

# Projected slabs: approximation of perspective projection and error analysis

By A. Vilanova Bartrolí\*, R. Wegenkittl and E. Gröller



*Virtual endoscopy is a promising medical application for volume-rendering techniques where perspective projection is mandatory. Most of the acceleration techniques for direct volume rendering use parallel projection. This paper presents an algorithm to approximate perspective volume rendering using parallel projected slabs. The introduced error due to the approximation is investigated. An analytical study of the maximum and average error is made. This method is applied to VolumePro 500. Based on the error analysis, the basic algorithm is improved. This improvement increases the frame rate, keeping the global maximum error bounded. The usability of the algorithm is shown through the virtual endoscopic investigation of various types of medical data sets. Copyright © 2002 John Wiley & Sons, Ltd.*

Received: 30 November 2001; Revised: 12 December 2001

KEY WORDS: direct volume rendering; virtual endoscopy; perspective projection; volumePro technology

## Introduction

The visualization of medical volume data produced by 3D imaging techniques (e.g., CT and MRI) has been intensively investigated in the last decades. Its application to daily medical care can highly improve the quality of current medical procedures.

Virtual endoscopy is an application which deals with the exploration of hollow organs and anatomical cavities using volume data. Virtual endoscopy has the potential of being used as a non-invasive diagnostic technique.

Several virtual endoscopy systems have been proposed.<sup>1–3</sup> These systems are basically concerned with two visualization techniques: surface rendering and direct volume rendering. Surface rendering leads to a reduction of the data information from 3D (volume) to 2D (surfaces). This incurs a loss of information and accuracy while requiring surface extraction as a preprocessing step. On the other hand, the advantage is that common graphics hardware can be used to accelerate the rendering step.

Direct volume rendering uses the 3D information and projects it onto a 2D image plane (e.g., with

ray-casting<sup>4</sup>). It has the disadvantage that projection algorithms are computationally expensive. This paper concentrates on direct volume-rendering techniques which require no preprocessing and achieve a higher accuracy.

There are several software<sup>5–7</sup> and hardware<sup>8,9</sup> acceleration techniques to improve the frame rate of direct volume rendering. The software-accelerated techniques usually need additional data storage and preprocessing.

3D texture mapping<sup>8</sup> is the most often used hardware acceleration technique. This method achieves interactive frame rates on an SGI Reality Engine, but it is difficult to incorporate this technique into a desktop machine like a PC. The basic method does not support the possibility to estimate gradients which is required to employ lighting models like the Phong model with diffuse and specular lighting effects. However, several approaches to overcoming this problem have been proposed.<sup>10</sup>

The VolumePro board<sup>9</sup> is a hardware implementation of ray-casting using shear-warp factorization.<sup>6</sup> It provides real-time rendering with compositing, classification with densitybased transfer functions and Phong shading. One of the main drawbacks of this board, with regard to usage in virtual endoscopy, is that it does not produce perspective projection. For

\*Correspondence to: A. Vilanova Bartrolí, Eindhoven University of Technology, Biomedical Engineering, Biomedical Imaging, PO Box 513, WH 2.103, 5600 MB Eindhoven, The Netherlands.  
E-mail: anna@cg.tuwien.ac.at

Copyright © 2002 John Wiley & Sons, Ltd.

Based on 'Perspective Projection through Parallel Projected Slabs for Virtual Endoscopy' by A. Vilanova, R. Wegenkittl, A. König and E. Gröller which appeared in *Proceedings of the Seventeenth Spring Conference on Computer Graphics*, April 2001, Budmerice, Slovakia, edited by Roman Durikovic and Silvester Czanner. © 2001 IEEE.

outside views, parallel projection gives good results but for inside views (e.g., endoscopic views), perspective projection is mandatory to provide a correct depth impression.

In this paper, a method to approximate perspective projection from several parallel projected slabs, similar to slicing<sup>8</sup> and to the slab subdivision,<sup>11</sup> is presented. An error estimation of the approximation is studied. An analysis of the error bounds and average error is given. Furthermore an improvement in the performance of the initial algorithm by using the error estimation is described. Finally, a study using several clinical data sets and performance issues is discussed.

The presented algorithm is not only restricted to the VolumePro application. More generally, the concept can be applied wherever perspective projection is used.

### Projected-Slabs Algorithm

The VolumePro 500 system is able to produce high-quality volume renderings of about 30 frames per second for a 256 cubic-size volume data.

The VolumePro technology implements a shear-warp algorithm<sup>6</sup> in hardware. It renders the base-plane image and the 2D warp operation is done using common graphics hardware.

The basic idea of the presented algorithm, called the projected-slabs algorithm, is as follows: generate a perspective rendering of the entire 3D data set, approximated by consecutive parallel projections of slabs of the volume data (see Figure 1). A slab is part of the volume data in two cutting planes which are orthogonal to the viewing direction. The thickness of a slab is selected such that the difference between a parallel projection and a perspective projection of the volume data contained within a slab is tolerable (i.e. below a certain error threshold). Using the cutting plane feature of the VolumePro system, each slab is rendered using parallel projection. The resulting base-plane image of an individual slab is then warped and transformed according to the perspective parameters of the defined camera.

All the images, one per slab, are finally blended to obtain the image of the entire data set. Figure 2 illustrates which data values are accumulated along a viewing ray with the projected-slabs methods compared to the correct perspective solution.

This algorithm uses an entire VolumePro 500 rendering cycle for every rendered slab and therefore the

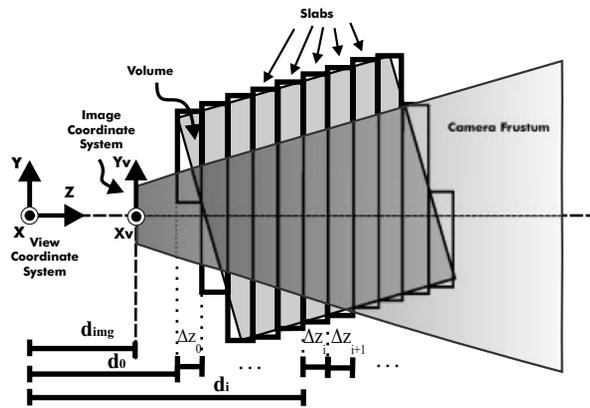


Figure 1. Illustration of the perspective approximation using the projected-slabs algorithm.

rendering frame rate is decreasing in the order of the number of slabs that are needed for the perspective approximation. Furthermore, there is an overhead due to the blending of the slabs.

Given the view position and the viewing direction, the slabs are numbered using the distance to the viewpoint in the following way (see Figure 1):

$$d_j = d_0 + \sum_{i=0}^{j-1} \Delta z_i \tag{1}$$

where  $j \geq 1$  and  $d_0$  is the distance from the viewpoint to the front plane of the first slab, and  $\Delta z_i$  is the thickness of the slab which starts at distance  $d_i$ .

If  $\Delta z_i$  is a constant value smaller than a voxel size for all the slabs, it is intuitive to see that the result produces good-quality perspective rendering. On the other hand, it also produces an intolerably high number of slabs and consequently decreases the frame rate. So the thickness of the slabs must be set to a value larger than one voxel size to obtain a reasonable performance.

Since we are approximating perspective projection, it is important to be able to evaluate the error produced due to this approximation.

### Error Estimation of the Projected-Slabs Algorithm

In this section, we study the error that results from the use of parallel projected slabs to produce the perspective view.

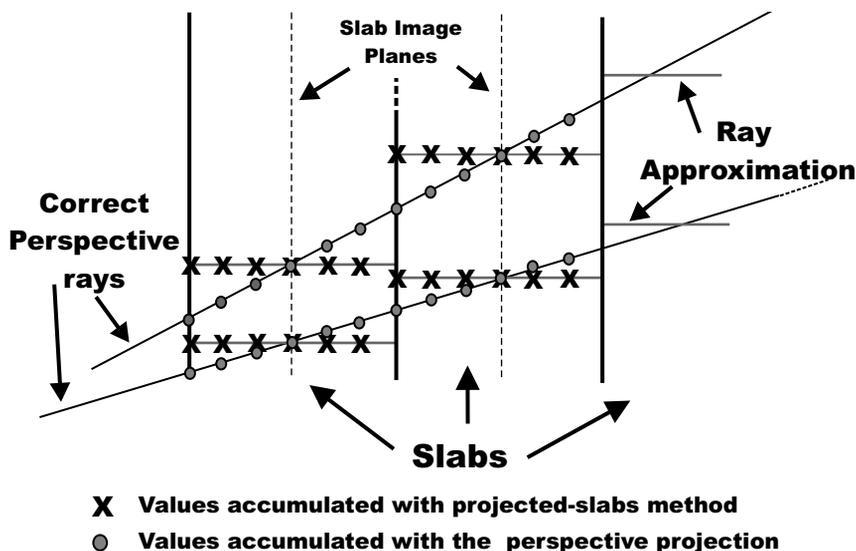


Figure 2. Accumulated values in a correct perspective projection as compared to the projected-slabs algorithm.

The basic error is that the sample points are projected to the wrong position in the image plane, and therefore they are accumulated to the wrong ray. Based on that, the error estimation is defined as the distance in the image plane between the correct perspective projected point and the point produced by the projected-slabs algorithm.

In the rendering pipeline, the difference between parallel and perspective projection appears after the world coordinates have been transformed to view coordinates. To transform from view coordinates to image plane coordinates, the appropriate projection matrix is used. For simplicity and a more intuitive explanation, we assume a left-handed camera system (see Figure 1) where the viewpoint is in the origin of the view coordinates. The image plane is orthogonal to the Z-direction and located at a distance  $d_{img}$  from the viewpoint.

We define a point  $P_v = (X_v, Y_v, Z_v)$  as a point resulting from applying a view-coordinate transformation to an arbitrary point in world coordinates.

The perspective projection matrix for a left-handed camera system, supposing left accumulation matrix notation, is:

$$M_{persp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{d} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where  $d$  is the distance from the viewpoint to an

arbitrary projection plane. If  $d = d_{img}$  then the projection plane is the image plane. The parallel transformation to image coordinates is simply the transformation of the Z-coordinate to the projection plane position  $d$ . It can be expressed by the matrix:

$$M_{paral} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & d & 1 \end{bmatrix}$$

If these transformations are applied to  $P_v$  we obtain the following equalities ( $P^h$  are points expressed in homogeneous coordinates).

$$P_{persp}^h = P_v^h * M_{persp} = \left[ X_v, Y_v, Z_v, \frac{Z_v}{d} \right]$$

$$P_{paral}^h = P_v^h * M_{paral} = [X_v, Y_v, d, 1]$$

The points  $P_{persp}^h$  and  $P_{paral}^h$  are transformed to the 2D projection plane coordinate system. This coordinate system is defined with the same X and Y-directions as the view coordinate system. The projection plane origin corresponds to  $[0,0,d]$  in view coordinates. Then,  $P_{persp}^h$  corresponds to  $P_{persp}$  and  $P_{paral}^h$  corresponds to  $P_{paral}$  with:

$$P_{persp} = \frac{d}{Z_v} [X_v, Y_v]$$

$$P_{paral} = [X_v, Y_v]$$

Error  $e_p$  is defined as the distance between the



quadrilateral defined by the intersection of the frustum and the image plane.

Developing the inner integral, we obtain the following equation:

$$\bar{e}_i = \frac{\left(\frac{\Delta z_i}{2}\right)A}{2\left(d_i + \frac{\Delta z_i}{2}\right)X_c Y_c} \quad (7)$$

where

$$A = \int_0^{Y_c} \int_0^{X_c} \|[X_p, Y_p]\| dX_p dY_p \quad (8)$$

Solving the integrals in equation (8) we obtain:

$$A = \frac{1}{3} X_c Y_c \sqrt{X_c^2 + Y_c^2} - \frac{1}{6} X_c^3 \ln(X_c) - \frac{1}{6} (Y_c)^3 \ln(Y_c) + \frac{1}{6} Y_c^3 \ln\left(X_c + \sqrt{X_c^2 + Y_c^2}\right) + \frac{1}{6} X_c^3 \ln\left(Y_c + \sqrt{X_c^2 + Y_c^2}\right)$$

In order to simplify this equation we assume that our images are squares, which is usually the case. This means that  $X_c = Y_c = D$ , and therefore equation (9) can be simplified to:

$$A = \frac{\sqrt{2}}{3} D^3 - \frac{2}{6} D^3 \ln(D) + \frac{2}{6} D^3 \ln\left((1 + \sqrt{2})D\right) = \left(\frac{\sqrt{2} + \ln(1 + \sqrt{2})}{3}\right) D^3 \quad (9)$$

If this result is combined with equation (7), then the average error within a slab is expressed by:

$$\bar{e}_i = \frac{\left(\frac{\Delta z_i}{2}\right)\left(\frac{\sqrt{2} + \ln(1 + \sqrt{2})}{3}\right) D^3}{2\left(d_i + \frac{\Delta z_i}{2}\right) D^2} = \left(\frac{\frac{\Delta z_i}{2}}{d_i + \frac{\Delta z_i}{2}}\right) \frac{\sqrt{2} + \ln(1 + \sqrt{2})}{6} D \quad (10)$$

This equation gives the average error for the points within a slab. The global average error,  $\bar{e}$ , for all the points that are projected to the image plane is:

$$\bar{e} = \frac{\sum_{i=0}^{n-1} \Delta z_i X_c Y_c \bar{e}_i}{\sum_{i=0}^{n-1} \Delta z_i X_c Y_c} \quad (11)$$

where  $n$  is the number of slabs.

## Maximum Error

In this section, we bound the error produced by the projected-slab algorithm. The value of  $e_{img_i}$  represents

the error or distance in the image plane between the projected-slabs algorithm and the correct perspective projection of a point  $P_v$  situated in the slab  $i$ . The point is situated at a distance  $\Delta z_v$  from the slab image plane and its perspective projection to the image plane gives  $[X_p, Y_p]$ . From equation (6), we can deduce several properties:

1. The error  $e_{img_i}$  is proportional to  $\|[X_p, Y_p]\|$ , the distance of  $P_p$  to the origin of the image plane.
2.  $\forall [X_p, Y_p]: |\Delta z_v| \rightarrow 0 \Rightarrow e_{img_i} \rightarrow 0$ . This behavior has already been intuitively described in the section on 'Projected-Slabs Algorithm'. Furthermore, it means that for points on the slab image plane the error is 0.
3. The error value increases, if  $|\Delta z_v|$  increases.
4. For fixed  $\|[X_p, Y_p]\|$ , if  $d_i$  increases, then the variation in  $e_{img_i}$  due to the changes in  $|\Delta z_v|$  decreases.

The minimum error or lower bound is 0, this is when  $\|[X_p, Y_p]\| = 0$  (i.e., the distance of  $P_p$  to the image plane origin is 0), and when the points are in the slab image plane (see property 2). Further, we will concentrate on finding the maximum error or upper bound for all values of  $e_{img_i}$ .

Due to property 3, it is clear that  $e_{img_i}$  is maximal when  $|\Delta z_v| = \frac{\Delta z_i}{2}$ , i.e., when the point  $P_v$  is situated on the front or back plane of the slab.

Due to property 1, it is clear that the upper bound value of  $e_{img_i}$  within slab  $i$  occurs when the highest value of  $\|[X_p, Y_p]\|$  is reached. That implies that  $[X_p, Y_p]$  can be fixed to the farthest point from the origin of the image plane that contributes to the final image (i.e.,  $[X_c, Y_c]$ ). Therefore, we define  $e_i^{max}$  as the maximum error in the final image inferred by slab  $i$ :

$$e_i^{max} = \left(\frac{\frac{\Delta z_i}{2}}{d_i + \frac{\Delta z_i}{2}}\right) \|[X_c, Y_c]\| \quad (12)$$

The maximum error in the final image is the maximum value of  $e_i^{max}$  for any slab  $i$ .

$$e^{max} = \max\{e_i^{max} | i \geq 0\} \quad (13)$$

Due to property 4 we see that if  $\Delta z_i$  is a constant for all  $i$ , then every slab has a different  $e_i^{max}$  and its value decreases when  $d_i$  increases. So, the maximal error produced in the final blended image corresponds to the maximal error of the first slab,  $e^{max} = e_0^{max}$ .

In the section on 'Error-Induced Variation of Slab Thickness' we use the previous observations to optimize the projected-slabs algorithm, without increasing the maximal error value  $e^{max}$  produced in the final image.

## Average vs. Maximum Error

In this section, we compare the average and maximal error. For simplicity, we suppose that our image planes are squares (i.e.,  $X_c = Y_c = D$ ) then:

$$e_i^{max} = \left( \frac{\frac{\Delta z_i}{2}}{d_i + \frac{\Delta z_i}{2}} \right) \sqrt{2}D \quad (14)$$

Comparing the maximum error with the average error (see equation 10) for square images, we obtain the following:

$$\frac{\bar{e}_i}{e_i^{max}} = \frac{\sqrt{2} + \ln(1 + \sqrt{2})}{6\sqrt{2}} = 0.2705 \quad (15)$$

For each slab the average error is approximately a 27% of the maximum error (i.e.,  $\bar{e}_i = 0.2705e_i^{max}$ ).

## Error-Induced Variation of Slab Thickness

In the previous section, it has been observed that the maximum and average error for slab  $i$  due to the projected-slabs algorithm,  $e_i^{max}$  and  $e_i$ , depend on the distance of the slab to the viewpoint and on the slab thickness.

From property 4 in the 'Maximum Error' section and equation (13), it can be deduced that slabs further away from the viewpoint may have a greater thickness than slabs closer to the viewpoint, keeping the same maximum image error  $e_i^{max}$ .

In this section, we present the criterion for selecting the slab thickness depending on the distance to the viewpoint, the camera characteristics and the maximum error. The rule is to have as few slabs as possible while keeping the error tolerance unchanged. Using equation (12),  $\Delta z_i$  can be isolated in the following way:

$$\Delta z_i = 2 * \left( \frac{e_i^{max} * d_i}{\| [X_c, Y_c] \| - e_i^{max}} \right) \quad (16)$$

The  $e_i^{max}$  is set to a constant value *DistanceError*, for all the slabs. We defined the incremental slab thickness algorithm using equations (16) and (1). The thickness of the slabs is defined in an iterative way, assuring that the error will be kept smaller than the defined value *DistanceError*. As was mentioned in the 'Projected-Slabs Algorithm' section, the projected-slabs algorithm decreases the frame rate if the number of slabs increases. Calculating the thickness using equation

(16), the slab thickness will increase with the value of  $d_i$ . So, fewer slabs are needed and therefore the frame rate is higher.

Apart from the *DistanceError*, equation (16) also depends on the camera characteristics:  $[X_c, Y_c]$  is specified by the intersection of the frustum and the image plane.  $\Delta z_i$  just needs to be computed when the camera characteristics, the *DistanceError* or the first slab distance  $d_0$  are modified. In the case of a square projection image and using equation (15), we deduce that when fixing a maximum error *DistanceError*, we are also defining an average error which is 27% of the *DistanceError*. This average error is constant and the same for each slab, and therefore the global average error for the complete image (see equation 11) is also 27% of the maximum error.

The presented result, obtained using mathematical derivation, illustrates the fact that in perspective projection objects far away are projected smaller to the image plane. This intuitive idea has been already used in other adaptive approaches.<sup>5,11,12</sup>

*DistanceError* is a parameter of the algorithm. The estimation of the error tolerance (i.e. *DistanceError*) depends on the volume data to visualize. To illustrate this, we study one of the worst cases for the projected-slabs algorithm. This case occurs when the camera is in the center of a straight tube and the camera is pointing in the direction of the tube axis.

In parallel projection, a ring with the tube thickness would be projected to the image plane. If the projected-slabs algorithm is used, a ring with the tube thickness for any slab will be projected. If the tube has a small thickness (see Figure 4(a)), it looks like several concentric rings, giving the impression of having different objects. This is because the specified error tolerance is larger than the projected thickness of the tube. Therefore we see the tube as a discontinuous surface. If the *DistanceError* is decreased to a value that approximates the projection of the thickness in the image plane, a better result is obtained (see Figure 4(b)).

## Results

In this section, we present some timings and the evaluation realized for three types of data sets. The calculation times were obtained by executing the algorithm on a 400 MHz Pentium II. The performance of the algorithm presented in this article basically depends on the time necessary for the VolumePro 500

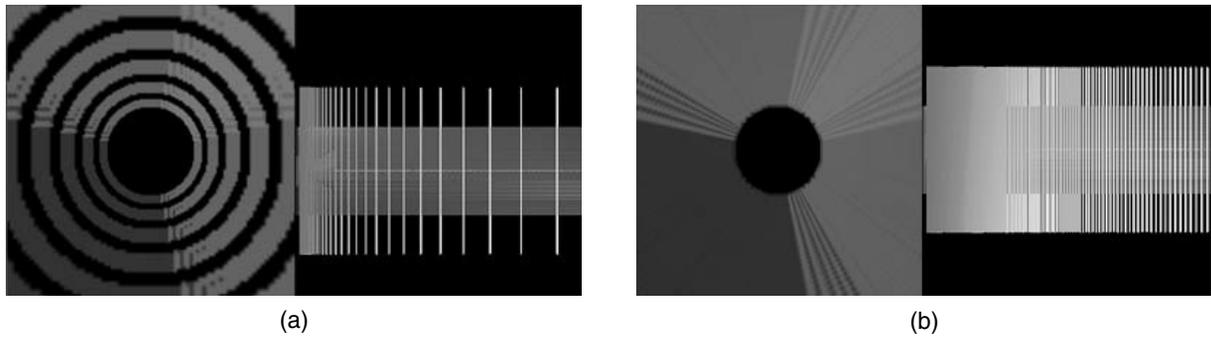


Figure 4. An illustration of the error tolerance behavior. Two endoscopic views using the projected-slabs algorithm of a voxelized tube are shown together with the corresponding outside view with the slab image planes. The error tolerance is different for each endoscopic view: (a)  $DistanceError = 5\%$  of the image size; (b)  $DistanceError = 2\%$  of the image size.

board to render a slab, and on the time for warping and blending. We provide timings relative to the times that can be achieved on a rendering cycle: time needed for the VolumePro 500 to render the volume once. A rendering cycle consists of the rendering of one slab using the VolumPro 500 board and the warping of the resulting base-plane image. The speed and quality of the warping and blending steps depend on the graphics hardware used.

In Tables I–III, the first column denotes the maximal error expressed as a percentage of the image size, the second column gives the average error, the third gives the number of slabs used, the fourth column represents the frame rate and the last column denotes the slow-down compared with the time of one rendering cycle.

The first dataset is a CT scan of a trachea of size  $292 \times 136 \times 114$ . The results of this visualization with

different maximal errors are shown in Figure 5. The CT was acquired from a corpse. After the scanning, a real bronchoscopy was performed. Figure 5 compares the results of the projectedslabs approach with the real endoscopic view from a similar camera position. It can be observed that the difference between incremental slab thickness (Figure 5a) and the constant slab thickness (Figure 5b) can be neglected. Numerical results are given in Table 1.

The second data set is a CT volume data of a trachea with a resolution of  $205 \times 83 \times 105$ . Figure 6 contains the images produced with different opacities for the walls of the trachea. The times are presented in Table 2. It can be observed that it is necessary to change the opacity of the trachea walls interactively, since it is difficult to recognize the structure of the trachea when it is semitransparent.

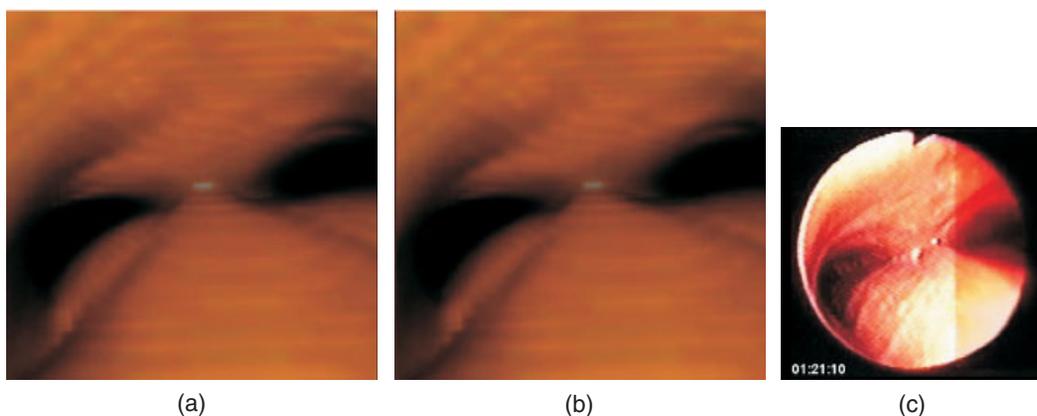


Figure 5. Visualization of the CT trachea data set of a corpse compared to a real endoscopic view: (a) projected-slabs algorithm with incremental slab thickness (37 slabs); (b) projected-slabs algorithm with constant slab thickness equal to one voxel distance (162 slabs); (c) real bronchoscopy snapshot.

$e^{max}$	$\bar{e}$	# slabs	f.p.s.	Slow-down factor
2.5%	0.675%	37	0.4	40.32
0%	0%	162	0.093	173.40

**Table 1. Times for the CT trachea of the corpse. One rendering cycle takes 65 ms. The error in the second row is 0% since a constant slab thickness of 1 voxel size per slab has been defined**

The third data set of size  $198 \times 115 \times 100$  is a portion of a Spiral CT of a colon. In Figure 7, a comparison between the projected-slabs technique and brute force volume rendering is presented. For the projected-slabs technique results are presented for both a constant and an incremental slab thickness. The times and frame rates are given in Table 3. We have observed that structures which are wide cavities produce good results since the firstly projected voxels fill a smaller image plane area and aliasing affects the performance of the algorithm less.

We have experienced that around 30 slabs are sufficient for the visualizations. As has been mentioned in the section on ‘Error-Induced Variation of Slab Thickness’, the projected-slabs algorithm is an approximation which depends on the characteristics of the structure to visualize. This incurs that depending on the structure and the parameters of the visualization, some artifacts may appear. For example, the transition between planes can be observed in some cases where the visualization parameters are not adequate.

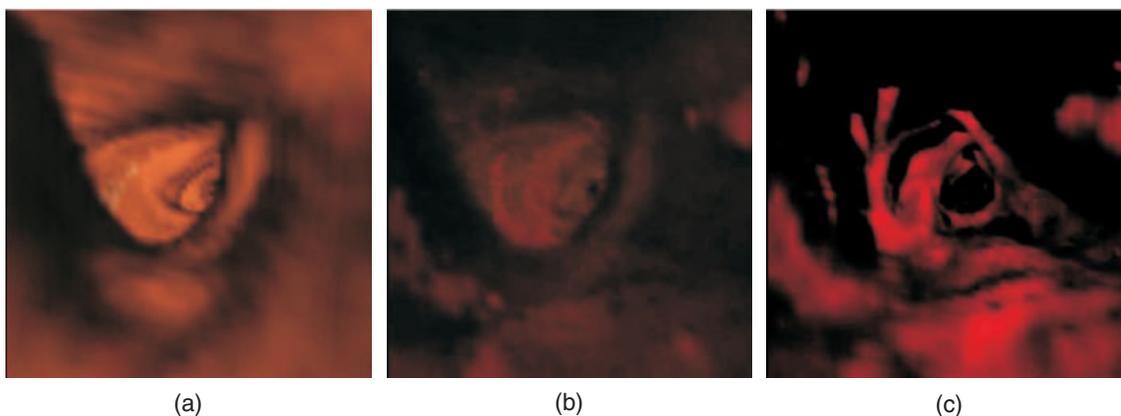
$e^{max}$	$\bar{e}$	# slabs	f.p.s.	Slow-down factor
4%	1.028%	21	0.94	22.6

**Table 2. CT of a trachea using transparency. One rendering cycle takes 47 ms**

## Conclusions and Future Work

An approach to produce perspective projection views using parallel volume rendering techniques (i.e., projected-slabs algorithm) has been presented. The algorithm uses consecutive parallel projected slabs of the volume. The error produced due to the approximation of perspective projection is investigated. The error bounds and average error are analyzed. Besides, based on the error studies, we introduce a criterion to vary the thickness of the slab and therefore to improve the algorithm performance. The usability of the algorithm has been tested using perspective views for virtual endoscopy in a common desktop machine using the VolumePro 500 system.

As future work, the algorithm can be speeded up by using the new version of VolumePro: VolumePro 1000. In the projected-slabs algorithm, VolumePro 500 processes the entire volume for every slab, although just a small portion of it contributes to the final image. VolumePro 1000, instead of rendering the entire volume, just processes the voxels that contribute to the final image. The projected-slabs algorithm may take advantage of this, and it could considerably improve its performance.



*Figure 6. CT trachea data set rendered with different transfer functions. From left to right, from opaque to transparent trachea walls, for the same camera position.*

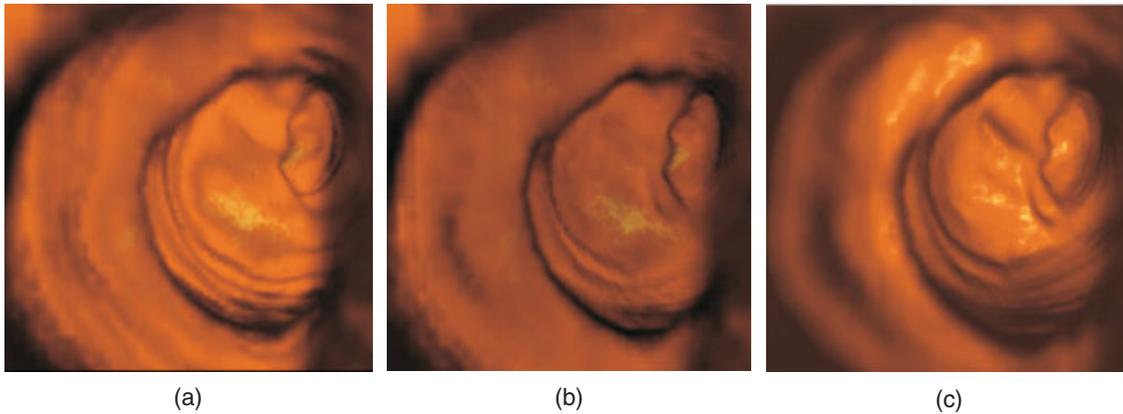


Figure 7. Spiral CT colon data set visualization: (a) projected-slabs algorithm with incremental slabs thickness (28 slabs); (b) projected-slabs algorithm with constant slab thickness and with the maximal error equal to (a) (179 slabs); (c) brute force ray-casting algorithm.

An investigation of how to automate the estimation of the parameters of the algorithm (e.g., the error tolerance) is another topic for future work.

A study of other uses of the presented error estimation algorithm should be performed. More generally, the algorithm subdivides view space into slabs, such that within the slabs approximated but faster operations are possible. In our case, parallel projection is used to approximate the perspective view. The concept could also be applied to other algorithms where perspective projection is used.

### ACKNOWLEDGEMENTS

The work presented in this publication has been funded by the VisMed project. VisMed is supported by Tiani Medgraph, Vienna (<http://www.tiani.com>), and the Forschungsförderungs fonds für die gewerbliche Wirtschaft, Austria. See <http://www.vismed.at> for further information on this project.

We thank Dr Martin C. Freund and the Department of Radiology at Leopold-Franzens University of Innsbruck for

their collaboration and for providing the data used in this paper.

### References

- Hong L, Muraki S, Kaufman A, Bartz D, He T. Virtual voyage: interactive navigation in the human colon. In *SIGGRAPH'97, Conference Proceedings, Annual Conference series, 1997*; pp 27–34.
- Satava RM, Robb RA. Virtual endoscopy: application of 3D visualization to medical diagnosis. *Presence 1997*; 6(2): 179–197.
- Vilanova Bartrolí A, König A, Gröller E. VirEn: a virtual endoscopy system. *Machine Graphics & Vision 1999*; 8(3): 469–487.
- Levoy M. Display of surfaces from volume data. *IEEE Computer Graphics and Applications 1988*; 8(5): 29–37.
- Brady ML, Jung KK, Nguyen HT, Nguyen T. Interactive volume navigation. *IEEE Transactions on Visualization and Computer Graphics 1998*; 4(3): 243–255.
- Lacroute P, Levoy M. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH'94, Conference Proceedings. ACM SIGGRAPH. ACM Press, July 1994*; pp 451–458.
- Wan M, Kaufman A, Bryson S. High performance presence-accelerated ray casting. In *IEEE Visualization'99, Conference Proceedings. IEEE, November 1999*; pp 379–386.
- Cullip TJ, Neumann U. Accelerating volume reconstruction with 3D texture hardware. Technical Report TR93-027, Department of Computer Science at the University of North Carolina, Chapel Hill, 1993.
- Pfister H, Hardenbergh J, Knittel J, Lauer H, Seiler L. The VolumePro real-time ray-casting system. In *SIGGRAPH'99, Conference Proceedings, 1999*; pp 251–260.
- Westermann R, Ertl T. Efficiently using graphics hardware in volume rendering applications. In *SIGGRAPH'98*,

$e^{max}$	$\bar{e}$	# slabs	f.p.s.	Slow-down factor
2.5%	0.675%	28	0.8	26.58
2.5%	0.675%	179	0.13	163.61

**Table 3. Spiral CT colon data set. One rendering cycle takes 47 ms. The first row corresponds to use incremental slab thickness, while in the second row a constant slab thickness was used.**

*Conference Proceedings, ACM SIGGRAPH, 1998; pp 169–177.*

11. Wan M, Li W, Kreeger K, Bitter I, Kaufman A, Liang Z, Chen D, Wax M. 3D virtual colonoscopy with real-time volume rendering. In *SPIE's International Symposium on Medical Imaging 2000, Conference Proceedings, 2000.*
12. Kreeger K, Bitter I, Dachille F, Chen B, Kaufman A. Adaptive perspective ray casting. In *1998 Symposium on Volume Visualization, Conference Proceedings 1998; pp 19–20.*



**Authors' biographies:**



**Anna Vilanova Bartrolí** is currently working as a scientific staff member at the Institute of Computer Graphics and Algorithms at the Vienna University of Technology. In October 2001, she received her PhD from the same university. She is involved in the Vismed project (<http://www.vismed.at>). Her research interests include computer graphics, medical volume visualization, virtual Endoscopy and image processing. She received here MS in computer Science from the Universitat Politencnica de Catalunya (UPC).

**Eduard Groeller** is an associate professor at the Institute of Computer Graphics and Algorithms, Vienna University of Technology. In 1993 he received his PhD from the same university. His research interests include computer graphics, flow visualization and medical visualization (<http://www.cg.tuwien.ac.at/research/vis/>). He is heading the visualization group at the Institute of Computer Graphics and Algorithms. He is a member of the IEEE Computer Society, ACM (Association of Computing Machinery), GI (Gesellschaft für Informatik), OCG (Austrian Computer Gesellschaft).



**Rainer Wegenkittl** is a software engineer employed by Tiani Medgraph. In 1997 he received his PhD from Vienna University of Technology. He is a key software engineer in the VisMed project (<http://www.vismed.at>) and he is a key researcher at the VRVis Research. Center for Virtual Reality and Visualization (<http://www.vrvis.at>). His research interests include flow visualization and medical visualization.