Multiresolution Maximum Intensity Volume Rendering by Morphological Pyramids

Jos B.T.M. Roerdink

Institute for Mathematics and Computing Science
University of Groningen
P.O. Box 800, 9700 AV Groningen, The Netherlands
roe@cs.rug.nl

Abstract We propose a multiresolution representation for maximum intensity projection (MIP) volume rendering, based on morphological pyramids which allow progressive refinement and have the property of perfect reconstruction. The pyramidal analysis and synthesis operators are composed of morphological erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis. The structure of the multiresolution MIP representation is very similar to wavelet splatting, the main differences being that (i) linear summation of voxel values is replaced by maximum computation, and (ii) linear wavelet filters are replaced by (nonlinear) morphological filters.

1 Introduction

Interactive rendering and transfer of volume data is still a demanding problem due to the sizes of the data sets. For this purpose multiresolution models are developed, which can be used to visualize data incrementally (‘progressive refinement’). An extensively studied class of such multiresolution models is based on wavelets [4, 12, 18]. Recent methods for X-ray rendering include wavelet splatting [7, 8], which extends splatting [19] by using wavelets as reconstruction filters, and Fourier-wavelet volume rendering [14, 17], which extends standard Fourier volume rendering [10], and uses a frequency domain implementation of the wavelet transform.

The goal of this paper is to propose a multiresolution representation for Maximum Intensity Projection (MIP) volume rendering, where one computes not the (opacity-weighted) integral, but the *maximum* along the line of sight. Because of its computational simplicity, this algorithm is widely used in the display of magnetic resonance angiography (MRA) and ultrasound data. Our approach makes use of the concept of *morphological pyramids*, following recent work of Goutsias and Heijmans [3, 6], who present a general framework for multiresolution signal decomposition, which includes linear wavelet analysis as a special case. Even though the morphological operators are nonlinear and non-invertible, the pyramid scheme does allow perfect reconstruction as well as progressive refinement, just as in the linear wavelet case. We restrict ourselves here to the so-called *flat* pyramids, where minima and maxima are computed in a local neighbourhood of each voxel, requiring only integer computations. Flatness in particular means that no new grey values are introduced in the analysis of a signal. Also, flat pyramids allow global error control, since they have the property that the approximation error decreases monotonically as we add detail signals. Note also that the morphological pyramids used in this paper are not auxiliary data, but an exact representation of the initial data. After the pyramid has been constructed, the original volume data can be discarded, since the pyramid allows perfect reconstruction of the data.

Morphological methods have a well-established mathematical basis and are widely used in image processing for filtering, segmentation, and shape analysis [5, 15]. Applications of morphological methods in visualization have so far mostly been restricted to preprocessing of volume data, but this is beginning to change. For example, Lürig and Ertl [9] used multiscale morphological operators as an alternative to transfer functions in traditional colour-opacity volume rendering. Visualization of solids defined by morphological operators was considered in [13].

Morphological pyramids are useful in the context of MIP for several reasons. First, from a mathematical point of view, the morphological operations of erosion and dilation (involving minimum and maximum computation) are exactly the right ones for the case of MIP, which involves maximum computation, just as linear wavelet representations are the right tool for the case of linear X-ray rendering. Second, the feature extraction capabilities of morphological operators can be incorporated within the volume rendering process. This allows processing based on geometric information, not just on grey value properties, as usually is the case. For example, when processing angiographic data, the multiresolution scheme will systematically remove small veins when going higher up in the pyramid, while keeping larger ones. Whether or not this is a desired property can only be answered in the context of the concrete medical application. Finally, pyramids are one of many possibilities for accelerating MIP. Many methods already exist for that purpose, including distance encoding [20], splatting in sheared object space [2], or MIP at warp speed [11], which preprocesses the data to remove non-contributing voxels from the volume.

We stress that in this paper the main issue is the presentation of a new multiresolution MIP representation. Computational efficiency is a separate issue: any existing fast MIP implementation can in principle be used for computing the maximum projections which are required to render different levels of the pyramid, as long as such an implementation can work directly on the data structures used to represent the pyramid. In the examples below we will use the voxel projection method of Mroz *et al.* [11]. A detailed study of computational aspects will be presented in future work.

The organization of this paper is as follows. Section 2 gives a few preliminaries on morphological operators, and summarizes the work of Goutsias and Heijmans [3, 6] on morphological pyramids. Section 3 contains the new material, i.e. the derivation of a multiresolution MIP rendering algorithm (MMIP) allowing progressive refinement. An example is given in section 4. Section 5 contains a discussion of future work.

2 Morphological pyramids

Before we consider multiresolution signal decomposition, first some elementary morphological operators are introduced.

Morphological operators Morphological operations for grey value images have been defined in analogy with the binary case [16]. For a mathematical treatment, see e.g. [5]. We consider signals or functions, defined on a subset of the discrete grid $\mathbb{Z}^d$, where $d = 2$ or $d = 3$ (image and volume data).

Let f be a signal with domain $F \subseteq \mathbb{Z}^d$, and A a subset of $\mathbb{Z}^d$ called the structuring element. Then the *dilation* $\delta_A(f)$ and *erosion* $\varepsilon_A(f)$ of f by A are defined by

$$\delta_A(f)(x) = \max_{y \in A, x-y \in F} f(x-y), \quad \varepsilon_A(f)(x) = \min_{y \in A, x+y \in F} f(x+y). \tag{1}$$

So dilation and erosion simply replace each value by the maximum or minimum in a neighbourhood defined by the structuring element A. By taking products of dilation and erosion we can construct *openings* and *closings*. The opening $\alpha_A(f)$ and closing $\phi_A(f)$ of f by A are defined by

$$\alpha_A(f)(x) = \delta_A(\varepsilon_A(f))(x), \qquad \phi_A(f)(x) = \varepsilon_A(\delta_A(f))(x). \tag{2}$$

The opening has the property that it is increasing ($f \leq g$ implies that $\alpha_A(f) \leq \alpha_A(g)$), anti-extensive ($\alpha_A(f) \leq f$) and idempotent ($\alpha_A(\alpha_A(f)) = \alpha_A(f)$). Similar properties hold for the closing, with the difference that closing is extensive ($\phi_A(f) \geq f$). The opening eliminates peaks, the closing valleys.

Pyramids We outline here the multiresolution signal decomposition scheme as recently introduced by Goutsias and Heijmans [3, 6], which encompasses linear (e.g. laplacian) and nonlinear pyramid schemes.

Consider signals in a d-dimensional signal space V_0, which is assumed to be the set of functions on (a subset of) the discrete grid $\mathbb{Z}^d$ that take values in a finite set of nonnegative integers. The goal is to decompose the original signal $f \in V_0$ into a number of coarser signals $f_j, j = 0, 1, 2, \ldots$. Here j is called the level of the decomposition. It is assumed that the signals f_j are elements of associated signal spaces V_j, which have the same structure as V_0.

Signal decomposition or *analysis* proceeds by analysis operators $\psi_j^\uparrow : V_j \rightarrow V_{j+1}$, which map a signal to a level higher in the pyramid, thereby reducing information. Signal reconstruction or *synthesis* proceeds by synthesis operators $\psi_j^\downarrow : V_{j+1} \rightarrow V_j$, which map a signal to a level lower in the pyramid. To guarantee that information lost during analysis can be recovered in the synthesis phase in a non-redundant way, one needs the so-called *pyramid condition*:

$$\psi_j^\uparrow \psi_j^\downarrow(f) = f \text{ for all } f \text{ on } V_{j+1}. \tag{3}$$

Decomposition of a signal $f \in V_0$ proceeds by

$$\begin{aligned} f_0 &= f \\ f_{j+1} &= \psi_j^\uparrow(f_j), \quad j \geq 0 \\ d_j &= f_j \dot{-} \psi_j^\downarrow(f_{j+1}). \end{aligned}$$

In a decomposition of L levels, this results in a sequence $d_0, d_1, \ldots, d_{L-1}, f_L$, where $\{d_j\}$ are detail signals and f_L an approximation signal at the coarsest level. Here $\dot{-}$

is a generalized subtraction operator. Assuming there exists an associated generalized addition operator $\dot{+}$ such that

$$\hat{f} \dot{+} (f \dot{-} \hat{f}) = f, \quad \text{if } f \in V_j \text{ and } \hat{f} = \psi_j^{\downarrow}\psi_j^{\uparrow}(f),$$

we have perfect reconstruction, that is, $f \in V_0$ can be *exactly* reconstructed from the sequence $d_0, d_1, \ldots, d_{L-1}, f_L$ by the recursion

$$f_j = \psi_j^{\downarrow}(f_{j+1}) \dot{+} d_j. \tag{4}$$

The operators $\dot{+}$ and $\dot{-}$ can be ordinary addition and subtraction, but other choices are possible, as we will see below.

By *approximations* of $f \in V_0$ we will mean signals which are reconstructed from higher levels by omitting some of the detail signals. To make this notion precise, we introduce the multilevel analysis operator $\psi_{i,j}^{\uparrow} = \psi_{j-1}^{\uparrow}\psi_{j-2}^{\uparrow}\cdots\psi_i^{\uparrow}, j > i$, which maps an element of V_i to an element of V_j. Similarly, the multilevel synthesis operator $\psi_{i,j}^{\downarrow} = \psi_i^{\downarrow}\psi_{i+1}^{\downarrow}\cdots\psi_{j-1}^{\downarrow}, j > i$, maps an element of V_j back to an element of V_i. The operator $\hat{\psi}_{i,j} = \psi_{i,j}^{\downarrow}\psi_{i,j}^{\uparrow}$ can be regarded as an *approximation operator* that maps the information obtained at level j by the analysis operator $\psi_{i,j}^{\uparrow}$ back to level i by the synthesis operator $\psi_{i,j}^{\downarrow}$. Now we define a level-j approximation $\hat{f}_{0,j}$ of $f \in V_0$ as

$$\hat{f}_{0,j} = \hat{\psi}_{0,j}(f) = \psi_{j,0}^{\downarrow}\psi_{0,j}^{\uparrow}(f) = \psi_{j,0}^{\downarrow}(f_j).$$

Adjunction pyramids We now introduce the class of so-called *morphological adjunction pyramids* [3], for which (i) the analysis and synthesis operators are independent of level ($\psi_j^{\uparrow} = \psi^{\uparrow}, \psi_j^{\downarrow} = \psi^{\downarrow}$), and (ii) $\psi^{\uparrow} : V_0 \to V_1$ and $\psi^{\downarrow} : V_1 \to V_0$ form a so-called *adjunction* between V_0 and V_1, implying that $\psi^{\uparrow}$ is an *erosion*, i.e. commutes with minima, and $\psi^{\downarrow}$ is a *dilation*, i.e. commutes with maxima[1]. In this case, the analysis and synthesis operators acting on a d-dimensional signal f have the form

$$\psi_A^{\uparrow}(f)(n) = \sigma^{\uparrow}\,\varepsilon_A(f), \qquad \psi_A^{\downarrow}(f)(k) = \delta_A\,\sigma^{\downarrow}(f). \tag{5}$$

Here $\delta_A(f)$ and $\varepsilon_A(f)$ are the dilation and erosion defined in (1), whereas $\sigma^{\uparrow}$ and $\sigma^{\downarrow}$ denote dyadic downsampling and dyadic upsampling in each spatial dimension:

$$\sigma^{\uparrow}(f)(n) = f(2n)$$
$$\sigma^{\downarrow}(f)(m) = \begin{cases} f(n), & \text{if } m = 2n \\ 0, & \text{otherwise} \end{cases}$$

So in the analysis phase we first compute an erosion, and then downsample; in the synthesis phase we first upsample and then dilate. Note that the notation is somewhat confusing: the arrow on σ for downsampling points upwards, and vice versa for upsampling. This is because downsampling is related to going to coarser levels in the pyramid,

[1] In a more general setting, ‘maxima’ and ‘minima’ should be replaced by ‘suprema’ and ‘infima’, respectively.

which traditionally are the higher levels. We could have inverted the arrows, so that the pyramid in upside-down, but decided to adhere to the notation of [3]. The pyramid condition (3) is satisfied, if there exists an $a \in A$ such that the translates of a over an even number of grid steps are never contained in the structuring element A; see [3] for more details.

In an adjunction pyramid, the product $\psi_A^{\downarrow}\psi_A^{\uparrow}$ is an *opening*, i.e. an operator which is increasing, anti-extensive and idempotent. The anti-extensivity property means that $\psi_A^{\downarrow}\ \psi_A^{\uparrow}\ (f) \leq f$. Therefore, we can define the generalized addition and subtraction operators by (cf. [3]):

$$t \dot{+} s = t \vee s = \max(t, s) \tag{6}$$

$$t \dot{-} s = \begin{cases} t, & \text{if } t > s \\ 0, & \text{if } t = s \end{cases} \tag{7}$$

where 0 is the smallest element, that is, the smallest image or voxel value. As a consequence, the detail signals are non-negative:

$$d_j(n) = f_j(n) \dot{-} \psi_A^{\downarrow}(f_{j+1})(n) = f_j(n) \dot{-} \psi_A^{\downarrow}\ \psi_A^{\uparrow}\ (f_j)(n) \geq 0. \tag{8}$$

Note that (7) implies that the detail signal $d_j(n)$ equals $f_j(n)$, except at points n for which $f_j(n) = \psi_A^{\downarrow}\ \psi_A^{\uparrow}\ (f_j)(n)$, where $d_j(n) = 0$. So, detail signals are not 'small' in regions where the structuring element does not fit well to the data. As long as we only look at the approximation signals, this is not a problem, but for compression purposes other choices of addition and subtraction operators are more suitable.

For an adjunction pyramid with the addition operator defined by (6), the reconstruction takes a special form. Making use of the fact that $\psi_A^{\downarrow}$ is a dilation, hence commutes with maxima, we derive from (4) and (6):

$$f = {\psi_A^{\downarrow}}^{L}(f_L) \vee \bigvee_{k=0}^{L-1} {\psi_A^{\downarrow}}^{k}(d_k), \tag{9}$$

where L is the decomposition depth and ${\psi_A^{\downarrow}}^{k}$ denotes k-fold composition of $\psi_A^{\downarrow}$ with itself. This representation is quite similar to the (linear) laplacian pyramid representation [1]. The main difference is that sums have been replaced by maxima.

3 Multiresolution maximum intensity projection

Now we come to the new part of this paper, which is the derivation of a multiresolution MIP volume rendering algorithm with progressive refinement based on morphological pyramids. To emphasize the main ideas, we first consider projections[2] along one of the coordinate axes, and then briefly indicate how the method can be extended to arbitrary viewing directions.

[2] By 'projection', we mean maximum intensity projection in what follows.

Axial projections Consider a 3-D volume data set f, and project parallel to the z-axis by computing the maximum value. The result is denoted by $\mathcal{M}(f)$:

$$\mathcal{M}(f)(x,y) = \max_z f(x,y,z), \qquad (x,y,z) \in \mathbb{Z}^3.$$

Applying the pyramid representation (9), and the fact that $\mathcal{M}$ evidently distributes over maxima, we get

$$\mathcal{M}(f) = \Big(\mathcal{M}({\psi_A^{\downarrow}}^L(f_L))\Big) \vee \bigvee_{k=0}^{L-1} \Big(\mathcal{M}({\psi_A^{\downarrow}}^k(d_k))\Big) \tag{10}$$

In principle, this formula allows us to do multiresolution MIP. Computationally, however, this expression is inefficient, because to compute the projections at a certain level k, we have to reconstruct first to full resolution by ${\psi_A^{\downarrow}}^k$ and then apply the maximum operator $\mathcal{M}$. It would be desirable to first compute the maxima along the line of sight on a coarse level, where the size of the data is reduced, before applying a synthesis operator to perform reconstruction to a finer resolution level. This is possible, as is shown next.

Computing the maxima before synthesis As (5) shows, the synthesis operator $\psi_A^{\downarrow}$ is composed of upsampling, followed by a dilation. Therefore, our problem is to rewrite $\mathcal{M}\,\psi_A^{\downarrow}\,(f) = \mathcal{M}\delta_A\,\sigma^{\downarrow}(f)$ such that the projection operator $\mathcal{M}$ is 'moved to the right'. The problem can be split in two parts. First we consider the projection of a dilated function, then the projection of an upsampled function, and finally combine the two results.

Now, both $\mathcal{M}$ and δ_A involve the computation of maxima. Therefore, it is easy to see that to compute $\mathcal{M}\delta_A(f)$, we can first project f along the z-axis, and then dilate the resulting 2-D function by a structuring element $\tilde{A}$, which is the projection of A. So,

$$\mathcal{M}\,\delta_A(f) = \delta_{\tilde{A}}\,\mathcal{M}(f), \tag{11}$$

with $\tilde{A} := \{(x,y) \in \mathbb{Z}^2 | (x,y,z) \in A \text{ for some } z \in \mathbb{Z}\}$. Note that δ_A is a 3-D dilation, while $\delta_{\tilde{A}}$ is a 2-D dilation (both defined by the formula (1) which holds for any dimension).

Next, consider projection of an upsampled function: $\mathcal{M}\,\sigma^{\downarrow}(f)$. Upsampling has the effect of inserting zeroes between neighbouring voxels in all three spatial dimensions. If we project the upsampled function, then for those (x,y) which are in the projection of the support of the original function f the outcome will be unaffected, since the inserted zero values never contribute to the maximum, zero being the minimum data value possible. On the other hand, for those (x,y) which are not in the projection of the support of the original function f, projection means computing the maximum of a vertical line of zeroes, which results in a zero at (x,y). Therefore,

$$\mathcal{M}\,\sigma^{\downarrow}(f) = \sigma^{\downarrow}\,\mathcal{M}(f), \tag{12}$$

where $\sigma^{\downarrow}$ on the right-hand side is a 2-D upsampling operator (the dimension of $\sigma^{\downarrow}$ is clear from the dimension of the functions on which it acts).

Now we can take the final step, which is to combine (11) and (12). We find,

$$\mathcal{M}\,\psi_A^{\downarrow}\,(f) = \mathcal{M}\delta_A\,\sigma^{\downarrow}(f) = \delta_{\tilde{A}}\,\mathcal{M}\,\sigma^{\downarrow}(f) = \delta_{\tilde{A}}\,\sigma^{\downarrow}\mathcal{M}(f) = \psi_{\tilde{A}}^{\downarrow}\,\mathcal{M}(f),$$

where $\psi_{\tilde{A}}^{\downarrow} = \delta_{\tilde{A}}\,\sigma^{\downarrow}$ is a 2-D synthesis operator of the same form as $\psi_A^{\downarrow}$ (the 3-D structuring element A has only been replaced by a 2-D structuring element $\tilde{A}$). It is evident that a similar formula holds for iterated versions of $\psi_A^{\downarrow}$.

As a result of the above analysis, we have proved the main result of this paper, which is a multiresolution representation of the maximum intensity projection $\mathcal{M}(f)$ of a 3-D voxel array f:

$$\mathcal{M}(f) = \left(\psi_{\tilde{A}}^{\downarrow}{}^{L}\mathcal{M}(f_L)\right) \vee \bigvee_{k=0}^{L-1} \left(\psi_{\tilde{A}}^{\downarrow}{}^{k}\mathcal{M}(d_k)\right) \tag{13}$$

As long as a user is interacting with the data (preview mode), only a coarse approximation $\hat{\mathcal{M}}_j(f)$ may be used, which can be refined to full resolution for close inspection.

The MMIP algorithm The multiresolution MIP algorithm can be summarized as follows.

- *Preprocessing*. Compute an L-level 3-D morphological pyramid of the volume data, resulting in a sequence $d_0, d_1, \ldots, d_{L-1}, f_L$
- *Actual MIP volume rendering*.
  1. Compute a low resolution approximation $\hat{\mathcal{M}}_L(f)$ by projecting f_L followed by applying the 2-D synthesis operator $\psi_{\tilde{A}}^{\downarrow}{}^{L}$.
  2. Refine the image progressively by taking the detail signals d_k, $k = L-1, \ldots, 0$ into account. From a level j approximation $\hat{\mathcal{M}}_j(f)$, compute a level $j-1$ approximation $\hat{\mathcal{M}}_{j-1}(f)$ by projecting d_{j-1}, applying the 2-D pyramid synthesis operator $\psi_{\tilde{A}}^{\downarrow}$ to the projection, and finally taking the maximum of the 2-D signal so obtained with the previous approximation:

$$\hat{\mathcal{M}}_{j-1}(f) = \psi_{\tilde{A}}^{\downarrow}{}^{j-1}(\mathcal{M}(d_{j-1})) \vee \hat{\mathcal{M}}_j(f). \tag{14}$$

  3. The recursion terminates with $\hat{\mathcal{M}}_0(f) = \mathcal{M}(f)$, the exact MIP of f.

The structure of this algorithm is very similar to that of wavelet splatting [7, 8, 17], with the difference that (i) linear summation of voxel values is replaced by maximum computation, and (ii) linear wavelet filters have been replaced by morphological filters (dilation and erosion).

From (14) we immediately deduce that $\hat{\mathcal{M}}_j(f) \leq \hat{\mathcal{M}}_{j-1}(f)$. So if we define a global error measure by the maximum (or L_∞) norm, then approximation error decreases monotonically as we go down the pyramid.

Implementation We implemented the MIP projections $\mathcal{M}$ required in the MMIP algorithm by means of the object order voxel projection method of Mroz *et al.* [11], that treats voxels as cells which a constant data value, which are simply projected on the viewing plane, each voxel contributing to exactly one pixel. The method also uses an efficient volume data storage scheme, by histogram-based sorting of interesting voxels according to grey value, and storing these in a value-sorted array of voxel positions. An additional array contains the cumulative histogram values. For our case of multiresolution data, all levels of the pyramid were created and stored as value-sorted arrays. To prevent holes forming in the projection image for non-axial viewing directions, we post-processed the projection images by performing a closing with a flat structuring element of size 2×2. By $\mathcal{M}_{\rm discrete}$ we will denote the operator consisting of voxel projection followed by the final closing. In the experiments to be discussed in Section 4, we define interesting voxels simply as those with a non-zero grey value (i.e., we did not use the preprocessing scheme of [11] to identify and remove other types of non-contributing voxels). In practice, especially for angiographic data, a substantial reduction (up to 99%) in the amount of voxels to be processed is possible when only nonempty voxels are stored.

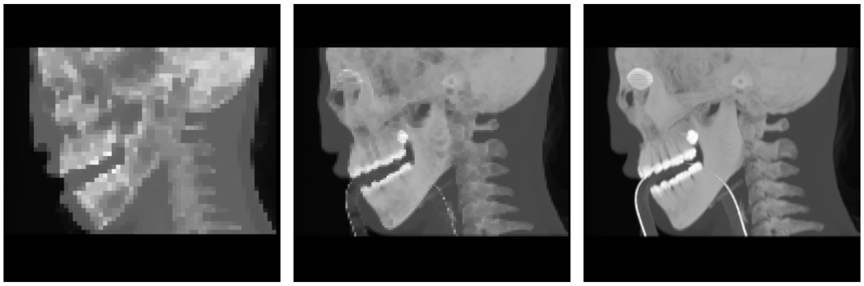
Arbitrary view directions To deal with general view directions a continuous function has first to be reconstructed from the discrete samples of the data set f. The MIP operator $\mathcal{M}$ now computes the maximum of the reconstructed function along a given direction vector. The reconstruction operator is assumed to be a dilation (one choice is the voxel model, which treats voxels as cells which a constant data value). Then formulas (9) and (10) still hold for the reconstructed functions. Formulas (11) and (12) have to be slightly adapted to deal with discretization effects. In the experiments below, we have simply replaced $\mathcal{M}$ by the discrete MIP operator $\mathcal{M}_{\rm discrete}$ of the previous subsection. The structuring element $\tilde{A}$ in (11) is replaced by the result of applying $\mathcal{M}_{\rm discrete}$ to the 3-D structuring element A. In the experiments, we found this approximation to be quite accurate. A careful study of the associated discretization error will be presented elsewhere.

4 Examples

Experiments were carried out on a PC with a 500 MHz Pentium III processor and 128 Mb memory. We performed MMIP rendering of a CT head data set and an MR angiography data set, both of size 256^3, using a 2-level pyramid. Dilations and erosions with a $2 \times 2 \times 2$ structuring element (3-D morphological Haar pyramid) were used. The sampling distance in the view plane was taken equal to the sampling distance of the original volume data. For the CT data, about 26% of the data consisted of nonzero voxels; for the angiography data, this was 1.25%. Creation of the pyramid took about 25 seconds in both cases. Rendering times were found to be almost independent of view angle. Sizes in value-sorted array format and rendering times of the successive levels of the pyramid are given in Table 1. For comparison, the numbers for direct MIP rendering of the full-size volume data are given as well. All times are excluding I/O. The timings show that computing a level-2 or level-1 approximation takes considerably less time than a full-size MIP, especially for data sets with a relatively large number of nonzero voxels. Figure 1 shows successive approximations for the CT data. MMIP approximations quickly remove details of the data, due to the fact that the approximations essentially are morphological openings by a structuring element whose size increases with level. Note in particular in Fig. 1 that small details such as the tube from the mouth almost disappear in the level 1 approximation. To be useful for angiographic data, the method has to be adapted so that small details are better preserved in higher levels of the pyramid (see discussion).

Table1. Data sizes (value-sorted array format) and rendering times of MIP (full image) and MMIP (progressive renderings of approximation and detail data).

MRA data $256 \times 256 \times 256$	size (kbytes)	time (s)	CT data $256 \times 256 \times 256$	size (kbytes)	time (s)
full image	838.5	0.423	full image	17433	6.92
level 2 approximation	0.812	0.110	level 2 approximation	253	0.20
add detail level 1	30.2	0.129	add detail level 1	1861	0.87
add detail level 0	801.6	0.417	add detail level 0	15171	6.04



level 2 approx. $\hat{\mathcal{M}}_2(f)$ level 1 approx. $\hat{\mathcal{M}}_1(f)$ original $\mathcal{M}(f)$

Figure1. MMIP reconstruction from a 2-level morphological adjunction pyramid using a $2 \times 2 \times 2$ structuring element.

5 Discussion

Several extensions to the proposed multiresolution extension of MIP volume rendering are possible. Adjunction pyramids with non-flat structuring functions can be used. The shape of this structuring function can be adapted in such a way that smaller details of the data at higher approximation levels are retained. For the same purpose, other operators than erosions, such as openings, can be used for the analysis phase. This implies however, that the representation formula (9) no longer holds. To maintain an acceptable level of efficiency, we still require that the synthesis operator $\psi^{\downarrow}$ is a dilation, so that it commutes with maxima. For compression purposes, other choices of addition and subtraction operators can be considered, and morphological wavelets can be used [6], which have the advantage that they provide a non-redundant multiresolution representation.

References

1. Burt, P. J., and Adelson, E. H. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun. 31* (1983), 532–540.
2. Cai, W., and Sakas, G. Maximum intensity projection using splatting in sheared object space. *Computer Graphics Forum (Proc. Proc. Eurographics'98) 17*, 3 (1998), C113–124.
3. Goutsias, J., and Heijmans, H. J. A. M. Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids. Tech. Rep. PNA-R9810, Centre for Mathematics and Computer Science, Amsterdam, Oct. 1998.
4. Grosso, R., and Ertl, T. Biorthogonal wavelet filters for frequency domain volume rendering. In *Proceedings of Visualization in Scientific Computing '95* (1995), J. van Wijk, R. Scateni, and P. Zanarini, Eds.
5. Heijmans, H. J. A. M. *Morphological Image Operators*, vol. 25 of *Advances in Electronics and Electron Physics, Supplement*. Academic Press, New York, 1994.
6. Heijmans, H. J. A. M., and Goutsias, J. Multiresolution signal decomposition schemes. Part 2: morphological wavelets. Tech. Rep. PNA-R9905, Centre for Mathematics and Computer Science, Amsterdam, June 1999.
7. Lippert, L., and Gross, M. H. Fast wavelet based volume rendering by accumulation of transparent texture maps. *Computer Graphics Forum 14*, 3 (1995), 431–443.
8. Lippert, L., Gross, M. H., and Kurmann, C. Compression domain volume rendering for distributed environments. In *Proc. Eurographics'97* (1997), pp. 95–107.
9. Lürig, C., and Ertl, T. Hierarchical volume analysis and visualization based on morphological operators. In *Proc. IEEE Visualization '98* (1998), IEEE Computer Society Press, pp. 335–341.
10. Malzbender, T. Fourier volume rendering. *ACM Transactions on Graphics 12*, 3 (1993), 233–250.
11. Mroz, L., König, A., and Gröller, E. Maximum intensity projection at warp speed. *Computers & Graphics 24* (2000), 343–352.
12. Muraki, S. Volume data and wavelet transforms. *IEEE Computer Graphics and Applications 13*, 4 (1993), 50–56.
13. Roerdink, J. B. T. M., and Blaauwgeers, G. S. M. Visualization of Minkowski operations by computer graphics techniques. In *Mathematical Morphology and its Applications to Image Processing*, J. Serra and P. Soille, Eds. Kluwer Acad. Publ., Dordrecht, 1994, pp. 289–296.
14. Roerdink, J. B. T. M., and Westenberg, M. A. Wavelet-based volume visualization. *Nieuw Archief voor Wiskunde 17 (Fourth Series)*, 2 (July 1999), 149–158.
15. Serra, J. *Image Analysis and Mathematical Morphology*. Academic Press, New York, 1982.
16. Sternberg, S. R. Grayscale morphology. *Comp. Vis. Graph. Im. Proc. 35* (1986), 333–355.
17. Westenberg, M. A., and Roerdink, J. B. T. M. Frequency domain volume rendering by the wavelet X-ray transform. *IEEE Trans. Image Processing 9*, 7 (2000), 1249–1261.
18. Westermann, R., and Ertl, T. A multiscale approach to integrated volume segmentation and rendering. In *Proc. Eurographics'97, Vienna*, D. Fellner and L. Szirmay-Kalos, Eds., vol. 16. 1997, pp. C–117–C–127.
19. Westover, L. A. Footprint evaluation for volume rendering. *Computer Graphics 24*, 4 (1990), 367–376.
20. Zuiderveld, K. J., Koning, A. H. J., and Viergever, M. A. Techniques for speeding up high-quality perspective Maximum Intensity Projection. *Pattern Recognition Letters 15* (1994), 507–517.