EUROGRAPHICS 2000 / M. Gross and F.R.A. Hopgood
(Guest Editors)

Volume 19 (2000), Number 3

Gradient Estimation in Volume Data using 4D Linear Regression

László Neumann, Balázs Csébfalvi, Andreas König, Eduard Gröller

Institute of Computer Graphics, Vienna University of Technology,
Wien, Karlsplatz 13/186/2, A-1040, AUSTRIA
lneumann@elender.hu, {csebfalvi, koenig, groeller}@cg.tuwien.ac.at

Abstract

In this paper a new gradient estimation method is presented which is based on linear regression. Previous contextual shading techniques try to fit an approximate function to a set of surface points in the neighborhood of a given voxel. Therefore a system of linear equations has to be solved using the computationally expensive Gaussian elimination. In contrast, our method approximates the density function itself in a local neighborhood with a 3D regression hyperplane. This approach also leads to a system of linear equations but we will show that it can be solved with an efficient convolution. Our method provides at each voxel location the normal vector and the translation of the regression hyperplane which are considered as a gradient and a filtered density value respectively. Therefore this technique can be used for surface smoothing and gradient estimation at the same time.

1. Introduction

In direct volume rendering the quality of the generated image is strongly influenced by the estimation method used for the normal vector computation. Volumetric data is usually obtained by sampling continuous objects and after the discretization the exact surface normals are not available anymore. Therefore the inclination of the surfaces is estimated investigating a close neighborhood of a given voxel. A possible way of evaluation of different normal computation techniques is to discretize continuous geometrical models and to compare the estimated normals to the exact original ones. Unfortunately, this strategy cannot be used for practical data sets, like medical CT scans, because the exact normal vectors are not known. In a typical volume, there are no sharp edges and the surfaces are rather smooth. Therefore one can expect that the surfaces and the contours of different organs are displayed smoothly with reduced staircase artifacts. In order to fulfill this requirement our method integrates the filtering and the normal computation into one process.

In [3] Yagel overviews several methods for discrete normal estimation and analyses their performance. Depending on the neighborhood considered, these techniques can be classified into two fundamentally different categories as *image space* and *object space* methods.

Image space techniques take only the 2D neighborhood in the projected image into account, therefore they are view-dependent. *Depth-gradient shading* [4, 5] as a representative of image space methods, approximates the gradient from the z-buffer calculating the differences between the depths of the given and the neighboring pixels. This approach produces sharp contours where in the neighboring pixels different objects are visible or where there is a drastic jump between the depth values. In order to avoid this artifact *context sensitive normal estimation* can be used which takes also these object and slope discontinuities into account. The basic idea of this approach can be applied to object space techniques too[3].

Object space methods estimate the normal according to the 3D neighborhood of the given voxel. *Constant shading* [6] which is based on the cuberille method is an early example of this category. The voxels are considered as unit cuboids, and the normals at each point of a boundary surface are the true normals of the corresponding cuboid faces. *Normal-based contextual shading* [7, 8] is based on the cuberille method as well. This technique additionally takes the orientation of the adjacent visible faces into account increasing the number of possible normal vectors and giving a better impression about the inclination of a boundary surface. *Gray-level gradient shading* [9, 10] is used for volumes, where each voxel represents a gray-level value. The gradient vectors are estimated according to the neighborhood of the voxels using traditional derivative filters [1]. *Contextual shading* fits a local approximate plane [11] or a biquadratic [12, 13] function to the set of points that belong to the same isosurface. These methods are time-consuming and limited to a certain neighborhood. Bryant and Krumvieda [11] solve a set of linear equations by Gaussian elimination in order to obtain an approximate tangent plane at a given surface point. Webber's technique[12, 13] is similar, but in a 26-neighborhood the surface is approximated by a biquadratic function producing accurate results for objects with C^1 continuous faces.

According to our approach the normal estimation is extended to a 4D linear regression problem and not restricted to the approximation of an isosurface. In a local neighborhood the density function is approximated with a 3D hyperplane taking not only the surface points but all the neighboring voxels into account, using an appropriate weighting function. Since a plane is defined by a normal vector and a translation, a 4D linear equation system is solved in order to minimize the error of the approximation. This seems to be more complicated than the previous contextual shading techniques but we will show that it leads to a computationally efficient convolution, thus the linear equation does not need to be solved using the traditional time-consuming methods of linear algebra, like Gaussian elimination. Furthermore, our technique provides not only an estimated normal vector but a translation value as well, which can be considered as a filtered value for the given voxel location. By substituting the original density with the filtered value, smooth surfaces can be displayed and the staircase artifacts can be reduced.

In the next section the mathematical background of our regression-based normal-estimation method is discussed. In section 3 some ideas are presented how to use the approximating hyperplanes for interpolation inside a cubic cell instead of using the traditional trilinear interpolation. Section 4 describes how to weight the error contributions of neighboring voxels in calculation of the global regression error. In section 5 we present the implementation results and in section 6 we summarize the contribution of this paper.

2. Linear regression

Assuming that the origin of the coordinate system is translated into the position of the current voxel, the density function $f(x,y,z)$ in a close neighborhood can be approximated linearly according to the following formula:

$$f(x,y,z) \approx A\cdot x + B\cdot y + C\cdot z + D. \qquad (1)$$

This approximation tries to fit a 3D regression hyperplane onto the measured density values assuming that the density function changes linearly in the direction of the plane normal $n = [A,B,C]$. The value of D which is the approximate density value at the origin of the local coordinate system determines the translation of the plane.

Evaluating this approximation for the voxels of the local neighborhood the error can be measured using the following mean square error calculation:

$$E(A,B,C,D) = \sum_{k=0}^{26} w_k \cdot (A\cdot x_k + B\cdot y_k + C\cdot z_k + D - f_k)^2. \qquad (2)$$

The coordinates x_k, y_k, z_k denote the components of the neighboring voxel locations in the coordinate system translated into the center of the subvolume representing the local neighborhood. The measured density value in the kth voxel position is denoted by f_k. The error of the kth sample contributes to the global mean square error with weight w_k. The weighting function assumed to be an arbitrary, spherically symmetric function, which is monotonically decreasing as the distance from the origin is getting larger.

The k indices are assigned to the neighboring voxel locations row-continuously (Figure 1). For the sake of clarity but without loss of generality, we assume that only the 26-neighborhood is taken into account. In this case, the index k of voxel v[x,y,z] in the 26-neigborhood of the current voxel $c[0,0,0]$ is defined as:

$$k = (z+1)\cdot 9 + (y+1)\cdot 3 + x + 1. \qquad (3)$$

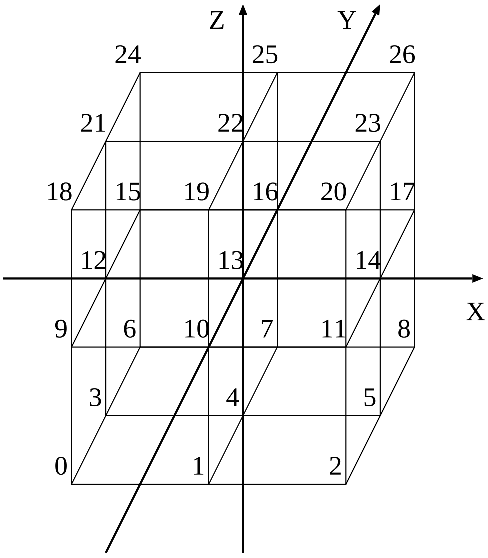

Figure 1: *Indexing of the neighboring voxels.*

In order to minimize the 4D error function $E(A,B,C,D)$, the partial derivatives according to the four unknown variables A, B, C, D are investigated:

$$\frac{\partial E}{\partial A} = 2 \cdot \sum_{k=0}^{26} w_k \cdot (A \cdot x_k + B \cdot y_k + C \cdot z_k + D - f_k) \cdot x_k, \quad (4)$$

$$\frac{\partial E}{\partial B} = 2 \cdot \sum_{k=0}^{26} w_k \cdot (A \cdot x_k + B \cdot y_k + C \cdot z_k + D - f_k) \cdot y_k, \quad (5)$$

$$\frac{\partial E}{\partial C} = 2 \cdot \sum_{k=0}^{26} w_k \cdot (A \cdot x_k + B \cdot y_k + C \cdot z_k + D - f_k) \cdot z_k, \quad (6)$$

$$\frac{\partial E}{\partial D} = 2 \cdot \sum_{k=0}^{26} w_k \cdot (A \cdot x_k + B \cdot y_k + C \cdot z_k + D - f_k). \quad (7)$$

In a minimum location of the error function these partial derivatives equal to zero. This condition leads to the following system of linear equations:

$$M \cdot \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \sum w_k f_k x_k \\ \sum w_k f_k y_k \\ \sum w_k f_k z_k \\ \sum w_k f_k \end{bmatrix}, \quad (8)$$

where

$$M = \begin{bmatrix} \sum w_k x_k^2 & \sum w_k x_k y_k & \sum w_k x_k z_k & \sum w_k x_k \\ \sum w_k x_k y_k & \sum w_k y_k^2 & \sum w_k y_k z_k & \sum w_k y_k \\ \sum w_k x_k z_k & \sum w_k y_k z_k & \sum w_k z_k^2 & \sum w_k z_k \\ \sum w_k x_k & \sum w_k y_k & \sum w_k z_k & \sum w_k \end{bmatrix}.$$

Note that the elements of the coefficient matrix M are constants, thus only the right side of the matrix equation depends on the measured f_k values. Assuming that the voxels are located at regular grid points, where the sampling distance is the same in the three major directions the equation is further simplified. In this case, the coefficient matrix M is a diagonal matrix since all the elements except the diagonal ones equal to zero because of symmetry reasons ($x_k, y_k, z_k \in \{-1, 0, 1\}$ and the weights w_k are symmetric to the origin, therefore each non-zero term in the sum has a pair with an opposite sign):

$$M = \begin{bmatrix} \sum w_k x_k^2 & 0 & 0 & 0 \\ 0 & \sum w_k y_k^2 & 0 & 0 \\ 0 & 0 & \sum w_k z_k^2 & 0 \\ 0 & 0 & 0 & \sum w_k \end{bmatrix}. \quad (9)$$

Such a linear equation can be solved very easily, since the inverse of the coefficient matrix is also a diagonal matrix containing in the diagonal the reciprocals of the original matrix elements. Thus the unknown vector $[A, B, C, D]$ is calculated by weighting the components of the right side. Let us introduce the following weights for each unknown variable:

$$w_A = \frac{1}{\sum_{k=0}^{26} w_k x_k^2}, \; w_B = \frac{1}{\sum_{k=0}^{26} w_k y_k^2},$$

$$w_C = \frac{1}{\sum_{k=0}^{26} w_k z_k^2}, \; w_D = \frac{1}{\sum_{k=0}^{26} w_k}. \quad (10)$$

The solution of the matrix equation leads to a simple linear convolution:

$$A = w_A \sum_{k=0}^{26} w_k f_k x_k, \; B = w_B \sum_{k=0}^{26} w_k f_k y_k,$$

$$C = w_C \sum_{k=0}^{26} w_k f_k z_k, \; D = w_D \sum_{k=0}^{26} w_k f_k. \quad (11)$$

Assuming that the sampling distances along the three major axes are the same the weights w_A, w_B, w_C are equal to each other. Thus these weights can be ignored since the estimated gradient $[A, B, C]$ has to be normalized anyway in order to obtain a surface normal of unit length. The gradient magnitude might also be used in the rendering stage for emphasizing the boundaries of isosurfaces. In this case the weights w_A, w_B, w_C can be ignored as well, since only the relative differences between the gradient magnitudes are important.

Note that the value of variable D is a normalized weighted sum of the measured values in the local neighborhood thus it can be considered as a filtered value. This is the result of the approximation in the origin of the local coordinate system ($f(0,0,0) = D$). Together with the approximate normal components these filtered values are stored in a newly generated volume. In this volume there is a strong correlation between the data values and the corresponding normals since in the grid points the error of the linear approximation which has been assumed in the normal estimation is minimal. Therefore in the ray casting process this volume is used instead of the original one. In a typical volume-rendering application, in order to reduce the noise in the data set and to smooth the surfaces low-pass filtering is used. This filtering process is completely separated from the gradient estimation. In our approach the smoothing and the normal estimation are performed in one step in a consistent way, using the same function for weighting the contribution of the neighboring voxels.

3. Interpolation

In direct volume rendering the approximate gradient vectors are usually calculated in advance at the grid points, in a preprocessing step. In the ray casting stage, a normal vector at an arbitrary sample point is calculated from the gradients of the eight closest voxels using trilinear interpolation. Wherever there is a big difference between the gradients at the eight corner voxels of the given cubic cell the typical staircase artifacts appear.

In order to avoid this problem the linear regression can be evaluated at the sample points along the viewing rays as well yielding a continuous reconstruction of the density function. Generally, the coefficient matrix M in Equation 8 will not be diagonal because of the asymmetric weights (the distances from the grid points of the neighborhood are different). Furthermore the entries depend on the position. Although the solution of the linear equation requires just a matrix multiplication, the evaluation of the inverse coefficient matrix is rather expensive computationally. This problem can be handled by dividing each cell into subcells with a regular subgrid (Figure 2). The inverse matrix is evaluated in advance for each corner point of the subcells. This has to be done only once for one generic cell. In the ray casting process trilinear interpolation is applied for the subcells, where the normals at the corner points are calculated using the precalculated inverse matrices. This modification provides a more accurate approximation of the density function although it increases the rendering time.

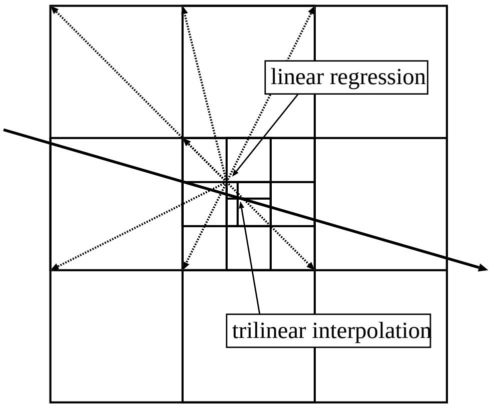

Figure 2: *Subdivision of the original grid.*

Another alternative is to use the approximating hyperplanes for density interpolation. First a density d_0 is computed from the filtered values at the eight closest grid points (which are the translations of the approximating hyperplanes), using trilinear interpolation. The obtained value d_0 cannot be larger than the maximum corner density of a cubic cell. Taking also the inclination of the surface into account another density d_1 is calculated the following way. The current sample location is substituted into the plane equations at the eight closest grid points and d_1 is trilinearly interpolated from the obtained values. Since this computation is not restricted to a cubic cell, the value d_1 might be larger than the maximum corner density depending on the influence of the neighboring cells. In order to sample the densities along the rays an arbitrary normalized weighted sum $\mu_0 \cdot d_0 + (1-\mu_0) \cdot d_1$ can be used. Increasing the weight μ_0 the influence of the local inclination is getting stronger and setting μ_0 to one results the traditional trilinear interpolation. In our experiments we used the value $\mu_0 = 0.5$ in order to interpolate the densities with a quadratic function. It can be considered as an acceptable compromise between trilinear and B-spline interpolation.

4. The weighting function

The weighting function w_k of the convolution can be an arbitrary, spherically symmetric function which is, apart of the origin, monotonically decreasing as the distance from the origin is getting larger. For example, the reciprocal of the square of the Euclidean (or Manhattan) distance can be used for weighting the neighboring voxels:

$$w_k = \begin{cases} 0 \text{ if } k = 13 \\ \frac{1}{d_k} \text{ otherwise,} \end{cases} \tag{12}$$

where d_k is the distance of the kth neighboring voxel from the central voxel. Note that the classical gradient estimation based on central differences is the special case of our method using the following weighting function:

$$w_k = \begin{cases} 1 \text{ if } k = 4 \text{ or } k = 22 \\ \text{ or } k = 10 \text{ or } k = 16 \\ \text{ or } k = 12 \text{ or } k = 14 \\ 0 \text{ otherwise.} \end{cases} \tag{13}$$

Thürmer‘s technique [2] which has been proposed for normal estimation in binary volumes is also a special case of our method. According to this approach the N_x, N_y, and N_z components of the estimated normal vector are calculated according to the following formula:

$$N_x = \sum_{k=0}^{26} w_k \sigma_k x_k,\ N_y = \sum_{k=0}^{26} w_k \sigma_k y_k,\ N_z = \sum_{k=0}^{26} w_k \sigma_k z_k, \tag{14}$$

where $\sigma_k = 1$ if the value of the kth binary voxel in the certain neighborhood is one and zero otherwise. Having a binary volume the density function f takes only the values of zero and one therefore in this special case our method provides the same normal components. It can be considered as a generalization of the previous normal estimation techniques, and can be used for gray-scale and binary data sets as well. Furthermore, our approach provides also a filtered value which is consistent to the estimated normal vector. Substituting the original densities with the filtered values the typical staircase artifacts of direct volume rendering can be reduced.

5. Implementation

The proposed normal estimation method has been tested on binary and gray-scaled data sets. Figure 3 shows a binary

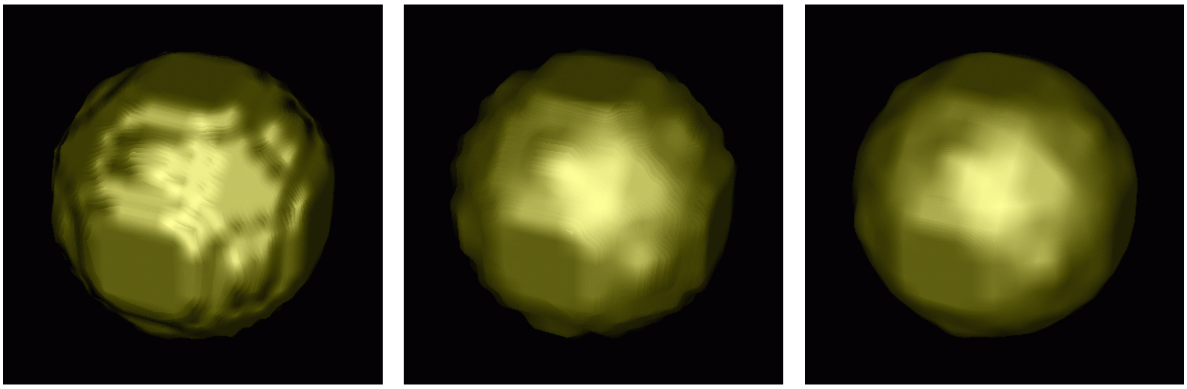

Figure 3: *Normal estimation on binary volume data using central differences (left), Thürmer's method (middle), and linear regression, where* $\mu_0 = 0.5$ *(right).*

volume of resolution $20 \times 20 \times 20$ obtained by discretization of a sphere.

The left image was rendered calculating central differences to estimate the surface normals, therefore the typical staircase artifacts appear. In the middle image, where normals are estimated from the 26-neighborhood of the voxels using Thürmer's method the surface is much smoother but the contour of the object has the same discontinuities as in the left image. The right image was rendered using our method, where the regression plane at each voxel location was calculated from the voxels of the 26-neighborhood. According to the linear regression the original data values are allowed to be modified in order to minimize the error of the approximation. Therefore the contour of the object is smoother and approximates the original contour much better than in the previous two images. Processing binary volumes, Thürmer's technique [2] and our method provides exactly the same normal vectors at the grid points. Nevertheless, the intersection points, where the normals are evaluated using trilinear interpolation, are different since the linear regression slightly changes the original data values. Although the estimated normal components are the same the surface in the right image is much smoother since the calculated intersection points are closer to the exact intersection points of the sphere and the viewing rays. These images clearly show that in rendering binary data sets not only the estimated normal components are important but also the sample locations, where the interpolated normals are evaluated.

Figure 4 shows a gray-scale data set obtained by a CT scan of a lobster. The data set has been rendered calculating central differences (left image) and using linear regression (right image) for gradient estimation. Although, in the right image some high frequency details are filtered, the contours of the different parts of the body are much sharper then in the left one, therefore they can be distinguished more easily. For example, the location and the shape of the legs can be perceived much better in the right image providing stronger spatial impression. In contrast, the left image contains some noisy regions, where the topology of the object cannot be recognized at all.

Having high resolution data sets, it is worthwhile to take a larger neighborhood into account for the linear regression calculation without significant loss of high frequency details. Figure 5 (see color plates) shows a human skull segmented from a CT scan of resolution $256 \times 256 \times 225$. The left image was rendered calculating the normals from the 3^3 neighborhood while in the right image the normals were estimated according to the 4^3 neighborhood. Note that the top of the skull in the right image is much smoother than in the left one and the staircase artifacts are less recognizable, while in high-frequency areas there is no significant difference.

Our method has been tested using also complex transfer functions. The images in Figure 6 (see color plates) have been rendered with an opacity function emphasizing the soft tissue and the bone. Using linear regression for gradient estimation (right) rather than calculating the central differences (left) the rendered image seems to be more realistic because of the antialiasing.

6. Conclusion

In this paper a new gradient estimation approach has been presented which is based on 4D linear regression. It has been shown that it is worthwhile to use the same function weighting the contribution of the neighboring voxels for filtering and for gradient computation yielding strong correlation between the filtered data values and the estimated normal vectors. Some previous normal computation techniques are special cases of our method thus it can be considered as a generalized solution with a clarified mathematical background. The presented technique can be used for gray-scale and binary data sets as well. Previous contextual shading methods are rather expensive computationally since they try to fit a linear or biquadratic function on the set of surface points

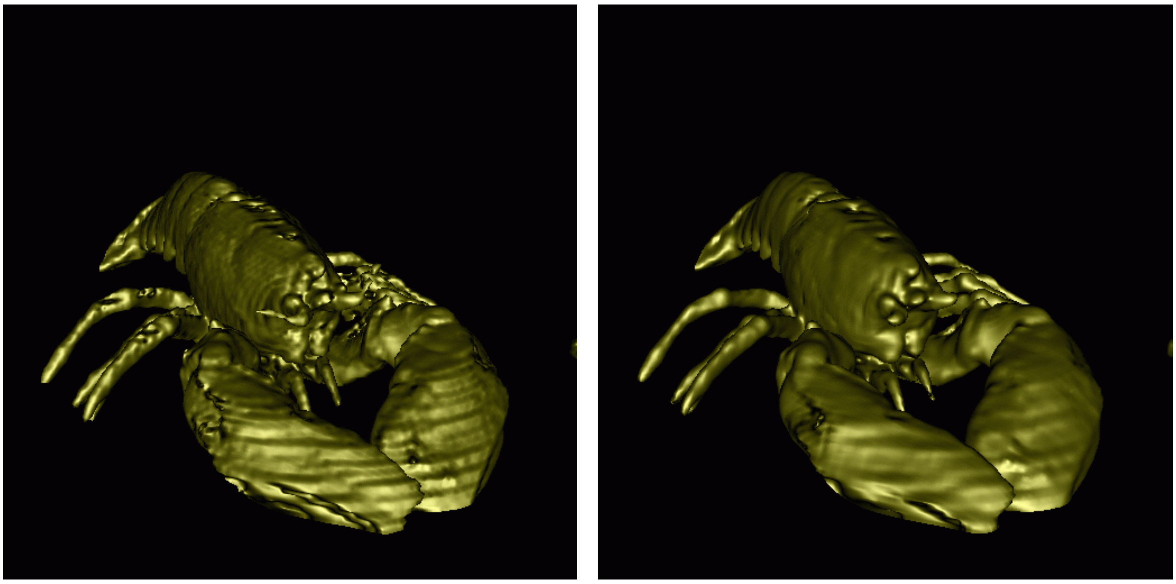

Figure 4: *A lobster rendered calculating central differences for gradient estimation (left) and using linear regression (right).*

and it requires the solution of a system of linear equations. In contrast, our approach approximates the density function itself with a 3D regression hyperplane and it leads to a computationally efficient convolution.

Acknowledgements

This work has been funded by the $\mathrm{V}^{\mathrm{is}}\mathrm{M}^{\mathrm{ed}}$ project (`http://www.vismed.at`). $\mathrm{V}^{\mathrm{is}}\mathrm{M}^{\mathrm{ed}}$ is supported by *Tiani Medgraph*, Vienna (`http://www.tiani.com`), and by the *Forschungsförderungsfond für die gewerbliche Wirtschaft*. The CT scan of the human head was obtained from the Chapel Hill Volume Rendering Test Dataset. The data was taken on the General Electric CT scanner and provided courtesy of North Carolina Memorial Hospital. The other CT data sets were provided by *Tiani Medgraph*.

References

1. T. Möller, R. Machiraju, K. Müller, R. Yagel. A Comparision of Normal Estimation Schemes. *Proceedings of IEEE Visualization '97*, pages 19–26, 1997.

2. G. Thürmer, C. A. Wüthrich. Normal Computation for Discrete Surfaces in 3D Space. *Computer Graphics Forum (Proceedings of EUROGRAPHICS '97)*, pages 15–26, 1997.

3. R. Yagel, D. Cohen, A. Kaufman. Normal Estimation in 3D Discrete Space. *The Visual Computer*, 8, pages 278–291, 1992.

4. D. Cohen, A. Kaufman, R. Bakalash, S. Bergman. Real Time Discrete Shading. *The Visual Computer*, 6, pages 16–27, 1990.

5. Y. W. Tam, W. A. Davis. Display of 3D Medical Images. *Proceedings of Graphics Interface '88*, pages 78–86, 1988.

6. G. T. Herman, H. K. Liu. Three-dimensional Display of Human Organs from Computed Tomograms. *Computer Graphics and Image Processing*, 9(1), pages 1–121, 1979.

7. G. T. Herman, J. K. Udupa. Display of Three-dimensional Discrete Surfaces. *Proceedings of the SPIE*, vol. 283, pages 90–97, 1981.

8. L. S. Chen, G. T. Herman, R. A. Reynolds, J. K. Udupa. Surface Shading in the Cuberille Environment. *IEEE Computer Graphics and Applications*, 5(December), pages 33–43, 1985.

9. K. H. Höhne, R. Bernstein. Shading 3D Images from CT using Gray-level Gradients. *IEEE Transactions on Medical Imaging*, MI-5(1), pages 45–47, 1986.

10. K. H. Höhne, M. Bomans, A. Pommert, M. Riemer, U. Tiede, G Wiebecke. Rendering Tomographic Volume Data: Adequacy of Methods for Different Modalities and Organs. *3D Imaging in Medicine*, Springer-Verlag, pages 197–215, 1990.

11. J. Bryant, C. Krumvieda. Display of 3D Binary Objects: I-shading. *Computers and Graphics*, 13(4), pages 441–444, 1989.

12. R. E. Webber. Ray Tracing Voxel Data via Biquadratic Local Surface Interpolation. *The Visual Computer*, 6(1), pages 8–15, 1990.

13. R. E. Webber. Shading Voxel Data via Local Curved-surface Interpolation. *Volume Visualization*, (A. Kaufmann, ed.), IEEE Computer Society Press, pages 229–239, 1991.

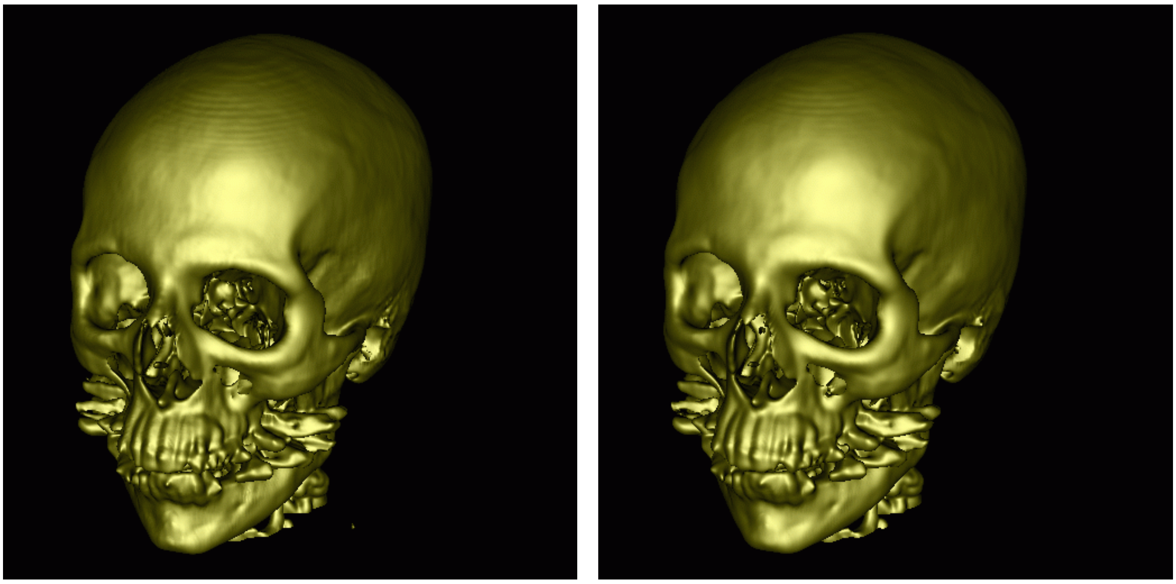

Figure 5: *Rendering of a human skull taking the* 3^3 *(left) and the* 4^3 *(right) neighborhood into account in the normal estimation using linear regression.*

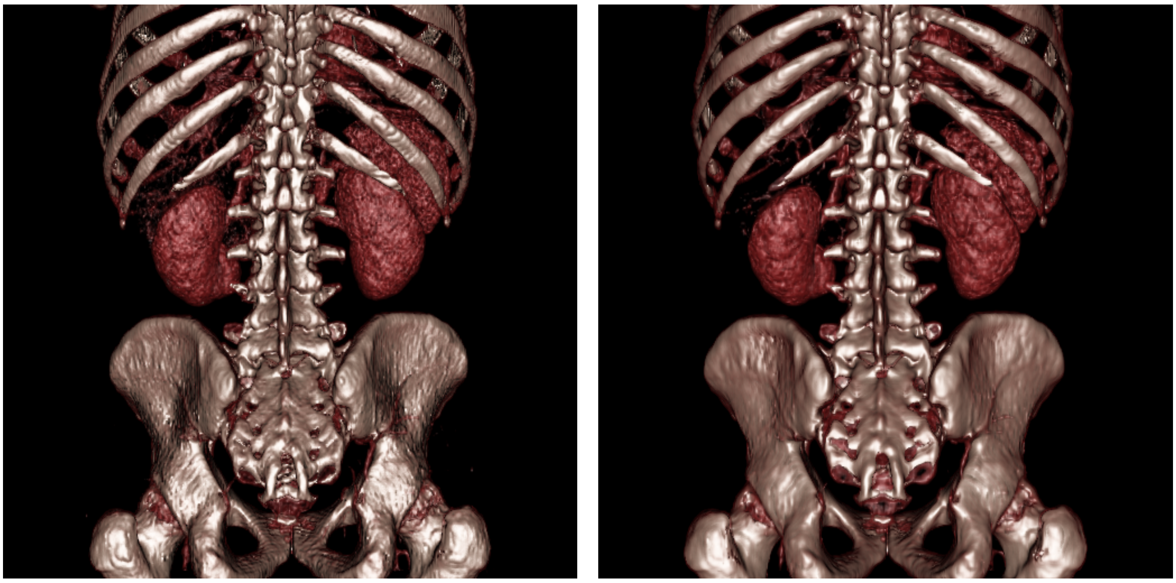

Figure 6: *Rendering of the kidneys and the skeleton using central differences (left) and linear regression (right) for gradient estimation.*