Fast and Accurate 3D Edge Detection for Surface Reconstruction

Christian Bähnisch, Peer Stelldinger, and Ullrich Köthe

University of Hamburg, 22527 Hamburg, Germany
University of Heidelberg, 69115 Heidelberg, Germany

Abstract. Although edge detection is a well investigated topic, 3D edge detectors mostly lack either accuracy or speed. We will show, how to build a highly accurate subvoxel edge detector, which is fast enough for practical applications. In contrast to other approaches we use a spline interpolation in order to have an efficient approximation of the theoretically ideal sinc interpolator. We give theoretical bounds for the accuracy and show experimentally that our approach reaches these bounds while the often-used subpixel-accurate parabola fit leads to much higher edge displacements.

1 Introduction

Edge detection is generally seen as an important part of image analysis and computer vision. As a fundamental step in early vision it provides the basis for subsequent high level processing such as object recognition and image segmentation. Depending on the concrete analysis task the accurate detection of edges can be very important. For example, the estimation of geometric and differential properties of reconstructed object boundaries such as perimeter, volume, curvature or even higher order properties requires particularly accurate edge localization algorithms.

However, especially for the 3D image domain performance and storage considerations become very present. While edge detection in 3D is essentially the same as in 2D, the trade-off of computational efficiency and geometric accuracy makes the design of usable 3D edge detectors very difficult. In this work we propose a 3D edge detection algorithm which provides accurate edges with subvoxel precision while being computationally efficient.

The paper is organized as follows: First, we give an overview about previous work. Then we describe our new approach for edge detection, followed by a theoretical analysis of the edge location errors of an ideal subvoxel edge detector under the influence of noise. This analysis is based on the same optimality constraints as used in the Canny edge detector. Finally, we show experimentally that (in contrast to a parabola fit method) our algorithm is a good approximation of an ideal edge detector, and that the computational costs are negligible.

J. Denzler, G. Notni, and H. Süße (Eds.): DAGM 2009, LNCS 5748, pp. 111–120, 2009.

2 Previous Work

Since edge detection is a fundamental operation in image analysis, there exists an uncountable number of different approaches. Nevertheless, any new edge detection algorithm must compete with the groundbreaking algorithm proposed by Canny in his MS thesis in 1986 [2]. Due to his definition, an edge is detected as local maximum of the gradient magnitude along the gradient direction. This idea proved to be advantageous to other approaches, mostly because of its theoretical justification and the restriction to first derivatives, which makes it more robust against noise. The Canny edge detection algorithm has been extended to 3D in [10] by using recursive filters. However both methods return only edge points with pixel/voxel accuracy, i.e. certain pixels (respectively voxels in 3D) are marked as edge points.

Since generally the discrete image is a sampled version of a continuous image of the real world, attempts had been made to locate the edge points with higher accuracy. The edge points are then called *edgels* (i.e. *edge elements* in analogy to pixels as picture elements and *voxels* as volume elements). Since a 2D edge separates two regions from each other, the analogon in 3D is a surface. Thus 3D edge points are also called *surfels*.

One often cited example of a subpixel precise edge detection algorithm is given in [3], where 2D edgels are detected as maxima on a local parabola fit to the neighboring pixels. The disadvantage of this parabola fit approach is, that the different local fits do not stitch together to a continuous image. The same is true for the approaches presented in [12]. A different method for subvoxel precise edge detection based on local moment analysis is given in [8], but it simply oversamples the moment functions and thus there is still a discretization error being only on a finer grid.

An interpolation approach having higher accuracy has been proposed for 2D in [7,13,14]. Here, the continuous image is defined by a computationally efficient spline interpolation based on the discrete samples. With increasing order of the spline, this approximates the signal theoretic optimal sinc interpolator, thus in case of sufficiently bandlimited images the approximation error converges to zero. An efficient implementation of the spline interpolation can be found in the VIGRA image processing library [6].

3 The 3D Edge Detection Algorithm

In this section we first introduce the concepts and mathematical notions needed to give the term "3D edge" an exact meaning. A discussion of our algorithm to actually detect them follows.

3.1 Volume Function and 3D Edge Model

In the following our mathematical model for a 3D image is the scalar valued *volume function* with *shape* $(w, h, d)^\mathsf{T} \in \mathbb{N}^3$:

$$f : \underline{w} \times \underline{h} \times \underline{d} \to D \quad \text{with}$$

$\underline{n} = \{0, \ldots, n-1\}$ and an appropriately selected domain D, e.g. $D = \underline{255}$. The gradient of f at position $\boldsymbol{p}$ is defined as $\nabla f := \nabla g_\sigma \star f$ with ∇g_σ denoting the vectors of spatial derivatives of the Gaussian $g_\sigma(\boldsymbol{p}) := {}^1\!/\!\sqrt{2\pi}\sigma \exp(-\frac{\|\boldsymbol{p}\|^2}{2\sigma^2})$ at scale σ. Note that the gradient can be efficiently computed using the separability property of the Gaussian. The gradient of the volume function is the basis for our 3D edge model. The *boundary indicator* $b := \|\nabla f\|$ expresses strong evidence for a boundary at a certain volume position in terms of a high scalar value. Adapting Canny's edge model [2] to 3D images, we define *surface elements (surfels)* as maxima of the boundary indicator b along the gradient direction ∇f of the 3D image function.

Our detection algorithm for these maxima can be divided into two phases: A voxel-precise surfel detection phase and a refinement phase which improves the localization of the surfels to sub-voxel precision.

3.2 Phase 1: Voxel-Based Edge-Detection

The detection of the voxel-precise surfels is basically an adaption of the classical Canny edge detection algorithm [2]. However, we do not perform the commonly used edge tracing step with hysteresis thresholding as this step becomes especially complicated in the 3D image domain. A second reason is, that this is the most time consuming part of the Canny edge detection algorithm. Additionally, although non-maxima-suppression is performed, the classical Canny edge detection algorithm can lead to several pixel wide edges which is only alleviated by hysteresis. Therefore, we propose to use only one threshold (corresponding to the lower of the hysteresis thresholds) in combination with a fast morphological thinning with priorities which ensures one voxel thick surfaces while preserving the topology of the detected voxel-set. This also reduces the number of initial edges for subvoxel refinement resulting in a significant speedup of the further algorithm. Thus, the steps of the first phase are:

1. Compute the gradient ∇f and the boundary indicator function b.
2. Compute a binary volume function $a : \underline{w} \times \underline{h} \times \underline{d} \to \{0, 1\}$ marking surfels with "1" and background voxels with "0" by thresholding b with t and by using non maximum suppression along the gradient direction:
$$a(\boldsymbol{p}) := \{1 \text{ if } b(\boldsymbol{p}) > t \wedge b(\boldsymbol{p}) > b(\boldsymbol{p} \pm \boldsymbol{d}), \text{ else } 0\}$$
with $\boldsymbol{d}$ being a vector such that $\boldsymbol{p} + \boldsymbol{d}$ is the grid point of the nearest neighbouring voxel in direction of ∇f, i.e.,
$$\boldsymbol{d} = \left(\left\lfloor u + \tfrac{1}{2} \right\rfloor, \left\lfloor v + \tfrac{1}{2} \right\rfloor, \left\lfloor w + \tfrac{1}{2} \right\rfloor\right)^\mathsf{T} \text{ and } (u, v, w)^\mathsf{T} = \frac{1}{2\sin(\pi/8)} \frac{\nabla f(\boldsymbol{p})}{\|\nabla f(\boldsymbol{p})\|}$$
where $\lfloor \cdot \rfloor$ is the floor operation.
3. Do topology preserving thinning with priorities on a.

For the last step of our algorithm we use a modified version of the 3D morphological thinning algorithm described in [5]: Thinning is not performed in scan line order of the volume; instead surfels in a with a small boundary indicator value are preferred for removal. The outcome of the thinning step is then a one voxel thin set of surfels such that its number of connected components is unchanged. A detailed description of thinning with priorities can be found in [7].

3.3 Phase 2: Subpixel Refinement

The first phase of our algorithm yields surface points located at voxel grid positions. In the second phase the localization accuracy of these points is improved to sub-voxel precision by means of iterative optimization methods based on line searches.

Therefore a continuous version of the boundary indicator is needed, i.e. it has to be evaluable at arbitrary sub-voxel positions. Here we use B-spline based interpolators for this purpose. They are especially suited as they provide an optimal trad-off between interpolation quality and computational costs: With increasing order n spline interpolators converge to the ideal sinc interpolator and have already for small values of n a very good approximation quality. While the computational burden also grows with n their are still efficiently implementable for order $n = 5$ used in our experiments.

The continuous boundary indicator can now be defined via a discrete convolution of recursively defined B-spline basis function β_n of *order n*:

$$b(\boldsymbol{p}) := \sum_{i,j,k} c_{ijk} \beta_n(i-x)\beta_n(j-y)\beta_n(k-z)$$

with $\beta_n := \frac{1}{n}\left[\beta_{n-1}\left(\frac{n+1}{2}+x\right)\left(x+\frac{1}{2}\right)+\beta_{n-1}\left(\frac{n+1}{2}-x\right)\left(x-\frac{1}{2}\right)\right]$and $\beta_0 := \{0 \text{ if } x < 0, \text{ else } 1\}$. The coefficients c_{ijk} can be efficiently computed from the discrete version of the boundary indicator by recursive linear filters. Note that there is one coefficient for each voxel and that they have to be computed only once for each volume. The overall algorithmic complexity for this is linear in the number of voxels with a small constant factor. More details on the corresponding theory and the actual implementation can be found in [7,13,14].

A B-spline interpolated boundary indicator has also the advantage of being $(n-1)$-times continuously differentiable. Its derivatives can also be efficiently computed at arbitrary sub-voxel positions which is very important for optimization methods which rely on gradient information.

We can now work on the continuous boundary indicator to get sub-voxel accurate surfels. As we are adapting Canny's edge model to 3D images, we shift the already detected surfels along the gradient direction of the 3D image function such that they are located at maxima of the boundary indicator. This can be formulated as a constrained line search optimization problem, i.e. we search for the maximizing parameter α of the one dimensional function

$$\phi(\alpha) := b(\boldsymbol{p} + \alpha \cdot \boldsymbol{d})$$

with the constraint $\alpha \in (\alpha_{\min}, \alpha_{\max})$ and with $\boldsymbol{d}$ being an unit length vector at position $\boldsymbol{p}$ collinear with ∇f such that b increases in its direction, i.e.

$$\boldsymbol{d} := \operatorname{sgn}(\nabla f^{\mathsf{T}} \nabla b) \cdot \nabla f / \|\nabla f\| .$$

The interval constraint on α can be fixed for every surfel or computed dynamically with e.g. bracketing (see e.g. [15]) which we use here. Maximizing ϕ can then be done via standard line search algorithms like the algorithm of Brent [1] or the algorithm of Moré and Thuente [11]. Here we use the modification of Brent's algorithm presented in [15], which takes advantage of the available gradient information.

In order to get even higher accuracy, the line searches defined by ϕ can be iterated several times. Any line search based optimization algorithms should be suitable for this. Here, we choose the common conjugate gradient method (see e.g. [15]) and compare its accuracy improvement to the single line search approach in sec. 5.

4 Theoretical Analysis: Accuracy

In order to justify the localization accuracy of our algorithm we perform experiments on synthetic test volumes based on simple 3D edge models for which the true surfel positions are known from theory. For this it is necessary to carefully model and implement the corresponding image acquisition process which we define as a convolution of a continuous volume with a 3D isotropic Gaussian point spread function (PSF) with scale σ_{PSF} followed by sampling and quantization with possible addition of white Gaussian noise. The volume is modeled via a binary volume function $f_0 : \mathbb{R}^3 \to \{0, S\}$ with $S \in \mathbb{R}$ such that the support $f_0^{-1}(S)$ is either an open half space, ball or a cylinder, i.e. its surface corresponds to a planar, spherical or cylindrical shell. We investigate these three types of functions, since they allow to estimate the localization accuracy for every possible case of a 3D surface. For example, if a surface is hyperbolic in some point $\boldsymbol{p}$ with principal curvatures $\kappa_1 > 0, \kappa_2 < 0$, then the localization errors should be bounded by the errors of two opposing cylinders having curvatures κ_1, respectively κ_2. The function f_0 is blurred by convolution with a Gaussian before sampling. The resulting function $f = f_0 \star g_{\sigma_{\text{PSF}}}$ defines the ground truth for edge detection.

More precisely, for a planar surface with normal unit vector $\boldsymbol{n}$ and distance $s \in \mathbb{R}$ from the origin the corresponding volume function reads

$$f_{\text{plane}}(\boldsymbol{p}) := S \cdot \varPhi_{\sigma_{\text{PSF}}}(\boldsymbol{p}^\top \boldsymbol{n} + s) \text{ with } \varPhi_\sigma(x) := \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right)\right).$$

Maxima of f_{plane} then occur exactly at positions $\boldsymbol{p}$ with $\boldsymbol{p}^\top \boldsymbol{n} + s = 0$. For a ball $\mathcal{B}_R$ with radius R, a closed form solution of the convolution integral with a Gaussian can be derived by exploiting the rotational symmetry of both functions and the separability of the Gaussian:

$$\begin{aligned} f_{\text{sphere}}(\boldsymbol{p}) &:= \iiint_{\mathcal{B}_R} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{p}\|^2}{2\sigma^2}} d\boldsymbol{x} = \int_{-R}^{R} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r'-r)^2}{2\sigma^2}} dr' \\ &= \frac{S}{2}\left(\operatorname{erf}\left(\frac{R-r}{\sqrt{2}\sigma}\right) + \operatorname{erf}\left(\frac{R+r}{\sqrt{2}\sigma}\right)\right) - \frac{e^{-\frac{(R+r)^2}{2\sigma^2}}\left(e^{\frac{2rR}{\sigma^2}} - 1\right) S\sigma}{\sqrt{2\pi} r} \end{aligned}$$

with $r = \|\boldsymbol{p}\|$. The gradient magnitude of a blurred sphere is then the derivative of f_{sphere} with respect to r:

$$\|\nabla f(\boldsymbol{p})\| = \frac{1}{\sqrt{2\pi} r^2 \sigma} e^{-\frac{(R+r)^2}{2\sigma^2}} \left(\sigma^2 + rR + e^{\frac{2rR}{\sigma^2}}\left(rR - \sigma^2\right)\right) \tag{1}$$

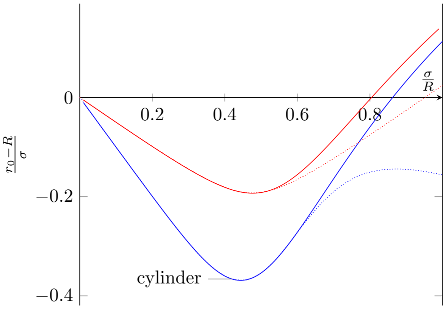


Fig. 1. Normalized bias of a blurred sphere and cylinder with radius R and scaling σ of the PSF. Approximating functions given by (2) and (3) are indicated with dotted lines.

As there is no closed form solution giving the maxima r_0 of (1), fig. 1 shows numeric results. It plots the normalized displacement $(r_0 - R)/\sigma$ against the ratio σ/R, in order to apply to arbitrary scales and radii. In practice, the most interesting part is in the interval $0 \le \sigma/R \le 0.5$, since otherwise the image content is too small to be reliably detectable after blurring. In this interval an approximation with error below $3 \cdot 10^{-4}$ is given by

$$\frac{r_0 - R}{\sigma} = \frac{0.04 - 145.5\frac{\sigma}{R} + 345.8(\frac{\sigma}{R})^2 - 234.8(\frac{\sigma}{R})^3}{142.6 - 308.5\frac{\sigma}{R} + 59.11(\frac{\sigma}{R})^2 + 327.7(\frac{\sigma}{R})^3}. \tag{2}$$

In case of a cylinder of radius R, a closed form solution exists neither for the convolution integral nor for the position of the maxima of its gradient magnitude (but for the gradient magnitude itself). This case is mathematically identical to the 2D case of a disc blurred by a Gaussian, which has been analyzed in detail in [9]. An approximating formula for the relative displacement with error below $3 \cdot 10^{-4}$ is given in [7]:

$$\frac{r_0 - R}{\sigma} = 0.52\sqrt{0.12^2 + \left(\frac{\sigma}{R} - 0.476\right)^2} - 0.255 \tag{3}$$

4.1 Noisy 3D Images

Canny's noise model is based on the assumption that the surface is a step which has been convolved with both the PSF and the edge detection filters. Therefore, the total scale of the smoothed surface is $\sigma_{\text{edge}} = \sqrt{\sigma_{\text{PSF}}^2 + \sigma_{\text{filter}}^2}$. The second directional derivative in the surface's normal direction equals the first derivative of a Gaussian (and it is a constant in the tangential plane of the surface). Near the true surface position, this derivative can be approximated by its first order Taylor expansion:

$$s_{xx}(x) \approx s_{xxx}(x = 0) \cdot x = -\frac{S \cdot x}{\sqrt{2\pi}\sigma_{\text{edge}}^3},$$

where S is the step height. The observed surface profile equals the true profile plus noise. The noise is only filtered with the edge detection filter, not with the PSF. The observed second derivative is the sum of the above Taylor formula and the second derivative of the smoothed noise

$$f_{xx}(x) \approx s_{xxx}|_{x=0} \cdot x + n_{xx}(x).$$

Solving for the standard deviation of x at the zero crossing $f_{xx}(x) = 0$ gives

$$\mathrm{StdDev}[x] = \frac{\sqrt{\mathrm{Var}[n_{xx}]}}{|s_{xxx}(x=0)|}.$$

According to Parseval's theorem, the variance of the second directional derivative of the noise can be computed in the Fourier domain as

$$\mathrm{Var}[n_{xx}] = N^2 \iiint_{-\infty}^{\infty} \left(4\pi^2 u^2 G(u) G(v) G(w)\right)^2 du\, dv\, dw = N^2 \frac{3}{32\pi^{3/2}\sigma^7},$$

where $G(.)$ is the Fourier transform of a Gaussian at scale σ_{filter}, and N^2 is the variance of the noise before filtering. Inserting, we get the expected localization error as

$$\mathrm{StdDev}[x] = \frac{N}{S} \frac{\sqrt{3}(\sigma^2_{\text{PSF}} + \sigma^2_{\text{filter}})^{3/2}}{4\pi^{1/4}\sigma^{7/2}_{\text{filter}}} \tag{4}$$

$\frac{N}{S}$ is the inverse signal-to-noise ratio. In contrast to 2D edge detection this error goes to zero as the filter size approaches infinity

$$\lim_{\sigma_{\text{filter}} \to \infty} \mathrm{StdDev}[x] = \frac{N}{S} \frac{\sqrt{3}}{4\pi^{1/4}\sqrt{\sigma_{\text{filter}}}}.$$

However, this limit only applies to perfectly planar surfaces. In case of curved surfaces, enlarging the edge detection filter leads to a bias, as shown above, and there is a trade-off between noise reduction and bias.

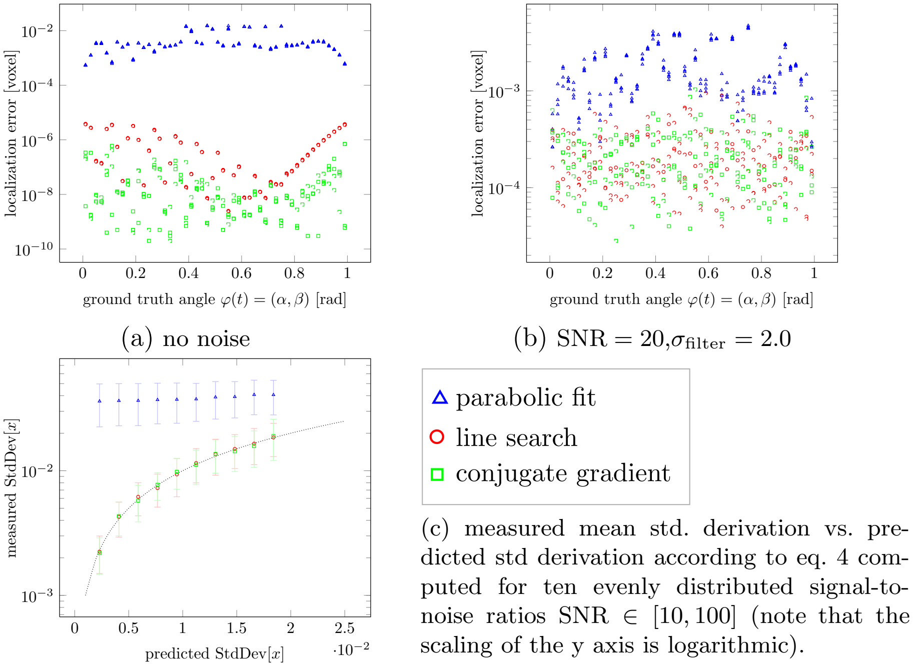
5 Experiments: Accuracy and Speed

In this section we present the results of experiments which confirm our claims about the accuracy and speed characteristics of our algorithm.

We start with artificial volume data generated by sampling of the simple continuous volume functions given above. For the cylindrical cases we used 8-fold oversampled binary discrete volumes with subsequent numeric convolution and downsampling. In the following we always use spline interpolation with order five for the line search and conjugate gradient based algorithms.

Accuracy results for test volumes generated from f_{plane} are shown in fig. 2. As model parameters we used unit step height, $\sigma_{\text{PSF}} = 0.9$ and $\sigma_{\text{filter}} = 1$ and various values for the sub-voxel shift s and the plane normal $\boldsymbol{n}$. For the directions of $\boldsymbol{n}$ we evenly distributed fifty points on the hemisphere located at the origin.

In fig. 2a results for noise free volumes are shown. As one can see both the line search and the conjugate gradient based methods possess very high accuracy and are several orders of magnitudes better than the parabolic fit. In the presence of noise the accuracy of our method is still almost one order of magnitude better than the parabolic fit for a rather bad signal-to-noise ration of SNR = 20, see



(a) no noise

(b) SNR $= 20, \sigma_{\text{filter}} = 2.0$

(c) measured mean std. derivation vs. predicted std derivation according to eq. 4 computed for ten evenly distributed signal-to-noise ratios SNR $\in [10, 100]$ (note that the scaling of the y axis is logarithmic).

Fig. 2. Comparison of sub-voxel accuracy of the three algorithms on sampled instances of f_{plane} with $\sigma_{\text{PSF}} = 0.9$, $\sigma_{\text{filter}} = 1.0$, $s \in \{0.1, 0.25, 0.5, 0.75\}$ and using fifty evenly distributed points on the hemisphere for $\boldsymbol{n}$

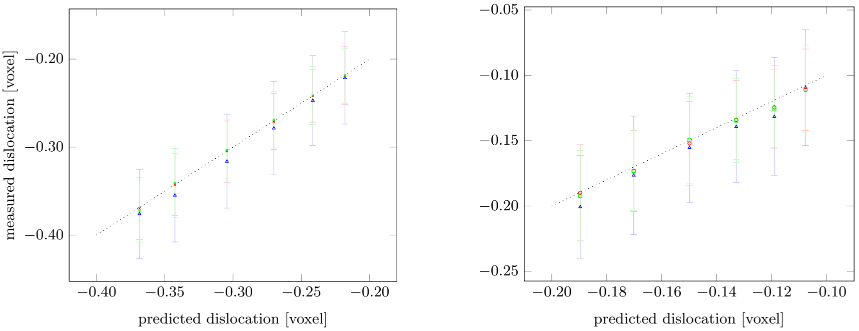


Fig. 3. comparison of predicted and measured localization bias for spherical (left) and cylindrical (right) surfaces using $R = 5$, $\sigma_{\text{PSF}} = 0.9$ with SNR $= 10$ for six evenly distributed filter scales $\sigma_{\text{filter}} \in [0.2, 0.4]$. Values have been averaged over 10 instances with different sub-voxel shift.

fig. 2b. Finally, fig. 2c shows that the estimated standard derivation matches the prediction from theory very well.

In fig. 3 we compare the predicted localization bias for spherical and cylindrical surfaces according to eq. 2 and eq. 3 respectively. Test-volumes have been generated from eq. 1 for spheres and using oversampling as described above for cylinders. As model parameters we used $R = 5$, $\sigma_{\mathrm{PSF}} = 0.9$ and various values for σ_{filter} with addition of Gaussian noise such that $\mathrm{SNR} = 10$. For each set of model parameters radii have then been estimated from 10 instances with same model parameters but with different sub-voxel shift. From these figures we conclude that our algorithms correctly reproduces the localization bias prevailing over the parabolic fit which possesses a systematic error.

For performance comparison, we measured execution time on a Linux PC with a Pentium D 3.4 GHz processor and 2 GB of RAM for test-volumes with shape $(200, 200, 200)^{\mathsf{T}}$ and two real CT-volumes. Results are given in table 1. As one can see the line search based method is only $\approx 35\%$ slower than the parabolic fit and the conjugate gradient based method only from $\approx 50\%$ to $\approx 90\%$ slower.

Table 1. performance results for various test volumes and real CT-volumes. Columns in the middle give run-times in seconds.

volume	shape	p. fit	l.search	cg	n. surfels
plane	$(200, 200, 200)^{\mathsf{T}}$	9.24	11.63	14.13	≈ 39500
sphere	$(200, 200, 200)^{\mathsf{T}}$	9.76	14.03	18.80	≈ 48000
cylinder	$(200, 200, 200)^{\mathsf{T}}$	10.91	15.51	20.74	≈ 75200
lobster	$(301, 324, 56)^{\mathsf{T}}$	6.22	8.24	10.19	21571
foot	$(256, 256, 207)^{\mathsf{T}}$	20.01	26.74	34.00	74411

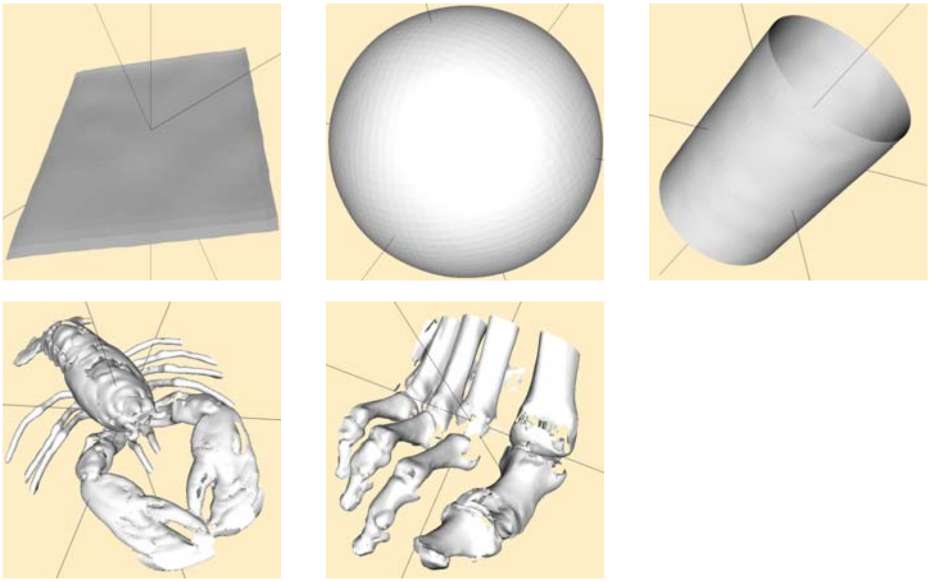


Fig. 4. Surface reconstructions for test-volumes and real CT-volumes using α-shapes [4] ($\alpha = 1$) with $\mathrm{SNR} = 10$ for the test-volumes

6 Conclusions

Based on the well-known Canny edge detector, we presented a new algorithm for subvoxel-precise 3D edge detection. The accuracy of our method is much better than the accuracy of the subvoxel refinement based on a parabola fit. Due to an efficient implementation of the spline interpolation and due to the use of fast voxel-accurate computations where-ever possible, our algorithm is still computationally efficient. In order to justify the accuracy, we theoretically analyzed the measurement errors of an ideal Canny-like edge detector on an infinite sampling resolution in case of 3D planar, spherical and cylindrical surfaces. Our analysis showed, that all experimental results are in full agreement with the theory, while this is not the case for the parabola fit method.

References

1. Brent, R.P.: Algorithms for Minimisation Without Derivatives. Prentice-Hall, Englewood Cliffs (1973)
2. Canny, J.: A computational approach to edge detection. TPAMI 8(6), 679–698 (1986)
3. Devernay, F.: A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical Report 2724, INRIA Sophia Antipolis (1995)
4. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Trans. Graph. 13(1), 43–72 (1994)
5. Jonker, P.P.: Skeletons in n dimensions using shape primitives. Pattern Recognition Letters 23, 677–686 (2002)
6. Köthe, U.: Vigra. Web Resource, `http://hci.iwr.uni-heidelberg.de/vigra/` (visited March 1, 2009)
7. Köthe, U.: Reliable Low-Level Image Analysis. Habilitation thesis, University of Hamburg, Germany (2008)
8. Luo, L., Hamitouche, C., Dillenseger, J., Coatrieux, J.: A moment-based three-dimensional edge operator. IEEE Trans. Biomed. 40(7), 693–703 (1993)
9. Mendonça, P.R.S., Padfield, D.R., Miller, J., Turek, M.: Bias in the localization of curved edges. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 554–565. Springer, Heidelberg (2004)
10. Monga, O., Deriche, R., Rocchisani, J.: 3d edge detection using recursive filtering: application to scanner images. CVGIP: Image Underst. 53(1), 76–87 (1991)
11. Moré, J.J., Thuente, D.J.: Line search algorithms with guaranteed sufficient decrease. ACM Trans. Math. Software 20, 286–307 (1994)
12. Udupa, J.K., Hung, H.M., Chuang, K.S.: Surface and volume rendering in three dimensional imaging: A comparison. J. Digital Imaging 4, 159–169 (1991)
13. Unser, M., Aldroubi, A., Eden, M.: B-Spline signal processing: Part I—Theory. IEEE Trans. Signal Process. 41(2), 821–833 (1993)
14. Unser, M., Aldroubi, A., Eden, M.: B-Spline signal processing: Part II—Efficient design and applications. IEEE Trans. Signal Process. 41(2), 834–848 (1993)
15. Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C++: The Art of Scientific Computing. Cambridge University Press, Cambridge (2002)