Directional 3D Edge Detection in Anisotropic Data: Detector Design and Performance Assessment

Marek Brejl and Milan Sonka
Department of Electrical and Computer Engineering
The University of Iowa, Iowa City, IA 52242

Running Title: Directional 3D Edge Detection in Anisotropic Data

Submitted to: Special Issue of Computer Vision and Image Understanding on Analysis of Volumetric Images

Address for correspondence:
Milan Sonka
Department of Electrical and Computer Engineering
The University of Iowa, Iowa City, Iowa, 52242
phone: (319)-335-6052, fax: (319)-335-6028

e-mail: milan-sonka@uiowa.edu

Abstract

A new directional 3D edge detector designed for anisotropic image data is reported. The detector is based on interpolating the image intensity function in a small neighborhood of every voxel by a tri-cubic polynomial. The analytical approximation of the image intensity function is used to compute the intensity function gradients. The developed edge detector uses a maximum average of directional derivatives of the approximated image intensity function over a small neighborhood to determine the gradient direction. Our method is directly applicable to anisotropic image data and it models the integrative character of data acquisition. With all these features, it remains computationally as expensive as any other convolution-based directional edge detector.

Quantitative measures of the 3D edge detection accuracy were employed to compare the performance of our new edge detector to that of the 3D Canny edge detector. 3D edges with step and ramp profiles with varying surface curvatures at the edge point as well as several levels of noise were used for the performance testing. The reported edge detector significantly outperformed the Canny edge detector in most experiments in anisotropic data as well as in data with superimposed noise. Another important property of the new edge detector is the ease of its implementation. Although its design required complex steps, the implementation employs straightforward 3D convolution in the volumetric image data using three pre-computed directional masks. Complete description of the gradient implementation is presented.

M. Brejl and M. Sonka are with the Department of Electrical and Computer Engineering, The University of Iowa, Iowa City IA 52242, e-mail: milan-sonka@uiowa.edu, http://www.icaen.uiowa.edu/~sonka

I. Introduction

Progress in volumetric data acquisition as well as cost-effectiveness of large-capacity data storage broadened the opportunities for an extended use of volumetric image data. Primarily in medical imaging and non-destructive testing, volumetric data sets are becoming standard. For example, methods of the volumetric medical data analysis allow probing of large portions of the human body and provide more complete information than 2D images are capable of. Despite the demonstrated utility, truly volumetric methods of image processing and analysis are not yet common and multi-dimensional data sets are frequently analyzed sequentially using two-dimensional analysis tools. The presented work offers an inherently three-dimensional (3D) solution to one of very important volumetric image processing problems – edge detection in 3D [1].

The first significant extensions of known 2D edge detectors into the 3D space were developed very recently. Zhang summarized the idea of 3D edge detection and proposed several general methods for extending 2D edge detectors into the 3D space [2]. As a simplest method, he proposed to use a 2D mask applied to the data in the three directions to determine the three derivatives. Zhang also suggested how to extend the 3D detectors into full 3D masks. Bhattacharya and Wild derived a directional $3 \times 3 \times 3$ Sobel-like edge detector, claiming a need for a simple and efficient 3D edge detector [3]. Mehrotra and Zhan derived an optimal 3D zero-crossing-based edge detector [4]. Their criteria of optimality were based on the Canny's criteria for edge detection [5]. However, as this was a Laplacian-like edge detector, it did not provide information about direction of edges, it provided only information about edge localization and approximate estimate of edge magnitude, which may not be sufficient in many real-life applications. Canny derived an optimal 1D edge detector according to his own criteria [5]. He proposed a circular symmetric extension to his 1D edge detector to construct edge detector of any higher dimension. Similarly, Spacek derived an optimal 1D edge detector based on a modification of Canny's criteria and suggested its extension to higher dimensions [6]. The extension uses the following substitution to allow solving a multi-dimensional edge detection problem using single-dimensional tools

$$r = \sqrt{\sum_{i=1,\ldots,N} x_i^2}$$

where r is the distance variable used in the 1D edge detector, N is the space dimension and x_i are the multi-dimensional space coordinates. However, both Canny and Spacek realized that extensions of the originally asymmetric 1D edge detectors to higher dimensional circular symmetric edge detectors leads to a loss of optimality.

One of the frequent problems faced by 3D edge detectors is their computational complexity. Simple, small-size edge detectors do not achieve good performance, while more complex large edge detectors are impractical because of the required computational time. Even more complicated is the need to reliably determine edges in volumetric image data with anisotropic voxels. Many of today's data acquisition methods use a different sampling rate in the z direction than in the x and y directions, x, y, z corresponding to a standard Cartesian coordinate system. None of the above-mentioned edge detectors can deal with anisotropic data. The typical approach is either to apply the 3D detectors to anisotropic data at the cost of obtaining biased results or to first employ some interpolation method to obtain isotropic voxel data – with the unfortunate byproduct of an additional increase of the already substantial data size.

In the following sections, a new approach to edge detection in anisotropic volumetric image data sets is reported together with a complete description of its simple and fast convolution-based implementation. Quantitative assessment of the edge detection performance and its comparison with that achieved by the 3D extension of the Canny edge detector [7] is provided. While the description is simplified to enhance readability, full details of the method's development are given in the Appendices.

II. Methods

A new directional 3D edge detector is reported that was inspired by the facet model of Haralick and Zuniga [8]. Our new 3D edge detection is based on approximation of the original volumetric data in a small 3D voxel neighborhood by a tri-cubic polynomial. All edge-related computations are performed using an analytical approximation of the original data. The approximation is obtained by fitting a tri-cubic polynomial into the image data in the least square error sense. The integrative process of data acquisition is incorporated to further improve the approximation. The image gradient is determined as the gradient of the image intensity function in a direction in which the average directional derivative in some neighborhood of the central voxel is maximal. The resulting method for gradient computation yields accurate results and can be implemented as

a mask convolution with the mask size corresponding to the size of the voxel neighborhood within which the polynomial is fitted. Thus, the implementation complexity is the same as the complexity of other mask-based 3D edge detectors. Importantly, the tri-cubic polynomial data model is well suited for anisotropic data since it can incorporate different sampling rates in different directions with the cubic interpolation of the image data performed directly in the edge detector.

Let a standard Cartesian coordinate system be used (Fig. 1). Any vector in this space can be described using its x, y and z elements or by its magnitude and orientation. The orientation is described by the angle θ between the x axis and the vector projection into the $x-y$ plane and by the angle ϕ between the vector projection into the $x-y$ plane and the vector itself (Fig. 1).

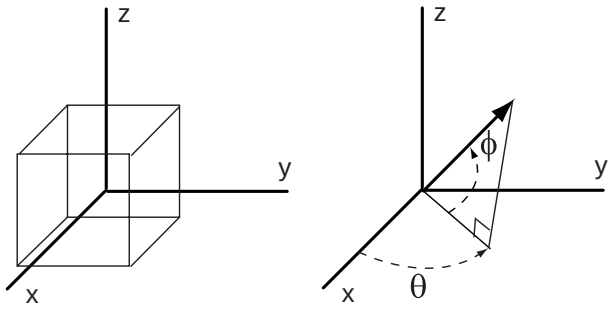


Fig. 1. Vector representation in the employed Cartesian coordinate system. The local coordinate system has its origin at the center of the investigated voxel.

A. Tri-cubic polynomial data model

For every voxel in the volumetric data, a tri-cubic polynomial is fitted into its 3D neighborhood. The tri-cubic polynomial $p(x,y,z)$ is defined as

$$\begin{aligned} p(x,y,z) = {} & K_1 + K_2x + K_3y + K_4z + K_5x^2 + K_6xy + \\ & + K_7xz + K_8yz + K_9y^2 + K_{10}z^2 + K_{11}x^3 + K_{12}x^2y + \\ & + K_{13}x^2z + K_{14}xy^2 + K_{15}xz^2 + K_{16}y^2z + K_{17}yz^2 + \\ & + K_{18}y^3 + K_{19}z^3 + K_{20}xyz \end{aligned} \tag{1}$$

where (x,y,z) denotes coordinates in the continuous space. Furthermore,

$$\begin{aligned} x &= x_{im}v_x \\ y &= y_{im}v_y \\ z &= z_{im}v_z \end{aligned} \tag{2}$$

where (x_{im}, y_{im}, z_{im}) are coordinates in the discrete (image) space and (v_x, v_y, v_z) are voxel dimensions. The local coordinate system is centered at the investigated voxel. Therefore, $(x,y,z) = (x_{im}, y_{im}, z_{im}) = (0,0,0)$ corresponds to the center of the investigated voxel. Realizing that 3D data acquisition frequently does not reflect ideal case of δ-function data sampling but rather it often includes integration over a small volume, this fact should be reflected in the edge detection process. The original data can be represented by the polynomial $p(x,y,z)$ from Eq. 1. After integrating and ideal sampling, the polynomial representation of the discrete image data is obtained

$$P(x,y,z) = \int_{x-dx}^{x+dx}\int_{y-dy}^{y+dy}\int_{z-dz}^{z+dz} p(\hat{x},\hat{y},\hat{z})\,d\hat{x}\;d\hat{y}\;d\hat{z} \tag{3}$$

where $(2d_x, 2d_y, 2d_z)$ are dimensions in continuous data space of the prism over which the integration is performed. As P models the discrete data, it can be expressed in image coordinates (x_{im}, y_{im}, z_{im}) by solving the integral (Eq. 3):

$$\begin{aligned} P(x_{im}v_x, y_{im}v_y, z_{im}v_z) &= K_1 + K_2x_{im}v_x + K_3y_{im}v_y + K_4z_{im}v_z + K_5(x_{im}^2v_x^2 + d_x^2/3) \\ &+ K_6x_{im}v_xy_{im}v_y + K_7x_{im}v_xz_{im}v_z + K_8y_{im}v_yz_{im}v_z + K_9(y_{im}^2v_y^2 + d_y^2/3) \\ &+ K_{10}(z_{im}^2v_z^2 + d_z^2/3) + K_{11}(x_{im}^3v_x^3 + x_{im}v_xd_x^2) \\ &+ K_{12}y_{im}v_y(x_{im}^2v_x^2 + d_x^2/3) + K_{13}z_{im}v_z(x_{im}^2v_x^2 + d_x^2/3) \\ &+ K_{14}x_{im}v_x(y_{im}^2v_y^2 + d_x^2/3) + K_{15}x_{im}v_x(z_{im}^2v_z^2 + d_z^2/3) \end{aligned}$$

$$
\begin{aligned}
&+ \quad K_{16} z_{im} v_z (y_{im}^2 v_y^2 + d_y^2/3) + K_{17} y_{im} v_y (z_{im}^2 v_z^2 + d_z^2/3) \\
&+ \quad K_{18} (y_{im}^3 v_y^3 + y_{im} v_y d_y^2) + K_{19} (z_{im}^3 v_z^3 + z_{im} v_z d_z^2) \\
&+ \quad K_{20} x_{im} v_x y_{im} v_y z_{im} v_z
\end{aligned} \tag{4}
$$

The function $P(x_{im}v_x, y_{im}v_y, z_{im}v_z)$ shall be fitted into the discrete volumetric data. The fit is computed in the least squares error sense so that

$$
e(x_{im}, y_{im}, z_{im}) = \sum_{x_{im} \in X_{im}} \sum_{y_{im} \in Y_{im}} \sum_{z_{im} \in Z_{im}} \{P(x_{im}v_x, y_{im}v_y, z_{im}v_z) - f(x_{im}, y_{im}, z_{im})\}^2 \tag{5}
$$

is minimized, where $f(x_{im}, y_{im}, z_{im})$ denotes the image intensity function and X_{im}, Y_{im} and Z_{im} represent the voxel neighborhood to which the polynomial is fitted (e.g., $X_{im} = Y_{im} = Z_{im} = \{-2, -1, 0, 1, 2\}$ for the $5 \times 5 \times 5$ neighborhood). The computation of coefficients $K_1, \ldots, K_{20}$ is presented in Appendix I.

B. Gradient computation

The ultimate goal is to compute the gradient of the intensity function in the continuous space. Therefore, the polynomial in Eq. 1 is used to represent the interpolation of the image intensity function in an analytical form and to compute derivatives. The simplest way to compute the gradient is to compute partial derivatives of $p(x, y, z)$ in x, y and z directions:

$$
grad(p) = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right)_{(x,y,z)=(0,0,0)} = (K_2, K_3, K_4) \tag{6}
$$

The gradient is computed at the point $(0, 0, 0)$ since the origin of the local coordinate system corresponds to the center of the investigated voxel. This would be a perfectly valid approach if the analytical polynomial perfectly corresponded to the image data. Unfortunately, this is not usually the case. An enhancement to gradient computation from an interpolating function proposed by Haralick and Zuniga in [8] for 2D images can be extended into the 3D space.

Let a gradient $p'_{\theta,\phi}(x, y, z)$ of $p(x, y, z)$ be computed in a direction described by (θ, ϕ) (the direction is represented by a unit vector (n_x, n_y, n_z), see Appendix II-A).

$$
p'_{\theta,\phi}(x, y, z) = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right) (n_x, n_y, n_z) = \frac{\partial p}{\partial x} \cos\theta \cos\phi + \frac{\partial p}{\partial y} \sin\theta \cos\phi + \frac{\partial p}{\partial z} \sin\phi \tag{7}
$$

The orientation in which the directional derivative reaches its maximum is specified by

$$
\begin{aligned}
\tan\theta &= \frac{\partial p/\partial y}{\partial p/\partial x} \\
\tan\phi &= \frac{\partial p/\partial z}{\sqrt{(\partial p/\partial x)^2 + (\partial p/\partial y)^2}}
\end{aligned} \tag{8}
$$

which corresponds to the gradient in Eq. 7. Now, instead of computing the direction in which $p'_{\theta,\phi}(0, 0, 0)$ is maximal, a direction is computed for which the average directional derivative in some neighborhood of the central voxel is maximal. Thus, local inaccuracy resulting from inaccurate interpolation of the image data can be better compensated for. Particularly, the average is computed over a prism rotated into the direction of (θ, ϕ);

$$
F_{\theta,\phi} = \frac{1}{8LVW} \int_{-W}^{W} \int_{-V}^{V} \int_{-L}^{L} p'_{\theta,\phi}(x, y, z) d\bar{x} d\bar{y} d\bar{z} \tag{9}
$$

where $(\bar{x}, \bar{y}, \bar{z})$ are coordinates in the coordinate system rotated into (θ, ϕ) direction, $(2L, 2V, 2W)$ are dimensions of the prism (Fig. 2); $p'_{\theta,\phi}(x, y, z)$ is computed in the new coordinates as

$$
\begin{aligned}
p'_{\theta,\phi}(x, y, z) \quad = \quad & p'_{\theta,\phi}(\bar{x} \cos\theta \cos\phi - \bar{y} \sin\theta - \bar{z} \cos\theta \sin\phi, \\
& \bar{x} \sin\theta \cos\phi + \bar{y} \cos\theta - \bar{z} \sin\theta \cos\phi, \\
& \bar{x} \sin\phi + \bar{z} \cos\phi)
\end{aligned} \tag{10}
$$

Transformation matrices for computation of the rotated coordinate systems are shown in Appendix II-A. Solving the integral (Eq. 9) yields

$$
F_{\theta,\phi} = \frac{1}{3} A(K_i, \theta, \phi) L^2 + \frac{1}{3} B(K_i, \theta, \phi) V^2 + \frac{1}{3} C(K_i, \theta, \phi) W^2 + D(K_i, \theta, \phi) \tag{11}
$$

where functions $A(K_i, \theta, \phi), B(K_i, \theta, \phi), C(K_i, \theta, \phi)$ and $D(K_i, \theta, \phi)$ are given in Appendix II-B.

Complexity of computing maxima of the integral $F_{\theta,\phi}$ (Eq. 11) is prohibitive for practical implementation. Therefore, instead of using an arbitrary prism, a cubical voxel neighborhood is considered for directional

derivative averaging ($L = V = W$). Under this condition, significant simplification occurs and the integral can be evaluated as

$$\begin{aligned} F_{\theta,\phi} &= (K_2 + \frac{1}{3}L^2K_{14} + \frac{1}{3}L^2K_{15} + L^2K_{11})\cos\theta\cos\phi \\ &+ (K_3 + \frac{1}{3}L^2K_{12} + \frac{1}{3}L^2K_{17} + L^2K_{18})\sin\theta\cos\phi \\ &+ (K_4 + \frac{1}{3}L^2K_{13} + \frac{1}{3}L^2K_{16} + L^2K_{19})\sin\phi \end{aligned} \tag{12}$$

Then, the direction (θ, ϕ) that maximizes $F_{\theta,\phi}$ must be determined. Denoting

$$\begin{aligned} F_X &= K_2 + \frac{1}{3}L^2K_{14} + \frac{1}{3}L^2K_{15} + L^2K_{11} \\ F_Y &= K_3 + \frac{1}{3}L^2K_{12} + \frac{1}{3}L^2K_{17} + L^2K_{18} \\ F_Z &= K_4 + \frac{1}{3}L^2K_{13} + \frac{1}{3}L^2K_{16} + L^2K_{19} \end{aligned} \tag{13}$$

and setting the first partial derivatives of $F_{\theta,\phi}$ equal to zero

$$\begin{aligned} \frac{\partial F_{\theta,\phi}}{\partial\theta} &= -F_X\sin\theta\cos\phi + F_Y\cos\theta\cos\phi = 0 \\ \frac{\partial F_{\theta,\phi}}{\partial\phi} &= -F_X\cos\theta\sin\phi - F_Y\sin\theta\sin\phi + F_Z\cos\theta = 0 \end{aligned} \tag{14}$$

the final solution is obtained

$$\begin{aligned} \theta^{MAX} &= \arctan\frac{F_Y}{F_X} \\ \phi^{MAX} &= \arctan\frac{F_Z}{\sqrt{F_X^2 + F_Y^2}} \\ F_{\theta,\phi}^{MAX} &= \sqrt{F_X^2 + F_Y^2 + F_Z^2} \end{aligned} \tag{15}$$

where the MAX symbol denotes the values of θ and ϕ for which $F_{\theta,\phi}$ reaches its maximum value $F_{\theta,\phi}^{MAX}$.

Thus, the gradient can be represented as (F_X, F_Y, F_Z). It can be observed that for $L = 0$ the result reduces to (K_2, K_3, K_4), which is the maximum gradient at the center of the processed voxel (Eq. 6).

C. Implementation

Although our edge detector is derived following a complicated process, its implementation is fairly easy. Careful observation can reveal that the 3D edge detector can be implemented by computing three 3D masks M_X, M_Y and M_Z that are then used for convolution with the image data, to obtain gradients (F_X, F_Y, F_Z) at every voxel. The overall edge detection algorithm can be summarized as:

1. Set the following properties of the detector according to the input image data
- voxel size (v_x, v_y, v_z),
- size of the prism over which the data are integrated during the acquisition process – (d_x, d_y, d_z),
- mask size (n_x, n_y, n_z) (odd numbers) to get the sets

$$\begin{aligned} X_{im} &= \{-n_x/_{int}2, \ldots, n_x/_{int}2\} \\ Y_{im} &= \{-n_y/_{int}2, \ldots, n_y/_{int}2\} \\ Z_{im} &= \{-n_z/_{int}2, \ldots, n_z/_{int}2\} \end{aligned}$$

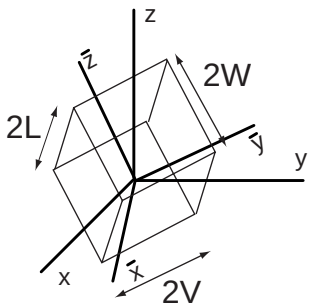


Fig. 2. Rotated prism over which integral $F_{\theta,\phi}$ is computed.

(where "$/_{int}$" denotes integer division), and

- neighborhood size for the directional gradient averaging L (recommended empirically derived value for the mask size of $5 \times 5 \times 5$ is $L = 2.1$).

2. Compute the three convolution masks. It can be seen from Eq. 13 that (F_X, F_Y, F_Z) may be computed from the polynomial coefficients K_i and from the integration dimension L. The coefficients K_i can be computed from Eqs. 23, 22, and 20. The masks M_X, M_Y and M_Z are applied to the image data to obtain F_X, F_Y and F_Z, respectively.

$$\begin{aligned}
M_X(x,y,z) &= \left[A_2 - \frac{1}{U_2}X_{21}(x^2+d_x^2) - \frac{1}{V_2}Y_{31}(y^2+d_y^2/3) - \frac{1}{W_2}Z_{31}(z^2+d_z^2/3)\right]x + \\
&\quad \frac{1}{3V_2}L^2\left[(y^2+d_y^2/3)Y_0 - Y_{31}\right]x + \frac{1}{3W_2}L^2\left[(z^2+d_z^2/3)Z_0 - Z_{31}\right]x + \\
&\quad \frac{1}{U_2}L^2\left[(x^2+d_x^2)X_2 - X_{21}\right]x \\
M_Y(x,y,z) &= \left[A_3 - \frac{1}{U_3}X_{31}(x^2+d_x^2/3) - \frac{1}{W_3}Y_{21}(y^2+d_y^2) - \frac{1}{V_3}Z_{31}(z^2+d_z^2/3)\right]y + \\
&\quad \frac{1}{3U_3}L^2\left[(x^2+d_x^2/3)X_0 - X_{31}\right]y + \frac{1}{3V_3}L^2\left[(z^2+d_z^2/3)Z_0 - Z_{31}\right]y + \\
&\quad \frac{1}{W_3}L^2\left[(y^2+d_y^2)Y_2 - Y_{21}\right]y \\
M_Z(x,y,z) &= \left[A_4 - \frac{1}{U_4}X_{31}(x^2+d_x^2/3) - \frac{1}{V_4}Y_{31}(y^2+d_y^2/3) - \frac{1}{W_4}Z_{21}(z^2+d_z^2)\right]z + \\
&\quad \frac{1}{3U_4}L^2\left[(x^2+d_x^2/3)X_0 - X_{31}\right]z + \frac{1}{3V_4}L^2\left[(y^2+d_y^2/3)Y_0 - Y_{31}\right]z + \\
&\quad \frac{1}{W_4}L^2\left[(z^2+d_z^2)Z_2 - Z_{21}\right]z
\end{aligned} \tag{16}$$

In the above equations, all necessary variables are directly computed from Eqs. 22 and 20, and the sets X, Y, Z are symmetric.

3. Compute the gradient (F_X, F_Y, F_Z) for every voxel (i, j, k) in the image volume as

$$\begin{aligned}
F_X(i,j,k) &= \sum_{x_{im}\in X_{im}}\sum_{y_{im}\in Y_{im}}\sum_{z_{im}\in Z_{im}} M_X(x_{im}v_x, y_{im}v_y, z_{im}v_z) f(i+x_{im}, j+y_{im}, k+z_{im}) \\
F_Y(i,j,k) &= \sum_{x_{im}\in X_{im}}\sum_{y_{im}\in Y_{im}}\sum_{z_{im}\in Z_{im}} M_Y(x_{im}v_x, y_{im}v_y, z_{im}v_z) f(i+x_{im}, j+y_{im}, k+z_{im}) \\
F_Z(i,j,k) &= \sum_{x_{im}\in X_{im}}\sum_{y_{im}\in Y_{im}}\sum_{z_{im}\in Z_{im}} M_Z(x_{im}v_x, y_{im}v_y, z_{im}v_z) f(i+x_{im}, j+y_{im}, k+z_{im})
\end{aligned} \tag{17}$$

where $f(i, j, k)$ is the image intensity function.

4. Determine gradient magnitude and orientation using Eq. 15.

III. Experimental methods

To provide comprehensive verification of the performance of the developed edge detector, many of its features were tested under a variety of conditions. The detector's behavior was assessed in edges of various gradient profiles and of various geometrical properties (different surface curvatures). The edge detection performance was also studied in anisotropic data, for different ratios of voxel dimensions, and in images with superimposed Gaussian noise.

Image data

To achieve an objective measurement of the gradient detection accuracy, simulated images of objects of known shapes and locations were used. 3D images were generated with the varying edge profiles, geometric properties, and noise levels:

1. Gradient profiles:

- step edge (the image intensity function f changed discontinuously at the edge location from $f_{LOW} = 50$ to $f_{HIGH} = 200$),

• ramp edge (the image intensity function was linearly dependent on the distance from the edge, intensity at the edge position being 128 with the slope of change of 10 units per voxel).
2. Geometric properties:
• planar edges,
• spherical edges (spheres of various radii were used ranging from 3 to 21 to test the detector's behavior in objects with different surface curvatures).
3. Data anisotropicity:
• The images were generated for various ratios of voxel dimensions x:y:z ranging from 1:1:1 to 1:1:6.
4. Noise levels:
• The images were generated for various values of variance of superimposed Gaussian noise with zero mean. The variance values ranged from 0% to approximately 100% of the edge magnitude.

In the computer-generated images, the gray level value of a voxel was determined as an integral of image intensity function over that voxel with the integration factor covering 100% of the voxel volume.

The experiments were performed by generating small $5 \times 5 \times 5$ subimages (equal to the neighborhood size of the employed detector). The gradient was computed at the center of these subimages. The characteristics were assessed using over 100 edge instances of different directions for each experiment. The overall number of tested edges was over $50,000$. The results reported below are based on assessing the edge detector's performance in $6,000$ edge instances.

Compared 3D edge detectors

The obtained edge information was compared to that obtained by using previously reported state-of-the-art 3D extension of the Canny edge detector, and its anisotropic modification. To clearly demonstrate the advantage of our gradient averaging approach, a simplified version our detector not containing the averaging step was also included in the comparisons. To summarize, four 3D edge detectors were tested with the parameter setting as follows:
• the new anisotropic edge detector with gradient averaging, $L = 2.1$,
• the new anisotropic edge detector without gradient averaging, $L = 0$. In this case, gradient computation was based solely on computing derivatives of the fitted interpolation polynomial,
• 3D extension of the Canny edge detector with $\sigma = 1.0$, and
• anisotropic modification of 3D extension of the Canny edge detector with $\sigma = 1.0$. The modification was done in the following way: the obtained gradients in every direction were divided by the relative voxel sizes in those directions. Rationale for this modification is based on the gradient detector behavior in ramp edges.
In all studied cases, $5 \times 5 \times 5$ edge detectors were employed.

Performance indices

To quantitatively assess the accuracy of edge detection, error indices were determined by comparison to the true edge properties.
• mean absolute error of the θ angle,
• mean absolute error of the ϕ angle,
• standard deviation of the gradient magnitude divided by its mean value (to assess the consistency of gradient strength measurement – the gradient magnitudes cannot be compared directly since edge detectors do not provide normalized magnitude values).

To determine the statistical significance of the achieved results, the edge orientation errors were compared using paired t-statistic, p value of 0.05 was considered significant.

IV. Results

Figs. 3 – 7 demonstrate the qualities of our new anisotropic edge detector. The results are divided into several groups according to the character of the underlying image data.

Edge detection performance in anisotropic data

Fig. 3 gives the comparison of edge detection performance in *planar step* edges in zero-noise data. In *planar ramp* edges in zero-noise data, the 3D version of the Canny edge detector yielded high errors in both magnitude and orientation while both version of our new edge detector as well as the modified Canny detector yielded zero errors. No graphical results are therefore presented. Fig. 4 gives the performance comparison in *spherical*

step edges, sphere radius 15, noise-free data. Similarly, Fig. 5 demonstrates edge detection errors in *spherical ramp* edges, radius 15, noise-free data. To keep the charts comprehensible, the θ errors are not shown since they are negligible in comparison to the ϕ angle errors. Rather, a range of θ errors is depicted.

Edge detection performance for various surface curvatures

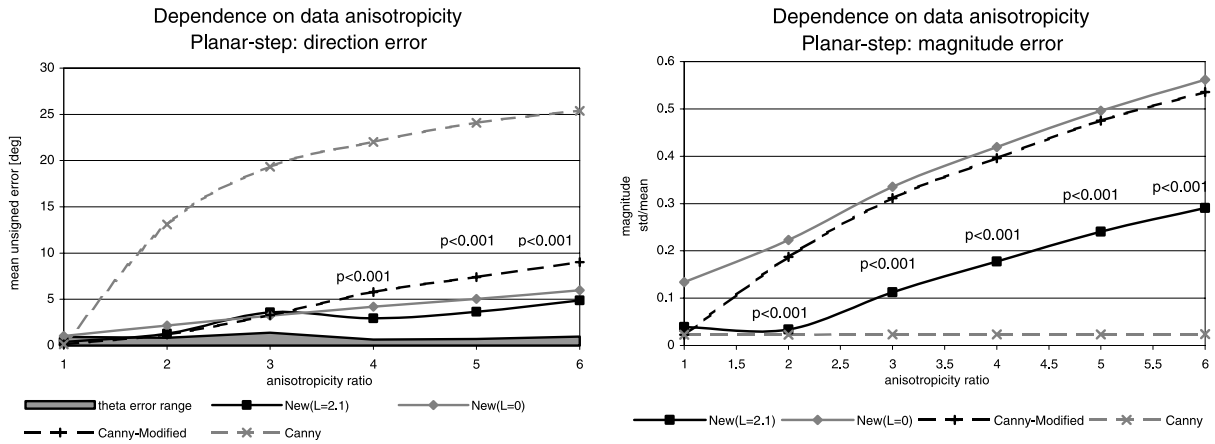
Figs. 6 a,b,e show errors of *spherical step* edge detection in noise-free isotropic data. Figs. 6 c,d,f provide similar comparison for noise-free *spherical ramp* edges. In both cases, the sphere radii ranged from 3 to 21. Notice that in Figs. 6 c,d,f y axes use a logarithmic scale to better depict the differences of detection errors.

Edge detection performance in noisy data

Figs. 7 a,b,e give edge detection errors for *spherical step* edges, radius 5 in isotropic data for the 8 noise levels. Fig. 7 c,d,f provide the same information for *spherical ramp* edges, radius 5.

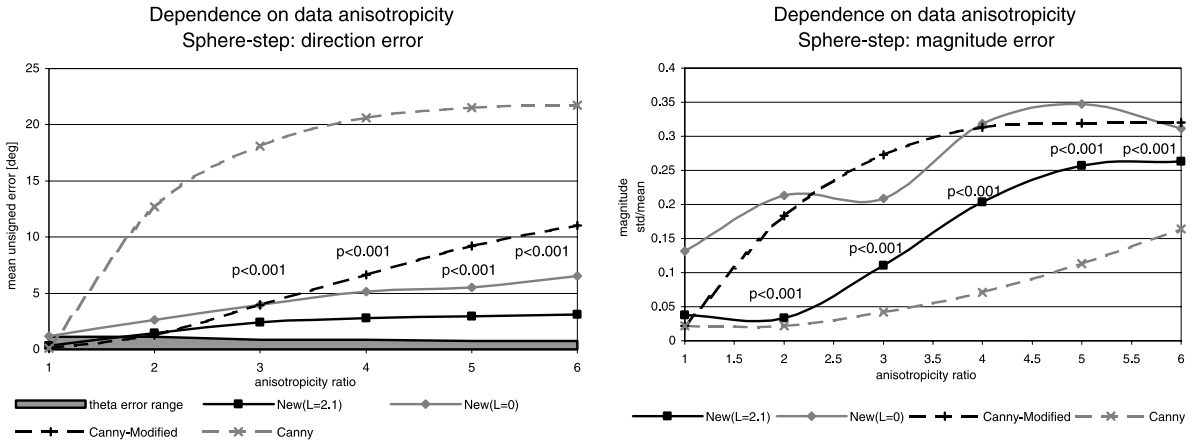
As was demonstrated in the error charts, our anisotropic edge detector exhibits consistently good performance in all the tested cases. Its simplified version that does not perform the gradient averaging step provides good results in noise-free data but it fails to yield acceptable results in noisy images. The Canny edge detector provides good results in all cases when isotropic data are used. However, it fails in anisotropic data yielding high edge-orientation errors. The employed modification to the Canny detector that divides the partial gradients by directional voxel dimensions was shown to reduce this error but not to fully compensate it. Statistical comparisons of the achieved results show that the novel anisotropic edge detector significantly outperforms the other tested detectors in most experiments in anisotropic image data as well as in data with superimposed noise (p values are given in charts).

The presented results give just a sample of all error assessment experiments that were performed to demonstrate the new edge detector qualities. While additional error indices can be presented, they would not alter the assessment of the new edge detection performance. Rather, the overall results are discussed in the Discussion section.



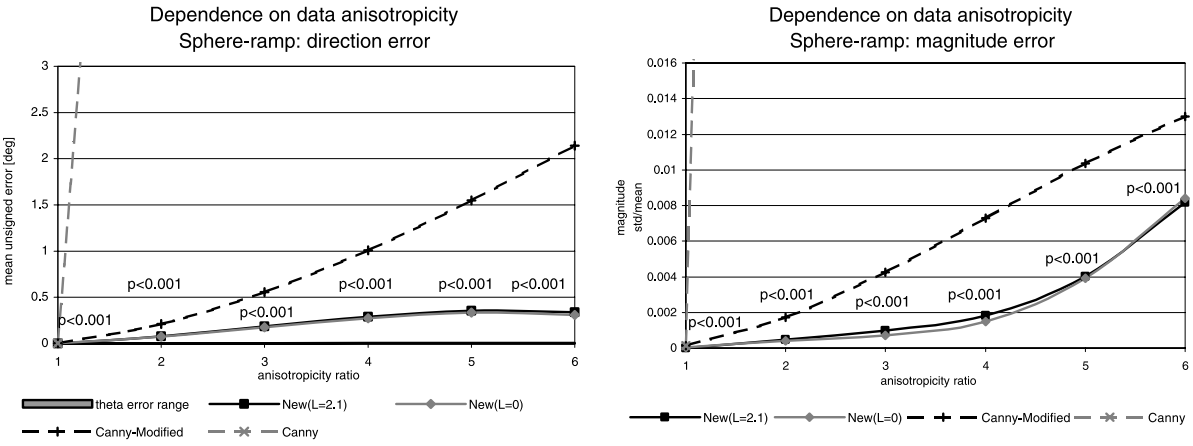
(a) (b)

Fig. 3. Detector performance in anisotropic planar noise-free step edges (New (L=2.1) ... novel anisotropic edge detector (parameter L=2.1), New (L=0) ... the simpler version of the novel anisotropic detector (without gradient averaging), Canny ... 3D extension of Canny edge detector, Canny-Modified ... anisotropic modification of Canny edge detector). (a) Directional errors, (b) magnitude errors. In experiments, in which paired t-statistic showed a significant difference between performance of the novel anisotropic edge detector (New (L=2.1)) and the modified 3D Canny edge detector (Canny-Modified), the p value is shown.



(a) (b)

Fig. 4. Detector performance in anisotropic spherical noise-free step edges (New (L=2.1) ... novel anisotropic edge detector (parameter L=2.1), New (L=0) ... the simpler version of the novel anisotropic detector (without gradient averaging), Canny ... 3D extension of Canny edge detector, Canny-Modified ... anisotropic modification of Canny edge detector). (a) Directional errors, (b) magnitude errors. In experiments, in which paired t-statistic showed a significant difference between performance of the novel anisotropic edge detector (New (L=2.1)) and the modified 3D Canny edge detector (Canny-Modified), the p value is shown.



(a) (b)

Fig. 5. Detector performance in anisotropic spherical noise-free ramp edges (New (L=2.1) ... novel anisotropic edge detector (parameter L=2.1), New (L=0) ... the simpler version of the novel anisotropic detector (without gradient averaging), Canny ... 3D extension of Canny edge detector, Canny-Modified ... anisotropic modification of Canny edge detector). (a) Directional errors, (b) magnitude errors. In experiments, in which paired t-statistic showed a significant difference between performance of the novel anisotropic edge detector (New (L=2.1)) and the modified 3D Canny edge detector (Canny-Modified), the p value is shown.

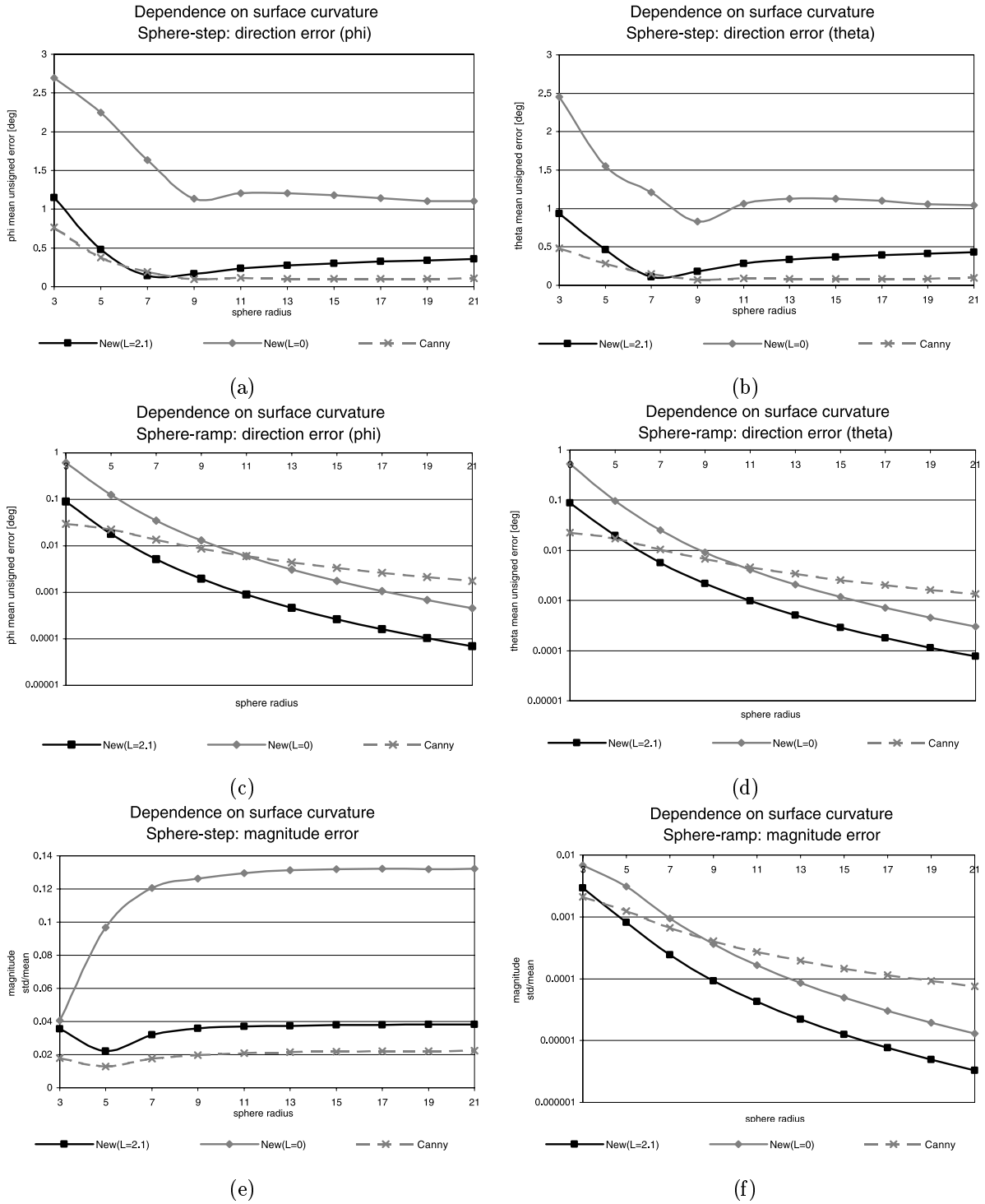


Fig. 6. Detector performance in isotropic spherical noise-free edges of varying curvature (New (L=2.1) ... novel anisotropic edge detector (parameter L=2.1), New (L=0) ... the simpler version of the novel anisotropic detector (without gradient averaging), Canny ... 3D extension of Canny edge detector, Canny-Modified ... anisotropic modification of Canny edge detector). (a) Directional errors of ϕ angle for step edges, (b) directional errors of θ angle for step edges, (c) directional errors of ϕ angle for ramp edges, (d) directional errors of θ angle for ramp edges, (e) magnitude errors for step edges, (f) magnitude errors for ramp edges.

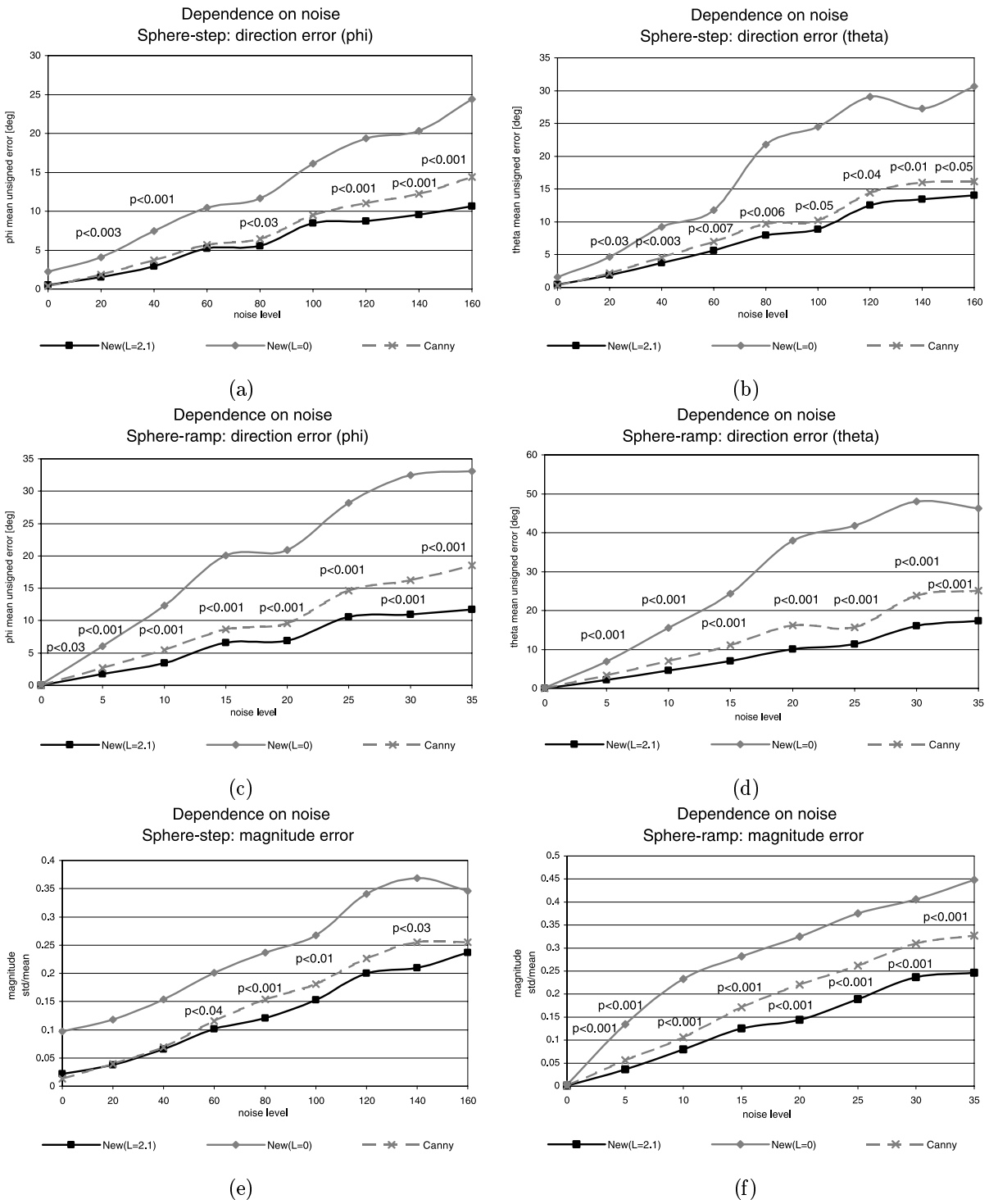


Fig. 7. Detector performance in isotropic spherical edges with varying noise (New (L=2.1) ... novel anisotropic edge detector (parameter L=2.1), New (L=0) ... the simpler version of the novel anisotropic detector (without gradient averaging), Canny ... 3D extension of Canny edge detector, Canny-Modified ... anisotropic modification of Canny edge detector). (a) Directional errors of ϕ angle for step edges, (b) directional errors of θ angle for step edges, (c) directional errors of ϕ angle for ramp edges, (d) directional errors of θ angle for ramp edges, (e) magnitude errors for step edges, (f) magnitude errors for ramp edges. In experiments, in which paired t-statistic showed a significant difference between performance of the novel anisotropic edge detector (New (L=2.1)) and the modified 3D Canny edge detector (Canny-Modified), the p value is shown.

V. Discussion

The reported anisotropic 3D edge detector has proved to be accurate over a broad spectrum of edge instances. Its comparison to the Canny edge detector revealed that the novel anisotropic edge detector significantly outperforms the Canny edge detector in both anisotropic data and data with superimposed noise. The new edge detector utilizes analytically computed gradients from polynomial interpolation of the image intensity function. Consequently, the results are influenced by the quality of the fit – the better the fit, the better the edge detection results. Generally, a better fit is obtained for edges with gradient profiles that that can be described by polynomials of lower orders (e.g., ramp edges) than for edges with abrupt changes in the image intensity function (e.g., step edges). Comparing, for example the absolute errors of ϕ for step and ramp edges as given in Figs. 4 and 5, it is seen that the ϕ mean error for spherical step edges is about 3° for the relative anisotropicity of 6, while it is less than 0.5° for the ramp edges under otherwise identical conditions. Similar behavior can be observed in planar step edges with the ϕ error of about 5° (Fig. 3), while the ramp edges are determined after an exact fit and thus yield zero errors. Our performance assessment uses these step and ramp edge profiles as examples from the two ends of the application spectrum, the step edge as one of the least-suited edge profiles for the novel detector and the ramp edge as one of the well-suited profiles. The detector is expected to perform comparably well in comparison to the 3D Canny edge detector in other edge profiles.

The main feature of the reported edge detector is its designed ability to perform in anisotropic image data. The experimental validation demonstrated this feature in data sets for which the anisotropicity was created by changing the z dimension of the voxel. Consequently, the increased error is likely to occur in the edge properties that are z-directional. In other words, errors of the ϕ angle can be expected. Errors in the θ angle are expected to be negligible. The error of the ϕ angle, while having an increasing tendency with the increased anisotropicity, remains low. It is consistently below 5° for z-voxel dimensions up to 6 times the x and y dimensions. In comparison, it is important to realize that the 3D extension of the Canny edge detector performs poorly when applied to anisotropic data. Its mean unsigned error of ϕ approaches 30°. When the Canny edge detector is modified to reflect anisotropicity as proposed above, its edge orientation accuracy improves substantially. Still, in the majority of the performed experiments, it is outperformed by our new anisotropic edge detector. Even the simplified version of our new detector that does not include the gradient averaging step ($L = 0$) performed well in the anisotropic data with orientation and magnitude errors comparable to those achieved by the full version of the new detector.

Our new 3D edge detector is designed to work in anisotropic data with different voxel sizes in any (or all) direction. Considering the fact, that a local fit to the image intensity function is performed, it can also be successfully applied to data with changing relative voxel sizes over the image volume, as long as the changes of the relative voxel dimensions are not abrupt (because of region overlapping). This may become useful, e.g., in potential variable-resolution medical image data of the future. Even though our detector performed very well in anisotropic data, it has to be kept in mind that for any edge detector to yield good results, the sampled data have to preserve the edge information. In other words, the sampling theorem must not be violated.

Testing the detector performance on objects with various surface curvatures (Fig. 6) showed that higher directional errors are obtained for surfaces with higher curvature (lower sphere radius). With the decreasing curvature, the errors converge to the values observed on planar surfaces. By design, the image data were isotropic and so the errors of θ and ϕ are approximately the same. As far as the four edge detectors are concerned, they all performed comparably well. The obvious difference is in a higher error of the proposed detector in step edges and lower error in ramp edges as discussed earlier.

Influence of a superimposed Gaussian zero mean noise was also studied (Fig. 7). The Canny edge detector was designed to perform in noisy data and was expected to yield accurate results what was indeed the case. Importantly, our new detector performed even better in the noisy data, yielding consistently lower errors than the Canny edge detector. In most experiments the statistical comparison revealed statistically significant difference between the two detectors. The incorporated gradient averaging step is primarily responsible for this good performance. Due to the gradient averaging, many of the local inaccuracies caused by noise get eliminated. The mean directional error was never higher than 20° even in data with standard deviation of noise being 100% of the edge magnitude. In this context, note the comparison between the full and the simplified versions of our anisotropic detector. Even though the simplified detector performed comparably well in all other performed experiments, it failed in the noisy data further demonstrating the importance of

the gradient averaging step. In the simplified version, the gradient at the voxel center is computed directly from the interpolated polynomial employing no compensation for the error caused by noise. This justifies the increased design complexity of our full version of the anisotropic 3D edge detector.

VI. Conclusion

A new directional 3D edge detector designed for anisotropic image data sets was reported. The detector is based on interpolating the image intensity function in a small neighborhood of every voxel by a tri-cubic polynomial. The analytical approximation of the image intensity function is used to compute the intensity function gradients. The developed edge detector uses a maximum average of directional derivatives of the approximated image intensity function over a small neighborhood to determine the gradient direction thus achieving robust edge detection in the presence of noise.

Quantitative measures of the 3D edge detection accuracy were employed to compare the performance of our new edge detector to that of the 3D Canny edge detector. 3D edges with step and ramp profiles with varying surface curvatures at the edge point as well as several levels of noise were used for the performance testing. The reported edge detector statistically significantly outperformed the Canny edge detector in most experiments in anisotropic data as well as in data with superimposed noise. Another important property of the new edge detector is the ease of its implementation. Although its design required complex steps, the implementation employs straightforward 3D convolution in the volumetric image data using three pre-computed directional masks.

Appendix

I. Tri-cubic polynomial fit

The constants of the tri-cubic polynomial are computed to minimize the error function $e(x_{im}, y_{im}, z_{im})$ from Eq. 5. Partial derivatives of $e(x, y, z)$ with respect to the polynomial constants $K_1, \ldots, K_{20}$ are

$$\begin{aligned}\left[\frac{\partial e}{\partial K_1}, \ldots, \frac{\partial e}{\partial K_{20}}\right]^T &= 2 \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} \{[P(x,y,z) - f(x,y,z)] \\ &\quad \left[1, x, y, z, (x^2 + d_x^2/3), xy, xz, yz, (y^2 + d_y^2/3), (z^2 + d_z^2/3),\right. \\ &\quad (x^3 + d_x^2 x), y(x^2 + d_x^2/3), z(x^2 + d_x^2/3), x(y^2 + d_y^2/3), x(z^2 + d_z^2/3), \\ &\quad \left. z(y^2 + d_y^2/3), y(z^2 + d_z^2/3), (y^3 + d_y^2 y), (z^3 + d_z^2 z), xyz\right]^T\}\end{aligned} \tag{18}$$

where X, Y, Z are neighborhood sets in the continuous space directly corresponding to X_{im}, Y_{im} and Z_{im} in the discrete space. When considering a symmetric neighborhood, a significant number of cancellations will occur:

$$
\begin{bmatrix}
\frac{\partial e}{\partial K_1} \\ \frac{\partial e}{\partial K_2} \\ \frac{\partial e}{\partial K_3} \\ \frac{\partial e}{\partial K_4} \\ \frac{\partial e}{\partial K_5} \\ \frac{\partial e}{\partial K_6} \\ \frac{\partial e}{\partial K_7} \\ \frac{\partial e}{\partial K_8} \\ \frac{\partial e}{\partial K_9} \\ \frac{\partial e}{\partial K_{10}} \\ \frac{\partial e}{\partial K_{11}} \\ \frac{\partial e}{\partial K_{12}} \\ \frac{\partial e}{\partial K_{13}} \\ \frac{\partial e}{\partial K_{14}} \\ \frac{\partial e}{\partial K_{15}} \\ \frac{\partial e}{\partial K_{16}} \\ \frac{\partial e}{\partial K_{17}} \\ \frac{\partial e}{\partial K_{18}} \\ \frac{\partial e}{\partial K_{19}} \\ \frac{\partial e}{\partial K_{20}}
\end{bmatrix}
= 2 \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z}
\begin{bmatrix}
K_1 + K_5(x^2 + d_x^2/3) + K_9(y^2 + d_y^2/3) + K_{10}(z^2 + d_z^2/3) - f(x,y,z) \\
K_2 x^2 + K_{11} x^2 (x^2 + d_x^2) + K_{14} x^2 (y^2 + d_y^2/3) + K_{15} x^2 (z^2 + d_z^2/3) - x f(x,y,z) \\
K_3 y^2 + K_{12} y^2 (x^2 + d_x^2/3) + K_{17} y^2 (z^2 + d_z^2/3) + K_{18} y^2 (y^2 + d_y^2) - y f(x,y,z) \\
K_4 z^2 + K_{13} z^2 (x^2 + d_x^2/3) + K_{16} z^2 (y^2 + d_y^2/3) + K_{19} z^2 (z^2 + d_z^2) - y f(x,y,z) \\
K_1 (x^2 + d_x^2/3) + K_5 (x^2 + d_x^2/3)^2 + K_9 (x^2 + d_x^2/3)(y^2 + d_y^2/3) + \\
+ K_{10} (x^2 + d_x^2/3)(z^2 + d_z^2/3) - (x^2 + d_x^2/3) f(x,y,z) \\
K_6 x^2 y^2 - xy f(x,y,z) \\
K_7 x^2 z^2 - xz f(x,y,z) \\
K_8 y^2 z^2 - yz f(x,y,z) \\
K_1 (y^2 + d_y^2/3) + K_5 (y^2 + d_y^2/3)(x^2 + d_x^2/3) + K_9 (y^2 + d_y^2/3)^2 + \\
+ K_{10} (y^2 + d_y^2/3)(z^2 + d_z^2/3) - (y^2 + d_y^2/3) f(x,y,z) \\
K_1 (z^2 + d_z^2/3) + K_5 (z^2 + d_z^2/3)(x^2 + d_x^2/3) + K_9 (z^2 + d_z^2/3)(y^2 + d_y^2/3) + \\
+ K_{10} (z^2 + d_z^2/3)^2 - (z^2 + d_z^2/3) f(x,y,z) \\
K_2 x^2 (x^2 + d_x^2) + K_{11} x^2 (x^2 + d_x^2)^2 + K_{14} x^2 (x^2 + d_x^2)(y^2 + d_y^2/3) + \\
+ K_{15} x^2 (x^2 + d_x^2)(z^2 + d_z^2/3) - x (x^2 + d_x^2) f(x,y,z) \\
K_3 y^2 (x^2 + d_x^2/3) + K_{12} y^2 (x^2 + d_x^2/3)^2 + K_{17} y^2 (x^2 + d_x^2/3)(z^2 + d_z^2/3) + \\
+ K_{18} y^2 (x^2 + d_x^2/3)(y^2 + d_y^2) - y (x^2 + d_x^2/3) f(x,y,z) \\
K_4 z^2 (x^2 + d_x^2/3) + K_{13} z^2 (x^2 + d_x^2/3)^2 + K_{16} z^2 (x^2 + d_x^2/3)(y^2 + d_y^2/3) + \\
+ K_{19} z^2 (x^2 + d_x^2/3)(z^2 + d_z^2) - z (x^2 + d_x^2/3) f(x,y,z) \\
K_2 x^2 (y^2 + d_y^2/3) + K_{11} x^2 (x^2 + d_x^2) + (y^2 + d_y^2/3) K_{14} x^2 (y^2 + d_y^2/3)^2 + \\
+ K_{15} x^2 (y^2 + d_y^2/3)(z^2 + d_z^2/3) - x (y^2 + d_y^2/3) f(x,y,z) \\
K_2 x^2 (z^2 + d_z^2/3) + K_{11} x^2 (x^2 + d_x^2)(z^2 + d_z^2/3) + \\
K_{14} x^2 (y^2 + d_y^2/3)(z^2 + d_z^2/3) + K_{15} x^2 (z^2 + d_z^2/3)^2 - x (z^2 + d_z^2/3) f(x,y,z) \\
K_4 z^2 (y^2 + d_y^2/3) + K_{13} z^2 (x^2 + d_x^2/3)(y^2 + d_y^2/3) + K_{16} z^2 (y^2 + d_y^2/3)^2 + \\
+ K_{19} z^2 (y^2 + d_y^2/3)(z^2 + d_z^2) - z (y^2 + d_y^2/3) f(x,y,z) \\
K_3 y^2 (z^2 + d_z^2/3) + K_{12} y^2 (x^2 + d_x^2/3)(z^2 + d_z^2/3) + K_{17} y^2 (z^2 + d_z^2/3)^2 + \\
+ K_{18} y^2 (z^2 + d_z^2/3)(y^2 + d_y^2) - y (z^2 + d_z^2/3) f(x,y,z) \\
K_3 y^2 (y^2 + d_y^2) + K_{12} y^2 (x^2 + d_x^2/3)(y^2 + d_y^2) + K_{17} y^2 (y^2 + d_y^2)(z^2 + d_z^2/3) + \\
+ K_{18} y^2 (y^2 + d_y^2)^2 - y (y^2 + d_y^2) f(x,y,z) \\
K_4 z^2 (z^2 + d_z^2) + K_{13} z^2 (x^2 + d_x^2/3)(z^2 + d_z^2) + K_{16} z^2 (y^2 + d_y^2/3)(z^2 + d_z^2) + \\
+ K_{19} z^2 (z^2 + d_z^2)^2 - z (z^2 + d_z^2) f(x,y,z) \\
K_{20} x^2 y^2 z^2 - xyz f(x,y,z)
\end{bmatrix}
\tag{19}
$$

After setting the derivatives equal to zero, coefficients K_6, K_7, K_8 and K_{20} can be computed directly. For the remaining coefficients, four systems of equations with four unknown variables can be constructed. The

substitution from Eq. 20 yields a simplified Eq. 21.

$$
\begin{array}{lll}
X_0 = \sum_{x\in X} 1 & Y_0 = \sum_{y\in Y} 1 & Z_0 = \sum_{z\in Z} 1 \\
X_2 = \sum_{x\in X} x^2 & Y_2 = \sum_{y\in Y} y^2 & Z_2 = \sum_{z\in Z} z^2 \\
X_{31} = \sum_{x\in X} (x^2 + d_x^2/3) & Y_{31} = \sum_{y\in Y} (y^2 + d_y^2/3) & Z_{31} = \sum_{z\in Z} (z^2 + d_z^2/3) \\
X_{32} = \sum_{x\in X} (x^2 + d_x^2/3)^2 & Y_{32} = \sum_{y\in Y} (y^2 + d_y^2/3)^2 & Z_{32} = \sum_{z\in Z} (z^2 + d_z^2/3)^2 \\
X_{11} = \sum_{x\in X} (x^2 + d_x^2) & Y_{11} = \sum_{y\in Y} (y^2 + d_y^2) & Z_{11} = \sum_{z\in Z} (z^2 + d_z^2) \\
X_{12} = \sum_{x\in X} (x^2 + d_x^2)^2 & Y_{12} = \sum_{y\in Y} (y^2 + d_y^2)^2 & Z_{12} = \sum_{z\in Z} (z^2 + d_z^2)^2 \\
X_{21} = \sum_{x\in X} x^2(x^2 + d_x^2) & Y_{21} = \sum_{y\in Y} y^2(y^2 + d_y^2) & Z_{21} = \sum_{z\in Z} z^2(z^2 + d_z^2) \\
X_{22} = \sum_{x\in X} x^2(x^2 + d_x^2)^2 & Y_{22} = \sum_{y\in Y} y^2(y^2 + d_y^2)^2 & Z_{22} = \sum_{z\in Z} z^2(z^2 + d_z^2)^2
\end{array}
\tag{20}
$$

The four simplified systems are

$$
\begin{bmatrix} X_0Y_0Z_0 & X_{31}Y_0Z_0 & X_0Y_{31}Z_0 & X_0Y_0Z_{31} \\ X_{31}Y_0Z_0 & X_{32}Y_0Z_0 & X_{31}Y_{31}Z_0 & X_{31}Y_0Z_{31} \\ X_{31}Y_0Z_0 & X_{31}Y_{31}Z_0 & X_0Y_{32}Z_0 & X_0Y_{31}Z_{31} \\ X_0Y_0Z_{31} & X_{31}Y_0Z_{31} & X_0Y_{31}Z_{31} & X_0Y_0Z_{32} \end{bmatrix} \begin{bmatrix} K_1 \\ K_5 \\ K_9 \\ K_{10} \end{bmatrix} = \begin{bmatrix} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} (x^2 + d_x^2/3) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} (y^2 + d_y^2/3) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} (z^2 + d_z^2/3) f(x,y,z) \end{bmatrix}
$$

$$
\begin{bmatrix} X_2Y_0Z_0 & X_{21}Y_0Z_0 & X_2Y_{31}Z_0 & X_2Y_0Z_{31} \\ X_{21}Y_0Z_0 & X_{22}Y_0Z_0 & X_{21}Y_{31}Z_0 & X_{21}Y_0Z_{31} \\ X_2Y_{31}Z_0 & X_{21}Y_{31}Z_0 & X_2Y_{32}Z_0 & X_2Y_{31}Z_{31} \\ X_2Y_0Z_{31} & X_{21}Y_0Z_{31} & X_2Y_{31}Z_{31} & X_2Y_0Z_{32} \end{bmatrix} \begin{bmatrix} K_2 \\ K_{11} \\ K_{14} \\ K_{15} \end{bmatrix} = \begin{bmatrix} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} x f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} x(x^2 + d_x^2) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} x(y^2 + d_y^2/3) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} x(z^2 + d_z^2/3) f(x,y,z) \end{bmatrix}
\tag{21}
$$

$$
\begin{bmatrix} X_0Y_2Z_0 & X_{31}Y_2Z_0 & X_0Y_2Z_{31} & X_0Y_{21}Z_0 \\ X_{31}Y_2Z_0 & X_{32}Y_2Z_0 & X_{31}Y_2Z_{31} & X_{31}Y_{21}Z_0 \\ X_0Y_2Z_{31} & X_{31}Y_2Z_{31} & X_0Y_2Z_{32} & X_0Y_{21}Z_{31} \\ X_0Y_{21}Z_0 & X_{31}Y_{21}Z_0 & X_0Y_{21}Z_{31} & X_0Y_{22}Z_0 \end{bmatrix} \begin{bmatrix} K_3 \\ K_{12} \\ K_{17} \\ K_{18} \end{bmatrix} = \begin{bmatrix} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} y f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} y(x^2 + d_x^2/3) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} y(z^2 + d_z^2/3) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} y(y^2 + d_y^2) f(x,y,z) \end{bmatrix}
$$

$$
\begin{bmatrix} X_0Y_0Z_2 & X_{31}Y_0Z_2 & X_0Y_{31}Z_2 & X_0Y_0Z_{21} \\ X_{31}Y_0Z_2 & X_{32}Y_0Z_2 & X_{31}Y_{31}Z_2 & X_{31}Y_0Z_{21} \\ X_{31}Y_0Z_2 & X_{31}Y_{31}Z_2 & X_0Y_{32}Z_2 & X_0Y_{31}Z_{21} \\ X_0Y_0Z_{21} & X_{31}Y_0Z_{21} & X_0Y_{31}Z_{21} & X_0Y_0Z_{22} \end{bmatrix} \begin{bmatrix} K_4 \\ K_{13} \\ K_{16} \\ K_{19} \end{bmatrix} = \begin{bmatrix} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} z f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} z(x^2 + d_x^2/3) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} z(y^2 + d_y^2/3) f(x,y,z) \\ \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} z(z^2 + d_z^2) f(x,y,z) \end{bmatrix}
$$

The systems of linear equations can be solved to receive the values of coefficients minimizing the error fit function $e(x,y,z)$. To further simplify the equations, the following substitutions are used:

$$
\begin{aligned}
U_1 &= Y_0 Z_0 (X_0 X_{32} - X_{31}^2) \\
V_1 &= X_0 Z_0 (Y_0 Y_{32} - Y_{31}^2) \\
W_1 &= X_0 Y_0 (Z_0 Z_{32} - Z_{31}^2) \\
A_1 &= \frac{1}{X_0 Y_0 Z_0} \left[1 - \frac{X_{31}^2}{X_{31}^2 - X_0 X_{32}} - \frac{Y_{31}^2}{Y_{31}^2 - Y_0 Y_{32}} - \frac{Z_{31}^2}{Z_{31}^2 - Z_0 Z_{32}} \right] \\
U_2 &= Y_0 Z_0 (X_2 X_{22} - X_{21}^2) \\
V_2 &= X_2 Z_0 (Y_0 Y_{32} - Y_{31}^2) \\
W_2 &= X_2 Y_0 (Z_0 Z_{32} - Z_{31}^2) \\
A_2 &= \frac{1}{X_2 Y_0 Z_0} \left[1 - \frac{X_{21}^2}{X_{21}^2 - X_2 X_{22}} - \frac{Y_{31}^2}{Y_{31}^2 - Y_0 Y_{32}} - \frac{Z_{31}^2}{Z_{31}^2 - Z_0 Z_{32}} \right] \\
U_3 &= Y_2 Z_0 (X_0 X_{32} - X_{31}^2) \\
V_3 &= X_0 Y_2 (Z_0 Z_{32} - Z_{31}^2) \\
W_3 &= X_0 Z_0 (Y_2 Y_{22} - Y_{21}^2) \\
A_3 &= \frac{1}{X_0 Y_2 Z_0} \left[1 - \frac{X_{31}^2}{X_{31}^2 - X_0 X_{32}} - \frac{Y_{21}^2}{Y_{21}^2 - Y_2 Y_{22}} - \frac{Z_{31}^2}{Z_{31}^2 - Z_0 Z_{32}} \right] \\
U_4 &= Y_0 Z_2 (X_0 X_{32} - X_{31}^2) \\
V_4 &= X_0 Z_2 (Y_0 Y_{32} - Y_{31}^2) \\
W_4 &= X_0 Y_0 (Z_2 Z_{22} - Z_{21}^2) \\
A_4 &= \frac{1}{Y_0 X_0 Z_2} \left[1 - \frac{X_{31}^2}{X_{31}^2 - X_0 X_{32}} - \frac{Y_{31}^2}{Y_{31}^2 - Y_0 Y_{32}} - \frac{Z_{21}^2}{Z_{21}^2 - Z_2 Z_{22}} \right]
\end{aligned}
\tag{22}
$$

The final solution of the least squares error fit of the tri-cubic polynomial is then

$$
\begin{aligned}
K_1 &= \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} \left(A_1 - \frac{1}{U_1} X_{31}(x^2 + \frac{d_x^2}{3}) - \frac{1}{V_1} Y_{31}(y^2 + \frac{d_y^2}{3}) - \frac{1}{W_1} Z_{31}(z^2 + \frac{d_z^2}{3}) \right) f(x,y,z) \\
K_2 &= \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} \left(A_2 - \frac{1}{U_2} X_{21}(x^2 + d_x^2) - \frac{1}{V_2} Y_{31}(y^2 + \frac{d_y^2}{3}) - \frac{1}{W_2} Z_{31}(z^2 + \frac{d_z^2}{3}) \right) x f(x,y,z) \\
K_3 &= \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} \left(A_3 - \frac{1}{U_3} X_{31}(x^2 + \frac{d_x^2}{3}) - \frac{1}{W_3} Y_{21}(y^2 + d_y^2) - \frac{1}{V_3} Z_{31}(z^2 + \frac{d_z^2}{3}) \right) y f(x,y,z) \\
K_4 &= \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} \left(A_4 - \frac{1}{U_4} X_{31}(x^2 + \frac{d_x^2}{3}) - \frac{1}{V_4} Y_{31}(y^2 + \frac{d_y^2}{3}) - \frac{1}{W_4} Z_{21}(z^2 + d_z^2) \right) z f(x,y,z) \\
K_5 &= \frac{1}{U_1} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} \left((x^2 + \frac{d_x^2}{3}) X_0 - X_{31} \right) f(x,y,z) \\
K_6 &= \frac{1}{X_2 Y_2 Z_0} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} xy f(x,y,z) \\
K_7 &= \frac{1}{X_2 Y_0 Z_2} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} xz f(x,y,z) \\
K_8 &= \frac{1}{X_0 Y_2 Z_2} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} yz f(x,y,z) \\
K_9 &= \frac{1}{V_1} \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} \left((y^2 + \frac{d_y^2}{3}) Y_0 - Y_{31} \right) f(x,y,z)
\end{aligned}
$$

$$
\begin{aligned}
K_{10} &= \frac{1}{W_1}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((z^2+\frac{d_z^2}{3})Z_0 - Z_{31}\right) f(x,y,z) \qquad (23)\\
K_{11} &= \frac{1}{U_2}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((x^2+d_x^2)X_2 - X_{21}\right) xf(x,y,z)\\
K_{12} &= \frac{1}{U_3}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((x^2+\frac{d_x^2}{3})X_0 - X_{31}\right) yf(x,y,z)\\
K_{13} &= \frac{1}{U_4}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((x^2+\frac{d_x^2}{3})X_0 - X_{31}\right) zf(x,y,z)\\
K_{14} &= \frac{1}{V_2}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((y^2+\frac{d_y^2}{3})Y_0 - Y_{31}\right) xf(x,y,z)\\
K_{15} &= \frac{1}{W_2}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((z^2+\frac{d_z^2}{3})Z_0 - Z_{31}\right) xf(x,y,z)\\
K_{16} &= \frac{1}{V_4}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((y^2+\frac{d_y^2}{3})Y_0 - Y_{31}\right) zf(x,y,z)\\
K_{17} &= \frac{1}{V_3}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((z^2+\frac{d_z^2}{3})Z_0 - Z_{31}\right) yf(x,y,z)\\
K_{18} &= \frac{1}{W_3}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((y^2+d_y^2)Y_2 - Y_{21}\right) yf(x,y,z)\\
K_{19} &= \frac{1}{W_4}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z}\left((z^2+d_z^2)Z_2 - Z_{21}\right) zf(x,y,z)\\
K_{20} &= \frac{1}{X_2Y_2Z_2}\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} xyz\ f(x,y,z)
\end{aligned}
$$

II. Directional Gradient Averaging

A. Rotation transformation matrices

The transformation matrix that represents a θ rotation about the z axis followed by a ϕ rotation about the y axis in the local coordinate system is

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi & -\sin\theta & -\cos\theta\sin\phi \\ \sin\theta\cos\phi & \cos\theta & -\sin\theta\sin\phi \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \qquad (24)$$

Therefore, a unit vector (n_x, n_y, n_z) in the direction (θ, ϕ) from Eq. 7 can be computed as the rotation of a unit vector in the direction of the x axis by the angle (θ, ϕ):

$$(n_x, n_y, n_z)^T = R \cdot (1,0,0)^T = (\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi)^T \qquad (25)$$

To express the coordinate transformation in Eq. 10, the original x, y, z coordinates must be derived from the rotated coordinates $\bar{x}, \bar{y}, \bar{z}$. Therefore, any point in the space after rotation described by the matrix R can be expressed in the original coordinates as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \cdot \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{pmatrix} \qquad (26)$$

As a result of this transformation, the coordinates can be expressed as

$$(x,y,z) = (\bar{x}\cos\theta\cos\phi - \bar{y}\sin\theta - \bar{z}\cos\theta\sin\phi, \bar{x}\sin\theta\cos\phi + \bar{y}\cos\theta - \bar{z}\sin\theta\cos\phi, \bar{x}\sin\phi + \bar{z}\cos\phi) \qquad (27)$$

B. Functions A,B,C and D in the Integral of Eq. 11

$$
\begin{aligned}
A(K_i,\theta,\phi) &= K_{11}\cos^3\theta\cos^3\phi + K_{12}\cos^2\theta\sin\theta\cos^3\phi + K_{13}\cos^2\theta\cos^2\phi\sin\phi + \\
&\quad K_{14}\cos\theta\sin^2\theta\cos^3\phi + K_{15}\cos\theta\cos\phi\sin^2\phi + K_{16}\sin^2\theta\cos^2\phi\sin\phi + \\
&\quad K_{17}\sin\theta\cos\phi\sin^2\phi + K_{18}\sin^3\theta\cos^3\phi + K_{19}\sin^3\phi + K_{20}\cos\theta\sin\theta\cos^2\phi\sin\phi \\
B(K_i,\theta,\phi) &= K_{11}\cos\theta\sin^2\theta\cos\phi + K_{12}\frac{1}{12}\cos\phi(\sin\theta - 3\sin(3\theta)) + K_{13}\frac{1}{3}\sin^2\theta\sin\phi + \\
&\quad K_{14}\frac{1}{12}\cos\phi(\cos\theta + 3\cos(3\theta)) + K_{16}\frac{1}{3}\cos^2\theta\sin\phi + K_{18}\cos^2\theta\sin\theta\cos\phi - \\
&\quad K_{20}\frac{1}{3}\cos\theta\sin\theta\sin\phi \qquad (28) \\
C(K_i,\theta,\phi) &= K_{11}\cos^3\theta\cos\phi\sin^2\phi + K_{12}\cos^2\theta\sin\theta\cos\phi\sin^2\phi + K_{13}\frac{1}{12}\cos^2\theta(\sin\phi - 3\sin(3\phi)) + \\
&\quad K_{14}\cos\theta\sin^2\theta\cos\phi\sin^2\phi + K_{15}\frac{1}{12}\cos\theta(\cos\phi + 3\cos(3\phi)) + \\
&\quad K_{16}\frac{1}{12}\sin^2\theta(\sin\phi - 3\sin(3\phi)) + K_{17}\frac{1}{12}\sin\theta(\cos\phi + 3\cos(3\phi)) + K_{18}\sin^3\theta\cos\phi\sin^2\phi + \\
&\quad K_{19}\cos^2\phi\sin\phi + K_{20}\frac{1}{12}\cos\theta\sin\theta(\sin\phi - 3\sin(3\phi)) \\
D(K_i,\theta,\phi) &= K_2\cos\phi\cos\theta + K_4\sin\phi + K_3\cos\phi\sin\theta
\end{aligned}
$$

References

[1] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision.* Boston: PWS, 2nd ed., 1998.

[2] Y. J. Zhang, "Quantitative study of 3D gradient operators," *Image and Vision Computing*, vol. 11, pp. 611–622, 1993.

[3] P. Bhattacharya and D. Wild, "A new edge detector for gray volumetric data," *Comput. Biol. Med.*, vol. 26, pp. 315–328, 1996.

[4] R. Mehrotra and S. Zhan, "A zero-crossing-based optimal three-dimensional edge detector," *CVGIP Image Understanding*, vol. 59, pp. 242–253, 1994.

[5] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[6] L. Spacek, "Edge detection and motion detection," *Image and Vision Computing*, pp. 43–52, 1986.

[7] W. K. Pratt, *Digital Image Processing.* New York: John Wiley and Sons, 2nd ed., 1991.

[8] R. M. Haralick and O. A. Zuniga, "Integrated directional derivative gradient operator," *IEEE Trans. Systems, Man and Cybernetics*, vol. 17, pp. 508–517, 1987.

List of Figures