Journal of Mathematical Imaging and Vision 22: 143–157, 2005

Morphological Pyramids in Multiresolution MIP Rendering of Large Volume Data: Survey and New Results

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 survey and extend nonlinear signal decompositions based on morphological pyramids, and their application to multiresolution maximum intensity projection (MIP) volume rendering with progressive refinement and perfect reconstruction. The structure of the resulting multiresolution rendering algorithm is very similar to wavelet splatting. Several existing classes of pyramids are discussed, and their limitations indicated. To enhance the approximation quality of visualizations from reduced data (higher levels of the pyramid), two approaches are explored. First, a new class of morphological pyramids, involving connectivity enhancing operators, is considered. In the pyramidal analysis phase, a conditional dilation operator is used, with a given number n of iterations. The corresponding pyramids for $n = 0$ and $n = 1$ are known as the adjunction pyramid and Sun-Maragos pyramid, respectively. We show that the approximation quality when rendering from higher levels of the pyramid does increase as a function of the number of iterations n of the conditional dilation operator, but the improvement for $n > 1$ is limited. The second new approach, called streaming MIP-splatting, again starts from the adjunction pyramid. The new element is that detail coefficients of all levels are considered simultaneously and are resorted with respect to decreasing magnitude of a suitable error measure. All resorted coefficients are projected successively, until a desired accuracy of the resulting MIP image is obtained. We show that this method outperforms the previous methods based on morphological pyramids, both with respect to image quality with a fixed amount of detail data, and in terms of flexibility of controlling approximation error or computation time.

Keywords: multiresolution signal decomposition, volume rendering, maximum intensity projection, progressive refinement, morphological pyramids, approximation error, streaming MIP-splatting

1. Introduction

Multiresolution signal decomposition schemes have a long history in image processing and analysis. Such schemes can be useful for noise suppression, more robust detection of signal features, and improved computational efficiency by analyzing a signal in a coarse-to-fine fashion. Well-known are linear multiresolution schemes, such as the Laplacian pyramid [1] or decomposition methods based on wavelets [9]. More recent are nonlinear multiresolution signal decompositions based on morphological pyramids. A detailed study of such pyramids was made by Goutsias and Heijmans [4, 6]. Morphological pyramids systematically split the input signal into approximation and detail signals by repeatedly applying a pyramidal analysis operator which involves morphological filtering followed by downsampling. As the level of the pyramid is increased, spatial features of increasing size are retained. The original signal can be recovered from the pyramid decomposition by repeated application of a pyramid synthesis operator. An important element of the morphological pyramid schemes is the so-called *pyramid condition*, which states that synthesis of a signal followed by analysis returns the original signal [4].

In this paper we survey and extend the application of morphological pyramids to the problem of multiresolution visualization of large volume data sets.

Volume visualization, or volume rendering, is a technique which produces two-dimensional images of three-dimensional data from different viewpoints, using advanced computer graphics techniques such as illumination, shading and colour [3]. Expecially when interactive rendering rates are required (i.e., there must be a fast response of the rendering system to actions of the user), this is a very demanding problem when the sizes of the volume data are large. For this reason, a multiresolution approach is an obvious choice, allowing to quickly visualize reduced versions of the data which can be progressively refined if needed. In so-called preview mode, when a user explores the data from different viewpoints, only approximation data on the highest decomposition level L are used, so that the amount of data, and therefore the rendering time, is roughly reduced by a factor of 8^L. This leads to improved user interaction. A second reason for applying morphological methods in volume rendering is the feature extraction capability of morphological operators, which can be incorporated in the volume rendering process.

A volume rendering method which is widely used in medical imaging is *maximum intensity projection* (MIP). Here one generates, for each pixel of the view plane, a ray through the data parallel to the line of sight (i.e., perpendicular to the view plane), and assigns the maximum data value encountered along this ray to the pixel. Because of its computational simplicity, MIP is widely used in the display of magnetic resonance angiography (MRA) and ultrasound data. Since the MIP transform is nonlinear, the standard linear multiresolution models based on wavelets [5, 7, 11, 16, 20, 21] are not applicable. Instead, morphological pyramids can be used for multiresolution rendering. Even though the morphological operators are nonlinear and noninvertible, the pyramid scheme does allow perfect reconstruction as well as progressive refinement, just as in the linear case. Pyramids where the synthesis operator is a dilation are particularly appropriate. For brevity, we will refer to this class of pyramids as *dilation pyramids*. These pyramids have the advantage that it is not necessary to first reconstruct the 3-D data to full resolution before computing the MIP operation. Instead, the maxima along the line of sight can be first computed from the data on a coarse level (where the size of the data is reduced), after which a fast 2-D morphological synthesis operator is used to perform reconstruction of the projection image to full grid resolution.

We have recently investigated two cases of dilation pyramids for MIP volume rendering. The first is the *adjunction pyramid* [12,14], where the pyramidal analysis and synthesis operators are composed of morphological erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis [4]. When applied to MIP volume rendering, we found that too few small features present in the data were retained in high levels of the pyramid. To put it differently, the detail signals are 'too large'. To improve the effectiveness of feature extraction, we studied in [13] an alternative pyramid scheme in which a morphological opening instead of an erosion is used for pyramidal analysis (the pyramidal synthesis operator is still a dilation). This pyramid has been studied in morphological image processing by Sun and Maragos [18], see also [4], and is referred to as the *Sun-Maragos pyramid* below. When applying the Sun-Maragos pyramid to MIP volume rendering, the approximation quality improved, essentially because erosions are replaced by openings, which keep image features to a larger extent, so that the chance that (parts of) these features survive the downsampling step is larger. Nevertheless, small structures such as veins and arteries in angiographic data become disconnected during the pyramid decomposition process. This suggests that connectivity preserving operators might do even better.

The goal of this paper is to improve upon the approximation quality of the morphological pyramids mentioned above along two distinct routes. First, we study the application of a new class of dilation pyramids in which the pyramidal synthesis operator $\psi^{\downarrow} = \delta_A \sigma^{\downarrow}$ is fixed to be a dilation δ_A by a structuring element A (preceded by an upsampling operator $\sigma^{\downarrow}$), but where the pyramid analysis operator has the form $\psi^{\uparrow} = \sigma^{\uparrow} \eta$ where $\sigma^{\uparrow}$ denotes downsampling and η may be chosen in different ways. This class was recently introduced in [15] for analyzing 2-D images, and in this paper we study its usefulness to rendering of volume data. Choosing the operator η to be an erosion ε_A, followed by an arbitrary number of conditional dilations with structuring element A, also leads to a valid synthesis operator, that is, the pair $(\psi^{\uparrow}, \psi^{\downarrow})$ still satisfies the pyramid condition. We refer to these pyramids as *conditional dilation pyramids*. Note that this class also contains the opening by reconstruction, which is the connected filter obtained by iterating the conditional dilations until idempotence [17]. We present experiments to show that indeed slightly more details can be retained in high pyramid levels when the iteration

number n of the conditional dilation increases. However, the improvement is quite limited, and comes at the expense of increased computation times during the analysis phase of the pyramid, especially for large n.

The second new approach, called *streaming MIP-splatting* after the analogous approach in the linear case of wavelet splatting [8], again starts from the adjunction pyramid. The difference with the original algorithm [12, 14] is that detail coefficients of all levels are considered simultaneously and are resorted with respect to decreasing magnitude of a suitable error measure. Detail coefficients in the adjunction pyramid as defined in [12,14] are not 'small' in the usual sense, because they are defined by the application of an unusual difference operator. Therefore, in our new approach we also consider the detail coefficients one would obtain by ordinary subtraction, sort these by decreasing magnitude, and after that replace them again by the original detail coefficients at the corresponding location and scale. In the rendering phase, all resorted coefficients are projected successively, until a desired accuracy of the resulting MIP image is obtained. We show that this method outperforms the previous methods based on morphological pyramids, both with respect to image quality with a fixed amount of detail data, and in terms of flexibility of controlling approximation error or computing time.

The remainder of this paper is organized as follows. Section 2 supplies a few preliminaries on morphological pyramids, and gives the definition of adjunction, Sun-Maragos and conditional dilation pyramids. In Section 3 we first recall the MIP algorithm based on adjunction pyramids, and then introduce the new multiresolution MIP algorithm based on conditional dilation pyramids. An evaluation of the approximation quality of the different types of conditional dilation pyramids is presented. In Section 4, the streaming MIP-splatting algorithm is introduced, and we present experimental results which show its superior performance when compared to the previous approaches. Section 5 contains a summary and discussion.

2. Preliminaries

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 $\mathbb{Z}^d$, where $d = 2$ or $d = 3$ (image and volume data), that take values in a finite set of nonnegative integers.

The general structure of linear as well as nonlinear pyramids is as follows. From an initial signal f_0, approximations $\{f_j\}$ of increasingly reduced size are computed by a decomposition or analysis operator $\psi^\uparrow$:

$$f_j = \psi^\uparrow(f_{j-1}), \quad j = 1, 2, \ldots L.$$

Here j is called the level of the decomposition.

The set $\{f_0, f_1, \ldots, f_L\}$ is referred to as an *approximation pyramid*. In the case of a Gaussian pyramid, the analysis operator consists of Gaussian low-pass filtering, followed by downsampling [1]. An approximation error associated to f_{j+1} may be defined by taking the difference between f_j and an expanded version of f_{j+1}:

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

Here $\dot{-}$ is a generalized subtraction operator. The set $d_0, d_1, \ldots, d_{L-1}, f_L$ is referred to as a *detail pyramid*. Assuming there exists an associated generalized addition operator $\dot{+}$ such that, for all j,

$$\hat{f}_j \dot{+} (f_j \dot{-} \hat{f}_j) = f_j, \quad \text{where } \hat{f}_j = \psi^\downarrow(\psi^\uparrow(f_j)),$$

we have *perfect reconstruction*, that is, f_0 can be exactly reconstructed by the recursion

$$f_j = \psi^\downarrow(f_{j+1}) \dot{+} d_j, \quad j = L-1, \ldots, 0. \tag{2}$$

For the linear case, the detail pyramid is called a Laplacian pyramid, and the synthesis operation consists of upsampling, followed by Gaussian low-pass filtering [1]. In the case of morphological pyramids, the analysis and synthesis operators involve morphological filtering instead of Gaussian filtering [4, 6]. It should be noted that, in principle, the analysis and synthesis operators may depend on level, but we assume them to be the same for all levels throughout this paper.

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* [4]:

$$\psi^\uparrow(\psi^\downarrow(f)) = f \text{ for all } f. \tag{3}$$

By approximations of f we mean signals in V_0 of the same size as the initial signal f which are reconstructed from higher levels of the pyramid by omitting

some of the detail signals. More precisely, a level-j approximation $\hat{f}_j^{(0)}$ of f is defined as

$$\hat{f}_j^{(0)} = \psi^{\downarrow j}(f_j), \tag{4}$$

where $\psi^{\downarrow j}$ means repeating the $\psi^{\downarrow}$ operator j times.

The generalized addition and subtraction operators $\dot{+}$ and $\dot{-}$ in (1) and (2) are usually taken as ordinary addition and subtraction. For the case of the adjunction pyramid, another choice is possible, see below.

2.1. Dilation Pyramids

We now consider the general class of pyramids whose analysis/synthesis operator pairs have the form

$$\psi^{\uparrow}(f) = \sigma^{\uparrow}(\eta(f)), \tag{5}$$
$$\psi^{\downarrow}(f) = \delta_A(\sigma^{\downarrow}(f)), \tag{6}$$

where $\delta_A(f)$ is the flat[1] grey-value dilation with structuring element A, and $\sigma^{\uparrow}$ and $\sigma^{\downarrow}$ denote dyadic downsampling and upsampling by a factor of 2 in each spatial dimension:

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

Here the arrows indicate transformations to higher (coarser) or lower (finer) levels of the pyramid.

The (not yet specified) operator η, and the structuring element A, are assumed to satisfy certain conditions, in order that the pyramid condition (3) is satisfied. These conditions will be considered next, when we present several types of dilation pyramids which differ only in the form of the analysis operator η.

2.1.1. Adjunction Pyramid. In this case, $\eta(f) = \varepsilon_A(f)$, i.e. the erosion with structuring element A. As shown in [4], 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. Introducing the notation

$$\mathbb{Z}^d[n] = \{k \in \mathbb{Z}^d \mid k - n \in 2\mathbb{Z}^d\}$$
$$A[n] = A \cap \mathbb{Z}^d[n]$$

the pyramid condition can be expressed as

$$A[a] = \{a\} \text{ for some } a \in A. \tag{7}$$

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. [4]):

$$t \dot{+} s = t \vee s = \max(t, s),$$
$$t \dot{-} s = \begin{cases} t, & \text{if } t > s \\ 0, & \text{if } t = s \end{cases} \tag{8}$$

where 0 is the smallest image or voxel value possible. As a consequence, the detail signals are nonnegative:

$$\begin{aligned} d_j(n) &= f_j(n) \dot{-} \psi_A^{\downarrow}(f_{j+1})(n) \\ &= f_j(n) \dot{-} \psi_A^{\downarrow}\psi_A^{\uparrow}(f_j)(n) \geq 0. \end{aligned} \tag{9}$$

Note that the definition of $\dot{-}$ in (8) 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 fit approximately, but not exactly, to the data.

For an adjunction pyramid with the generalized addition being defined as the maximum operation (see (8)), the reconstruction takes a special form [12,14]. Making use of the fact that $\psi_A^{\downarrow}$ is a dilation, hence commutes with the maximum operation, we derive from (2) and (8):

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

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.

Remark 1. One may also use ordinary addition and subtraction to define the detail signals in an adjunction pyramid. However, unless indicated otherwise, by 'adjunction pyramid' we will always mean the adjunction pyramid with the special addition and subtraction given by (8).

2.1.2. Sun-Maragos Pyramids. This pyramid has the analysis operator $\eta(f) = \alpha_A(f)$, where $\alpha_A = \delta_A \varepsilon_A$

is the opening by structuring element A. Under the condition that

$$A[\mathbf{0}] = \{\mathbf{0}\}, \tag{11}$$

where $\mathbf{0}$ is the origin of $\mathbb{Z}^d$, the pyramid condition (3) is satisfied (see [4, Proposition 5.9]), that is,

$$\psi^{\uparrow}\,\psi^{\downarrow} = \sigma^{\uparrow}\,\delta_A\,\varepsilon_A\,\delta_A\sigma^{\downarrow} = \mathsf{id},$$

where id denotes the identity operator. Since $(\varepsilon_A, \delta_A)$ is an adjunction, we have that $\delta_A\,\varepsilon_A\,\delta_A = \delta_A$. Therefore, when A satisfies (11), the previous formula implies that

$$\sigma^{\uparrow}\,\delta_A\,\sigma^{\downarrow} = \mathsf{id}. \tag{12}$$

2.1.3. Trivial Pyramid. This pyramid has the analysis operator $\eta(f) = f$, i.e. the identity operator. Again, assuming that (11) holds, the pyramid condition is satisfied, since $\psi^{\uparrow}\,\psi^{\downarrow} = \sigma^{\uparrow}\,\delta_A\sigma^{\downarrow} = \mathsf{id}$, where (12) was used. In this type of pyramid, analysis consists only of downsampling, which is why we refer to it as the trivial pyramid (compare to the lazy wavelet in the lifting scheme [19]).

2.1.4. Conditional Dilation Pyramid. This pyramid has the analysis operator $\eta(f) = R_A^{(n)}(f)$, where the sequence of operators $R_A^{(n)}$ for $n = 0, 1, 2, \ldots$ is defined by the following recursion:

$$R_A^{(0)}(f) = \varepsilon_A(f) \tag{13}$$

$$R_A^{(n)}(f) = f \wedge \delta_A\big(R_A^{(n-1)}(f)\big), \quad n = 1, 2, \ldots. \tag{14}$$

The operator in (14) is a conditional dilation, that is, after each dilation step the infimum with the original signal f is taken. Therefore, $R_A^{(\infty)}(f)$ is the opening by reconstruction of f from its erosion $\varepsilon_A(f)$. In practice, f is defined on a finite subset $D \subseteq \mathbb{Z}^d$ and the recursion terminates after a finite number of steps.

By observing that $R_A^{(0)}(f)$ is the erosion ε_A of f and $R_A^{(1)}(f)$ is the opening $\alpha_A = \delta_A\,\varepsilon_A$ of f, we see that the cases $n = 0$ and $n = 1$ correspond to the adjunction pyramid and Sun-Maragos pyramid, respectively. Again assuming the structuring element A to satisfy condition (11), we showed in [15] that the pyramid condition is satisfied. To make this paper self-contained, the proof is reproduced in the Appendix.

3. Multiresolution MIP Volume Rendering

The basic idea of multiresolution MIP is to use a pyramid representation in which it is possible to interchange the MIP operator (computing maxima along the line of sight) with the pyramidal synthesis operator. Then the MIP operation can be performed on a coarse level, where the size of the data is reduced, before performing a fast 2-D morphological synthesis operation to reconstruct the projection image to full grid resolution, thus leading to a computationally efficient algorithm. For the dilation pyramids defined above, such commutativity of MIP and pyramid synthesis holds because both the upsampling operation and the dilation δ_A commute with the maximum operation [12, 14].

In the following, the MIP operation is denoted by $\mathcal{M}_{\boldsymbol{\Theta}}$, where the viewing coordinates are denoted by a vector $\boldsymbol{\Theta} = (\theta, \phi, \alpha)$. Here θ and ϕ are two angles defining the projection direction vector which is perpendicular to the view plane, and the angle α gives the orientation of the view plane with respect to this projection direction, cf. Fig. 1. Successive approximations of the MIP of f are denoted by $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(j)}(f)$, $j = L, L-1, \ldots, 0$, where L is the decomposition depth. These approximations all have the same size in the image plane, because they are reconstructed from higher levels in the pyramid to the size of the MIP of f.

For a discrete volume data set f, the MIP operator $\mathcal{M}_{\boldsymbol{\Theta}}$ projects all voxels on the view plane, and takes the maximum of all voxel values projecting to the same pixel. For non-axial projections, where the viewing coordinate system is not aligned with the Cartesian coordinate system in which the data are defined, additional interpolation issues arise. In order to emphasize the basic structure of the algorithm, these issues are not considered in what follows, see [14] for more details.

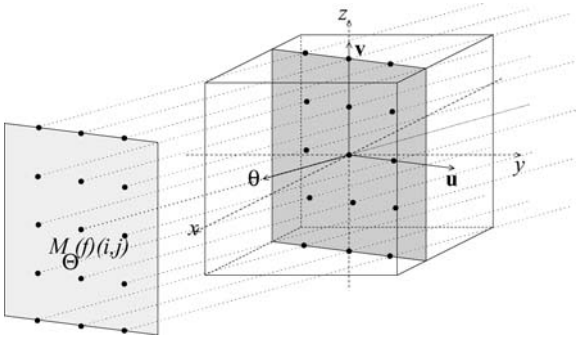

Figure 1. Data cube and view plane defined by unit vectors $\mathbf{u}, \mathbf{v}$ orthogonal to the viewing vector.

3.1. The Multiresolution MIP Algorithm for Dilation Pyramids

The multiresolution MIP algorithm for dilation pyramids defined by (5)–(6) can be summarized as follows.

- *Preprocessing*. Compute an L-level 3-D morphological pyramid with structuring element A of the volume data f, resulting in a sequence of volumes d_0, $d_1, \ldots, d_{L-1}$, f_L (detail pyramid), or in a sequence of volumes f_0, f_1, …, f_{L-1}, f_L (approximation pyramid). Here f_L is a signal at the coarsest level.
- *Actual MIP volume rendering*. For a given orientation $\boldsymbol{\Theta}$ of the viewing coordinate system, do:
  1. Compute a low resolution approximation $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L)}(f)$ by first applying the voxel projection operator $\mathcal{M}_{\boldsymbol{\Theta}}$ to f_L, followed by the 2-D synthesis operator $\psi_{\tilde{A}}^{\downarrow L}$:
  $$\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L)}(f) = \psi_{\tilde{A}}^{\downarrow L}(\mathcal{M}_{\boldsymbol{\Theta}}(f_L)). \tag{15}$$
  Here $\tilde{A} = \mathcal{M}_{\boldsymbol{\Theta}}(A)$ is a 2-D structuring element, which is obtained as the projection of the 3-D structuring element A on the view plane.
  2. Refine the image progressively.
     - *Detail pyramid*. For each level $j < L$, compute an approximation $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(j)}(f)$ on level j as follows. Reconstruct f_{j+1} to level j, and add the result to d_j to obtain f_j. Then apply the MIP operator $\mathcal{M}_{\boldsymbol{\Theta}}$, and finally perform a 2-D reconstruction operator j times:
     $$f_j = \psi_A^{\downarrow}(f_{j+1}) \dotplus d_j \tag{16}$$
     $$\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(j)}(f) = \psi_{\tilde{A}}^{\downarrow j}(\mathcal{M}_{\boldsymbol{\Theta}}(f_j)) \tag{17}$$
     - *Approximation pyramid*. Since f_j is available, directly apply formula (17) above.
  3. The recursion terminates with $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(0)}(f)$, which equals the MIP $\mathbf{M}_{\boldsymbol{\Theta}}(f)$ of the original data f.

3.2. The Multiresolution MIP Algorithm for Adjunction Pyramids

For the special case of the adjunction pyramid with the generalized addition and subtraction defined by (8), one may directly project all detail signals and refine in the image plane, as follows from (10). This leads to the following algorithm [12, 14].

- *Preprocessing*. Compute an L-level 3-D morphological adjunction pyramid with structuring element A of the volume data, resulting in a sequence d_0, d_1, …, d_{L-1}, f_L, where $\{d_j\}$ are detail signals and f_L is a signal at the coarsest level.
- *Actual MIP volume rendering*. For a given orientation $\boldsymbol{\Theta}$ of the viewing coordinate system, do:
  1. Compute a low resolution approximation $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L)}(f)$ by first applying the voxel projection operator $\mathcal{M}_{\boldsymbol{\Theta}}$ to f_L, followed by the 2-D synthesis operator $\psi_{\tilde{A}}^{\downarrow L}$:
  $$\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L)}(f) = \psi_{\tilde{A}}^{\downarrow L}(\mathcal{M}_{\boldsymbol{\Theta}}(f_L)). \tag{18}$$
  Here $\tilde{A} = \mathcal{M}_{\boldsymbol{\Theta}}(A)$ is a 2-D structuring element, which is obtained as the projection of the 3-D structuring element A on the view plane.
  2. Refine the image progressively by taking the detail signals d_k, $k = L-1, \ldots, 0$ into account. From a level-j approximation $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(j)}(f)$, compute an approximation $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(j-1)}(f)$ on level $j-1$ by projecting d_{j-1}, applying the 2-D pyramid synthesis operator $\psi_{\tilde{A}}^{\downarrow j-1}$ to the projection, and taking the maximum of the image so obtained with the previous approximation:
  $$\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(j-1)}(f) = \psi_{\tilde{A}}^{\downarrow j-1}(\mathcal{M}_{\boldsymbol{\Theta}}(d_{j-1})) \vee \hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(j)}(f). \tag{19}$$
  3. The recursion terminates with $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(0)}(f)$, which equals the MIP $\mathbf{M}_{\boldsymbol{\Theta}}(f)$ of the original data f.

3.3. Remarks on Implementation and Efficiency

For MIP rendering, the time complexity is $\mathcal{O}(N^3)$ for a volume data set of size N^3, since for each of the N^2 pixels in the image plane, the maximum of N voxel values has to be computed. Various acceleration schemes exist, but these do not change the complexity [2,10,22]. For multiresolution MIP, there is first the preprocessing step, only to be executed once, in which the pyramid is constructed. The analysis operator involves 3-D erosions which are linear in the number of voxels. A complete reconstruction is also linear in the number of voxels. In preview mode, rendering uses only the highest decomposition level L, and therefore the rendering time is reduced by a factor of 8^L.

Another issue is memory usage. For a volume data set of size N^3, the size of an L-level analysis pyramid

is given by

$$N^3\left(1+\frac{1}{8}+\cdots+\left(\frac{1}{8}\right)^L\right) = \frac{8}{7}N^3(1-(1/8)^{L+1}) \le \frac{8}{7}N^3. \quad (20)$$

i.e., at most 14% more memory is needed than for the input data. In the experiments of this and previous papers [12–14], the MIP projections $\mathcal{M}_\Theta$ have been implemented by means of the object order voxel projection method of Mroz et al. [10]. In this method, one loops through the volume, projects all voxels to the image plane with each voxel contributing to exactly one pixel, and accumulates values at pixel locations by maximum computation. The result is independent of the order in which the voxels are visited. An efficient volume data storage scheme is used, comprising 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. In our implementation, interesting voxels are simply defined as those with a nonzero grey value (zero voxel values never contribute to pixel maxima), which may lead to a large reduction in the amount of voxels to be processed.

When performing progressive refinement with a detail pyramid, the use of value-sorted arrays presents a problem. During synthesis, one must first do a one-level reconstruction of f_j from f_{j+1} which involves a 3-D dilation (see (16)). This dilation cannot be efficiently computed on data stored as value-sorted arrays. However, because of the large amount of zero voxels, the use of value-sorted arrays is more or less mandatory, since in practice it may save up to 95% of memory as compared to 3-D arrays. Therefore, it is most efficient to perform progressive refinement from an approximation pyramid (stored as as value-sorted arrays), instead of a detail pyramid. As long as no thresholding of detail data is used, storing the approximation data $f_0, f_1, \ldots, f_{L-1}, f_L$ does require memory storage of the same order of magnitude as storing the detail and coarsest approximation data $d_0, d_1, \ldots, d_{L-1}, f_L$.

The algorithm for the detail pyramid also applies to adjunction pyramids where detail signals and reconstructions are computed by using ordinary addition and subtraction. However, when the special operators of (8) are used, one may directly project all detail signals and refine in the image plane, see Section 3.2. This approach is computationally more efficient than that given in the general algorithm of Section 3.1, and will be used in Section 4. However, as far as approximation quality is concerned, there is no difference. More details and experimental results on computation time and memory usage for various data sets can be found in [12–14].

3.4. *Experimental Results*

We performed multiresolution MIP rendering using a 2-level decomposition with a $2 \times 2 \times 2$ structuring element. Larger structuring element sizes remove too much of the original data in higher levels of the pyramid. In the case of conditional dilation pyramids, a $3 \times 3 \times 3$ structuring element was used for the conditional dilations, in order to recover as much as possible of the original data in all directions.

Several types of pyramid are compared: the trivial, adjunction, and conditional dilation pyramid for several values of the iteration parameter n. Three data sets were used,[2] all of size $256 \times 256 \times 256$: *aneurism*, an X-ray scan of the arteries (with aneurism) of the right half of a human head (1% nonzero voxels); *neuron*, a confocal microscopy data set of a neuron (1.8% nonzero voxels), and *bonsai*, CT data of a bonsai tree (16.5% nonzero voxels), see Fig. 2. The sampling distance in the view plane was taken equal to the sampling distance of the original volume data. Only axial projection was used. For the case of the adjunction and Sun-Maragos pyramids, more details on sizes of approximation and detail data and behaviour of rendering times as a function of pyramid level can be found in [12, 14].

In Table 1 we show the relative L_1-error $\mathcal{E}^{(j)}$ between a level-j approximation image $\hat{\mathbf{M}}_\Theta^{(j)}$ $(j = 1, 2)$ and the

Table 1. Approximation error $\mathcal{E}^{(j)}$ for level $j = 1, 2$ for the trivial pyramid and conditional dilation pyramids with various values of the pyramid parameter n. Cases $n = 0$ and $n = 1$ correspond to the adjunction pyramid and Sun-Maragos pyramid, respectively.

	Aneurism		*Bonsai*		*Neuron*	
Pyramid type	$\mathcal{E}^{(2)}$	$\mathcal{E}^{(1)}$	$\mathcal{E}^{(2)}$	$\mathcal{E}^{(1)}$	$\mathcal{E}^{(2)}$	$\mathcal{E}^{(1)}$
$n = 0$	0.859	0.532	0.499	0.225	0.938	0.676
$n = 1$	0.639	0.366	0.282	0.125	0.851	0.547
$n = 5$	0.584	0.328	0.244	0.113	0.824	0.520
$n = 15$	0.546	0.311	0.230	0.109	0.804	0.504
$n = 25$	0.534	0.305	0.228	0.108	0.799	0.499
Trivial pyramid	0.523	0.284	0.207	0.098	0.808	0.478

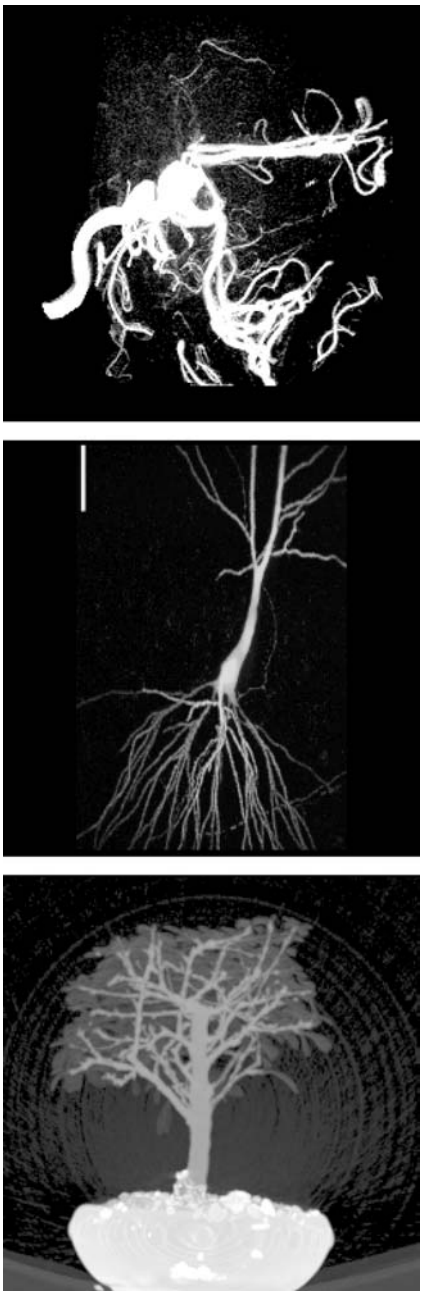

Figure 2. MIP rendering at full size of original data sets used in the experiments. Top: *aneurism*; middle: *neuron*; bottom: *bonsai*.

full image $\hat{\mathbf{M}}_\Theta^{(0)}$:

$$\mathcal{E}^{(j)} = \|\hat{\mathbf{M}}_\Theta^{(0)} - \hat{\mathbf{M}}_\Theta^{(j)}(f)\| \big/ \|\hat{\mathbf{M}}_\Theta^{(0)}\|. \qquad (21)$$

Figures 3–5 show successive approximations, for all pyramid types considered. To evaluate the results, we

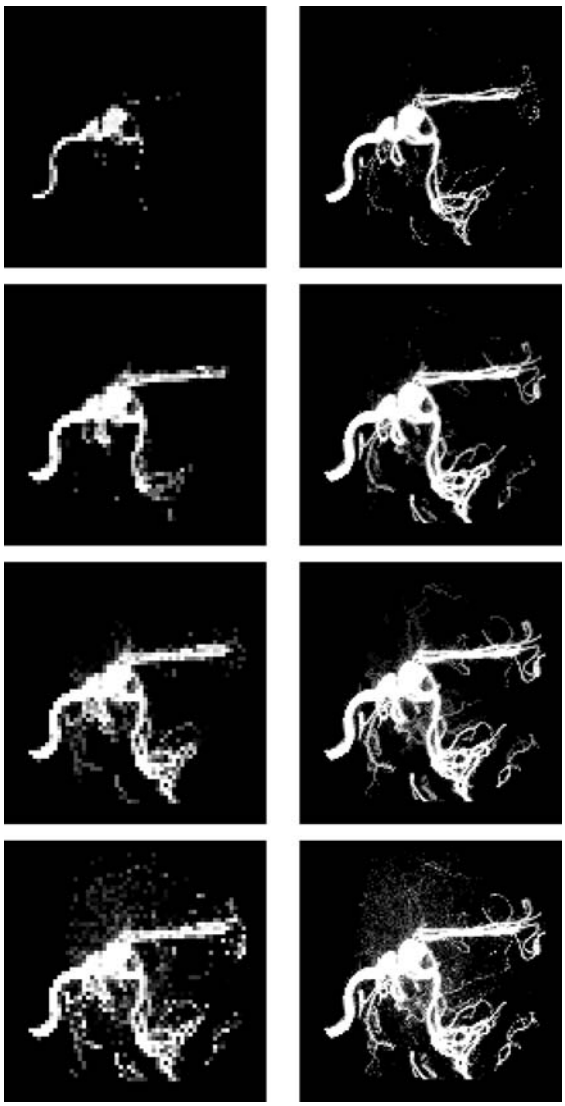

Figure 3. Multiresolution MIP reconstruction from a 2-level morphological pyramid, *aneurism* data. Left column: level-2 reconstruction; right column: level-1 reconstruction. First row: adjunction pyramid; second row: Sun-Maragos pyramid; third row: conditional dilation pyramid with $n = 25$; fourth row: trivial pyramid.

take two criteria into account. First, the size of the approximation error; second, the extent to which the noise present in the data is reduced. From the table, it is observed that conditional dilation pyramids considerably reduce the approximation error as compared to adjunction pyramids. For iteration numbers higher than 15, not much further improvement in the error is obtained. From visual inspection of the figures, it is clear that in higher levels of the adjunction pyramid small details are quickly removed. The conditional

Figure 4. Multiresolution MIP reconstruction from a 2-level morphological pyramid, *neuron* data. Left column: level-2 reconstruction; right column: level-1 reconstruction. First row: adjunction pyramid; second row: Sun-Maragos pyramid; third row: conditional dilation pyramid with $n = 25$; fourth row: trivial pyramid.

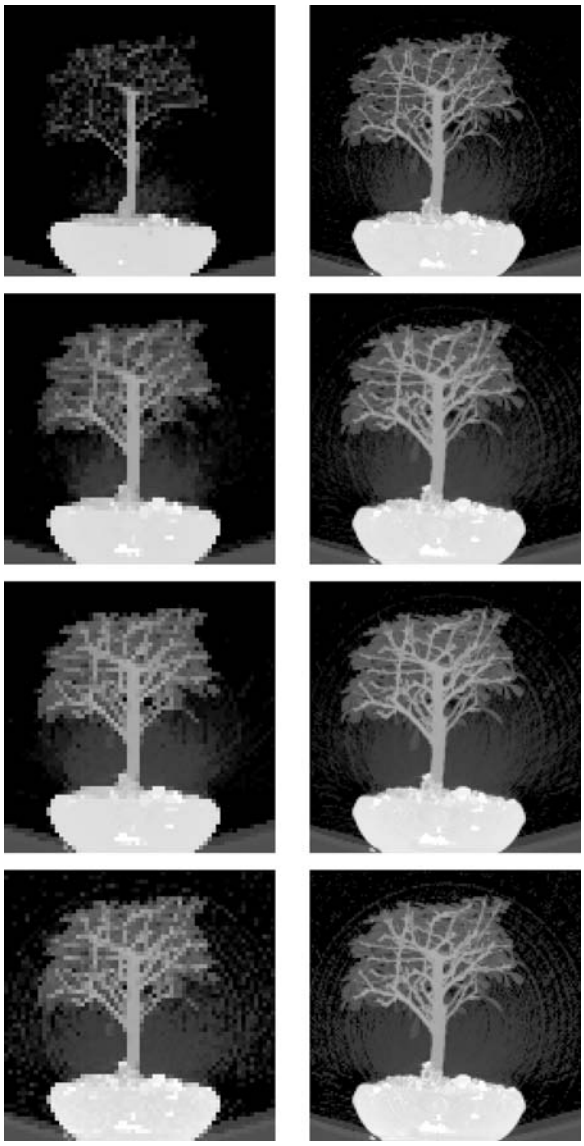

Figure 5. Multiresolution MIP reconstruction from a 2-level morphological pyramid, *bonsai* data. Left column: level-2 reconstruction; right column: level-1 reconstruction. First row: adjunction pyramid; second row: Sun-Maragos pyramid; third row: conditional dilation pyramid with $n = 25$; fourth row: trivial pyramid.

dilation pyramid preserves small details much better, especially for $n \geq 5$. The improvement for higher iteration numbers of the conditional dilation pyramid is especially noticeable for the aneurism data set, which contains many small elongated veins. Note that this accuracy improvement is possible with rendering times and memory usage comparable to those of adjunction pyramids. Noise reduction is significant in level-2 and level-1 approximations of the adjunction and conditional dilation pyramids (compare to the full-size MIP in Fig. 2). The trivial pyramid has the smallest errors, which may seem obvious since it retains most of the input data. However, it does not reduce the noise. In fact, for volume data with many isolated noise voxels which survive the downsampling step in the analysis phase, the error may become large, since in the synthesis phase

these voxels will be expanded by the structuring element. It would of course be possible to carry out a noise reduction step of the original data, before applying the morphological pyramids. But this would still leave us with a limit on the accuracy, since one can never do better than a level-1 approximation (unless one does a full reconstruction). What is really needed is a method with continuous error control. Such a method is presented in the next section.

4. Streaming MIP-Splatting

A major problem of the approaches presented above is that the approximation quality is completely determined by the discrete levels of the pyramid. For example, in a pyramid with $L = 2$ levels, the level-2 approximation may be too coarse, and the level-1 approximation too fine for meeting a certain upper bound on computation time. Therefore, we develop in this section a new approach, which allows to a priori fix the desired approximation error and/or computation time in a more or less continuous way. This approach is called *streaming MIP-splatting*, after the analogous approach in the linear case of wavelet splatting [8].

Our approach is to consider detail coefficients of all levels simultaneously, and order these coefficients in the same way as an auxiliary set of detail coefficients, which are sorted with respect to decreasing magnitude. After the resorting process, these auxiliary coefficients are no longer used.

4.1. Construction of the Pyramid

Again we start from the adjunction pyramid, cf. Section 2.1.1, with the generalized addition being defined as the maximum operation, and the generalized subtraction by (8). Recall that in this case the following multiresolution representation holds:

$$f = \psi_A^{\downarrow L}(f_L) \vee \bigvee_{k=0}^{L-1} \psi_A^{\downarrow k}(d_k). \tag{22}$$

Note that in volume rendering, f_L and d_k are 3-D data arrays.

As discussed in Section 3.3, the representation (22) allows one to project the elements of the approximation and detail coefficients in any order on the image plane. It is not necessary to do this projection level by level, as we have done so far. The novel idea then is to join the entries of the 3-D detail coefficient arrays of all levels $k = 0, 1, \ldots, L-1$ into a one-dimensional coefficient array, and sort these detail coefficients with respect to decreasing magnitude. Then, in the rendering phase, one would start projecting all detail coefficients, largest first, until a certain desired accuracy has been obtained.

Recall that in order for (22) to hold, the detail signals should be defined by:

$$\begin{aligned} d_j(n) &= f_j(n) \dot{-} \psi_A^{\downarrow}(f_{j+1})(n) \\ &= f_j(n) \dot{-} \psi_A^{\downarrow} \psi_A^{\uparrow}(f_j)(n) \geq 0. \end{aligned} \tag{23}$$

The problem we have to overcome is that the detail coefficients in the arrays $\{d_k\}$ are not small in regions where the structuring element does fit approximately, but not exactly, to the data, cf. Section 2.1.1.

To address this problem, we note that instead of the special subtraction $\dot{-}$, we can use ordinary subtraction to define a related set of detail signals, which we denote by $\bar{d}_j$:

$$\begin{aligned} \bar{d}_j(n) &= f_j(n) - \psi_A^{\downarrow}(f_{j+1})(n) \\ &= f_j(n) - \psi_A^{\downarrow} \psi_A^{\uparrow}(f_j)(n) \geq 0. \end{aligned} \tag{24}$$

The analysis and synthesis operators $\psi_A^{\downarrow}$ and $\psi_A^{\uparrow}$ are not changed. This results in a pyramid $\bar{d}_0, \bar{d}_1, \ldots, \bar{d}_{L-1}$, f_L. Note that the approximation signal f_L is independent of the choice of the difference operator. When the initial signal f is a 3-D array of size $S = N_1 \times N_2 \times N_3$, then d_j and $\bar{d}_j$ are 3-D arrays of size $8^{-j}\, S$.

In contrast to the original detail signals, the elements of the auxiliary detail signals $\bar{d}_0, \bar{d}_1, \ldots, \bar{d}_{L-1}$ are small when the structuring element approximately fits the data. We sort these elements with respect to magnitude and put them into one large 1-D array $\bar{D}$ with elements $\bar{D}_1, \ldots, \bar{D}_M$, where M is bounded by the sum of the sizes of the detail arrays $\{\bar{d}_j\}$, see (20). In practice, we will delete any zero entries during the sorting process, so that the actual value of M is much smaller. Note that an element of $\bar{d}_j$ is zero exactly when this is the case for the corresponding element of d_j.

After the sorting step, the entries of the array $\bar{D}$ are no longer ordered by level. From the initial dimensions N_1, N_2, N_3, and the index k of each element $\bar{D}_k$, the original level j and 3-D position (x, y, z) of the corresponding detail coefficient can be easily reconstructed. That is, there is a 1-1 mapping π, such that $k = \pi(x, y, z, j)$ and $(x, y, z, j) = \pi^{-1}(k)$, and $\bar{D}_k = \bar{d}_j(x, y, z)$.

As a final step, we replace the value of each entry of the array $\bar{D}$ by the corresponding entry of the original

detail signals:

$$\bar{D}_k \leftarrow d_j(x, y, z) \quad \text{where } (x, y, z, j) = \pi^{-1}(k). \quad (25)$$

4.2. Rendering the Pyramid

As a result of the construction phase, we have obtained an ordered list of detail coefficients $d_j(x, y, z)$, which can be used to compute the MIP of the input data. By construction, the order is such that each successive coefficient, when taken into account, maximally reduces the L_1-error between the partial reconstruction and the original data.

In the MIP process, all sorted coefficients are projected successively, until a desired accuracy of the resulting MIP image is obtained. When a coefficient k is projected, its value $val = \bar{D}_k$ is compared to the current value *curval* at the point of projection in the image plane, and *curval* is overwritten when $curval < val$. In addition, a local dilation of size j has to be carried out for coefficients with level $j > 0$, directly after projection. The reason is that, because of the resorting, detail coefficients with varying scale index j are successively projected. So we cannot do the dilation globally, as in the case of the level-by-level projection, where the scale index is constant per level. As before, let $\tilde{A}$ denote the projection of the 3-D structuring element A on the view plane, and let $j \cdot \tilde{A}$ denote this structuring element scaled by a factor of j. Then, when the level of the coefficient is j, all pixels in the neighborhood $j \cdot \tilde{A}$ around the point of projection are overwritten by val when their current value is smaller than val (this formulation applies for flat structuring elements).

When all entries of the sorted array $\bar{D}$ are processed, the exact MIP of the input volume is obtained. To compute approximations, one may stop projecting detail coefficients when a maximal index $K_{\max}$ of the array $\bar{D}$ has been reached. This limit $K_{\max}$ can be determined in various ways. It may be chosen so that an a-priori chosen error threshold is reached. Or, when computational efficiency is the major issue, one may simply choose $K_{\max}$ directly, as it determines the total number of coefficients to be projected, and thus the required computation time.

4.3. The Streaming MIP-Splatting Algorithm

The multiresolution MIP algorithm for streaming MIP-splatting can be summarized as follows.

– *Preprocessing*.

1. Compute an L-level 3-D morphological adjunction pyramid of the volume data f, resulting in a sequence of volumes $d_0, d_1, \ldots, d_{L-1}, f_L$. Here f_L is a signal at the coarsest level, and the details signals are defined by (23).
2. Compute auxiliary detail signals $\bar{d}_0, \bar{d}_1, \ldots, \bar{d}_{L-1}$ by (24), sort their entries with respect to magnitude (deleting zero entries), and finally use (25) to replace each entry of the resulting list by the corresponding original detail coefficient.

– *Actual MIP volume rendering*. For a given orientation $\boldsymbol{\Theta}$ of the viewing coordinate system, and a given maximum size $K_{\max}$, do:

1. Compute a low resolution approximation $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L)}(f)$ by first applying the voxel projection operator $\mathcal{M}_{\boldsymbol{\Theta}}$ to f_L, followed by the 2-D synthesis operator $\psi_{\tilde{A}}^{\downarrow L}$:

$$\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L)}(f) = \psi_{\tilde{A}}^{\downarrow L}(\mathcal{M}_{\boldsymbol{\Theta}}(f_L)). \quad (26)$$

   Here $\tilde{A} = \mathcal{M}_{\boldsymbol{\Theta}}(A)$ is a 2-D structuring element, which is obtained as the projection of the 3-D structuring element A on the view plane.
2. Refine the image progressively. For $k = 0, 1, \ldots, K_{\max}$, do:
   - Reconstruct the position and level corresponding to index k of array $\bar{D}$ by the mapping $(x, y, z, j) = \pi^{-1}(k)$.
   - Project the coefficient (x, y, z, j) to the correct position in the image plane (which already contains the level L approximation $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L)}(f)$), and carry out a local dilation of scale j by replacing all pixel values in a neighborhood $j \cdot \tilde{A}$ by $\bar{D}_k$ when their current value is smaller than $\bar{D}_k$.

   This results in an approximation which we denote by $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(L, K_{\max})}(f)$.

4.4. Experimental Results

In Table 2 we present the L_1-approximation error $\mathcal{E}^{(2, frac)}$ as a function of the percentage $frac$ of detail coefficients used. That is, $frac = K_{\max}/size(\bar{D})$, where $size(\bar{D})$ is the total number of entries in $\bar{D}$, i.e., the total number of original detail coefficients with zero ones deleted. The same data sets as before are used,

Table 2. L_1-approximation error $\mathcal{E}^{(2,frac)}$ for streaming MIP-splatting with various values of the fraction *frac* of detail coefficients used.

frac	*Aneurism*	*Bonsai*	*Neuron*
0.0	0.859	0.449	0.938
0.1	0.331	0.165	0.127
0.2	0.197	0.077	0.073
0.3	0.129	0.041	0.053
0.4	0.087	0.027	0.037
0.5	0.057	0.014	0.029
0.6	0.034	0.008	0.023
0.7	0.017	0.003	0.014
0.8	0.007	0.001	0.009
0.9	0.002	0.000	0.005
1.0	0.000	0.000	0.000

Table 3. L_1-approximation error for streaming MIP-splatting. The fraction of detail coefficients taken into account is determined by $frac = size(d_1)/size(\bar{D})$. The last row contains the corresponding level-1 error $\mathcal{E}^{(1)}$, taken from Table 1, case $n = 0$.

Data set	*frac*	$\mathcal{E}^{(2,frac)}$	$\mathcal{E}^{(1)}$
Aneurism	0.045	0.497	0.532
Bonsai	0.087	0.173	0.225
Neuron	0.034	0.301	0.676

with the same settings, i.e., a 2-level decomposition with a $2 \times 2 \times 2$ flat structuring element, and axial projection. When $frac = 0$, the error corresponds to that of a level-2 approximation, compare to the error $\mathcal{E}^{(2)}$ for $n = 0$ in Table 1. When $frac = 1$, all data have been taken into account and the error is zero.

For comparison with the earlier algorithm for an adjunction pyramid, we also present results on approximation error in Table 3 for the value of *frac* determined by $size(d_1)/size(\bar{D})$, where $size(d_1)$ is the number of nonzero level-1 detail coefficients. In this case, exactly the same number of detail coefficients is used as for a level-1 approximation in the algorithm which projects level by level (compare to $\mathcal{E}^{(1)}$ for $n = 0$ in Table 1).

We observe that the error of streaming MIP-splatting is smaller than that of a usual level-1 MIP. In the case of the *Neuron* data, the difference is substantial, a phenomenon which we also observed for other data sets (data not shown here). Note also that the fraction of detail coefficients taken into account is quite small (less than 1%). The streaming algorithm has the advantage that the fraction of included detail coefficients can be chosen at will, depending on the computation time one wants to spend or the maximal error one allows.

For visual inspection, we also give the corresponding MIP-images in Fig. 6. When we compare these images to the corresponding level-1 MIP images in Figs. 3–5, we observe that the streaming results are much better, with more fine details preserved (see also the full-size MIP images in Fig. 2). This obviously is due to the property of the streaming algorithm that important level-0 detail coefficients will get a small index k in the resorting process, and thus are included during partial reconstruction.

Another important observation we can make is that use of the L_1-error has a quite limited value. Although the errors of the level-by-level algorithm and the streaming algorithm in Table 3 are comparable, the corresponding images are perceptually quite different. Of course, this limitation is well known in image processing and computer vision, and active research is still going on to find better error metrics. Although some other metrics have been developed, they are often very hard to compute. Inclusion of these metrics is beyond the scope of this paper.

5. Discussion

We have surveyed and extended nonlinear multiresolution signal decompositions based on morphological pyramids for MIP volume rendering with progressive refinement and perfect reconstruction. Such pyramids combine the feature extraction capabilities of morphological operators with the acceleration gained by rendering multiresolution data. The morphological operators used in constructing these pyramids remove spatial details of size proportional to 2^j, where j is the level of the pyramid. However, in contrast to linear multiresolution approaches, such as those based on wavelets, no smoothing of data takes place. Typical use of such pyramids is in preview mode, where the user is continuously changing the viewpoint. For such interactive display, some loss in image quality is generally acceptable. When interaction ceases, details of the data can then be successively taken into account to quickly generate a high resolution view.

Our main object of interest in this paper was to see how the approximation error and persistence of small object features depend on pyramid level, for the different pyramid types. To improve the approximation

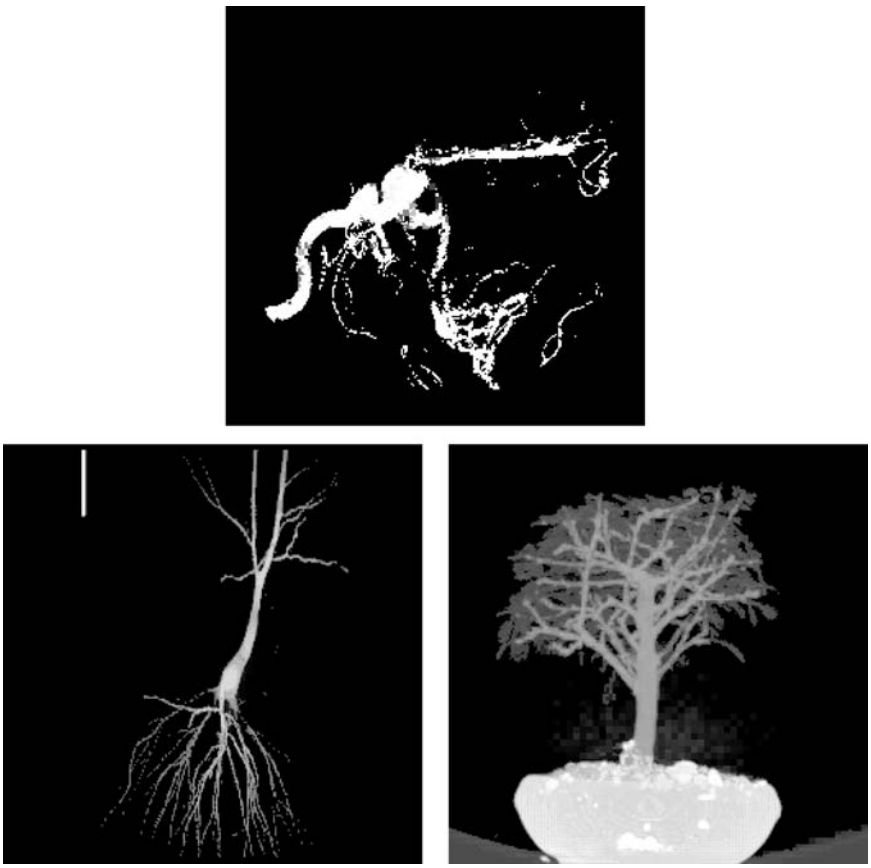

Figure 6. Streaming MIP reconstruction from a 2-level morphological pyramid. The fraction of detail coefficients taken into account is determined by $frac = size(d_1)/size(\bar{D})$. For comparison, see the level-1 MIP images in Figs. 3–5 which use the same number of detail coefficients.

quality of visualizations from reduced data (higher levels of the pyramid), two new approaches were explored. First, the class of conditional dilation pyramids, involving connectivity enhancing operators, was introduced. As a reference, we used the adjunction pyramid and Sun-Maragos pyramid, which remove spatial details without regard to the connectivity properties of the volume data [12–14]. We found that conditional dilation pyramids reduce the relative L_1-error between the full-size MIP and the approximate MIP from reduced pyramid data as the number n of conditional dilations increases. This accuracy improvement is possible with rendering times and memory usage comparable to those of the adjunction pyramid. For values of n larger than 15 no further improvement was observed. Also, noise is reduced in higher levels of the pyramid, as indicated by enhanced visual quality. Nevertheless, the improvement for $n > 1$ is fairly limited. Only the time needed to create the pyramids rises significantly with increasing n, but this is less of a problem since this step is carried out only once as a preprocessing step, whereas rendering is usually performed many times for different view angles.

The second new approach, called streaming MIP-splatting, again starts from the adjunction pyramid. The difference with the original algorithm [12,14] is that detail coefficients of all levels are considered simultaneously. They are ordered in the same way as an auxiliary set of detail coefficients, which are sorted with respect to decreasing magnitude. These auxiliary coefficients are defined by using ordinary subtraction, whereas the detail coefficients to be used during rendering are defined by a special subtraction operator which allows the max-representation of Eq. (10).

During rendering, all ordered detail coefficients are projected successively, until a desired accuracy of the

resulting MIP image is obtained. We have shown that this method outperforms all earlier methods, both with respect to image quality with a fixed amount of detail data, and in terms of flexibility of controlling error or maximum computation time. This is due to the fact that an arbitrary fraction of detail coefficients may be chosen for computing an approximation, whereas the other pyramids all work level-by-level.

Appendix A: Validity of the Pyramid Condition for Conditional Dilation Pyramids

Here we prove that the conditional dilation pyramid as defined in Section 2.1 satisfies the pyramid condition. First, the following lemma is proved.

Lemma 2. *Consider a morphological pyramid with analysis operator* $\psi^\uparrow = \sigma^\uparrow \eta$ *and synthesis operator* $\psi^\downarrow = \delta_A \sigma^\downarrow$, *satisfying the following assumptions*:

1. η *is an anti-extensive operator*
2. $\eta \delta_A \geq \mathsf{id}$
3. *The structuring element A satisfies condition* (11).

Then the pyramid condition holds.

Proof: By Assumption 2 we have that $\psi^\uparrow \psi^\downarrow = \sigma^\uparrow \eta \delta_A \sigma^\downarrow \geq \sigma^\uparrow \sigma^\downarrow = \mathsf{id}$. On the other hand, from Assumption 1, $\psi^\uparrow \psi^\downarrow \leq \sigma^\uparrow \delta_A \sigma^\downarrow$. By Assumption 3, formula (12) holds, that is, $\sigma^\uparrow \delta_A \sigma^\downarrow = \mathsf{id}$. Hence we found that $\psi^\uparrow \psi^\downarrow \leq \mathsf{id}$ and that $\psi^\uparrow \psi^\downarrow \geq \mathsf{id}$, so $\psi^\uparrow \psi^\downarrow = \mathsf{id}$. □

This lemma can be used to show that the pair (5)–(6), with $\eta = R_A^{(n)}$ given by (14), satisfies the pyramid condition for each n. It is sufficient to show that the operator η satisfies Assumptions 1 and 2 of the lemma, since Assumption 3 was assumed to hold anyhow.

1. The operator $R_A^{(0)} = \varepsilon_A$ is anti-extensive, because (11) implies that $\mathbf{0} \in A$, and hence the erosion ε_A is anti-extensive. For $n > 0$, Eq. (14) trivially implies that $R_A^{(n)}(f) \leq f$. Hence $R_A^{(n)}$ is anti-extensive for all $n \geq 0$.
2. We prove by induction that assumption 2 holds. First, $R_A^{(0)}(\delta_A(f)) = \varepsilon_A(\delta_A(f)) \geq f$ since $\varepsilon_A \delta_A$ is a closing. Second, for $n > 0$

$$R_A^{(n)}(\delta_A(f)) = \delta_A(f) \wedge \delta_A\big(R_A^{(n-1)}(\delta_A(f))\big).$$

Applying the induction hypothesis, i.e. $R_A^{(n-1)}\delta_A \geq \mathsf{id}$, we find

$$R_A^{(n)}(\delta_A(f)) \geq \delta_A(f) \wedge \delta_A(f) = \delta_A(f).$$

Finally, (11) implies that $\mathbf{0} \in A$, and hence the dilation δ_A is extensive. Therefore $R_A^{(n)}(\delta_A(f)) \geq f$, and we are done.

Notes

1. Some results for non-flat structuring functions can be found in [14].
2. Available from http://www.volvis.org

References

1. P.J. Burt and E.H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Communications*, Vol. 31, pp. 532–540, 1983.
2. W. Cai and G. Sakas, "Maximum Intensity projection using splatting in sheared object space," *Computer Graphics Forum (Proc. Eurographics'98)*, Vol. 17, No. 3, pp. C113–C124, 1998.
3. R.A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *Computer Graphics (SIGGRAPH '88 proceedings)*, Vol. 22, No. 4, pp. 65–74, 1988.
4. J. Goutsias and H.J.A.M. Heijmans, "Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids," *IEEE Trans. Image Processing*, Vol. 9, No. 11, pp. 1862–1876, 2000.
5. R. Grosso and T. Ertl, "Biorthogonal wavelet filters for frequency domain volume rendering," in *Proceedings of Visualization in Scientific Computing '95*, J. van Wijk, R. Scateni, and P. Zanarini (Eds.), 1995.
6. H.J.A.M. Heijmans and J. Goutsias, "Multiresolution signal decomposition schemes. Part 2: morphological wavelets," *IEEE Trans. Image Processing*, Vol. 9, No. 11, pp. 1897–1913, 2000.
7. L. Lippert and M.H. Gross, "Fast wavelet based volume rendering by accumulation of transparent texture maps," *Computer Graphics Forum*, Vol. 14, No. 3, pp. 431–443, 1995.
8. L. Lippert, M.H. Gross, and C. Kurmann, "Compression domain volume rendering for distributed environments," in *Proc. Eurographics'97*, 1997, pp. 95–107.
9. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press: New York, 1998.
10. L. Mroz, A. König, and E. Gröller, "Maximum intensity projection at warp speed," *Computers & Graphics*, Vol. 24, pp. 343–352, 2000.
11. S. Muraki, "Volume data and wavelet transforms," *IEEE Computer Graphics and Applications*, Vol. 13, No. 4, pp. 50–56, 1993.
12. J.B.T.M. Roerdink, "Multiresolution maximum intensity volume rendering by morphological pyramids," in *Data Visualization 2001. Proc. Joint Eurographics—IEEE TCVG Symposium on Visualization*, D. Ebert, J.M. Favre, and R. Peikert (Eds.),

May 28–30, 2001, Ascona, Switzerland, Springer, Wien, New York, 2001, pp. 45–54.
13. J.B.T.M. Roerdink, "Comparison of morphological pyramids for multiresolution MIP volume rendering," in *Data Visualization 2002. Proc. Eurographics—IEEE TCVG Symposium*, D. Ebert, P. Brunet, and I. Navazo (Eds.), May 27–29, 2002, Barcelona, Spain, ACM: New York, 2002, pp. 61–70.
14. J.B.T.M. Roerdink, "Multiresolution maximum intensity volume rendering by morphological adjunction pyramids," *IEEE Trans. Image Processing*, Vol. 12, No. 6, pp. 653–660, 2003a.
15. J.B.T.M. Roerdink, "A new class of morphological pyramids for multiresolution image analysis," in *Geometry, Morphology, and Computational Imaging*, Vol. 2616 of *Lecture Notes in Computer Science*, T. Asano, R. Klette, and C. Ronse (Eds.), Springer, Wien: New York, 2003b, pp. 165–175.
16. J.B.T.M. Roerdink and M.A. Westenberg, "Wavelet-based volume visualization," *Nieuw Archief voor Wiskunde*, Vol. 17 (Fourth Series), No. 2, pp. 149–158, 1999.
17. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.
18. F.K. Sun and P. Maragos, "Experiments on image compression using morphological piramids," in *SPIE Conf. Vis. Comm. Im. Proc. IV*, Vol. 1199, 1989, pp. 1303–1312.
19. W. Sweldens, "Wavelets and the lifting scheme: A 5 minute tour," *Z. Angew. Math. Mech.*, Vol. 76 (Suppl. 2), pp. 41–44, 1996.
20. M.A. Westenberg and J.B.T.M. Roerdink, "Frequency domain volume rendering by the wavelet X-ray transform," *IEEE Trans. Image Processing*, Vol. 9, No. 7, pp. 1249–1261, 2000.
21. R. Westermann and T. Ertl, "A multiscale approach to integrated volume segmentation and rendering," in *Proc. Eurographics'97*, D. Fellner and L. Szirmay-Kalos (Eds.), Vienna, Vol. 16(3). 1997, pp. C-117–C-127.
22. K.J. Zuiderveld, A.H.J. Koning, and M.A. Viergever, "Techniques for speeding up high-quality perspective Maximum Intensity Projection," *Pattern Recognition Letters*, Vol. 15, pp. 507–517, 1994.

Jos B.T.M. Roerdink received his M.Sc. (1979) in theoretical physics from the University of Nijmegen, the Netherlands. Following his Ph.D. (1983) from the University of Utrecht and a two-year position (1983–1985) as a Postdoctoral Fellow at the University of California, San Diego, both in the area of stochastic processes, he joined the Centre for Mathematics and Computer Science in Amsterdam. There he worked from 1986–1992 on image processing and tomographic reconstruction. He was appointed associate professor (1992) and full professor (2003), respectively, at the Institute for Mathematics and Computing Science of the University of Groningen, where he currently holds a chair in Scientific Visualization and Computer Graphics. His current research interests include morphological image processing, biomedical visualization, neuroimaging and bioinformatics.