Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Line segment rasterization in n-dimensional space

C. Wuthrich

Computer Science/Department of Interactive Systems

CS-R9414 1994

Line segment rasterization in n-dimensional space

Charles A. Wüthrich
ERCIM
European Research Consortium in Informatics and Mathematics
Domaine de Voluceau
BP 105
78153 Le Chesnay Cedex, France

Abstract

The rapid development of scanning and measuring hardware for medical imaging and for scientific experiments, the introduction of animation techniques into common use have created the need to understand n-dimensional raster geometry, where $n > 3$. After a brief review of continuous geometry in $\mathbb{R}^n$, this paper presents the discrete regular structure called a *hyperlattice* used for defining n-dimensional raster geometry. The concept of neighbourhood in hyperlattices is studied in detail, introducing a new definition which extends existing definitions by taking into account the border of the Voronoi sets associated with neighbouring lattice points. A general definition of discrete curves based on these new definitions is then introduced. Finally a general definition of rasterization onto hyperlattices is given, and algorithms for rasterizing straight lines onto n-dimensional lattices are presented. A detailed analysis of line-drawing algorithms onto hyperlattices is then presented.

C.R. Subject Classification (1991): I.3.3 [Picture/Image Generation] Line and curve generation, I.3.1 [Hardware Architectures] Raster display devices, I.4.1 [Digitization] Sampling, I.4.1 [Digitization] Scanning.

Keywords and Phrases: N-dimensional geometry, hyperlattices, neighbourhood relations, n-dimensional rasterization, straight line segments.

1 Introduction

The rapid development of scanning and measuring hardware for medical imaging and for scientific experiments, the introduction of animation techniques into common use and the development of multimodal user interfaces have created new problems regarding sampling, representation and rendering of data defined in n-dimensional spaces for use in the discrete world representable in computers.

Note: This work has been supported by the *ERCIM* fellowship programme, during stays at *SERC, Rutherford Appleton Laboratory, Chilton, Didcot, Oxon OX11 0QX, England*, at *INESC (Instituto de Engenharia de Sistemas e Computadores), Rua Alves Redol 9, 1017 Lisboa, Portugal*, and at *CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands.*

While in the Computer Graphics literature there are abundant references to surface and volume modelling and rendering in three-dimensional space [4, 11, 30, 10, 14], and in Signal Processing there is a well established theory on the sampling of n-dimensional signals [15], references on curve and surface rasterization in general are rare, and only recently has a comprehensive definition of rasterization in two-dimensional spaces been introduced [25, 16, 41]. In brief, rasterization is the operation that governs the passing from the human continuous representation of the world to the discrete digital world of computing devices. Studies on rasterization in n-dimensional spaces, with $n \geq 3$, are even rarer [39, 24, 1, 16, 29, 2, 36] and are not organized systematically, and are thus insufficient for the formulation of a global theory that allows the understanding of the mechanisms and artifacts that affect rasterization in such highly complex spaces.

The main studies available to date for the treatment and visualization of n-dimensional spaces can be found in the field of statistical computing [37, 31, 9], but the visualization requirements for statistical analyses are relatively limited, and there has been no particular need to extend these early techniques to encompass today's sophisticated visualization tools.

Due to the recent progress in scientific visualization, there has been a renewed interest in new methods for the visualization of n-dimensional spaces [44, 5, 28, 38]. Also, problems arising from digital medical imaging posed researchers, such as G.T. Herman, new geometric problems in more general digital spaces, such as the definition of oriented surfaces [23]. However, these studies either are very practically oriented, or are limited to well-circumscribed issues of the theoretical problems involved. To avoid the development of mutually incompatible theories, a general framework for the development of the theory for n-dimensional rasterization is needed.

The present work tries to address exactly this issue. After a brief introduction to the geometry of n-dimensional spaces (Section 2), a model for n-dimensional discrete spaces based on *hyperlattices*, or *discrete lattices*, will be proposed. Discrete lattices are both sufficiently regular to be tractable and general enough to include a variety of discrete point dispositions in the real space $\mathbf{R}^n$ (Section 3). Briefly, a discrete lattice is the set of points that are the linear combination with integer coefficients of n linearly independent vectors called *generators* of the lattice. The relations between different lattice generator sets and the lattice generated will then be analyzed, and a necessary and sufficient condition for two lattice generator sets to generate the same lattice will be presented.

Discrete lattices provide sufficient generality to encapsulate both traditionally used structures, such as the hypercubic lattice, where the generators are mutually orthogonal and have the same length, and less conventional lattices based on non-orthogonal generators. Such an extension is needed for example to allow more freedom in the direction chosen for data scanning and display in devices such as Nuclear Magnetic Resonance (NMR) and Computer Assisted Tomography (CAT) scanners.

Section 4 will introduce $l_{\leq}$-neighbourhoods on hyperlattices: two points will be called neighbours if the corresponding Voronoi sets have in common a particular $(n-k)$-dimensional subset of the hyperplane which lies halfway between them, where k is an integer smaller than or equal to l. Hyperlattice neighbourhoods are an extension of the traditional neighbourhood concept in two- and three-dimensions, and allow to take into account different degrees of "diagonality" between neighbours. In the three-dimensional orthogonal case, this is equivalent to making a distinction between point-, line- and face-connected neighbours. The number of available neighbours depends on the mutual orthogonality of the lattice generators, and a necessary condition for two lattice points to be neighbours is presented. Curves on hyperlattices are finally defined in the same way as paths on graphs as sequences of neighbouring lattice points, thus binding together for the first time lattice connectedness to the definition of curve.

Section 5 introduces different rasterization methods for hyperlattices. First, the straightforward extension of two-dimensional nearest neighbour rasterization methods is presented: grid intersection and cellular rasterization methods are indicated as effective methods to generate respectively $n_{\leq}$- and $1_{\leq}$-connected rasterizations onto the n-dimensional space. A generic algorithm for obtaining the rasterization of a curve into an l-connected path is then presented. Finally, an algorithm that generates the rasterization of a straight line segment onto an n-dimensional $l_{\leq}$-connected lattice is derived in Section 6.

The inherent difficulty of the matter involved in this paper may be off-putting for some readers. Full proofs have been provided to aid understanding of the methods employed for reasoning, and this might result in a slight indigestibility of some parts of the paper. However, in this introduction we have tried to provide the reader with a roadmap through the perils of reasoning in n-dimensional spaces.

2 Basic geometry in n-dimensional space

Before introducing the operation of rasterization in n-dimensional spaces, it is sensible to recall some of the geometry of the Euclidean n-dimensional space $\mathbf{R}^n$, so that the reader can later agree on simple geometric concepts such as planes, straight lines and perpendicularity. Readers interested in the general definitions of the quantities introduced hereafter are advised to consult basic mathematics text books such as [13, 35, 21]. We shall partially follow here the definitions used in [7], which trade off generality for ease of understanding by stressing the geometric interpretation.

In the Euclidean real n-dimensional space $\mathbf{R}^n$, the distance function commonly used is the *Euclidean distance.* Given two points $\mathbf{x} = (x_0, \ldots, x_{n-1})$ and $\mathbf{y} = (y_0, \ldots, y_{n-1})$ of $\mathbf{R}^n$, their distance is the real positive number $d(\mathbf{x}, \mathbf{y}) = (\sum_{i=0}^{n-1} |x_i - y_i|^2)^{\frac{1}{2}}$. The *norm* of a vector $\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbf{R}^n$ is the real number $\|\mathbf{x}\| = (\sum_{i=0}^{n-1} |x_i|^2)^{\frac{1}{2}}$. The *scalar product* or *inner product* $\langle \mathbf{x}, \mathbf{y} \rangle$ of two vectors $\mathbf{x} = (x_0, \ldots, x_{n-1})$ and $\mathbf{y} = (y_0, \ldots, y_{n-1})$ of $\mathbf{R}^n$ is defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=0}^{n-1} x_i y_i \; .$$

Note that $\langle \mathbf{x}, \mathbf{x} \rangle = \|\mathbf{x}\|^2$ and that $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$. Moreover, in the Euclidean n-dimensional space both the *triangle inequality*

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \; ,$$

and the *cosine rule*

$$\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \cdot \|\mathbf{y}\| \cos \alpha \; ,$$

hold, where α represents the angle formed by the vectors $\mathbf{x}$ and $\mathbf{y}$ in the origin. From a geometric point of view, the importance of the definition of inner products lies in the fact that they are used to define orthogonality: two non-zero vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be *orthogonal* or *perpendicular* if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. If $\mathbf{x}$ and $\mathbf{y}$ are orthogonal we shall write $\mathbf{x} \perp \mathbf{y}$.

A *unit vector* $\mathbf{u} = (u_0, \ldots, u_{n-1})$ in $\mathbf{R}^n$ is a vector such that $\|\mathbf{u}\| = 1$. A unit vector uniquely determines a direction in the n-dimensional space: directions are therefore characterized by the corresponding unit vector n-tuple $(u_0, \ldots, u_{n-1})$. Two directions $\mathbf{u}$ and $\mathbf{u}'$ are said to be opposite if $\mathbf{u}' = -\mathbf{u}$. The *direction* associated to a non-zero vector $\mathbf{x} = (x_0, \ldots, x_{n-1})$ is the direction expressed by the corresponding unit vector $\mathbf{u}_\mathbf{x} = (\frac{x_0}{\|\mathbf{x}\|}, \ldots, \frac{x_{n-1}}{\|\mathbf{x}\|})$. Two vectors $\mathbf{x} = (x_0, \ldots, x_{n-1})$ and $\mathbf{y} = (y_0, \ldots, y_{n-1})$ are said to be *parallel* if the directions associated to them are equal or opposite. Note that this implies that there exists a constant $k \in \mathbf{R} \setminus \{0\}$ such that $\forall i,\ x_i = k y_i$.

Given two vectors ξ and $\mathbf{x}$ of $\mathbf{R}^n$, with $\xi = (\xi_0, \ldots, \xi_{n-1})$ and $\mathbf{x} = (x_0, \ldots, x_{n-1}) \neq 0$, the set

$$r : \{\xi + t\mathbf{x},\ t \in \mathbf{R}\}$$

is called the *straight line* passing through ξ in the direction $\mathbf{u}_\mathbf{x}$. Given any two distinct points $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$, the straight line passing through them is the set $r_{\mathbf{xy}} : \{\mathbf{x} + t(\mathbf{y} - \mathbf{x}),\ t \in \mathbf{R}\}$. If the parameter t is restricted to varying in the real interval $[0, 1]$, then $r_{\mathbf{xy}}$ contains only the points of the *straight line segment* joining $\mathbf{x}$ and $\mathbf{y}$.

Given ξ and $\mathbf{v} \neq 0$ in $\mathbf{R}^n$, the *hyperplane* passing through ξ and orthogonal to $\mathbf{v}$ is the set

$$H = \{\mathbf{x} \in \mathbf{R}^n \ : \ \langle \mathbf{x} - \xi, \mathbf{v} \rangle = 0\}. \tag{1}$$

The unit vector $\mathbf{u}_\mathbf{v}$ associated with $\mathbf{v}$ is called the *normal* of the hyperplane H. A hyperplane H is uniquely determined by assigning the direction $\mathbf{u}_\mathbf{v}$ of its normal and a point ξ of the space. Note that a hyperplane in the n-dimensional space can also be defined as the locus of points $\mathbf{x} = (x_0, \ldots, x_{n-1})$ satisfying the linear equation

$$c_0 x_0 + \ldots + c_{n-1} x_{n-1} = c\,, \tag{2}$$

where, of course, not all the coefficients $c_i \in \mathbf{R}$ are simultaneously zero. The two definitions proposed above are equivalent: consider the equation $\langle \mathbf{x} - \xi, \mathbf{v} \rangle = 0$. This means that $(x_0 - \xi_0)v_0 + \ldots + (x_{n-1} - \xi_{n-1})v_{n-1} = 0$, where the x_i are variables. Thus we have $v_0 x_0 + \ldots + v_{n-1} x_{n-1} = c$, where $c = \langle \xi, \mathbf{v} \rangle$. Conversely, suppose that the vector $\xi = (\xi_0, \ldots, \xi_{n-1})$ satisfies Equation 2. Let $\mathbf{v} = (c_0, \ldots, c_{n-1})$. Since not all the c_i are zero, $\mathbf{v} \neq 0$. Let $\mathbf{x} = (x_0, \ldots, x_{n-1})$ be a vector satisfying Equation 2. It is trivial to show that $\mathbf{x}$ also satisfies the equation $\langle \mathbf{x} - \xi, \mathbf{v} \rangle = 0$.

All hyperplanes in $\mathbf{R}^n$ are isomorphic to $\mathbf{R}^{n-1}$. Note that in $\mathbf{R}^3$ a hyperplane is a plane, and in $\mathbf{R}^2$ it is a straight line. Two hyperplanes H and H' are said to be parallel if their normals are parallel. Two parallel hyperplanes have no points in common.

The set of points $\mathbf{x} = (x_0, \ldots, x_{n-1})$ of $\mathbf{R}^n$ the coordinates of which satisfy the inequality $c_0 x_0 + \ldots + c_{n-1} x_{n-1} > c$ is called a *semispace* in the n-dimensional space: its border set is the hyperplane $c_0 x_0 + \ldots + c_{n-1} x_{n-1} = c$. A hyperplane subdivides the n-dimensional space into two semispaces.

Let $\mathbf{v} = v_0, \ldots, v_{n-1}$ be a point of $\mathbf{R}^n$, and let H be the hyperplane $c_0 x_0 + \ldots + c_{n-1} x_{n-1} - c = 0$. The real number $H(\mathbf{v}) = \sum_{i=0}^{n-1} c_i v_i$ is called the *evaluation* of the hyperplane H in $\mathbf{v}$. Let $\mathbf{v}$ and $\mathbf{w}$ be two points of $\mathbf{R}^n$ which do not belong to H. From the definition of the evaluation of a hyperplane in a point, the following trivial remark can be deduced:

Remark 2.1 *If $H(\mathbf{v}) > 0$ and $H(\mathbf{w}) > 0$, the segment joining $\mathbf{v}$ and $\mathbf{w}$ does not intersect the hyperplane H, and $\mathbf{v}$ and $\mathbf{w}$ belong to the same semispace defined by H in $\mathbf{R}^n$.*

Consider the two distinct hyperplanes H_0, of equation $h_{0,0}x_0 + \ldots + h_{0,n-1}x_{n-1} - c_0 = 0$, and H_1, of equation $h_{1,0}x_0 + \ldots + h_{1,n-1}x_{n-1} - c_1 = 0$. If there exist two non-zero real numbers μ and ν such that for all values of the variables x_i we have

$$\mu(h_{0,0}x_0 + \ldots + h_{0,n-1}x_{n-1} - c_0) + \nu(h_{1,0}x_0 + \ldots + h_{1,n-1}x_{n-1} - c_1) = 0 \,,$$

H_0 and H_1 are said to be *linearly dependent*, otherwise they are said to be *linearly independent.*

Let H_0 and H_1 be two linearly independent hyperplanes. The hyperplanes of equation

$$\mu(h_{0,0}x_0 + \ldots + h_{0,n-1}x_{n-1} - c_0) + \nu(h_{1,0}x_0 + \ldots + h_{1,n-1}x_{n-1} - c_1) = 0$$

are said to be the *linear combinations* of H_0 and H_1.

Consider n distinct hyperplanes $H_0, \ldots, H_{n-1}$, and let $c_{i0}x_0 + \ldots + c_{in}x_n$ be the equation of H_i. The intersection

$$\bigcap_{i=1}^{n} H_i$$

is called a *flat* or *affine set* of $\mathbf{R}^n$.

Remark 2.2 *Let $H_0 : c_{00}x_0 + \ldots + c_{0n}x_n = 0$ and $H_1 : c_{10}x_0 + \ldots + c_{1n}x_n = 0$ be two non-parallel and linearly independent hyperplanes in the n-dimensional space. The set $H_0 \cap H_1$ is isomorphic to a hyperplane in the $(n-1)$-dimensional space.*

Proof. Since the hyperplanes are not parallel they must have at least one point in common. The coordinates of such a point must satisfy the system of equations

$$\begin{cases} c_{00}x_0 + \ldots + c_{0n}x_n &= c_0 \\ c_{10}x_0 + \ldots + c_{1n}x_n &= c_1 \,. \end{cases}$$

Let c_{0m} be the first of the $c_{0i} \neq 0$. Solving the first equation of the linear system with respect to x_m and substituting in the second one yelds a linear equation in $(n-1)$ variables, which represents a hyperplane in the $(n-1)$-dimensional space. $H_0 \cap H_1$ is thus isomorphic to this hyperplane. □

Similarly, let

$$\begin{array}{llcl} H_0 : & c_{00}x_0 + \ldots + c_{0n}x_n & = & c_0 \\ \vdots & \vdots \qquad\qquad\quad \vdots & & \vdots \\ H_k : & c_{k0}x_0 + \ldots + c_{kn}x_n & = & c_k \end{array}$$

be $k+1$ linearly independent hyperplanes in the n-dimensional space. By definition the set $F = H_0 \cap \ldots \cap H_k$ is a flat in the $(n-1)$-dimensional space; its points are the points $(x_0, \ldots, x_{n-1})$ satisfying the system

$$\begin{cases} c_{00}x_0 + \ldots + c_{0n}x_n &= c_0 \\ \vdots \qquad\qquad\quad \vdots &= \vdots \\ c_{k0}x_0 + \ldots + c_{kn}x_n &= c_k \,. \end{cases} \tag{3}$$

If the hyperplanes are not parallel nor linearly dependent, then the system above can be solved with respect to k variables, and the results can be substituted in the remaining equations leading to one linear equation in $n-k$ variables. The locus of points in question is therefore isomorphic to the $(n-k)$-dimensional space. If even only two of the hyperplanes are parallel, then system (3) has no solution, and the intersection flat is the empty set.

A flat included in the n-dimensional space and isomorophic to the k-dimensional space $(0 \leq k < n)$ will be called a k-dimensional flat. A flat isomorphic to $\mathbf{R}^2$ will be called a *plane*.

Note that all linear combinations of two linearly independent hyperplanes H_0 and H_1 share the $(n-2)$-dimensional hyperplane $H_0 \cap H_1$. Thus if three hyperplanes are linearly dependent, their intersection is still an $(n-2)$-dimensional hyperplane.

Finally, let us introduce curves, surfaces in $\mathbf{R}^n$, polygons and hyperpolygons. A *curve* γ in the n-dimensional space is a continuous function

$$\gamma : [0,1] \subset \mathbf{R} \longrightarrow \mathbf{R}^n .$$

For example, straight lines and segments are curves in the n-dimensional space. Note that any curve γ can always be seen as a subset of $\mathbf{R}^{n+1}$. A *surface* in the n-dimensional space is a continuous function

$$f : [0,1] \times [0,1] \subset \mathbf{R}^2 \longrightarrow \mathbf{R}^n .$$

Planes are surfaces in the n-dimensional space; finite subsets of planes delimited by a finite number of straight lines are called *polygons*. The definition of a *hyperpolygon* has to be introduced recursively. A 1-dimensional hyperpolygon is a segment. A 2-dimensional hyperpolygon is a polygon. A k-dimensional hyperpolygon is a portion of the k-dimensional space the borders of which are a finite number of $(k-1)$-dimensional hyperpolygons.

The building blocks presented in this section provide the definitions for the basic geometric entities that we shall use in this paper. In order to be able to introduce n-dimensional raster geometry, the definition of the underlying raster space has to be provided. The next section will define the discrete geometric structures onto which the real continuous world will be mapped.

3 Simple properties of n-dimensional lattices

In this section we present the regular subspace of $\mathbf{R}^n$ that will be used to define rasterization. The notation used here is based on the notation used in [15, 41].

Let $\mathbf{R}$ be the set of real numbers, and let $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$ be n linearly independent vectors of $\mathbf{R}^n$. The set of all linear combinations of these vectors with integer coefficients, i.e. the set

$$\begin{aligned} \Lambda &= \Lambda(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) = \\ & \{\mathbf{v} = n_0\mathbf{v}_0 + n_1\mathbf{v}_1 + \ldots + n_{n-1}\mathbf{v}_{n-1} \mid n_i \in \mathbf{Z}\, ,\ (i = 0, \ldots, n-1)\} \end{aligned}$$

is called a *lattice* in $\mathbf{R}^n$, and the vectors $\mathbf{v}_i$ are called lattice *generators* or *basis vectors* of Λ. Given a lattice Λ, each of its points can be uniquely identified by the ordered n-uple of integer coordinates $(n_0, n_1, \ldots, n_{n-1})$. Such coordinates are called *lattice coordinates* of the point. Note that all lattices include the origin. Whenever the circumstances will allow us to do so, and no confusion can be generated, we shall indicate the points of a lattice with their n-tuple.

The most common lattice example is $\mathbf{Z}^n$, the basis vectors of which are the unit vectors of $\mathbf{R}^n$ and which we shall call the *hypercubic lattice.* Other interesting lattices are the *hyperrectangular lattice*, the generators of which are the vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$, where all the components of $\mathbf{v}_i$ are equal to zero except the i-th component. Notice that different sets of generators may generate the same lattice. For example, in $\mathbf{R}^2$, $\Lambda((1,0),(0,1)) = \Lambda((1,0),(1,1))$.

For the purpose of n-dimensional raster device modeling, we shall restrict the discrete sets under consideration to n-dimensional lattices, since they are sufficient for our modeling purpose, as we shall see later on, in section 5. Because of the importance of lattices, the rest of the current section will be dedicated to the analysis of the correlation between generators and generated lattices.

Let $\Lambda_v = \Lambda(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1})$ and $\Lambda_w = \Lambda(\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1})$. The following lemma is true:

Lemma 3.1 $\Lambda_w \subseteq \Lambda_v \iff \forall i,\ \mathbf{w}_i \in \Lambda(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1})$.

Proof. $\Longrightarrow$ Since Λ_w is included in Λ_v, all of its points are expressible as linear combinations with integer coefficients of the generators of Λ_v, thus the thesis is true.

$\Longleftarrow$ The hypothesis implies that $\forall i,\ \mathbf{w}_i \in \Lambda(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1})$. Thus $\mathbf{w}_i = n_{i,0}\mathbf{v}_0 + \ldots + n_{i,n-1}\mathbf{v}_{n-1} = \sum_{j=0}^{n-1} n_{i,j}\mathbf{v}_j$.

Let $\mathbf{w} \in \Lambda_w$. This means that there exist $m_0, \ldots, m_{n-1} \in \mathbf{Z}$ such that $\mathbf{w} = m_0\mathbf{w}_0 + \ldots + m_{n-1}\mathbf{w}_{n-1} = \sum_{i=0}^{n-1} m_i\mathbf{w}_i$. Substituting for the $\mathbf{w}_i$ and grouping similar terms leads to

$$\mathbf{w} = \sum_{i=0}^{n-1} m_i(\sum_{j=0}^{n-1} n_{i,j}\mathbf{v}_j) = \sum_{j=0}^{n-1}\sum_{i=0}^{n-1} m_i n_{i,j}\mathbf{v}_j \ ,$$

and thus $\mathbf{w} \in \Lambda_v$. Since this is valid $\forall \mathbf{w} \in \Lambda_w$, $\Lambda_w \subseteq \Lambda_v$. □

Let $\Lambda_w \subseteq \Lambda_v$. Thus

$$\begin{array}{lllllllll} \mathbf{w}_0 & = & n_{0,0}\mathbf{v}_0 & + & n_{0,1}\mathbf{v}_1 & + & \ldots & + & n_{0,n-1}\mathbf{v}_{n-1} \\ \mathbf{w}_1 & = & n_{1,0}\mathbf{v}_0 & + & n_{1,1}\mathbf{v}_1 & + & \ldots & + & n_{1,n-1}\mathbf{v}_{n-1} \\ \vdots & & \vdots & & \vdots & & \ddots & & \vdots \\ \mathbf{w}_{n-1} & = & n_{n-1,0}\mathbf{v}_0 & + & n_{n-1,1}\mathbf{v}_1 & + & \ldots & + & n_{n-1,n-1}\mathbf{v}_{n-1} \ , \end{array} \tag{4}$$

where $n_{i,j} \in \mathbf{Z}$ for all i,j. Equation 4 can be rewritten in a more concise form by putting $\mathbf{W} = (\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1})$ and $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1})$, and by using matrix notation[1]. This leads to

$$\mathbf{W} = N\mathbf{V} \ ,$$

where of course $N = [n_{i,j}]_{i,j=0}^{n-1}$. Let $N_{i,j}$ be the cofactor of the term $n_{i,j}$, i.e. the $(n-1)\times(n-1)$ matrix formed by deleting row i and column j from the matrix N, and let C_N be the matrix formed by the determinants of the cofactors of the elements of N, *i.e.* let

$$C_N = [c_{i,j}] = \Big[(-1)^{i+j}\det(N_{i,j})\Big]_{i,j=0,\ldots,n-1} \ . \tag{5}$$

Note that if the elements of N are integers, the elements of C_N are also integers, since the determinant of a matrix is obtained by performing only multiplications and additions. The following theorem holds:

[1] Note that here the components of $\mathbf{W}$ and $\mathbf{V}$ are n-dimensional vectors

Theorem 3.1 $\Lambda_w = \Lambda_v \iff$ *the elements of* C_N *are divisible by* $\det(N)$.

Proof. The hypothesis $\det(N) \neq 0$ is not necessary, because both the $\mathbf{v}_i$ and the $\mathbf{w}_i$ are linearly independent, thus N must be non-singular. Since by hypothesis $\Lambda_w \subseteq \Lambda_v$, we already know that $\exists N$ such that $\mathbf{W} = N\mathbf{V}$.

$\Longrightarrow$ This is straightforward, since $\Lambda_w = \Lambda_v$ *iff* $(\Lambda_w \subseteq \Lambda_v) \wedge (\Lambda_w \supseteq \Lambda_v)$. But this means that

$$\begin{aligned} \mathbf{W} &= N\mathbf{V} \\ \mathbf{V} &= M\mathbf{W} \end{aligned}$$

for some $n \times n$ integer matrices N and M. Thus, $M = N^{-1}$. But then the elements of N^{-1} are integers. Linear algebra references, such as [20], show that the elements $n'_{i,j}$ of N^{-1} are the elements of the transposed matrix C_N^T of the matrix C_N divided by the determinant of N, i.e.

$$n'_{i,j} = \frac{c_{j,i}}{\det(N)} \ (i, j = 0, \ldots, n-1) \ . \tag{6}$$

Since the $n'_{i,j}$ are integers, each $c_{i,j}$ must be divisible by $\det(N)$.

$\Longleftarrow$ To prove that $\Lambda_w = \Lambda_v$, we must prove that $\Lambda_w \supseteq \Lambda_v$, which by Lemma 3.1 is equivalent to prove that $\forall j$, $\mathbf{v}_j \in \Lambda(\mathbf{w}_0, \ldots, \mathbf{w}_{n-1})$. By hypothesis, the elements of C_N are divisible by $\det(N)$, thus the elements of N^{-1} are integers. Also by hypothesis, $\Lambda_w \subseteq \Lambda_v$, which means that there exists an integer matrix N such that

$$\mathbf{W} = N\mathbf{V} \ . \tag{7}$$

We also know that N^{-1} exists and that its elements are integers. Multiplying 7 on the left by N^{-1}, we obtain

$$\mathbf{V} = N^{-1}\mathbf{W} \ . \tag{8}$$

This means that all the generators of Λ_v are linear combinations with integer coefficients of the generators of Λ_w, and that therefore $\Lambda_w \supseteq \Lambda_v$. □

Corollary 3.1 *Let* $\Lambda = \Lambda(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1})$, *let* $w_j = v_j \ \forall j \neq i$, *and let*

$$\mathbf{w}_i = \mathbf{v}_i + \sum_{j \neq i} c_j \mathbf{v}_j \ ,$$

where $c_j \in \mathbf{Z} \ \forall j$. *Then*

$$\Lambda(\mathbf{v}_0, \ldots, \mathbf{v}_{n-1}) = \Lambda(\mathbf{w}_0, \ldots, \mathbf{w}_{n-1})$$

Proof. Since the $\mathbf{w}_i$ are linear combinations of the $\mathbf{v}_i$, we already know that $\Lambda_w \subseteq \Lambda_v$. From Theorem 3.1 we know that the thesis is true if and only if $\mathbf{V} = M\mathbf{W}$ for some integer matrix

M. The definition of the $\mathbf{w}_j$ says that

$$\begin{array}{lcl} \mathbf{w}_0 & = & \mathbf{v}_0 \\ \mathbf{w}_1 & = & \mathbf{v}_1 \\ \vdots & & \vdots \\ \mathbf{w}_{i-1} & = & \mathbf{v}_{i-1} \\ \mathbf{w}_i & = & \sum_{j=0}^{i-1} c_j \mathbf{v}_j + \mathbf{v}_i + \sum_{j=i+1}^{n-1} c_j \mathbf{v}_j \\ \mathbf{w}_{i+1} & = & \mathbf{v}_{i+1} \\ \vdots & & \vdots \\ \mathbf{w}_{n-1} & = & \mathbf{v}_{n-1}\ . \end{array} \tag{9}$$

This means that

$$\mathbf{W} = N\mathbf{V}\ , \tag{10}$$

where

$$N = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ c_0 & c_1 & \dots & c_{i-1} & 1 & c_{i+1} & \dots & c_{n-1} \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{bmatrix}, \tag{11}$$

and the row of N containing the c_j coefficients is the i-th row. Multiplying the left hand side of equation 10 by N^{-1}, we obtain

$$N^{-1}\mathbf{W} = \mathbf{V}\ . \tag{12}$$

Consider N^{-1}. The determinant of the matrix N is equal to the determinant of the matrix obtained from N by multiplying one of its rows by any real coefficient and by subtracting the resulting row from any other row of the matrix [20]. If in N we subtract from the i-th line of c_0 times the first row, c_1 times the second row and so on (excluding the i-th row), we obtain the unit matrix, the determinant of which is 1. But then N^{-1} is an integer matrix, since its terms can be computed through Equation 5, and the denominator in their expression equals 1. All $\mathbf{v}_j$ can therefore be expressed as linear combinations with integer coefficients of the $\mathbf{w}_i$, and are therefore elements of Λ_w. The thesis is therefore true. □

Note that the lattices generated also coincide if the $\mathbf{w}_i$ are defined as follows

$$\mathbf{w}_i = \sum_{j=0}^{i-1} c_j \mathbf{v}_j + \mathbf{v}_i\ , \tag{13}$$

since again the determinant of the matrix N equals one.

The theorems introduced above point out that more than one set of generators can generate the same lattice. For instance, in the example in Figure 1, corollary 3.1 guarantees that if v_0 is chosen as first lattice generator, then the pairs (v_0, v_1), (v_0, v_1') and (v_0, v_1'') generate the same

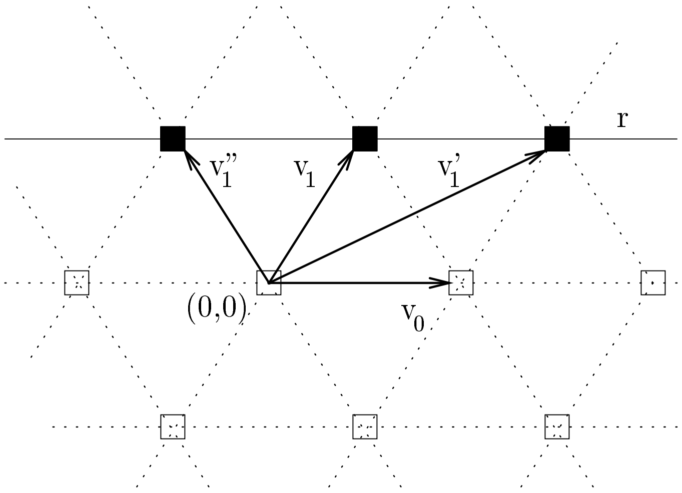

Figure 1: Equivalent two-dimensional lattices.

points but with different lattice coordinates. In fact, the points of the form $cv_0 + v_1$ (where $c \in \mathbf{Z}$) belong always to the straight line r of Figure 1. Note also that the lattice generated by $(\mathbf{w}_0, \dots, \mathbf{w}_{n-1})$ is also generated by $((-)^{i_1}\mathbf{w}_0, \dots, (-)^{i_{n-1}}\mathbf{w}_{n-1})$, i.e. by any combination of the opposite vectors to the $\mathbf{w}_i$. Moreover, by introducing the following equivalence relation among n-uples of lattice generators

$$\begin{array}{c} (\mathbf{v}_0, \dots, \mathbf{v}_{n-1}) \cong (\mathbf{w}_0, \dots, \mathbf{w}_{n-1}) \\ \Updownarrow \\ \Lambda(\mathbf{v}_0, \dots, \mathbf{v}_{n-1}) = \Lambda(\mathbf{w}_0, \dots, \mathbf{w}_{n-1}) \end{array} \tag{14}$$

the set of all possible lattice bases can be partitioned into equivalence classes[2], each of which is formed by all bases that generate a certain lattice.

Finally, note that between two different lattices Λ_v and Λ_w, there exists always an affine transform that transforms lattice generators into lattice generators. Such transformations preserve hyperplanes, straight lines and midpoints.

In the next paragraph we shall introduce the Voronoi tessellation of the n-dimensional space associated with a lattice. Voronoi tessellations will allow us to introduce the concept of neighbourhood, on which the definitions of curves and surface will be based.

4 Neighbourhood relations for n-dimensional lattices

The definition introduced in the previous section defines a lattice Λ as a discrete subset of $\mathbf{R}^n$, the elements of which are embedded regularly in the n-dimensional space. In discrete structures

[2] The relation defined above is an equivalence relation, since it is trivially reflexive, symmetric and transitive

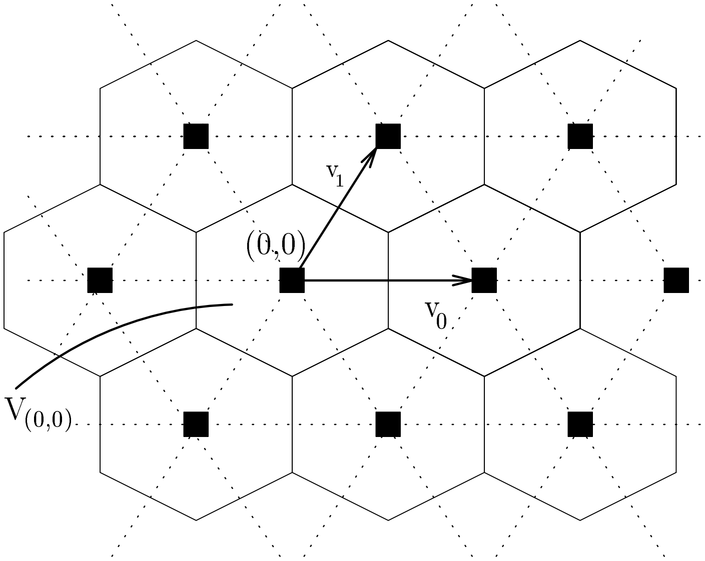

Figure 2: A two-dimensional lattice and the Voronoi sets associated to it.

such as lattices, neighbourhoods are the first structure introduced, and are used to define directly basic concepts such as distance and connectivity, which in turn are used for the definition of curves.

In the early days of Computer Graphics, Rosenfeld defined neighbourhoods in planar rectangular lattices [33]. Later, topological properties such as connectedness were studied for different lattice configurations in the plane [26, 19]. A first attempt to introduce a discrete geometry for graphics purposes was also carried out by Luby [27]. However, to date, researchers have not yet agreed upon the definition of neighbourhood or the notation to use, not even in the two-dimensional case.

Recently, Eker [16] proposed a more comprehensive definition of neighbourhood in the n-dimensional space. Unfortunately, his definition of l-connectedness leads to a different notation than the one used classically, for example in [34, 24, 17], both in the two- and in the three-dimensional cases. We propose here a new notation that agrees with the common definition of neighbourhood in two and three dimensional spaces.

A good place to start from in order to introduce a clean notation for neighbourhoods is the Voronoi tessellation of the space defined by the lattice Λ. Let Λ be a lattice in $\mathbf{R}^n$, and let $\lambda \in \Lambda$. The Voronoi set associated with a point $\lambda \in \Lambda$ is the set V_λ of all points of $\mathbf{R}^n$ that are strictly closer to λ than to any other point of the lattice. Figure 2 shows the points of a two-dimensional lattice and the Voronoi sets associated to them. Note that by definition each Voronoi set does not include its borders, and is therefore an open set in the Euclidean topology of $\mathbf{R}^n$. Thus the union of the Voronoi sets associated to all points of a lattice Λ does not cover entirely $\mathbf{R}^n$. If each Voronoi set V_λ is closed with its border points to obtain a new set $\overline{V}_\lambda$, the union of the resulting sets covers the n-dimensional space. The set $\overline{V}_\lambda$ will be called hereafter the *closed Voronoi set* associated to the point λ. In each of the equivalence classes defined by 14, from now onwards we shall consider only those n-uples of generators such that $\overline{V}_{(0,0,\ldots,0)} \cap \overline{V}_{(1,1,\ldots,1)}$ is non-empty.

Due to the fact that if two different n-uples generate the same lattice they also generate the same Voronoi sets associated with the lattice, this excludes from consideration lattices whose generators can be expressed as the vectors $\mathbf{w}_i$ in Equations 9 and 13. In fact, from the point of view of neighbourhoods such lattices differ only in the notation used but not in their intrinsic topology[3].

Note that the Voronoi sets associated with a discrete subset A of $\mathbf{R}^n$ are enumerable intersections of n-dimensional semispaces, since for each $a, b \in \mathbf{R}^n$, the set of points that lie closer to a than to b is the semispace π_{ab} that includes a and that is delimited by the $(n-1)$-dimensional hyperplane of the points equidistant from a and b. If the discrete subset A is a lattice, then the number of intersections is finite. In fact, consider the lattice point $\lambda = n_0\mathbf{v}_0 + \ldots + n_{n-1}\mathbf{v}_{n-1}$: using the triangle inequality it is easy to prove that the only points that have to be considered for defining the Voronoi set associated with it are the points $\lambda' \neq \lambda$ of the form $\lambda' = m_0\mathbf{v}_0 + \ldots + m_{n-1}\mathbf{v}_{n-1}$, where m_i can be either $n_i - 1$, n_i, or $n_i + 1$, and the number of such points is finite. Moreover, since only a finite number of $(n-1)$-dimensional hyperplanes delimit the Voronoi sets associated to a lattice Λ, the borders of such Voronoi sets are the union of a finite number of $(n-1)$-dimensional hyperpolygons.

Note also that the Voronoi set associated with any point $\lambda \in \Lambda$ is the Voronoi set associated with the origin translated by λ. In symbols, consider $\mathbf{v} \in \mathbf{R}^n$, and a subset $A \subset \mathbf{R}^n$, if $A + \mathbf{v}$ denotes the translated set of the set A through the vector $\mathbf{v}$, i.e. the set $A + \mathbf{v} = \{\mathbf{w} \in \mathbf{R}^n \mid \mathbf{w} = \mathbf{a} + \mathbf{v},\ \mathbf{a} \in A\}$, we have[4] that $V_\lambda = V_O + \lambda$.

The closures of the Voronoi sets associated with a lattice tessellate the n-dimensional space. A tessellation of the n-dimensional space is the generalization of a tiling in a two-dimensional space [6, 12]. More precisely, a *tessellation* in the n-dimensional space consists of a connected compact subset $A \subset \mathbf{R}^n$ and a subgroup G of the group of orientation preserving isometries of $\mathbf{R}^n$, such that the interior $\mathring{A}$ of A is non-empty and such that the following conditions are satisfied:

- $\bigcup_{g \in G} g(A) = \mathbf{R}^n$.

- If $g, h \in G$, $g(A) = h(A)$ whenever $g(\mathring{A}) \cap h(\mathring{A}) \neq 0$.

Simply put, the first condition requires the isometric copies of A to cover $\mathbf{R}^n$, while the second one guarantees that the interiors of the copies of A never overlap. The group G is called a *crystallographic group*.

Consider a lattice point λ, and let $\overline{V}_\lambda$ be the closed Voronoi set associated with it. Consider the closed Voronoi sets $\overline{V}_{\lambda_1}, \ldots, \overline{V}_{\lambda_m}$ such that their intersection with $\overline{V}_\lambda$ is non-empty. The points $\lambda_1, \ldots, \lambda_m$ will be called *neighbour candidates* of λ. The relation of neighbourhood in Λ will be based upon the type of intersection that exists among the sets $\overline{V}_\lambda$ and $\overline{V}_{\lambda_i}$.

[3] Of course at least one coefficient c_i has to be non-zero.

[4] This can be trivially proved by checking that all points of $V_\lambda - \lambda$ are in V_O and vice versa

In general, since the points $\lambda_i = (m_{i0}, \ldots, m_{i,n-1})$ (where $\forall i, j,\ m_{ij} \in \{n_j - 1, n_j, n_j + 1\}$) are the only possible neighbour candidates for the point $\lambda = (n_0, \ldots, n_{n-1})$, the number of neighbour candidates of a point of a lattice is smaller than or equal to the number of dispositions of three objects (i.e. $n_j - 1, n_j, n_j + 1$) in n positions, which is 3^n, minus the disposition corresponding to the point λ itself. Thus, the number N_Λ of neighbours of any point of a lattice $\Lambda(v_0, \ldots, v_{n-1}) \subset \mathbf{R}^n$ will be such that

$$N_\Lambda \leq 3^n - 1 \, .$$

The actual number of neighbour candidates of a lattice point depends on the angles formed by the lattice generators. It is difficult and would be beyond the scope of this paper to give an exact count of the number of neighbours of the origin. We will provide here only the equations for the possible borders of the Voronoi set associated with the origin. Since the Voronoi sets corresponding to the points of a lattice are invariant by translation, it is only necessary to give the equations of the Voronoi set associated to the origin.

Let $\mathbf{v}_i = (v_{i,0}, \ldots, v_{i,n-1})$ $(i = 0, \ldots, n-1)$ be n linearly independent vectors of $\mathbf{R}^n$, let O be the origin, and consider the lattice $\Lambda = \Lambda(v_0, \ldots, v_{n-1})$. Let V_λ denote the Voronoi set associated with the lattice point λ. Let $H_\beta = H_{b_0,\ldots,b_{n-1}}$, with $b_i \in \{0, 1\}$, be the hyperplane of the points equidistant from the origin and from the lattice point $\mathbf{v}_\beta = b_0\mathbf{v}_0 + \ldots + b_{n-1}\mathbf{v}_{n-1}$, i.e. the hyperplane passing through the point $\frac{1}{2}(b_0\mathbf{v}_0 + \ldots + b_{n-1}\mathbf{v}_{n-1})$ and perpendicular to the straight line joining the origin and $b_0\mathbf{v}_0 + \ldots + b_{n-1}\mathbf{v}_{n-1}$. Since this notation is quite complicated, whenever more convenient, the hyperplane of the points equidistant from O and $\mathbf{v}_i$ will be denoted H_i, while the hyperplane of the points equidistant from O and $\mathbf{v}_i + \mathbf{v}_j$ will be denoted $H_{i,j}$ and the hyperplane of the points equidistant from O and $\mathbf{v}_0 + \ldots + \mathbf{v}_{k-1}$ will be denoted $H_{0,\ldots,k-1}$. Note that H_i is the H_β such that $b_i = 1$ and $b_j = 0$ for $i \neq j$.

Let $\alpha_{i,j}$ be the angle[5] formed by the vectors $\mathbf{v}_i$ and $\mathbf{v}_j$.

The hyperplanes H_β define the semispaces the intersection of which is the Voronoi set associated with the origin. Let us write the equation of such a hyperplane: since the points of the straight line joining the origin and the point $b_0\mathbf{v}_0 + \ldots + b_{n-1}\mathbf{v}_{n-1}$ are the points $\mathbf{y} = y_0, \ldots, y_{n-1}$ such that

$$y_i = (b_0 v_{0,i} + \ldots + b_{n-1} v_{n-1,i})t \tag{15}$$

where $i = 0, \ldots, n-1$ and t is a real number, we have that the equation of H_β is of the form

$$\sum_{i=0}^{n-1} [(b_0 v_{0,i} + \ldots + b_{n-1} v_{n-1,i})x_i + p_i] = 0 \, . \tag{16}$$

Substituting in this equation the coordinates of the point $\frac{1}{2}(b_0 v_{0,i} + \ldots + b_{n-1} v_{n-1,i})$, remembering the principle of identity for polynomials, and solving with respect to p_i, we obtain the values of the p_i. Substituting these values into Equation 16 we obtain the equation of H_β:

$$\sum_{i=0}^{n-1} \left[(b_0 v_{0,i} + \ldots + b_{n-1} v_{n-1,i})x_i - \frac{1}{2}(b_0 v_{0,i} + \ldots + b_{n-1} v_{n-1,i})^2 \right] = 0 \tag{17}$$

[5] According to the standard notation, angles are considered to be oriented anticlockwise.

Starting from this equation, it is trivial to prove that H_i, H_{ij} and $H_{0\ldots k-1}$ are the loci of points respectively satisfying the equations

$$\sum_{k=0}^{n-1} v_{i,k}x_k - \sum_{k=0}^{n-1} \frac{(v_{i,k})^2}{2} = 0 \tag{18}$$

$$\sum_{k=0}^{n-1}(v_{i,k} + v_{j,k})x_k - \sum_{k=0}^{n-1} \frac{(v_{i,k} + v_{j,k})^2}{2} = 0 \tag{19}$$

$$\sum_{i=0}^{n-1} \left[(v_{0,i} + \ldots + v_{k-1,i})x_i - \frac{(v_{0,i} + \ldots + v_{k-1,i})^2}{2}\right] = 0\,. \tag{20}$$

Let us focus on the last term in Equation 17. We have

$$\begin{aligned}
&\frac{1}{2}\sum_{i=0}^{n-1}(b_0 v_{0,i} + \ldots + b_{n-1}v_{n-1,i})^2 = \\
&\quad \frac{1}{2}\sum_{i=0}^{n-1}\left[\sum_{j=0}^{i-1} b_j v_{j,i}^2 + 2\sum_{l=1}^{n-1} b_l v_{l,i} \sum_{j=0}^{l-1} b_j v_{j,i}\right] = \\
&\quad \frac{1}{2}\sum_{i=0}^{n-1}\sum_{j=0}^{i-1} b_j v_{j,i}^2 + \sum_{i=0}^{n-1}\sum_{l=1}^{n-1} b_l v_{l,i} \sum_{j=0}^{l-1} b_j v_{j,i} = \\
&\quad \frac{1}{2}\sum_{i=0}^{n-1}\sum_{j=0}^{i-1} b_j v_{j,i}^2 + \sum_{l=1}^{n-1}\sum_{i=0}^{n-1} b_l v_{l,i} \sum_{j=0}^{l-1} b_j v_{j,i} = \\
&\quad \frac{1}{2}\sum_{i=0}^{n-1}\sum_{j=0}^{i-1} b_j v_{j,i}^2 + \sum_{l=1}^{n-1}\sum_{i=0}^{n-1}\sum_{j=0}^{l-1} b_l b_j v_{l,i} v_{j,i} = \\
&\quad \frac{1}{2}\sum_{i=0}^{n-1}\sum_{j=0}^{i-1} b_j v_{j,i}^2 + \sum_{l=1}^{n-1}\sum_{j=0}^{l-1}\left[\sum_{i=0}^{n-1} b_l b_j v_{l,i} v_{j,i}\right]\,,
\end{aligned} \tag{21}$$

and thus we have that H_β is the plane of equation

$$\begin{aligned}
&\sum_{i=0}^{n-1}(b_0 v_{0,i} + \ldots + b_{n-1}v_{n-1,i})x_i - \\
&\quad \frac{1}{2}\sum_{i=0}^{n-1}\sum_{j=0}^{i-1} b_j v_{j,i}^2 - \sum_{l=1}^{n-1}\sum_{j=0}^{l-1}\left[\sum_{i=0}^{n-1} b_l b_j v_{l,i} v_{j,i}\right] = 0\,.
\end{aligned} \tag{22}$$

Let us rewrite the sum

$$\sum_{i=0}^{n-1} b_l b_j v_{l,i} v_{j,i} \;=\; b_l b_j \sum_{i=0}^{n-1} v_{l,i} v_{j,i}. \tag{23}$$

Note that if b_l or b_j are zero, i.e. if $\mathbf{v}_l$ or $\mathbf{v}_j$ do not contribute to the point $\mathbf{v}_\beta$, then the sum is zero. Note also that the sum in the right hand expression is the inner product of the vectors $\mathbf{v}_l$

and $\mathbf{v}_j$. We can therefore rewrite the equation of H_β as

$$\sum_{i=0}^{n-1}(b_0 v_{0,i} + \ldots + b_{n-1}v_{n-1,i})x_i - \frac{1}{2}\sum_{i=0}^{n-1}\sum_{j=0}^{i-1} b_j v_{j,i}^2 - \sum_{l=1}^{n-1}\sum_{j=0}^{l-1} b_l b_j \|\mathbf{v}_l\|\|\mathbf{v}_j\| \cos\alpha_{l,j} = 0\,, \tag{24}$$

where $\alpha_{l,j}$ is the angle formed by the vectors $\mathbf{v}_l$ and $\mathbf{v}_j$.

Equation 24 is very important, since it expresses the equation of all the potential borders of the Voronoi set associated with the origin. The first two terms of the left hand side of the equation are the sum of the left hand terms of the equations of H_i, $i = 0, \ldots, n$ (cfr. of Equation 18). This implies that given a point $\mathbf{y} \in \mathbf{R}^n$ and the hyperplane H_β, the evaluation of H_β in $\mathbf{y}$ is equal to

$$H_\beta(\mathbf{y}) = \sum_{i=0}^{n-1} b_i H_i(\mathbf{y}) - \sum_{l=1}^{n-1}\sum_{j=0}^{l-1} b_l b_j \|\mathbf{v}_l\|\|\mathbf{v}_j\| \cos\alpha_{l,j} \tag{25}$$

Note that Expression 25 contains only evaluations of hyperplanes perpendicular to the lattice generators, and that the last sum contains only the planar angles formed by pairs of lattice generators.

The next theorem introduces a necessary condition for two points to be neighbour candidates.

Theorem 4.1 *Consider the point $\mathbf{v}_\beta$, and consider the sum*

$$S = \sum_{l=1}^{n-1}\sum_{j=0}^{l-1} b_l b_j \|\mathbf{v}_l\|\|\mathbf{v}_j\| \cos\alpha_{l,j}. \tag{26}$$

If $S > 0$, then $\mathbf{v}_\beta$ cannot be a neighbour candidate of the origin.

Proof. Consider the origin O. From Equation 18 we have that

$$H_i(O) = -\sum_{k=0}^{n-1} \frac{(v_{i,k})^2}{2} \tag{27}$$

and this expression is always negative. Obviously, the hyperplanes H_i contribute to the definition of V_O, since the vectors $\mathbf{v}_i$ are possible neighbours of the origin. If it is possible to prove that for all points $\mathbf{y} = (y_1, \ldots, y_n) \in H_\beta$ there is at least one H_i such that $H_i(\mathbf{y}) > 0$, then none of the points of H_β lies on the border of V_O, and consequently $\mathbf{v}_\beta$ is not a neighbour candidate of the origin O.

Consider therefore a point $\mathbf{y} \in H_\beta$. Clearly, $H_\beta(\mathbf{y}) = 0$, and thus we have that

$$\sum_{i=0}^{n-1} b_i H_i(\mathbf{y}) = \sum_{l=1}^{n-1}\sum_{j=0}^{l-1} b_l b_j \|\mathbf{v}_l\|\|\mathbf{v}_j\| \cos\alpha_{l,j} \tag{28}$$

or, more simply put,

$$\sum_{i=0}^{n-1} b_i H_i(\mathbf{y}) = S\,. \tag{29}$$

Since by the hypothesis $S > 0$, and since Equation 29 must hold, and since all the b_i coefficients are positive or null, there must be at least one H_j such that $H_j(\mathbf{y}) > 0$. The point $\mathbf{y}$ therefore does not lie in the semispace defined by H_j that includes the origin, and therefore cannot belong to the border of V_0, since V_0 is defined as the intersection of the semispaces defined by the hyperplanes of type $H_{\beta'}$ and containing the origin. Since this fact is true for all $\mathbf{y} \in H_\beta$, H_β lies outside the boundaries of the Voronoi set associated with the origin, and therefore $\mathbf{v}_\beta$ is not a neighbour candidate of the origin. □

In the most common cases, the theory of n-dimensional rasterization has to be applied to lattices generated by orthogonal vectors. In this case it is therefore sensible to give some more detail on the characteristics of the borders of the Voronoi sets.

Theorem 4.2 *Suppose that the lattice generators $v_{i_0}, \ldots, v_{i_{k-1}}$ (with $i_j \neq i_m$ for $j \neq m$) are orthogonal. Then the closures of the Voronoi sets associated with the origin and with the point $\mathbf{v}_{i_0} + \ldots + \mathbf{v}_{i_{k-1}}$ have at most a subset of the $(n-k)$-dimensional space in common.*

Proof. By the axiom of choice, we can always suppose that the orthogonal vectors are the vectors $\mathbf{v}_0, \ldots, \mathbf{v}_{k-1}$.

Note that from the definition of the hyperplanes H_i, their evaluation in the origin O of the n-dimensional space is such that $\forall i$,

$$H_i(O) < 0 \tag{30}$$

Consider the hyperplane H_β, where this time $b_i = 1 \; \forall i < k$ and $b_i = 0 \; \forall i \geq k$. Also here H_β is defined by Equation 22. Since the vectors $\mathbf{v}_0, \ldots, \mathbf{v}_{k-1}$ are mutually orthogonal, the inner product $\sum_{i=0}^{n-1} v_{l,i} v_{m,i} = 0$ for all $l, m < k$ $(l \neq m)$, and therefore the equation of H_β becomes

$$\sum_{i=0}^{n-1} (v_{0,i} + \ldots + v_{k-1,i}) x_i - \frac{1}{2} \sum_{i=0}^{n-1} \sum_{j=0}^{k-1} v_{j,i}^2 = 0 \,. \tag{31}$$

Remembering the definitions of H_i, we have that for a point $\mathbf{y} \in H_\beta$,

$$H_\beta(\mathbf{y}) = H_0(\mathbf{y}) + \ldots + H_{k-1}(\mathbf{y}) = 0 \,. \tag{32}$$

There are two cases possible: if for all possible i $(i = 0, \ldots, k-1)$, $H_i(\mathbf{y}) = 0$, then $\mathbf{y} \in H_i$ for all i, and thus $\mathbf{y} \in \bigcap_{i=0}^{k} H_i$.

If instead there is at least one $H_i(\mathbf{y}) \neq 0$, by the axiom of choice we can always suppose that the hyperplane in question is H_0. If $H_0(\mathbf{y}) > 0$, then the origin O and the point $\mathbf{y}$ lie in different semispaces with respect to the hyperplane H_0, and since the hyperplane H_i contributes to the definition of the Voronoi set associated with the origin, then $\mathbf{y}$ does not belong to the border of V_O. If instead $H_0(\mathbf{y}) < 0$, then equation 32 implies that the sum $H_1(\mathbf{y}) + \ldots + H_{k-1}(\mathbf{y}) > 0$, and thus that there is at least one H_i such that $H_i(\mathbf{y}) > 0$. Therefore, in this case also, $\mathbf{y}$ does not belong to the border of V_O.

A point $\mathbf{y} \in H_\beta$ thus is either external to the Voronoi set associated with the origin or belongs to all H_i. The hyperplane H_β does not therefore contribute to the border of V_O other than the points of $\bigcap_{i=0}^{k} H_i$, and since this is the intersection of k linearly independent hyperplanes, it is an $(n-k)$-dimensional flat. V_O and V_β can at most share this flat, and this flat is not empty, since it contains at least the point $\frac{1}{2}\mathbf{v}_\beta$, and since the hyperplanes H_i are perpendicular to linearly independent vectors. □

Even if some of the lattice generators are orthogonal, little more can be said about the number of neighbour candidates of a lattice point. If instead all lattice generators are orthogonal, then the theorem above proves indirectly that all points $\mathbf{v} = m_0\mathbf{v}_0 + \ldots + m_{n-1}\mathbf{v}_{n-1}$ (where $m_i \in \{-1, 0, 1\}$) are neighbour candidates of the origin, since it proves that all the Voronoi sets V_β, i.e. the points $\mathbf{v}$ having positive or null m_i coefficients, share a flat with V_O, and since some simple reasoning on the symmetries involved proves an analogous result for the vectors $\mathbf{v}$ having negative m_i coefficients.

We are now ready to introduce the concept of l-neighbourhood in n-dimensional space.

Definition 4.1 *Two lattice points λ and λ' are said to be* l-neighbours *if the closures of the corresponding Voronoi sets V_λ and $V_{\lambda'}$ share an $(n-l)$-dimensional hyperface.*

Definition 4.2 *The* l-neighbourhood *of a lattice point λ is the set of lattice points which are l-neighbours to it.*

Definition 4.3 *Two lattice points λ and λ' are said to be* $l_\leq$-neighbours *if the closures of the corresponding Voronoi sets V_λ and $V_{\lambda'}$ share an $(n-k)$-dimensional hyperface (where $k \leq l$).*

Definition 4.4 *The* $l_\leq$-neighbourhood *of a lattice point λ is the set of lattice points which are $l_\leq$-neighbours to it.*

Figure 3 illustrates the concept of l-neighbours for three-dimensional hypercubic lattices. Three mirrors have been placed behind and underneath the subject to help the viewing of the hidden faces. In hypercubic lattices, Voronoi sets are hypercubes centered in the lattice points. The reference lattice point λ is the point the Voronoi set of which is the front bottom left voxel, i.e. the voxel with no striping. Equal thickness of the striping of the Voronoi set surface indicates that the corresponding lattice points are l-neighbours of the reference point. Thicker striping indicates smaller values of l. Obviously, $l_\leq$-neighbours are all the points corresponding to the voxels having thinner or equal striping to the l-neighbours of λ.

The number of $l_\leq$ neighbours of a lattice point is easily computed when the lattice generators are all mutually orthogonal. In fact, the following theorems hold:

Theorem 4.3 *Let Λ be the lattice of $\mathbf{R}^n$ formed by n mutually orthogonal vectors. The number N of l-neighbours of a lattice point λ is*

$$N = 2^l \binom{n}{l} \tag{33}$$

Proof. Since the Voronoi sets associated with a lattice Λ are invariant by translation, the theorem will be proved for the neighbourhood of the origin.

All l-neighbours of the origin have all their lattice coordinates equal to 0,1 or -1. The l-neighbours of the origin are the lattice points having all but l coordinates equal to 0, and the remaining coordinates equal to either 1 or -1. Thus their number equals the number of distinct ordered n-uples from an alphabet of two signs (the coordinate equals zero or it doesn't) having

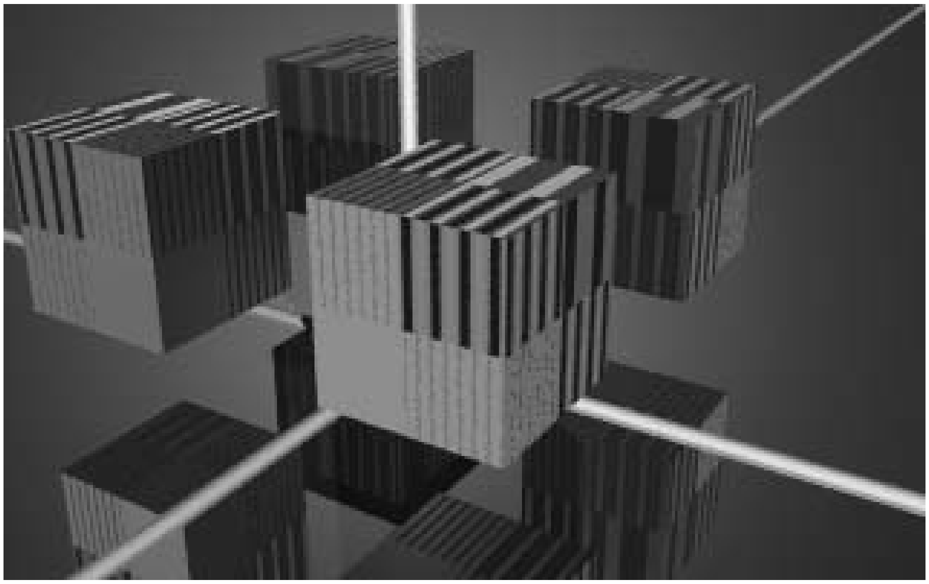

Figure 3: Neighbour candidates for the origin in a three-dimensional rectangular lattice in the positive directions of the three axes.

l coordinates of one type (the coordinate is not zero), i.e. $\binom{n}{l}$, multiplied by the number of possible dispositions of 1 and -1 over l positions, i.e. 2^l. □

Given the definition of $l_{\leq}$-neighbours, the proof of the following theorem is trivial:

Theorem 4.4 *The number $N_{\leq}$ of $l_{\leq}$-neighbours of a lattice point λ is*

$$N_{\leq} = \sum_{i=1}^{l} 2^i \binom{n}{i} \tag{34}$$

Despite the more complicated definition, in general in the literature $l_{\leq}$-neighbourhoods are considered [24, 17, 34], and only Eker [16] in his thesis uses l-neighbourhoods[6]. In this paper we shall adhere to the classic notation.

The introduction of neighbourhoods in lattices allows the definition of arc and curve in an n-dimensional lattice. In order to do this, it is worthwhile to remember that a lattice onto which a neighbourhood relation has been defined can also be seen as an infinite graph $\langle \Lambda, E \rangle$ embedded in $\mathbf{R}^n$, the nodes of which are the lattice points and the edges of which link neighbouring lattice points. Arcs can then be introduced on an n-dimensional lattice with the same procedure used on infinite graphs: given two lattice points A and B, an *$l_{\leq}$-arc* from A to B is a finite sequence of lattice elements $P_0, P_1, \ldots, P_n$ such that $\forall i$, P_i and P_{i+1} are $l_{\leq}$-neighbours and such that $P_0 = A$ and $P_n = B$ [34]. The points A and B are called *endpoints* of the arc. Note how this definition corresponds to the definition of the path between two nodes of a graph. An arc is said to be *closed* if the two endpoints coincide. An *$l_{\leq}$-curve* is an infinite arc, i.e. a sequence of lattice points $\{P_i\}_{i=-\infty}^{+\infty}$ such that P_i and P_{i+1} are $l_{\leq}$-neighbours $\forall i$.

[6] In the classical literature, $l_{\leq}$-neighbourhoods are denoted after the number of neighbours of a lattice point, $N_{\leq}$. Brevity considerations (the number of neighbours can be very big in n-dimensional spaces) suggested using $l_{\leq}$-neighbourhoods instead.

It is useful to remark here that 1-connected arcs are such that in the sequence of neighbours building the path each element shares with the next all lattice coordinates except one. Such a path is often said to be built by steps *along the directions of the lattice generators.* Note also how an $l_{\leq}$-connected arc can be seen as being built from a 1-connected arc by grouping together up to l steps in the directions of different generators into one single multi-dimensional diagonal step, called *hop*.

Now that $l_{\leq}$-arcs and $l_{\leq}$-curves have been defined directly on n-dimensional lattices, we are ready to introduce the operation of rasterization, which is the mechanism that governs the transformation of continuous subsets of the n-dimensional real space into subsets of n-dimensional discrete lattices.

5 Rasterization in n-dimensional space

Discrete lattices can be used to model the output space of most common raster devices. By raster device we understand here a device which can represent a set of discrete values $i_1, i_2, \ldots, i_m$ on a discrete subset L of the n-dimensional real space. In symbols, a raster device A is a device that can represent a discrete-valued function

$$D : L \longrightarrow I_1 \times I_2 \times \ldots I_m \ ,$$

where $I_1, I_2, \ldots I_m$ are discrete (usually finite) subsets of the real numbers. Typical examples of raster devices in two-dimensional space are traditional Computer Graphics square-based output devices, such as raster screens and dot-matrix, inkjet or laser printers, the domain of which (L) is usually a rectangular subset[7] of $\mathbf{Z}^2$ and the codomain of which is the set $\{0, 1\}$ for black and white raster screens and printers, the set $RGB = \{0, \ldots, 255\} \times \{0, \ldots, 255\} \times \{0, \ldots, 255\}$ for 24 bit display devices, or a set of colours $c_1, \ldots, c_N$ [8] for colour lookup table (CLUT) devices. Examples of three-dimensional raster devices are medical imaging devices, such as MRI (Magnetic Resonance Imaging) and CT (Computerized Tomography: TAC) devices, which have $\mathbf{Z}^3$ as a domain, and for imaging purposes RGB as a codomain, while there are also four-dimensional devices such as PET (Positron Emitting Tomography) scanners, which scan the variations of three-dimensional data in time, and thus data in four dimensions.

The use of a unified model for raster devices allows to utilize the same notation both for raster input devices, such as scanners, and for raster output devices, such as printers and raster screens. In particular, the operation of scanning, i.e. of sampling a continuous signal through a raster device into a discrete dataset for its elaboration through a computer, looks similar, in principle, to the operation of rasterizing, i.e. of discretizing the representation of data through continuous curves, polygons, and all the components of a geometric model, into a discrete set of data representable on a raster output device. In both cases, due to the discrete nature of the device, the sampling process can only be done at fixed frequencies, thus causing a noticeable loss of detail as well as aliasing problems. The similarities in the two sampling processes end here, though. Scanning devices are limited by their resolution, and data can be sampled only at

[7] The range of which, of course, depends on the resolution of the device.

[8] Here each colour c_i is a triplet of integers in the set of all possible colours representable by the device, which, in general, is a set of type $\{0, \ldots, N_1\} \times \{0, \ldots, N_2\} \times \{0, \ldots, N_3\}$.

fixed frequencies. The resolution limits of the device cannot be overcome without an increase of the sampling resolution, i.e. without buying a higher resolution device, or through new devices that place the sampling sets more efficiently, as has been proposed recently [32, 15, 42]. Raster output devices, instead, permit a different approach, since the continuous model that has to be sampled is readily available in the computer. The sampling frequency of the model can thus be increased at will (super-sampling) and sampling errors can be averaged, or partially corrected, before display on the physical device. The study of such techniques, called anti-aliasing, is well-advanced in Computer Graphics, and wide literature is available on the subject [18]. However, such techniques are in general very expensive, and do not address the core problems caused by the rasterization process. As many authors have pointed out [25, 19, 26, 16], there is a real lack of study on the mathematical foundations of rasterization, and the notation used is either derived from the theory of sampling, and thus is too heavy, or is too incomplete for productive use. In this section we will focus on the introduction of a notation for n-dimensional lattices.

The operation of sampling a continuous m-dimensional signal

$$f : \mathbf{R}^n \longrightarrow \mathbf{R}^m$$

to obtain an m-dimensional discrete signal of a discrete variable

$$D : L \longrightarrow I_1 \times I_2 \times \ldots I_m$$

which can be output on a raster display device is called *discretization* of the signal f.

A discretization function ρ therefore associates with a function f which has to be discretized its discretization D, *i.e.*

$$\rho : \ \mathcal{F} \longrightarrow \ \mathcal{D} \, ,$$

where $\mathcal{F} = \{f : \mathbf{R}^n \longrightarrow \mathbf{R}^m\}$ and $\mathcal{D} = \{D : L \longrightarrow I_1 \times \ldots \times I_m\}$. From the above definitions it can be observed that discretization is, in fact, a particular sampling process in the n-dimensional space that can be characterized by the fact that also the output set $I = I_1 \times I_2 \times \ldots I_m$ is also discrete.

In general, for Computer Graphics purposes, both L and I are subsets respectively of $\mathbf{R}^n$ and $\mathbf{R}^m$, and the discretization is done so that some visually important characteristics of the function f are preserved. In particular, discretization is performed in such a way that the embedding of $\rho(f)$ in $\mathcal{F}$ is an "acceptable" approximation of f. The criteria for the definition of acceptable vary widely, and depend mainly on the function ρ used.

Some rasterization schemes can be directly derived as the n-dimensional extension of existing rasterization schemes for two-dimensional rasterizations. Let $\Lambda = \Lambda(\mathbf{v}_0, \ldots, \mathbf{v}_{n-1})$ be an n-dimensional lattice, consider the Voronoi set associated with the origin V_O. Consider a simply connected set K_ρ such that $K_\rho \subseteq V_O$. The set K_ρ is called the *basic domain of the rasterization scheme.* Let $\lambda \in \Lambda$, and let $K_\rho(\lambda)$ be the translated set of K_ρ through the vector λ, in other words, let $K_\rho(\lambda) = \{\mathbf{y} \in \mathbf{R}^n \, : \, \mathbf{y} = \lambda + \mathbf{x}, \text{where}\, \mathbf{x} \in K_\rho\}$. The rasterization of a point and of a set can be defined in the following way:

Definition 5.1 *Let $\mathbf{v} \in \mathbf{R}^n$. The rasterization $Dig_{K_\rho}(\mathbf{v})$ of $\mathbf{v}$ is defined by the following relation*

$$Dig_{K_\rho}(\mathbf{v}) = \begin{cases} \{\lambda\} & \textit{iff } (\lambda \in \Lambda) \wedge (\mathbf{v} \in K_\rho(\lambda)) \\ \\ \emptyset & \textit{otherwise.} \end{cases} \tag{35}$$

Definition 5.2 *Let $A \subseteq \mathbf{R}^n$. The rasterization of A is the set*

$$Dig_{K_\rho}(A) = \bigcup_{\mathbf{x} \in A} Dig_{K_\rho}(\mathbf{x}) \,. \tag{36}$$

The rasterization function introduced by these definitions is called *nearest neighbour* rasterization. Although it is defined exactly in the same way as two-dimensional rasterization is, the operation of rasterizing in n-dimensional space is, of course, more complicated, and requires particular care in the choice of the set K_ρ. If $K_\rho = V_O$, then the rasterization function is called *cellular rasterization*. The major drawback of cellular rasterization is the fact that in the case of n orthogonal lattice generators, whenever a curve has to be rasterized, its cellular rasterization in Λ is an $n_{\leq}$-curve. Although this is a desirable feature for hypervoxel[9] traversal algorithms, in some cases [3] it is more desirable to compute $l_{\leq}$-curves for rasterization in the n-dimensional lattice, where $l < n$, just as in the two-dimensional case, where 8-connected curves are usually preferred to 4-connected curves.

As in the two-dimensional case, there is a particular choice of the set K_ρ that generates 0-connected curve digitalizations in the case of orthogonal lattice generators. If the vectors $\mathbf{v}_i$ are orthogonal, by letting one of the axes coincide with the straight line containing one generator, we can always lay an orthogonal cartesian coordinate system in $\mathbf{R}^n$ such that all the coordinates of the vector $\mathbf{v}_i$ are zero except the i-th coordinate, $v_{i,i}$. Let A_i be the hyperplane passing through the origin and perpendicular to the vector $\mathbf{v}_i$. Consider the Voronoi set associated with the origin V_O, and let $A'_i = A_i \cap V_0$. Consider now as basic domain of the rasterization scheme the set

$$K_\rho = \bigcup_{i=0}^{n-1} A'_i \,.$$

The scheme defined in this way computes the intersections of the curve γ to be rasterized with the hyperplanes of the form

$$x_i = j v_{i,i}$$

where j is an integer, and then approximates the resulting points to their nearest lattice point[10]. The resulting lattice points build the rasterization of the curve. This last type of rasterization is called *grid intersection* rasterization.

Grid intersection rasterization can be extended to non-orthogonal lattices by substituting the hyperplanes A_i with the hyperplanes $\overline{A}_i$ passing through the origin and containing all vectors $\mathbf{v}_j$ for all $j \neq i$. In the lattice coordinate system, such hyperplanes have the form $x_i = 0$, with k integer. To rasterize a curve γ in this case, its intersections with the grid of hyperplanes of the

[9] Where a hypervoxel is by definition the set V_λ.

[10] Such a point is not always unique. However, this ambiguity can be trivially overcome by forcing uniqueness.

form $x_i = j$ (in the lattice coordinate system) have to be computed, and the resulting points are then approximated to the nearest lattice point on the hyperplane.

In general, cellular rasterization schemes guarantee that the rasterization of any given point P exists, since the Voronoi sets associated with a lattice tessellate $\mathbf{R^n}$, and should be therefore used to rasterize single points, whereas grid intersection rasterizations are more convenient whenever a curve has to be rasterized, since they perform intersections with a set of parallel hyperplanes, the equations of which are easy to handle.

Nearest neighbour rasterization schemes provide a fairly simple mechanism for the definition of rasterization. However, they are not extensible to take into account the concept of $l_{\leq}$-neighbourhoods. In order to generate the discretization of a set A, such schemes use replicas of one single (fixed) set, and compute the intersections of the set to be rasterized with these copies. The result of this procedure is then a connected set of lattice points that represent the set A on the lattice, and that lies the closest possible to the set to be rasterized, within the limits imposed by the discrete nature of lattices and by the relation of neighbourhood involved. With the schemes introduced above, there is one and only one optimal choice of the resulting rasterization set[11].

To extend the operation of rasterization to lattices onto which a generic $l_{\leq}$-neighbourhood is defined, from now on in this paper we restrict ourselves to the definition of *curve rasterization* onto n-dimensional lattices, since a definition of rasterization for generic sets would imply a thorough analysis of the topology of n-dimensional lattices, and this would be well beyond the scope of this paper. The notation presented above for the definition of rasterization, although more general, thus, turns out to be too heavy: in fact, curve rasterization is the rasterization of subsets of $\mathbf{R}^{n+1}$. As we mentioned in Section 2, a curve is a particular subset of $\mathbf{R}^{n+1}$, namely a continuous function $\gamma : \mathbf{R} \longrightarrow \mathbf{R}^n$, and therefore its rasterization must be a curve in a lattice $\Lambda \subset \mathbf{R}^n$, i.e. a sequence $S : \mathbf{Z} \longrightarrow \Lambda$ such that $S(i)$ and $S(i+1)$ are neighbours. The discrete curve resulting from the rasterization process will therefore be a connected subset of points of Λ that lies the closest possible to the curve to be rasterized.

The keywords for the definition are the words *connected* and *closest.* Connectivity is the characteristic which is most important from an observer's standpoint: a rasterization of a curve must be connected. As far as closeness is concerned, an observer is usually less categorical about it, and depending on the requirements of the system this closest criterion can be relaxed [17, 27]. For example, in two-dimensional rasterizations, for speed reasons a polygon inscribed in a curve might be rasterized instead of the curve itself [22, 40]. The rasterization scheme, therefore, must ensure curve connectedness first.

The choice of the closest $l_{\leq}$-path to the curve γ to be rasterized is, in general, more complicated than the choice of a closest $0_{\leq}$-path or $n_{\leq}$-path to γ, because 0- and n-rasterizations involve only local decisions in the rasterization process, whereas generic $l_{\leq}$-rasterizations have also to take into account conditions for the neighbours of the current point.

Similarly to grid intersection rasterizations, the starting point for the definition of an $l_{\leq}$-path generating scheme, is calculating the intersection of the continuous curve γ with a grid of lines, so as to partition the rasterization procedure into an (enumerable) sequence of steps. However here the choice of the next rasterization point has to take into account the local neighbourhood

[11] In fact, all the schemes introduced to date avoid in one way or another the ambiguities derived from the points on the border of V_λ, which inherently are points shared by more than one K_λ set, by arbitrarily attributing these points to one and only one K_λ

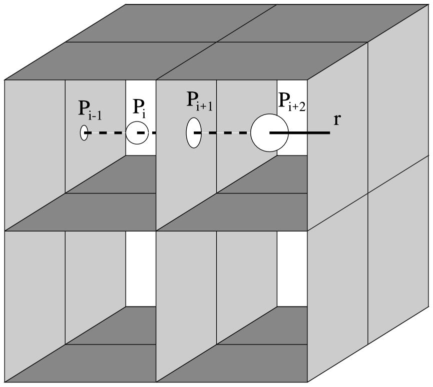

Figure 4: Intersection of a curve r with the grid hyperplanes.

configurations, which impose additional conditions on the choice of the points of the discrete curve.

Let $\Lambda(\mathbf{v}_0, \ldots, \mathbf{v}_{n-1})$ be the lattice onto which we want to perform rasterization, let $\gamma : [0,1] \longrightarrow \mathbf{R^n}$ be the curve to be rasterized, and let A and B be its endpoints. Let the coordinates of the generator $\mathbf{v}_i$ be $(v_{i,0}, \ldots, v_{i,n-1})$ in the orthogonal coordinate system. Consider the coordinate system introduced by the lattice generators. In this coordinate system, the coordinates of the generator $\mathbf{v}_i$ will all be zero, except the i-th coordinate which will be 1. In the coordinate system introduced by the lattice generators in $\mathbf{R^n}$, consider the hyperplanes of the form $x'_i = k$. In other words, consider the hyperplanes $H_{k\mathbf{v}_i}$ whose equation in the orthogonal coordinate system is

$$v_{i,0}x_0 + \ldots + v_{i,n-1}x_{n-1} = k(v_{i,0}^2 + \ldots + v_{i,n-1}^2) \, ,$$

where $k \in \mathbf{Z}$. The hyperplanes $H_{k\mathbf{v}_i}$ form a grid over $\mathbf{R^n}$, which will be intersected with the curve to be rasterized to obtain the lattice rasterization points. Note that, in the case of rectangular lattices, the grid defined here coincides with the grid which derives from grid intersection rasterization.

Let λ_A and λ_B respectively be the grid point digitalizations of A and B,[12] and consider the intersections of γ with the grid of the hyperplanes $H_{k\mathbf{v}_i}$. For each point P, intesection between γ and the hyperplane $H_{k_P\mathbf{v}_i}$, let P' be its nearest lattice point on the hyperplane $H_{k_P\mathbf{v}_i}$. The paramenter t induces an ordering on the points of the curve γ, and thus also on the intersections of γ with the grid. Let $S = \{P_0, \ldots, P_r\}$ be the ordered sequence of intersections of γ with the hyperplanes $H_{k\mathbf{v}_i}$. This sequence represents the order in which the hyperplanes $H_{k\mathbf{v}_i}$ will be intersected by the curve γ for growing values of t, as illustrated in Figure 4. The sequence S induces also an ordering on the sequence S' of the P'_i, i.e. of the nearest neighbours defined above. Note that the P'_i are lattice points, and will be used to build the $l_{\leq}$-path, rasterization

[12] From a logical point of view, if we denote the rasterization of γ with Γ, both λ_A and λ_B should belong to Γ. However, in order to maintain consistency, in traditional rasterization algorithms this is not always the case, and here we shall follow the same approach.

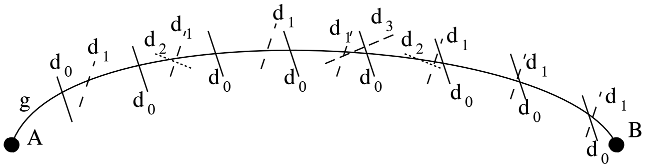

Figure 5: Grouping 1-steps into $l_{\leq}$-steps for producing the rasterisation of a curve r.

of γ. From this sequence, it is easy to extract an $l_{\leq}$-connected path: up to l steps along the directions of the generators can be grouped into a single diagonal step, provided that a single ("combined") step is not composed of more than one step in each direction.

To illustrate this a little, imagine an observer travelling in the n-dimensional space from A to B along the curve γ. Such an observer would "bump" into the hyperplanes of the grid at the points P_i in a certain order, which defines a sequence S of points on the grid hyperplanes. Each time the observer encounters a hyperplane in a point P, he scrupulously takes note of the point encountered, looks for the closest lattice point to it, say P', and takes note of it too. Once the observer arrives at B, he has built two sequences of points S and S', respectively containing the points P and P', from which it is possible to build a shortest $l_{\leq}$-path from A to B. Let us call this path Γ. In order to do this, it suffices to note that each point P' in S' represents a step in the direction d perpendicular to the grid hyperplane containing P'. The sequence S' therefore defines a sequence D of single steps in the directions of the axes which builds a 1-path from A to B. From the sequence D we can obtain an $l_{\leq}$ path Δ by simply grouping together in each hop up to l different steps in the directions of the generators.

Let $D = \{d_{i_0}, \ldots, d_{i_m}\}$. Here, naturally, the elements of D are directions parallel to the lattice generators. Let i_{h_0} be the minimum index of the sequence D that has appeared before in D. Let $k_0 = Min(l, h_0)$. In the first hop, the first $k_0 \leq l$ steps of D in different directions of the single generators can be grouped in a single hop. Let us denote this fact by rewriting D and enclosing grouped steps in brackets. We have $D = \{(d_{i_0}, \ldots, d_{i_{k_0}}), d_{i_{k_0+1}}, \ldots, d_{i_m}\}$. Consider now the subsequence extracted from D starting from i_{k_0+1}, i.e. the subsequence $D_{k_0} = \{d_{i_{k_0+1}}, \ldots, d_{i_m}\}$ and, again, let i_{h_1} be the minimum index of the sequence that has appeared before in D_{k_0}. Let $k_1 = Min(l, h_1)$. In the second hop of the $l_{\leq}$ path, the steps from $d_{i_{k_0+1}}$ to $d_{i_{k_1}}$ can be grouped together in a single step. We can thus rewrite D as

$$D = \{(d_{i_0}, \ldots, d_{i_{k_0}}), (d_{i_{k_0+1}}, \ldots, d_{i_{k_1}}), d_{i_{k_1+1}}, \ldots, d_{i_m}\} .$$

This can be repeated until all the directions of D are grouped in hops. The result of this procedure is the grouping of the directions in D into $l_{\leq}$ hops which define an $l_{\leq}$ path from A to B. Such a path represents the rasterization of the curve γ.

Figure 5 illustrates this in an example. The curve $g \in \mathbf{R^4}$ has to be rasterized onto an orthogonal $3_{\leq}$-connected[13] lattice Λ from the point A to the point B. Since we are working in 4-dimensional space, there will be four lattice generators, and therefore four parallel sheafs

[13] Note that orthogonality is required here to allow $l_{\leq}$-connectedness.

of hyperplanes (perpendicular to the lattice generators) which will be used to compute the rasterization of g. In the figure, the projection of g onto a convenient plane is shown. The diagonal segments intersecting the curve represent the intersections of the curve with the sheafs of hyperplanes of equation $x_i = k$ $(i = 0, \ldots, 3)$, where parallel hyperplanes have been represented by parallel segments. The intersections found define a 1-connected path from A to B, the steps of which build the sequence of directions

$$D = \{d_0, d_1, d_0, d_2, d_1, d_0, d_1, d_0, d_1, d_3, d_0, d_2, d_1, d_0, d_1, d_0, d_0, d_1\} .$$

The directions of D are grouped to form an $l_{\leq}$ path in the following way:

$$\begin{aligned} D = \; & \{(d_0, d_1), (d_0, d_2, d_1), (d_0, d_1), (d_0, d_1, d_3), \\ & (d_0, d_2, d_1), (d_0, d_1), (d_0), (d_0, d_1)\} . \end{aligned}$$

Note that here one step in the direction d_0 cannot be grouped together with any other step, due to the fact that d_0 occurs twice in a row. Note also that the chosen path is not the only $3_{\leq}$-path possible: for example also the path

$$\begin{aligned} D = \; & \{(d_0, d_1), (d_0, d_2, d_1), (d_0, d_1), (d_0, d_1, d_3), \\ & (d_0, d_2, d_1), (d_0, d_1), (d_0), (d_0, d_1)\} . \end{aligned}$$

is an $l_{\leq}$-connected path from A to B, and represents g accurately. This non-uniqueness is due to the nature of $l_{\leq}$ paths. However, all possible paths derived from D by grouping together up to 3 different directions will be close enough to the original curve to represent it "well". A more precise definition of rasterization which would compute a unique path would have to compromise greatly in speed, and is thus not desirable, since all possible legal groupings within D would have to be analyzed. The algorithm defined above follows the curve from A to B and groups directions as soon as the occasion arises.

To recapitulate, in this section several rasterization techniques have been introduced. The use of each of them depends on the object to be rasterized, on the lattice onto which rasterization is to be performed and on the neighbourhood relation defined onto it. If, as in most cases, there is no requirement for $l_{\leq}$ connectedness, then it is much easier to use nearest neighbour rasterizations. Cellular rasterization is preferred if single points or generic objects have to be rasterized, while curves are preferably rasterized through grid intersection rasterization schemes. However, there are cases, for example when conditions on the smoothness of the rasterized object are imposed, or in interconnection network routing, where the $l_{\leq}$-connectedness of the rasterization is required. In these cases the last method presented is capable of finding an $l_{\leq}$-connected rasterization of the curve. In the next section, we will define an algorithm that rasterizes straight line segments.

6 Straight line rasterization

Although there is plenty of literature on straight line rasterization onto square planar lattices [8], only recently has line rasterization onto hexagonal lattices been introduced [41, 43]. In three-dimensional space, straight line drawing algorithms have been developed in the context of ray-tracing for following a ray through the scene to be rendered [1]. Lately there has been growing interest on line segment rasterization algorithms in n-dimensional space. Badouel and Wüthrich [3] presented a face connected algorithm which rasterized straight line segments onto n-dimensional hypercubes. In an independent work, Slater [36] published two algorithms that generated both the $n_{\leq}$-connected and the $1_{\leq}$-connected rasterization of straight line segments onto a hyperrectangular lattice.

The purpose of this section is to illustrate the definition of rasterization presented in the last chapter through the development of an algorithm for the rasterization of a straight line onto an $l_{\leq}$-connected hypercubic lattice. We shall start by introducing one such algorithm for the hypercubic unitary lattice Λ_u, i.e. such that its generators are all of unit length.

Let $P = (p_0, p_1, \ldots, p_{n-1})$ and $Q = (q_0, q_1, \ldots, q_{n-1})$ be two distinct hyperlattice points. Let $n_i = q_i - p_i$. The straight line from P to Q is the set of points $X = (x_0, x_1, \ldots, x_{n-1})$ such that $x_i = (q_i - p_i)t + p_i$, where $t \in [0, 1]$. As seen in the previous section, the parameter t introduces an ordering on the points of the straight line. For each dimension i, consider the straight line points P_{j,h_i} obtained for $t = \frac{h_j}{n_i}$, where $h_j = 1, \ldots, n_i$, and order these points in increasing order of their corresponding parameter value. The segment PQ results subdivided into n_i equal parts for each dimension i, and the points obtained on the straight line segment are ordered by increasing values of the parameter t to form a sequence $A = \{A_1, \ldots, A_r\}$ which in turn is used to build the $l_{\leq}$-connected path as in the previous section.

The algorithm outlined above can be efficiently implemented to perform integer calculations only, and to generate one hop for each of its steps. To do this, for each dimension i, an integer counter d_i is allocated ($i = 0, \ldots, n-1$). The values of n_i, together with the least common multiple $L = LCM(\{n_i\})$ are then computed. Let $n''_i = \frac{L}{n_i}$, and let $n'_i = 2n''_i$. The n'_is represent the increment that will be used for the counter d_i throughout the whole algorithm. The counters d_i are initialized to n''_i. At each step, the set D of the l smallest counter cells d_i is considered. Let d_{i_0} be the smallest counter cell, and let n'_{i_0} be the increment corresponding to it. First n'_{i_0} is added to d_{i_0}. Then the set D is reordered by increasing values of the counter cells. Again, the corresponding increment n'_{i_1} is added to the smallest d_i, until either all l elements of D have been incremented, or the smallest d_i is a counter cell that has already been incremented in the current step. Once one of these two conditions is fulfilled, the directions corresponding to the elements of D that have been incremented in the step are grouped together in one single hop. The steps are repeated until all counters d_i have reached the value of $2L + n''_i$, which is equivalent to the condition $\forall i, d_i \geq 2L$. To avoid missing the endpoint Q, whenever a counter d_i is greater than or equal to $2L$, it cannot contribute any more to the hops, and it is therefore removed from the list of the incrementable d_i counters. This condition is unnecessary if there is the need to draw an "infinite" line. Whenever two counter cells d_i have equal values in a certain step, an arbitrary choice can be made: here the direction corresponding the lowest dimension (i.e. corresponding to the smallest i) is considered first (as if it was smaller), unless it has already been incremented in the current step in which case it is considered to be bigger than all non-incremented counter cells equal to it.

Note that the initialization of the counters d_i is important, since otherwise in the first step all d_i would be equal to zero, and this would imply an arbitrary choice of the directions of the first step.

The algorithm translates into the following pseudocode:

$\boxed{\text{A}}$ Algorithm:

```
• Initialization:
      $\forall i,\ n_i \leftarrow q_i - p_i$ END$\forall$
      $L \leftarrow \text{LCM}_i(n_i)$
      $\forall i\ n''_i \leftarrow \frac{L}{n_i}\ n'_i \leftarrow 2 \cdot n''_i$ END$\forall$.
• Translation:
      $\forall i,\ d_i \leftarrow n''_i$ END$\forall$
• Loop:
      WHILE $(\forall i,\ d_i \leq 2L)$
         Consider the $l$ minimum $d_i$,
         and order them in the sequence $d_{i_0} \leq \ldots \leq d_{i_{l-1}}$
         IF$(d_{i_k} + n'_{i_k} \geq d_{i_l},\ \forall k < l)$
            $\forall k < l$,
               $d_{i_k} \leftarrow d_{i_k} + n'_{i_k}$
               $x[i_k] \leftarrow x[i_k] + 1$
            END$\forall$
         ENDIF
         ELSE
            Let $m$ be the minimum $d_{i_r} + n'_{i_r}$, and let $l'$
            be the maximum $i_k$ such that $d_{i_{l'}} \leq m$:
            $\forall k \leq l'$,
               $d_{i_k} \leftarrow d_{i_k} + n'_{i_k}$
               $x[i_k] \leftarrow x[i_k] + 1$
            END$\forall$
         ENDELSE
         Discard directions such that $d_i > 2L$
         Write $X = (x[0], \ldots, x[n-1])$
      ENDWHILE
```

The algorithm presented above is a little slower than traditional algorithms developed for the 1- and n-connected cases. This is due to the inherent complexity of $l_{\leq}$-connectedness. In case 1- and n-connectedness are required, it is thus better to use the already mentioned algorithms.

The algorithm presented above can be extended trivially to a generic hypercubic lattice Λ_c by using an affine transform for transforming the points of the generic hypercubic lattice into points of a unitary hypercubic lattice Λ_u and transforming back the resulting rasterization through the inverse affine transform. More generally, the algorithm above can be trivially extended to hyperrectangular lattices. In fact, given a hyperrectangular lattice Λ_r and a hypercubic lattice Λ_c, there is an affine transformation A that converts the generators of Λ_r into the generators of Λ_c. To compute a hyperrectangular $l_{\leq}$-connected straight path between two points P and Q in Λ_r, it suffices again to compute an $l_{\leq}$-connected straight path from $A(P)$ to $A(Q)$, and to transform the resulting rasterization points through the inverse of A.

Finally, a small remark has to be made on the generation of the rasterization of a straight line segment from P to Q onto a non-orthogonal lattice Λ. In this case also, there exists a simple affine transformation $\overline{A}$ mapping the generators of Λ into the generators of Λ_u. As for all affine transformations, this transformation guarantees that lattice points will be mapped into lattice points, and that both grid hyperplanes and segment midpoints will be preserved. To find the 1-connected rasterization of a segment onto a generic lattice Λ, the 1-connected rasterization of the segment from $\overline{A}(P)$ to $\overline{A}(Q)$ can be computed and transformed back through the inverse of $\overline{A}$. The resulting rasterization points coincide with the points that would have been reached by applying grid intersection rasterization directly on the grid hyperplanes defined as in the previous section.

7 Conclusions

This paper has presented a first definition of rasterization in n-dimensional spaces using hyperlattices as a model for n-dimensional raster output devices. The model used does not require the lattice generators to be orthogonal, allowing thus non-orthogonal rasterization and sampling operations in higher order spaces. Rasterization schemes have been linked to the degree of connectivity required by the rasterized object in the discrete space. Such links will permit in future some control of the smoothness of the generated curve before the rasterization process begins. Both a generic algorithm for the rasterization of a curve, and an algorithm for the generation of a straight line segment have been presented: each of these algorithms generates the degree of smoothness required in the output.

The immediate application of this work in Computer Graphics lies in the visualization of n-dimensional datasets from scientific and experimental data, as well as in the development of the theory necessary for the display and representation of n-dimensional virtual worlds. In Image Processing the main contribution of this study lies in the allowed non-orthogonality of the data acquisition process through the scanning device. For example, the theory developed allows scanning devices, such as NMR and PET (Positron-Emission Tomography) devices, to acquire data along non-orthogonal directions, and to display such data in computer visualization systems, thus granting more flexibility to the data analysis.

The definition of a model for n-dimensional raster output devices and for the operation of rasterization allows the application of rasterization to new fields, whenever continuous objects have to be represented through discrete ones.

To be fully usable, however, the theory presented in this paper has to be refined: although curve rasterization is defined here, to the author's knowledge there are no studies available on surface and hypersurface rasterization. There are not even algorithms for the rasterization of hyperplanes in n-dimensional spaces. The link between lattice connectivity and the smoothness of rasterized objects is still more an intuition than a fact.

As the power of computing devices increases, so does the quest for visualizing progressively more complex spaces. Advances can be made only if the concepts of discrete n-dimensional geometry are clear. It is thus reasonable to assume that in the near future many of the unresolved issues mentioned above will be tackled and resolved.

Acknowledgments

The author wishes to thank John Ashby, David Duce and Olivia Wüthrich-Martone for the useful discussions on various issues of this paper. Robert Van Liere has contributed food for thought on n-dimensional visualization issues. Support and encouragement provided by my supervisors during the ERCIM fellowship, David Duce and Chris Wadsworth at RAL, Mario Rui Gomes at INESC and Fons Kujik and Paul Ten Hagen at CWI are also gratefully acknowledged.

Raytraced figures were produced using the *Rayshade* raytracer by Craig Kolb and Rod Bogart.

References

[1] Amanatides, J., and Woo, A. A fast voxel traversal algorithm for ray tracing. In *Proceedings of Eurographics '87* (Amsterdam, 1987), North Holland, pp. 3–9.

[2] Badouel, D., and Wüthrich, C. Face-connected line segment generation in an n-dimensional space. In *Graphics Gems III*, D. Kirk, Ed. Academic Press, Toronto, 1992, pp. 89–91,460.

[3] Badouel, D., Wüthrich, C. A., and Fiume, E. L. An analysis of connectivity of k-ary n-cube m-diag interconnection networks. Technical Report CSRI-266, Computer Systems Research Institute, University of Toronto, Toronto, 1992.

[4] Bartels, R. H., Beatty, J. C., and Barsky, B. A. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Morgan Kaufman, Los Altos, CA., 1987.

[5] Beddow, J. Shape coding of multidimensional data on a microcomputer display. In *Proceedings of IEEE Visualization '90* (Los Alamitos, CA., 1990), IEEE Computer Society Press, pp. 238–246.

[6] Berger, M. *Geometry*, vol. I–II. Springer Verlag, Berlin, 1987.

[7] Binmore, K. *Mathematical Analysis – A Straightforward Approach.* Cambridge University Press, Cambridge, 1982.

[8] Bresenham, J. E. Algorithm for computer control of a digital plotter. *IBM System Journal*, 4 (1965), 25–30.

[9] Chernoff, H. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Asoociation* (1973), 361–368.

[10] Cohen, M., and Greenberg, D. A radiosity solution for complex environment. *ACM Computer Graphics 19*, 3 (July 1985), 31–40.

[11] Cook, R., and Torrance, K. A reflectance model for computer graphics. *ACM Transactions on Graphics 1*, 1 (1982), 7–24.

[12] Coxeter, H. S. M. *Introduction to Geometry.* John Wiley and Sons, New York, N.Y., 1969.

[13] Dieudonné, J. *Foundations of Modern Analysis.* Academic Press, New York, N.Y., 1960.

[14] Drebin, R., Carpenter, L., and Hanrahan, P. Volume rendering. *ACM Computer Graphics 22*, 4 (Aug. 1988), 65–74.

[15] Dubois, E. The sampling and reconstruction of time-varying imagery with application in video systems. *Proceedings of the IEEE 73*, 4 (Apr. 1985), 502–522.

[16] Eker, S. *Formal Foundations for the Design of Rasterization Algorithms and Architectures.* PhD thesis, University of Leeds, School of Computer Studies, Leeds, Sept. 1990.

[17] Fiume, E. L. *The Mathematical Structure of Raster Graphics.* Academic Press, New York, N.Y., 1989.

[18] Forrest, A. Antialiasing in practice. In *Fundamental Algorithms for Computer Graphics*, R. Earnshaw, Ed., vol. F-17 of *NATO ASI Series.* Springer Verlag, Berlin, 1985, pp. 113–133.

[19] Françon, J. Topologie discrète et algorithmique graphique. *Bigre+Globule*, 61-62 (Apr. 1989).

[20] Franklin, J. W. *Matrix Theory.* Prentice Hall, Englewood Cliffs, N.J, N.Y., 1968.

[21] Fuller, G. *Analytic Geometry.* Addison-Wesley, Reading, MA., 1980.

[22] Gonczarowski, J. Fast generation of unfilled and filled outline characters. In *Raster imaging and digital typography* (Cambridge, 1989), J. Andrè and R. D. Hersch, Eds., Cambridge University Press, pp. 261–270.

[23] Herman, G. Oriented surfaces in digital spaces. *CVGIP: Graphical Models and Image Processing 55*, 5 (Sept. 1993), 381–396.

[24] Kim, C. E. Three-dimensional digital planes. *IEEE Transactions on Pattern Anasysis and Machine Intelligence PAMI-6*, 5 (Apr. 1984), 639–645.

[25] Klette, R. The m-dimensional grid point space. *Computer Vision, Graphics and Image Processing 30* (1985), 1–12.

[26] Kovalevsky, V. A. Finite topology as applied to image analysis. *Computer Vision, Graphics and Image Processing 46* (1989), 141–161.

[27] Luby, M. G. Grid geometries which preserve properties of euclidean geometry: A study of line drawing algorithms. In *Theoretical Foundations of Computer Graphics and CAD*, R. Earnshaw, Ed., vol. F-40 of *NATO ASI Series.* Springer Verlag, Berlin, 1988, pp. 397–432.

[28] Mihalisin, T., Schweiger, J., and Timlin, J. Hierarchical multivariate visualization. In *Proceedings of Interface 92* (1992).

[29] Montani, C., and Scopigno, R. Rendering volumetric data using the sticks representation scheme. *ACM Computer Graphics 24*, 5 (Nov. 1990), 87–93.

[30] Nishita, T., and Nakamae, E. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. *ACM Computer Graphics 19*, 3 (July 1985), 22–30.

[31] Noll, A. A computer technique for displaying n-dimensional hyperobjects. *Comm. of the ACM 10*, 8 (Aug. 1967), 469–473.

[32] Rogers, C. A. *Packing and Covering.* Cambridge University Press, Cambridge, 1964.

[33] Rosenfeld, A. Connectivity in digital pictures. *Journal of the ACM 17* (1970), 146–160.

[34] Rosenfeld, A. Arcs and curves in digital pictures. *Journal of the ACM 20*, 1 (1973), 81–87.

[35] Shilov, G. *An Introduction to the Theory of Linear Spaces.* Prentice Hall, London, 1961.

[36] Slater, M. Tracing a ray through uniformly subdivided n-dimensional space. *The Visual Computer 9* (1992), 39–46.

[37] Somerville, D. *An Introduction to the Geometry of N Dimensions.* Dover Publications, Inc., New York, 1958.

[38] van Wijk, J., and van Liere, R. Hyperslice - visualization of scalar functions of many variables. In *Proceedings of IEEE Visualization '93* (1993).

[39] Wu, L.-D. On the chain code of a line. *IEEE Transactions on Pattern Anasysis and Machine Intelligence PAMI-4*, 3 (1982), 347–353.

[40] Wüthrich, C. A. An analysis of digitalization algorithms for non-linear curves. In *Computer Graphics and Mathematics*, I. Herman, B. Falcidieno, and C. Pienovi, Eds., Proceedings of the Eurographics Workshop on Computer Graphics and Mathematics. Springer Verlag, Berlin, 1992, pp. 285–297.

[41] Wüthrich, C. A., and Stucki, P. An algorithmic comparison between square- and hexagonal-based grids. *CVGIP: Graphical Models and Image Processing 53*, 4 (July 1991), 324–339.

[42] Wüthrich, C. A., Stucki, P., and Ghezal, A. A frequency domain analysis of square and hexagonal based images. In *Raster imaging and digital typography, Proceedings of RIDT'89* (Cambridge, 1989), J. Andrè and R. Hersch, Eds., Cambridge University Press, pp. 261–270.

[43] Yong-Kui, L. The generation of straight lines on hexagonal grids. *Computer Graphics Forum 12*, 1 (1993), 27–31.

[44] Young, F., Kent, D., and Kuhfeld, W. *Dynamic Graphics for Exploring Multivariate Data.* Wadsworth Inc., 1988.