Surface Area Estimation of Digitized Planes Using Weighted Local Configurations

Joakim Lindblad

Centre for Image Analysis, Uppsala University
Lägerhyddsv. 3, SE–75237 Uppsala, Sweden
joakim@cb.uu.se

Abstract. We describe a method for estimating surface area of three-dimensional binary objects. The method assigns a surface area weight to each $2 \times 2 \times 2$ configuration of voxels. The total surface area is given by a summation of the local area contributions for a digital object. We derive optimal area weights, in order to get an unbiased estimate with minimum variance for randomly oriented planar surfaces. This gives a coefficient of variation (CV) of 1.40% for planar regions. To verify the results and to address the feasibility for area estimation of curved surfaces, the method is tested on convex and non-convex synthetic test objects of increasing size. The algorithm is appealingly simple and uses only a small local neighbourhood. This allows efficient implementations in hardware and/or in parallel architectures.

Keyword: Surface area estimation, marching cubes, optimal weights, digital planes, local voxel configurations

1 Introduction

Surface area of three-dimensional (3D) objects is an important feature for image analysis. In digital image analysis, we are given only a digitized version of the original continuous object. Digital surface area measurements can therefore only be estimates of the true surface area of the original object. Quantitative analysis of digital images requires that such estimates are both accurate and precise, i.e., that the estimates agree well with the true measure on the continuous object and that we get similar values for repeated measurements.

A good estimator should be unbiased, i.e., the expected value of the estimate should be equal to the true value. To be precise, an estimator should also have as small Mean Squared Error (MSE) as possible. For an unbiased estimator the MSE is equal to the variance σ^2.

From a theoretical point of view, multigrid convergence, is a very appealing property of an estimator (see e.g. [6]). This ensures that the estimate converges toward the true value, as the grid resolution increases. However, from a practical viewpoint, grid resolution is rarely a parameter that can be easily increased.

A property that is attractive from a practical point of view, is locality. Local techniques compute features using information only from a local part of the image. Computation of different parts of the image are independent, and can thus

I. Nyström et al. (Eds.): DGCI 2003, LNCS 2886, pp. 348–357, 2003.

be performed in parallel. Local algorithms are, in general, simple to implement and efficient in terms of computing power. This together with their inherent parallelism makes them most suitable for demanding real-time applications. Unfortunately, due to the limited distance that information is allowed to travel in local algorithms, it is believed that they never can be made multigrid convergent in a general sense.

This leads to a trade-off situation, between the desire for a local algorithm, due to reasons of simplicity and speed, and the improved performance at higher resolutions given by multigrid convergent estimators. The best estimator for one particular situation may well differ from the best choice in another situation.

In this paper, we present a method to obtain accurate surface area estimates with high precision, using only local computations and avoiding strong assumptions about the object of study. The estimator is based on assigning area weights to local configurations of binary voxels. The weights for the different configurations are optimized in order to give an unbiased estimate with minimal MSE for an isotropic distribution of flat surfaces of infinite size. We verify the performance of the estimator by applying it to synthetic test objects of increasing size with randomized alignment in the digitization process.

2 Background

In 2D image analysis, the perimeter of a digital object can be estimated as the cumulative distance from pixel centre to pixel centre along the border of the object. This is straightforward to accomplish using the Freeman chain code [4], but results in an incorrect estimate. The weights 1 for isothetic and $\sqrt{2}$ for diagonal steps are not optimal when measuring digitized line segments. By assigning optimized weights [8,12] to the steps, a more accurate perimeter estimate is obtained. Weights for the 2D case have been optimized for infinitely long straight lines and have then been proven to perform even better for curved contours [3]. In addition to the above local type of estimators, different multigrid convergent perimeter estimators exist, most of them based on finding straight line segments and performing a polygonalization of the object. See e.g [2], for a compact overview.

A straightforward and simple approach to get a surface area estimate of a 3D object is to count the number of foreground voxels with a neighbour in the background. This, however, results in a quite severe underestimate. Assigning a weight to each border voxel of 1.2031 instead of 1, we get an unbiased estimate for an isotropic distribution of planar surfaces [11]. The variance of that estimate is however rather large, due to the fact that we do not differentiate between different types of surface voxels.

Mullikin and Verbeek [11] propose a method where each voxel in the foreground is classified depending on the number of six-connected neighbours it has in the background. This gives a total of 9 different types of configurations of boundary voxels. Three of these configurations, namely voxels with one, two, and three neighbouring voxels in the background, occur for flat surfaces and are also, the by far most commonly appearing for smooth objects sampled at reasonable resolution. Mullikin and Verbeek derive optimal weights for these three

cases, in order to get an unbiased estimate with minimal MSE for planar surfaces. The weights are $W_1 = 0.8940$, $W_2 = 1.3409$, and $W_3 = 1.5879$, respectively, in sample grid units squared. This gives an unbiased estimate with a coefficient of variation (CV=σ/μ) of 2.33% for planar surfaces.

Since their method is dependent on what is defined to be foreground and background in the image, the estimate will change if applied to the complementary image. The true area of the surface between a continuous object and the background does not change if we interchange what is object and what is background, and a desired property of a surface area estimator is, therefore, that it gives the same result on the complementary image. Mullikin and Verbeek suggest to use the average of the estimate for the original and the complementary image, to achieve a symmetric estimate. Note that this property of symmetry with respect to foreground and background is slightly dependent on the chosen digitization method. We have in this paper used Gauss centre point digitization, where the digitized object is defined to be the set of all grid points (voxel centres) contained in the continuous set.

Recent publications [5,7,1] have studied multigrid convergent surface area estimators with promising results. Klette et al. [7] use global polyhedrization techniques to arrive at a surface area estimate, whereas Coeurjolly et al. [1] present an efficient algorithm based on discrete normal vector field integration, where the problem of surface area estimation is transformed into a problem of normal vector estimation.

3 Surface Area Estimation

The method proposed in this paper has similarities to the one described in [11]. Both methods count the occurrence of a set of local configurations of binary valued voxels. Each configuration is assigned an area contribution and the total area is calculated as the sum of the local area contributions over the surface of the object. Where Mullikin and Verbeek use the six-connected neighbourhood, we use a $2 \times 2 \times 2$ neighbourhood. This gives a total of 13 different types of surface configuration, five of them appearing for flat surfaces. The increased number of cases gives a better discrimination between different normal directions, and therefore an improved surface estimate is achieved. In addition, since the $2 \times 2 \times 2$ neighbourhood is symmetric with respect to foreground and background, the estimate does not change if we apply it to the complementary image.

The use of the $2 \times 2 \times 2$ neighbourhood makes the configurations appearing in the method similar to the ones of the *Marching Cubes* algorithm [10]. In fact, assigning a surface area to each configuration equal to the surface area of the triangles of the Marching Cubes algorithm, is not a bad idea. This leads to an overestimate of 8%. If we divide the result with the factor 1.08, we get an unbiased estimate with a remaining CV of 2.25%. In [9], empirical optimization when varying the weight of one of the configurations (case 5) was studied with promising results. Note that the optimization that we perform in this paper is based only on the binary voxel configurations, and we do not care about possible triangulations of the actual surface.

3.1 *m*-Cubes

An *m-cube* (short for Marching Cube), is the cube bounded by the eight voxels in a $2 \times 2 \times 2$ neighbourhood. Hence, each corner of the m-cube corresponds to a voxel. In a binary image, the possible number of configurations of the eight voxels is 256. Using symmetry, the 256 configurations can be grouped into 14 (or 15) cases [10], see Fig. 1. We number the cases according to [10], except for the mirrored cases 11 and 14, which we group into one, case 11.

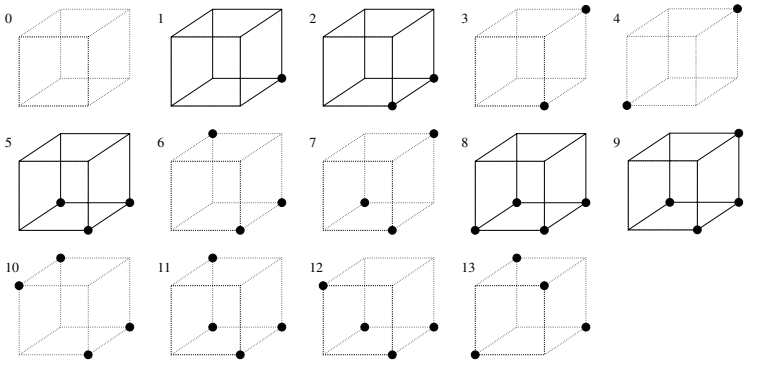


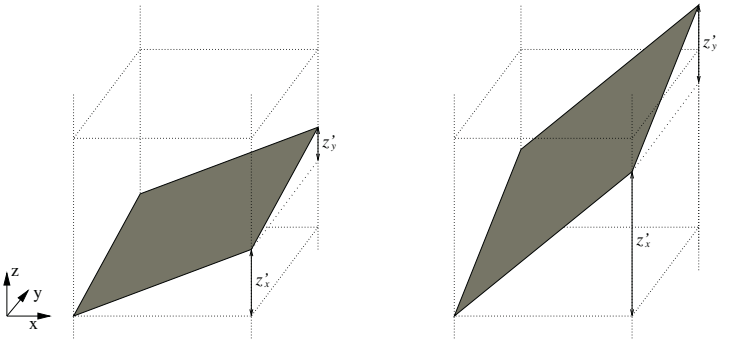
Fig. 1. m-cubes of $2 \times 2 \times 2$ voxels. Voxels denoted by a • are inside the object. The complementary cases are classified to be the same as the original cases. Only cases 1, 2, 5, 8, and 9, (*emphasized*) appear for planar surfaces.

Surface area weights, A_i, are assigned to the different cases. One of the configurations, case 0, does not represent a boundary situation, and therefore has zero area contribution. The sum of the area weights for all surface m-cubes of an object gives an estimate of the total surface area of that object. The histogram presenting the cardinality, N_i, of each of the 13 surface configurations (skipping case 0) is computed for each digitized object. The surface area estimate of the particular object is then calculated as

$$\hat{A} = \sum_{i=1}^{13} A_i N_i \,. \tag{1}$$

3.2 Planar Surfaces

We optimize the weights, A_i, in order to get an unbiased estimate with minimal MSE for planar surfaces of isotropic orientation. This can be justified by the fact that the surface of an object becomes locally planar as the sampling density increases if the maximum spatial frequency is kept constant. Furthermore, planar surfaces of distinct orientation represents the worst case objects for the estimation method. Since all other objects have a more isotropic distribution of



(a) $z'_{\mathrm{x}} + z'_{\mathrm{y}} \leq 1$ (b) $z'_{\mathrm{x}} + z'_{\mathrm{y}} > 1$

Fig. 2. The surface between object and background, $z = z'_{\mathrm{x}}x + z'_{\mathrm{y}}y + w$, here shown for $w = 0$.

normal directions, the variance of the estimated surface area of any reasonably shaped object should be lower than that of planar objects. This is verified on the synthetic test objects in Sect. 4.

To optimize the surface area weights, we need to study the different types of configurations that appear when a volume of voxels is divided by a randomly oriented and positioned plane. We can, without loss of generality, restrict the analysis to planes with a reduced set of normal directions, due to the symmetry of the sampling grid. We have chosen, in spherical coordinates, the region

$$-\pi \leq \phi < -\tfrac{3\pi}{4}, \qquad 0 \leq \theta < \tfrac{\pi}{2} + \arctan(\cos(\phi))\,, \tag{2}$$

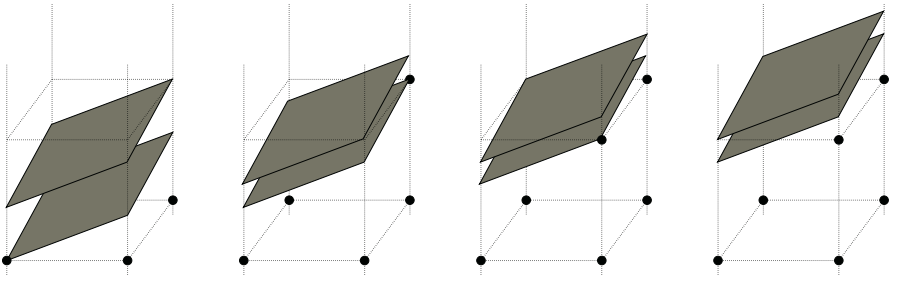
where the transformation from spherical to Cartesian coordinates of the normal vector is given by $\boldsymbol{n} = (\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$. The reason for choosing this set of normal directions is that it allows us to represent the surface plane as a function of x and y,

$$z(x,y) = z'_{\mathrm{x}}x + z'_{\mathrm{y}}y + w\,, \qquad 0 \leq z'_{\mathrm{y}} \leq z'_{\mathrm{x}} < 1\,, \tag{3}$$

where voxels with a centre on, or below, the plane are included in the object. Figure 2 shows two such surfaces of different slopes.

Depending on if $z'_{\mathrm{x}} + z'_{\mathrm{y}}$ is less or greater than 1, we get two different sets of configurations appearing, as we vary the offset term w. This is illustrated in Figs. 3 and 4. We keep track of the intersection between the surface and all m-cubes. For example, in Fig. 3(b) the lower m-cube is a case 5 and the upper one is a case 1. Note that only five of the 13 possible surface configurations appear for planar surfaces.

Since the offset term for randomly aligned planes is uniformly distributed, we can calculate the probability, $P(c_i)$, that an intersected m-cube is of type i, given a specific normal direction $\boldsymbol{n}$, directly from Fig. 3 for $z'_{\mathrm{x}} + z'_{\mathrm{y}} \leq 1$,



(a) Case 8 $0 \le w < 1-z'_{\mathrm{x}}-z'_{\mathrm{y}}$ (b) Case 5+1 $1-z'_{\mathrm{x}}-z'_{\mathrm{y}} \le w < 1-z'_{\mathrm{x}}$ (c) Case 2+2 $1-z'_{\mathrm{x}} \le w < 1-z'_{\mathrm{y}}$ (d) Case 1+5 $1-z'_{\mathrm{y}} \le w < 1$

Fig. 3. The different cases appearing for $z'_{\mathrm{x}} + z'_{\mathrm{y}} \le 1$ as w is varied are shown.

$$P(c_1|\boldsymbol{n}) = 2z'_{\mathrm{y}}/z_{\mathrm{tot}}\,, \tag{4a}$$
$$P(c_2|\boldsymbol{n}) = 2(z'_{\mathrm{x}} - z'_{\mathrm{y}})/z_{\mathrm{tot}}\,, \tag{4b}$$
$$P(c_5|\boldsymbol{n}) = 2z'_{\mathrm{y}}/z_{\mathrm{tot}}\,, \tag{4c}$$
$$P(c_8|\boldsymbol{n}) = (1 - z'_{\mathrm{x}} - z'_{\mathrm{y}})/z_{\mathrm{tot}}\,, \tag{4d}$$

and from Fig. 4 for $z'_{\mathrm{x}} + z'_{\mathrm{y}} > 1$,

$$P(c_1|\boldsymbol{n}) = 2z'_{\mathrm{y}}/z_{\mathrm{tot}}\,, \tag{5a}$$
$$P(c_2|\boldsymbol{n}) = 2(z'_{\mathrm{x}} - z'_{\mathrm{y}})/z_{\mathrm{tot}}\,, \tag{5b}$$
$$P(c_5|\boldsymbol{n}) = 2(1 - z'_{\mathrm{x}})/z_{\mathrm{tot}}\,, \tag{5c}$$
$$P(c_9|\boldsymbol{n}) = (z'_{\mathrm{x}} + z'_{\mathrm{y}} - 1)/z_{\mathrm{tot}}\,, \tag{5d}$$

where $z_{\mathrm{tot}} = 1 + z'_{\mathrm{x}} + z'_{\mathrm{y}}$.

The total number of intersected m-cubes for a plane of area A and normal direction $\boldsymbol{n}$ is

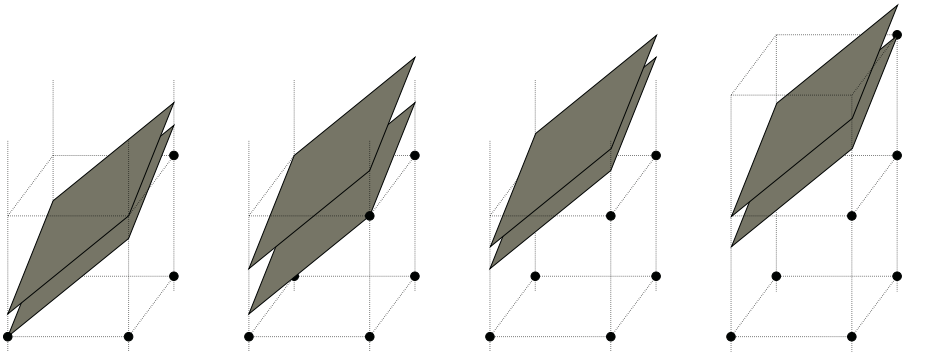
$$N_{\mathrm{tot}}(\boldsymbol{n}) = \frac{1+z'_{\mathrm{x}}+z'_{\mathrm{y}}}{\sqrt{1+{z'_{\mathrm{x}}}^2+{z'_{\mathrm{y}}}^2}}\,A\,. \tag{6}$$

The cardinality of a specific configuration, i, is

$$N_i(\boldsymbol{n}) = P(c_i|\boldsymbol{n})N_{\mathrm{tot}}(\boldsymbol{n})\,, \qquad i = 1\ldots 13 \tag{7}$$

and the total estimated surface area is given by

$$\hat{A}(\boldsymbol{n}) = \sum_{i=1}^{13} A_i N_i(\boldsymbol{n})\,. \tag{8}$$



(a) Case 5+1 $0 \leq w < 1-z'_x$ (b) Case 2+2 $1-z'_x \leq w < 1-z'_y$ (c) Case 1+5 $1-z'_y \leq w < 2-z'_x-z'_y$ (d) Case 1+9+1 $2-z'_x-z'_y \leq w < 1$

Fig. 4. The different cases appearing for $z'_x + z'_y > 1$ as w is varied are shown.

3.3 Optimization

We wish to optimize (8) over all normal directions, in order to get an unbiased estimate with minimal MSE. However, since the cardinality of cases 1, 5, and 9, are linearly dependent,[1] $N_1 = N_5 + 2N_9$, the solution becomes non-unique.

By grouping the cases that co-appear, we can still get a unique solution. That is, instead of assigning a surface area to each individual m-cube, we assign an area to the different situations shown in Figs. 3 and 4. Optimizing area contributions, using standard methods, to this new set of cases, leads to the following grouped set of area contributions, in sample grid units squared.

$$A_1 + A_5 = 1.1897, \quad 2A_2 = 1.3380, \quad A_8 = 0.9270, \quad 2A_1 + A_9 = 1.6942. \quad (9)$$

This gives an unbiased area estimate with a CV of 1.40% for planar surfaces.

For non-planar objects the relation between cases 1, 5, and 9, no longer holds. Therefore, we need area contributions for the individual m-cubes. Since the Marching Cubes triangulation of cases 1 and 9 together represent a well behaved and flat surface (Fig. 5), we have used the ratio given by the corresponding triangle areas, i.e., $\frac{A_1}{A_9} = \frac{1}{6}$. Inserting this into (9) we get the following set of area contributions.

$$A_1 = 0.2118, \; A_2 = 0.6690, \; A_5 = 0.9779, \; A_8 = 0.9270, \; A_9 = 1.2706. \quad (10)$$

This specific choice, of how to distribute the area between cases 1, 5, and 9, is not indisputable. Since the optimization for planar surfaces does not supply enough information to give a unique solution, further optimization on some other type of objects is required. Note, however, that the distribution of area between cases 1, 5, and 9, affects neither the CV nor the maximum error for planar surfaces, as long as (9) holds. The maximum absolute error is reached for planes

[1] Easily verified by observing Figs. 3 and 4. For cases 5 and 9, the neighbouring voxel (voxels for case 9) will always contain a complementary case 1.

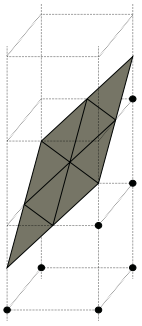


Fig. 5. Possible triangulation of cases 1 and 9.

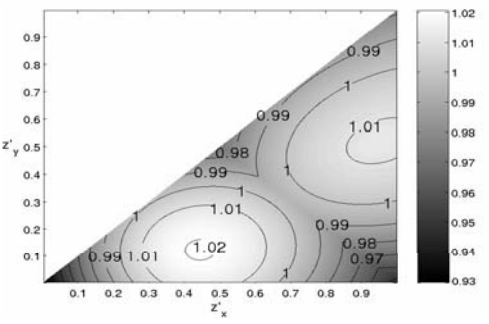


Fig. 6. Estimated surface area divided by true surface area, for different values of z'_x and z'_y. The maximum error is reached for $z'_x = z'_y = 0$.

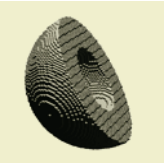


Fig. 7. Digitized spherical cap of radius 40 pixels.

aligned with the digitization grid, and the error is then $1 - A_8 = 0.0730$. Figure 6 shows the estimate divided by the true area as a function of the normal direction.

4 Simulations

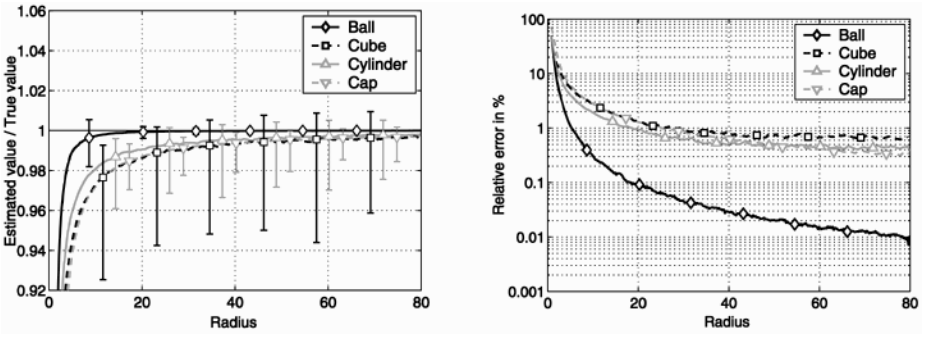
To verify the results and to address the feasibility for area estimation of curved and non-convex surfaces, the method is tested on synthetic objects of known surface area. The used test objects are balls of radii 0–80, cubes of side length 0–160, cylinders of $height = 2 \cdot radius$, and, to get a concave object, thick spherical caps (see Fig 7) where $radius_{\text{cavity}} = \frac{1}{2} \cdot radius_{\text{cap}}$. The objects are generated in the continuous space and then digitized using Gauss centre point digitization in different sizes, and with random rotation and position in the digitization grid.

Since our test objects are not only planar surfaces, some additional m-cube cases will be present. To calculate a total area for our test objects we need to assign an area contribution also to those additional cases. Since these cases constitute not more than 0.2% of the total number of surface m-cubes of the test objects, this area contribution will have a very small impact on the overall result. We have assigned the triangle area of a Marching Cubes triangulation to these additional cases.

5 Results

Surface area estimates and average relative errors for digitized objects of increasing resolution can be seen in Fig. 8. Surface area estimates for 10,000 digitized balls of radius 70 pixels, 20,000 cubes of side length 140, 10,000 cylinders of radius 70 and height 140, and 10,000 spherical caps of radius 70 pixels, are summarized in Table 1. The results are (on average) a slight underestimate of the true surface area. This is due to the cutting of corners and edges [13]. For large objects this effect can be neglected, though.

The surface of a large ball is a good sampling of planes in all directions and should thus exhibit very low variance. This is verified by the simulations, where



(a) (b)

Fig. 8. (a) Surface area estimates divided by true surface area for 150,000 digitizations of objects of increasing size. (b) Log plot of relative error for the different objects.

Table 1. Performance on synthetic test objects of radius 70 / side length 140.

Object	$\text{mean}(\hat{A})/A$	CV	$\text{mean}(\lvert\hat{A}-A\rvert/A)$	$\max(\lvert\hat{A}-A\rvert/A)$
Ball	0.9999	0.015%	0.012%	0.082%
Cube	0.9963	0.89%	0.68%	7.14%
Cylinder	0.9975	0.63%	0.45%	3.36%
Spherical Cap	0.9965	0.32%	0.36%	1.99%

superlinear convergence $O(r^{-\alpha}), \alpha \approx 1.5$, is observed. The cube have planar surfaces which are aligned so that it represents a worst case situation for the cubic digitization grid, accordingly it shows the worst performance in the simulations.

6 Discussion and Conclusions

We have presented a method for estimating surface area of binary 3D objects using local computations. The algorithm is appealingly simple and uses only a very small local neighbourhood, allowing efficient implementations in hardware and/or in parallel architectures. The estimated surface area is computed as a sum of local area contributions. We have derived optimal area weights for the $2 \times 2 \times 2$ configuration of voxels that appear on digital planar surfaces. The method gives an unbiased estimate with minimum variance for randomly oriented planar surfaces. Theoretic worst case CV for the suggested surface area estimator is 1.40%, and the maximum absolute error is 7.30%. The maximum error is reached for planar surfaces aligned with the digitization grid. The performance of the surface area estimator is verified on more than 200,000 convex and non-convex digitized synthetic objects. Due to the local nature of the method, it cannot be made multigrid convergent. However, for objects of size less than a few hundred voxels in diameter, it is competitive in terms of precision with existing multigrid convergent methods. A more detailed comparison is of inter-

est. Further work on finding an optimal and unique distribution of surface area between cases 1, 5, and 9 will follow.

Acknowledgements

We thank Doc. Ingela Nyström, Prof. Gunilla Borgefors, and Nataša Sladoje, for their strong scientific support.

References

1. D. Coeurjolly, F. Flin, O. Teytaud, and L. Tougne. Multigrid convergence and surface area estimation. In *Theoretical Foundations of Computer Vision "Geometry, Morphology, and Computational Imaging"*, volume 2616 of *LNCS*, pages 101–119. Springer-Verlag, 2003.
2. D. Coeurjolly and R. Klette. A comparative evaluation of length estimators. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR)*, pages IV: 330–334. IEEE Computer Science, 2002.
3. L. Dorst and A. W. M. Smeulders. Length estimators for digitized contours. *Computer Vision, Graphics and Image Processing*, 40:311–333, 1987.
4. H. Freeman. Boundary encoding and processing. In B. S. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*, pages 241–266, New York, 1970. Academic Press.
5. Y. Kenmochi and R. Klette. Surface area estimation for digitized regular solids. In L. J. Latecki, R. A. Melter, D. M. Mount, and A. Y. Wu, editors, *Vision Geometry IX*, pages 100–111. Proc. SPIE 4117, 2000.
6. R. Klette. Multigrid convergence of geometric features. In G. Bertrand, A. Imiya, and R. Klette, editors, *Digital and Image Geometry*, volume 2243 of *LNCS*, pages 314–333. Springer-Verlag, 2001.
7. R. Klette and H. J. Sun. Digital planar segment based polyhedrization for surface area estimation. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *Visual Form 2001*, volume 2059 of *LNCS*, pages 356–366. Springer-Verlag, 2001.
8. Z. Kulpa. Area and perimeter measurement of blobs in discrete binary pictures. *Computer Graphics and Image Processing*, 6:434–454, 1977.
9. J. Lindblad and I. Nyström. Surface area estimation of digitized 3D objects using local computations. In *Proceedings of the 10th International Conference on Discrete Geometry for Computer Imagery (DGCI)*, volume 2301 of *LNCS*, pages 267–278. Springer-Verlag, 2002.
10. W. E. Lorensen and H. E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th ACM SIGGRAPH on Computer Graphics*, volume 21, pages 163–169, 1987.
11. J. C. Mullikin and P. W. Verbeek. Surface area estimation of digitized planes. *Bioimaging*, 1(1):6–16, 1993.
12. D. Proffit and D. Rosen. Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges. *Computer Graphics and Image Processing*, 10:318–332, 1979.
13. I. T. Young. Sampling density and quantitative microscopy. *Analytical and Quantitative Cytology and Histology*, 10(4):269–275, 1988.