Surface Volume Estimation of Digitized Hyperplanes Using Weighted Local Configurations

Joakim Lindblad

Centre for Image Analysis, Uppsala University, Uppsala, Sweden
joakim@cb.uu.se

Abstract. We present a method for estimating the surface volume of four-dimensional objects in discrete binary images. A surface volume weight is assigned to each $2 \times 2 \times 2 \times 2$ configuration of image elements. The total surface volume of a digital 4D object is given by a summation of the local volume contributions. Optimal volume weights are derived in order to provide an unbiased estimate with minimal variance for randomly oriented digitized planar hypersurfaces. Only 14 out of 64 possible boundary configurations appear on planar hypersurfaces. We use a marching hypercubes tetrahedrization to assign surface volume weights to the non-planar cases. The correctness of the method is verified on four-dimensional balls and cubes digitized in different sizes. The algorithm is appealingly simple; the use of only a local neighbourhood enables efficient implementations in hardware and/or in parallel architectures.

Keywords: surface volume estimation, marching cubes, digital hyperplanes, 4D, cell tiling.

1 Introduction

In many applications of digital image analysis, quantitative geometrical measures, such as length and area of objects, are of foremost interest. When working with three-dimensional (3D) digital images, an often desired measure is the surface area of a digitized object. With modern imaging techniques and powerful computers, it has become interesting to look at higher dimensional data volumes. The four-dimensional (4D) counterpart to surface area is surface volume. In this paper we present a method to perform accurate surface volume estimations of 4D objects in binary digital images using a technique based on local cell tiling.

In [11, 12] we presented a surface area estimator for 3D images that utilises only local computations and a small local neighbourhood to obtain an estimate that is very fast to calculate and still exhibits good performance in terms of accuracy, precision and robustness. In this paper we extend this methodology to four dimensions, and derive optimal surface volume weights for the hyxel (hyper volume picture element) configurations that appear on planar hypersurfaces.

E. Andres et al. (Eds.): DGCI 2005, LNCS 3429, pp. 252–262, 2005.

2 Previous and Related Work

Visualization methods for high dimensional data are relatively well developed. There exist tools for 4D plane-tracing (a 4D extension of ray-tracing) and many techniques based on splatting. The popular Marching Cubes algorithm [13], for generating a triangulated iso-surface from voxel data, has also been extended to higher dimensions [1]. Far less is available in the field of image analysis and the task of extracting quantitative data from high-dimensional images. Concerning the geometry of digital 4D objects, work on, e.g., distance transforms [2] and skeletonization [6] in 4D has been presented. To the best of our knowledge, no surface volume estimation technique for discrete 4D data has previously been presented in literature. However, many similarities with surface area estimates of 3D objects and perimeter estimates of 2D objects do exist.

The perimeter of a digitized 2D object can be estimated as the cumulative distance from pixel centre to pixel centre along the border of the object, where an isothetic step is given weight 1 and a diagonal one is given weight $\sqrt{2}$. This is straightforward to accomplish using the Freeman chain code [5], but results in rather big over-estimates. Starting from an assumption that the boundary of an object is locally linear, optimal weights for the local steps have been derived, leading to an unbiased estimator with a minimal mean squared error (MSE) [10, 14]. A similar approach can be taken in order to estimate the surface area of digitized 3D objects. By counting the local configurations of voxels that appear on the boundary of a digital object a fast and accurate area estimate is achieved. In [11, 12] optimal surface area weights were derived, providing an unbiased estimator with minimal MSE. The method described in this paper is a direct extension of this technique to the 4D case.

In addition to the local type of estimators mentioned above, different multigrid convergent perimeter and surface area estimators exist, see, e.g., [4] for an overview of perimeter estimators, and [7, 9, 3] for examples of multigrid convergent surface area estimators. This class of estimators ensure convergence toward the true value as the grid resolution increases [8]. Many multigrid convergent estimators are based on finding straight line/plane segments. However, in order to do so we can no longer use local algorithms. Coeurjolly et al. [3] have presented efficient algorithms based on discrete normal vector field integration, where the problem of perimeter/surface area estimation is transformed into a problem of normal vector estimation. It seems that this approach may be extended to higher dimensions. To our knowledge no one has so far attempted to do so.

3 Surface Area Estimation

To introduce the methodology, we present, in this section, a brief derivation of the 3D version of the method; for measuring the surface area of a digitized 3D object. For a more detailed description see [12].

The estimation is based on counting local configurations of $2 \times 2 \times 2$ voxels. In a binary image, the number of possible configurations of the eight voxels is

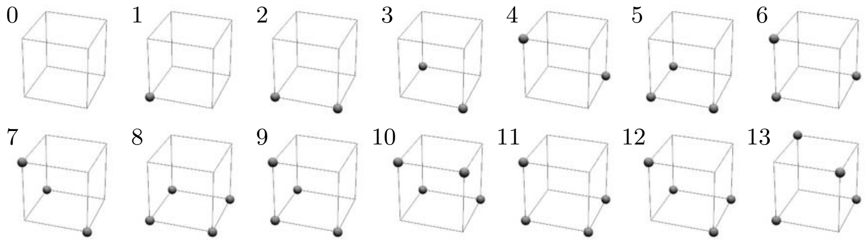


Fig. 1. The 14 major 3-cubes of 2 × 2 × 2 voxels. Marked voxel centres are inside the object. The complementary cases are classified to be the same as the original cases. Only cases c_1, c_2, c_5, c_8, and c_9, appear for planar surfaces

$2^{2^3} = 256$. Using rotation, mirror, and complement symmetry, the 256 configurations can be grouped into 14 major cases c_i, see Fig. 1.

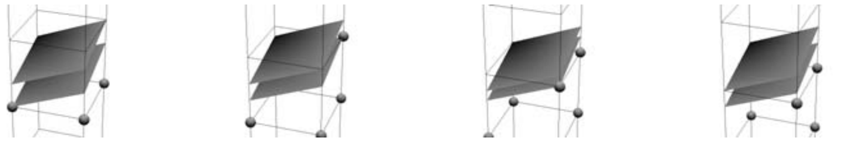
A surface area contribution A_i is assigned to each case (c_0 does not represent a boundary situation, and therefore has zero area contribution). The number of occurrences N_i of each of the 13 surface configurations is computed for the digitized object, and the surface area estimate $\hat{A}$ of the object is calculated as

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

The area contributions A_i are optimally selected so as to provide an unbiased estimate with a minimal MSE when the method is applied to infinite planes (the surfaces of half-spaces) digitized over an isotropic distribution of normal directions. This optimization can be justified by the fact that the surface of an object with limited curvature becomes locally planar as the sampling density increases. Only five of the 13 possible surface configurations appear for planar surfaces. We call these five cases the planar cases. They dominate the boundary of most objects digitized at a resolution high enough to capture the details of the surface structure.

When performing the optimization, we can, due to the symmetry of the sampling grid, without loss of generality, restrict the study to planes that can be expressed as a function $z(x, y) = z'_{\mathrm{x}}x + z'_{\mathrm{y}}y + k$, $0 \leq z'_{\mathrm{y}} \leq z'_{\mathrm{x}} < 1$. Voxels with a centre on, or below, the plane are included in the object. We vary the offset term k and observe the configurations that appear when a plane of a given normal direction cuts a column of cubes (an infinite stack of cubes in the z-direction). Depending on if $z'_{\mathrm{x}} + z'_{\mathrm{y}}$ is less or greater than 1, two different sets are observed. This is shown in Figs. 2 and 3. We keep track of the intersections between the surface and all cubes in the column. For example, in Fig. 2(b) the lower cube is of type c_5 and the upper one is of type c_1.

For a plane in general position the offset term is uniformly distributed. Given a specific normal direction $\boldsymbol{n}$, the expected number of occurrences of each case $E_i = E[N_i|\boldsymbol{n}]$ per column of intersected cubes, can be directly calculated from Figs. 2 and 3.



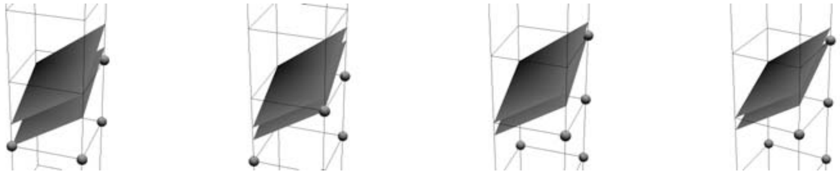
(a) Case 8 $0 \leq k < 1{-}z'_{\mathrm{x}}{-}z'_{\mathrm{y}}$

(b) Case 5+1 $1{-}z'_{\mathrm{x}}{-}z'_{\mathrm{y}} \leq k < 1{-}z'_{\mathrm{x}}$

(c) Case 2+2 $1{-}z'_{\mathrm{x}} \leq k < 1{-}z'_{\mathrm{y}}$

(d) Case 1+5 $1{-}z'_{\mathrm{y}} \leq k < 1$

Fig. 2. The different cases appearing for $z'_{\mathrm{x}} + z'_{\mathrm{y}} \leq 1$ as k is varied



(a) Case 5+1 $0 \leq k < 1{-}z'_{\mathrm{x}}$

(b) Case 2+2 $1{-}z'_{\mathrm{x}} \leq k < 1{-}z'_{\mathrm{y}}$

(c) Case 1+5 $1{-}z'_{\mathrm{y}} \leq k < 2{-}z'_{\mathrm{x}}{-}z'_{\mathrm{y}}$

(d) Case 1+9+1 $2{-}z'_{\mathrm{x}}{-}z'_{\mathrm{y}} \leq k < 1$

Fig. 3. The different cases appearing for $z'_{\mathrm{x}} + z'_{\mathrm{y}} > 1$ as k is varied

For $z'_{\mathrm{x}} + z'_{\mathrm{y}} \leq 1$,

$E_1 = 2z'_{\mathrm{y}}, \quad E_2 = 2(z'_{\mathrm{x}} - z'_{\mathrm{y}}), \quad E_5 = 2z'_{\mathrm{y}}, \quad E_8 = 1 - z'_{\mathrm{x}} - z'_{\mathrm{y}},$

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

$E_1 = 2z'_{\mathrm{y}}, \quad E_2 = 2(z'_{\mathrm{x}} - z'_{\mathrm{y}}), \quad E_5 = 2(1 - z'_{\mathrm{x}}), \quad E_9 = (z'_{\mathrm{x}} + z'_{\mathrm{y}} - 1).$

The number of intersected columns of cubes, for a planar surface segment of area A and normal direction $\boldsymbol{n}$, is $N_{\mathrm{col}}(\boldsymbol{n}) = \frac{1}{\sqrt{1+{z'_{\mathrm{x}}}^2+{z'_{\mathrm{y}}}^2}} A$. The estimated surface area is given by

$$\hat{A}(\boldsymbol{n}) = \sum_{i=1}^{13} A_i E_i(\boldsymbol{n}) N_{\mathrm{col}}(\boldsymbol{n}) \,. \tag{2}$$

The MSE of (2) is minimized over all normal directions, while keeping zero bias, in order to find optimal values for A_i. On a planar surface cases 1, 5, and 9 always appear together, which leads to a non-unique solution. The reason for this is that the planar surface continues into the neighbouring cubes and divides them in a way that creates the co-appearing cases. Leaving A_1 as a variable we obtain the following solution:

$$A_2 = 0.669, \quad A_5 = 1.190 - A_1, \quad A_8 = 0.927, \quad A_9 = 1.694 - 2A_1. \tag{3}$$

These weights provide, independent on how we choose A_1, an unbiased area estimate with a coefficient of variation (CV=μ/σ) of 1.40% for planar surfaces.

In order to estimate the surface area of general object boundaries, area contributions have to be assigned to all the 13 surface cases. Since the non-planar

Table 1. Table of elementary areas assigned to the different 3-cube cases. A_1 is left undefined

$A_1 =$ undefined	$A_2 = 0.6690$	$A_3 = 2A_1$	$A_4 = 2A_1$	
$A_5 = 1.1897 - A_1$	$A_6 = A_1 + A_2$	$A_7 = 3A_1$	$A_8 = 0.9270$	
$A_9 = 1.6942 - 2A_1$	$A_{10} = 2A_2$	$A_{11} = 1.5731$	$A_{12} = A_1 + A_5$	$A_{13} = 4A_1$

cases are, in general, scarcely appearing in real object volumes, the area contribution assigned to these cases will have a limited impact on the overall surface area estimate. Dividing the cubes into (locally) face-connected components, all but one of the additional cases can be decomposed into the five planar ones. This way we introduce a minimal number of new values. The only truly new configuration is c_{11}, which we here assign an area of 1.5731 (derived from a Marching Cubes triangulation of that case). The surface area weights of all 13 cases are summarized in Table 1, where A_1 is left undefined. Note, however, that this subdivision of cases is not uniquely determined. For example, c_7 can either be split into $3c_1$ or $c_9 + c_1$ depending on if we look at the original or the complementary case.

For curved surfaces, the relation between cases c_1, c_5, and c_9 no longer holds, and the specific choice of the free parameter A_1 will affect the estimation result. In [12] the freedom to choose the value of A_1 is used to minimize the estimation error of the method when applied to a distribution of digitized balls of increasing radii.

4 Surface Volume Estimation

For 4D surface volume estimation we use configurations of 2 × 2 × 2 × 2 hyxels, which in a binary image gives $2^{2^4} = 65\,536$ different configurations. Using rotation, mirror, and complement symmetry, they can be grouped into 222 major cases. 14 of these, shown in Fig. 4, appear on the surfaces of planar volumes. Just as in the 3D case, we restrict the study to hyperplanes that can be expressed as a function $w(x, y, z) = w'_{\mathrm{x}}x + w'_{\mathrm{y}}y + w'_{\mathrm{z}}y + k$, $0 \leq w'_{\mathrm{z}} \leq w'_{\mathrm{y}} \leq w'_{\mathrm{x}} < 1$. We vary

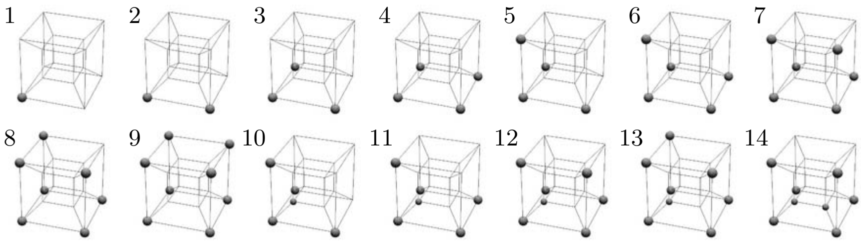
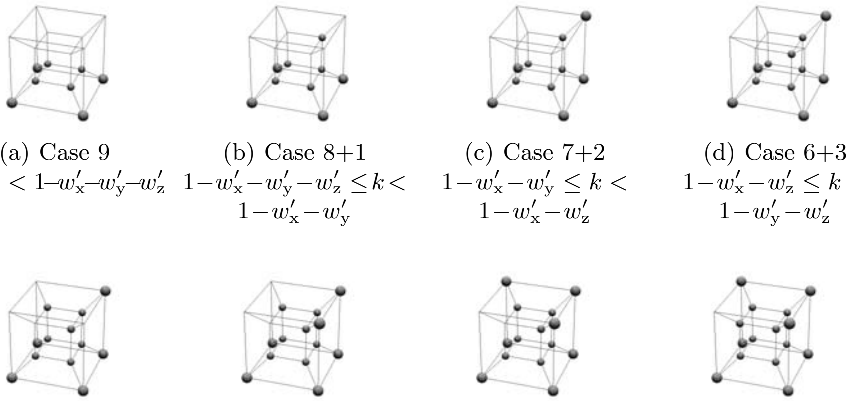


Fig. 4. The 14 planar 4-cubes, appearing on the surface of planar volumes. Marked hyxel centres are inside the object



(a) Case 9
$0 \le k < 1 - w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}$

(b) Case 8+1
$1 - w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}} \le k < 1 - w'_{\mathrm{x}} - w'_{\mathrm{y}}$

(c) Case 7+2
$1 - w'_{\mathrm{x}} - w'_{\mathrm{y}} \le k < 1 - w'_{\mathrm{x}} - w'_{\mathrm{z}}$

(d) Case 6+3
$1 - w'_{\mathrm{x}} - w'_{\mathrm{z}} \le k < 1 - w'_{\mathrm{y}} - w'_{\mathrm{z}}$

(e) Case 5+5
$1 - w'_{\mathrm{y}} - w'_{\mathrm{z}} \le k < 1 - w'_{\mathrm{x}}$

(f) Case 3+6
$1 - w'_{\mathrm{x}} \le k < 1 - w'_{\mathrm{y}}$

(g) Case 2+7
$1 - w'_{\mathrm{y}} \le k < 1 - w'_{\mathrm{z}}$

(h) Case 1+8
$1 - w'_{\mathrm{z}} \le k < 1$

Fig. 5. The different cases appearing for $w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} \le 1, w'_{\mathrm{x}} \le w'_{\mathrm{y}} + w'_{\mathrm{z}}$ as k is varied

the offset term k and study the configurations that appear when a hyperplane of a given normal direction cuts a column (in the w-direction) of 4-cubes.

Depending on the slope of the hyperplane, one out of 14 different sets of configurations appear. The set, $w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} \le 1, w'_{\mathrm{x}} \le w'_{\mathrm{y}} + w'_{\mathrm{z}}$, is shown in Fig. 5. Just as in the 3D case, it is straightforward, albeit a bit tedious, to calculate the expected number of cells of each type that appear when a column of 4-cubes is intersected by a hyperplane of a given normal direction. The configurations are listed in the Appendix. The number of intersected columns of 4-cubes, for a planar surface segment of volume V and normal direction $\boldsymbol{n}$, is $N_{\mathrm{col}}(\boldsymbol{n}) = \frac{1}{\sqrt{1+w_{\mathrm{x}}'^2+w_{\mathrm{y}}'^2+w_{\mathrm{z}}'^2}} V$. The estimated surface volume is given by

$$\hat{V}(\boldsymbol{n}) = \sum_{i=1}^{14} V_i E_i(\boldsymbol{n}) N_{\mathrm{col}}(\boldsymbol{n}) \,. \tag{4}$$

We minimize the MSE of (4) over all normal directions, while keeping zero bias, in order to find optimal values for V_i. Just as in the 3D case, we do not get enough information from using only planar objects to find a unique solution. Leaving V_1, V_2, V_3, V_5 as variables the optimization leads to the weights presented in Table 2. These weights provide an unbiased volume estimate with a CV of

Table 2. Table of elementary volumes assigned to the planar 4-cubes

$V_1 =$ undefined	$V_2 =$ undefined	$V_3 =$ undefined	$V_4 = 0.668$
$V_5 =$ undefined	$V_6 = 0.609 - V_3 + V_5$	$V_7 = 1.194 - V_2$	$V_8 = 1.707 - V_1 - V_5$
$V_9 = 0.920$	$V_{10} = 0.972 - V_1$	$V_{11} = 1.558 - V_1 - V_3$	
$V_{12} = 2.113 - V_1 - V_2 - V_5$	$V_{13} = 2.630 - 2V_1 - 2V_5$	$V_{14} = 1.680 - 2V_2$	

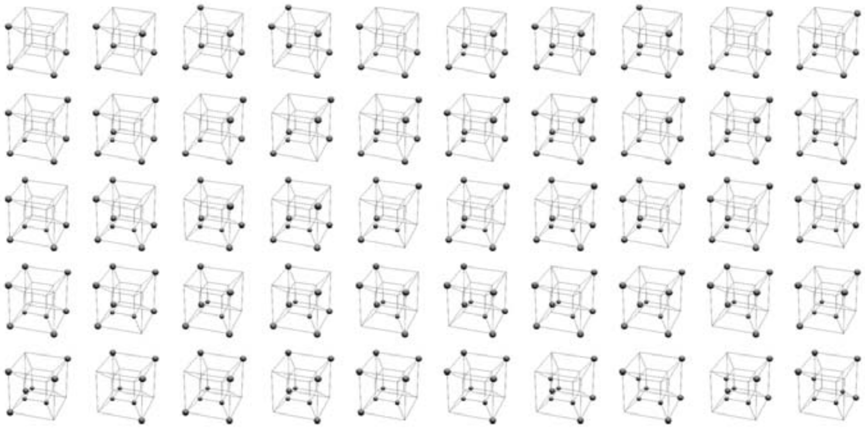


Fig. 6. The 50 non-planar 4D sub-cases

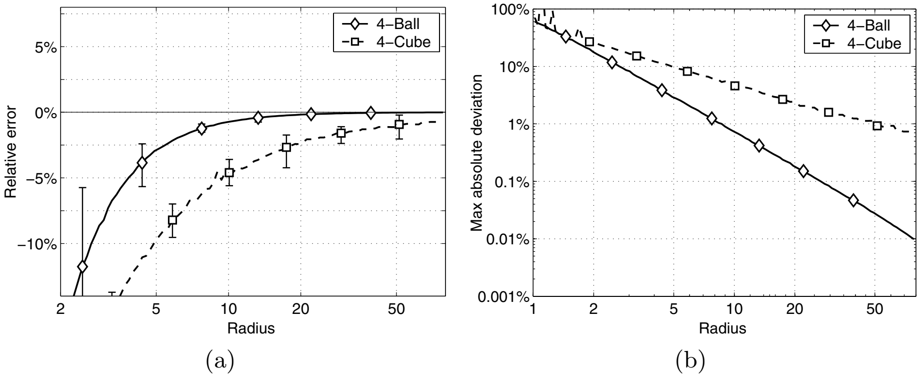
0.77% for planar hypersurfaces, independent on how we choose V_1, V_2, V_3, V_5. The maximum absolute error, AD=7.97% is reached for hyperplanes aligned with the digitization grid.

4.1 Curved Volumes

In order to estimate the surface volume of curved 4D objects, we need to assign volume contributions to all 221 surface cases. Similar as for the 3D case, we can reduce this number by dividing the 4-cubes into (locally) volume connected components. This way the 222 major cases can be reduced to 65 sub-cases. Note that this is less than the number given by Roberts [15]. The reason for this is that we consider it valid to look at the complement also when splitting the cells into components (since complement is used anyway to reach the 222 major cases). This is similar to the optional splitting of c_7 into $c_9 + c_1$ in the 3D case. It is an open question how to assign an optimal surface volume weight to the 50 non-planar sub-cases (shown in Fig. 6).

5 Simulations

To verify and evaluate the performance of the estimator, we test the method on synthetic objects of known surface volumes. The used test objects are 4-balls of radii 1–80 hyxels and 4-cubes of side lengths 2–160 hyxels. We generate 30 000 instances of each object in the continuous space and digitize them using Gauss digitization, with a random orientation and position in the digitization grid. We have assigned surface volume weights derived from a marching hypercubes tetrahedrization [1] to the non-planar cases that appear on the edges of the 4-cubes, and to the unassigned planar cases, c_1, c_2, c_3, and c_5.



(a) (b)

Fig. 7. Relative error (a) and absolute deviation (b) of the surface volume estimates for digitized objects of increasing size. The error bars indicate minimum and maximum values of the estimate

Average relative error and maximum absolute deviation for the surface volume estimation method are shown in Fig. 7. The surface of a large 4-ball is a good sampling of hyperplanes of all normal directions and the described surface volume estimator is therefore expected to exhibit very low variance on such objects. This is verified by the simulations, where superlinear convergence $O(r^{-\alpha})\,, \alpha \approx 2$, is observed. Improved low resolution performance can be achieved by adjusting the weights V_1, V_2, V_3, and V_5. This is beyond the scope of this paper.

6 Summary

We have presented a method for estimating the surface volume of binary 4D objects using local computations. The surface volume is computed as a sum of local volume contributions. Optimal volume weights for the $2 \times 2 \times 2 \times 2$ configurations of hyxels that appear on digital planar hypersurfaces are derived. The method gives an unbiased estimate with minimum variance for randomly oriented planar hypersurfaces. Theoretic worst case CV for the suggested surface volume estimator is 0.77%, and the maximum absolute error is 7.97%. The maximum error is reached for planar hypersurfaces aligned with the digitization grid. The solution for planar volumes is not unique and freedom in the choice of parameters may be used to improve the performance at lower resolutions in a manner similar to what was done for the 3D case in [12]. For curved volumes additional cases appear. It is an open question how to assign optimal volume weights to these cases.

Acknowledgements. We thank Nataša Sladoje for valuable scientific support, Dr. Rephael Wenger for making his programs for generating iso-surface lookup tables freely available on the Internet, and Prof. Nahun Kiryati for (gently) pushing us out into the fourth dimension.

References

[1] P. Bhaniramka, R. Wenger, and R. Crawfis. Isosurface construction in any dimension using convex hulls. *IEEE Trans. on Vision and Computer Graphics*, 10(2):130–141, 2004.

[2] G. Borgefors. Weighted digital distance transforms in four dimensions. *Discrete Applied Mathematics*, 125(1):161–176, 2003.

[3] D. Coeurjolly, F. Flin, O. Teytaud, and L. Tougne. Multigrid convergence and surface area estimation. In *Theoretical Foundations of Computer Vision*, volume 2616 of *LNCS*, pages 101–119. Springer-Verlag, 2003.

[4] 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, Quebec, 2002. IEEE Computer Science.

[5] 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.

[6] P. P. Jonker. Skeletons in N dimensions using shape primitives. *Pattern Recognition Lett*, 23(4):677–686, 2002.

[7] 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.

[8] 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.

[9] 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, Capri, Italy, 2001. Springer-Verlag.

[10] Z. Kulpa. Area and perimeter measurement of blobs in discrete binary pictures. *Computer Graphics and Image Processing*, 6:434–454, 1977.

[11] J. Lindblad. Surface area estimation of digitized planes using weighted local configurations. In I. Nyström, G. Sanniti di Baja, and S. Svensson, editors, *DGCI*, volume 2886 of *LNCS*, pages 348–357, Naples, Italy, 2003. Springer-Verlag.

[12] J. Lindblad. Surface area estimation of digitized 3D objects using weighted local configurations. *Image and Vision Computing*, 23(2):111–122, 2005. Special issue on Discrete Geometry for Computer Imagery.

[13] 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.

[14] 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.

[15] J. C. Roberts and S. Hill. Piecewise linear hypersurfaces using the marching cubes algorithm. In R. F. Erbacher and A. Pang, editors, *Visual Data Exploration and Analysis VI, Proceedings of SPIE*, volume 3643, pages 170–181, 1999.

Appendix

The expected number of occurrences of the 14 planar 4D-configurations when a column of 4-cubes is intersected by hyperplanes of different slopes. Cases which are not mentioned for a specific slope do not appear. Cases c_1 and c_2 appear with the same frequency for all slopes; $E_1 = 2w'_{\mathrm{z}}$, $E_2 = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}})$.

1a: $w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} \leq w'_{\mathrm{y}} + w'_{\mathrm{z}}.$

$E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$ $E_5 = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - w'_{\mathrm{x}}),$ $E_6 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$
$E_7 = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_8 = 2w'_{\mathrm{z}},$ $E_9 = 1 - w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}.$

1b: $w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} > w'_{\mathrm{y}} + w'_{\mathrm{z}}.$

$E_3 = 2w'_{\mathrm{z}},$ $E_4 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_6 = 2w'_{\mathrm{z}},$
$E_7 = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_8 = 2w'_{\mathrm{z}},$ $E_9 = 1 - w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}.$

2a: $w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} \leq 1, \quad w'_{\mathrm{x}} \leq w'_{\mathrm{y}} + w'_{\mathrm{z}}.$

$E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$ $E_5 = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - w'_{\mathrm{x}}),$ $E_6 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$
$E_7 = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_8 = 2(1 - w'_{\mathrm{x}} - w'_{\mathrm{y}}),$ $E_{13} = w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} - 1.$

2b: $w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} \leq 1, \quad w'_{\mathrm{x}} > w'_{\mathrm{y}} + w'_{\mathrm{z}}.$

$E_3 = 2w'_{\mathrm{z}},$ $E_4 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_6 = 2w'_{\mathrm{z}},$
$E_7 = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_8 = 2(1 - w'_{\mathrm{x}} - w'_{\mathrm{y}}),$ $E_{13} = w'_{\mathrm{x}} + w'_{\mathrm{y}} + w'_{\mathrm{z}} - 1.$

3a: $w'_{\mathrm{x}} + w'_{\mathrm{y}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} \leq w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 \leq w'_{\mathrm{z}}.$

$E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$ $E_5 = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - w'_{\mathrm{x}}),$ $E_6 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$
$E_7 = 2(1 - w'_{\mathrm{x}} - w'_{\mathrm{z}}),$ $E_{12} = 2(w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1),$ $E_{13} = 1 + w'_{\mathrm{z}} - w'_{\mathrm{x}} - w'_{\mathrm{y}}.$

3b: $w'_{\mathrm{x}} + w'_{\mathrm{y}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} > w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 \leq w'_{\mathrm{z}}.$

$E_3 = 2w'_{\mathrm{z}},$ $E_4 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_6 = 2w'_{\mathrm{z}},$
$E_7 = 2(1 - w'_{\mathrm{x}} - w'_{\mathrm{z}}),$ $E_{12} = 2(w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1),$ $E_{13} = 1 + w'_{\mathrm{z}} - w'_{\mathrm{x}} - w'_{\mathrm{y}}.$

3c: $w'_{\mathrm{x}} + w'_{\mathrm{y}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} \leq w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 > w'_{\mathrm{z}}.$

$E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$ $E_5 = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - w'_{\mathrm{x}}),$ $E_6 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$
$E_7 = 2(1 - w'_{\mathrm{x}} - w'_{\mathrm{z}}),$ $E_{12} = 2w'_{\mathrm{z}},$ $E_{14} = w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 - w'_{\mathrm{z}}.$

3d: $w'_{\mathrm{x}} + w'_{\mathrm{y}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} > w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 > w'_{\mathrm{z}}.$

$E_3 = 2w'_{\mathrm{z}},$ $E_4 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_6 = 2w'_{\mathrm{z}},$
$E_7 = 2(1 - w'_{\mathrm{x}} - w'_{\mathrm{z}}),$ $E_{12} = 2w'_{\mathrm{z}},$ $E_{14} = w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 - w'_{\mathrm{z}}.$

4a: $w'_{\mathrm{x}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{y}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} \leq w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 \leq w'_{\mathrm{z}}.$

$E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}),$ $E_5 = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - w'_{\mathrm{x}}),$ $E_6 = 2(1 - w'_{\mathrm{y}} - w'_{\mathrm{z}}),$
$E_{11} = 2(w'_{\mathrm{x}} + w'_{\mathrm{z}} - 1),$ $E_{12} = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_{13} = 1 + w'_{\mathrm{z}} - w'_{\mathrm{x}} - w'_{\mathrm{y}}.$

4b: $w'_{\mathrm{x}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{y}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} > w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 \leq w'_{\mathrm{z}}.$

$E_3 = 2w'_{\mathrm{z}},$ $E_4 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_6 = 2(1 - w'_{\mathrm{x}}),$
$E_{11} = 2(w'_{\mathrm{x}} + w'_{\mathrm{z}} - 1),$ $E_{12} = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}}),$ $E_{13} = 1 + w'_{\mathrm{z}} - w'_{\mathrm{x}} - w'_{\mathrm{y}}.$

4c: $w'_{\mathrm{x}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{y}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} \leq w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 > w'_{\mathrm{z}}.$

$$\begin{array}{lll} E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}), & E_5 = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - w'_{\mathrm{x}}), & E_6 = 2(1 - w'_{\mathrm{y}} - w'_{\mathrm{z}}), \\ E_{11} = 2(w'_{\mathrm{x}} + w'_{\mathrm{z}} - 1), & E_{12} = 2(1 - w'_{\mathrm{x}}), & E_{14} = w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 - w'_{\mathrm{z}}. \end{array}$$

4d: $w'_{\mathrm{x}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{y}} + w'_{\mathrm{z}} \leq 1, \quad w'_{\mathrm{x}} > w'_{\mathrm{y}} + w'_{\mathrm{z}}, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 > w'_{\mathrm{z}}.$

$$\begin{array}{lll} E_3 = 2w'_{\mathrm{z}}, & E_4 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}} - w'_{\mathrm{z}}), & E_6 = 2(1 - w'_{\mathrm{x}}), \\ E_{11} = 2(w'_{\mathrm{x}} + w'_{\mathrm{z}} - 1), & E_{12} = 2(1 - w'_{\mathrm{x}}), & E_{14} = w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 - w'_{\mathrm{z}}. \end{array}$$

5a: $w'_{\mathrm{y}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 \leq w'_{\mathrm{z}}.$

$$\begin{array}{lll} E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}), & E_5 = 2(1 - w'_{\mathrm{x}}), & E_{10} = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - 1), \\ E_{11} = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}), & E_{12} = 2(w'_{\mathrm{y}} - w'_{\mathrm{z}}), & E_{13} = 1 + w'_{\mathrm{z}} - w'_{\mathrm{x}} - w'_{\mathrm{y}}. \end{array}$$

5c: $w'_{\mathrm{y}} + w'_{\mathrm{z}} > 1, \quad w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 > w'_{\mathrm{z}}.$

$$\begin{array}{lll} E_3 = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}), & E_5 = 2(1 - w'_{\mathrm{x}}), & E_{10} = 2(w'_{\mathrm{y}} + w'_{\mathrm{z}} - 1), \\ E_{11} = 2(w'_{\mathrm{x}} - w'_{\mathrm{y}}), & E_{12} = 2(1 - w'_{\mathrm{x}}), & E_{14} = w'_{\mathrm{x}} + w'_{\mathrm{y}} - 1 - w'_{\mathrm{z}}. \end{array}$$