

Boundaries in Digital Spaces

GABOR T. HERMAN*

ABSTRACT. Intuitively, a boundary in an N -dimensional digital space is a connected component of the $(N - 1)$ -dimensional surface of a connected object. In this paper we make these concepts precise, and show that the boundaries so specified have properties that are intuitively desirable. We provide some efficient algorithms for tracking such boundaries. We illustrate that the algorithms can be used, in particular, for computer graphic display of internal structures (such as the skull and the spine) in the human body based on the output of medical imaging devices (such as CT scanners). In the process some interesting mathematical results are proven regarding “digital Jordan boundaries,” such as a specification of a local condition that guarantees the global condition of “Jordanness.”

2000 AMS Classification: 54A99, 52C99, 68R99, 68W05

Keywords: Boundaries, surfaces, tracking, algorithms, digital topology, geometry, digital space.

1. THE BOUNDARY TRACKING PROBLEM

1.1. A sample application of boundary tracking.

Many imaging devices will produce estimated values of a physical quantity at certain points (referred to as *grid points*) in three-dimensional space. For example, a CT (computerized tomography) scanner estimates the X-ray attenuation coefficient inside a human body at points of a three-dimensional rectangular grid. When displaying the results of such an estimation, we usually use a sequence of two-dimensional images, such as that shown in Figure 1, in which bone appears light, softer tissues appear grey, and air appears dark.

*The research of the author is currently supported by NIH grant HL070472 and NSF grant DMS0306215.

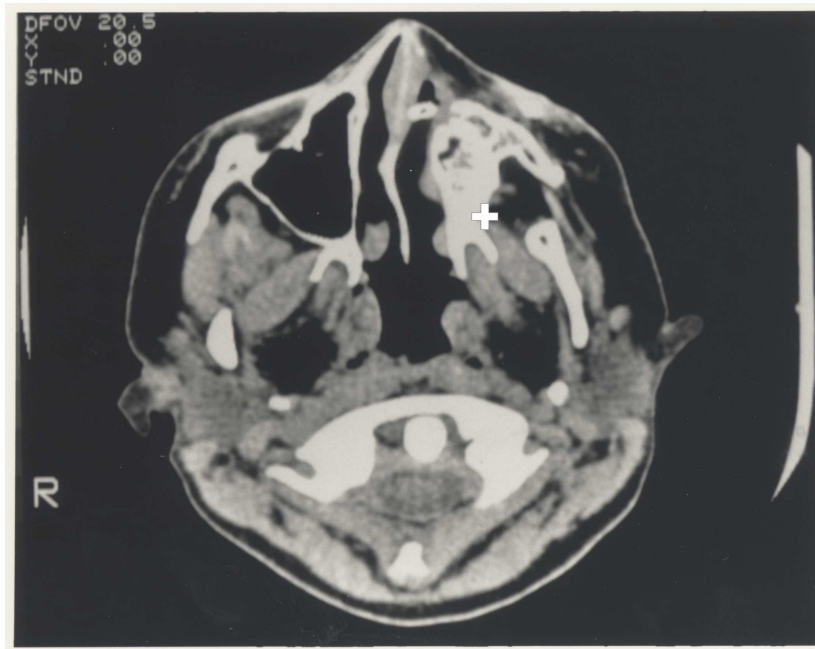


FIGURE 1. A slice of a CT scan of the head of a trauma victim.

In Figure 1 a cursor is placed at the right edge of a large piece of bone in the upper part of the image. Since this CT scan was taken of a trauma victim, an important question to ask is: are all other bones which are normally “connected” to this main piece of bone still connected for this patient? For example, the biggish piece of bone just to the right and below the cursor (it is the mandible) is not connected to any other bone in this slice, but it should be connected to the rest of the skull (at our resolution) somewhere in the whole three-dimensional array. By this we mean that somewhere in the three-dimensional space the mandible should be so close to the rest of the skull that, at the resolution of our imaging process, the two would be judged to be “connected.” Making precise the sense in which the word “connected” is used here and the analysis of the resulting connectivity properties in three-dimensional images is the main topic of the material that follows.

We turn aside for a moment to introduce some notation. We use \mathcal{R} to denote the set of real numbers and \mathcal{R}^N to denote the set of N -dimensional (row) vectors, all of whose components are real numbers. If $v \in \mathcal{R}^N$, v_n denotes the n th component of v , for $1 \leq n \leq N$. The *norm* of a vector v is denoted by $\|v\|$. We use the algebraic notion of a *vector* in \mathcal{R}^N and the geometric notion of a *point* in an N -dimensional Euclidean space quite interchangeably, and actually refer to \mathcal{R}^N as the N -dimensional (Euclidean) space. We use Z to denote the set of all integers. For any positive real number δ and for any positive integer

N , we define

$$(1.1) \quad \delta Z^N = \{(\delta c_1, \dots, \delta c_N) \mid c_n \in Z, \text{ for } 1 \leq n \leq N\}.$$

Specifically, we use Z^N to abbreviate $1Z^N$.

Let G be any set of points (a *grid*) in \mathcal{R}^N . The *Voronoi neighborhood* in G of any element g of G is defined as

$$(1.2) \quad N_G(g) = \{v \in \mathcal{R}^N \mid \text{for all } h \in G, \|v - g\| \leq \|v - h\|\}.$$

In other words, the Voronoi neighborhood of g consists of all those vectors which are not nearer to any other point of G than they are to g . For example, the Voronoi neighborhood of (c_1, c_2) in Z^2 is

$$(1.3) \quad \left\{ (v_1, v_2) \in \mathcal{R}^2 \mid \max(|v_1 - c_1|, |v_2 - c_2|) \leq \frac{1}{2} \right\}.$$

This is a closed square with side-length 1 centered at the point (c_1, c_2) . When perceived as a set of points in \mathcal{R}^2 , δZ^2 is referred to as a *square grid*. The Voronoi neighborhoods in a grid in \mathcal{R}^2 are referred to as *pixels*.

We now return to the way in which the two-dimensional image of Figure 1 was produced. Let us assume that the X-ray attenuation coefficients have been estimated at grid points with integer coordinates (c_1, c_2, c_3) , $1 \leq c_1 \leq I$, $1 \leq c_2 \leq J$, $1 \leq c_3 \leq K$ and that Figure 1 is the k th of these K slices. The region over which the images are defined is the union of the pixels associated with those $(c_1, c_2) \in Z^2$ for which $1 \leq c_1 \leq I$ and $1 \leq c_2 \leq J$. The grey value assigned to the pixel associated with one of these (c_1, c_2) s is proportional to the estimated X-ray attenuation coefficient at (c_1, c_2, k) .

The overall aim of the type of work on which we are reporting is the display and analysis of selected objects based on the estimated values of a physical quantity at grid points in three- (and sometimes higher) dimensional space. Examples for bones in the head are shown in Figure 2. On the top we show a computer graphic display of the boundary of the connected piece of bone which is indicated by the cursor in Figure 1. (The cursor location in the top image in Figure 2 corresponds to the cursor location in Figure 1.) We now see that the mandible is disconnected from this bone in the whole three-dimensional data set. The boundary of the mandible can be extracted separately, and then the two boundaries can be displayed together in their correct relative locations, as is shown in the bottom image of Figure 2.

1.2. A methodology for extracting object boundaries.

We now discuss a particular methodology for extracting boundaries of objects based on values assigned to points of the grid Z^3 in \mathcal{R}^3 . For $\delta > 0$, δZ^3 is referred to as a *cubic grid*. The Voronoi neighborhoods associated with a grid in \mathcal{R}^3 are referred to as *voxels*. The voxel associated with the element (c_1, c_2, c_3) of Z^3 is the closed unit cube $\{v \in \mathcal{R}^3 \mid \max(|v_1 - c_1|, |v_2 - c_2|, |v_3 - c_3|) \leq \frac{1}{2}\}$. The tessellation of \mathcal{R}^3 into voxels of a cubic grid is sometimes referred to as a *cube*. In Figure 3, we illustrate a grid point g of Z^3 together with all other

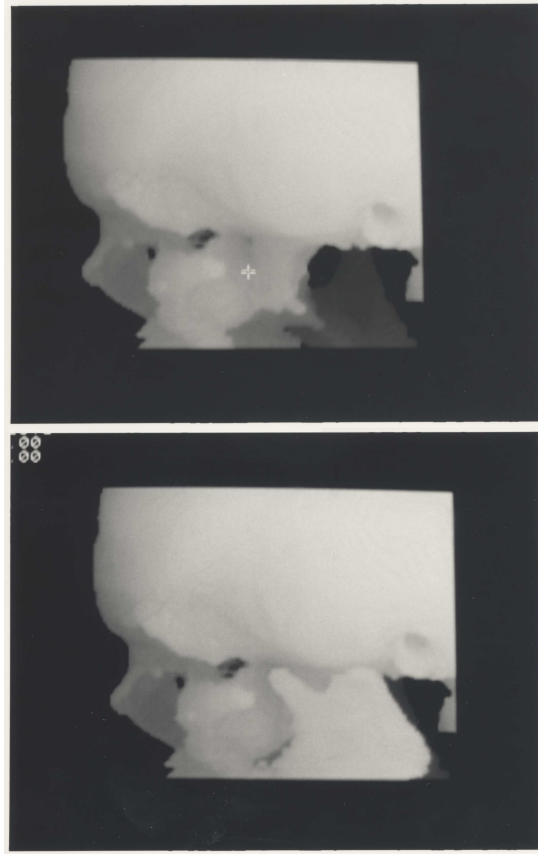


FIGURE 2. Computer graphic displays of detected boundaries of bones in the head, based on the data set which contains the slice shown in Figure 1.

grid points which lie in the $2 \times 2 \times 2$ cube whose center is g , as well as the Voronoi neighborhood of g .

Now suppose that we have some methodology to determine which grid points belong to what kind of material. For example, in CT bone attenuates X-rays more than any other type of tissue in the human body and so we may say that if the value assigned to a grid point is above a certain value, then that grid point is in bone. So, as an approximation, the part of space that is occupied by bone may be considered to be the union of the voxels associated with those pixels for which the estimated X-ray attenuation coefficient is above the threshold for bone. This collection of voxels may not necessarily form a connected subset of three-dimensional space: some pieces of bone may be disconnected from the rest and some voxels may be included because noise in the estimation process

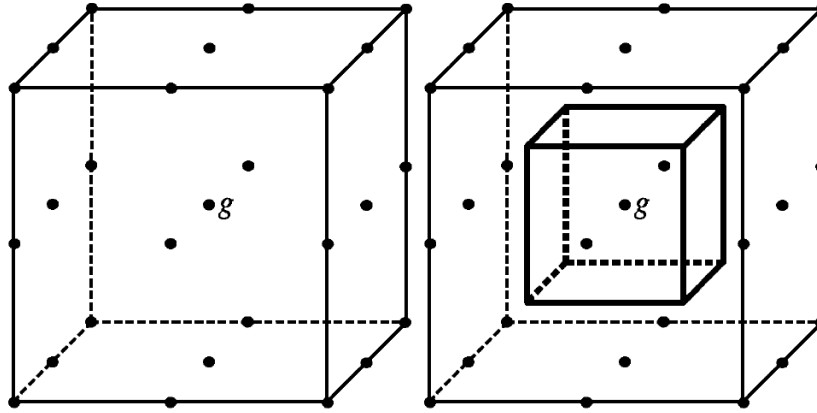


FIGURE 3. On the left we show a point g of the cubic grid Z^3 together with the other 26 grid points which lie in the $2 \times 2 \times 2$ cube whose center is g . On the right we show (using heavy lines) the Voronoi neighborhood of g (a voxel).

caused us to identify a grid point as being in bone, when in fact it is not. Let us say that the “object” whose boundary we wish to display is a *component* of the set of points occupied by the “bone” voxels.

Even this specification is not precise enough to describe the intuitive notion of “a boundary” of an object. This is because an object of the kind we described in the previous paragraph may have cavities inside it (just look at Figure 1; the inside of bones contains lots of less X-ray attenuating tissue and may well be identified as a result of thresholding as “not bone”) and so it may have multiple boundaries (one exterior one and possibly many interior ones). Let us say that our task is to identify exactly one of these boundaries.

How do we specify which one? One way is to point at a boundary face, that is at a face which separates a bone voxel from a not-bone voxel, and say that we wish to display that boundary of the object which contains that boundary face. (At this point, it is not even clear that this specification is legitimate. Is the “boundary containing a boundary face” a well defined concept? We will show that it is; we will set up an environment in which for any boundary face there will be one and only one boundary containing it.)

An intuitively useful picture is the following. We consider the cuberille, the tessellation of three-dimensional space into cubic voxels. A finite number of these voxels are occupied by sugar cubes of just the right size. We point at an uncovered face of one of these sugar cubes (i.e., a face such that the voxel on the other side of it is not occupied by a sugar cube) and ask that there be delivered to us the boundary surface which contains that face. The problem is to design an algorithm which is guaranteed to do this for all possible arrangements of the sugar cubes.

At this point a topologist may reasonably ask: Are we not reinventing the wheel here? After all, the boundary of a set is a classical topological concept. The problem is that there is no topology (in the classical sense) for the cuberille (and, indeed, for the other digital spaces to be discussed below) that would result in the same sets being “connected” according to the topological definition as those that would be considered connected in the digital environment. This somewhat surprising (and disturbing) fact is stated precisely and proved in Section 4.3 of [13]. It is fascinating that, in spite of this, we are able to do many things similar to topics that are normally considered in classical topology: the Jordan curve theorem, simple connectedness, etc. However, we are going to do these things not as a consequence of classical topology, but rather as a self-standing theory for digital spaces.

The approach presented here is one of a number of different alternatives that can be taken to insert topological ideas into the digital environment. Examples of alternatives can be found in [4], [16], [17], [19], [20] and in their references.

From an even more general point of view, this paper concentrates on one family of methods for producing computer graphic displays of three-dimensional objects, as illustrated in Figure 2. Indeed there are many alternatives, for three surveys see [9], [10] and Chapter 1 of [25]. Broadly speaking, the approaches taken fall into two categories: the so-called “volumetric” and “boundary-based” approaches. What is presented here is an example of the latter, but it is by no means the only one. As discussed above, our approach produces a single boundary of a single component, specified by a user-supplied initial face in the boundary. As opposed to this, the so-called “marching cubes algorithm” [22] automatically detects all surfaces of all components. It is not clear that this really is an advantage: after all, in practical visualization only a few of these surfaces should be displayed (as in Figure 2). Besides, the mathematical theorems proven below regarding the boundaries produced by our algorithms having well-defined interiors and exteriors allow us to detect automatically all boundaries by repeated applications of the algorithm for the tracking of a single boundary. In fact, as discussed in [21], the set of all surfaces produced by the marching cubes algorithm can also be produced by creating “continuous analogs” of the boundaries produced by the algorithms discussed below, and the total process of tracking all the boundaries (in our sense) and creating their continuous analogs requires less computer time than what is needed by the marching cubes algorithm, provided only that the number of grid points needed for the objects of interest (i.e., $I \times J \times K$) is large; and this is even more so if the values originally estimated for these grid points are noisy (i.e., inaccurate).

1.3. Flies in Flatland.

Before attacking the three-dimensional problem, we consider its simpler two-dimensional analog. Following Abbott [1], we refer to the two-dimensional plane as Flatland and to the computing device which we hope will deliver us the required boundary surfaces as a Flat Fly. As an example, consider a

configuration of two-dimensional sugar cubes in Flatland shown in Figure 4 with a Flat Fly on one of the flat faces (i.e., on an edge) of one of these flat sugar cubes.

Algorithm 1.1. *Algorithm for Flat Flies*

- (1) Dirty the face on which you are standing.
- (2) Crawl onto the face which meets the one on which you are standing at the vertex in front of you.
- (3) See if it is dirty.
 - (a) If it is, fly away.
 - (b) If it is not, start again at Instruction (1).

This algorithm actually does something that can be made precise. To achieve that precision, we need to introduce some mathematical terminology.

1.4. Components determined by binary relations.

Let F be any set and ρ be a *binary relation* on F . If $(c, d) \in \rho$, then we say that c is ρ -adjacent to d and that d is ρ -adjacent from c and, in case ρ is a *symmetric relation* (meaning that $(c, d) \in \rho$ if, and only if, $(d, c) \in \rho$), that c and d are ρ -adjacent. In case F is finite, we define for any d in F its *indegree* (denoted by $in_p(d)$) as the number of elements a of F such that $(a, d) \in \rho$.

In the case when $F = Z^N$, we will be repeatedly dealing with two symmetric binary relations, α_N and ω_N , defined as follows.

$$(1.4) \quad (c, d) \in \alpha_N \Leftrightarrow (c \neq d \text{ and, for } 1 \leq n \leq N, |c_n - d_n| \leq 1).$$

$$(1.5) \quad (c, d) \in \omega_N \Leftrightarrow \sum_{n=1}^N |c_n - d_n| = 1.$$

For $N = 2$, $(c, d) \in \alpha_2$ means that the associated pixels have a nonempty intersection, but are not identical, and $(c, d) \in \omega_2$ means that the associated pixels have exactly one edge in common. These are often used notions and go under a number of commonly used names: ω_2 is referred to both as the edge-adjacency and as the 4-adjacency, while α_2 is referred to both as the edge-or-vertex-adjacency and as the 8-adjacency.

For any c and d in a subset A of F , we call a sequence $\langle c^{(0)}, \dots, c^{(K)} \rangle$ of elements of A a ρ -path in A connecting c to d , if $c^{(0)} = c$, $c^{(K)} = d$ and, for $1 \leq k \leq K$, $c^{(k-1)}$ is ρ -adjacent to $c^{(k)}$. If such a path exist, we say that c is ρ -connected to d in A . A nonempty subset A of F is said to be a ρ -connected if, for any c and d in A , c is ρ -connected in A to d .

If ρ -connectedness in A is symmetric, then it is an *equivalence* relation. (Note that if ρ itself happens to be symmetric, then ρ -connectedness in A is automatically symmetric.) Hence, it partitions A into ρ -components (i.e., into nonempty ρ -connected subsets which are not proper subsets of any other ρ -connected subset of A). Referring to Figure 4, we see that there are three ω_2 -components of the set of grid points whose pixels are painted gray: one has

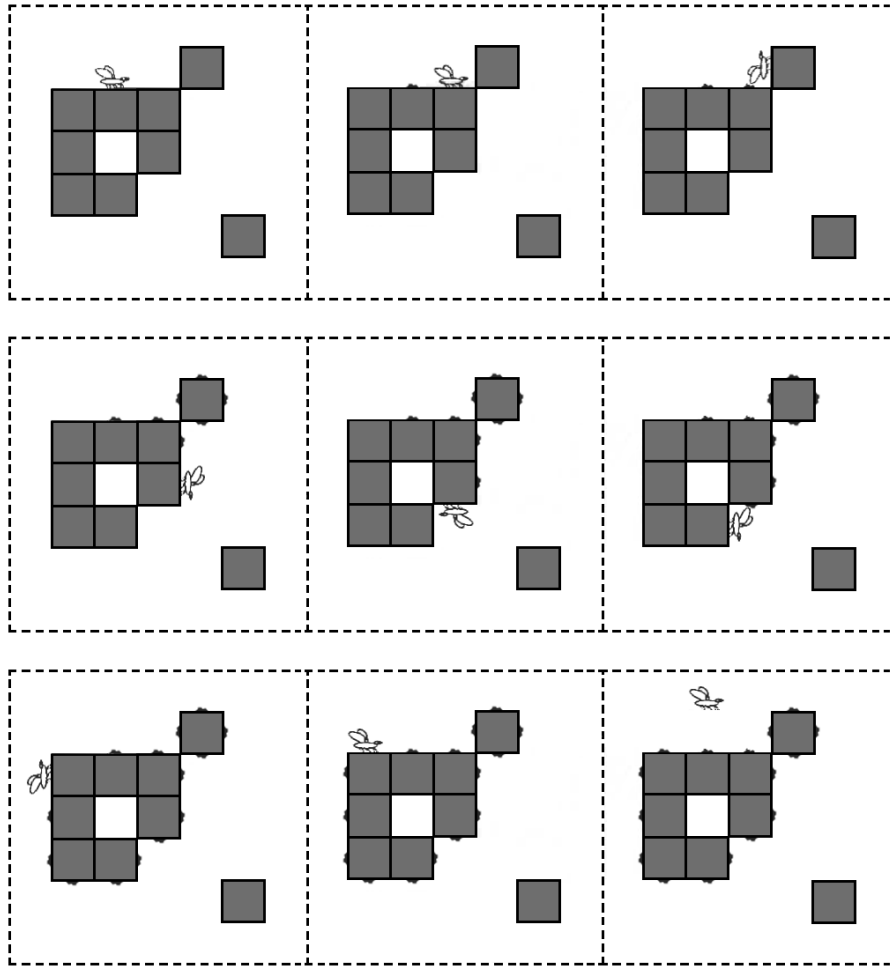


FIGURE 4. Illustration of boundary tracking in two dimensions by the Algorithm for Flat Flies when a flat fly is initially placed on a flat face (i.e., on an uncovered edge of an occupied pixel): the first row is the beginning, the second row is the middle, and the third row is the end of the execution of the algorithm.

seven element and the other two have one element each. On the other hand, there are only two α_2 -components.

We denote the *complement* $F - A$ of A in F by \overline{A} . In Figure 4, \overline{A} (the set of grid points whose pixels are not painted grey) is an α_2 -connected subset (and hence it has only one α_2 -component). However, it is not ω_2 -connected. In the

special case when $A = F$, we use the phrases ρ -path and ρ -connected instead of ρ -path in A and ρ -connected in A .

For any binary relation ρ on a set F , we define its *transitive closure* as the binary relation ρ^* on F by: $(c, d) \in \rho^*$ if, and only if, c is ρ -connected to d . If ρ^* is symmetric, then we refer to ρ as an *adjacency* (on F). In this case F is partitioned into ρ -components. Note, in particular, that (for all $N \geq 1$) both α_N and ω_N are adjacencies on Z^N (and, for these adjacencies, there is only one component in the partition of Z^N).

1.5. What does a Flat Fly do?

We first specify the outcome of the Algorithm for Flat Flies in the specific case identified in Figure 4. The Flat Fly is initially put on the face which separates the grid point g (painted grey) from the grid point h (not painted grey). After a while it flies away (see Figure 4), leaving behind a bunch of faces that have been dirtied. This set of faces can be described as consisting of exactly those faces which are between a grid point painted grey that is α_2 -connected in the set of grey grid points to g and a grid point not painted grey that is ω_2 -connected in the set of not grey grid points to h .

This is, in fact, representative of the general behavior of the Flat Fly. We can say the following. Suppose that in the square grid Z^2 the pixels of a finite number of grid points are filled with flat sugar cubes and that a Flat Fly is put on top of one of these sugar cubes into a pixel which is not occupied by a sugar cube (see Figure 4). Let g be the name of the grid point of the sugar cube on top of which the Flat Fly is placed into the pixel of a grid point named h (see Figure 4).

Claim 1.2. *After a finite number of loops through Instructions (1), (2), and (3) of the Algorithm for Flat Flies, the Flat Fly will fly away. At that time, the set of dirty faces is exactly the set of those faces which are between a pixel associated with an element of the α_2 -component of the set of grid points with sugar cubes that contains g and a pixel associated with an element of the ω_2 -component of the set of grid points without sugar cubes that contains h .*

We do not give a proof of this claim in this section; such a proof will be a consequence of the general results that we will be deriving in the sections that follow. We will return to proving the claim at the appropriate point later in the paper. However, before going on to other things we state another claim that shows that the set of faces dirtied by the Flat Fly has some desirable properties. First we need to introduce some additional notation and terminology.

We first observe that the face onto which the fly is initially placed can be identified by the pair (g, h) of grid points on either side of it. Note that this is an ordered pair, interpreted as “the fly is put into h , with its feet touching g .” If we were to draw the Flat Fly on (h, g) , then its feet would be touching the same edge, but it would be hanging upside down. Similarly, every ordered pair of ω_2 -adjacent grid points can be thought of as such an edge with an orientation across it from the first grid point in the pair toward the second

and, conversely, all such oriented edges can be described by a unique element of ω_2 . Thus we make the fundamental observation that ω_2 has a dual purpose: it is an adjacency on Z^2 and its elements can be used as unique identifiers of elements of the set of all oriented faces between pixels of this square grid. We note immediately that, for arbitrary positive integer N , ω_N is an adjacency on the grid Z^N and its elements can be used as unique identifiers of elements of the set of all oriented (hyper-)faces between points of the grid. We now introduce two important concepts for the grid Z^N .

For any two subsets O and Q of Z^N we define the *boundary* between them by:

$$(1.6) \quad \vartheta(O, Q) = \{(c, d) \mid c \in O, d \in Q, (c, d) \in \omega_N\}.$$

Let A consist of the nonempty finite set of grid points which are filled with sugar cubes and let the Flat Fly be placed on (g, h) , where $g \in A$ and $h \in \overline{A}$. Let O be the α_2 -component of A containing g and Q be the ω_2 -component of \overline{A} containing h . Then Claim 1.2 says that the set of faces dirtied by the Flat Fly is exactly $\vartheta(O, Q)$.

Let $\langle c^{(0)}, \dots, c^{(K)} \rangle$ be an ω_N -path and S be any subset of ω_N . We say that $\langle c^{(0)}, \dots, c^{(K)} \rangle$ *crosses* S , if there is a k , $1 \leq k \leq K$, such that either $(c^{(k-1)}, c^{(k)}) \in S$ or $(c^{(k)}, c^{(k-1)}) \in S$.

Claim 1.3. *Let A be any nonempty finite subset of Z^2 . Let O be an α_2 -component of A and Q be an ω_2 -component of \overline{A} , such that $\vartheta(O, Q)$ is not empty. Then there exist two uniquely defined subsets I and E of Z^2 , which have the following properties.*

- (i) $O \subseteq I$ and $Q \subseteq E$.
- (ii) $\vartheta(O, Q) = \vartheta(I, E)$.
- (iii) $I \cup E = Z^2$ and $I \cap E = \emptyset$. (\emptyset denotes the empty set.)
- (iv) I is α_2 -connected and E is ω_2 -connected.
- (v) Every ω_2 -path connecting an element of I to an element of E crosses $\vartheta(O, Q)$.

The proof of this claim is also delayed till later; in fact it will be proven in a much more general context. Here we discuss its relevance with respect to the Algorithm for Flat Flies. Since, according to Claim 1.2, the output of this algorithm is a boundary $\vartheta(O, Q)$ satisfying the premises of Claim 1.3, the latter claim says that there are two sets of grid points I and E , of which we will think intuitively as the interior and the exterior, which have certain properties reminiscent of the famous Jordan Curve Theorem [23]. According to that theorem, the set of all points in the plane which are not on a simple closed curve can be partitioned into two subsets, one may be called the interior and the other the exterior. Both of these are connected sets in the sense that we can get from any point of the interior to any other point by drawing continuously a curve between them which never leaves the interior (and similarly for the exterior), but one cannot draw continuously a curve from a point in the interior to a point

in the exterior which does not contain at least one point of the simple closed curve, which is in fact the boundary between the interior and the exterior.

The boundary that is the output of the Algorithm for Flat Flies shares important properties with simple closed curves in \mathcal{R}^2 : its interior and exterior partition the whole space (iii), both are connected in some sense (iv), but one cannot get from the interior to the exterior without crossing the boundary (v) which, in fact, is the boundary between the interior and the exterior (ii). (Mathematically speaking there is also a difference: a simple closed curve is a subset of the plane, and the Jordan Curve Theorem says that the complement of the curve, rather than the whole plane, has precisely two components. Since our boundary is a subset of ω_2 , rather than of Z^2 , it is possible to demand that the interior and exterior partition the whole of Z^2 , if anything, a more attractive-sounding aim than that of the classical theorem.)

It is important to recognize that, in spite of this formal similarity to the Jordan Curve Theorem, Claim 1.3 is not a result about the continuous plane \mathcal{R}^2 but a thoroughly digital theorem. Notions of “connectedness,” “path,” and “crosses” have all been defined for the square grid Z^2 and although they are analogous to corresponding notions in \mathcal{R}^2 , the analogy breaks down sometimes. It is the difficulty of applying the continuous notions directly to the digital environment to prove claims regarding algorithms operating in discrete domains which motivates us to develop a geometry directly for digital spaces.

For example, it is tempting to reinterpret the boundary output by the Algorithm for Flat Flies as the point set in \mathcal{R}^2 which is the union of all the points in all the dirtied faces. However, in spite of the discrete Jordan curve properties of this boundary (Claim 1.3), the Jordan Curve Theorem is not applicable to the corresponding point set in \mathcal{R}^2 , since it does not form a simple closed curve. (It “touches itself” at one vertex and so it is not a simple curve.)

Having said this, we must admit that the argument is only half convincing. It is all very well that we can give a self-contained theory in digital spaces with nice theorems that resemble those of famous results of continuous mathematics; nevertheless, an interpretation in the underlying continuous space is unavoidably needed for some applications. For example, no physician would think of the boundaries shown in Figure 2 as collections of ordered pairs of voxels; rather they would be perceived as continuous biological surfaces. Hence, it is not really desirable to have boundaries which touch themselves in the fashion of the boundary consisting of the dirtied faces at the end of Figure 4.

While we are pondering this, we may as well ask the following question: Is it reasonable that two pixels which only share a vertex are considered connected to each other? This came about quite naturally in the way the Flat Fly crawled about on the sugar cubes in Figure 4, but - looking at the result - the connection is rather suspect. Would it not be better to use only ω_2 to define connectivity and avoid such flimsy looking connections?

Once we raise this objection on physical grounds, we also see that the use of α_2 is not particularly elegant even from a mathematical point of view. Why do we use different types of components for grid points with sugar cubes and for

those without in Claim 1.2? Why do we use different types of components for A and for \bar{A} in Claim 1.3? Since in (v) of Claim 1.3 only ω_2 is used, would it not be much nicer to replace α_2 by ω_2 everywhere in Claim 1.3? Unfortunately, the resulting claim would not be true; one can easily produce counterexamples [11]. Thus, we need ω_2 (it is used to describe the edges which make up the boundary), and we need a broader adjacency for the validity of the rather desirable Claim 1.3.

1.6. On the cuberille.

In three-dimensional space we would like to have an Algorithm for Fat Flies (in Figure 5 we show one placed on a sugar cube), which would have behavior resembling that of Flat Flies as expressed in Claims 1.2 and 1.3.

We first need to decide on the appropriate analog of α_2 for the case of a cuberille. One candidate is α_3 . The geometrical interpretation of α_3 is: two points of the cubic grid Z^3 are α_3 -adjacent if, and only if, the corresponding voxels have either exactly one face, or one edge, or one vertex in common (see Figure 3). The question is: for the purpose of the three-dimensional analog of Claim 1.2, do we want to consider two sugar cubes connected if they share only a vertex? Even in two dimensions such a connectivity was undesirable, but was forced upon us in order to have Claim 1.3. In three dimensions we need ω_3 to define the boundary surface, but also something wider to obtain an analog of Claim 1.3. The difference is that now we have an alternative, another analog of α_2 in which two grid points are adjacent if the corresponding voxels have either exactly one face in common or exactly one edge in common.

Formally, for any positive integer N , we define the adjacency δ_N on Z^N as follows. For any c and d in Z^N ,

$$(1.7) \quad (c, d) \in \delta_N \Leftrightarrow \left[(c, d) \in \alpha_N \text{ and } \sum_{n=1}^N |c_n - d_n| \leq 2 \right].$$

Clearly, $\delta_2 = \alpha_2$, and so δ_3 and α_3 are equally legitimate three-dimensional analogs of α_2 . However, δ_3 is intuitively preferable over α_3 , inasmuch as it does not allow adjacency by vertices only; see Figure 6. (For obvious reasons, ω_3 has been referred to both as face-adjacency and as 6-adjacency, δ_3 has been referred to both as face-or-edge-adjacency and as 18-adjacency, and α_3 has been referred to both as face-or-edge-or-vertex-adjacency and as 26-adjacency.) In view of this, our aim is as follows.

Problem 1.4. *Design an Algorithm for Fat Flies that will have the following property. Suppose that in the cubic grid Z^3 (see Figure 3) the voxels of a finite number of grid points are filled with sugar cubes and that a Fat Fly is put on top of one of these sugar cubes into a voxel which is not occupied by a sugar cube (see Figure 5). Let g be the name of the grid point of the sugar cube on top of which the Fat Fly is placed into the voxel of a grid point named h . Let O be the δ_3 -component of the set of grid points with sugar cubes which contains g and Q be the ω_3 -component of the set of grid points without sugar cubes which*

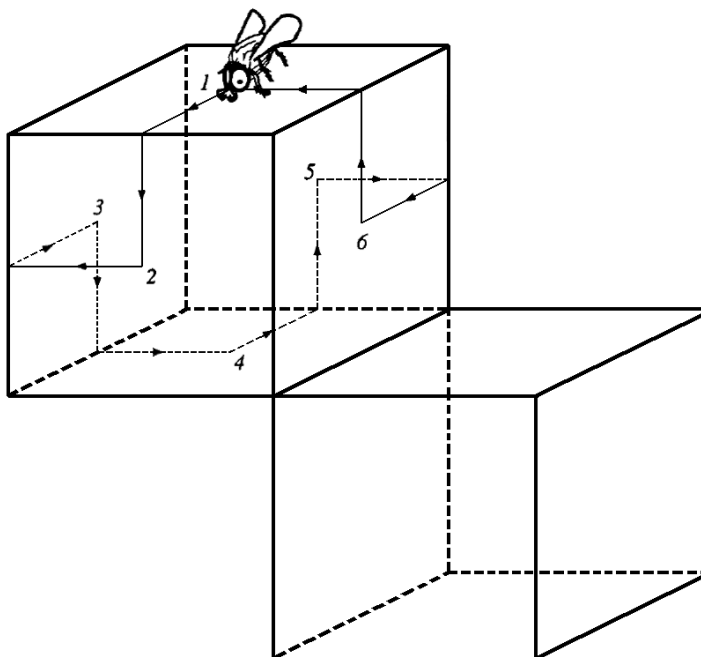


FIGURE 5. The route of a Model 1 Fat Fly on one of a pair of sugar cubes.

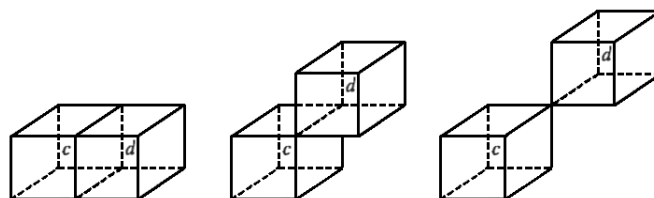


FIGURE 6. The grid points c and d on the left are α_3 -, δ_3 -, and ω_3 -adjacent; those in the middle are α_3 - and δ_3 -adjacent but not ω_3 -adjacent; those on the right are α_3 -adjacent but are not δ_3 -adjacent or ω_3 -adjacent.

contains h . After a finite number of steps of the Algorithm for Fat Flies, the Fat Fly should fly away and, at that time, the set of faces that have been dirtied should be exactly $\vartheta(O, Q)$.

The output $\vartheta(O, Q)$ of such an algorithm is an intuitively desirable boundary, due to the truth of the following claim (whose proof is also postponed

until we have reached the appropriate point in the development of our general theory).

Claim 1.5. *Let A be any nonempty finite subset of Z^3 . Let O be a δ_3 -component of A and Q be an ω_3 -component of \overline{A} such that $\vartheta(O, Q)$ is not empty. Then there exist two uniquely defined subsets I and E of Z^3 with the following properties.*

- (i) $O \subseteq I$ and $Q \subseteq E$.
- (ii) $\vartheta(O, Q) = \vartheta(I, E)$.
- (iii) $I \cup E = Z^3$ and $I \cap E = \emptyset$.
- (iv) I is δ_3 -connected and E is ω_3 -connected.
- (v) Every ω_3 -path connecting an element of I to an element of E crosses $\vartheta(O, Q)$.

We see that this claim is a digital three-dimensional version of the Jordan Curve Theorem. Apart from it being a mathematically pleasing fact, this observation is of the utmost practical importance. In applications we are interested in finding boundaries of objects for two reasons. One is to create computer graphic displays of them, such as the ones shown in Figure 2, and the other is to analyze certain properties of the objects, such as their volumes. From the graphical display point of view, property (v) can be used to prove that when we display the boundary $\vartheta(O, Q)$ as it appears from any exterior point, then the surface displayed will hide all the points interior to the object.

For the purpose of analysis, Claim 1.5 works as follows. Suppose that a device has given us estimates of some physical quantity at points of the cubic grid Z^3 . Suppose further that we can identify from such data that set, A , of grid points which are in cardiac muscle. Assuming that A itself is δ_3 -connected and that the set of grid points in any one of the chambers of the heart is an ω_3 -component of \overline{A} , we can estimate the volume of the left ventricle, say, by selecting ω_3 -adjacent voxels g and h , so that g is in the cardiac muscle and h is in the left ventricle, defining O and Q as in Problem 1.4, and estimating the volume of the left ventricle of the heart as the combined volume of the voxels associated with the grid points of the uniquely defined exterior E , whose existence is guaranteed by Claim 1.5. Alternatively, if we wish to find the volume of the whole heart, then assuming that the set of grid points outside the heart form an ω_3 -component of \overline{A} , we can select ω_3 -adjacent voxels g and h , so that g is in the cardiac muscle and h is outside the heart, defining O and Q as in Problem 1.4, and estimating the volume of the whole heart as the combined volume of the voxels associated with the grid points of the uniquely defined interior I , whose existence is guaranteed by Claim 1.5. This also indicates how the achievement of Problem 1.4 would allow us to detect individual boundaries of the cardiac muscle.

1.7. Algorithms for Fat Flies.

First we observe that just by simply changing the word “vertex” in the Algorithm for Flat Flies into the word “edge” we get a set of instructions that are meaningful in the three-dimensional case. However, the resulting algorithm will not be a solution to Problem 1.4. Even if there is only one grid point with a sugar cube, this simply modified version will fail. The Fat Fly will keep moving forward and, having dirtied only four faces of the cube, it will get back to the original face and will fly away. Clearly, $\vartheta(O, Q)$ in this case should consist of all six faces of the sugar cube, and the proposed algorithm resulted in only four of them being dirtied. So consider instead the following.

Algorithm 1.6. *Algorithm for Fat Flies Model 1*

- (1) *Dirty the face on which you are standing.*
- (2) *Crawl onto the face which meets the one on which you are standing at the edge in front of you.*
- (3) *Turn to the right, and dirty the face on which you are standing.*
- (4) *Crawl onto the face which meets the one on which you are standing at the edge in front of you.*
- (5) *See if it is dirty.*
 - (a) *If it is, fly away.*
 - (b) *If it is not, turn left and start again at Instruction (1).*

It turns out that this algorithm will behave as required for the case of a single sugar cube. It can be seen in Figure 5 that the Fat Fly under the control of the Model 1 algorithm will visit (and hence dirty) every face of a single sugar cube.

However, the route of the Fat Fly as indicated in Figure 5 tells us that the Model 1 algorithm does not behave as is needed for Problem 1.4. Since there are edges not traversed by its route, if we attach a second sugar cube to one of these edges, the grid points of the two sugar cubes are δ_3 -adjacent, and so $\vartheta(O, Q)$ consists of the twelve faces of the pair of sugar cubes. However, the route of the Fat Fly does not change; it still dirties only six faces; see Figure 5.

In fact, we see that this indicates a general principle. If the proposed algorithm is such that the edge toward which the Fat Fly starts crawling is not influenced by the absence or presence of nearby sugar cubes, then for the algorithm to work, it must traverse each edge when started on a face of a single sugar cube. (Otherwise, we can attach a second sugar cube to the untraversed edge, just as indicated in Figure 5.) However, if each edge is traversed once during a route on a single sugar cube, then each face must be visited twice. So the simple mechanism of dirtying the faces the first time they are visited and flying away the second time one is visited will not do. This by itself is not a problem; we can have the fly “mark” a face the first time it visits it, “dirty” it when it finds that it has already been marked, and fly away when it finds that the face has already been dirtied. The interesting question is: what should be the rule for deciding the edge towards which the Fat Fly starts crawling when it is on a particular face?

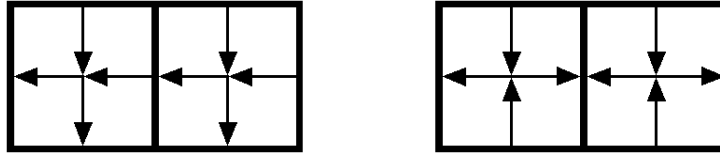


FIGURE 7. Pairs of faces with two types of arrow orientation; only the one on the left is acceptable.

We now establish a further principle. Having accepted that the route of the Fat Fly must traverse each edge, let us aim for the shortest route on a single sugar cube which satisfies this condition. If we draw the route of such a fly on the sugar cube as was done for Figure 5, then we see that on each face there have to be four arrows (one for each edge of the face), two of which point from the center of the face towards an edge and two which point from the other two edges towards the center of the face. Further, there has to be some consistency between these arrows: if an edge has an arrow pointing to it on one face, on the face which meets this edge the arrow has to be pointing away from the edge. Finally, we consider the desirable simplicity of the Algorithm for Fat Flies. We impose the condition that the way in which the arrows point on a face depends only on the orientation of that face. (There are six possible orientations, corresponding to the six faces of a single sugar cube.) Now consider the arrangements shown in Figure 7. We see that the only type of arrangement that satisfies all the conditions just discussed is the one in which the two inward arrows are next to each other (making an angle of 90°) as is illustrated on the left in the figure.

This implies that the route of the Fat Fly for a single sugar cube must have the general form indicated by Figure 8. We say general form, since the considerations above still leave us with some freedom. For example, the horizontal cycle of arrows around the vertical faces could be reversed without violating the conditions described above. The same is true for the two vertical cycles of arrows, leaving us with eight possible legitimate arrangements. However, these are symmetrical versions of each other and, for what we are going to do, there is no harm in restricting our attention to the arrangement shown in Figure 8. A *legitimate route* for the Fat Fly is defined as one which follows the arrows and traverses each edge once.

The following is a fascinating fact. If we restrict (in a reasonable way) the form that an Algorithm for Fat Flies may take, then it can be shown [14] that there is no such algorithm which causes the Fat Fly to cover a legitimate route if placed on a single sugar cube and which, at the same time, satisfies Problem 1.4. In other words, there is no straightforward three-dimensional generalization of the Algorithm for Flat Flies. One way of overcoming this difficulty is to allow Fat Flies with outlandish capabilities, such as being able to create a clone of themselves [14]. We discuss boundary tracking algorithms based on

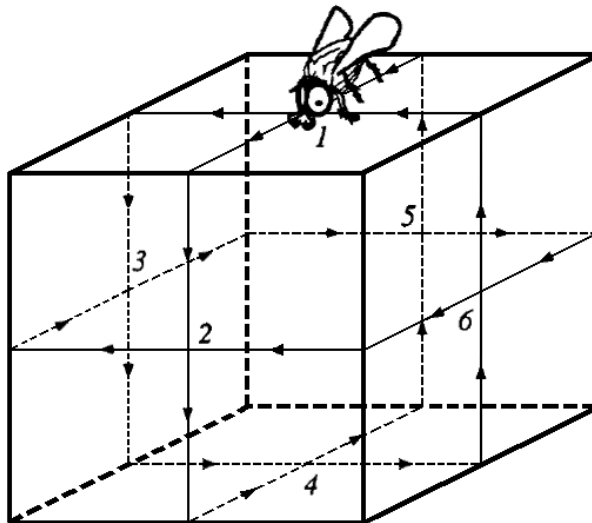


FIGURE 8. Legitimate routes of the Fat Fly on a single sugar cube follow the arrows and cross every edge once. The numbers refer to the orientations of the faces and not to the order in which they are visited.

such ideas in the following sections.

1.8. A formulation of the boundary tracking problem.

We now say good-bye to our flies and prepare ourselves for the more mathematical treatment of the following sections. Let G be an arbitrary grid in \mathcal{R}^3 and let π be the adjacency defined on this grid by $(c, d) \in \pi$ if, and only if, the intersection of the voxels associated with c and d is not a subset of a line in \mathcal{R}^3 . (In case $G = Z^3$, this definition implies that $\pi = \omega_3$.) We redefine the concept of a boundary for this more general situation. For any two subsets O and Q of G the *boundary* between them is

$$(1.8) \quad \vartheta(O, Q) = \{(c, d) \mid c \in O, d \in Q, (c, d) \in \pi\}.$$

A (finite binary) *picture over* a grid G is an assignment of 1 to finitely many points of G and 0 to the rest of the points. (This is a mathematical way of identifying whether or not the voxel associated with the grid point is occupied by a sugar cube.) We refer to the grid points that have 1 assigned to them as *1-spels* and to the others as *0-spels*. (The word *spel* is an abbreviation of “spatial element.”) We say that $(c, d) \in \pi$ is a *bel* (short for “boundary element”) if c is a 1-spel and d is a 0-spel. The following is a generalization of Problem 1.4 in which the adjacencies δ_3 and ω_3 on Z^3 are replaced by arbitrary adjacencies

κ and λ on the grid G . (The general notion of an adjacency between spels will be rigorously defined in Section 3. In Section 2 the discussion will be restricted to specific cases for κ and λ .)

Problem 1.7. *Design an algorithm for the grid G and adjacencies κ and λ , with the following property. For any picture over G and for any bel (g, h) , let O be the κ -component of the set of 1-spels which contains g and Q be the λ -component of the set of 0-spels which contains h . After a finite number of steps the algorithm should terminate and, at that time, its output should be exactly $\vartheta(O, Q)$.*

These notions are easily generalized to N -dimensional space. We define the adjacency π on a given grid G by $(c, d) \in \pi$ if, and only if, the intersection of the Voronoi neighborhoods of c and d is not a subset of any set formed by all linear combinations of $N - 2$ vectors in \mathcal{R}^N . (In case $G = Z^N$, this definition implies that $\pi = \omega_N$.) Defining boundaries, pictures, 1-spels, 0-spels, and bels exactly as above Problem 1.7, the aim itself can be restated without any changes for the N -dimensional case.

In what follows we will not fully achieve this very general aim: the algorithms that we will provide in the next section will perform as desired only for a restricted class of the adjacencies κ and λ . However, this class is sufficiently large to justify us claiming to have achieved the N -dimensional generalization of Problem 1.4 and some instances of Problem 1.7 that are not included in the N -dimensional generalization of Problem 1.4.

2. SOME BOUNDARY TRACKING ALGORITHMS

2.1. Boundary tracking on the cuberille.

In this and the next subsection we concentrate our attention to solving Problem 1.4 or, more mathematically, Problem 1.7 with $G = Z^3$, $\kappa = \delta_3$ and $\lambda = \omega_3$. We first discuss a way of defining the “adjacency of one bel to another” and then use this concept to describe the algorithms.

Looking at Figure 8 we see that the arrangement of arrows on the sugar cube can be interpreted as a digraph with nodes $\{1, 2, 3, 4, 5, 6\}$. We now redefine this digraph in a way that will be easy to generalize later on.

Let, for $1 \leq n \leq N$, u^n denote the *unit vector in direction n* , which is defined as the element of Z^n for which $u^n_n = 1$ and all other components are 0. For the case $N = 3$, consider the digraph with nodes $M = \{u^1, u^2, u^3, -u^1, -u^2, -u^3\}$ and with arcs

$$(2.1) \quad \rho = \left\{ \begin{array}{l} (u^1, u^2), (u^2, -u^1), (-u^1, -u^2), (-u^2, u^1), \\ (u^1, u^3), (u^3, -u^1), (-u^1, -u^3), (-u^3, u^1), \\ (u^2, u^3), (u^3, -u^2), (-u^2, -u^3), (-u^3, u^2) \end{array} \right\}$$

(see Figure 9). This (M, ρ) is an example of the general concept of a basic digraph, which will be defined in Subsection 2.4. Note that if $(c, d) \in \omega_3$, then $d - c \in M$ and is ρ -adjacent to exactly two other unit vectors.

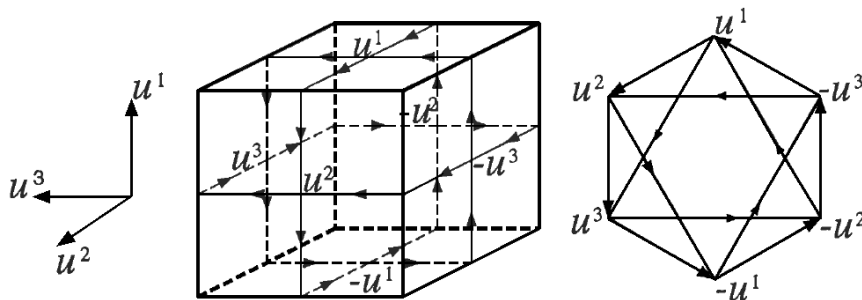


FIGURE 9. Illustration of a basic digraph (on the right) and its interpretation as a set of directions taken while tracking the boundary between a single voxel and the set of all other voxels in Z^3 (in the middle, where each face is labelled by the unit vector normal to the face and exterior to the cube).

The basic digraph gives rise to a bel-adjacency β , defined as follows. If $a = (c, d)$ is a bel (which means that c is a 1-spel, d is a 0-spel, and $(c, d) \in \omega_3$), then there are exactly two unit vectors $u \in M$ which are ρ -adjacent from $d - c$ and each of these gives rise to exactly one bel b_u , which is β -adjacent from (c, d) , according to the following definition (see Figure 10). If $d + u$ is a 1-spel (case (iii) in Figure 10), then $b_u = (d + u, d)$. If $d + u$ is a 0-spel and $c + u$ is a 1-spel (case (ii) in Figure 10), then $b_u = (c + u, d + u)$. If both $d + u$ and $c + u$ are 0-spels (case (i) in Figure 10), then $b_u = (c, c + u)$. Based on these definitions, the following algorithm fulfills the requirements of Problem 1.7 with $G = Z^3$, $\kappa = \delta_3$ and $\lambda = \omega_3$. In the algorithm L is a queue [24] of bels and S is a set of bels (which is considered the output of the algorithm when it terminates). The algorithm can be considered to be an instance of the well-known breadth-first search algorithm for traversing a strongly-connected component of a digraph from an initial node.

Algorithm 2.1. *Inefficient Bel-Tracking Algorithm*

- (1) Put (g, h) into L and S .
- (2) Remove a bel a from L . For both bels b which are β -adjacent from a and which are not in S , put b into L and S .
- (3) Check if L is empty.
 - (a) If it is, STOP.
 - (b) If it is not, start again at Instruction (2).

We will not bother proving the correctness (in the sense of fulfilling the requirements of Problem 1.7 with $G = Z^3$, $\kappa = \delta_3$ and $\lambda = \omega_3$) of this algorithm. This is because we will be able to give a more efficient alternative. The inefficiency of the algorithm above is hidden in the ostensibly innocuous phrase “which are not in S ?” How do we check that a bel is not in S ? In the early

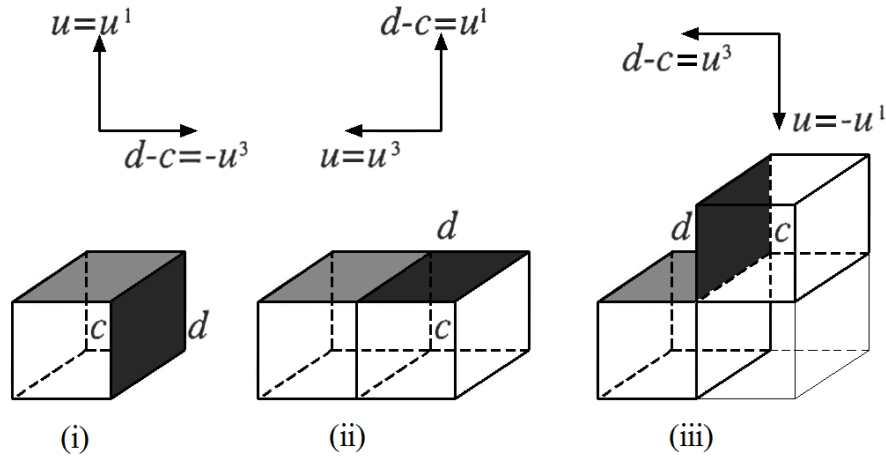


FIGURE 10. Illustration of the definition of bel-adjacency using the basic digraph of Figure 9. In each case, the darkly-shaded bel is β -adjacent to the lightly-shaded one. Note that in addition to the bel that is indicated in this figure, there is a second bel that is β -adjacent to the lightly-shaded one.

stages of the algorithm S has only a few elements and so it is easy to carry out such a check. As the algorithm proceeds, S gets larger and larger (quite typically containing well over a million elements in a medical application) and checking for a particular bel belonging to it gets demanding on computer resources (time and/or storage). We next discuss an alternative to the Inefficient Bel-Tracking Algorithm which goes a long way towards reducing this computational expense.

Prior to doing that we point out a common property of these algorithms. At the end of Subsection 1.7 we discussed the need for “Fat Flies with outlandish capabilities, such as being able to create a clone of themselves.” The mathematical equivalent of this is in the phrase “For both bels b which are β -adjacent from a .” We can represent this as a Fat Fly on bel a cloning itself into two, one clone crawling onto one and the other clone crawling onto the other bel β -adjacent from a .

2.2. Artzy’s Algorithm.

A way of decreasing the computational burden of the Inefficient Bel-Tracking Algorithm was proposed by Artzy *et al.* [2]. Their trick is to make use of an auxiliary data structure T of once-visited bels and to check for membership in T instead of for membership in S . The important underlying fact is that each bel is β -adjacent from exactly two bels, and so it will be reached exactly twice. Hence if a bel is found in T , it has already been reached once before and so it will not be reached again, and can be removed from T . Consequently,

the size of T is likely to be a tiny fraction of the size of S and one can expect orders of magnitude of reduction in computational requirements over the Inefficient Bel-Tracking Algorithm when dealing with typical problems arising in many application areas, such as medicine. For a detailed discussion of the computational considerations associated with the algorithm of this section, see [6].

Algorithm 2.2. *Artzy's Algorithm*

- (1) Put (g, h) into L and S and put two copies of (g, h) into T .
- (2) Remove a bel a from L . For both bels b which are β -adjacent from a , try to find a copy of b in T .
 - (a) If successful, remove this copy of b from T .
 - (b) If not, then put b into L , S and T .
- (3) Check if L is empty.
 - (a) If it is, STOP.
 - (b) If it is not, start again at Instruction (2).

We illustrate the output of Artzy's Algorithm in Figure 11. CT scans with $0.8\text{mm} \times 0.8\text{mm}$ pixels were taken at 7 mm interval covering the spine and the ribs of a patient. From these we estimated (by interpolation) values for a cubic grid assuming a unit distance of 0.65 mm and then we assigned 1s and 0s to the grid points using a threshold appropriate for distinguishing bone (1) from not bone (0). We applied Artzy's Algorithm to the resulting picture. It produced, upon its termination, an output S containing 1,661,728 bels. This is displayed in Figure 11 using a methodology described in [3].

We will not prove the correctness (in the sense of fulfilling the requirements of Problem 1.7 with $G = Z^3$, $\kappa = \delta_3$ and $\lambda = \omega_3$) of Artzy's Algorithm, since in Subsection 2.4 we will state an N -dimensional version of it and the proof of correctness of that version will imply the correctness of the three-dimensional version of this subsection. However, before getting into that generalization we discuss a remarkable adaptation of Artzy's Algorithm to an alternative grid in the three-dimensional Euclidean space.

2.3. A variant of Artzy's Algorithm for the fcc grid.

Near the end of Subsection 1.5 we raised some objections to the use of α_2 as one of the adjacencies in Claim 1.2, but then discarded them on the basis that using only ω_2 would make the claim false. The same consideration lead us to using δ_3 , in addition to ω_3 , in Problem 1.4 (and consequently to allowing such a flimsy-looking connectivity as is illustrated in Figure 5). It turns out, however, that this difficulty is not inherent in the nature of our general problem; rather, it arises due to choosing a cubic (or a square) grid. We now discuss an alternative family of grids in three-dimensional space, one for which objections such as those stated at the end of Subsection 1.5 disappear. (There are also other advantages of these grids over the cubic grids, but we will not get into a

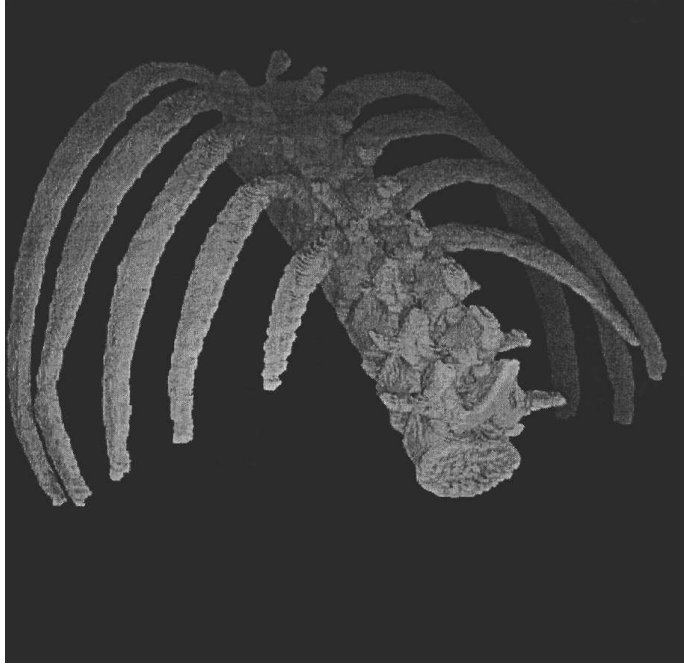


FIGURE 11. Three-dimensional display of the spine and the ribs of a patient (seen from behind and downward from the cervical vertebrae) approximated as a collection of 1,661,728 faces of cubic voxels. The area of the individual faces is the same as in Figure 13.

discussion of these and instead refer the interested reader to Subsection 2.1 of [13].)

For any positive real number ϕ , we define a *face-centered cubic grid* (or *fcc grid*, for short) as the set of points

$$(2.2) \quad F_\phi = \{(\phi c_1, \phi c_2, \phi c_3) \mid c_1, c_2, c_3 \in Z \text{ and } c_1 + c_2 + c_3 \equiv 0 \pmod{2}\}.$$

Each of the associated voxels is a *rhombic dodecahedron* (i.e., a twelve-faced solid, for which every face is an identical rhombus), as can be seen from Figure 12. We will sometimes abbreviate F_1 as F .

Clearly, F is a subset of Z^3 . A remarkable property of F is that for the adjacency π for the grid F , as defined in Subsection 1.8, we have that, for any g and h in F , $(g, h) \in \pi$ if, and only if, $(g, h) \in \delta_3$ (see Figure 12). On the other hand, it follows from the definition of F that, for any g and h in F , it cannot possibly be that $(g, h) \in \omega_3$.

Since F is a subset of Z^3 , every picture over F gives rise to a *derived picture* over Z^3 , simply by assigning 0 to those elements of Z^3 which are not in F . This

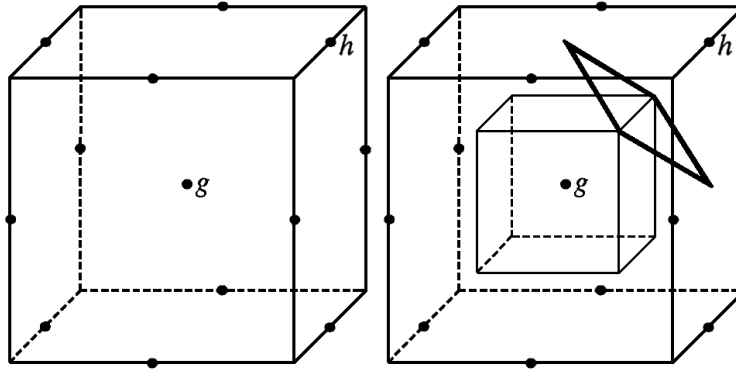


FIGURE 12. On the left we show a point g of the fcc grid $F = F_1$ together with the other 12 grid points which lie in the $2 \times 2 \times 2$ cube whose center is g . On the right we show (using heavy lines) one of the faces of the Voronoi neighborhood of g (whose shape is a rhombic dodecahedron) and also (using lighter lines) the cubic voxel associated with g when considered a point in Z^3 . Note that the edge of the cubic voxel which is specified by (g, h) is a diagonal of that face of the rhombic dodecahedral voxel which is specified by (g, h) .

is the observation which allows us to adapt Artzy's Algorithm to boundary tracking in the fcc grid. The idea [7] is to consider the nature of the bel-adjacency β in the derived picture.

Observe Figure 10. In case (i), $(c, d + u)$ is a bel of the picture over F . Case (ii) cannot possibly arise in a derived picture. In case (iii), $(c, d + u)$ is not a bel of the picture over F , since both c and $d + u$ are 1-spels. Note that case (i) can be distinguished from the other cases by the fact that it is the only one in which the bel (e, f) which is β -adjacent from (c, d) has the property that $e = c$ (and we also happen to have that $f = c + u$). We make use of this observation by running Artzy's Algorithm on the derived picture and producing an output S which contains bels of the picture over F .

In order to do this, we need to find a suitable starting point to Artzy's Algorithm, since in this case the (g, h) of Problem 1.7 is not a bel of the derived picture. However, if we define, for $1 \leq n \leq 3$,

$$(2.3) \quad h'_n = \begin{cases} h_n, & \text{if } \sum_{i=1}^n |g_i - h_i| \leq 1, \\ g_n, & \text{otherwise,} \end{cases}$$

then (g, h') will be a bel of the derived picture in Z^3 .

Algorithm 2.3. *Tracking Algorithm for the FCC Grid*

- (1) Put (g, h') into L and put two copies of (g, h') into T .
- (2) Remove a bel (c, d) of the derived picture from L . For both bels (e, f) of the derived picture that are β -adjacent from (c, d) , do the following. If $e = c$, then put $(c, d + f - c)$ into S . Try to find one copy of (e, f) in T .
 - (a) If successful, remove this copy of (e, f) from T .
 - (b) If not, then put (e, f) into L and T .
- (3) Check if L is empty.
 - (a) If it is, STOP.
 - (b) If it is not, start again at Instruction (2).

We illustrate the output of the Tracking Algorithm for the FCC Grid in Figure 13. The same CT scans were used as those employed to produce Figure 11. From these we estimated (by interpolation) values for an fcc grid assuming a unit distance of $2^{0.25} \times 0.65\text{mm}$ and then we assigned 1s and 0s to the grid points using the same threshold that was employed to produce Figure 11 for distinguishing bone (1) from not bone (0). We changed the unit of distance so that the faces of the voxels in the two grids will have the same size; we considered this to be reasonable when comparing the boundaries output by two tracking algorithms. Consequently, the volumes of the rhombic dodecahedra are more than 3.36 times larger than the volumes of the cubes and the number of fcc grid points needed to cover a particular volume of space is less than a third of the number of cubic grid points needed for the same purpose. (This is one of the advantages of using an fcc grid over using a cubic grid.) We applied the Tracking Algorithm for the FCC Grid to the resulting picture. It produced, upon its termination, an output S containing 1,258,482 bels. This is displayed in Figure 13 using the same methodology as was employed to produce Figure 11.

The appearance of the two computer graphic displays is very similar, but there are some less obvious, but nevertheless significant, differences. First (even though the same set of CT slices, the same threshold, and the same area for the faces were used to produce the boundaries) the number of bels in the approximation to the spine and the ribs based on the cubic grid is approximately a third more than the number of bels in the approximation based on the fcc grid. This is because the larger number of orientations (twelve, as opposed to six) of the rhombic dodecahedral faces allows us to fit the underlying biological surface more tightly. This also has a consequence on computational costs: the time required to track the boundary displayed in Figure 13 is less than the time required to track (by essentially the same algorithm) the boundary displayed in Figure 11. The time required to produce the computer graphic displays of the already tracked boundary was approximately one and one-half times longer when using the cubic grid than when using the fcc grid.



FIGURE 13. Three-dimensional display of the spine and ribs of a patient (seen from behind and downward from the cervical vertebrae) approximated as a collection of 1,258,482 faces of rhombic dodecahedral voxels. The area of the individual faces is the same as in Figure 11.

2.4. Artzy's Algorithm in N -dimensional space.

We now generalize Artzy's Algorithm to the N -dimensional Euclidean space, with $N > 1$. The intent is to fulfill the generalization of Problem 1.7 which is stated in the last paragraph of Subsection 1.8 with $G = Z^N$, $\kappa = \delta_N$ and $\lambda = \omega_N$. To do this we need to generalize the concept of "adjacency of one bel to another," which is achieved by introducing an appropriate basic digraph using a generalization of (2.1).

Consider the *basic digraph* (M, ρ) with the following properties:

- (i) $M = \{u^n \mid 1 \leq n \leq N\} \cup \{-u^n \mid 1 \leq n \leq N\}$;
- (ii) $\rho = \bigcup_{1 \leq i < j \leq N} \rho_{i,j}$, where

$$(2.4) \quad \rho_{i,j} = \{(u^i, u^j), (u^j, -u^i), (-u^i, -u^j), (-u^j, u^i)\}.$$

(Note that the three rows of (2.1) correspond to $\rho_{1,2}$, $\rho_{1,3}$ and $\rho_{2,3}$, respectively.) It is an obvious consequence of this definition that every element of M is ρ -adjacent from exactly $N - 1$ elements of M .

We now use this basic digraph to define, for a picture over Z^N , a bel-adjacency β . If $b = (c, d)$ is a bel, then we have that c is a 1-spel, d is a 0-spel, and $(c, d) \in \omega_N$. From the definition of ω_N it follows that $d - c$ is a node of the basic digraph (M, ρ) . Every node ρ -adjacent from $d - c$ gives rise to a single bel β -adjacent from b . (Consequently, every bel is β -adjacent from exactly $N - 1$ bels.) Because of the utility of this in the proofs of the next section, we define β as $\bigcup_{1 \leq i < j \leq N} \beta_{i,j}$, where every node $\rho_{i,j}$ -adjacent from $d - c$ gives rise to a single bel $\beta_{i,j}$ -adjacent from b . The last sentence of the previous paragraph can now be restated by saying that, for any bel b , $in_\beta(b) = N - 1$.

In following the details of the specification of how a node u which is $\rho_{i,j}$ -adjacent from $d - c$ gives rise to a unique bel which is $\beta_{i,j}$ -adjacent from (c, d) , consult again Figure 10. (In that figure we are illustrating the situation when u has been selected using $\rho_{1,3}$. Three separate cases are considered: in (i) $d - c = -u^3$ and, hence, $u = u^1$, in (ii) $d - c = u^1$ and, hence, $u = u^3$, and in (iii) $d - c = u^3$ and, hence, $u = -u^1$.) We distinguish between three mutually exclusive possibilities. The first is that $d + u$ is a 1-spel (this corresponds to (iii) in Figure 10). In this case the bel $\beta_{i,j}$ -adjacent from (c, d) is specified to be $(d + u, d)$. The second is that $d + u$ is a 0-spel and $c + u$ is a 1-spel (this corresponds to (ii) in Figure 10). In this case the bel $\beta_{i,j}$ -adjacent from (c, d) is specified to be $(c + u, d + u)$. The third is that both $d + u$ and $c + u$ are 0-spels (this corresponds to (i) in Figure 10). In this case the bel $\beta_{i,j}$ -adjacent from (c, d) is specified to be $(c, c + u)$.

Algorithm 2.4. *Artzy's Algorithm in N -Dimensional Space ($N \geq 2$)*

- (1) Put (g, h) into L and S and put $in_\beta((g, h)) = N - 1$ copies of (g, h) into T .
- (2) Remove a bel a from L . For all bels b which are β -adjacent from a , try to find a copy of b in T .
 - (a) If successful, remove this copy of b from T .
 - (b) If not, then put b into L and S and put $in_\beta(b) - 1$ copies of b into T .
- (3) Check if L is empty.
 - (a) If it is, STOP.
 - (b) If it is not, start again at Instruction (2).

Claim 2.5. *For $N \geq 2$, for any picture over the grid Z^N and for any bel (g, h) , let O be the δ_N -component of the set of 1-spels which contains g and Q be the ω_N -component of the set of 0-spels which contains h . After a finite number of steps Artzy's Algorithm in N -Dimensional Space terminates and, at that time, $S = \vartheta(O, Q)$.*

This claim will be proved in Section 4. It is clear that what we called Artzy's Algorithm earlier is exactly the same as Artzy's Algorithm in 3-Dimensional Space. In particular, Claim 2.5 implies that Artzy's Algorithm provides a solution to Problem 1.4. A little reflection will also reveal that the Algorithm

for Flat Flies of the last section is essentially similar to Artzy's Algorithm in 2-Dimensional Space and, consequently, Claim 1.2 is a corollary of Claim 2.5 and will not have to be proved separately.

2.5. The Gordon-Udupa Algorithm.

Motivated by the desire of speeding up the operation of Artzy's Algorithm, Gordon and Udupa [8] introduced an alternative basic digraph (M, ρ) , in which $M = \{u^1, u^2, u^3, -u^1, -u^2, -u^3\}$ and

$$(2.5) \quad \rho = \left\{ (u^1, u^2), (u^2, -u^1), (-u^1, -u^2), (-u^2, u^1), \right. \\ \left. (u^1, u^3), (u^3, -u^1), (-u^1, -u^3), (-u^3, u^1) \right\}$$

(see Figure 14). For this digraph, if $(c, d) \in \omega_3$, then $d - c \in M$ and is ρ -adjacent to either one or two other unit vectors.

This basic digraph gives rise to a bel-adjacency β . If $a = (c, d)$ is a bel, then there may be either one or two unit vectors $u \in M$ which are ρ -adjacent from $d - c$ and each of these gives rise to exactly one bel b_u which is β -adjacent from (c, d) . This is specified in exactly the same manner as was done in Subsection 2.1 using Figure 10. Note that for the resulting β , $in_\beta(b)$ will be either 1 or 2, depending on the "direction" of the bel b (see Figure 14).

Based on this definition, the following is the algorithm of Gordon and Udupa [8]. In the algorithm L is a queue [24] of bels and S is a set of bels (which is considered the output of the algorithm when it terminates).

Algorithm 2.6. Gordon-Udupa Algorithm

- (1) Put (g, h) into L and S and put $in_\beta((g, h))$ copies of (g, h) into T .
- (2) Remove a bel a from L . For the bels b which are β -adjacent from a , try to find a copy of b in T .
 - (a) If successful, remove this copy of b from T .
 - (b) If not, then put b into L, S and put $in_\beta(b) - 1$ copies of b into T .
- (3) Check if L is empty.
 - (a) If it is, STOP.
 - (b) If it is not, start again at Instruction (2).

The advantage of the Gordon-Udupa Algorithm over Artzy's algorithm is that on the whole the expectation is that for two thirds of the bels there will be only one bel adjacent from them and so T is likely to remain smaller than the T of Artzy's Algorithm for the same picture and the same starting bel (g, h) . It is reported in [8] that the Gordon-Udupa Algorithm is indeed less demanding on computer resources (storage and time) than Artzy's Algorithm. Indeed, this is not surprising; every bel is visited twice during the execution of Artzy's Algorithm, while there will always be some bels that are visited only once during the execution of the Gordon-Udupa Algorithm.

It is easy to check that the Gordon-Udupa Algorithm detects all the bels when started on a face of a single sugar cube (a picture with only one 1-spel) or when started on a face of a pair of sugar cubes arranged as in Figure 5. However, if the only common edge of a pair of sugar cubes were a vertical one

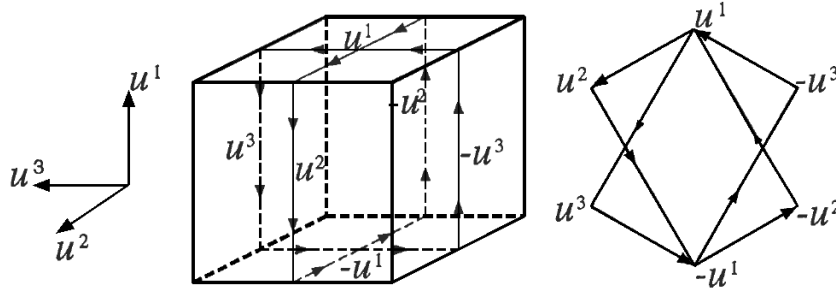


FIGURE 14. Illustration of a basic digraph of Gordon and Udupa (on the right) and its interpretation as a set of directions taken while tracking the boundary between a single voxel and the set of all other voxels in Z^3 (in the middle).

(i.e., parallel to u^1), then the Gordon-Udupa Algorithm would detect the six faces of the sugar cube on which it is started, but Artzy's Algorithm would detect all twelve faces of the two sugar cubes. So the outputs of the two algorithms are sometimes not the same. To make this mathematically precise we need define on Z^N the adjacency γ_N by

$$(2.6) \quad (c, d) \in \gamma_N \Leftrightarrow \left[(c, d) \in \alpha_N \text{ and } \sum_{n=2}^N |c_n - d_n| \leq 1 \right].$$

This is an adjacency between face-adjacency and face-or-edge-adjacency. Roughly speaking we can say that while Artzy's Algorithm (Algorithm 2.2) tracks the boundaries of face-or-edge-adjacent components, the Gordon-Udupa Algorithm detects boundaries of γ_N -components.

2.6. A plan of action.

In this section we have proposed various ways of defining boundaries and made some claims that our definitions result in boundaries with some desirable properties (Claims 1.3 and 1.5). We now generalize these claims. Here $\vartheta(O, Q)$ is defined by (1.8) with π as defined in the paragraph following Problem 1.7.

Claim 2.7. (*Desirability Claim.*) *For any picture over the grid G in \mathcal{R}^N , let O be a κ -component of the set of 1-spels and Q be a λ -component of the set of 0-spels, such that $\vartheta(O, Q)$ is not empty. Then there exist two uniquely defined subsets I and E of G with the following properties.*

- (i) $O \subseteq I$ and $Q \subseteq E$.
- (ii) $\vartheta(O, Q) = \vartheta(I, E)$.
- (iii) $I \cup E = G$ and $I \cap E = \emptyset$.
- (iv) I is κ -connected and E is λ -connected.
- (v) Every π -path connecting an element of I to an element of E crosses $\vartheta(O, Q)$.

We will prove this Desirability Claim for the following three cases.

Case (i). $G = Z^N$, $\kappa = \gamma_N$ and $\lambda = \omega_N$;

Case (ii). $G = Z^N$, $\kappa = \delta_N$ and $\lambda = \omega_N$;

Case (iii). $G = F$ and $\kappa = \lambda = \delta_3$.

Note that Case (ii) is a generalization of Claims 1.3 and 1.5, as can be seen by setting in those claims A to be the finite set of 1-spels and $N = 2$ and $N = 3$, respectively.

We have also proposed some algorithms for boundary tracking. The usefulness of such an algorithm is greatly enhanced if one can prove about it something like the following.

Claim 2.8. (*Correctness Claim.*) *For any picture over the grid G in \mathcal{R}^N and for any $bel(g, h)$, let O be the κ -component of the set of 1-spels which contains g and Q be the λ -component of the set of 0-spels which contains h . After a finite number of steps the algorithm terminates and, at that time, its output (S) is $\vartheta(O, Q)$.*

Note that the Correctness Claim is simply saying that the algorithm solves Problem 1.7. We will provide a proof of the Correctness Claim for Artzy's Algorithm in N -dimensional space (with $G = Z^N$, $\kappa = \delta_N$ and $\lambda = \omega_N$, corresponding to Case (ii) above), but as stated before that includes also Artzy's Algorithm and the Algorithm for Flat Flies. The first mathematical proof of correctness of Artzy's Algorithm (Algorithm 2.2) appeared in [15] and of the Gordon-Udupa Algorithm (Algorithm 2.6) appeared in [18].

3. DESIRABILITY OF BOUNDARIES

3.1. Surfaces in digital spaces.

We now generalize the ideas expressed in Subsection 1.8. A *digital space* [13] is a pair (V, π) , where V is an arbitrary nonempty set and π is a symmetric binary relation on V such that V is π -connected. Sometimes we will refer to π as the *proto-adjacency* of the digital space (V, π) . Already discussed examples of digital spaces are (Z^N, ω_N) and, for the fcc grid, (F, δ_3) . Elements of V will be called *spels* (short for spatial elements) and elements of π will be called *surfels* (short for surface elements).

A thorough presentation of this very general approach is given in [13]. Here we compromise: we abandon the full generality of [13] and opt instead for a level of generality which is sufficient to cover the three special cases of the last section. In particular, the sets of objects determined by particular definitions given below will be subsets of the sets of corresponding objects in [13]; to clarify these relationships, when we give a definition below we usually also indicate the qualifying adjectives that would be applied in [13] to define the same concept. We already had an informal example of this when we introduced the notion of a "(finite binary) picture."

Any nonempty subset S of π will be called an (antisymmetric) *surface* in (V, π) if whenever $(c, d) \in S$, then $(d, c) \notin S$. We will be particularly concerned with a special class of surfaces, which we refer to as boundaries. These are defined as follows. Let (V, π) be a digital space and let O and Q be subsets of V . Then the *boundary in (V, π) between O and Q* is $\vartheta(O, Q)$, defined exactly as in (1.8). In what follows, we will normally be dealing with nonempty boundaries between disjoint sets; clearly, such boundaries are surfaces. We say that a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ *crosses* the surface S if there is a k , $1 \leq k \leq K$, such that either $(c^{(k-1)}, c^{(k)}) \in S$ or $(c^{(k)}, c^{(k-1)}) \in S$.

Before going any further into our theoretical development, we restate the Desirability Claim of the last section in the more general terminology of this section, so that we can better see where we are heading and why. For this we need some further definitions.

A (finite binary) *picture* over the digital space (V, π) is a triple (V, π, f) , where f is a function whose domain is V and whose value is 0 except that, for a finite number of spels c in V , the value of $f(c)$ is 1. We refer to these finitely-many spels as *1-spels* and to all the other spels as *0-spels*.

A (very tight) *spel-adjacency* in the digital space (V, π) is a binary relation ρ on V , such that

- (i) ρ is symmetric,
- (ii) $\pi \subseteq \rho$, and
- (iii) for any $(c, d) \in \rho$, there exists a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ from c to d such that $(c^{(0)}, c^{(k)}) \in \rho$, for $1 \leq k \leq K$.

Note that π is always a spel-adjacency in (V, π) and that, in addition, γ_N and δ_N are spel-adjacencies in (Z^N, ω_N) , for $N > 1$.

Another spel-adjacency in (Z^3, ω_3) is σ that is defined as follows. In addition to all elements of ω_3 , σ contains a pair of spels (c, d) if one of the following three conditions is satisfied:

- (i) $c_1 = d_1$ and $(c_2 - d_2) \times (c_3 - d_3) = -1$,
- (ii) $c_2 = d_2$ and $(c_3 - d_3) \times (c_1 - d_1) = -1$,
- (iii) $c_3 = d_3$ and $(c_1 - d_1) \times (c_2 - d_2) = -1$.

The following is a generalization of Claim 2.7 to arbitrary digital spaces.

Claim 3.1 (Desirability Claim). *Let κ and λ be spel-adjacencies in a digital space (V, π) . For any picture over (V, π) , let O be a κ -component of the set of 1-spels and Q be a λ -component of the set of 0-spels, such that $\vartheta(O, Q)$ is not empty (and, hence, it is a surface). Then there exist two uniquely defined subsets I and E of V with the following properties.*

- (i) $O \subseteq I$ and $Q \subseteq E$.
- (ii) $\vartheta(O, Q) = \vartheta(I, E)$.
- (iii) $I \cup E = V$ and $I \cap E = \emptyset$.
- (iv) I is κ -connected and E is λ -connected.
- (v) Every π -path connecting an element of I to an element of E crosses $\vartheta(O, Q)$.

We call an ordered pair (κ, λ) of spel-adjacencies in a digital space (V, π) *desirable* for (V, π) , if the Desirability Claim as stated above is valid for that particular choice of κ and λ . We see that, in order to prove the Desirability Claim of Subsection 2.6 for the three special cases listed there, what we need to do is to show that:

- Case (i). (γ_N, ω_N) is desirable for (Z^N, ω_N) ;
- Case (ii). (δ_N, ω_N) is desirable for (Z^N, ω_N) ;
- Case (iii). (δ_3, δ_3) is desirable for (F, δ_3) .

This section is aimed at achieving this, but in a context which provides a general approach to proving the desirability of pairs of spel-adjacencies.

3.2. Interiors and exteriors.

Let (V, π) be a digital space and let S be a surface in it. We define the *immediate interior* $II(S)$, the *immediate exterior* $IE(S)$, and the *immediate neighborhood* $IN(S)$ of S as follows:

$$(3.1) \quad \begin{aligned} II(S) &= \{c \mid (c, d) \in S \text{ for some } d \text{ in } V\}, \\ IE(S) &= \{d \mid (c, d) \in S \text{ for some } c \text{ in } V\}, \\ IN(S) &= II(S) \cup IE(S). \end{aligned}$$

The *interior* $I(S)$ and the *exterior* $E(S)$ of S are defined as follows:

$$(3.2) \quad \begin{aligned} I(S) &= \{c \in V \mid \text{there exists a } \pi\text{-path connecting } c \text{ to} \\ &\quad \text{an element of } II(S) \text{ that does not cross } S\}, \\ E(S) &= \{c \in V \mid \text{there exists a } \pi\text{-path connecting } c \text{ to} \\ &\quad \text{an element of } IE(S) \text{ that does not cross } S\}. \end{aligned}$$

Lemma 3.2. *For any surface S in a digital space (V, π) ,*

$$(3.3) \quad I(S) \cup E(S) = V.$$

Proof. Let c be an arbitrary element of V and d be an arbitrary element of $IN(S)$. (By definition, neither of the sets V and $IN(S)$ is empty.) Since V is π -connected, there exists a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ from c to d . Let k be the smallest nonnegative integer such that $c^{(k)}$ is in $IN(S)$. Then $\langle c^{(0)}, \dots, c^{(k)} \rangle$ is a π -path connecting c to an element of $II(S) \cup IE(S)$ which does not cross S . Hence c is in at least one of $I(S)$ and $E(S)$. \square

A surface S in a digital space (V, π) is said to be *near-Jordan* if every π -path from any element of $II(S)$ to any element of $IE(S)$ crosses S . We remark that if S is not a near-Jordan surface, then the intersection of its interior and its exterior is necessarily nonempty. (This is because there is a path, not crossing S , from its immediate interior to its immediate exterior; all the spels in this path are in both the interior and the exterior of S , as can be seen from (3.2), the symmetry of proto-adjacency and the symmetrical definition of ‘‘crosses.’’)

Lemma 3.3. *Let S be a surface in a digital space (V, π) . Then the following three conditions are equivalent.*

- (i) S is near-Jordan.
- (ii) Every π -path from any element of $I(S)$ to any element of $E(S)$ crosses S .
- (iii) $I(S) \cap E(S) = \emptyset$.

Furthermore, if these conditions are satisfied, then it is also the case that

$$(3.4) \quad S = \vartheta(I(S), E(S)).$$

Proof. Let S be any surface in a digital space (V, π) . Suppose there is a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ not crossing S connecting a c in $I(S)$ to a d in $E(S)$. By (3.2), there is also a π -path $\langle e^{(0)}, \dots, e^{(L)} \rangle$ not crossing S connecting c to an element of $II(S)$ and a π -path $\langle d^{(0)}, \dots, d^{(M)} \rangle$ not crossing S connecting d to an element of $IE(S)$. Then $\langle e^{(L)}, \dots, e^{(0)} = c = c^{(0)}, \dots, c^{(K)} = d = d^{(0)}, \dots, d^{(M)} \rangle$ is a π -path not crossing S which connects an element of $II(S)$ to an element of $IE(S)$. Therefore, by definition, S is not near-Jordan. This argument shows that (i) implies (ii). If $I(S)$ and $E(S)$ have an element c in common, then the π -path $\langle c \rangle$ is from an element of $I(S)$ to an element of $E(S)$ and does not cross S . This shows that (ii) implies (iii). That (iii) implies (i) was proven in the paragraph preceding the statement of the lemma. Finally, it is the case for any surface S that it is a subset of $\vartheta(I(S), E(S))$. (This follows trivially from the definitions of immediate interior and exterior and of interior and exterior.) Suppose now that (i) - (iii) hold and that the surfel (c, d) is in $\vartheta(I(S), E(S))$. Then d belongs to $E(S)$, and so by (iii) d does not belong to $I(S)$; hence (d, c) is not in S . The π -path $\langle c, d \rangle$ is from an element of $I(S)$ to an element of $E(S)$ and so, as condition (ii) is satisfied, it crosses S . Hence (c, d) is in S . This proves that any of the conditions (i) - (iii) implies (3.4). \square

The following theorem provides two further characterizations of a near-Jordan surface. The first of these has a significant property that characterizations (ii) and (iii) in Lemma 3.3 do not have: it gives an easy way to construct near-Jordan surfaces. It should also help the reader to grasp the essential nature of such surfaces.

Theorem 3.4. *Let S be a surface in a digital space (V, π) . Then the following three conditions are equivalent.*

- (i) S is near-Jordan.
 - (ii) There exists a nonempty proper subset O of V such that
- $$(3.5) \quad S = \vartheta(O, \overline{O}).$$
- (iii) There exists a nonempty proper subset O of V such that $I(S) = O$ and $E(S) = \overline{O}$.

Furthermore, if these conditions are satisfied, then the O of (ii) has to be $I(S)$.

Proof. Let S be any surface in a digital space (V, π) . Since, by definition, S is nonempty, both $I(S)$ and $E(S)$ are nonempty. If S is also near-Jordan, then it follows from Lemma 3.2 and Lemma 3.3(iii) that $I(S)$ is a nonempty proper subset of V and $E(S)$ is the complement of $I(S)$ in V . By setting $O = I(S)$, (i) implies (ii) from (3.4). Now assume that there is an O that satisfies (ii). To show that for this O we have that $I(S) = O$ and $E(S) = \overline{O}$, it suffices (by Lemma 3.2) to show that (3.5) implies

$$(3.6) \quad E(S) \cap O = \emptyset$$

and

$$(3.7) \quad I(S) \cap \overline{O} = \emptyset.$$

First we note that (3.5) implies that $II(S)$ is a subset of O and that $IE(S)$ is a subset of \overline{O} and also that any π -path from an element of O to an element (necessarily not the same) of \overline{O} must cross S . Now suppose that c is in O . Then any π -path from c to an element of $IE(S)$ must cross S , showing that c is not in $E(S)$, which means that (3.6) is true. Now suppose that d is in \overline{O} . Then any π -path from d to an element of $II(S)$ must cross S , showing that d is not in $I(S)$, which means that (3.7) is true. Thus, (ii) implies (iii) and the O of (ii) has to be $I(S)$. Finally, (iii) of this theorem implies (iii) of Lemma 3.3, which is shown there to be equivalent to (i). \square

3.3. Connectedness in digital spaces.

The following is a useful tool for proving connectedness of the interiors and/or the exteriors of surfaces.

Theorem 3.5. *Let S be a surface, and let κ and λ be spel-adjacencies in a digital space (V, π) .*

(i) *If for some κ -connected subset O of V*

$$(3.8) \quad II(S) \subseteq O \subseteq I(S),$$

then $I(S)$ is κ -connected.

(ii) *If for some λ -connected subset Q of V*

$$(3.9) \quad IE(S) \subseteq Q \subseteq E(S),$$

then $E(S)$ is λ -connected.

Proof. We prove only (i) since the proof of (ii) is entirely analogous. We assume the truth of the premise of (i). Let c_1 and c_2 be two spels in $I(S)$. We need to prove that they are κ -connected in $I(S)$. By (3.2), for $1 \leq i \leq 2$, there exists a d_i in $II(S)$ such that there is a π -path (hence, by the definition of a spel-adjacency, a κ -path) not crossing S (and hence entirely in $I(S)$) connecting c_i to d_i . By (3.8), d_1 and d_2 are in O and are therefore κ -connected in O and hence (again by (3.8)) in $I(S)$. By combining these three κ -paths in $I(S)$ into a single κ -path (recall that κ is symmetric by the definition of a spel-adjacency), we get the desired result. \square

3.4. Boundaries in pictures.

Let S be a surface and κ and λ be spel-adjacencies in a digital space (V, π) . We say that S is a $\kappa\lambda$ -boundary in the picture (V, π, f) , if there is κ -component O of the set of 1-spels and a λ -component Q of the set of 0-spels such that $S = \vartheta(O, Q)$.

Theorem 3.6. *Let (V, π, f) be a picture and κ and λ be spel-adjacencies in (V, π) . If O is a κ -component of the set of 1-spels, Q is a λ -component of the set of 0-spels, and S is the $\kappa\lambda$ -boundary $\vartheta(O, Q)$, then $O \subseteq I(S)$ and $Q \subseteq E(S)$.*

Proof. We prove only that $O \subseteq I(S)$, since the proof that $Q \subseteq E(S)$ is strictly analogous. Let $d \in O$, $c \in II(S) (\subseteq O)$, and $\langle c^{(0)}, \dots, c^{(K)} \rangle$ be a κ -path in O from c to d . We prove by induction that, for $0 \leq k \leq K$, $c^{(k)} \in I(S)$. This suffices to complete our proof.

Clearly, $c^{(0)} \in I(S)$. Now assume that the same is true for $c^{(k-1)}$, for some k , $1 \leq k \leq K$. Since $(c^{(k-1)}, c^{(k)}) \in \kappa$, there exists a π -path $\langle e^{(0)}, \dots, e^{(T)} \rangle$ from $c^{(k-1)}$ to $c^{(k)}$, such that, for $1 \leq t \leq T$, $(e^{(0)}, e^{(t)}) \in \kappa$. Now we prove by induction that, for $0 \leq t \leq T$,

$$(3.10) \quad e^{(t)} \in I(S) \cup Q.$$

Since $e^{(T)} = c^{(k)}$ is in O and so cannot be in Q , this shows that $c^{(k)} \in I(S)$ and so completes our proof.

If $t = 0$, then (3.10) is satisfied since $e^{(0)} = c^{(k-1)}$ is in $I(S)$ by the hypothesis for the induction on k . Now assume that (3.10) is satisfied for some $t < T$. In case $e^{(t)} \in I(S)$, either $e^{(t+1)} \in I(S)$ (in which case we are done) or $(e^{(t)}, e^{(t+1)}) \in S$ (see (3.2)), and so $e^{(t+1)} \in Q$ (and again we are done). In case $e^{(t)} \in Q$, either $e^{(t+1)}$ is a 0-spel and therefore is in Q since $\pi \subseteq \lambda$ (in which case we are done) or $e^{(t+1)} \in O$, since $c^{(k-1)}$ is in O . In this latter case, $(e^{(t+1)}, e^{(t)}) \in S$, and so $e^{(t+1)} \in I(S)$ (and again we are done). \square

Theorem 3.7. *Let κ and λ be spel-adjacencies in a digital space (V, π) . If S is a $\kappa\lambda$ -boundary in a picture (V, π, f) , then $I(S)$ is κ -connected and $E(S)$ is λ -connected.*

Proof. Let $S = \vartheta(O, Q)$, for some κ -component O of the set of 1-spels and some λ -component Q of the set of 0-spels. Then, by Theorem 3.6, $O \subseteq I(S)$ and $Q \subseteq E(S)$ and so, by Theorem 3.5, $I(S)$ is κ -connected and $E(S)$ is λ -connected. \square

What we have proved so far allows us to establish the following powerful result for proving the desirability of a pair of spel-adjacencies.

Theorem 3.8. *If κ and λ are spel-adjacencies in a digital space (V, π) such that every $\kappa\lambda$ -boundary in every picture is near-Jordan, then (κ, λ) is desirable for (V, π) .*

Proof. Let (V, π, f) be a picture over (V, π) and let O be a κ -component of the set of 1-spels and Q be a λ -component of the set of 0-spels, such that $S = \vartheta(O, Q)$ is not empty (and hence it is a $\kappa\lambda$ -boundary). By the assumption of the theorem, S is near-Jordan. Let $I = I(S)$ and $E = E(S)$.

In Claim 3.1, (i) is satisfied, by Theorem 3.6. Near-Jordanness of S implies the validity of (3.4), which is the same as (ii) in Claim 3.1. On the other hand, (iii) in Claim 3.1 follows from Lemma 3.2 and (iii) of Lemma 3.3. Theorem 3.7 gives us (iv) of Claim 3.1 and (v) follows from (ii) of Lemma 3.3. That $I(S)$ is the only choice for I (and hence that $E(S)$ is the only choice for E), satisfying (ii) of Claim 3.1 follows from Theorem 3.4. \square

3.5. Isomorphic digital spaces.

An *isomorphism* i from a digital space (V, π) to a digital space (V', π') is a one-to-one function that maps V onto V' so that

$$(3.11) \quad (c, d) \in \pi \Leftrightarrow (i(c), i(d)) \in \pi'.$$

Theorem 3.9. *The digital spaces (Z^3, σ) and (F, δ_3) are isomorphic.*

Proof. Define, for $c \in Z^3$, $i(c) = (c_2 + c_3, c_3 + c_1, c_1 + c_2)$. Since the sum of its components is even, $i(c) \in F$. To show that i is one-to-one, assume that $i(c') = i(c'')$, and let $c = c' - c''$. Then $i(c) = i(c') - i(c'') = 0$, which implies that $c_2 = -c_3$, $c_1 = -c_3$, and $-2c_3 = c_1 + c_2 = 0$. From this it follows that all three components of c are 0, and hence $c' = c''$. Finally, to show that i maps Z^3 onto F , let $d \in F$, and define $c_1 = (-d_1 + d_2 + d_3)/2$, $c_2 = (d_1 - d_2 + d_3)/2$, and $c_3 = (d_1 + d_2 - d_3)/2$. Since the sum of the components of d is even, c_1 , c_2 , and c_3 are all integers, and, clearly, $i(c) = d$. This shows that i is a one-to-one function mapping Z^3 onto F .

To show that i is an isomorphism from (Z^3, σ) to (F, δ_3) , first we observe that $(i(c), i(d)) \in \delta_3$ if, and only if, $i(c)$ and $i(d)$ differ by 1 in two coordinates and are the same in the third. A consequence of this observation is that if we let $a = i(c) - i(d)$, then $(i(c), i(d)) \in \delta_3$ if, and only if, two of $|a_1|$, $|a_2|$, and $|a_3|$ have the value 1 and the third has the value 0. By the definition of i , for $a = i(c) - i(d)$,

$$(3.12) \quad \begin{aligned} a_1 &= (c_2 - d_2) + (c_3 - d_3), \\ a_2 &= (c_3 - d_3) + (c_1 - d_1), \\ a_3 &= (c_1 - d_1) + (c_2 - d_2). \end{aligned}$$

Now suppose that $(c, d) \in \sigma$. There are two different reasons why this might be the case. The first is that $(c, d) \in \omega_3$. In this case, exactly one of $|c_1 - d_1|$, $|c_2 - d_2|$, and $|c_3 - d_3|$ has the value 1 and the other two have the value 0. The other possibility is that there exist two integers k and j ($1 \leq k \neq j \leq 3$) such that $(c_k - d_k) \times (c_j - d_j) = -1$ and $c_n = d_n$ if n is neither k nor j . Looking at the components of a , as expressed in the equation above, we see that either case implies that $(i(c), i(d)) \in \delta_3$.

Conversely, suppose that $(i(c), i(d)) \in \delta_3$ and so two of $|a_1|$, $|a_2|$, and $|a_3|$ have the value 1 and the third has the value 0. One way that this may

come about is by both the terms in the zero-valued sum being themselves zero-valued. In that case, the other (common) term in the other two sums must have absolute value 1. This implies that $(c, d) \in \omega_3$. The alternative possibility is that the terms in the zero-valued sum are not zero-valued but are the negatives of each other. Then we have the situation that there exist two integers k and j ($1 \leq k \neq j \leq 3$) such that $(c_k - d_k) = -(c_j - d_j) \neq 0$ and, if n is neither k nor j , $|(c_n - d_n) + (c_k - d_k)| = 1$ and $|(c_n - d_n) - (c_k - d_k)| = 1$. It is easy to see that this can happen only if $(c_k - d_k) \times (c_j - d_j) = -1$ and $c_n = d_n$. In either case, $(c, d) \in \sigma$. \square

Theorem 3.10. *Let i be an isomorphism from a digital space (V, π) to a digital space (V', π') , and let S be a surface in (V, π) . If we define*

$$(3.13) \quad S' = \{(i(c), i(d)) \mid (c, d) \in S\},$$

then S is a near-Jordan surface in (V, π) if, and only if, S' is a near-Jordan surface in (V', π') .

Proof. First we point out that S' is indeed a surface in (V', π') , since the fact that S is a nonempty subset of π implies, in view of (3.11), that S' is a nonempty subset of π' .

The rest of the theorem is proved by the following sequence of equivalences, each one of which is an immediate consequence of the definitions. S is not a near-Jordan surface in (V, π) if, and only if, there exists a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ from an element of $II(S)$ to an element of $IE(S)$ which does not cross S . This happens if, and only if, there exists a π' -path $\langle i(c^{(0)}), \dots, i(c^{(K)}) \rangle$ from an element of $II(S')$ to an element of $IE(S')$ that does not cross S' , which means that S' is not a near-Jordan surface in (V', π') . \square

If there is an isomorphism i from the digital space (V, π) to a digital space (V', π') and ρ is an adjacency in (V, π) , then we define its *isomorphic image* (due to i) ρ' in (V', π') by

$$(3.14) \quad (i(c), i(d)) \in \rho' \Leftrightarrow (c, d) \in \rho.$$

It is trivial to check that the ρ' defined in this fashion is an adjacency in (V', π') .

Since here we will be discussing multiple definitions simultaneously, for any spel-adjacency ρ in a digital space (V, π) and for any subsets O and Q of V , we introduce the notation

$$(3.15) \quad \vartheta_\rho(O, Q) = \{(c, d) \mid c \in O, d \in Q, (c, d) \in \rho\}.$$

Note that $\vartheta_\pi(O, Q)$ is just the boundary in (V, π) between O and Q .

Theorem 3.11. *If i is an isomorphism from the digital space (V, π) to the digital space (V', π') , (κ, λ) is a desirable pair of adjacencies for (V, π) and κ' and λ' are the respective isomorphic images of κ and λ , then (κ', λ') is desirable for (V', π') .*

Proof. By Theorem 3.8 we need to prove that any $\kappa'\lambda'$ -boundary S' in any picture (V', π', f') over (V', π') is near-Jordan.

Let $S' = \vartheta_{\pi'}(O', Q')$, where O' is a κ' -component of the set of 1-spels and Q' is a λ' -component of the set of 0-spels of (V', π', f') . Define the function f on V by $f(c) = f'(i(c))$. Then (V, π, f) is a picture over (V, π) , and $O = \{c \mid i(c) \in O'\}$ and $Q = \{d \mid i(d) \in Q'\}$ are clearly a κ -component of the set of 1-spels and a λ -component of the set of 0-spels, respectively, of (V, π, f) . Defining $S = \vartheta_{\pi}(O, Q)$ we get

$$(3.16) \quad \begin{aligned} (c, d) \in S &\Leftrightarrow c \in O \text{ and } d \in Q \text{ and } (c, d) \in \pi \\ &\Leftrightarrow i(c) \in O' \text{ and } i(d) \in Q' \text{ and } (i(c), i(d)) \in \pi' \\ &\Leftrightarrow (i(c), i(d)) \in S'. \end{aligned}$$

This implies that S is a surface (since it is nonempty), it is near-Jordan (by Lemma 3.3) and so S' is also near-Jordan (by Theorem 3.10). \square

3.6. Simple connectedness in digital spaces.

Let S be a surface in a digital space (V, π) . For any practical application, it would be impossible to determine whether S is near-Jordan by examining all π -paths from $II(S)$ to $IE(S)$. We need instead a result which says that S is near-Jordan if some local condition is satisfied at every surfel of S . This is the subject matter of this and the next subsection.

In classical topology, a simply connected space is (intuitively speaking) a connected space in which every loop can be continuously pulled to a point without leaving the space. Here we present one of the corresponding notions for digital spaces (for others, see Section 6.1 of [13] and Section 3 of [5]).

If

$$(3.17) \quad P = \langle c^{(1)}, \dots, c^{(m)}, d^{(0)}, \dots, d^{(n)}, e^{(1)}, \dots, e^{(l)} \rangle$$

and

$$(3.18) \quad P' = \langle c^{(1)}, \dots, c^{(m)}, f^{(0)}, \dots, f^{(k)}, e^{(1)}, \dots, e^{(l)} \rangle$$

are π -paths in V , such that

$$(3.19) \quad f^{(0)} = d^{(0)}, \quad f^{(k)} = d^{(n)}, \quad \text{and } 1 \leq k + n \leq 4,$$

then P and P' are said to be *elementarily equivalent* in the digital space (V, π) . (Note that in this definition, m or l or both in (3.17) may be zero; in other words, the difference between elementarily equivalent π -paths may be at their “head” or at their “tail.”) Two π -paths, P and P' in V are said to be *equivalent* in the digital space (V, π) , if there is a sequence of π -paths P_0, \dots, P_L ($L \geq 0$) in V , such that $P_0 = P$, $P_L = P'$ and, for $1 \leq l \leq L$, P_{l-1} and P_l are elementarily equivalent in the digital space (V, π) .

We demonstrate the notion of equivalent π -paths in (Z^2, ω_2) in Figure 15. That $\langle a, b, c, d, e, f, g, h, i, j, k, l, a \rangle$ is elementarily equivalent to $\langle a, b, c, d, e, d, g, h, i, j, k, l, a \rangle$ follows by substituting in (3.17) $n = 2$, $d^{(0)} = e$, $d^{(1)} = f$, $d^{(2)} = g$ and in (3.18) $k = 2$, $f^{(0)} = e$, $f^{(1)} = d$, $f^{(2)} = g$. That

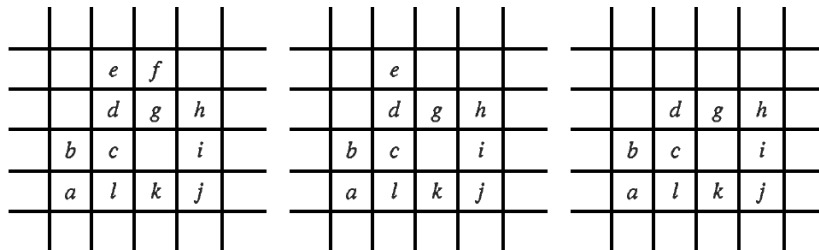


FIGURE 15. Demonstration of equivalence of π -paths in (Z^2, ω_2) : $\langle a, b, c, d, e, f, g, h, i, j, k, l, a \rangle$ is equivalent to $\langle a, b, c, d, g, h, i, j, k, l, a \rangle$, since it is elementarily equivalent to $\langle a, b, c, d, e, d, g, h, i, j, k, l, a \rangle$, which is elementarily equivalent to $\langle a, b, c, d, g, h, i, j, k, l, a \rangle$.

$\langle a, b, c, d, e, d, g, h, i, j, k, l, a \rangle$ is elementarily equivalent to $\langle a, b, c, d, g, h, i, j, k, l, a \rangle$ follows by substituting in (3.17) $n = 2$, $d^{(0)} = d$, $d^{(1)} = e$, $d^{(2)} = d$ and in (3.18) $k = 0$, $f^{(0)} = d$.

A loop (of length K) in a digital space (V, π) is a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ such that $c^{(K)} = c^{(0)}$. In particular, for any spel c , $\langle c \rangle$ is a loop, and is called a *trivial loop*. We note that any loop of length 1, 2, or 3 is automatically equivalent to a trivial loop. A digital space is said to be *simply connected* if every loop in the digital space is equivalent to a trivial loop.

Theorem 3.12. *For any positive integer N , (Z^N, ω_N) is simply connected.*

Proof. We show that every loop in (Z^N, ω_N) is equivalent to a trivial loop. We do this by induction on the length of the loop. We have already noted that, in general, every loop of length 1 or 2 is equivalent to a trivial loop. Suppose that every loop in (Z^N, ω_N) of length less than some $K > 2$ is equivalent to a trivial loop. Consider a loop $\langle c^{(0)}, \dots, c^{(K)} \rangle$ in (Z^N, ω_N) of length K . We now show that it is equivalent to a loop of length $K - 1$ or $K - 2$ and thus, by the induction hypothesis, is equivalent to a trivial loop.

Since $c^{(1)}$ is ω_N -adjacent to $c^{(0)}$, we have $c^{(1)} \neq c^{(0)}$. (To illustrate our argument, consider Figure 15. In that figure $c^{(0)} = a = (0, 0)$ and $c^{(1)} = b = (0, 1)$.) Then there is a unique j such that $c_j^{(1)} \neq c_j^{(0)}$. Without loss of generality, assume that $c_j^{(1)} > c_j^{(0)}$. (In Figure 15, $j = 2$.) Let $z = \max_{1 \leq k \leq K} \{c_j^{(k)}\}$. (In Figure 15, $z = 3$.) Let l be the largest integer in the range $0 < l < K$ such that $c_j^{(l)} = z$. (In Figure 15, in the left column $l = 5$ with $c^{(5)} = f = (2, 3)$ and in the middle column $l = 4$ with $c^{(4)} = e = (1, 3)$.) Clearly, $c_j^{(l+1)} = z - 1$. (In Figure 15, in the left column $c^{(l+1)} = c^{(6)} = g = (2, 2)$ and in the middle column $c^{(l+1)} = c^{(5)} = d = (1, 2)$.) Let k be the smallest integer such that, for all i in the range $0 < k \leq i \leq l < K$, we have $c_j^{(i)} = z$. (In Figure 15, $k = 4$ in

both the left and the middle column.) Clearly, $c_j^{(k-1)} = z - 1$. (In Figure 15, $c^{(k-1)} = c^{(3)} = d = (1, 2)$ in both the left and middle columns.) We will now use induction on $l - k$.

If $l - k = 0$, then $k = l$ and therefore $c^{(k-1)} = c^{(k+1)}$. (This case is illustrated in Figure 15 by the middle column, for which $k = l = 4$ and $c^{(k-1)} = c^{(k+1)} = d$.) In this case the loop

$$(3.20) \quad \langle c^{(0)}, \dots, c^{(k-2)}, c^{(k-1)}, c^{(k)}, c^{(k+1)}, c^{(k+2)}, \dots, c^{(K)} \rangle$$

is elementarily equivalent to the loop

$$(3.21) \quad \langle c^{(0)}, \dots, c^{(k-2)}, c^{(k-1)} = c^{(k+1)}, c^{(k+2)}, \dots, c^{(K)} \rangle$$

which has length $K - 2$ and we are done. (The loop in (3.20) is illustrated by the loop of the middle column of Figure 15, while the loop in (3.21) is illustrated by the loop of the right column of Figure 15.)

Suppose now (induction hypothesis) that whenever $l - k = h$, then $\langle c^{(0)}, \dots, c^{(K)} \rangle$ is equivalent to a loop of length $K - 1$ or $K - 2$. We now show that the same conclusion holds if $l - k = h + 1$. Let

$$(3.22) \quad \langle c^{(0)}, c^{(1)}, \dots, c^{(k)}, \dots, c^{(l-1)}, c^{(l)}, c^{(l+1)}, \dots, c^{(K)} \rangle$$

be a loop. Define j , z , l , and k for this loop as above and suppose $l - k = h + 1$. (This is the case in Figure 15 for the loop associated with the left column with $l = 5$, $k = 4$ and, hence, $h = 0$. Note also that for this loop $j = 2$ and $z = 3$.) Let $c'^{(l)}$ be a spel such that, for $1 \leq n \leq N$,

$$(3.23) \quad c'_n{}^{(l)} = \begin{cases} z - 1, & \text{if } n = j, \\ c_n^{(l-1)}, & \text{otherwise.} \end{cases}$$

(In the left column of Figure 15, $c'^{(l)} = c'^{(5)} = (1, 2) = d$.) Clearly, $c'^{(l)}$ is proto-adjacent to $c^{(l-1)}$. Also, $c^{(l+1)}$ differs from $c^{(l)}$ in exactly the j th component (which is $z - 1$ for the former) and $c^{(l)}$ differs from $c^{(l-1)}$ in exactly one component which is other than the j th; thus $c'^{(l)}$ is proto-adjacent to $c^{(l+1)}$. (In the left column of Figure 15, $c^{(l-1)} = c^{(4)} = e$ and $c^{(l+1)} = c^{(6)} = g$, both of which are proto-adjacent to $c'^{(l)} = c'^{(5)} = d$.) It follows that

$$(3.24) \quad \langle c^{(0)}, c^{(1)}, \dots, c^{(k)}, \dots, c^{(l-1)}, c'^{(l)}, c^{(l+1)}, \dots, c^{(K)} \rangle$$

is also a loop and is easily seen to be elementarily equivalent to the loop in (3.22). (The loop in (3.22) is illustrated by the loop of the left column of Figure 15, while the loop in (3.24) is illustrated by the loop of the middle column of Figure 15.) Furthermore, it follows from (3.23) that for the loop in (3.24) the condition of the induction hypothesis holds. (Indeed, we have already seen that for the middle column of Figure 15 we have $l - k = 0 = h$.) So, by the induction hypothesis, the loop in (3.24) is equivalent to a loop of length $K - 1$ or $K - 2$. From this it follows that the loop in (3.22) is also equivalent to a loop of length $K - 1$ or $K - 2$. \square

3.7. Locally-Jordan surfaces.

Let S be a surface and $P = \langle c^{(0)}, \dots, c^{(K)} \rangle$ be a π -path in a digital space (V, π) . We say that the *crossing parity* $p_S P$ of P through S is *even* (or zero; i.e., $p_S P = 0$) if the number of π -paths among $\langle c^{(0)}, c^{(1)} \rangle, \dots, \langle c^{(K-1)}, c^{(K)} \rangle$ that cross S is even and we say that it is *odd* (or one; i.e., $p_S P = 1$) if this number is odd. We use the notation \oplus for *modulo 2 addition* of parities (i.e.; $0 \oplus 0 = 1 \oplus 1 = 0$ and $0 \oplus 1 = 1 \oplus 0 = 1$).

It is easy to see that, for any surface S in a digital space, the crossing parity through S is even for any loop in the digital space whose length is not greater than two. Also, cyclic permutation of a loop does not influence its crossing parity through a surface S , since, for $1 \leq k \leq K$,

$$(3.25) \quad \begin{aligned} p_S \langle c^{(0)}, \dots, c^{(k-1)}, c^{(k)}, \dots, c^{(K)} \rangle = \\ p_S \langle c^{(k)}, \dots, c^{(K)} = c^{(0)}, \dots, c^{(k-1)}, c^{(k)} \rangle. \end{aligned}$$

In addition, reversing a π -path does not change its crossing parity. It is also easy to see that if $\langle c^{(0)}, \dots, c^{(K)} \rangle$ and $\langle d^{(0)}, \dots, d^{(L)} \rangle$ are π -paths such that $c^{(K)} = d^{(0)}$, then

$$(3.26) \quad \begin{aligned} p_S \langle c^{(0)}, \dots, c^{(K)}, d^{(1)}, \dots, d^{(L)} \rangle = \\ p_S \langle c^{(0)}, \dots, c^{(K)} \rangle \oplus p_S \langle d^{(0)}, \dots, d^{(L)} \rangle. \end{aligned}$$

Theorem 3.13. *If S is a near-Jordan surface in a digital space, then the crossing parity through S is odd for any π -path $P = \langle c^{(0)}, \dots, c^{(K)} \rangle$ such that $(c^{(0)}, c^{(K)}) \in S$.*

Proof. First note that (according to Lemma 3.3), since S is near-Jordan, $I(S) \cap E(S) = \emptyset$ and $S = \vartheta(I(S), E(S))$. We prove by induction that, for any $0 \leq k \leq K$,

$$(3.27) \quad p_S \langle c^{(0)}, \dots, c^{(k)} \rangle = \begin{cases} 0, & \text{if } c^{(k)} \in I(S), \\ 1, & \text{otherwise.} \end{cases}$$

Since $c^{(K)} \in E(S)$, this is sufficient to prove the theorem.

Clearly, (3.27) is true for $k = 0$. Suppose that (3.27) is true for some $k - 1$, where $1 \leq k \leq K$. We prove that it is also true for k . We use the following special case of (3.26):

$$(3.28) \quad p_S \langle c^{(0)}, \dots, c^{(k-1)}, c^{(k)} \rangle = p_S \langle c^{(0)}, \dots, c^{(k-1)} \rangle \oplus p_S \langle c^{(k-1)}, c^{(k)} \rangle.$$

In case $c^{(k-1)} \in I(S)$, the first term on the right-hand side of (3.28) is 0, and the second term is 0 if $c^{(k)} \in I(S)$ and 1 otherwise. In case $c^{(k-1)} \notin I(S)$, the first term on the right-hand side of (3.28) is 1, and the second term is 1 if $c^{(k)} \in I(S)$ and 0 otherwise. In either case, (3.27) is true for k . \square

A surface S in a digital space (V, π) is said to be *locally-Jordan* if $p_S \langle c^{(0)}, \dots, c^{(K)} \rangle$ is odd for any π -path such that $(c^{(0)}, c^{(K)}) \in S$ and $2 \leq K \leq 3$. (This restriction on V enforces ‘‘locality.’’ It says that any path from the immediate interior of S to its immediate exterior must cross S exactly once if $K = 2$ and

must cross S either once or three times if $K = 3$.) By Theorem 3.13, if a surface S in a digital space (V, π) is near-Jordan, then it is locally-Jordan.

Lemma 3.14. *Any loop of length not more than 4 has even crossing parity through any locally-Jordan surface in any digital space (V, π) .*

Proof. We have already pointed out that the crossing parity through any surface is even for any loop of length not more than two.

Let S be a locally-Jordan surface. Consider a loop $L = \langle c^{(0)}, \dots, c^{(K)} \rangle$ with $3 \leq K \leq 4$. If there does not exist a k , $1 \leq k \leq K$, such that $\langle c^{(k-1)}, c^{(k)} \rangle$ crosses S , then we are done. Otherwise, for such a k , either $(c^{(k-1)}, c^{(k)}) \in S$ or $(c^{(k)}, c^{(k-1)}) \in S$ is true. If $(c^{(k)}, c^{(k-1)}) \in S$, $P = \langle c^{(k)}, \dots, c^{(K)} = c^{(0)}, \dots, c^{(k-1)} \rangle$ is a π -path of length $K-1$ with $2 \leq K-1 \leq 3$ and so the fact that S is locally-Jordan implies $p_S P = 1$. By application of (3.25) and (3.26), we have

$$(3.29) \quad p_S L = p_S P \oplus p_S \langle c^{(k-1)}, c^{(k)} \rangle = 1 \oplus 1 = 0.$$

If $(c^{(k-1)}, c^{(k)}) \in S$, a similar argument, which also makes use of the fact that reversing a π -path does not change its crossing parity, can be used to derive the same conclusion. \square

Lemma 3.15. *Let S be a locally-Jordan surface in a digital space (V, π) . If P and P' are equivalent π -paths, then they have the same crossing parity through S .*

Proof. By the definition of equivalent, it is sufficient to prove that if P and P' satisfy (3.17), (3.18) and (3.19), then they have the same crossing parity through S .

By applying (3.26), we get

$$(3.30) \quad \begin{aligned} p_S P &= p_S \langle c^{(1)}, \dots, c^{(m)}, d^{(0)} \rangle \oplus \\ &p_S \langle d^{(0)}, \dots, d^{(n)} \rangle \oplus p_S \langle d^{(n)}, e^{(1)}, \dots, e^{(l)} \rangle \end{aligned}$$

and

$$(3.31) \quad \begin{aligned} p_S P' &= p_S \langle c^{(1)}, \dots, c^{(m)}, f^{(0)} \rangle \oplus \\ &p_S \langle f^{(0)}, \dots, f^{(k)} \rangle \oplus p_S \langle f^{(k)}, e^{(1)}, \dots, e^{(l)} \rangle. \end{aligned}$$

Therefore, using (3.19), the invariance of crossing parity under reversal, and (3.26), we get

$$(3.32) \quad \begin{aligned} p_S P \oplus p_S P' &= p_S \langle d^{(0)}, \dots, d^{(n)} \rangle \oplus p_S \langle f^{(0)}, \dots, f^{(k)} \rangle \\ &= p_S \langle d^{(0)}, \dots, d^{(n)} = f^{(k)}, \dots, f^{(0)} = d^{(0)} \rangle = 0. \end{aligned}$$

The last equality follows from the previous lemma combined with (3.19). \square

The results up to now apply to digital spaces which do not have to be simply connected. The next lemma makes essential use of simple connectedness.

Lemma 3.16. *If S is a locally-Jordan surface in a simply connected digital space (V, π) , then S is near-Jordan if either (and hence both) of the following two equivalent conditions holds.*

- (i) *For any $c \in II(S)$ and $d \in II(S)$, there exists a π -path P from c to d such that $p_S P = 0$.*
- (ii) *For any $c \in IE(S)$ and $d \in IE(S)$, there exists a π -path P from c to d such that $p_S P = 0$.*

Proof. Evidently the two conditions are equivalent. Indeed, if $c \in II(S)$ and $d \in II(S)$, then there exist $c' \in IE(S)$ and $d' \in IE(S)$ such that $(c, c') \in S$ and $(d, d') \in S$; hence if there exists a π -path of even crossing parity from c' to d' , then there also exists one from c to d , so that (ii) implies (i). Similarly, (i) implies (ii).

In what follows, we prove that S is near-Jordan if (ii) holds. We do this by supposing that S is not near-Jordan and showing that this, together with (ii), leads to a contradiction.

First, we show that there exists a π -path $P_1 = \langle c^{(1)}, \dots, c^{(K)} \rangle$ such that $c^{(1)} \in II(S)$, $c^{(K)} \in IE(S)$ and $p_S P_1 = 0$. Indeed, since S is supposed to be not near-Jordan, there is a π -path from $II(S)$ to $IE(S)$ that does not cross S . Clearly, any such π -path has the required properties.

Next, we show that there exists a π -path $P_3 = \langle e^{(1)}, \dots, e^{(L)} \rangle$ such that $(e^{(1)}, e^{(L)}) \in S$ and $p_S P_3 = 0$. Let P_1 be the π -path of the last paragraph. Let $c^{(0)}$ be such that $(c^{(1)}, c^{(0)}) \in S$. By (ii), there exists a π -path $P_2 = \langle c^{(K)} = d^{(0)}, \dots, d^{(L)} = c^{(0)} \rangle$ from $c^{(K)}$ to $c^{(0)}$ such that $p_S P_2 = 0$. Then $P_3 = \langle c^{(1)}, \dots, c^{(K)}, d^{(1)}, \dots, d^{(L)} \rangle$ is a π -path from $c^{(1)}$ to $d^{(L)}$ such that $(c^{(1)}, d^{(L)}) \in S$ and, by (3.26), $p_S P_3 = p_S P_1 \oplus p_S P_2 = 0$.

For a π -path P_3 satisfying the properties listed at the beginning of the previous paragraph, let $e^{(0)} = e^{(L)}$. By (3.26), $P_4 = \langle e^{(0)}, e^{(1)}, \dots, e^{(L)} \rangle$ is a loop such that $p_S P_4 = p_S \langle e^{(0)}, e^{(1)} \rangle \oplus p_S P_3 = 1$. Since (V, π) is simply connected, P_4 is equivalent to a trivial loop, whose crossing parity through S is zero. Since S is locally-Jordan, according to Lemma 3.15, we must also have $p_S P_4 = 0$, contradicting the fact that $p_S P_4 = 1$. \square

Lemma 3.17. *Let (V, π, f) be a picture over the digital space (V, π) .*

- (i) *Let λ be a spel-adjacency in (V, π) . Let O be a union of π -components of 1-spels and Q be a λ -component of 0-spels in (V, π, f) such that $S = \vartheta(O, Q)$ is nonempty. For any c and d in Q , there exists a π -path P from c to d such that $p_S P = 0$.*
- (ii) *Let κ be a spel-adjacency in (V, π) . Let O be a κ -component of 1-spels and Q be a union of π -components of 0-spels in (V, π, f) such that $S = \vartheta(O, Q)$ is nonempty. For any c and d in O , there exists a π -path P from c to d such that $p_S P = 0$.*

Proof. We prove only (i), since the proof of (ii) is obviously similar.

First, we show that the result is true if $(c, d) \in \lambda$. In this case, there exists a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ from c to d such that, $0 \leq k \leq K$, $(c, c^k) \in \lambda$. Now we prove by induction that, for $0 \leq k \leq K$,

$$(3.33) \quad p_S \langle c^{(0)}, \dots, c^{(k)} \rangle = \begin{cases} 0, & \text{if } c^{(k)} \notin O, \\ 1, & \text{if } c^{(k)} \in O. \end{cases}$$

Since $c^{(K)}$ is a 0-spel, and so cannot be in O , this inductive proof meets the aim of this paragraph. Clearly, (3.33) is true for $k = 0$, since $c^{(0)}$ is a 0-spel. Now suppose that it is true for some $k - 1$ ($1 \leq k \leq K$). In the proof we repeatedly use (3.28). First we consider the case $c^{(k-1)} \in O$. In this case the first term on the right-hand side of (3.28) has value 1, by the induction hypothesis. If $c^{(k)}$ is a 1-spel, then it must also be in O and so the second term on the right-hand side of (3.28) has value 0. If $c^{(k)}$ is a 0-spel, then it must be in Q (since $(c, c^k) \in \lambda$ and Q is a λ -component of 0-spels that contains c) and so the second term on the right-hand side of (3.28) has the value 1. Either way, (3.33) is true for k . On the other hand, if $c^{(k-1)} \notin O$, the first term on the right-hand side of (3.28) has value 0 by the induction hypothesis. If $c^{(k-1)}$ is a 1-spel, then it cannot be in Q and so the second term on the right-hand side of (3.28) has value 0. At the same time $c^{(k)}$ cannot be in O , so (3.33) is true for k . If $c^{(k-1)}$ is a 0-spel, then (as before) it must be in Q and so the second term on the right-hand side of (3.28) has value 0 if $c^{(k)} \notin O$ and value 1 if $c^{(k)} \in O$. Thus, again (3.33) is true in either case for k . This completes the induction proof.

Now consider arbitrary c and d in Q . There exists a λ -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ of 0-spels in Q from c to d . By the result in the last paragraph, for $1 \leq k \leq K$, there exists a π -path $P_k = \langle c^{(k-1)} = c_k^{(0)}, \dots, c_k^{(m_k)} = c^{(k)} \rangle$ such that $p_S P_k = 0$. Therefore, $P = \langle c_1^{(0)}, \dots, c_1^{(m_1)}, c_2^{(1)}, \dots, c_2^{(m_2)}, \dots, c_K^{(1)}, \dots, c_K^{(m_K)} \rangle$ is a π -path from c to d such that

$$(3.34) \quad p_S P = \sum_{k=1}^K p_S \langle c_k^{(0)}, \dots, c_k^{(m_k)} \rangle = \sum_{k=1}^K p_S P_k = 0,$$

as can be shown by repeated application of (3.26). (Here \sum refers to modulo 2 additions.) \square

Theorem 3.18. *If κ and λ are spel-adjacencies in a simply connected digital space (V, π) , then every locally-Jordan $\kappa\lambda$ -boundary in a picture over (V, π) is near-Jordan.*

Proof. Let O be a κ -component of 1-spels and Q be a λ -component of 0-spels in a picture over (V, π) such that $S = \vartheta(O, Q)$ is a locally-Jordan $\kappa\lambda$ -boundary. Noting that Q has to be a union of π -components of 0-spels, we see from Lemma 3.17(ii) that for any c and d in O , there exists a π -path P from c to d such that $p_S P = 0$. Noting that $II(S) \subseteq O$, we see that this implies that condition (i) in Lemma 3.16 holds and so S is near-Jordan. \square

3.8. Desirable pairs of spel-adjacencies.

Lemma 3.19. *If S is a $\kappa\lambda$ -boundary in a picture over a digital space (V, π) and $P = \langle a, b, c \rangle$ is a π -path such that $(a, c) \in S$, then $p_S P = 1$.*

Proof. Suppose that $S = \vartheta(O, Q)$, where O is a κ -component of 1-spels and Q is a λ -component of 0-spels. If b is a 1-spel, then it is in O and $(b, c) \in S$. If b is a 0-spel, then it is in Q and $(a, b) \in S$. In either case, $p_S P = 1$. \square

Lemma 3.20. *A $\kappa\lambda$ -boundary S in a picture over a digital space (V, π) is locally-Jordan if $p_S P = 1$ for every π -path $P = \langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)} \rangle$ of length 3 such that $(c^{(0)}, c^{(3)}) \in S$, $c^{(0)} \neq c^{(2)}$, $c^{(1)} \neq c^{(3)}$, $c^{(1)}$ is a 0-spel and $c^{(2)}$ is a 1-spel.*

Proof. Let O be the κ -component of 1-spels and Q be the λ -component of 0-spels such that $S = \vartheta(O, Q)$. We need to show that $p_S P = 1$ for any π -path $P = \langle c^{(0)}, \dots, c^{(K)} \rangle$ such that $(c^{(0)}, c^{(K)}) \in S$ and $K = 2$ or 3 .

By Lemma 3.19, we need only show the result for $K = 3$. If $c^{(0)} = c^{(2)}$ or $c^{(1)} = c^{(3)}$, then it is easy to see that $p_S P = p_S \langle c^{(0)}, c^{(3)} \rangle = 1$. In the following, we assume that both $c^{(0)} \neq c^{(2)}$ and $c^{(1)} \neq c^{(3)}$. There are four possibilities: (i) both $c^{(1)}$ and $c^{(2)}$ are 1-spels, (ii) both $c^{(1)}$ and $c^{(2)}$ are 0-spels, (iii) $c^{(1)}$ is a 1-spel and $c^{(2)}$ is a 0-spel, and (iv) $c^{(1)}$ is a 0-spel and $c^{(2)}$ is a 1-spel. In cases (i), (ii) and (iii), it is easily seen that $p_S P = 1$. That this is also true in case (iv) is exactly the condition in our lemma. \square

Theorem 3.21. *For $N \geq 1$, every $\gamma_N \omega_N$ -boundary in every picture over the digital space (Z^N, ω_N) is near-Jordan.*

Proof. Let S be an arbitrary $\gamma_N \omega_N$ -boundary in a picture over (Z^N, ω_N) . Let O be the γ_N -component of 1-spels and Q be the ω_N -component of 0-spels such that $S = \vartheta(O, Q)$. Since γ_N and ω_N are both spel-adjacencies in the simply connected digital space (Z^N, ω_N) (see Theorem 3.12), all we have to do, according to Theorem 3.18, is to show that S is locally-Jordan. We use Lemma 3.20. Let $P = \langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)} \rangle$ be an ω_N -path of length 3 such that $(c^{(0)}, c^{(3)}) \in S$, $c^{(0)} \neq c^{(2)}$, $c^{(1)} \neq c^{(3)}$, $c^{(1)}$ is a 0-spel and $c^{(2)}$ is a 1-spel. Now all we need to show is that $p_S P = 1$.

The properties that $\langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)} \rangle$ is an ω_N -path, $(c^{(0)}, c^{(3)}) \in S$ (and hence $(c^{(3)}, c^{(0)}) \in \pi$), $c^{(0)} \neq c^{(2)}$ and $c^{(1)} \neq c^{(3)}$ imply that there exist integers i, j ($1 \leq i \neq j \leq N$) and u, v ($|u| = |v| = 1$) such that

$$(3.35) \quad c_i^{(1)} = c_i^{(0)} + u, \quad c_j^{(2)} = c_j^{(1)} + v, \quad c_i^{(3)} = c_i^{(2)} - u, \quad c_j^{(0)} = c_j^{(3)} - v.$$

If i or j is 1, then $c^{(2)}$ is γ_N -adjacent to $c^{(0)}$ and, consequently is in O . Under these circumstances, it is easy to see that, whether or not $c^{(1)}$ is in Q , we have $p_S P = 1$. On the other hand, if i and j are not 1, then we also have that either $c^{(2)} \in O$ or $c^{(1)} \in Q$ (and, consequently, $p_S P = 1$), as we prove now by showing that the alternative leads to a contradiction.

Assuming the alternative, we have a loop $\langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)}, c^{(0)} \rangle$ in (Z^N, ω_N) , such that $c_1^{(k)}$ is the same for $0 \leq k \leq 3$, and all of the following conditions hold: (i) $(c^{(0)}, c^{(3)}) \in S$, (ii) there exist integers i, j ($1 \leq i \neq j \leq N$) and u, v ($|u| = |v| = 1$) such that (3.35) is satisfied, and (iii) $c^{(1)}$ is a 0-spel not in Q and $c^{(2)}$ is a 1-spel not in O . Since S is assumed to be finite, we may assume without loss of generality that $\langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)}, c^{(0)} \rangle$ is such a loop for which the common value z of $c_1^{(0)} = c_1^{(1)} = c_1^{(2)} = c_1^{(3)}$ is as great as possible. Now consider the four spels $c'^{(0)}, \dots, c'^{(3)}$ such that $c'^{(k)}$ is proto-adjacent to $c^{(k)}$ and $c_1'^{(k)} = c_1^{(k)} + 1$ for $0 \leq k \leq 3$. Clearly, $\langle c'^{(0)}, c'^{(1)}, c'^{(2)}, c'^{(3)}, c'^{(0)} \rangle$ is a loop in (Z^N, ω_N) such that $c_1'^{(k)}$ is the same (namely, $z + 1$) for $0 \leq k \leq 3$. Condition (ii) also holds for this new loop. Looking at condition (iii), we see that $c'^{(1)}$ must be a 0-spel (otherwise $c^{(2)}$ would be in O) and not in Q (otherwise $c^{(1)}$ would be in Q). Similarly, $c'^{(3)}$ must be a 0-spel and hence must be in Q . In view of this, $c'^{(2)}$ must be a 1-spel (otherwise $c'^{(1)}$ would be in Q after all) and not in O (otherwise $c^{(2)}$ would be in O). Similarly, $c'^{(0)}$ must be a 1-spel and hence must be in O . Putting all this together, we see that the loop $\langle c'^{(0)}, c'^{(1)}, c'^{(2)}, c'^{(3)}, c'^{(0)} \rangle$ also satisfies conditions (i) and (iii). This contradicts the maximality of z and thus completes the proof. \square

Combining Theorem 3.21 with Theorem 3.8 yields the following.

Corollary 3.22. *For $N \geq 1$, (γ_N, ω_N) is desirable for (Z^N, ω_N) .*

Theorem 3.23. *If (κ, λ) is a desirable pair of adjacencies for a digital space (V, π) , then so is (κ', λ) for any adjacency such that $\kappa \subseteq \kappa'$.*

Proof. By Theorem 3.8 all we need to prove that any $\kappa'\lambda$ -boundary S in any binary picture (V, π, f) is near-Jordan. Let $c \in II(S)$ and $d \in IE(S)$. We need to show that every π -path from c to d crosses S .

Let O be the κ -component and O' be the κ' -component of the set of 1-spels in (V, π, f) which contain c and let Q be the λ -component of the set of 0-spels in (V, π, f) which contains d . Then $S = \vartheta(O', Q)$. Since $\kappa \subseteq \kappa'$, we have $O \subseteq O'$, and $\vartheta(O, Q) \subseteq S$. Furthermore, $\vartheta(O, Q)$ is not empty, since the fact that c (which is in O) is in $II(S)$ implies that there must be an e in $IE(S)$ (and hence in Q) such that $(c, e) \in \vartheta(O, Q)$. The fact that (κ, λ) is a desirable pair for the digital space (V, π) implies that the $\kappa\lambda$ -boundary $\vartheta(O, Q)$ is near-Jordan. By Theorem 3.6, we also know that c , which is in O , is in the interior of $\vartheta(O, Q)$ and d , which is in Q , is in the exterior of $\vartheta(O, Q)$. By Lemma 3.3, every π -path from c to d crosses $\vartheta(O, Q)$ and hence crosses S since S is a superset of $\vartheta(O, Q)$. \square

The followings are immediate consequences of the last two theorems and Theorem 3.8.

Corollary 3.24. *For $N \geq 1$, every $\delta_N\omega_N$ -boundary in every picture over the digital space (Z^N, ω_N) is near-Jordan.*

Corollary 3.25. *For $N \geq 1$, (δ_N, ω_N) is a desirable pair for (Z^N, ω_N) .*

Theorem 3.26. *In the digital space (Z^3, ω_3) , every $\sigma\sigma$ -boundary in every picture is near-Jordan.*

Proof. Let S be an arbitrary $\sigma\sigma$ -boundary in a picture over (Z^3, ω_3) . Let O be a σ -component of 1-spels and Q be a σ -component of 0-spels such that $S = \vartheta(O, Q)$. According to Theorem 3.18, all we have to do is to show that S is locally-Jordan. We use Lemma 3.20. Let $\langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)} \rangle$ be an ω_3 -path of length 3 such that $(c^{(0)}, c^{(3)}) \in S$, $c^{(0)} \neq c^{(2)}$, $c^{(1)} \neq c^{(3)}$, $c^{(1)}$ is a 0-spel and $c^{(2)}$ is a 1-spel. Now all we need to show is that $p_S P = 1$.

Just as in the proof of Theorem 3.21, we get that there exists a pair of integers i, j ($1 \leq i \neq j \leq 3$) and u, v ($|u| = |v| = 1$) such that (3.35) is satisfied. It follows that

$$(3.36) \quad \left(c_i^{(2)} - c_i^{(0)} \right) \times \left(c_j^{(2)} - c_j^{(0)} \right) = u \times v = - \left(c_i^{(3)} - c_i^{(1)} \right) \times \left(c_j^{(3)} - c_j^{(1)} \right)$$

and $c_k^2 = c_k^0$ and $c_k^3 = c_k^1$, if k is neither i nor j .

Now observing the definition of σ in Subsection 3.1 we see that exactly one of the diagonals $(c^{(0)}, c^{(2)})$ and $(c^{(1)}, c^{(3)})$ is in σ . If $(c^{(0)}, c^{(2)}) \in \sigma$, then $c^{(2)} \in O$ and we see that $p_S P = 1$, whether or not $c^{(1)} \in Q$. If $(c^{(1)}, c^{(3)}) \in \sigma$, then $c^{(1)} \in Q$, and again $p_S P = 1$. \square

Corollary 3.27. *In the digital space (Z^3, ω_3) , (σ, σ) is a desirable pair.*

Theorem 3.28. *Let ρ be a spel-adjacency in a digital space (V, π) such that (ρ, ρ) is desirable for (V, π) . Then (ρ, ρ) is desirable for (V, ρ) .*

Proof. By Theorem 3.8, all we need to show is that every $\rho\rho$ -boundary S in every binary picture (V, ρ, f) is near-Jordan. Let O be the ρ -component of 1-spels and Q be the ρ -component of 0-spels such that $S = \vartheta_\rho(O, Q)$.

Consider, $T = \vartheta_\pi(O, Q) = \vartheta_\rho(O, Q) \cap \pi$. We want to show first that T is a near-Jordan surface in (V, π) . For this it is sufficient to show that T is not empty, since in that case the desirability of (ρ, ρ) for (V, π) provides us what we seek.

We know that S is not empty (by the definition of a boundary). Let $(c, d) \in S$, and hence $(c, d) \in \rho$. By the definition of a spel-adjacency, there exists a π -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ from c to d such that $(c^{(0)}, c^{(k)}) \in \rho$, for $1 \leq k \leq K$. Let $c^{(k-1)}$ be the last 1-spel on this path; it exists since c is a 1-spel and d is a 0-spel. Then $c^{(k-1)}$ must be in O because $c^{(0)} \in O$ is ρ -adjacent to it, and $c^{(k)}$ must be in Q because $\langle c^{(k)}, \dots, c^{(K)} \rangle$ is a ρ -path of 0s and $c^{(K)}$ is in Q . This shows that $(c^{(k-1)}, c^{(k)}) \in T$.

Let $\langle c^{(0)}, \dots, c^{(K)} \rangle$ be an arbitrary ρ -path from the immediate interior of S to the immediate exterior of S . All we need to do now is to show that it crosses S .

We adopt the convention that when we talk about the interior of S , we mean its interior in (V, ρ) , and when we talk about the interior of T , we mean its

interior in (V, π) . Similar conventions apply to exteriors, immediate interiors and immediate exteriors.

By Theorem 3.6, $O \subseteq I(T)$ and $Q \subseteq E(T)$. By definition, $II(S) \subseteq O$ and $IE(S) \subseteq Q$. Therefore, $c^{(0)} \in I(T)$ and $c^{(K)} \in E(T)$. In view of Lemma 3.2, there must exist a k , $1 \leq k \leq K$, such that $c^{(k-1)} \in I(T)$ and $c^{(k)} \in E(T)$. Our proof is complete if we can show that in fact $c^{(k-1)} \in O$ and $c^{(k)} \in Q$, since this implies that $(c^{(k-1)}, c^{(k)}) \in S$.

We now show that $c^{(k-1)} \in O$. By the definition of a spel-adjacency, there exists a π -path $\langle e^{(0)}, \dots, e^{(L)} \rangle$ from $c^{(k-1)}$ to $c^{(k)}$ such that $(c^{(k-1)}, e^{(l)}) \in \rho$ for $0 \leq l \leq L$. Since this a π -path from the interior to the exterior of the surface T , which is near-Jordan in (V, π) , there must be an l , $1 \leq l \leq L$, such that $(e^{(l-1)}, e^{(l)}) \in T$. If $l = 1$, then $c^{(k-1)} = e^{(l-1)}$ is in O and we are done. Otherwise, if $c^{(k-1)}$ were a 0-spel, then it would be in Q (since it is ρ -adjacent to the spel $e^{(l)}$ in Q) and hence in $E(T)$. This would contradict Lemma 3.3(iii). Therefore, $c^{(k-1)}$ must be a 1-spel. Since it is ρ -adjacent to the spel $e^{(l-1)}$ in O , it itself must be in O .

To show that $c^{(k)} \in Q$, essentially we can repeat the same argument using a π -path from $c^{(k)}$ to $c^{(k-1)}$ such that $c^{(k)}$ is ρ -adjacent to every other point on the path. \square

From this theorem and the corollary that precedes it we get the following.

Corollary 3.29. *For the digital space (Z^3, σ) , (σ, σ) is a desirable pair.*

Theorem 3.30. *For the digital space (F, δ_3) , (δ_3, δ_3) is a desirable pair.*

Proof. By Theorem 3.9 (Z^3, σ) is isomorphic to (F, δ_3) , under an isomorphism i defined in the proof of that theorem. Since $i(\sigma) = \delta_3$, the previous corollary and Theorem 3.11 provide the result we seek. \square

Corollary 3.22 restates the Desirability Claim 3.1 for Case (i) in Subsection 3.1, Case (ii) is provided by Corollary 3.25, while Case (iii) is provided by Theorem 3.30.

4. CORRECTNESS OF TRACKING ALGORITHMS

4.1. General boundary tracking.

Let O be the set of all 1-spels and Q be the set of all 0-spels in a picture (V, π, f) . Recall that we refer to elements of $B = \vartheta(O, Q)$ as *bels* (short for boundary elements). Since, by the definition of a picture, O is finite, so is B . We now generalize the idea of a bel-adjacency (first introduced in Subsection 2.1).

We say that the binary relation β on B is a *bel-adjacency* if β is an adjacency (i.e., β^* is symmetric).

We propose an algorithm for the following task: Given a binary picture (V, π, f) , a bel-adjacency β , and a bel o , find the β -component S of B that contains o .

Algorithm 4.1. *General Bel-Tracking Algorithm*

- (1) Put o into L and S and put $in_\beta(o)$ copies of o into T .
- (2) Remove a bel a from L . For all bels b which are β -adjacent from a , try to find a copy of b in T .
 - (a) If successful, remove this copy of b from T .
 - (b) If not, then put b into L and S and put $in_\beta(b) - 1$ copies of b into T .
- (3) Check if L is empty.
 - (a) If it is, STOP.
 - (b) If it is not, start again at Instruction (2).

Prior to proving the correctness of this algorithm, two remarks are in order. First, in Instruction (2)b, $in_\beta(b) - 1$ is guaranteed to be nonnegative. This is because we only get to this point in the algorithm for a b which is β -adjacent from a bel a and, so, $in_\beta(b) \geq 1$. Second, the potential efficiency of this algorithm comes from the fact that we check for membership in T (rather than in S). While S keeps getting bigger and bigger as the algorithm is executed, due to Instruction (2)b, the same is not true for T : elements from T will be repeatedly removed due to Instruction (2)a. Hence the size of T is likely to be a small fraction of the size of S after the algorithm has been performing for a while on a large data set.

The essence of the proof of correctness is given in the next lemma. To state it easily we use a couple of abbreviations. We let $n_T(b)$ abbreviate “the number of copies of the bel b in the list T .” The other definition is more complicated. The value of $n_a(b)$ is 0 if $b \notin S$ and is “ $in_\beta(b)$ less the number of bels in $S - L$ which are β -adjacent to the bel b ” otherwise.

Lemma 4.2. *Both just prior and just after the execution of Instruction (2) in the General Bel-Tracking Algorithm it is the case that, for every bel b ,*

- (i) $n_T(b) = n_a(b)$,
- (ii) the bel b has so far been put into L and S - either due to Instruction (1) or due to Instruction (2)b - at most once, and
- (iii) if the bel b is in S , then o is β -connected in B to b .

Proof. Consider the situation just after the execution of Instruction (1). We have that $n_T(o) = in_\beta(o) = n_a(o)$ and the bel o has so far been put into L and S exactly once. For any bel b other than o , $n_T(b) = 0 = n_a(b)$ and b has not so far been put into L and S even once. Since only o has been put into S , (iii) is clearly satisfied at this time.

Now we show that if (i), (ii) and (iii) hold just prior the execution of Instruction (2), then they also hold just after its execution. This is sufficient, since the situation cannot change as a result of Instruction (3). Assume therefore that we are just at the beginning of executing Instruction (2). This means that at this time L is not empty and so we remove a bel a from it. This a must have been put into L and S earlier on and, since nothing is ever removed from S , we must have that $a \in S$. We leave it to the reader to supply the easy proof

of the fact that for those bels which are not β -adjacent from a , the inductive step is valid. For the bels b which are β -adjacent from a , we study separately two possibilities.

Case a: we find a copy of b in T . In this case, by Instruction (2)a, we remove this b from T . This reduces $n_T(b)$ by 1. Since b was in T , it also had to be in S (nothing is ever put into T without being put into S at the same time). Therefore the applicable part of the definition of $n_a(b)$ is that it is $in_\beta(b)$ less the number of bels in $S - L$ which are β -adjacent to the bel b . The only thing that changes in this definition is that a , which is β -adjacent to b , got removed from L . Hence $n_a(b)$ is also decreased by 1, proving (i) of the lemma for the bel b . Since nothing is put into L and S , (ii) and (iii) are also valid at the end of executing Instruction (2).

Case b: there is no copy of b in T ; i.e., $n_T(b) = 0$. First we show that under these circumstances, it cannot be the case that b has been previously put into L and S . This is so, since otherwise prior to the beginning of Instruction (2), the applicable part of the definition of $n_a(b)$ would be “ $in_\beta(b)$ less the number of bels in $S - L$ which are β -adjacent to the bel b .” Since at that time the bel a is still in L , the value of $n_a(b)$ has to be positive, contradicting the truth of (i) in the induction hypothesis. Upon executing Instruction (2)b, b has been put into L and S (for the first time) and $n_T(b) = in_\beta(b) - 1$. There is at least one bel, namely a , which is in $S - L$ and is β -adjacent to b . There cannot be another one, since whenever a bel is put into S , it is also put into L at the same time, and so if at a later time it is no longer in L , then it must have been removed from it. At that time, b would have been put into L and S and we would not be in Case b. This shows that in this case too, $n_T(b) = n_a(b)$ just after the execution of Instruction (2). Finally, since $a \in S$ just prior to the execution of Instruction (2), o is β -connected in B to a , by (iii) of the induction hypothesis. The same must be true for the new bel b in S , since a is β -adjacent to it. \square

Theorem 4.3. *If (V, π, f) is a binary picture, β is a bel-adjacency and o is a bel, then the General Bel-Tracking Algorithm terminates in a finite number of steps and, at that time, S is the β -component of B that contains o .*

Proof. From (iii) of the previous lemma it follows that anything that gets put into S is in the β -component of B which contains o . Termination in a finite number of steps now easily follows from (ii) of the previous lemma: since each of the finitely many bels in the β -component of B which contains o is put into L at most once (and nothing else ever gets put into L) and in each execution of Instruction (2) of the algorithm a bel is removed from L , sooner or later L has to become empty and the algorithm will stop due to Instruction (3). At that time, as all through the execution of the algorithm, o is β -connected in B to every element of S . That the converse is also true (and hence S is the β -component of B which contains o) can be shown as follows. For any bel b of the β -component of B which contains o , there is a β -path $\langle b^{(0)}, \dots, b^{(K)} \rangle$ from o to b . It is a trivial matter to show by induction that, for $1 \leq k \leq K$, $b^{(k-1)}$ will get put into L and S and, since L gets eventually emptied, $b^{(k-1)}$ must

get removed from L , resulting in $b^{(k)}$ being put into L and S (provided that it is not in T , which would imply that it has been put into L and S in some previous step). \square

This theorem shows that the General Bel-Tracking Algorithm is powerful stuff. Its practical usefulness depends on two properties of the bel-adjacency β . The first is, how easy is it to compute the bels β -adjacent from a given bel? Clearly, the efficiency of executing Instruction (2) depends on this (as well as on how easy it is to determine for a bel whether or not it is in T). The other property has to do with the usefulness of the resulting boundaries: are they in fact of the form $\vartheta(O, Q)$ for some appropriately specified O and Q ? In the following subsection we will discuss these questions for cases specified in Subsection 2.6.

4.2. Boundary tracking on hyper-cubes.

We now apply the General Bel-Tracking Algorithm to the tracking of boundaries in the spaces (Z^N, ω_N) with $N \geq 2$. A *basic digraph for Z^N* is a pair (M, ρ) for which:

- (i) $M = \{u^n \mid 1 \leq n \leq N\} \cup \{-u^n \mid 1 \leq n \leq N\}$;
- (ii) $\rho = \bigcup_{1 \leq i < j \leq N} \rho_{i,j}$, where $\rho_{i,j}$ is exactly one of
 - (a) $\{(u^i, u^j), (u^j, -u^i), (-u^i, -u^j), (-u^j, u^i)\}$,
 - (b) \emptyset ;

Note that it is permitted in the definition of ρ that different options ((a) or (b)) be selected for specifying $\rho_{i,j}$ for different pairs of i and j . Because of this, the definition given here is a shared generalization of two definitions given earlier: in Subsection 2.4 the specification uses Option (a) in all cases, and in Subsection 2.5 (in which only $N = 3$ is considered), the specification uses Option (a) for the pairs $(i, j) \in \{(1, 2), (1, 3)\}$ but uses Option (b) for the pair $(2, 3)$.

From now the phrase “basic digraph” will be used for the concept as defined in the previous paragraph, with the further assumption that Option (a) is to be used for at least one pair (i, j) .

The method of defining β from the basic digraph is described in the two paragraphs above Algorithm 2.4. (That definition is for Option (a), in case of Option (b) the corresponding $\beta_{i,j} = \emptyset$). This definition provides a satisfactory reply to one of the questions raised in the last paragraph of the previous subsection: how easy is it to compute the bels β -adjacent from a given bel? For the definition given in this subsection the answer is “it is very easy.” A basic digraph is an easily represented object: for each of the $2N$ nodes, there are at most $N - 1$ nodes adjacent from it or to it. For any bel $b = (c, d)$, we have for some i ($1 \leq i \leq N$) that $d - c \in \{u^i, -u^i\}$. For each (of the $N - 1$) $j \neq i$, there may (or may not) be a u which is $\rho_{i,j}$ -adjacent from $d - c$. If there is such a u , it gives rise to a single bel $\beta_{i,j}$ -adjacent from (c, d) and the specification of

this bel (as illustrated in Figure 10) requires the checking of the value of the binary function f at at most two spels.

A most important observation for our choice of β is the following. If Option (b) is not chosen for any (i, j) , then the General Bel-Tracking Algorithm (Algorithm 4.1) is exactly Artzy's Algorithm in N -dimensional space (Algorithm 2.4). If $N = 3$ and Option (b) is chosen only for $(2, 3)$, then the General Bel-Tracking Algorithm is exactly the Gordon-Udupa Algorithm (Algorithm 2.6).

The behavior of the General Bel-Tracking Algorithm (as expressed by Theorem 4.3) is guaranteed only if the adjacency β is a bel-adjacency; i.e., if β^* is symmetric. Prior to discussing whether or not this is the case for the $\beta = \bigcup_{1 \leq i < j \leq N} \beta_{i,j}$, we prove a technical lemma of a more general applicability.

Lemma 4.4. *Let ρ be a binary relation on a set M and L be a finite subset of M .*

- (i) *The transitive closure ρ^* of ρ is symmetric, provided that, for all $(c, d) \in \rho$, (d, c) is in ρ .*
- (ii) *If every element of L has exactly one element of L ρ -adjacent from it and at least one element of L ρ -adjacent to it, then every element of L has exactly one element of L ρ -adjacent to it.*
- (iii) *If every element of L has exactly one element of L ρ -adjacent from it and exactly one element of L ρ -adjacent to it, then for any c and d in L such that c is ρ -adjacent to d we have that d is ρ -connected in L to c .*

Proof. Assume that, for all $(c, d) \in \rho$, (d, c) is in ρ^* . We prove by induction on K that if $\langle c^{(0)}, \dots, c^{(K)} \rangle$ is a ρ -path, then $(c^{(K)}, c^{(0)}) \in \rho^*$. This is clearly the case, by the definition of transitive closure, if $K = 0$. If $K > 0$, then $(c^{(K)}, c^{(K-1)}) \in \rho^*$ (by the assumption) and $(c^{(K-1)}, c^{(0)}) \in \rho^*$ (by the induction hypothesis). Since ρ^* is clearly transitive, we have that $(c^{(K)}, c^{(0)}) \in \rho^*$. This proves (i).

We now define the binary relation τ on L by $(c, d) \in \tau$ if, and only if, $c \in L$, $d \in L$, and $(c, d) \in \rho$. Since it is assumed in (ii) that every element of L has exactly one element of L ρ -adjacent from it, the number of elements in τ must be exactly the number of elements in L . This shows that there cannot be an element of L which has more than one element of L ρ -adjacent to it, since otherwise we would have more elements in τ than in L (since it is also assumed that every element of L has at least one element of L ρ -adjacent to it).

To complete the proof, let us assume that the premise of (iii) is satisfied and c and d in L are such that c is ρ -adjacent to d . We define an infinite sequence of elements of L as follows: $c^{(0)} = d$ and, for $k \geq 1$, $c^{(k)}$ is the unique element of L that is ρ -adjacent from $c^{(k-1)}$. Since L is finite, there must be a smallest positive integer l such that $c^{(l)} = c^{(m)}$, for some $m < l$. If m were positive, then the uniqueness of the element of L that is ρ -adjacent to $c^{(l)} = c^{(m)}$ would imply that $c^{(l-1)} = c^{(m-1)}$, which would contradict the minimality of l . Hence,

$c^{(l)} = c^{(0)} = d$. Again by the uniqueness of the element of L which is ρ -adjacent to $c^{(l)} = d$, we must have that $c^{(l-1)} = c$, and so $\langle c^{(0)}, \dots, c^{(l-1)} \rangle$ is a ρ -path in L from d to c . \square

Theorem 4.5. *For $N \geq 2$, let $(M, \bigcup_{1 \leq i < j \leq N} \rho_{i,j})$ be a basic digraph for Z^N . Assume that the set of all bels B in the picture (Z^N, ω_N, f) is not empty and, for $1 \leq i < j \leq N$, define the binary relation $\beta_{i,j}$ on B as above. Let $B_{i,j}$ be the subset of those bels $b = (c, d)$ for which $d - c \in \{u^i, -u^i, u^j, -u^j\}$. Then, for $1 \leq i < j \leq N$ such that $\rho_{i,j} \neq \emptyset$ and for every bel b in $B_{i,j}$, the following hold.*

- (i) *There is one, and only one, bel b' that is $\beta_{i,j}$ -adjacent from b . Furthermore, $b' \in B_{i,j}$.*
- (ii) *There is one, and only one, bel b' in $B_{i,j}$ that is $\beta_{i,j}$ -adjacent to b . Furthermore, b is $\beta_{i,j}$ -connected in $B_{i,j}$ to b' .*

Moreover, $\beta = \bigcup_{1 \leq i < j \leq N} \beta_{i,j}$ is a bel-adjacency for (Z^N, ω_N, f) .

Proof. By the definition of $\rho_{i,j}$, there is one and only one u which is $\rho_{i,j}$ -adjacent from $d - c$ and this u is in $\{u^i, -u^i, u^j, -u^j\}$. According to the definition of $\beta_{i,j}$, u gives rise to a unique bel $b' = (c', d')$ that is $\beta_{i,j}$ -adjacent from (c, d) . Looking at the three possibilities in the definition of $\beta_{i,j}$ (c.f., Figure 10), we see that $d' - c'$ is (respectively) $d - (d + u) = -u$, $(d + u) - (c + u) = d - c$, or $(c + u) - c = u$. In any case, $b' \in B_{i,j}$. This proves (i).

In order to prove (ii) we first distinguish between three mutually exclusive possibilities (corresponding to the three cases represented in Figure 10) and show that in each one of them there is at least one bel $b' = (c', d')$ in $B_{i,j}$ that is $\beta_{i,j}$ -adjacent to b . We again denote by u the element of $\{u^i, -u^i, u^j, -u^j\}$ which is $\rho_{i,j}$ -adjacent from $d - c$.

The first possibility is that $d - u$ is a 1-spel. In this case we define $c' = d - u$ and $d' = d$. Then $b' = (c', d')$ is a bel, $d' - c' = u$, and so $b' \in B_{i,j}$. From the definition of $\rho_{i,j}$ it follows (since u is $\rho_{i,j}$ -adjacent from $d - c$) that $u' = c - d$ is $\rho_{i,j}$ -adjacent from $d' - c' = u$. Since $d' + u' = d + (c - d) = c$ is a 1-spel, the definition of $\beta_{i,j}$ implies $b' = (c', d')$ is $\beta_{i,j}$ -adjacent to $(d' + u', d') = (c, d) = b$.

The second possibility is that $d - u$ is a 0-spel and $c - u$ is a 1-spel. In this case we define $c' = c - u$ and $d' = d - u$. Then $b' = (c', d')$ is a bel, $d' - c' = d - c$, and so $b' \in B_{i,j}$. Since u is $\rho_{i,j}$ -adjacent from $d - c = d' - c'$ and $d' + u = d$ is a 0-spel and $c' + u = c$ is a 1-spel, the definition of $\beta_{i,j}$ implies $b' = (c', d')$ is $\beta_{i,j}$ -adjacent to $(c' + u, d' + u) = (c, d) = b$.

The third possibility is that both $d - u$ and $c - u$ are 0-spels. In this case we define $c' = c$ and $d' = c - u$. Then $b' = (c', d')$ is a bel, $d' - c' = -u$, and so $b' \in B_{i,j}$. From the definition of $\rho_{i,j}$ it follows (since u is $\rho_{i,j}$ -adjacent from $d - c$) that $u' = d - c$ is $\rho_{i,j}$ -adjacent from $d' - c' = -u$. Since $d' + u' = (c - u) + (d - c) = d - u$ and $c' + u' = c + (d - c) = d$ are both 0-spels, the definition of $\beta_{i,j}$ implies $b' = (c', d')$ is $\beta_{i,j}$ -adjacent to $(c', c' + u') = (c, d) = b$.

This shows that, for any bel b , there is at least one bel b' in $B_{i,j}$ that is $\beta_{i,j}$ -adjacent to b . According to (i), it is also the case that there is exactly one bel b' in $B_{i,j}$ that is $\beta_{i,j}$ -adjacent from b . Applying (ii) of the previous lemma we get that there is one, and only one, bel b' in $B_{i,j}$ that is $\beta_{i,j}$ -adjacent to b . Hence we can also apply (iii) of the previous lemma and obtain that b is $\beta_{i,j}$ -connected in $B_{i,j}$ to b' . This completes the proof of (ii).

To complete the proof, we make use of (i) of the previous lemma. If $(a, b) \in \beta$, then $(a, b) \in \beta_{i,j}$, for some $1 \leq i < j \leq N$ such that $\rho_{i,j} \neq \emptyset$. By (ii), this implies that b is $\beta_{i,j}$ -connected in $B_{i,j}$ to a and, consequently, $(b, a) \in \beta^*$. Hence, β^* is symmetric. \square

Corollary 4.6. *For $N \geq 2$, let (M, ρ) be a basic digraph for Z^N . Define the binary relation β on B as above. It is then the case that, for any bel o , the General Bel-Tracking Algorithm terminates in a finite number of steps and, at that time, S is that β -component of the set of bels that contains o .*

The following theorem takes care of the Correctness Claim 2.8 for $G = Z^N$, $\kappa = \alpha_N$ and $\lambda = \omega_N$.

Theorem 4.7. *For $N \geq 2$, let $(M, \bigcup_{1 \leq i < j \leq N} \rho_{i,j})$ be a basic digraph for Z^N such that $\rho_{i,j} \neq \emptyset$ for any $1 \leq i < j \leq N$. Define the binary relation β on B as above. It is then the case that, for any bel $o = (c, d)$, the General Bel-Tracking Algorithm terminates in a finite number of steps and, at that time, $S = \vartheta(O, Q)$, where O is the δ_N -component of the set of 1-spels that contains c and Q is the ω_N -component of the set of 0-spels that contains d .*

Proof. By the previous corollary we know that the algorithm terminates in a finite number of steps and that at that time S is that β -component of B that contains o .

First we show inductively that both just prior and just after the execution of Instruction (2) in the General Bel-Tracking Algorithm $L \subseteq \vartheta(O, Q)$ and $S \subseteq \vartheta(O, Q)$. This is true just prior to the first execution of Instruction (2), since only o gets put into L and S due to Instruction (1) and, clearly, $o = (c, d) \in \vartheta(O, Q)$. Assume now the validity of the induction hypothesis just prior to some execution of Instruction (2). If a bel b gets put into L and S during the execution of Instruction (2), then there must have been an a removed from $L \subseteq \vartheta(O, Q)$ such that b is $\beta_{i,j}$ -adjacent from a . By considering the three possible ways that a can be $\beta_{i,j}$ -adjacent to b (look at Figure 10) we see that $a \in \vartheta(O, Q)$ implies that $b \in \vartheta(O, Q)$. (Essential use is made here of the fact that O is a δ_N -component of the set of 1-spels.) Since nothing is put into L or S during the execution Instruction (3), this completes the induction.

Next we show that at the termination of the General Bel-Tracking Algorithm S is locally-Jordan. By definition, we need to show that $p_S \langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)} \rangle$ is odd for any ω_N -path such that $(c^{(0)}, c^{(3)}) \in S$. (We have made use here of the fact that there is no ω_N -path of length 2 from a spel to a spel ω_N -adjacent from it.) The sought-after result is trivially true if $c^{(0)} = c^{(2)}$ or $c^{(1)} = c^{(3)}$. When neither of these is the case, then the loop $\langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)}, c^{(0)} \rangle$ satisfies (3.35)

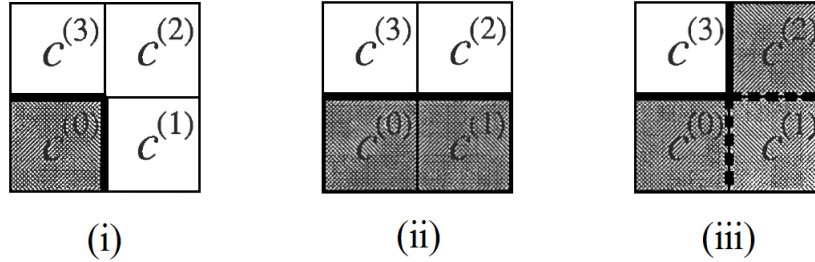


FIGURE 16. Illustration of the three cases in the proof which shows that the output of the General Bel-Tracking Algorithm based on the basic digraph $(M, \bigcup_{1 \leq i < j \leq N} \rho_{i,j})$ is locally-Jordan. The spels shaded dark grey are 1-spels, the spel shaded light grey may be either a 1-spel or a 0-spel, all other spels are 0-spels.

for some u and v such that $|u| = |v| = 1$. This implies that, for some $1 \leq i < j \leq N$, $c^{(3)} - c^{(0)} = c^{(2)} - c^{(1)} \in \{u^i, -u^i, u^j, -u^j\}$ and $c^{(3)} - c^{(2)} = c^{(0)} - c^{(1)} \in \{u^i, -u^i, u^j, -u^j\}$; see Figure 16. In that figure we distinguish between three cases, corresponding to the three cases of Figure 10. In each case, the bel $(c^{(0)}, c^{(3)})$ in Figure 16 corresponds to the bel painted light grey in Figure 10. In Case (i), $f(c^{(1)}) = f(c^{(2)}) = 0$. Both the bels $(c^{(0)}, c^{(3)})$ and $(c^{(0)}, c^{(1)})$ are in $B_{i,j}$ (as defined in the statement of the previous theorem) and, since $\rho_{i,j} \neq \emptyset$, one of them has to be $\beta_{i,j}$ -adjacent to the other. The previous theorem implies that they are in the same β -component of B . Since $(c^{(0)}, c^{(3)}) \in S$, this implies that $(c^{(0)}, c^{(1)}) \in S$. On the other hand, $(c^{(1)}, c^{(2)})$ and $(c^{(2)}, c^{(3)})$ are not bels and so cannot be in S . This proves that $p_S \langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)} \rangle = 1$ in Case (i). The proof in Case (ii) is very similar and is therefore not given. In Case (iii) there is an additional complication. One can argue, as in Case (i), to show that in Case (iii) $(c^{(2)}, c^{(3)}) \in S$, but the possibilities of $(c^{(0)}, c^{(1)}) \in S$ or $(c^{(1)}, c^{(2)}) \in S$ need also be investigated. Neither of these is true if $c^{(1)}$ is a 1-spel. On the other hand, if $c^{(1)}$ is a 0-spel, then both $(c^{(0)}, c^{(1)})$ and $(c^{(1)}, c^{(2)})$ are in $B_{i,j}$ and, consequently, either both are in S or neither is in S . In any case, we again have that $p_S \langle c^{(0)}, c^{(1)}, c^{(2)}, c^{(3)} \rangle$ is odd.

Next we show that at the termination of the General Bel-Tracking Algorithm S is near-Jordan. We use Lemma 3.16. We have just shown that S is a locally-Jordan surface in the digital space (Z^N, ω_N) , which is simply connected, by Theorem 3.12. Since S is a set of bels, it has to be antisymmetric. To show that it is near-Jordan, according to Lemma 3.16 it is sufficient to show that, for any two elements in the immediate exterior of S , there is an ω_N -path between them which does not cross S . Suppose that g and h are in $IE(S)$. That implies that there exist spels g' and h' , such that both (g', g) and (h', h) are in S . Since S

is a β -component of B , there is β -path in S from (g', g) to (h', h) . Since a β -path is a sequence of consecutively β -adjacent bels, the task of this paragraph is done if we can show that whenever a bel (g', g) is β -adjacent to a bel (h', h) , then there is an ω_N -path from g to h that does not cross B . This follows, since (g', g) must be $\beta_{i,j}$ -adjacent to (h', h) , for some $1 \leq i < j \leq N$, and in each of the three cases in the definition of $\beta_{i,j}$ the required result follows trivially. (For example, the three relevant ω_3 -paths in Figure 10 are: (i) $\langle d, d + u, c + u \rangle$, (ii) $\langle d, d + u \rangle$ and (iii) $\langle d \rangle$.)

Now we complete the proof by showing that at the termination of the General Bel-Tracking Algorithm $S = \vartheta(O, Q)$. We already know that $S \subseteq \vartheta(O, Q)$. To prove the converse, consider any $(g, h) \in \vartheta(O, Q)$; i.e., $g \in O$, $h \in Q$, $(g, h) \in \omega_N$. Since $g \in O$, there is a δ_N -path in O from c to g from which we can create an ω_N -path $\langle c^{(0)}, \dots, c^{(K)} \rangle$ from c to g with the following property: for $0 \leq k \leq K$, either $f(c^{(k)}) = 1$ or $f(c^{(k)}) = 0$ and $f(c^{(k-1)}) = f(c^{(k+1)}) = 1$. In the latter case, $(c^{(k-1)}, c^{(k)})$ and $(c^{(k+1)}, c^{(k)})$ are bels such that one of them is β -adjacent to the other and so either both are in S or neither is in S . It follows therefore that $p_S \langle c^{(0)}, \dots, c^{(K)} \rangle = 0$. Since $h \in Q$, there is an ω_N -path $\langle d^{(0)}, \dots, d^{(L)} \rangle$ in Q from h to d ; clearly $p_S \langle d^{(0)}, \dots, d^{(L)} \rangle = 0$. It follows that $(g, h) \in S$, for otherwise the crossing parity through the near-Jordan surface S would be even for the ω_N -path $\langle c = c^{(0)}, \dots, c^{(K)} = g, h = d^{(0)}, \dots, d^{(L)} = d \rangle$, contradicting Theorem 3.13. \square

This proves the correctness (Claim 2.8) of Artzy's Algorithm for N -dimensional spaces (Algorithm 2.4). For a discussion of the outputs of the Tracking Algorithm for the FCC Grid (Algorithm 2.3) and the Gordon-Udupa Algorithm (Algorithm 2.6) see pp. 194-199 of [13].

5. SUMMARY.

In this article we have motivated the general mathematical concept of a "digital space" by giving particular useful examples of it. We have demonstrated that the concept of a boundary can be defined in this general environment and that the resulting boundaries for specific spaces are "desirable" in the sense that they have properties similar to those specified by the Jordan curve theorem for simple closed curves. We have also proposed some algorithms for tracking such boundaries and proved some results about the behavior of such algorithms; in particular, we have proved the "correctness" of a boundary tracking algorithm for the N -dimensional cubic grid.

Acknowledgements. The author received a great deal of help in preparing this manuscript from T. Yung Kong and Laslo Cernetic. Numerous suggestions by three referees and the special issue editor T. Peters were also helpful in improving the clarity and comprehensiveness of the coverage.

REFERENCES

- [1] E. A. Abbott, *Flatland, A Romance of Many Dimensions*, Little, Brown and Co., Boston, 1899.
- [2] E. Artzy, G. Frieder and G. T. Herman, *The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm*, *Comput. Graph. Image Proc.* **15** (1981), 1–24.
- [3] L.-S. Chen, G. T. Herman, R. A. Reynolds and J. K. Udupa, *Surface shading in the cuberille environment*, *IEEE Comput. Graph. Appl.* **5**(12)(1985), 33–43. (Erratum appeared in **6**(2) (1986), 6749).
- [4] A. V. Evako, *Topological properties of closed digital spaces: One method of constructing digital models of closed continuous surfaces by using covers*, *Comput. Vision Image Underst.* **102** (2006), 134–144.
- [5] S. Fourey, T. Y. Kong and G. T. Herman, *Generic axiomatized digital surface-structures*, *Disc. Appl. Math.* **139** (2004), 65–93.
- [6] G. Frieder, G. T. Herman, C. Meyer and J. K. Udupa, *Large software problems for small computers: An example from medical imaging*, *IEEE Software* **2**(5) (1985), 37–47.
- [7] E. Garduño, G. T. Herman and H. Katz, *Boundary tracking in 3-D binary images to produce rhombic faces for a dodecahedral model*, *IEEE Trans. Med. Imag.* **17** (1998), 1097–1100.
- [8] D. Gordon and J. K. Udupa, *Fast surface tracking in three-dimensional binary images*, *Comput. Vision Graph. Image Proc.* **45** (1989), 196–241.
- [9] G. T. Herman, *A survey of 3D medical imaging technologies*, *IEEE Eng. Med. Biol. Mag.* **9**(4)(1990), 15–17.
- [10] G. T. Herman, *3D display: A survey from theory to applications*, *Comput. Med. Imag. Graph.* **17** (1993), 231–242.
- [11] G. T. Herman, *Oriented surfaces in digital spaces*, *CVGIP: Graph. Models Image Proc.* **55** (1993), 381–396.
- [12] G. T. Herman, *Finitary 1-simply connected digital spaces*, *Graph. Models Image Proc.* **60** (1998), 46–56.
- [13] G. T. Herman, *Geometry of Digital Spaces*. Birkhäuser, Boston, MA, 1998.
- [14] G. T. Herman and D. F. Robinson, *Digital boundary tracking*, *Pattern Anal. Appl.* **1** (1998), 2–17.
- [15] G. T. Herman and D. Webster, *A topological proof of a surface tracking algorithm*, *Comput. Vision Graph. Image Proc.* **23** (1983), 162–177.
- [16] T. Y. Kong, *Topological adjacency relations on Z^N* , *Theor. Comput. Sci.* **283** (2002), 3–28.
- [17] T. Y. Kong, *The Khalimsky topologies are precisely those simply connected topologies on Z^N whose connected sets include all $2n$ -connected sets but no $(3(n) - 1)$ -disconnected sets*, *Theor. Comput. Sci.* **305** (2003), 221–235.
- [18] T. Y. Kong and J. K. Udupa, *A justification of a fast surface tracking algorithm*, *CVGIP: Graph. Models Image Proc.* **54** (1992), 507–515.
- [19] R. Kopperman, *On storage of topological information*, *Disc. Appl. Math.* **147** (2005), 287–300.
- [20] R. Kopperman, P. R. Meyer and R. G. Wilson, *A Jordan surface theorem for three-dimensional digital spaces*, *Disc. Comput. Geom.* **6** (1991), 155–161.
- [21] J.-O. Lachaud and A. Montever, *Continuous analogs of digital boundaries: A topological approach to iso-surfaces*, *Graph. Models* **62** (2000), 129–164.
- [22] W. Lorensen and H. Cline, *Marching cubes: A high-resolution 3D surface construction algorithm*, *Comput. Graph.* **21**(4) (1987), 163–169.
- [23] J. R. Munkres, *Topology: A First Course*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [24] A. Ralston, E. D. Reilly and D. Hemmendinger, *Encyclopedia of Computer Science*, Nature Publishing Group, London, 4th edition, 2000.

- [25] J. K. Udupa and G. T. Herman, *3D Imaging in Medicine*, CRC Press, Boca Raton, FL, 2nd edition, 2000.

GABOR T. HERMAN (GaborTHerman@yahoo.com)

PhD Program in Computer Science, The Graduate Center, City University of
New York, 365 5th Avenue, New York, NY, 10016, USA.