Digital Representation Schemes for 3-D Curves

Amnon Jonas* **and Nahum Kiryati**†
Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel

Abstract

Digital representation of three dimensional curves is considered. Two dimensional representation schemes are first reviewed, then methods applicable to curves in three and higher dimensional spaces are described. Desirable properties of curve representation schemes are identified and used as a basis for qualitative and quantitative comparison of the various methods that have been suggested. Conclusions concerning the selection of a digital 3-D curve representation scheme for shape analysis and coding are drawn. It is shown that grid intersect quantization is a poor choice for curve representation in 3-D space and that cube quantization, that leads to 6-connected chain codes, meets all the identified requirements and should be preferred.

1 Introduction

Digital representation of curves is an important step in computerized shape analysis. Typical digital curve representation schemes consist of a *quantization* (or *digitization*) stage followed by chain encoding. Chain codes are the standard input format for numerous shape analysis and machine vision algorithms. Various 2-D curve representation schemes have been considered in the literature [13, 14, 16, 18, 23, 32, 40, 41, 42, 44, 51, 55]. A valuable comparison between the main techniques appears in [32].

The details of the digital curve representation scheme can have great influence on the performance of subsequent algorithms. For example, Kulpa [35] resolved a controversy between two approaches to area measurement by showing that they relate to different representation schemes. Consider the evolution of estimators for the original perimeter of a 2-D shape from

*Currently with RDC Communications Ltd., P.O.Box 34233, Jerusalem 91341, Israel
†To whom correspondence should be addressed

its chain code, from the original work of Freeman [13, 14] to recent contributions [11, 33]. It is apparent that careful analysis of the digitization schemes and their effects on the properties of digitized straight lines [6, 9, 10, 12, 13, 19, 20, 37, 39, 47, 56] (that are a limiting case for fine sampling of smooth curves) has been the key to breakthroughs in the performance of perimeter estimators.

The increasing importance of three dimensional shape analysis in medical imaging, range image processing, image sequence analysis, robotic navigation and other domains necessitates the definition, evaluation and well founded selection of digital curve representation schemes in three dimensions. The 3-D chain code was defined by Freeman in 1974 [14], but without explicit specification of the quantization method. In the last decade several 3-D digital curve representation methods have been suggested. One of them is a generalization of the 2-D Grid Intersect Quantization (GIQ) scheme [24, 30], but note an important correction made in [53] concerning the properties of that method. Other techniques have been suggested in [1, 7, 27, 30, 53].

The cumulative experience in the analysis of curves, paths and surfaces in 3-D has proven that better understanding of 3-D digital curve representation schemes is crucial for the development of higher performance 3-D shape analysis algorithms. The extension of 2-D schemes to 3-D is surprisingly nontrivial. Important properties of 3-D schemes are unobvious and somewhat counterintuitive, and have escaped the attention of several researchers.

In this paper the concept of a mathematically 'well-behaved' 3-D curve representation scheme is formalized and summarized as a list of requirements. These are used as a basis for qualitative and quantitative comparison of the various methods that have been suggested. It is shown that grid intersect quantization is a poor curve representation scheme in 3-D and that cube quantization, that leads to 6-connected chain codes, meets all the identified requirements and should be preferred.

2 Notation and Basic Definitions

A lattice in $\mathbf{R}^n$ is a set of points created by n independent vectors $v_i \in \mathbf{R}^n$ [36]:

$$\{p \in \mathbf{R}^n : \ p = \sum_{i=1}^{n} p_i v_i, \ p_i \in \mathbf{Z}, \ i = 1, 2, ..., n\} \tag{1}$$

The lattice created by the unit vectors in the direction of the coordinate system axes, i.e., the set of all points in $\mathbf{R}^n$ with integer coordinates, is the *(n-dimensional) unit square lattice.* In the sequel, unless noted otherwise, the term *(n-D) lattice* stands for (n-D) unit square lattice. An *n-D centered hypervoxel* is an n-dimensional hypercube of unit length edge, with its center on a lattice point and its edges parallel to the axes. An *n-dimensional shifted hypervoxel* is an n-dimensional unit length hypercube, with all its corners on lattice points. A two dimensional hypervoxel (centered or shifted) is a *pixel*. A three dimensional hypervoxel is a *voxel*. In [5] the term *rexel* is suggested for a four dimensional hypervoxel.

Two distinct points on an n-dimensional lattice are *O(r)-adjacent* if each coordinate of one of them differs from the corresponding coordinate of the other by at most 1, and at least $n - r$ coordinates are equal. The points that are O(r)-adjacent to a specific point are called its *O(r)-neighbors*. In two or three dimensions it is common to use the term *m-adjacent* instead of O(r)-adjacent, where m is the total number of O(r)-adjacent points. Hence in 2-D “4-adjacent” and “8-adjacent” mean “O(1)-adjacent” and “O(2)-adjacent”, and in 3-D, “6,18, or 26-adjacent” mean “O(1), O(2), or O(3)-adjacent”, respectively.

An *O(r)-chain* is a sequence of lattice points, in which every two consecutive points are O(r)-adjacent. A subset S of a lattice is *O(r)-connected* if for any two points in S, there is an O(r)-chain in S with the two points at its ends [45]. An O(r)-chain in which each point (except the first and the last) has exactly two O(r)-adjacent neighbors, and each of the two end points has a single O(r)-adjacent neighbor, is a *digital arc* (in the O(r)-connectivity sense).

The unordered set of points obtained by digitization of a continuous curve is its *digital image*. A *digital curve* is a chain that (according to a given scheme) represents a continuous curve. If the continuous curve is a straight line, the digital curve is a *digital straight line*.

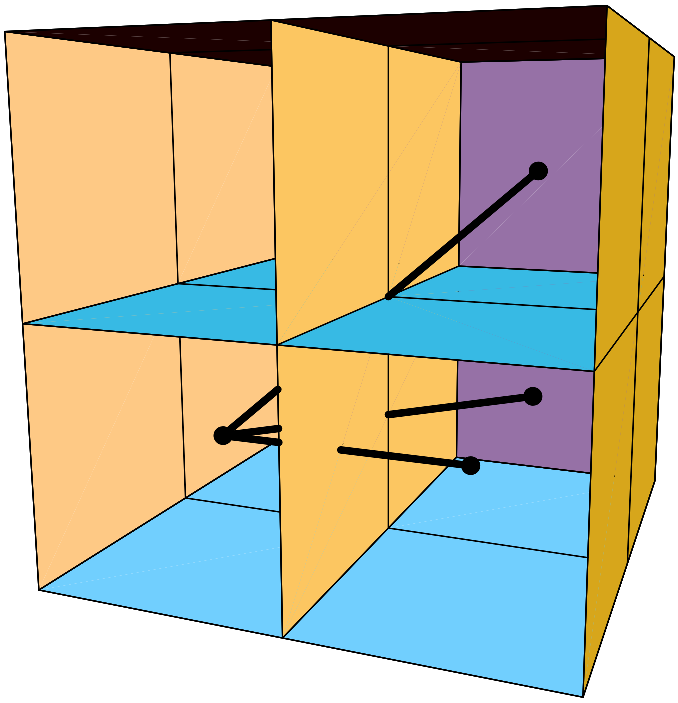

Figure 1: Three link types in three dimensional 26-chains

The segment connecting two consecutive chain points is a *link*. In *(m-chains)* each link is one of m possible vectors. A *chain code* representation of a digital curve consists of the coordinates of the starting point and the sequence of links [13]. Note that in two dimensional 8-chains the links can be classified according to their length: it is either 1 for links that are parallel to the axes, or $\sqrt{2}$ for diagonal links. In three dimensional 26-chains there are three types of links, type-I, type-II and type-III, of lengths 1, $\sqrt{2}$ and $\sqrt{3}$ respectively. See Fig. 1.

There are various measures for distance between lattice points (or other points) in n-dimensional space [5, 8, 30, 49, 50]. A common measure is the Euclidean distance between the two points. Another useful measure is the *chess-board distance* defined by

$$d_{CB}(p,q) = \max_{1 \le i \le n} |p_i - q_i|, \tag{2}$$

where p_i and q_i respectively denote the components of p and q.

Additional definitions and explanations about 2-D digital contours can be found in [13, 43, 46]. Extensions to 3-D are considered in [14, 48] and general formulations for multi-dimensions are described in [8, 54].

3 Brief Review of 2-D Curve Digitization Schemes

Consider first *Square Quantization* [13, 14, 23], in which the curve is implicitly segmented according to its intersections with centered pixels. The chain is taken to be the sequence of points at the centers of these pixels, following the order of the segments along the curve. For a curve in general position, the probability of corner crossing between pixels is zero and can be ignored, so diagonal links do not appear and a 4-chain is obtained. See Fig. 2.

Other schemes can be used to get 8-chains. Three of these are based on the intersection points between the curve and the edges (between two lattice points at the corners) of shifted pixels. In *Grid Intersection Quantization (GIQ)* [13, 14] the curve is represented by selecting, at each intersection, the closer lattice point of the pair connected by the intersected edge, as shown in Fig. 3. The order of points follows the order of intersections along the curve. (Note that Bresenham's algorithm [3] for digital display of straight lines is equivalent to GIQ of straight lines). Other approaches regard the curve as the boundary of an object. Then one of the pair of lattice points is in the object while the other is outside, i.e., in the background. *Object Boundary Quantization* selects from each pair the point in the object and *Background Boundary Quantization* selects the background point [18]. The points are ordered as in Grid Intersect Quantization (GIQ). See Fig. 4.

In *Convex Quantization* [32] (Fig. 5) a convex domain is placed around each lattice point. If the curve passes through the domain, the lattice point is included in the representation. Square quantization is a special case in which the convex domains are the centered pixels around each lattice point. *Circle Convex Quantization*, in which the convex domain is a circle

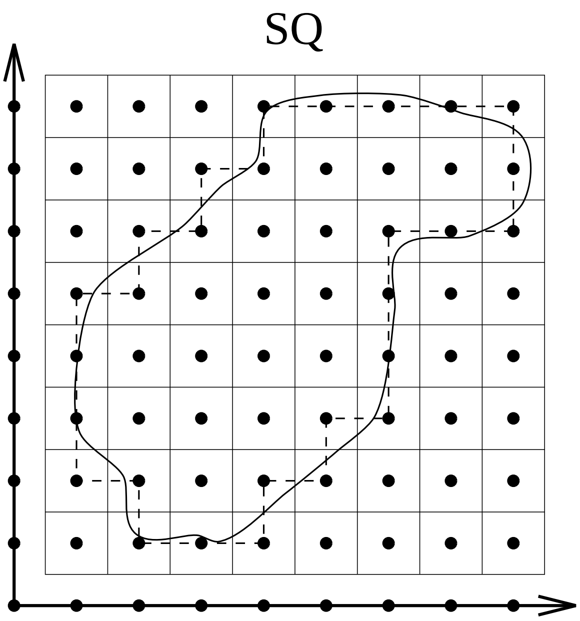

Figure 2: The Square Quantization method for 2-D curve representation.

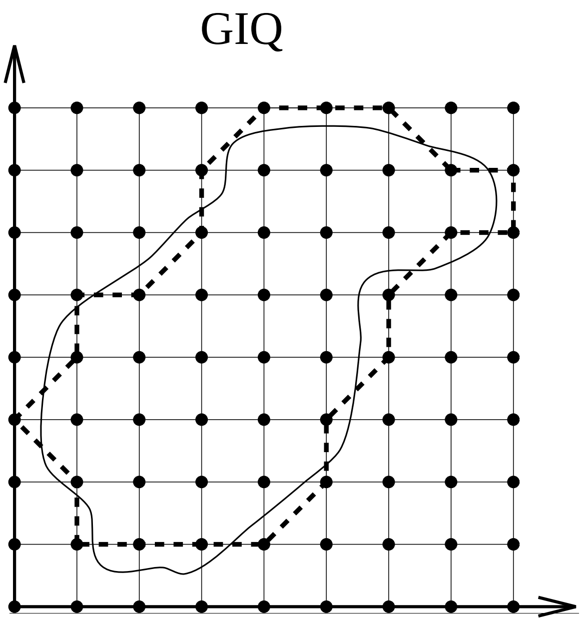

Figure 3: The Grid Intersect Quantization (GIQ) method for 2-D curve representation. At each intersection of the curve with an edge of a shifted pixel, the lattice point closer to the intersection is selected and included in the representation.

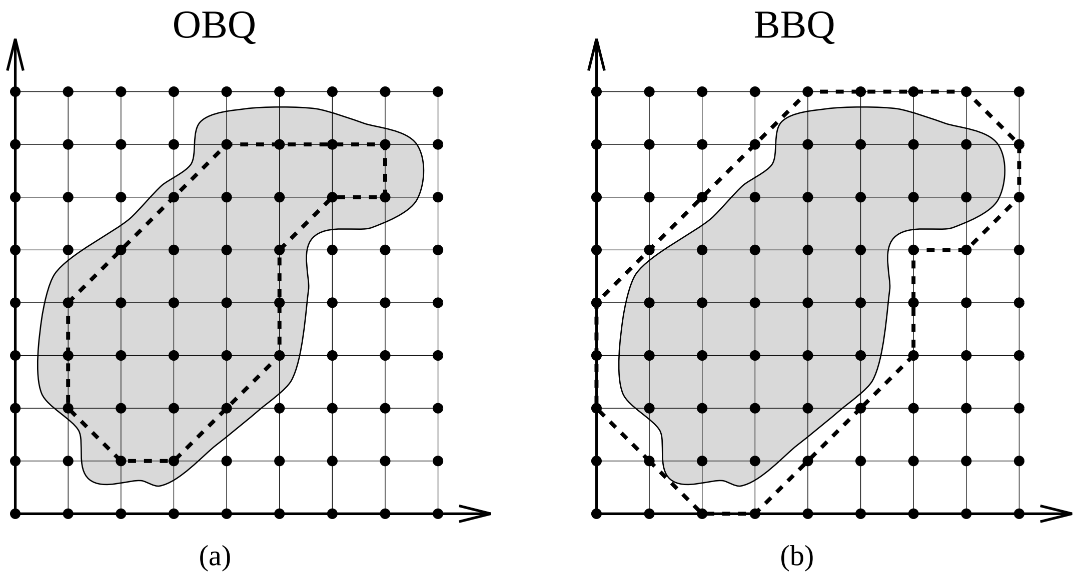

Figure 4: The Object Boundary Quantization (OBQ) and Background Boundary Quantization (BBQ) methods for 2-D curve representation. At each intersection of the curve with an edge of a shifted pixel, one of the two endpoints of the edge is selected and included in the representation. In OBQ it is the one outside the object, and in BBQ it it the point in the object.

of unit diameter centered at the lattice point leads to 8-chains. *Diamond Convex Quantization* has also been considered [32]. For straight lines and other curves that do not enter and exit a diamond through the same edge it is equivalent to Grid Intersect Quantization.

Generalized chain codes (consisting of longer steps between successive elements) [15, 17] and hexagonal grid quantization schemes [52] have also been suggested.

4 Three Dimensional and N-dimensional Methods

Most 2-D digital curve representation schemes do not lend themselves to generalization to higher dimensions. Object Boundary Quantization and Background Boundary Quantization cannot be used in three or higher dimensions since an object side and a boundary side of a curve do not exist. In 3-D Convex Quantization, connectivity is not always maintained, because the curve may pass between domains without crossing them. For example, if one uses spherical domains as a generalization of Circle Convex Quantization to three dimensions, then even an infinite line can pass between the spheres without intersecting any of them, see Fig. 6.

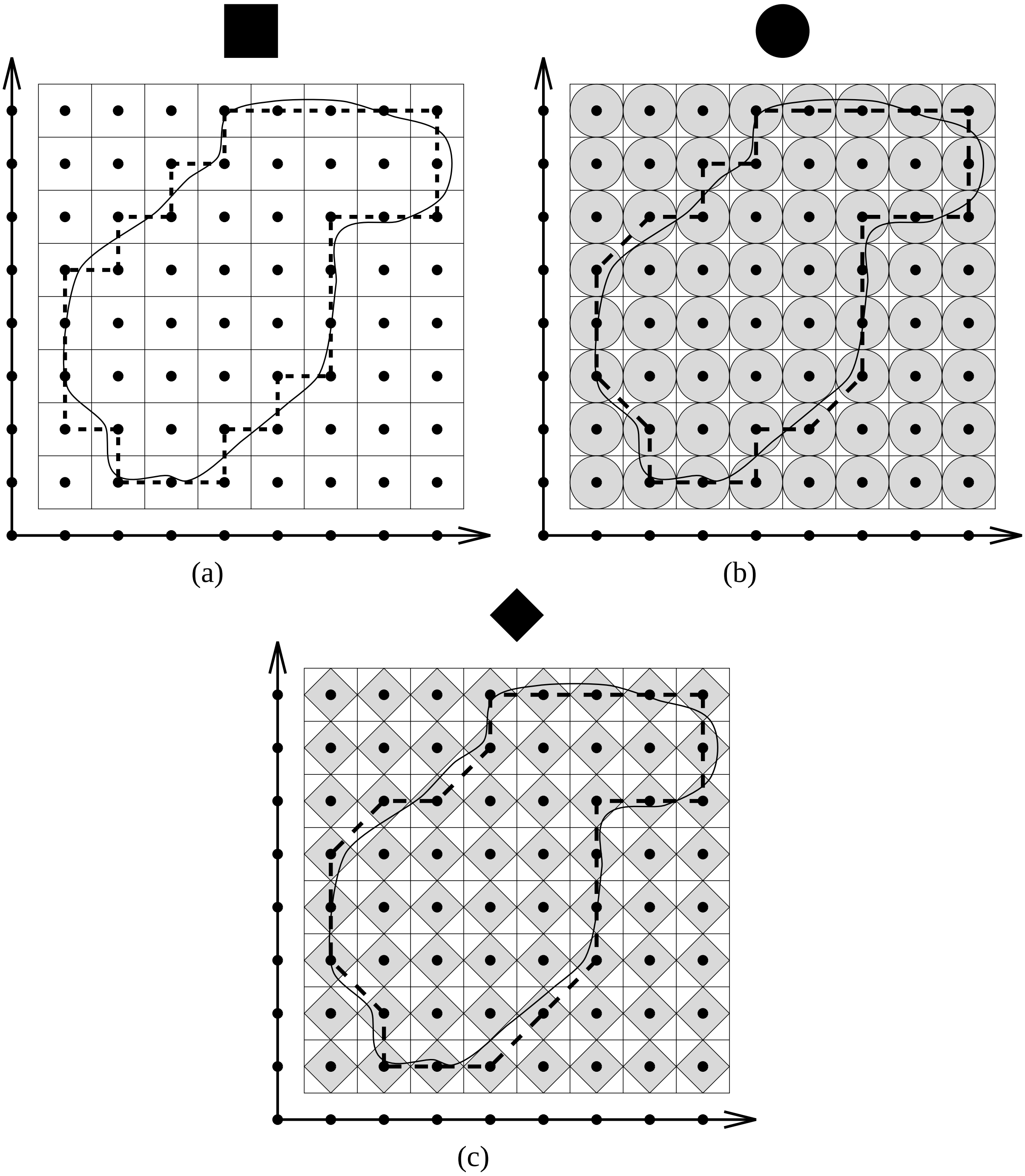

Figure 5: Two dimensional Convex Quantization. A convex domain is placed around each lattice point, and the point is included in the representation if the curve intersects the domain. (a) Square Quantization. (b) Circle Convex Quantization. (c) Diamond Convex Quantization.

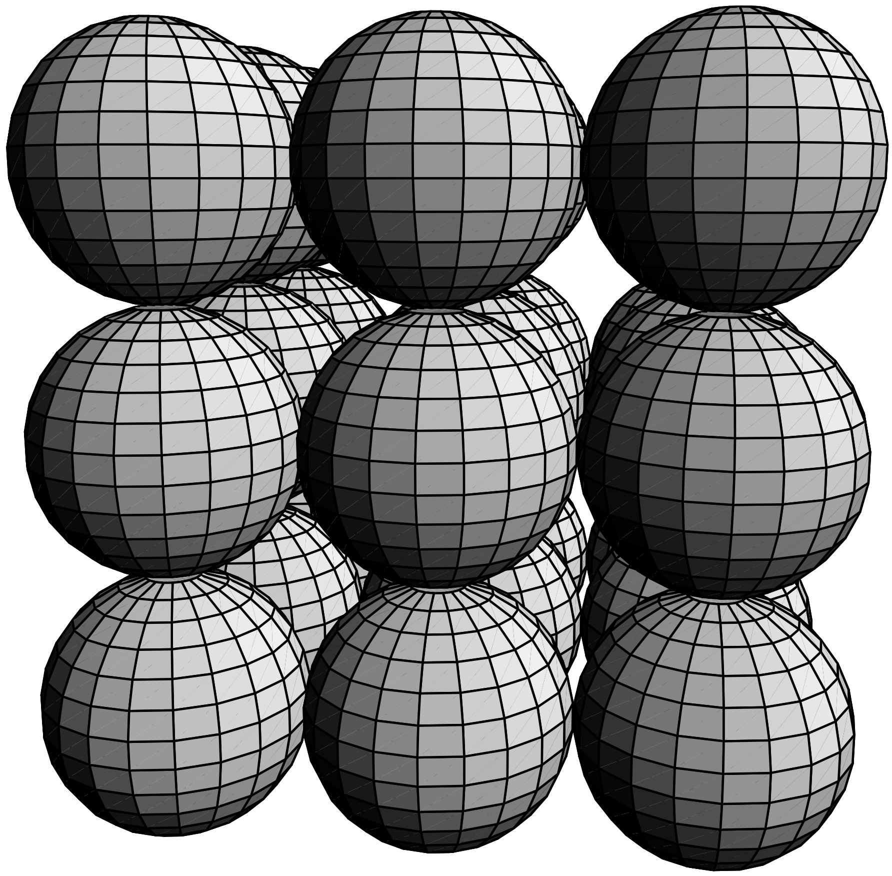

Figure 6: A naive attempt to generalize 2-D Circle Convex Quantization to 3-D Spherical Convex Quantization. Even an an infinite straight line can pass between the spheres without intersecting any of them.

Several 3-D digital curve representation schemes have been suggested in the literature. To obtain O(1)-chains (6-chains), Square Quantization can simply be generalized to *Cube Quantization (CQ)* [27, 30] by referring to "centered voxels" instead of "centered pixels". See Fig. 7. To obtain O(3)-chains (26-chains), Grid Intersect Quantization (GIQ) can be generalized to 3-D. In 3-D GIQ [24, 30], one takes the closest of the four lattice points to each intersection of the curve with the face of the shifted voxel as shown in Fig. 8. Several properties of 3-D GIQ are presented in [24], extended to multi-dimensions in [30] (and used in [2, 7]), but as shown in [53], not all are correct. Klette [30] developed a generalization of Convex Quantization that includes multidimensional Cube Quantization and Grid Intersect Quantization (GIQ) as special cases. GIQ was also extended [25, 38] into a 3-D *surface* quantization scheme. This seems to be the natural extension of 2-D GIQ to 3-D.

Maximum Length Quantization (MLQ) [22] also yields 26-chains in 3-D, and is readily applicable to higher dimensions. The curve is divided to segments by its intersections with the planes that are perpendicular to one of the coordinate axes and are midway between the integer coordinate planes. For each segment, the lattice point with the longest subsegment in its centered voxel is selected. This process is repeated for each of the axes. The selected points are then ordered according to the order of the corresponding subsegments on the curve. A 2-D illustration of MLQ is shown in Fig. 9. Unlike other curve quantization methods, in MLQ the quantization depends on the whole curve, not just on selected points such as grid intersection points. In 2-D, for a straight line, MLQ is equivalent to GIQ.

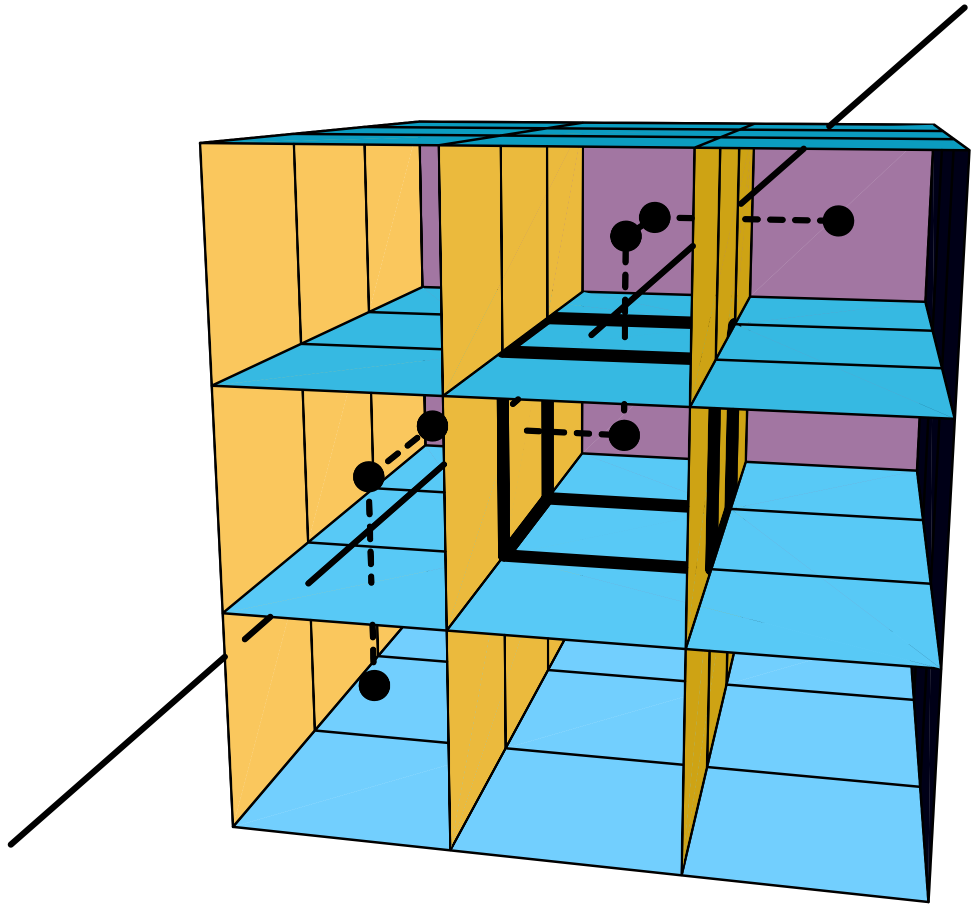

Figure 7: The Cube Quantization method for 3-D digital curve representation is a generalization of 2-D Square Quantization.

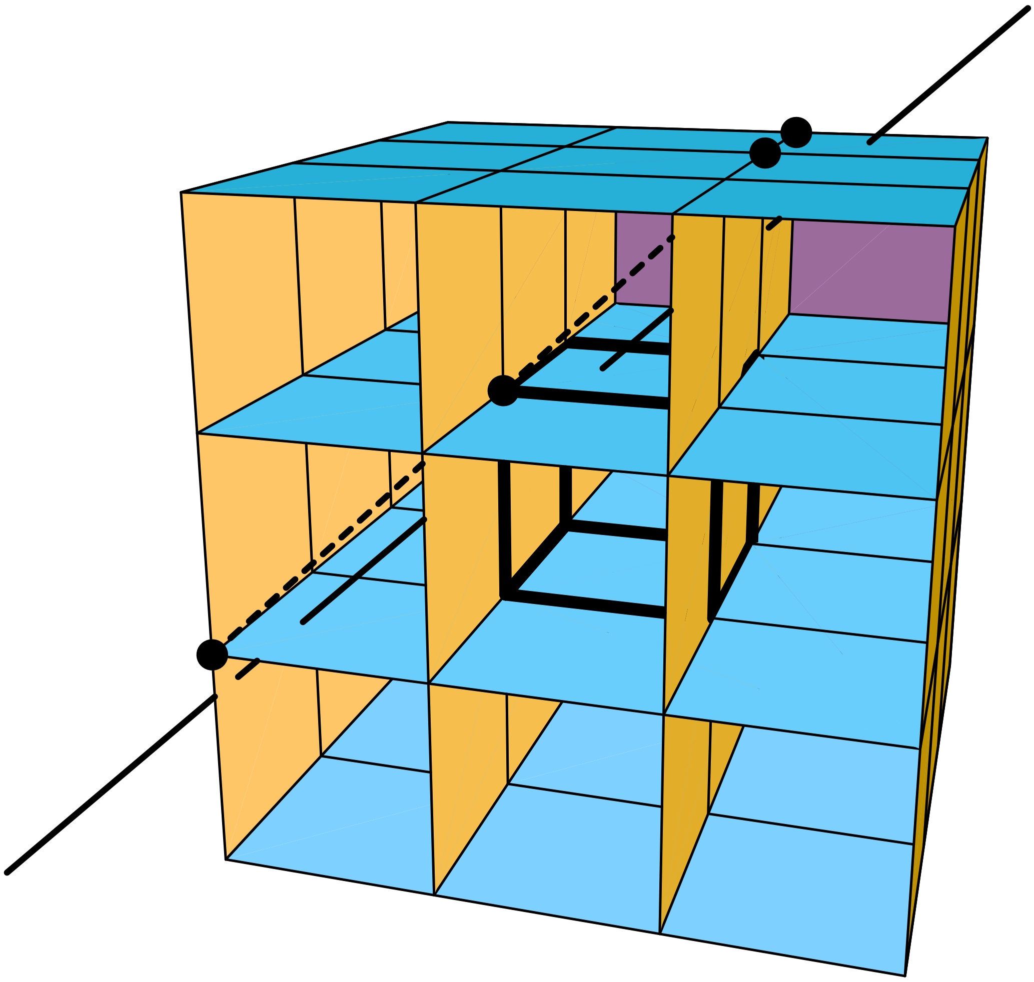

Figure 8: Three dimensional GIQ. At each intersection of the curve with the face of a shifted voxel, the nearest lattice point of the four corner points of the face is selected and included in the representation.

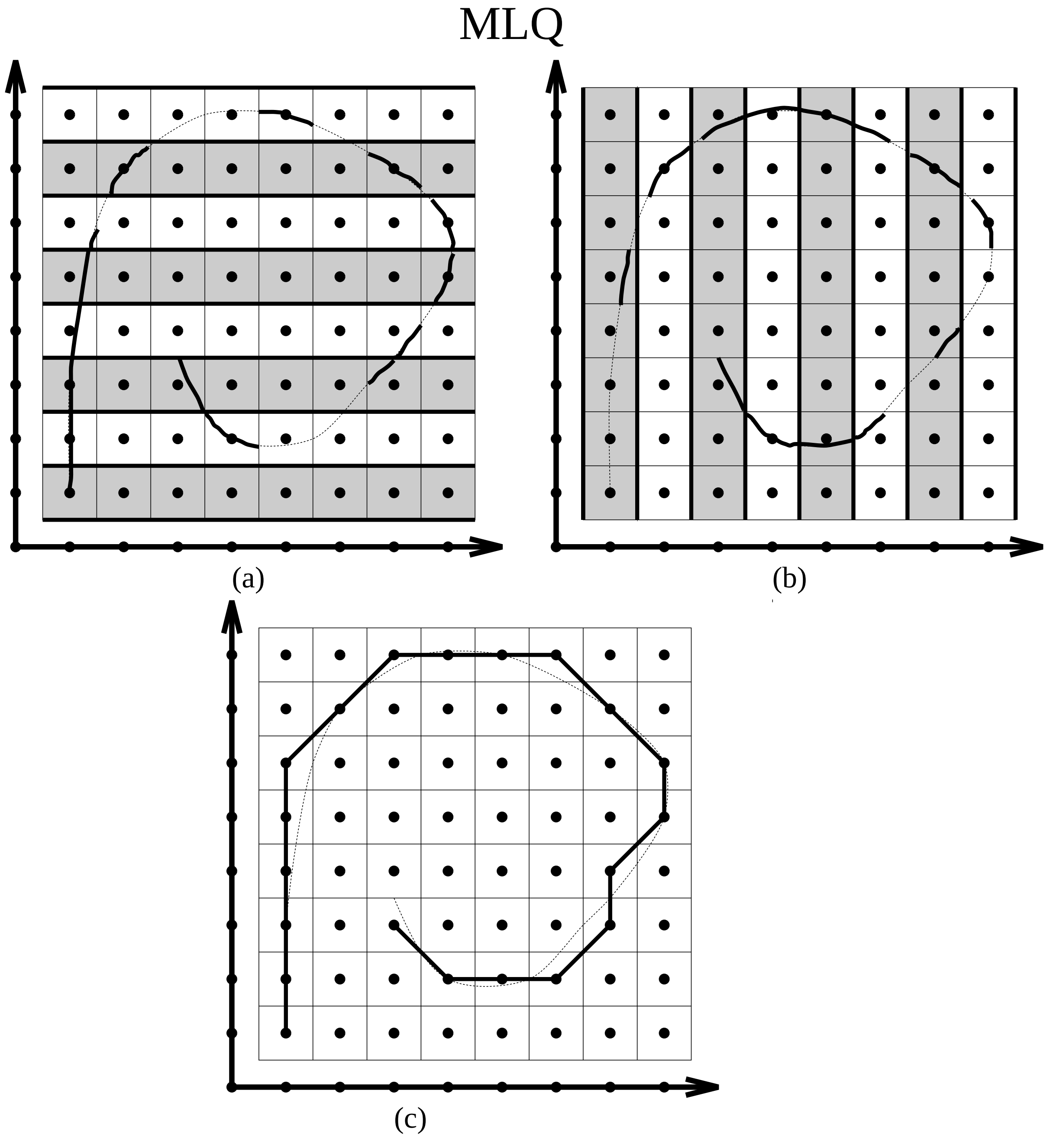

Figure 9: 2-D illustration of Maximum Length Quantization (MLQ). In (a) the curve is segmented according to its intersections with lines that are perpendicular to the Y axis and are midway between the integer coordinate lines. For each segment the lattice point with the longest subsegment in its centered pixel is selected. In (b) the process is repeated with respect to the X axis. The resulting chain is shown in (c).

A method used in [27] is referred to here as *Thinned Cube Quantization (TCQ)* (Fig. 10). The O(3)-chain obtained by this method is a subset of the O(1)-chain points obtained by Cube Quantization. The chain begins with the first Cube Quantization point and at each stage proceeds to the furthest possible point subject to maintaining O(3) connectivity, giving precedence to type-III links over type-II links and to type-II links over type-I links. This method minimizes the number of elements and, for a straight line, always yields a digital arc.

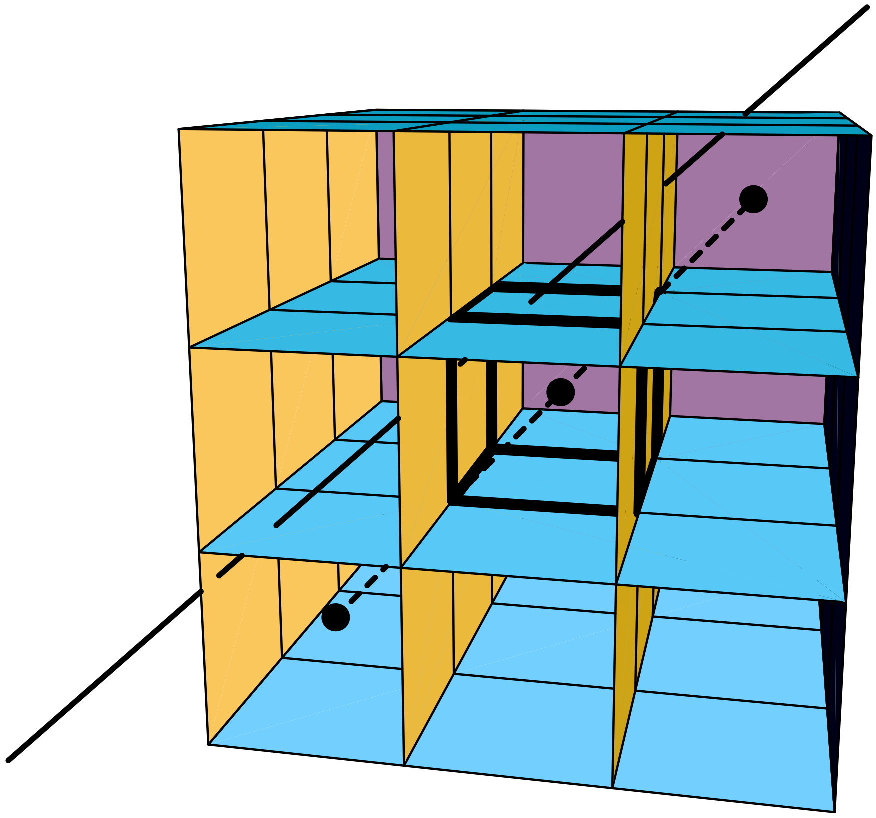

Figure 10: The Thinned Cube Quantization (TCQ) method for 3-D curve representation. The TCQ chain consists of a subset of the points in the Cube Quantization representation. The chain begins with the first Cube Quantization point and at each stage proceeds to the furthest possible point subject to maintaining O(3) connectivity, giving precedence to type-III links over type-II links and to type-II links over type-I links.

In [1, 4, 7, 53] a quantization scheme that is applicable only to straight lines has been considered. The axis with the smallest angle to the line is chosen to be the main axis, and the line is sampled at each integer value along that axis. The other coordinate values of the points are rounded [1, 53] or truncated [4, 7]. In [53] two extensions of these techniques to general curves are briefly suggested. One is based on approximation of the curve by straight line segments; this leads to large deviations from the original curve. The other uses the tangent to the curve at each coordinate crossing to locally define a main axis; chain connectivity in the general case is not ensured.

5 'Buyers Checklist' for Curve Representation Schemes

In this section the vague concept of a mathematically well-behaved 3-D curve representation scheme is formalized and summarized as a list of requirements. These are used in the sequel as guidelines for qualitative and quantitative comparison of the various methods that have been suggested.

Formally, Let $\bar{l}(t)$ be a (continuous) curve. The goal is to represent it by an O(r)-chain $\bar{L}_n$ in a square lattice in $\mathbf{R}^n$. Let Q denote the quantization operator, i.e., $\bar{L}_n = Q\bar{l}(t)$. Since Q transforms from a continuous space to a discrete one, it is a non-invertible operator, that is, an infinite number of continuous curves are generally represented by a given chain. Hence, $\bar{L}_n$ may be only an approximation of $\bar{l}(t)$. This approximation should preferably satisfy the following requirements.

(a) Coordinate system axis symmetry - The quantization should be symmetric with respect to changes in the labels and directions of the axes.

(b) Curve direction symmetry - Reversing the starting point and end point of the continuous curve should lead to the same chain in reversed order.

(c) Invariance to integer translation - Integer translation of the continuous curve (or equivalently of the coordinate system origin) should lead to the same chain translated.

(d) Finite memory - The representation of a curve segment should not depend on arbitrarily distant segments.

(e) Line to arc - The discrete representation of a straight line should be a digital arc. This requirement is necessary for minimization of the number of elements (subject to maintaining connectivity) in the discrete chain. Since the shortest chain between two lattice points is necessarily a digital arc, it is reasonable to require that a digital straight line should be a digital arc.

(f) The projection property - Let $R_{ij}x$ be the projection of a 3-D curve or chain on the ij plane, where i, j are any two coordinate axes. The projection property is that $QR_{ij}l = R_{ij}Ql$, i.e., the projection of a 3-D chain onto a plane perpendicular to any one of the coordinate system axes should be identical to the 2-D digital representation of the projection of the continuous curve onto that plane.

(g) Distance minimization - The process should minimize, in some sense, the distance between the discrete chain and the original continuous curve. There are obviously many ways to define the distance.

(h) Compactness - The chain should consist of as few elements as possible, subject to the other requirements. Alternatively, one can account for the different number of bits required for link representation in, say, 6-chains and 26-chains by requiring that the number of *bits* in the representation should be minimized.

These guidelines can be used in the selection or design of computational or mathematical quantization procedures. In physical quantization processes, such as in range-sensing or tomography, the application-specific physical constraints (that are often difficult to modify) will govern its mathematical characteristics.

6 Evaluation of 3-D Digital Curve Representation Schemes

In this section 3-D digital curve representation schemes are compared according to the list of requirements provided in the previous section. The methods considered are Grid Intersect Quantization (GIQ), Thinned Cube Quantization (TCQ), Maximum Length Quantization (MLQ) and Cube Quantization (CQ).

(a) Coordinate system axis symmetry is satisfied in all four 3-D digital curve representation schemes.

(b) Curve direction symmetry is satisfied in GIQ, MLQ and CQ, where the encoding is independent of ordering, but not in Thinned Cube Quantization (TCQ) in which the thinning of the cube quantization chain to obtain a 26-chain might be direction dependent as shown in Fig. 11.

(c) All four 3-D digital curve representation schemes are invariant to integer translation.

(d) Finite memory exists in GIQ, MLQ and CQ, but not in TCQ, as exemplified in Fig. 11.

(e) It is easy to show that in CQ the discrete representation of a straight line is a digital arc. This property holds in TCQ as well. Kim [24] and Klette [30] tried to prove that it also holds in 3-D GIQ, but Stojmenović and Tošić [53] provided a counter-example and have thus shown that the discrete representation of a straight line using 3-D GIQ is not necessarily a digital arc. To show that in MLQ the discrete representation of a straight line is not always a digital arc, consider the line

$$y = 0.9x + 0.4 \qquad z = 0.2x + 0.33$$

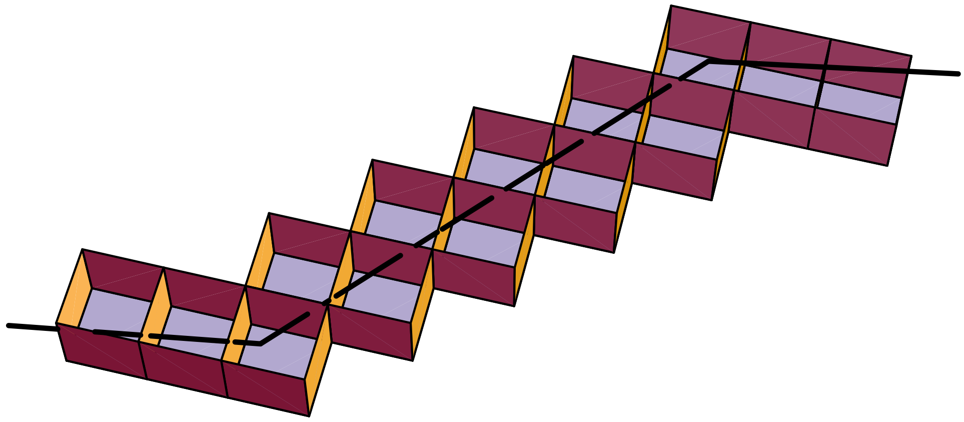

Figure 11: Violation of direction symmetry and finite memory in TCQ. The curve consists of three straight segments. Every other voxel that the middle segment traverses is included in the representation, but the selection depends on the direction of encoding, hence direction symmetry is violated. Furthermore, even if the middle segment had been made arbitrarily long, the selection would have still been influenced by the voxels that the first segment traverses.

The resulting MLQ chain contains the string

$$(0,0,0),(0,1,0),(1,1,1),(2,2,1)$$

The point $(1,1,1)$ has three neighbors, hence the chain is not a digital arc.

(f) Stojmenović and Tošić [53] have demonstrated that the projection property does not hold in 3-D GIQ. It can be further shown (see Appendix A) that any extension of 2-D GIQ (and 2-D OBQ) to three dimensions necessarily violates the projection property. TCQ and MLQ do not satisfy the projection property, see Fig. 12 and Fig. 13 respectively. In Cube Quantization (CQ), if a 3-D curve traverses a voxel then the 2-D projection of the curve traverses the 2-D projection of the voxel, and vice-versa. Thus, of the four methods only CQ satisfies the projection property.

Table 1 compares the four 3-D digital curve representation schemes according to requirements (a) through (f).

(g) One way to define the distance between the discrete chain and the original continuous curve is according to the *maximum distance between any chain lattice point and the curve.* An

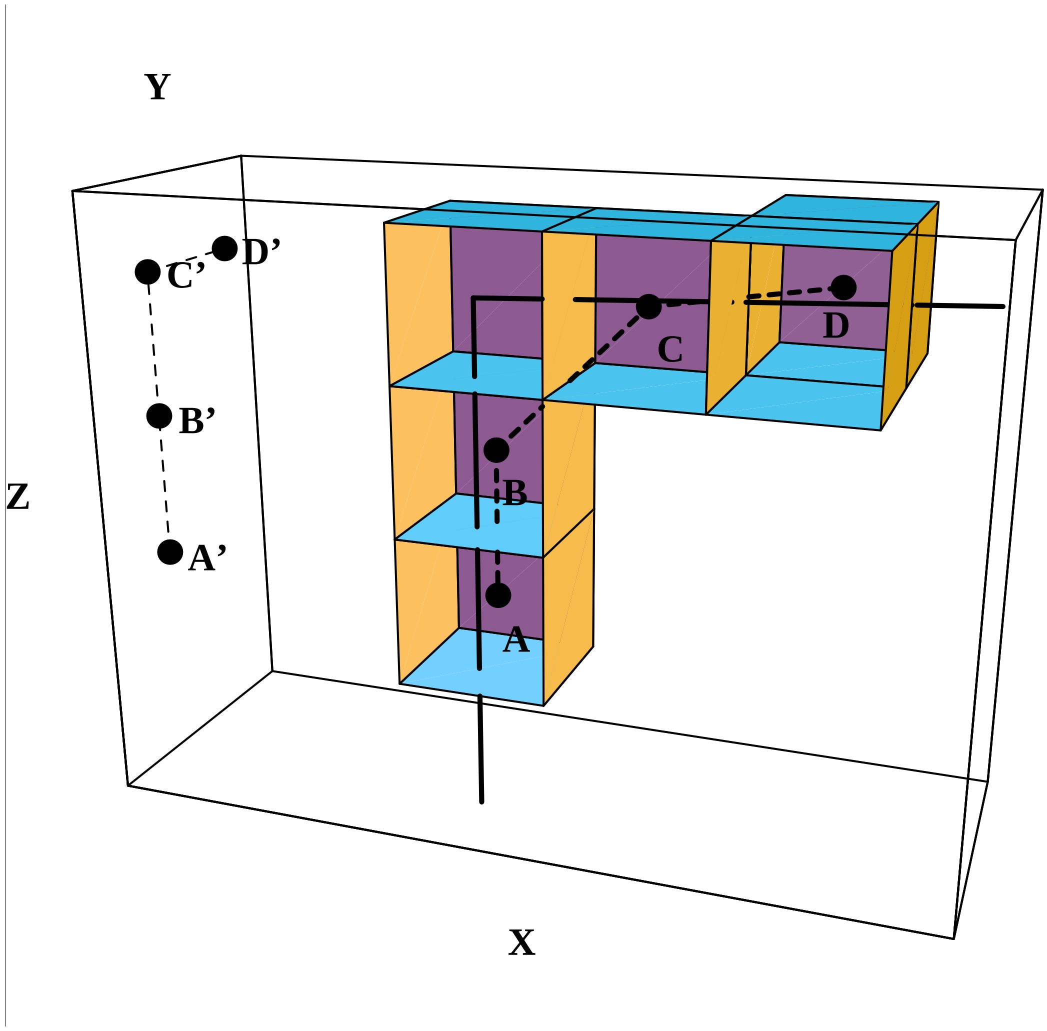

Figure 12: Violation of the projection property in TCQ. A 3-D curve (solid), its digital image, its TCQ representation and the projection of the chain onto the $Y-Z$ plane are shown. Point C is included in the 3-D TCQ chain but its projection C' will be thinned (deleted) in the 2-D TCQ representation of the projection of the curve.

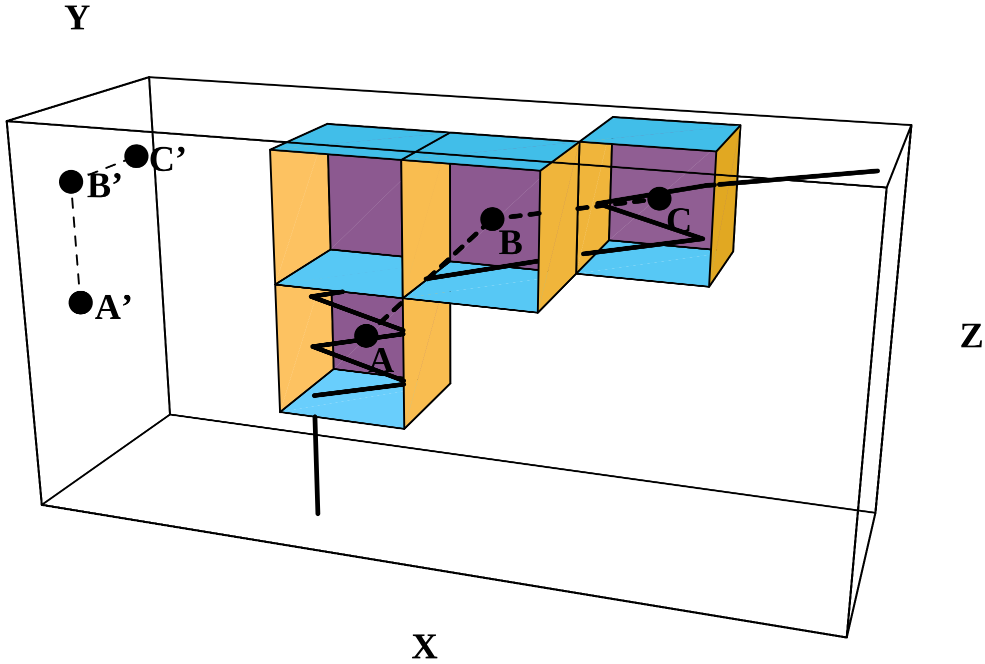

Figure 13: Violation of the projection property in MLQ. A 3-D curve (solid), its digital image, its MLQ representation and the projection of the chain onto the $Y-Z$ plane are shown. Point B is included in the 3-D MLQ chain but its projection B' is not included in the 2-D MLQ representation of the projection of the curve.

alternative definition is based on the *maximum distance between any curve point and the nearest chain lattice point*. In both approaches a definition of the distance between a given curve point and a given lattice point is needed. By specifying either the Euclidean or chess-board metrics, four alternative curve to chain distance measures are obtained. The four 3-D digital curve representation schemes were compared according to the four distance measures, and the results are summarized in Table 2.

(h) In order to compare the compactness of the four 3-D digital curve representations, the average number of chain elements per unit length of random straight lines is presented in the upper row of Table 3. The values for CQ and TCQ were taken from [27]. The average number of bits per unit length is shown in the bottom row of Table 3. It has been obtained by multiplying the average number of chain elements per unit length by $\log_2 6$ for CQ and by $\log_2 26$ for the other three schemes.

criterion	CQ	GIQ	MLQ	TCQ
a: coordinate system axes symmetry	+	+	+	+
b: curve direction symmetry	+	+	+	-
c: invariance to integer translation	+	+	+	+
d: finite memory	+	+	+	-
e: line to arc	+	-	-	+
f: the projection property	+	-	-	-

Table 1: Qualitative comparison of 3-D digital curve representation schemes.

measure	CQ	GIQ	MLQ	TCQ
A	0.866	0.707	0.866	0.866
B	0.5	0.5	0.5	0.5
C	0.866	1.732	1.66	1.66
D	0.5	1	1.5	1.5

Table 2: Comparison of the distance between the continuous curve and its discrete representation in four 3-D curve discretization schemes. *A:* Maximum Euclidean distance between any chain lattice point and the curve. Note that for GIQ the worst case is realized by a curve that passes through the center of a face of a voxel. For the other schemes the worst case is a curve that passes at the corner of a voxel. *B:* Maximum chess-board distance between any chain lattice point and the curve. *C:* Maximum Euclidean distance between any curve point and the nearest chain lattice point. *D:* Maximum chess-board distance between any curve point and the nearest chain lattice point.

measure	CQ	GIQ	MLQ	TCQ
A	1.5	0.900	0.835	0.831
B	3.88	4.23	3.92	3.91

Table 3: Compactness 3-D digital curve representations. *A:* Number of elements per unit length. *B:* Number of bits per unit length.

7 Discussion

More than thirty years of research have yielded a significant body of knowledge on digital representation schemes for planar curves. in 2-D space. This has been the foundation for the design and analysis of a multitude of shape analysis and pattern recognition algorithms. The prominent 2-D curve digitization techniques are generally 'well-behaved', and are in fact closely related among themselves.

As applications for 3-D shape analysis emerge, the need for digital representation schemes for 3-D curves arises. It is tempting to expect that 2-D schemes can be easily generalized to 3-D, but this is not the case: Several 2-D curve representation schemes cannot be extended to 3-D in a meaningful way, and others loose much of their elegance and mathematical robustness.

As a basis for comparison between digital representation schemes for 3-D curves a list of requirements has been presented. The suggested principles are quite general and intended to be relevant to many applications. They are certainly necessary for the partial extension of planar digital geometry theory to 3-D.

It turns out that Grid Intersect Quantization (GIQ) is a poor curve representation scheme in 3-D due to the lack of several important features. In fact, there is no 26-chain technique that we are aware of that can be recommended for general use. The main conclusion of this paper is that Cube Quantization (CQ), that leads to 6-chains, is the method of choice for 3-D curve representation. It meets all the identified requirements and provides superior coding efficiency in terms of bits per unit length.

Note that in certain applications, such as in the design of length estimators and chamfer Euclidean distance transformations, the natural classification of 26-chain links to three types is very convenient. Due to symmetry, there is no obvious "natural" classification of 6-chain links that result from Cube Quantization. However, simple and meaningful 6-chain link classification criteria have indeed been developed and applied to the design of extremely accurate length estimators for 3-D curves [21, 22].

Appendix A

We proceed to show that no digital 3-D curve representation scheme can satisfy the projection property if the respective 2-D scheme is GIQ or OBQ.

For GIQ consider the straight line

$$y = 0.25x + 0.45$$

$$z = 0.25x + 0.9$$

The following points belong to the 2-D GIQ representation of the projection of the line onto the $x - y$ plane in the vicinity of the origin:

$$(-1,0),(0,0),(1,1),(2,1)$$

Similarly for the $x - z$ plane:

$$(-2,0),(-1,1),(0,1),(1,1),(2,1)$$

and for the $y - z$ plane

$$(-1,-1),(0,0),(1,1),(2,2)$$

Assume that the projection property is satisfied. Then the point $(0,0)$ in the 2-D chain in the $x - y$ plane is the projection of a point in the 3-D chain whose coordinates are $(0,0,z')$. The projection of that point onto the $x - z$ plane is $(0,z')$ and by comparison to the list of points in that plane necessarily $z' = 1$. The projection of the point $(0,0,1)$ onto the $y - z$ plane is $(0,1)$, but that point is not included in the 2-D chain in the $y - z$ plane. Hence the projection property is not satisfied.

For OBQ consider the straight line

$$y = 0.25x + 0.95$$

$$z = 0.25x + 1.4$$

The digital representation of the projections of that line on the three planes is as above, hence the rest of the proof is similar.

Acknowledgments

This research was supported in part by the Israeli Ministry of Science and Arts, by the Fund for the Promotion of Research at the Technion and by the Ollendorff Center of the Electrical Engineering Department, Technion.

References

[1] T.M. Amarunnishad and P.P. Das, "Estimation of Length for Digitized Straight Lines in Three Dimensions", *Pattern Recognition Letters*, Vol. 11, pp. 207-213, 1990.

[2] T.A. Anderson and C.E. Kim, "Representation of Digital Line Segments and Their Preimages", *Comput. Vision Graphics Image Process.*, Vol. 30, pp. 279-288, 1985.

[3] J.E. Bresenham, "Algorithm for Computer Control of a Digital Plotter", *IBM Systems J.*, Vol. 4, pp. 25-30, 1965.

[4] A.L.D. Beckers and A.W.M. Smeulders, "Optimization of Length Measurements for Isotropic Distance Transformations in Three Dimensions", *CVGIP: Image Understanding*, Vol. 55, pp. 296-306, 1992.

[5] G. Borgefors, "Distance Transformations in Arbitrary Dimensions", *Comput. Vision Graphics Image Process.*, Vol. 27, pp. 321-345, 1984.

[6] A. M. Bruckstein, "Self-Similarity Properties of Digitized Straight Lines", *Contemporary Mathematics*, Vol. 119, pp. 1-20, 1991.

[7] S. Chattopadhyay and P.P. Das, "Estimation of the Original Length of a Straight Line Segment from its Digitization in Three Dimensions", *Pattern Recognition*, Vol. 25, pp. 787-798, 1992.

[8] P.P. Das, P.P. Chakrabarti and B.N. Chatterji, "Generalized Distances in Digital Geometry", *Inform. Sci.*, Vol. 42, pp. 51-67, 1987.

[9] L. Dorst and R.P.W Duin, "Spirograph Theory: A Framework for Calculations on Digitized Straight lines", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, pp. 632-639, Sep. 1984.

[10] L. Dorst and A.W.M. Smeulders, "Discrete Representation of Straight Lines", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, pp. 450-463, Jul. 1984.

[11] L. Dorst and A.W.M. Smeulders, "Length Estimators for Digitized Contours", *Computer Vision Graphics Image Processing*, Vol. 40, pp. 311-333, 1987.

[12] L. Dorst and A.W.M. Smeulders, "Discrete Straight Line Segments: Parameters, Primitives, and Properties", *Comtemporary Mathematics*, Vol. 119, pp. 45-62, 1991.

[13] H. Freeman, "Boundary Encoding and Processing", in B.S. Lipkin and A. Rosenfeld (eds.), *Picture Processing and Psychopictorics*, pp. 241-266, Academic Press, New York, 1970.

[14] H. Freeman, "Computer Processing of Line-Drawing Images", *Comput. Surveys*, Vol. 6, pp. 57-97, 1974.

[15] H. Freeman, "Application of the Generalized Chain Coding Scheme to Map Data Processing", in *Proc. IEEE Comput. Soc. Conf. Pattern Recognition and Image Process.*, pp. 220-226, Chicago, 1978.

[16] H. Freeman and J.M. Glass, "On the Quantization of Line-Drawing Data", *IEEE Trans. Sys. Sci. Cyber.*, Vol. SSC-5, pp. 70-79, 1969.

[17] H. Freeman and J.A. Saghri, "Comparative Analysis of Line-Drawing Modeling Schemes", *Computer Graphics Image Processing*, Vol. 12, pp. 203-223, 1980.

[18] F.C.A. Groen and P.W. Verbeek, "Freeman-Code Probabilities of Object Boundary Quantized Contours", *Computer Graphics Image Processing*, Vol. 7, pp. 391-402, 1978.

[19] S.H.Y. Hung, "On the Straightness of Digital Arcs", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-7, pp. 203-215, 1985.

[20] S.H.Y. Hung and T. Kasvand, "On the Chord Property and its Equivalences", *Proc. 7th International Conference on Pattern Recognition*, pp. 116-119, Montreal, 1984.

[21] A. Jonas and N. Kiryati, "Length Estimation in 3-D Using Cube Quantization", *Proc. SPIE Vision Geometry III*, Vol. 2356, pp. 220-230, Boston, 1994.

[22] A. Jonas, "Digital Representation and Length Estimation of Three Dimensional Curves", M.Sc. Thesis, Department of Electrical Engineering, Technion, May 1995.

[23] C.E. Kim, "On Cellular Straight Line Segments", *Computer Graphics Image Processing*, Vol. 18, pp. 369-381, 1982.

[24] C.E. Kim, "Three-Dimensional Digital Line Segments", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 231-234, 1983.

[25] C.E. Kim, "Three-Dimensional Digital Planes", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, pp. 639-645, 1984.

[26] R. Kimmel and N. Kiryati, "Finding Shortest Paths on Surfaces by Fast Global approximation and Precise Local Refinement", *Proc. SPIE Vision Geometry III*, Vol. 2356, pp. 198-209, Boston, 1994.

[27] N. Kiryati and O. Kübler, "On Chain Code Probabilities and Length Estimators for Digitized Three Dimensional Curves", *Pattern Recognition*, Vol. 28, pp. 361-372, 1995.

[28] R. Kimmel, N. Kiryati and A.M. Bruckstein, "Using Multi-Layer Distance Maps for Motion Planning on Surfaces with Moving Obstacles", *Proc. 12th International Conference on Pattern Recognition*, Vol. I, pp. 367-382, Jerusalem, 1994.

[29] R. Kimmel, N. Kiryati and A.M. Bruckstein, "Sub-pixel Distance Maps and Weighted Distance Transforms", *J. of Mathematical Imaging and Vision*, in press, 1995.

[30] R. Klette, "The m-Dimensional Grid Point Space", *Computer Vision Graphics Image Processing*, Vol. 30, pp. 1-12, 1985.

[31] T.Y. Kong and A. Rosenfeld, "Digital Topology: Introduction and Survey", *Comput. Vision Graphics Image Process.*, Vol. 48, pp. 357-393, 1989.

[32] J. Koplowitz, "On the Performance of Chain Codes for Quantization of Line Drawings", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-3, pp. 180-185, 1981.

[33] J. Koplowitz and A.M. Bruckstein, "Design of Perimeter Estimators for Digitized Planar Shapes", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-11, pp. 611-622, 1989.

[34] N. Kiryati and G. Szèkely, "Estimating Shortest Paths and Minimal Distances on Digitized Three Dimensional Surfaces", *Pattern Recognition*, Vol. 26, pp. 1623-1637, 1993.

[35] Z. Kulpa, "More About Areas and Perimeters of Quantized Objects", *Computer Vision Graphics Image Processing*, Vol. 22, pp. 268-276, 1983.

[36] C.G. Lekkerkerker, *Geometry of Numbers*, Wolters-Noordhoff, Amsterdam, 1969.

[37] M. Lindenbaum and J. Koplowitz, "A New Parametrization of Digital Straight Lines", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-13, pp. 847-852, 1991.

[38] X. Luo and L. Wu, "The Generalized Chord Property of Digital Plane Element", *Proc. 8th International Conference on Pattern Recognition*, pp. 1159-1161, Paris, 1986.

[39] M.D. McIlroy, "A Note on Discrete Representation of Lines", *AT&T Tech. J.*, Vol. 64, pp. 481-490 1984.

[40] U. Montanari, "On Limit properties in Digitization Schemes", *J. of the ACM*, Vol. 17, pp. 348-360, 1970.

[41] U. Montanari, "A Note on Minimal Length Polygonal Approximation to a Digitized Contour", *Comm. of the ACM*, Vol. 13, pp. 41-47, 1970.

[42] D.L. Neuhoff and K.G. Castor, "A Rate and Distortion Analysis of Chain Codes for Line Drawings", *IEEE Trans. Info. Th.*, Vol. IT-31, pp. 53-68, 1985.

[43] T. Pavlidis, *Algorithms for Graphics and Image Processing,* Springer-Verlag, Berlin-Heidelberg, 1982.

[44] D. Proffitt and D. Rosen, "Metrication Errors and Coding Efficiency of Chain-Encoding Schemes for the Representation of Lines and Edges", *Computer Graphics Image Processing*, Vol. 10, pp. 318-332, 1979.

[45] A. Rosenfeld, "Connectivity in Digital Pictures", *J. of the ACM*, Vol. 17, pp. 146-170, 1970.

[46] A. Rosenfeld, "Adjacency in Digital Pictures", *Inform. and Control*, Vol. 26, pp. 24-33, 1974.

[47] A. Rosenfeld, "Digital Straight Line Segments", *IEEE Trans. Comput.*, Vol. C-23, pp. 1264-1269, 1974.

[48] A. Rosenfeld, "Three-Dimensional Digital Topology", *Inform. and Control*, Vol. 50, pp. 119-127, 1981.

[49] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.

[50] A. Rosenfeld and J.L. Pfaltz, "Distance Functions on Digital pictures", *Pattern Recognition*, Vol. 1, pp. 33-61, 1968.

[51] J.A. Saghri and H. Freeman, "Analysis of the Precision of Generalized Chain Codes for the Representation of Planar Curves", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-3, pp. 533-539, 1981.

[52] D.K. Scholten and S.G. Wilson, "Chain Coding with a Hexagonal Lattice", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 526-533, 1983.

[53] I. Stojmenović and R. Tošić, "Digitization Schemes and the Recognition of Digital Straight Lines, Hyperplanes, and Flats in Arbitrary Dimensions", *Contemporary Mathematics*, Vol. 119, pp. 197-212, 1991.

[54] J.K. Udupa, S.N. Srihari and G.T. Herman, "Boundary Detection in Multidimensions", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-4, pp. 41-50, 1982.

[55] A.M. Vossepoel and A.W.M. Smeulders, "Vector Code Probability and Metrication Error in the Representation of Straight Lines of Finite Length", *Computer Graphics Image Processing*, Vol. 20, pp. 347-364, 1982.

[56] L.D. Wu, "On the Chain Code of a Line", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-4, pp. 347-353, 1982.