University of Groningen

Wavelet-based X-ray volume rendering

Westenberg, Michel Arjan

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2001

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Westenberg, M. A. (2001). *Wavelet-based X-ray volume rendering*. s.n.

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 07-06-2022

n. This was done
rder interpolation
arginally different
ction, we consider
show that bilinear
ge as 10 degrees

f the angle incre-
f the angle incre-
roximation coeffi-
ering always takes
angle increments

Chapter 6

Concluding remarks

6.1 Summary

Over the last decades, exploration and visualization of large three-dimensional (3-D) data volumes has become important in many scientific areas. Such volumes are obtained by physical measurements (magnetic resonance imaging, computerized tomography), numerical simulations (computational fluid dynamics), or modelling (computer aided design). The process of generating two-dimensional (2-D) images from volume data is called *volume rendering*. Chapter 1 provides an overview of different volume rendering methods.

Recently, researchers have developed an interest in exchanging volume data and performing collaborative visualization through the Internet. In the context of volume rendering, one may think of client/server systems in which volume data are stored on a central server and visualized on one or more remote clients. Bandwidth is a limiting factor, therefore, these systems require fast and efficient methods of transfer and display. This requires a mechanism to visualize data incrementally as they arrive through the network, a technique called *progressive refinement*. For this purpose, multiresolution models have been developed, allowing a systematic decomposition of the volume data into versions at different levels of resolution. *Wavelets* are a natural candidate for such a multiresolution approach. In Chapter 1, we summarize the basic facts about wavelets.

This thesis describes a number of wavelet-based volume rendering methods which are extensions of standard X-ray rendering. This method integrates the data values along the line of sight, resulting in images that look like X-ray photographs. We describe extensions to standard algorithms viz. *splatting* and *Fourier volume rendering*. Splatting is a method that reconstructs the underlying continuous function from the discrete volume data by convolution with a reconstruction filter, followed by a mapping to the view plane by superposition of building blocks called footprints. These are the result of integrating the reconstruction kernel along the line of sight. For orthographic projection, the footprints are the same for all voxels, and have to be computed only once for a given viewing direction, making splatting an efficient rendering method. Fourier volume rendering makes use of frequency domain techniques, and provides another efficient way to perform X-ray rendering. Fourier volume rendering starts by computing the 3-D Fourier transform of the volume data. For a given viewing direction, an image is made by computing the values of the Fourier transform in a slice plane parallel to the view plane and through the center of Fourier space, followed by an inverse 2-D Fourier transform of the slice plane.

Chapter 2 introduces *two-stage splatting*, a method that speeds up the rendering process in case of X-ray rendering by splatting. This is done by splitting the splatting process in two phases: (i) coefficient accumulation and (ii) a final convolution with the footprint. This saves many multiplications, and, therefore, reduces rendering time. Experimental results show that speedups up to a factor of three can be reached. In addition, the two-stage splatting method is incorporated into wavelet splatting. Two-stage splatting proves to be less dependent on the size of the footprint. This is important for wavelet splatting, because it is then possible to use wavelets with larger support, which produce visually more pleasing low resolution images.

Hierarchical wavelet splatting (HWS) is described in Chapter 3. This method is a combination of the wavelet splatting method described in Chapter 2, and the one described by Lippert *et al.*, taking advantage of the strengths of both methods. The first method uses the hierarchical ordering provided by the wavelet transform during rendering. The second method (WS) imposes a global ordering on the coefficients, which turns out to introduce artefacts in low resolution images. The hierarchical ordering does not suffer from these artefacts. The advantage of the global ordering is that images of high resolution are obtained using only a small number of wavelet coefficients. Hierarchical wavelet splatting separates the wavelet coefficients into two sequences: one containing the approximation coefficients and another containing all detail coefficients on which a global ordering is imposed. Experiments show that image quality for low resolution images computed during user interaction is higher in comparison to WS. Rendering during user interaction is faster with HWS, since it uses the two-stage splatting algorithm to increase rendering speed. Similar speeds can be reached by WS, albeit at a lower image quality. In the refinement phase, i.e. when user interaction ceases, HWS and WS show similar behaviour, both needing approximately the same number of wavelet coefficients to keep errors below a certain value. We define errors in terms of the L_∞ norm of the difference between an image obtained using all wavelet coefficients and an image obtained using a limited number of coefficients. Experiments with a CT volume data set show that a large number of wavelet coefficients may be discarded, resulting in only small image degradation.

Chapter 4 introduces Fourier-wavelet volume rendering (FWVR), a wavelet-based extension to Fourier volume rendering. The method combines integration along the line of sight with a simultaneous 2-D wavelet transform in the view plane perpendicular to this line. This is called the *wavelet X-ray transform*, and we derive an efficient algorithm to compute this transform by using a frequency domain implementation of the wavelet transform. Approximation images are obtained by a partial wavelet reconstruction in frequency space, followed by a 2-D inverse Fourier transform to obtain an image in the spatial domain. Progressive refinement is possible, and we show that it can be implemented in a straightforward manner in the frequency domain. Despite the general belief in the literature that Fourier rendering is an inaccurate method, it turns out that this is not the case when proper interpolation functions are chosen. The multiresolution variant offered by Fourier-wavelet volume rendering proves to be useful to implement client/server visualization systems.

In Chapter 5, we extend Fourier-wavelet volume rendering with a technique similar to *view interpolation*. View interpolation is used in computer graphics to speed up rendering of complex scenes, and is done by precomputing images for a number of viewing directions. Images for intermediate viewing directions are then obtained by interpolating the precomputed images. The extension proposed in this chapter uses a set of precomputed sequences of wavelet approximation coefficients in the Fourier domain for different viewing directions. The new algorithm computes images for intermediate viewing directions by interpolation of the precomputed coefficients. The main differences between ordinary view interpolation and view interpolation in Fourier-wavelet space are that (i) interpolation is performed on the wavelet approximation coefficients in the frequency domain and not in the image domain, and (ii) interpolation is performed during user interaction only. We use simple bilinear interpolation for view interpolation. This was done for two reasons: (i) it is computationally more efficient than higher order interpolation methods, and (ii) higher order interpolation methods give only marginally different results. Since view interpolation is applied *only* during user interaction, we consider speed to be more important than accuracy. Furthermore, the results show that bilinear interpolation gives acceptable errors, and angle increments as large as 10 degrees result in only a small degradation of image quality.

6.2 Perspectives

This thesis has presented some fundamental wavelet-based volume rendering methods which are extensions of standard X-ray rendering. A few simple extensions to the methods proposed were already mentioned in the previous chapters. Into both the wavelet splatting and Fourier-wavelet method, it is possible to incorporate depth-cueing and gradient-based shading. Standard wavelet compression methods could be used to reduce the number of coefficients for wavelet splatting. For the client/server based Fourier-wavelet rendering, compression is a little more challenging, since it cannot be done as preprocessing. When the viewing direction changes, the Fourier-wavelet decomposition of the slice plane is recomputed, and, as a consequence, the compression algorithm would be invoked to compress the resulting wavelet coefficients. This requires very fast compression methods, otherwise, the transmission speed gain obtained by sending only the wavelet approximation coefficients would be nullified. A last simple addition would be to include colour, which can be done easily by applying the methods on the separate colour channels.

An interesting area of research is the combination of wavelet theory and mathematical morphology. These so-called *morphological wavelets* (Heijmans and Goutsias 2000) offer interesting possibilities. Mathematical morphology considers an image as a geometric object, and analyzes its structure with nonlinear operators which make use of *structuring elements*. The structuring elements are building blocks for operators that, for instance, can find objects with particular shape or size attributes. Morphological wavelets offer a way to make nonlinear extensions to the standard linear wavelets, and they possess some useful properties (Heijmans and Goutsias 2000). (i) For integer-valued input, the output is integer-valued as well. This is useful for large 3-D volumes, so that it is not necessary to adopt a floating point representation. (ii) The operations are simple (addition, subtraction, min, max), enabling fast implementations. (iii) The nonlinear operators preserve geometric information well at low resolutions. One of the schemes proposed by the authors is called max-lifting, which preserves local maxima at lower resolutions. The question now is: Can we use morphological wavelets to construct new volume rendering methods? One possibility would be to develop a multiresolution variant of maximum intensity projection (MIP). A pilot study demonstrated that it is possible to construct a multiresolution MIP based on morphological pyramids (Roerdink 2001). Whether this result can be extended to morphological wavelets is an open issue.

A further area of research is the application of wavelets in volume rendering methods that support transparency. Lippert *et al.* (1995) have developed a multiresolution raycasting algorithm which renders directly from wavelet space. The problem, however, is that ray integration becomes very complicated. The authors do not mention rendering times, and do not make a comparison with standard raycasting, so it is not clear what is gained by rendering from wavelet space. The method does not allow progressive refinement, therefore, the rendering process has to be repeated to obtain images of higher resolution. It would be interesting to investigate if this problem can be resolved by an intermediate datastructure that allows both fast reconstruction from wavelet space *and* fast accumulation to obtain an image in the view plane.