_j = V_{j-1} \oplus W_{j-1}$ implies that $W_{j-1} \subset V_j$. Therefore, we can represent W_{j-1} in terms of a basis of V_j without any loss of information. For this, we need only project the wavelet components in the subspaces W_{j-1} back to the subspaces V_j. This method produces a multiscale description in terms of the scaling function (this structure is equivalent to the Laplacian pyramid (Burt, 1983)).

4.3 The B-Spline Scaling Basis

So far we have discussed the multiscale decomposition without defining the scaling function associated with it. The B-spline is a scaling function that is suited to our purposes.

The B-spline function has several desirable properties: symmetry, smoothness, compact support, good localization in space and scale, a simple analytical form in both spatial and frequency domains, and efficient implementation (Chui and Wang, 1990).

The only disadvantage is that the B-spline does not generate an orthogonal basis. But it is possible to construct a dual function which together with the B-spline, defines a pair $(\phi, \tilde{\phi})$ that is bi-orthogonal and can be used in the multiscale decomposition method (Cohen, 1992).

Figure 1 shows a plot of the cubic B-spline scaling and wavelet functions.

(a) (b)

Figure 1: Cubic B-spline (a) scaling function (b) wavelet function

5 The Multiscale Representation

The results in the previous section allow us to create an implicit piecewise representation based on a multiscale decomposition of the implicit function f. This representation is adapted to the variations of f at different scales.

As we mentioned in section 3, the multiscale representation contains the coefficients a_i of the multiscale decomposition. The indices $\gamma_i = (j_i, k_i)$ relate these coefficients to elements ϕ_{γ_i} of the dictionary $\mathcal{D}$.

In order to produce the representation, we compute the coefficients a_i of the B-spline multiscale decomposition and convert them to a suitable data structure.

5.1 Properties

The multiscale implicit description is:

- Hierarchical
- Structured

The multiscale representation describes a function f by its components $g_j \in V_j$. This representation is hierarchical because the spaces V_j form a ladder of approximation spaces. It is structured because the basis functions $\phi_{j,k}$ of V_j are located in a regular rectangular grid.

These two properties are the key for efficient computation with piecewise implicit models.

5.2 Computing the Representation

There are two equivalent procedures to compute the multiscale decomposition of a function $f \in V_0$: an extended wavelet transform, and a Laplacian transform.

In both cases, we assume that we have computed the coefficient sequence $\{c\}_0 = < f, \widetilde{\phi}_{0,k} >$ of the representation of f in the basis of V_0, such that f is written as

$$f(x) = \sum_k < f, \widetilde{\phi}_{0,k} > \phi_{0,k}(x).$$

In the extended wavelet transform we apply the wavelet decomposition to the coefficient sequence $\{c\}_0$ obtaining the wavelet coefficients $\{d\}_j$, $j = -1, \ldots, -n$, and then for each scale 2^j, we project the wavelet coefficients $\{d\}_j \in W_j$ back to V_{j+1} by applying the wavelet reconstruction from level j to $j+1$. This procedure is illustrated in the diagram below.

$$\begin{tikzcd}
\{c\}_j \arrow[r, "\widetilde{H}"] \arrow[dr, "\widetilde{G}"] & \{c\}_{j-1} \arrow[r, "\widetilde{H}"] \arrow[dr, "\widetilde{G}"] & \{c\}_{j-2} \arrow[dr, "\widetilde{G}"] & \cdots \\
\{d\}_j & \{d\}_{j-1} \arrow[dl, "G"] & \{d\}_{j-2} \arrow[dl, "G"] & \cdots \arrow[dl, "G"] \\
\{a\}_j & \{a\}_{j-1} & \{a\}_{j-2} & \cdots
\end{tikzcd}$$

where H, G, $\widetilde{H}$ and $\widetilde{G}$ are the pairs of digital filters associated with the B-spline scaling function, the B-spline wavelet and their duals (Cohen, Daubechies and Feauveau, 1991).

In the Laplacian transform, we exploit the fact that, since $W_{j-1} = V_j \ominus V_{j-1}$, the coefficients $\{a_j\}$ of the multiscale decomposition can be computed by subtracting the coefficients of the approximations at V_j and V_{j-1}. This procedure is illustrated in the diagram below.

$$\begin{tikzcd}
\{c\}_j \arrow[r, "\widetilde{H}"] & \{c\}_{j-1} \arrow[r, "\widetilde{H}"] \arrow[dl, "H"] & \{c\}_{j-2} \arrow[dl, "H"] & \cdots \arrow[dl, "H"] \\
- & - & - & \\
\{b\}_j & \{b\}_{j-1} & \{b\}_{j-2} & \cdots \\
= & = & = & \\
\{a\}_j & \{a\}_{j-1} & \{a\}_{j-2} & \cdots
\end{tikzcd}$$

In one dimension, these two procedures require roughly the same amount of computation. In higher dimensions, the Laplacian transform is more efficient.

5.3 Data Structures

We can use three alternative data structures to encode the coefficients of the multiscale decomposition:

- A list of the coefficients $l = \{a_i, \gamma_i\}$.
- A pyramid data structure, $A_j = (a_{j,k})$.
- A spatial hash table $H = [\uparrow a_i, \gamma_i]$.

The list structure is simply an enumeration of the coefficients of the multiscale decomposition a_i and the indices $\gamma_i = (s_i, u_i)$ corresponding to the functions $\phi_{2^j,k}$ of the dictionary $\mathcal{D}$.

The pyramid structure contains all coefficients associated with the basis functions $\phi_{(2^j,k)}$ of the approximating spaces V_j. This is essentially an enumeration of the coefficients corresponding to all elements in $\mathcal{D}$, such that the functions that are not in γ_i have a coefficient $a_{j,k} = 0$. This eliminates the need for including the indices γ_i in the representation.

The coefficients a_i may also be associated with a spatial hash table in which each cell has pointers to the elements whose support is in the cell. This cell complex can be formed by any adaptive subdivision of space, such as an Octree or a BSP tree (Carvalho, Gomes and Velho, 1992).

In practice, the choice of a particular data structure will be dictated by the characteristics of the implicit function, as well as the requirements of the application.

6 Examples

In this section we give examples of the multiscale description of implicit objects in two and three dimensions.

6.1 2D Example

The implicit object in the following example, is given in the form of 2D sample array. This volumetric representation is converted to the B-spline multiscale representation.

The implicit function is shown as an image, the boundary of the object as a curve in the plane, and the B-spline pyramid is depicted as a set of circles corresponding to the support of B-spline basis functions with non-zero coefficients.

The implicit object is a free-form shape created with an experimental painting program (Perlin and Velho, 1992). Figure 2 shows an image of its density function. Figure 3 shows the B-spline functions in this representation. Note how the B-spline functions at finer scales are concentrated in places where the object presents more detail. Figure 4 shows the curve generated from the multiscale representation.

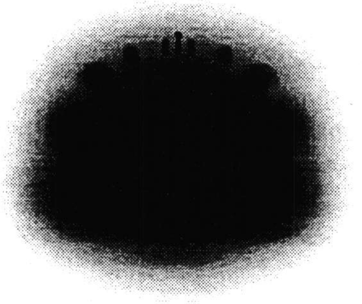

Figure 2: Implicit Shape Function

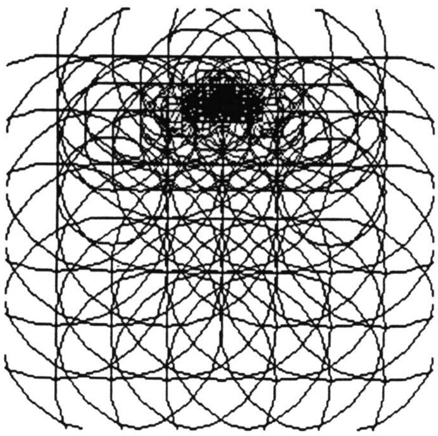

Figure 3: B-spline Representation

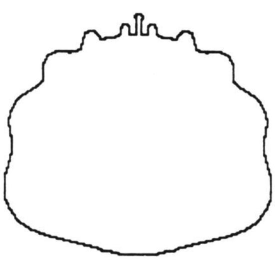

Figure 4: Implicit Boundary Curve

6.2 3D Example

The next example shows the use of the method in three dimensions. The input is a 3D sample array that is converted to the B-spline multiscale representation.

The implicit object is the "noisy sphere", a procedural implicit shape defined by functional composition of an object density function with density modulation functions (Perlin and Hoffert, 1989). In this example, the object density function is of a soft sphere and the modulation function is a bandlimited noise function.

Figure 5 shows the volume density array generated by the hypertexture procedure mentioned above.

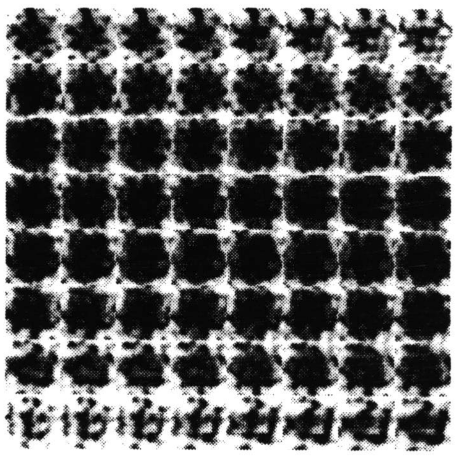

Figure 5: Slices of the Volume Density Function for the Noisy Sphere

Figure 6 shows one slice of each level of the B-spline pyramid.

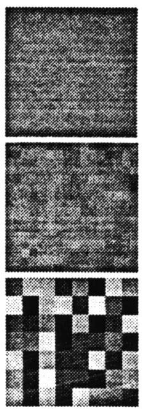

Figure 6: One Slice of Each Level of the B-spline Pyramid

In this figure, the coefficients are depicted according to the following convention: zero is middle gray, negative is darker and positive is lighter. Note that almost all the information is contained in the bottom level of the pyramid, indicating that most of variations of the implicit function are at that scale.

Figure 7 is a ray-traced image of the noisy sphere. It was produced by rendering the B-spline pyramid description of the data in Figure 5.

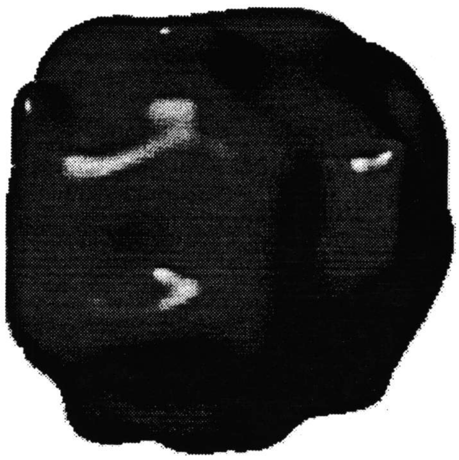

Figure 7: Noisy Sphere, Raytraced from its B-spline Pyramid

7 Applications

The multiscale decomposition has many applications in modeling and graphics. It has the potential to be the basis for the development of efficient computational methods in those areas.

Applications of the B-spline multiscale decomposition include the conversion of volumetric data to implicit analytic representations, variable level of detail models, ray tracing of implicit surfaces and volume rendering.

7.1 Conversion of Volumetric to Implicit Descriptions

The multiscale decomposition can be applied to the conversion of volumetric to implicit descriptions. The method takes as input a n-dimensional array of discrete samples and generates a piecewise implicit representation in terms of the B-spline scaling basis functions.

7.2 Variable Level of Detail Models

The multiscale representation describes the model by its components at different scales. This makes it particularly effective to represent implicit objects at variable level of detail.

7.3 Ray Tracing

The multiscale representation is suitable for ray casting techniques. There are several ray-tracing algorithms for implicit surfaces given by polynomial functions that could be used to compute the ray intersection with the B-spline primitives (Tonnesen, 1989), (Wyvill and Trotman, 1990). Similarly, the cone tracing algorithm (Amanatides, 1984) can be adapted to handle "fuzzy balls" producing antialiased images as well as soft shadows (Perlin and Zhu, 1990).

7.4 Volume Rendering

The multiscale representation is also well suited to volumetric rendering techniques. Direct projection methods, such as splatting (Laur and Hanrahan, 1991), and volume integration methods, such as ray marching, (Levoy, 1990), can exploit the multiscale representation in various ways.

8 Conclusions

In this paper we defined the notion of multiscale decomposition models and discussed the criteria to analyze such models. We identified the strategies to generate a multiscale decomposition and investigated a wavelet based method for producing this kind of description. We presented algorithms to compute the biorthogonal B-spline multiscale description, investigated its properties, and described alternative data structures for this representation.

The multiscale decomposition model has several applications in the area of computer graphics. Some of these applications are the conversion from volumetric to piecewise implicit representations, variable level of detail models, rendering of implicit surfaces using ray tracing, and volume visualization.

9 References

Amanatides, J. (1984). Ray tracing with cones. *Computer Graphics*, 18(3):129–136.

Blinn, J. F. (1982). A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256.

Burt, P. J. (1983). The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540.

Carvalho, P. C., Gomes, J. M., and Velho, L. (1992). Space decompositions: Theory and practice. *IMPA (preprint).*

Chui, C. and Wang, J. (1990). A cardinal spline approach. Technical Report CAT 211, Center for Approximation Theory, Department of Mathematics, Texas A&M University.

Cohen, A. (1992). Biorthogonal wavelets. In Chui, C. K., editor, *Wavelets: A Tutorial in Theory and Applications*, pages 123–152. Academic Press.

Cohen, A., Daubechies, I., and Feauveau, J. (1991). Bi-orthogonal bases of compactly supported wavelets. To appear in Comm. Pures and Appl. Math.

Daubechies, I. (1992). *Ten Lectures on Wavelets.* Number 61 in CBMS-NSF Series in Applied Mathematics. SIAM Publications, Philadelphia.

DeVore, R. A., Jawerth, B., and Lucier, B. J. (1992). Image compression through wavelet transform coding. *IEEE Trans. on Inf. Theory*, 38(2):719–746.

Gabor, D. (1946). Theory of communication. *J. Inst. Elec. Eng.*, Vol.93(Part III):429–457.

Goldstein, R. and Malin, L. (1979). 3-d modeling with the synthavision system. In *Proc. First Annl. Conf. Comp. Graphics in CAD/CAM Systems*, pages 244–247.

Laur, D. and Hanrahan, P. (1991). Hierarchical splatting: A progressive refinement algorithm for volume rendering. In Sederberg, T. W., editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 285–288.

Levoy, M. (1990). Efficient ray tracing of volume rendering. *ACM Transactions on Graphics*, 9(3):245–261.

Mallat, S. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. Technical report, Courant Institute, New York University.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 11(7):674–693.

Muraki, S. (1993). Volume data and wavelet transform. *IEEE Computer Graphics and Applications*, 13(4):50–56.

Nishimura, H., Hirai, M., Kawai, T., Kawata, T., Shirakawa, I., and Omura, K. (1985). Object modeling by distribution function and a method of image generation. *Japan Electronics Communication Conference 85*, J68-D(4):718–725.

Perlin, K. and Hoffert, E. (1989). Hypertexture. *Computer Graphics*, 23(3).

Perlin, K. and Velho, L. (1992). A wavelet representation for unbounded resolution painting. Technical report, Courant Institute of Mathematical Sciences, New York University, Dept. of Computer Science.

Perlin, K. and Zhu, B. (1990). Surflets. ACM Siggraph Course Notes. Photorealistic Volume Modeling and Rendering Techniques.

Rosenfeld, A. (1984). *Multiresolution Image Processing and Analysis.* Springer–Verlag, New York.

Tonnesen, D. (1989). Ray tracing implicit surfaces resulting from the summation of bounded polynomial functions. Technical report, Rensselaer Polytechnic Institute.

Totsuka, T. and Levoy, M. (1993). Frequency domain volume rendering. In Kajiya, J. T., editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 271–278.

Wyvill, B., McPheeters, C., and Wyvill, G. (1986). Animating soft objects. *The Visual Computer*, 2(4):235–242.

Wyvill, G. and Trotman, A. (1990). Ray tracing soft objects. ACM Siggraph Course Notes. Modeling and Animating with Implicit Surfaces.