Multiresolution Maximum Intensity Volume Rendering by Morphological Adjunction Pyramids

Jos B.T.M. Roerdink

Institute for Mathematics and Computing Science
University of Groningen
P.O. Box 800, 9700 AV Groningen, The Netherlands
Tel. +31-50-3633939, Fax. +31-50-3633800
E-mail: roe@cs.rug.nl

July 11, 2001

Abstract

We describe a multiresolution extension to maximum intensity projection (MIP) volume rendering, allowing progressive refinement and perfect reconstruction. The method makes use of morphological adjunction pyramids. The pyramidal analysis and synthesis operators are composed of morphological 3-D erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis. In this case the MIP operator can be interchanged with the synthesis operator. This fact is the key to an efficient multiresolution MIP algorithm, because it allows the computation of the maxima along the line of sight on a coarse level, before applying a two-dimensional synthesis operator to perform reconstruction of the projection image to a finer level. For interpolation and resampling of volume data, which is required to deal with arbitrary view directions, morphological sampling is used, an interpolation method well adapted to the nonlinear character of MIP. The structure of the resulting multiresolution algorithm 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.

Keywords: Multiresolution signal decomposition, volume rendering, maximum intensity projection, morphological adjunction pyramids.

1 Introduction

Volume rendering is a technique to produce two-dimensional images of three-dimensional data from different viewpoints, using advanced computer graphics techniques such as illumination, shading and colour. Although computing power is rapidly increasing, interactive rendering of volume data is still a demanding problem due to the sizes of the data sets, which may easily comprise several tens of megabytes. Also, the desire to exchange volume data through the internet has created a need for fast and efficient methods of transfer and display. For this purpose multiresolution models are developed, allowing systematic decomposition of the data into versions at different levels of detail, which can be used to visualize data incrementally as it arrives ('progressive refinement'), thus improving user interaction.

A well-known volume rendering method [5] is *X-ray volume rendering*, which is based upon integrating the 3-D data along the line of sight. An extensively studied class of multiresolution models in X-ray volume rendering is based on wavelets [7, 16, 26]. Recent developments include wavelet splatting [11,12], which extends splatting [27] by using wavelets as reconstruction filters, and Fourier-wavelet volume rendering [20, 24], which is a generalization of standard Fourier volume rendering [14], and uses a frequency domain implementation of the wavelet transform.

The goal of this paper, which extends some preliminary results in [17], is to present a multiresolution algorithm for maximum intensity projection (MIP) volume rendering, where one computes the *maximum*, instead of the integral, along the line of sight. Because of its computational simplicity, MIP is widely used in the display of magnetic resonance angiography (MRA) and ultrasound data. Our approach is based on the concept of *morphological pyramids*, following recent work of Goutsias and Heijmans [6,9], who present a general framework for multiresolution signal decomposition which includes linear wavelet analysis as a special case. Here, 'signals' are to be understood as multidimensional data, with image and volume data as special cases. 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.

For the case of maximum intensity projection, the class of *adjunction pyramids* [6] where one uses erosion for pyramid analysis and dilation for pyramid synthesis is particularly appropriate, because in this case the MIP operator can be interchanged with the synthesis operator. This fact is the key to an efficient MIP algorithm, because it allows to compute the maxima along the line of sight on a coarse level (where the size of the data is reduced), before applying a two-dimensional synthesis operator to perform reconstruction of the projection image to a finer level. Also, adjunction pyramids allow global error control, since they have the property that the global approximation error decreases monotonically as detail signals are added. If the data are of integer type (bytes or shorts, depending on the dynamic range), then in contrast to the case of wavelet-based volume rendering, no floating point computations are required, but all operations are carried out as calculations on integers.

In [17] we derived a multiresolution representation for the case of *axial* projections, that is, projections where the axes of the viewing coordinate system are parallel to the axes of the original grid of volume data. Here we present a full discussion of the general case of non-axial projections, which requires reconstruction of a continuous function from discrete data before projection. To that end morphological sampling is used [8, 10], an interpolation method well adapted to the nonlinear character of MIP. Also, in [17] we restricted ourselves to *flat* adjunction pyramids, where minima and maxima are computed in a local neighbourhood of each voxel. Flatness in particular means that no new grey values are introduced in the analysis of a signal. Here we also study *non-flat* adjunction pyramids, which are useful when one wants to extract small features, such as small veins in angiographic data.

A study of morphological pyramids in volume rendering is useful for various reasons. A first motivation is that it is one of many possibilities for accelerating the rendering process. In the case of MIP, many methods already exist for that purpose, including distance encoding [29], splatting in sheared object space [2], or MIP at warp speed [15], which preprocesses the data to remove non-contributing voxels from the volume. In many of these methods, pyramids are used as auxiliary data structures. In contrast, 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. A second, and perhaps more important, reason for studying morphological methods in volume rendering, is the feature extraction capability of morphological operators. Morphological methods have a well-established mathematical basis [8, 21] and are widely used in image processing for filtering, segmentation (see [19] for a recent review) and shape analysis. Applications of morphological methods in visualization have so

far mostly been restricted to preprocessing of volume data, but this is beginning to change now. For example, Lürig and Ertl [13] 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 [18]. We believe that the possibility of morphological methods to integrate feature extraction within the volume rendering process has great potential and deserves to be explored further.

The organization of this paper is as follows. Section 2 gives a few preliminaries on morphological operators, and reviews morphological pyramids. In particular, the class of adjunction pyramids is described. Section 3 then studies multiresolution maximum intensity projection based on morphological adjunction pyramids, and a multiresolution MIP rendering algorithm (MMIP) allowing progressive refinement is derived. For clarity of exposition, the case of axial projections is considered first, and then the analysis is extended to arbitrary view directions. Both flat and non-flat adjunction pyramids are considered. Experimental results are given in section 4. Section 5 contains a summary and discussion of future work.

2 Morphological pyramids

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

2.1 Morphological operators

Morphological operations for grey value images have been defined in analogy with the binary case [22]. For a mathematical treatment, see e.g. [8]. Consider signals in a d-dimensional signal space V_0, where 'signal' is interpreted as 'image' when $d = 2$ and 'volume' when $d = 3$. The structure of the signal space V_0 is assumed to be a complete lattice $\mathrm{Fun}(\mathbb{Z}^d, \mathcal{T})$ of functions on the discrete grid $\mathbb{Z}^d$, taking values in another complete lattice $\mathcal{T}$. For example, $\mathcal{T}$ may be the set of reals or integers. $\mathrm{Fun}(\mathbb{Z}^d, \mathcal{T})$ is itself a complete lattice, where the ordering relation is inherited from $\mathcal{T}$:

$$f \leq g \iff f(n) \leq g(n), \quad n \in \mathbb{Z}^d, f, g \in \mathrm{Fun}(\mathbb{Z}^d, \mathcal{T}). \tag{1}$$

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

$$\begin{aligned} \delta_A(f)(x) &= \bigvee_{y \in A, x-y \in F} f(x-y), \\ \varepsilon_A(f)(x) &= \bigwedge_{y \in A, x+y \in F} f(x+y). \end{aligned} \tag{2}$$

So dilation and erosion simply replace the value at a given point by the maximum or minimum in a neighbourhood defined by the structuring element A. By taking products of dilation and erosion one obtains the *opening* and *closing*. The opening $\alpha_A(f)$ and closing $\phi_A(f)$ of f by A are defined by

$$\alpha_A(f)(x) = \delta_A \varepsilon_A(f)(x), \qquad \phi_A(f)(x) = \varepsilon_A \delta_A(f)(x). \tag{3}$$

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.

2.2 Multiresolution signal decomposition

Here an outline is given of the basics of multiresolution signal decomposition, as recently introduced by Goutsias and Heijmans [6]. This scheme encompasses linear (e.g. laplacian) and nonlinear pyramid schemes. In particular, the class of morphological adjunction pyramids is considered, and a multiresolution representation is derived which forms the basis of the multiresolution maximum intensity projection algorithm of Section 3.

The goal is to decompose the original signal f which is an element of the signal space V_0 into a number of coarser signals $f_j, j = 0, 1, 2, \ldots$. Here j indicates the *level* of the decomposition. It is assumed that the signals $\{f_j\}$ are elements of associated signal spaces V_j, which all have the structure of the complete lattice $\mathrm{Fun}(\mathbb{Z}^d, \mathcal{T})$.

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

Definition 1 *The analysis and synthesis operators $\psi_j^{\uparrow}, \psi_j^{\downarrow}$ are said to satisfy the* pyramid condition *if $\psi_j^{\uparrow}\psi_j^{\downarrow} = \mathsf{id}$ on V_{j+1}. Here* id *denotes the identity operator, i.e.,* $\mathsf{id}f = f$.

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

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

In a level-L decomposition, a signal f results in a sequence d_0, d_1, $\ldots$, d_{L-1}, f_L, where $\{d_j\}$ are detail signals and f_L is a signal at the coarsest level. Here $\dot{-}$ is a generalized subtraction operator. Assuming there exists an associated generalized addition operator $\dot{+}$ such that

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

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

$$f_j = \psi_j^{\downarrow}(f_{j+1}) \mathbin{\dot{+}} d_j, \quad j = L-1, L-2, \ldots, 0. \tag{4}$$

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

In general, the pyramid representation d_0, d_1, $\ldots$, d_{L-1}, f_L of a signal f is redundant. A nonredundant representation results if one introduces a second family of analysis and synthesis operators to encode details signals, resulting in so-called *morphological wavelets* [9]. This representation is particularly useful if coding efficiency is important. In this paper, we restrict ourselves to morphological pyramids.

Approximations of a signal $f \in V_j$ are signals in V_j which are reconstructed from higher levels of the pyramid by omitting some of the detail signals. To make this notion precise, some notation is introduced first. The multilevel analysis operator

$$\psi_{i,j}^{\uparrow} = \psi_{j-1}^{\uparrow}\psi_{j-2}^{\uparrow}\cdots\psi_i^{\uparrow}, \qquad j > i$$

maps an element of V_i to an element of V_j. Similarly, the multilevel synthesis operator

$$\psi^{\downarrow}_{i,j} = \psi^{\downarrow}_i \psi^{\downarrow}_{i+1} \cdots \psi^{\downarrow}_{j-1}, \qquad j > i$$

maps an element of V_j back to an element of V_i. The operator

$$\hat{\psi}_{i,j} = \psi^{\downarrow}_{i,j} \psi^{\uparrow}_{i,j}$$

can be regarded as an *approximation operator* that maps the information obtained at level j by the analysis operator $\psi^{\uparrow}_{i,j}$ back to level i by the synthesis operator $\psi^{\downarrow}_{i,j}$.

Now define a level-j approximation $\hat{f}_{0,j}$ of $f \in V_0$ by

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

2.3 Adjunction pyramids

We now describe the class of *morphological adjunction pyramids* [6], which satisfy the following assumptions:

1. $V_j = \text{Fun}(\mathbb{Z}^d, \mathcal{T})$, the space of grey-value functions, where the grey-value set $\mathcal{T}$ is a complete lattice.
2. The analysis and synthesis operators are independent of level: $\psi^{\uparrow}_j = \psi^{\uparrow}, \psi^{\downarrow}_j = \psi^{\downarrow}$.
3. $\psi^{\uparrow} : V_0 \to V_1$ and $\psi^{\downarrow} : V_1 \to V_0$ form an *adjunction* between V_0 and V_1. This means that $\psi^{\uparrow}$ is an *erosion*, i.e. commutes with infima, and $\psi^{\downarrow}$ is a *dilation*, i.e. commutes with suprema. In particular, the product operator $\psi^{\downarrow}\psi^{\uparrow}$ is an opening.

First *flat* pyramids are considered, where local minima and maxima are computed in a local neighbourhood $A \subseteq \mathbb{Z}^d$ of each point. Non-flat pyramids are considered below (see Section 3.4).

For a d-dimensional signal f, the analysis and synthesis operators of an adjunction pyramid have the form

$$\psi^{\uparrow}_A(f)(n) = \sigma^{\uparrow}\, \varepsilon_A(f)(n) = \bigwedge_{k \in A} f(2n + k) \tag{5}$$

$$\psi^{\downarrow}_A(f)(k) = \delta_A\, \sigma^{\downarrow}(f)(k) = \bigvee_{n \in A[k]} f\left(\frac{k-n}{2}\right) \tag{6}$$

Here $\delta_A(f)$ and $\varepsilon_A(f)$ are the grey value dilation and erosion with structuring element A, whereas $\sigma^{\uparrow}$ and $\sigma^{\downarrow}$ denote dyadic downsampling and dyadic upsampling in each spatial dimension:

$$\sigma^{\uparrow}(f)(n) = f(2n)$$

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

Also,

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

$$A[n] = A \cap \mathbb{Z}^d[n]$$

The sets $A[n]$ form a disjoint partition of A of at most 2^d nonempty and mutually disjoint subsets. So the analysis phase consists of erosion followed by downsampling; the synthesis phase consists of upsampling followed by dilation. We note in passing 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, 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 [6].

In order that the pyramid condition of Definition 1 is satisfied, a condition has to be put on A. It is shown in [6] that the pyramid condition 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, that is, when

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

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 $\psi_A^{\downarrow}\,\psi_A^{\uparrow}\,(f) \leq f$. In this case, the generalized addition and subtraction operators can be defined by [6]

$$t \dot{+} s = t \vee s \tag{8}$$

$$t \dot{-} s = \begin{cases} t, & \text{if } t > s \\ \bot, & \text{if } t = s \end{cases} \tag{9}$$

where $\bot$ is the least element of $\mathcal{T}$. From now on, we will assume that $\mathcal{T}$ is the set of nonnegative integers, which means that $\bot = 0$. As a consequence, the detail signals are nonnegative:

$$d_j(n) = f_j(n) \dot{-} \psi_A^{\downarrow}(f_{j+1})(n) = f_j(n) \dot{-} \psi_A^{\downarrow}\,\psi_A^{\uparrow}\,(f_j)(n) \geq 0. \tag{10}$$

Note that (9) implies that the detail signal $d_j(n)$ equals $f_j(n)$ except at points where $f_j(n) = \psi_A^{\downarrow}\,\psi_A^{\uparrow}\,(f_j)(n)$. So, detail signals are not 'small' in regions where the structuring function does not match the data well. Since our primary interest is in the approximations, not the detail signals, this is not a problem, but for compression purposes other choices of addition and subtraction operators are advantageous.

For an adjunction pyramid with the addition operator defined by (8), the reconstruction takes a special form. Making use of the fact that $\psi_A^{\downarrow}$ is a dilation, hence commutes with suprema, one derives

$$\begin{aligned} f &= d_0 \vee \psi_A^{\downarrow}(f_1) = d_0 \vee \psi_A^{\downarrow}(d_1 \vee \psi_A^{\downarrow}(f_2)) \\ &= d_0 \vee \psi_A^{\downarrow}(d_1) \vee \psi_A^{\downarrow}\,\psi_A^{\downarrow}\,(f_2)) \\ &\dots \\ &= d_0 \vee \psi_A^{\downarrow}(d_1) \vee {\psi_A^{\downarrow}}^2(d_2) \vee \ldots \vee {\psi_A^{\downarrow}}^{L-1}(d_{L-1}) \vee {\psi_A^{\downarrow}}^{L}(f_L), \end{aligned}$$

where L is the decomposition depth and ${\psi_A^{\downarrow}}^k$ denotes k-fold composition of $\psi_A^{\downarrow}$ with itself. More succinctly written, the following representation has been found:

$$f = {\psi_A^{\downarrow}}^{L}(f_L) \vee \bigvee_{k=0}^{L-1} {\psi_A^{\downarrow}}^{k}(d_k). \tag{11}$$

This representation is quite similar to the (linear) laplacian pyramid representation [1], see also [6]. The main difference is that sums have been replaced by maxima.

2.3.1 Example: 2-D Haar pyramid

A simple example of an adjunction pyramid is the morphological Haar pyramid, which can be viewed as a nonlinear generalization of the Haar wavelet well-known from linear wavelet theory [4]. It can be defined in any dimension. For example, the 2-D morphological Haar pyramid is defined by the structuring element $A = \{(0,0),(0,1),(1,1),(1,0)\}$, that is, its analysis and synthesis operators are given by [6]:

$$\begin{aligned}\psi_A^{\uparrow}(f)(m,n) =& f(2m,2n) \wedge f(2m,2n+1) \\ & \wedge f(2m+1,2n) \wedge f(2m+1,2n+1) \\ \psi_A^{\downarrow}(f)(2m,2n) = \psi_A^{\downarrow}(f)&(2m,2n+1) = \psi_A^{\downarrow}(f)(2m+1,2n) \\ & = \psi_A^{\downarrow}(f)(2m+1,2n+1) = f(m,n)\end{aligned}$$

So, $\psi_A^{\uparrow}$ computes the maximum in a 2×2 neighbourhood of a pixel, and downsamples by a factor of 2, whereas $\psi_A^{\downarrow}$ simply upsamples by a factor of 2. In this case

$$\begin{aligned}&A[(0,0)] = \{(0,0)\}, A[(0,1)] = \{(0,-1),(0,1)\} \\ &A[(1,0)] = \{(-1,0),(1,0)\},\end{aligned}$$

so condition (7) is satisfied for $a = (0,0)$. An example of application of the Haar pyramid is given in Fig. 1.

3 Maximum intensity projection

Now we address the main problem of this paper, which is the derivation of a multiresolution maximum intensity projection (MMIP) volume rendering algorithm with progressive refinement. In MIP volume rendering, one computes maxima along straight lines through the data volume. To allow multiresolution reconstruction, the morphological adjunction pyramids described above are applied.

For clarity of exposition, we first consider the case of *axial* projections, that is, projections where the axes of the viewing coordinate system are parallel to the axes of the original grid of volume data. That is, the direction of projection is parallel to the z-axis, and the x-y plane is the view plane. Subsequently, the extension to arbitrary viewing directions is discussed. This extension requires that the volume data are interpolated and resampled. For this purpose, morphological sampling is used [8, 10], an interpolation method well adapted to the nonlinear character of MIP.

3.1 Axial projections

Consider a discrete volume data set f, with values $f(k,l,m)$, where $(k,l,m) \in S \subseteq \mathbb{Z}^3$. Project voxels parallel to the z-axis by computing the maximum value. The result is denoted by $\mathcal{M}_{\mathbf{z}}(f)$:

$$\mathcal{M}_{\mathbf{z}}(f)(k,l) = \bigvee_m f(k,l,m).$$

Applying the pyramid representation (11), and the fact that $\mathcal{M}_{\mathbf{z}}$ evidently distributes over maxima, one gets

$$\mathcal{M}_{\mathbf{z}}(f) = \mathcal{M}_{\mathbf{z}}\left(\psi_A^{\downarrow\,L}(f_L)\right) \vee \bigvee_{k=0}^{L-1} \mathcal{M}_{\mathbf{z}}\left(\psi_A^{\downarrow\,k}(d_k)\right) \tag{12}$$

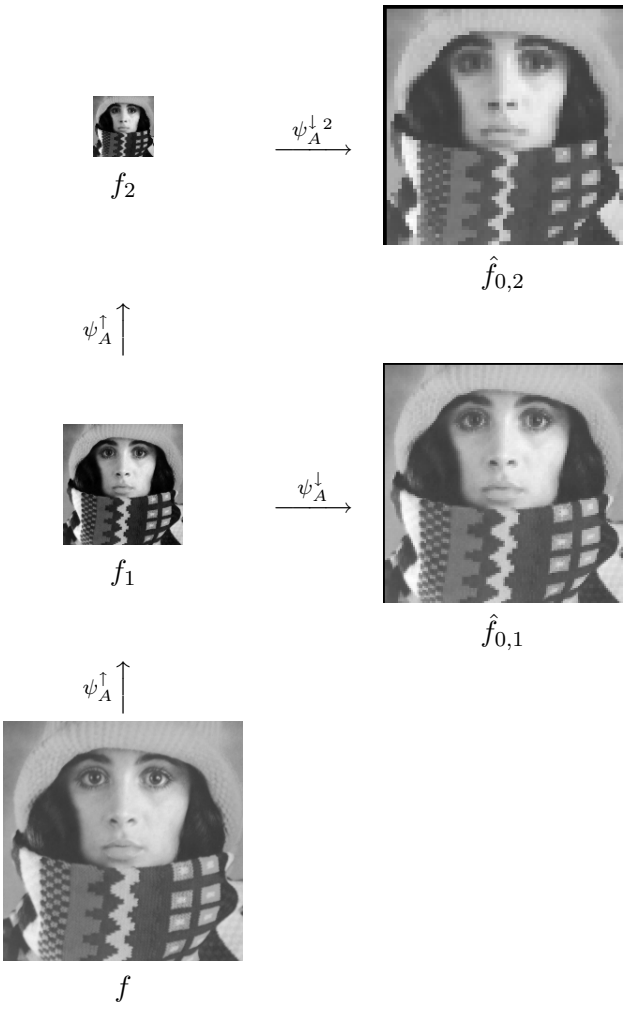


Figure 1: Signal decomposition and approximation from a 2-level morphological Haar pyramid.

Although, in principle, this formula allows us to do multiresolution MIP, this expression is computationally inefficient, because to compute the projections at a certain level of approximation j, one has to reconstruct f_L, d_{L-1}, $\ldots$, d_j first to full resolution and then apply the maximum operator $\mathcal{M}_{\mathbf{z}}$. What we really want is to 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 level. This is possible, as is shown next.

3.1.1 Computing the maxima before synthesis

According to (6), the synthesis operator $\psi_A^{\downarrow}$ is composed of upsampling, followed by a dilation. Therefore, our problem is to rewrite

$$\mathcal{M}_{\mathbf{z}}\, \psi_A^{\downarrow}\, (f) = \mathcal{M}_{\mathbf{z}} \delta_A\, \sigma^{\downarrow}(f)$$

such that the MIP operator $\mathcal{M}_{\mathbf{z}}$ is 'moved to the right'. The problem can be split in two parts. First the projection of a dilated function is considered, then the projection of an upsampled function, and finally the two results are combined.

First of all, both $\mathcal{M}_{\mathbf{z}}$ and δ_A involve the computation of maxima. Therefore, it is obvious that to compute $\mathcal{M}_{\mathbf{z}}\delta_A(f)$, one can first project f along the z-axis, and then dilate the resulting 2-D function by a flat 2-D structuring function whose support $\tilde{A}$ is the projection of A. That is,

$$\mathcal{M}_{\mathbf{z}}\, \delta_A(f) = \delta_{\tilde{A}}\, \mathcal{M}_{\mathbf{z}}(f), \tag{13}$$

with

$$\tilde{A} := \{(k,l) \in \mathbb{Z}^2 | (k,l,m) \in A \text{ for some } m \in \mathbb{Z}\}.$$

Note that δ_A is a 3-D dilation while $\delta_{\tilde{A}}$ is a 2-D dilation. If a set is identified with its indicator function, we can also write

$$\tilde{A} = \mathcal{M}_{\mathbf{z}}(A). \tag{14}$$

This form will be useful when considering arbitrary viewing directions and/or grey scale structuring functions.

Next, consider projection of an upsampled function: $\mathcal{M}_{\mathbf{z}}\, \sigma^{\downarrow}(f)$. Upsampling means inserting zeroes between neighbouring voxels in all three spatial dimensions. If the upsampled function is projected, then for those (k,l) 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 (k,l) 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 (k,l). Therefore,

$$\mathcal{M}_{\mathbf{z}}\, \sigma^{\downarrow}(f) = \sigma^{\downarrow}\, \mathcal{M}_{\mathbf{z}}(f), \tag{15}$$

where $\sigma^{\downarrow}$ on the right-hand side is a 2-D upsampling operator. Note that the dimension of $\sigma^{\downarrow}$ is not explicitly indicated, since this is clear from the dimension of the function on which it acts.

Now we can take the final step, which is to combine (13) and (15), yielding

$$\begin{aligned} \mathcal{M}_{\mathbf{z}}\, \psi_A^{\downarrow}\, (f) &= \mathcal{M}_{\mathbf{z}} \delta_A\, \sigma^{\downarrow}(f) = \delta_{\tilde{A}}\, \mathcal{M}_{\mathbf{z}}\, \sigma^{\downarrow}(f) \\ &= \delta_{\tilde{A}}\, \sigma^{\downarrow} \mathcal{M}_{\mathbf{z}}(f) = \psi_{\tilde{A}}^{\downarrow}\, \mathcal{M}_{\mathbf{z}}(f), \end{aligned} \tag{16}$$

where $\psi^{\downarrow}_{\tilde{A}} = \delta_{\tilde{A}}\,\sigma^{\downarrow}$ is a 2-D synthesis operator of the same form as $\psi^{\downarrow}_{A}$ (the structuring element A has only been replaced by a structuring element $\tilde{A}$). It is evident that a similar formula holds for projection of iterated versions ${\psi^{\downarrow}_{A}}^{k}$, $k > 1$.

As a result of the above analysis, we have proved the main result of this subsection, which is a multiresolution representation of the maximum intensity projection $\mathcal{M}_{\mathbf{z}}(f)$ of a 3-D discrete data set f for the case of axial projections:

$$\mathcal{M}_{\mathbf{z}}(f) = {\psi^{\downarrow}_{\tilde{A}}}^{L}\left(\mathcal{M}_{\mathbf{z}}(f_L)\right) \vee \bigvee_{k=0}^{L-1} {\psi^{\downarrow}_{\tilde{A}}}^{k}\left(\mathcal{M}_{\mathbf{z}}(d_k)\right) \tag{17}$$

This formula allows us to do multiresolution MIP progressively, starting from the coarsest signal f_L, and successively taking the detail signals d_k, $k = L-1, \ldots, 0$ into account as follows.

Define a level-L approximation $\hat{\mathcal{M}}_{\mathbf{z}}^{(L)}(f)$ by

$$\hat{\mathcal{M}}_{\mathbf{z}}^{(L)}(f) = {\psi^{\downarrow}_{\tilde{A}}}^{L}(\mathcal{M}_{\mathbf{z}}(f_L)),$$

and level-j approximations $\hat{\mathcal{M}}_{\mathbf{z}}^{(j)}(f)$, $j = L-1, \ldots, 0$, by

$$\hat{\mathcal{M}}_{\mathbf{z}}^{(j)}(f) = {\psi^{\downarrow}_{\tilde{A}}}^{L}\left(\mathcal{M}_{\mathbf{z}}(f_L)\right) \vee \bigvee_{k=j}^{L-1} {\psi^{\downarrow}_{\tilde{A}}}^{k}\left(\mathcal{M}_{\mathbf{z}}(d_k)\right). \tag{18}$$

Then, starting from $\hat{\mathcal{M}}_{\mathbf{z}}^{(L)}(f)$, (18) can be computed recursively by

$$\hat{\mathcal{M}}_{\mathbf{z}}^{(j-1)}(f) = {\psi^{\downarrow}_{\tilde{A}}}^{j-1}\left(\mathcal{M}_{\mathbf{z}}(d_{j-1})\right) \vee \hat{\mathcal{M}}_{\mathbf{z}}^{(j)}(f). \tag{19}$$

In particular, $\hat{\mathcal{M}}_{\mathbf{z}}^{(0)}(f) = \mathcal{M}_{\mathbf{z}}(f)$, the exact MIP of the original data f.

3.2 Arbitrary view directions

A simple method to produce MIP views for an arbitrary projection direction is *voxel projection*. This is a method where all voxels are projected on the view plane in arbitrary order, and each voxel v contributes only to the pixel p which is closest to the projection of the center of v on the view plane. The final value of pixel p is the maximum of the values of all voxels which project to p. A problem with this method is that holes may appear in the projection image for non-axial views. Holes are isolated pixels in the image plane to which no voxel projects because of undersampling, see Section 4.1 for an example. To deal with this problem, a continuous function has first to be reconstructed from the discrete data. The standard solution is to use some form of linear interpolation. However, linear interpolation does not combine well with the process of maximum computation. Therefore, morphological sampling is used instead, an interpolation method well adapted to the nonlinear character of MIP [8, 10]. The resulting MIP method will be a combination of voxel projection and morphological closing.

Let the direction of projection be defined by a unit vector $\boldsymbol{\theta}$, and define $\mathbf{u}$ and $\mathbf{v}$ to be two mutually orthogonal unit vectors perpendicular to $\boldsymbol{\theta}$, cf. Fig. 2. To fix the viewing coordinate system completely, two angles θ and ϕ are needed for defining the view vector $\boldsymbol{\theta}$, and a third angle α for defining the

orientation of the view plane with respect to the view vector $\boldsymbol{\theta}$. That is, let the triple $\mathbf{x}, \mathbf{y}, \mathbf{z}$ be the unit vectors of the Cartesian coordinate system. Then the triple $\mathbf{u}, \mathbf{v}, \boldsymbol{\theta}$ is the image of $\mathbf{x}, \mathbf{y}, \mathbf{z}$ under the composite rotation $R_{\mathbf{z}}(\phi)\, R_{\mathbf{y}}(\theta)\, R_{\mathbf{z}}(\alpha)$, where $R_{\mathbf{n}}(\beta)$ denotes the rotation over an angle β around the axis $\mathbf{n}$. That is, first the coordinate system is rotated around the z-axis over an angle α, then around the y-axis over an angle θ, and finally again around the z-axis over an angle ϕ. The three angles defining the viewing coordinate system will be collectively denoted by $\boldsymbol{\Theta} = (\theta, \phi, \alpha)$.

3.2.1 Single-scale projections

In appendix A, we show that the MIP projection process can be split in two phases: (i) voxel projection with maximum accumulation, and (ii) a final 2-D closing. This is analogous to the two-stage approach [25] developed for the so-called splatting method, which is a linear volume rendering technique based on the X-ray transform. In the first phase, all voxels in the volume data set $S \subseteq \mathbb{Z}^3$ are projected on the view plane $P \subseteq \mathbb{Z}^2$, such that a voxel with center (k, l, m) contributes to a pixel (i, j) if the projection of (k, l, m), which is denoted by $\pi_{\boldsymbol{\Theta}}(k, l, m)$, falls within a square $C(i, j)$ of size Δ_u by Δ_v with center (i, j), where Δ_u and Δ_v are the horizontal and vertical sampling distances in the view plane. Here $\boldsymbol{\Theta} = (\theta, \phi, \alpha)$ defines the orientation of the viewing coordinate system with respect to the world coordinate system. We will always take $\Delta_u = \Delta_v = \Delta$, where Δ equals the sampling distance of the original volume data, which are assumed to be sampled on a uniform Cartesian grid. Without loss of generality, we set $\Delta = 1$. The final pixel value is the maximum of the values of all voxels which project to the same pixel. In the second phase, a 2-D closing is performed on the image produced in phase 1. The resulting algorithm can be summarized as follows. The combined operator, which maps the discrete 3-D array f of volume data to a discrete 2-D array of pixel values, is denoted by $\mathbf{M}_{\boldsymbol{\Theta}}$.

1. Voxel projection with maximum accumulation:

$$\mathcal{M}_{\boldsymbol{\Theta}}(f)(i, j) = \bigvee_{(k,l,m) \in S} \{f(k, l, m) : \pi_{\boldsymbol{\Theta}}(k, l, m) \in C(i, j)\}, \tag{20}$$

2. 2-D closing ϕ_B of $\mathcal{M}_{\boldsymbol{\Theta}}(f)$ with structuring element B:

$$\mathbf{M}_{\boldsymbol{\Theta}}(f)(i, j) = \phi_B \left(\mathcal{M}_{\boldsymbol{\Theta}}(f)\right)(i, j), \quad (i, j) \in P.$$

The structuring element B depends on the 3-D neighbourhood used in the morphological interpolation, see the appendix for details. A simple choice is to take for B a structuring element of size 2×2 for all non-axial views. For axial projections, B reduces to a single pixel (i.e. the closing can be omitted); e.g. when $\mathbf{u} = \mathbf{x}, \mathbf{v} = \mathbf{y}, \boldsymbol{\theta} = \mathbf{z}$, $\mathbf{M}_{\boldsymbol{\Theta}}$ reduces to the axial projection $\mathcal{M}_{\mathbf{z}}$ defined in Section 3.1.

3.2.2 Multi-scale projections

Now we reconsider the multiresolution MIP projections discussed in Section 3.1, and extend the analysis to non-axial projections.

Starting point is again the discrete pyramid representation (11). The MIP operator $\mathbf{M}_{\boldsymbol{\Theta}}$ is composed of the voxel projection operator $\mathcal{M}_{\boldsymbol{\Theta}}$, and a closing ϕ_B. Clearly, $\mathcal{M}_{\boldsymbol{\Theta}}$ distributes over maxima, so

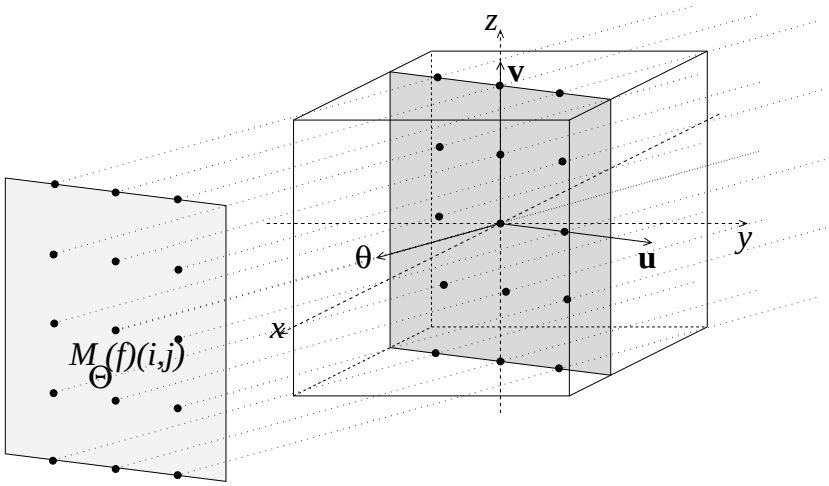


Figure 2: View plane defined by unit vectors $\mathbf{u}, \mathbf{v}$ orthogonal to the direction vector $\boldsymbol{\theta}$.

$$\mathbf{M}_{\mathbf{\Theta}}(f) = \phi_B \left(\mathcal{M}_{\mathbf{\Theta}} \left({\psi_A^{\downarrow}}^L (f_L) \right) \vee \bigvee_{k=0}^{L-1} \mathcal{M}_{\mathbf{\Theta}} \left({\psi_A^{\downarrow}}^k (d_k) \right) \right). \tag{21}$$

As before, the problem is to 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 level. The analysis in Section 3.1 leading to the commutativity relation (16) is easily generalized to arbitrary directions if one considers MIP projections of a continuous function f, with $\psi_A^{\downarrow}$ consisting of upsampling by a continuous factor σ followed by a dilation with a continuous structuring element A. Assuming this relation to hold in the discrete case as well, $\mathcal{M}_{\mathbf{\Theta}}$ is moved inside the $\psi_A^{\downarrow}$ operators, yielding

$$\mathbf{M}_{\mathbf{\Theta}}(f) = \phi_B \left({\psi_{\tilde{A}}^{\downarrow}}^L \left(\mathcal{M}_{\mathbf{\Theta}}(f_L) \right) \vee \bigvee_{k=0}^{L-1} {\psi_{\tilde{A}}^{\downarrow}}^k \left(\mathcal{M}_{\mathbf{\Theta}}(d_k) \right) \right), \tag{22}$$

where now (cf. (14))

$$\tilde{A} := \mathcal{M}_{\mathbf{\Theta}}(A). \tag{23}$$

This step involves a certain discretization error, which is assessed below experimentally (cf. Section 4.1).

Again we aim for progressive computation of (22). This requires that the additional closing ϕ_B in (22) can be handled 'recursively'. This is possible in view of the following property.

Lemma 2 *Let (ϵ, δ) be an adjunction on a lattice $\mathcal{L}$. Then the closing $\phi = \epsilon\delta$ satisfies*

$$\phi(f \vee g) = \phi(\phi(f) \vee g) = \phi(f \vee \phi(g)) = \phi(\phi(f) \vee \phi(g)).$$

Proof:

We prove only the first equality, the proof of the other two is similar. Using the fact that δ commutes with suprema and that $\delta = \delta\epsilon\delta$ (see [8]), one finds

$$\begin{aligned}\phi(f \vee g) &= \epsilon\delta(f \vee g) = \epsilon(\delta(f) \vee \delta(g)) = \epsilon(\delta\epsilon\delta(f) \vee \delta(g)) \\ &= \epsilon(\delta\phi(f) \vee \delta(g)) = \epsilon\delta(\phi(f) \vee g) = \phi(\phi(f) \vee g).\end{aligned}$$

∎

Define a level-L approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)$ by

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f) = \phi_B\left(\psi_{\tilde{A}}^{\downarrow\,L}(\mathcal{M}_{\mathbf{\Theta}}(f_L))\right),$$

and level-j approximations $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)$, $j = L-1, \ldots, 0$, by

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f) = \phi_B\left(\psi_{\tilde{A}}^{\downarrow\,L}(\mathcal{M}_{\mathbf{\Theta}}(f_L)) \vee \bigvee_{k=j}^{L-1} \psi_{\tilde{A}}^{\downarrow\,k}(\mathcal{M}_{\mathbf{\Theta}}(d_k))\right). \tag{24}$$

Then, using Lemma 2, (24) can be computed recursively by

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f) = \phi_B\left(\psi_{\tilde{A}}^{\downarrow\,j-1}(\mathcal{M}_{\mathbf{\Theta}}(d_{j-1})) \vee \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)\right). \tag{25}$$

Again, the recursion is initialized by $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)$ and terminates with $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(0)}(f)$, which equals the single-scale MIP $\mathbf{M}_{\mathbf{\Theta}}(f)$ of the original data f, up to discretization errors. For axial views the error will be exactly zero (cf. Section 4.1).

Remark 3 *In the case that the sampling distance in the view plane is equal to the sampling distance of the original volume data (as is the case in all the experiments reported below), holes in the projected images will only arise if the view vector $\boldsymbol{\theta}$ is along one of the grid axes of the volume data, but the view plane is rotated around the view vector. However, we have found that even when the closing ϕ_B is not necessary to prevent holes, it still diminishes the discretization error. Therefore the closing is applied for all non-axial views.*

3.3 The MMIP algorithm

The proposed multiresolution MIP algorithm can be summarized as follows.

1. *Preprocessing*. Compute an L-level 3-D morphological adjunction pyramid with structuring element A of the volume data, resulting in a sequence $d_0, d_1, \ldots, d_{L-1}, f_L$, where $\{d_j\}$ are detail signals and f_L is a signal at the coarsest level.
2. *Actual MIP volume rendering*. For a given orientation $\mathbf{\Theta}$ of the viewing coordinate system, do:
   - Compute a low resolution approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)$ by first applying the voxel projection operator $\mathcal{M}_{\mathbf{\Theta}}$ to f_L, followed by the 2-D synthesis operator $\psi_{\tilde{A}}^{\downarrow\,L}$ and a final closing ϕ_B:

     $$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f) = \phi_B\left(\psi_{\tilde{A}}^{\downarrow\,L}(\mathcal{M}_{\mathbf{\Theta}}(f_L))\right). \tag{26}$$

     Here $\tilde{A} = \mathcal{M}_{\mathbf{\Theta}}(A)$ is the voxel projection of the 3-D structuring element A, and B is a 2-D structuring element (for axial views, ϕ_B can be omitted).

- Refine the image progressively by taking the detail signals d_k, $k = L-1, \ldots, 0$ into account. From a level-j approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)$, compute an approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f)$ on level $j-1$ by projecting d_{j-1}, applying the 2-D pyramid synthesis operator $\psi_{\check{A}}^{\downarrow\, j-1}$ to the projection, taking the maximum of the image so obtained with the previous approximation, and finally applying a closing ϕ_B:

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f) = \phi_B \left(\psi_{\check{A}}^{\downarrow\, j-1}(\mathcal{M}_{\mathbf{\Theta}}(d_{j-1})) \vee \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f) \right). \tag{27}$$

- The recursion terminates with $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(0)}(f)$, which equals the MIP $\mathbf{M}_{\mathbf{\Theta}}(f)$ of the original data f (up to a discretization error, which is zero for axial projections).

As long as a user is interacting with the data, e.g. by continuously changing the viewing direction, only the coarsest approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)$ is used. When interaction ceases, finer approximations $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)$, $j < L$, are taken into account, so that the user can obtain reconstructions at higher levels of detail.

The structure of this algorithm is very similar to that of wavelet splatting [11, 12, 24]. The main differences are that (i) linear summation of voxel values is replaced by maximum computation, and (ii) linear wavelet filters are replaced by morphological filters.

3.3.1 Error decrease

From (25) one immediately deduces that $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f) \leq \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f)$, since from (10) the details signals d_{j-1} are nonnegative. So the projections increase pointwise as one goes down the pyramid:

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)(x) \leq \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L-1)}(f)(x) \leq \ldots \leq \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(0)}(f)(x). \tag{28}$$

Define a global approximation error by

$$\mathcal{E}^{(j)} = \|\mathbf{M}_{\mathbf{\Theta}}(f) - \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)\|,$$

where $\|\cdots\|$ is some error norm, such as the L_1, L_2 or L_∞ norm. Then (28) implies that the global approximation error decreases monotonically with decreasing j.

3.4 Grey scale structuring functions

Let the flat pyramid operators (5) and (6) be replaced by an analysis operator $\psi_a^{\uparrow}$ and synthesis operator $\psi_a^{\downarrow}$ involving grey-scale erosion and dilation by a structuring function $a \in \mathrm{Fun}(\mathbb{Z}^d, \mathcal{T})$,

$$\psi_a^{\uparrow}(f) = \sigma^{\uparrow}\, \varepsilon_a(f), \quad \psi_a^{\downarrow}(f) = \delta_a\, \sigma^{\downarrow}(f), \tag{29}$$

where $(\varepsilon_a, \delta_a)$ is a grey scale adjunction:

$$\delta_a(f)(x) = \bigvee_{y \in A, x-y \in F} f(x-y) + a(y),$$

$$\varepsilon_a(f)(x) = \bigwedge_{y \in A, x+y \in F} f(x+y) - a(y).$$

Here A and F denote the domain of a and f, respectively. Then, as shown in [6], the pyramid condition is still satisfied as long as (7) holds. Since in our case $\mathcal{T}$ is the lattice of nonnegative integers, we must ensure that the least element of the grey scale domain (i.e. the value zero) is conserved by the pyramid operators. This is the case if the structuring function a satisfies the requirements

$$\mathbf{0} \in A, \quad a(y) \leq a(\mathbf{0}) = 0 \text{ for all } y \in A,$$

where $\mathbf{0}$ is the origin of $\mathbb{Z}^d$. Under this condition, in the analysis step $\psi_a^{\uparrow}(f)$ will always be nonnegative since f itself is nonnegative. In the synthesis step the same holds: after the upsampling operation $\sigma^{\downarrow}$ the value at each point x is still nonnegative, and the final dilation δ_a does not change this. The reason is that, since $\mathbf{0} \in A$ and $a(\mathbf{0}) = 0$, the point x contributes its current —nonnegative— value during the local maximum computation around x. Therefore the outcome of $\psi_a^{\downarrow}(f)(x)$ can never be smaller than zero.

It is easy to show that in the case of non-flat structuring functions, the basic commutativity relation (16) still holds if the structuring element A is replaced by the structuring function a, and $\tilde{A}$ by $\tilde{a} = \mathcal{M}_{\mathbf{z}}(a)$. For non-axial views, we take $\tilde{a} = \mathbf{M}_{\boldsymbol{\Theta}}(a)$, which again will involve a discretization error.

3.5 Time and memory requirements

3.5.1 Time complexity

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. Of course, various acceleration schemes are possible, but these do not change the complexity [2, 15, 29].

For MMIP, there is first the preprocessing step, which has only to be executed once. The analysis operator involves 3-D erosions which are linear in the number of voxels, so preprocessing is $\mathcal{O}(N^3)$. Reconstruction is $\mathcal{O}(N^3)$ as well, since it is dominated by the projections of the volume data (the 2-D synthesis operators $\psi_{\tilde{A}}^{\downarrow j}$ and closing ϕ_B are $\mathcal{O}(N^2)$). The implementation of the morphological operators can also be accelerated by using structuring element decomposition techniques [3, 28]. But in our case, this is not really an issue, since only small sizes of the structuring element A are used (i.e., $M \times M \times M$, with $M = 2$ or $M = 3$).

3.5.2 Memory usage

For a volume data set of size N^3, the size of an L-level analysis pyramid is

$$N^3 \left(1 + \frac{1}{8} + \ldots + (\frac{1}{8})^L\right) = \frac{8}{7} N^3 \left(1 - (1/8)^{L+1}\right) \leq \frac{8}{7} N^3.$$

Hence, the pyramid takes at most 14% more memory than the input data. In principle, if morphological wavelet pyramids [9] are used, then this memory overhead can be reduced completely, but this requires the construction of new wavelets adapted to the special choice (8) of the addition/subtraction operations.

The MIP projections $\mathcal{M}_{\boldsymbol{\Theta}}$ required in the MMIP algorithm can be implemented by means of the object order voxel projection method of Mroz *et al.* [15]. 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 final result is independent of the order in which the voxels are visited. This 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. In the experiments to be discussed in Section 4, all levels of the pyramid were created and stored as value-sorted arrays. We define interesting voxels simply as those with a nonzero grey value (zero voxel values never contribute to pixel maxima). In practice, especially for angiographic data, a substantial reduction (sometimes more than 95%) in the amount of voxels to be processed is thus obtained. Further memory reductions are possible by using a preprocessing scheme to identify and remove other types of non-contributing voxels [15], and by applying compression techniques (thresholding, Huffman coding, etc.). For this so-called compression domain rendering in the case of wavelet splatting, see [12].

4 Experimental results

Multiscale MIP rendering experiments were carried out with a CT head data set and an MR angiography (MRA) data set, both of size 256^3, using a pyramid with two decomposition levels ($L = 2$). We used a PC with a 500 MHz Pentium III processor and 128 Mb memory. In the analysis and synthesis operations, dilations and erosions with a $2 \times 2 \times 2$ flat or non-flat structuring function 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%.

4.1 Multiscale discretization error

First we present some experimental results which give an indication of the multiscale discretization error due to morphological sampling which obtains when using the approximate equation (22). That is, the difference between $\mathbf{M}_{\boldsymbol{\Theta}}(f)$, the direct MIP projection of the original data, and $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(0)}(f)$ is computed according to the recursion (25). The relative error was defined as the L_1-norm of this difference, divided by the L_1-norm of $\mathbf{M}_{\boldsymbol{\Theta}}(f)$. The error will be a function of the angles θ, ϕ and α which define the viewing coordinate system, cf. Section 3.2. For axial projections this error will be exactly zero, because no morphological sampling is necessary.

The relative L_1-error was computed as a function of angle for the angiographic data set. Two cases were considered:

1. $\theta = \phi = 0$, α varying between 0 and π. This means that the projection direction is constant, i.e., $\boldsymbol{\theta}$ is parallel to the z-axis, but the view plane is rotating around $\boldsymbol{\theta}$.

2. $\phi = \alpha = 0$, θ varying between 0 and π, i.e., the view vector $\boldsymbol{\theta}$ is rotating around the y-axis.

For non-axial projections a closing ϕ_B was used with B a 2×2 structuring element. An example of holes arising when the closing is omitted is given in Fig. 4 for the MRA data set. In this example, the view vector $\boldsymbol{\theta}$ was along a grid axis, but the view plane was rotated over an angle of $\pi/4$ around $\boldsymbol{\theta}$, cf. Remark 3.

The results of the error computations are shown in Fig. 3. The left error curve is periodic with period $\pi/2$, since the uniform rectangular coordinate grid, which is rotating around the fixed projection vector $\boldsymbol{\theta}$, is invariant under rotation over $\pi/2$. The error is indeed zero at multiples of $\pi/2$. The maximum error remains below 1%, which is not noticeable in the MIP images. Also, we found no visually disturbing artifacts to occur.

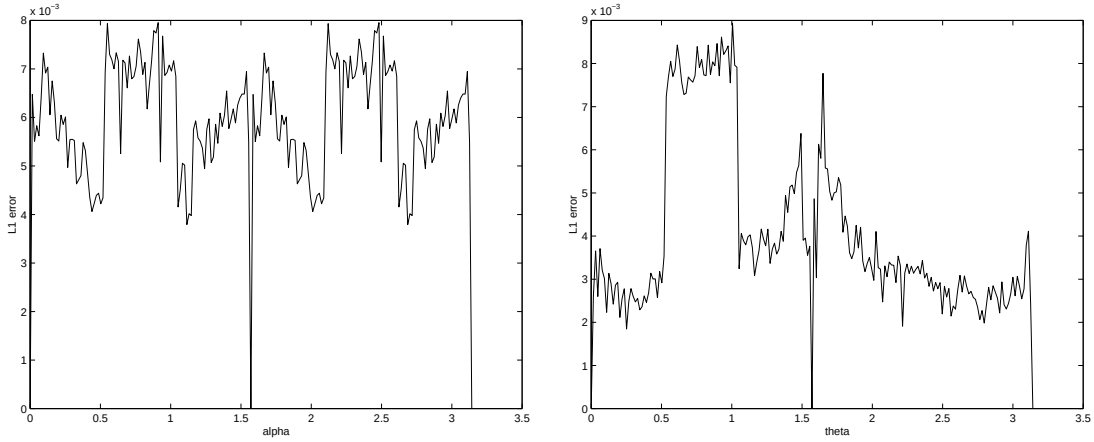


Figure 3: Multiscale discretization error $\|\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(0)}(f) - \mathbf{M}_{\boldsymbol{\Theta}}(f)\|_1 / \|\mathbf{M}_{\boldsymbol{\Theta}}(f)\|_1$ of the MRA data set as a function of rotation angle. Left: $\theta = \phi = 0$, $\alpha \in [0, \pi]$; right: $\phi = \alpha = 0$, $\theta \in [0, \pi]$.

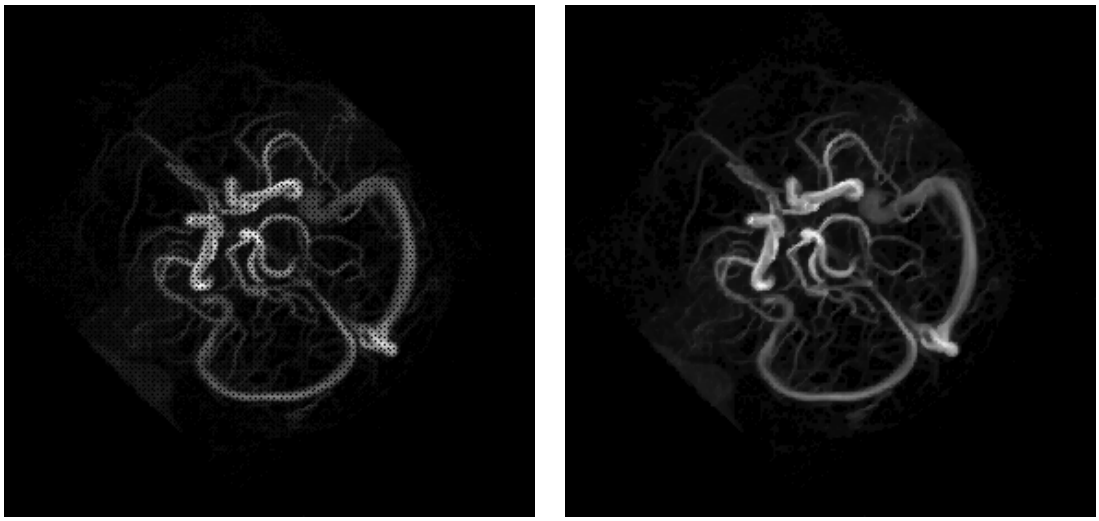


Figure 4: MIP projection of the MRA data for $\theta = \phi = 0, \alpha = \pi/4$, without (left) and with (right) additional closing with a 2×2 structuring element. Note the holes (regularly spaced isolated black dots) in the left image.

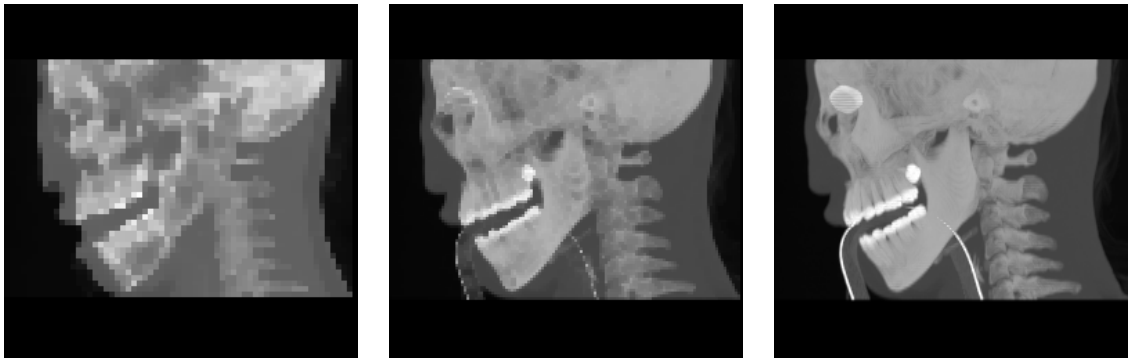
4.2 Rendering times and memory usage

Next, results on rendering times and memory usage for the CT and MRA data sets are presented. In both cases about 25 seconds was needed to create the pyramid. Sizes in value-sorted array format (see Section 3.5) 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. Rendering times were found to be almost independent of view angles. Figure 5 shows successive approximations for the CT data. Here, and in all cases below, axial projection was used.

Table 1: 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)
full image	838.5	0.423
level 2 approximation	0.812	0.110
add detail level 1	30.2	0.129
add detail level 0	801.6	0.417

CT data $256 \times 256 \times 256$	size (kbytes)	time (s)
full image	17433	6.92
level 2 approximation	253	0.20
add detail level 1	1861	0.87
add detail level 0	15171	6.04

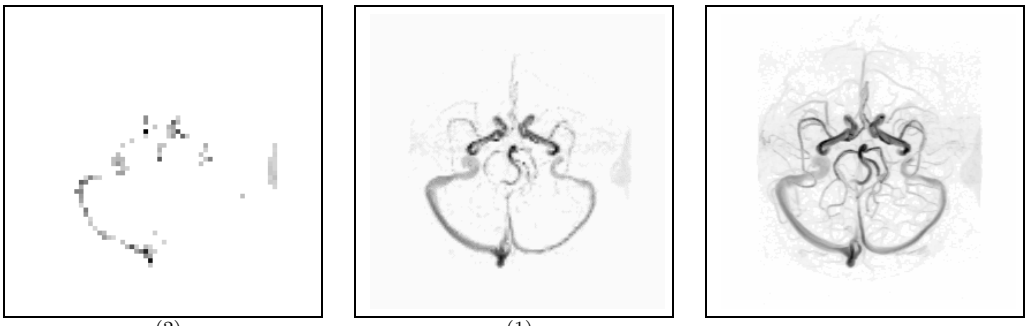


$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(2)}(f)$ $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(1)}(f)$ $\mathbf{M}_{\mathbf{\Theta}}(f)$

Figure 5: MMIP reconstruction from a 2-level morphological adjunction pyramid using a flat structuring function with a support of size $2 \times 2 \times 2$. From left to right: approximations on level 2, level 1 and level 0 (original).

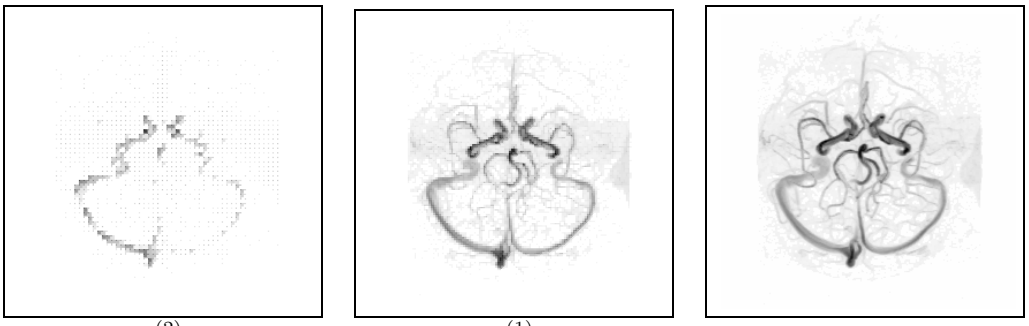
4.3 Non-flat adjunction pyramids

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. 5 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. One possibility is to use non-flat structuring functions, which can probe fine details more accurately. As an example, we computed MIP projections of the MRA data, both with a flat and with a non-flat structuring function with a support of size $2 \times 2 \times 2$. In the non-flat case, the structuring function had a peak of size 25. The results are shown in Fig. 6 and Fig. 7, respectively. For enhanced display purposes, we show the images in reverse-video mode (high intensity corresponding to low grey value). Clearly, the level-1 result where a non-flat structuring function is used contains more small details compared to the case with a flat structuring function, but in level 2 many details still have disappeared. To further improve on this, other types of pyramids are needed, see Section 5.



$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(2)}(f)$ $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(1)}(f)$ $\mathbf{M}_{\mathbf{\Theta}}(f)$

Figure 6: MMIP reconstruction from a 2-level morphological adjunction pyramid using a flat structuring function with a support of size $2 \times 2 \times 2$. From left to right: approximations on level 2, level 1 and level 0 (original).



$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(2)}(f)$ $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(1)}(f)$ $\mathbf{M}_{\mathbf{\Theta}}(f)$

Figure 7: MMIP reconstruction from a 2-level morphological adjunction pyramid using a non-flat structuring function with a support of size $2 \times 2 \times 2$. From left to right: approximations on level 2, level 1 and level 0 (original).

5 Discussion

In this paper, we have described a multiresolution extension to maximum intensity projection (MIP) volume rendering, allowing progressive refinement and perfect reconstruction. The method is based on a particular type of morphological pyramids, the *adjunction* pyramids. The pyramidal analysis and synthesis operators are composed of morphological 3-D erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis.

The proposed multiresolution MIP (MMIP) algorithm can be summarized as follows. In a preprocessing step, an L-level 3-D morphological adjunction pyramid of the volume data is computed, 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. After choosing a view direction, actual MIP volume rendering takes place by first computing a low resolution approximation from the coarsest signal f_L, and then refining the image progressively by taking the detail signals d_k, $k = L-1, \ldots, 0$ into account. We have shown that the operations of maximum projection and pyramidal 3-D synthesis can be interchanged (for non-axial directions, this step implies a certain discretization error, which was found to be small). As a consequence, maximum projection of the coarser signals can be computed first, followed by a 2-D pyramid synthesis operation and a 2-D closing, resulting in a computationally efficient algorithm. The algorithm is very similar to wavelet splatting [11, 24], the main differences being that (i) linear summation of voxel values is replaced by maximum computation, and (ii) linear wavelet filters are replaced by morphological filters (dilation and erosion). If the volume data are of integer type (bytes or shorts, depending on the dynamic range), then in contrast to the case of wavelet-based volume rendering, no floating point computations are required, but all operations are carried out as minimum and maximum calculations on integers. An efficient implementation was obtained by using an object order voxel projection scheme [15] acting on an ordered list of voxel values containing only the nonzero voxels. Further memory reduction is possible by compression domain rendering, just as in the case of wavelet splatting [12].

There are a number of issues to be addressed in future work. First, adjunction pyramids quickly remove details of the data at higher approximation levels. To improve on this, other operators instead of erosions can be used for the analysis phase, such as openings. This implies however, that the pyramid is no longer an adjunction pyramid, and in particular, that the representation formula (11) 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. Examples of such morphological pyramids are known, for example the so-called Sun-Maragos pyramid [23], see also [6]. Second, the special choice of subtraction operator in adjunction pyramids does have the disadvantage that the detail signals themselves are not 'small' in regions where the structuring function does not match the data well. So, for better compression other choices of addition and subtraction operators have to be investigated. Finally, we want to study the use of morphological wavelets [9] instead of morphological pyramids, which have the advantage that they provide a nonredundant multiresolution representation, thus further saving memory.

References

[1] Burt, P. J., and Adelson, E. H. The Laplacian pyramid as a compact image code. *IEEE Trans. Communications 31* (1983), 532–540.

[2] Cai, W., and Sakas, G. Maximum intensity projection using splatting in sheared object space. *Computer Graphics Forum (Proc. Eurographics'98) 17*, 3 (1998), C113–124.

[3] Camps, O. I., Kanungo, T., and Haralick, R. M. Gray-scale structuring element decomposition. *IEEE Transactions on Image Processing 5*, 1 (1996), 111–120.

[4] Chui, C. K. *An Introduction to Wavelets*. Academic Press, New York, 1992.

[5] Drebin, R. A., Carpenter, L., and Hanrahan, P. Volume rendering. *Computer Graphics (SIGGRAPH '88 proceedings) 22*, 4 (1988), 65–74.

[6] Goutsias, J., and Heijmans, H. J. A. M. Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids. *IEEE Trans. Image Processing 9*, 11 (2000), 1862–1876.

[7] 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.

[8] Heijmans, H. J. A. M. *Morphological Image Operators*, vol. 25 of *Advances in Electronics and Electron Physics, Supplement*. Academic Press, New York, 1994.

[9] Heijmans, H. J. A. M., and Goutsias, J. Multiresolution signal decomposition schemes. Part 2: morphological wavelets. *IEEE Trans. Image Processing 9*, 11 (2000), 1897–1913.

[10] Heijmans, H. J. A. M., and Toet, A. Morphological sampling. *Comp. Vis. Graph. Im. Proc.: Image Understanding 54* (1991), 384–400.

[11] 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.

[12] Lippert, L., Gross, M. H., and Kurmann, C. Compression domain volume rendering for distributed environments. In *Proc. Eurographics'97* (1997), pp. 95–107.

[13] 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.

[14] Malzbender, T. Fourier volume rendering. *ACM Transactions on Graphics 12*, 3 (1993), 233–250.

[15] Mroz, L., König, A., and Gröller, E. Maximum intensity projection at warp speed. *Computers & Graphics 24* (2000), 343–352.

[16] Muraki, S. Volume data and wavelet transforms. *IEEE Computer Graphics and Applications 13*, 4 (1993), 50–56.

[17] Roerdink, J. B. T. M. Multiresolution maximum intensity volume rendering by morphological pyramids. In *Data Visualization 2001. Proc. Joint Eurographics – IEEE TCVG Symposium on Visualization, May 28-30, 2001, Ascona, Switzerland*, D. Ebert, J. M. Favre, and R. Peikert, Eds. Springer, Wien, New York, 2001, pp. 45–54.

[18] 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.

[19] Roerdink, J. B. T. M., and Meijster, A. The watershed transform: definitions, algorithms, and parallellization strategies. *Fundamenta Informaticae 41* (2000), 187–228.

[20] 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.

[21] Serra, J. *Image Analysis and Mathematical Morphology*. Academic Press, New York, 1982.

[22] Sternberg, S. R. Grayscale morphology. *Comp. Vis. Graph. Im. Proc. 35* (1986), 333–355.

[23] Sun, F. K., and Maragos, P. Experiments on image compression using morphological piramids. In *SPIE Conf. Vis. Comm. Im. Proc. IV* (1989), vol. 1199, pp. 1303–1312.

[24] 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.

[25] Westenberg, M. A., and Roerdink, J. B. T. M. X-ray volume rendering through two-stage splatting. *Machine Graphics & Vision 9*, 1/2 (2000), 307–314.

[26] Westermann, R., and Ertl, T. A multiscale approach to integrated volume segmentation and rendering. In *Proc. Eurographics'97, Vienna* (1997), D. Fellner and L. Szirmay-Kalos, Eds., vol. 16 (3), pp. C–117–C–127.

[27] Westover, L. A. Footprint evaluation for volume rendering. *Computer Graphics 24*, 4 (1990), 367–376.

[28] Zhuang, X., and Haralick, R. M. Morphological structuring element decomposition. *Comp. Vis. Graph. Im. Proc. 35* (1986), 370–382.

[29] 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.

A Maximum Intensity Projection for arbitrary view angles

We choose one of the possible morphological sampling strategies described in [8, 10], by reconstructing a continuous function from discrete data by dilation, and using an erosion to sample a continuous function. In this appendix, tildes are used to indicate continuous functions. Starting point is a discrete data array $f(k,l,m)$, where the index vector (k,l,m) runs over a sampling grid $S \subseteq \mathbb{Z}^3$. First, a continuous function $\tilde{f}$ is reconstructed from the discrete data f by dilation with a structuring element $K \subseteq \mathbb{R}^3$:

$$\tilde{f}(x) = \delta_K(f)(x) = \bigvee_{s \in K_x \cap S} f(s), \quad x \in \mathbb{R}^3.$$

Then the reconstructed function $\tilde{f}$ is projected along an arbitrary view vector $\boldsymbol{\theta}$ on the view plane perpendicular to $\boldsymbol{\theta}$. Finally, we sample the projected function, say $\tilde{g}(p)$, $p \in \mathbb{R}^2$, on a Cartesian pixel grid in the view plane by erosion with K', where K' is the projection of K on the view plane:

$$g(t) = \varepsilon_{K'}(\tilde{g})(t) = \bigwedge_{p \in K'} \tilde{g}(t+p), \quad t \in \mathbb{Z}^2.$$

Recall from Section 3.2 that the projection geometry involves the direction vector $\boldsymbol{\theta}$ along which projection takes place, and two mutually orthogonal vectors $\mathbf{u}$ and $\mathbf{v}$ perpendicular to $\boldsymbol{\theta}$. Coordinates

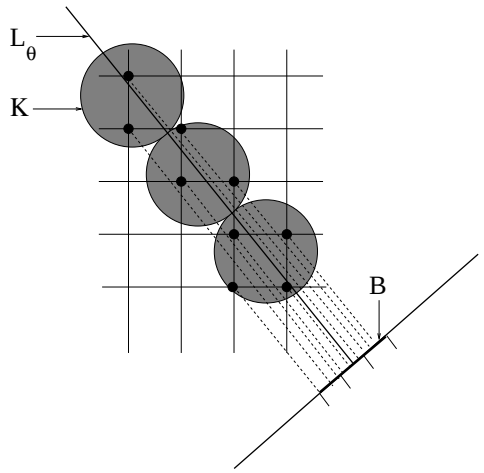


Figure 8: Projection of a dilated 2-D function on a line.

of the projection of a point $x \in \mathbb{R}^3$ are denoted by $(u, v) \in \mathbb{R}^2$. Let $L_{\boldsymbol{\theta}}(i, j)$ be the line parallel to $\boldsymbol{\theta}$ which passes through the pixel with center $(i, j) \in P \subseteq \mathbb{Z}^2$, where P denotes the pixel grid in the view plane. We construct a projected function which is piecewise constant on pixels. The value at any point $(u, v) \in C(i, j)$ is taken equal to the continuous MIP of $\tilde{f}$, say $m_{\boldsymbol{\Theta}}(\tilde{f})$, evaluated at the center (i, j) of pixel $C(i, j)$:

$$\begin{aligned} m_{\boldsymbol{\Theta}}(\tilde{f})(u, v) &= \bigvee_{x \in L_{\boldsymbol{\theta}}(i,j)} \bigvee_{(k,l,m) \in K_x} f(k, l, m) \\ &= \bigvee_{(k,l,m) \in L_{\boldsymbol{\theta}}(i,j) \oplus K} f(k, l, m), \end{aligned}$$

for $(u, v) \in C(i, j)$. Let $B \subseteq \mathbb{Z}^2$ be the set of pixels which are hit by the projection of K on the view plane. If the structuring element K is rotationally symmetric, the shape of B will be independent of the view direction. Taking into account the assumption of piecewise constancy, the previous equation can be rewritten as

$$\begin{aligned} m_{\boldsymbol{\Theta}}(\tilde{f})(u, v) = \bigvee_{(i',j') \in B_{(i,j)}} \bigvee_{(k,l,m)} \{f(k, l, m) : \\ \pi_{\boldsymbol{\Theta}}(k, l, m) \in C(i', j')\}, \end{aligned} \tag{30}$$

where $B_{(i,j)}$ is the translate of B over the discrete vector (i, j), and $\pi_{\boldsymbol{\Theta}}(k, l, m)$ is the projection of (k, l, m) on the view plane. The process is illustrated for the 2-D case in Figure 8.

We can rewrite (30) as

$$m_{\boldsymbol{\Theta}}(\tilde{f})(u, v) = \bigvee_{(i',j') \in B_{(i,j)}} \mathcal{M}_{\boldsymbol{\Theta}}(f)(i', j') = \delta_B(\mathcal{M}_{\boldsymbol{\Theta}}(f))(i, j),$$

for $(u, v) \in C(i, j)$, where $\mathcal{M}_{\boldsymbol{\Theta}}(f)$ is defined in (20). The final result is obtained by sampling $m_{\boldsymbol{\Theta}}(\tilde{f})(u, v)$ by erosion. The combined operator which maps the discrete 3-D array f to a discrete 2-D array is denoted by $\mathbf{M}_{\boldsymbol{\Theta}}$. Then one gets

$$\mathbf{M}_{\boldsymbol{\Theta}}(f)(i, j) = \epsilon_B(m_{\boldsymbol{\Theta}}(\tilde{f}))(i, j) = \phi_B(\mathcal{M}_{\boldsymbol{\Theta}}(f))(i, j),$$

where $\phi_B = \epsilon_B \delta_B$ is the closing with structuring element B.

To minimize the amount of flattening of the projected image due to the closing, the structuring element K should be as small as possible. On the other hand, K should be large enough to cover the complete volume by translation over integer steps. One obvious choice is to take K equal to a unit cube surrounding each voxel, so that the translates of K over the voxel lattice fill the volume without overlap. In that case the projected structuring element B becomes dependent upon the view direction. A simpler choice is to take for B a fixed structuring element of size 2×2, except for axial projections where B is a single pixel (i.e. the closing has no effect, so can be omitted).