

# Digitization Schemes and the Recognition of Digital Straight Lines, Hyperplanes, and Flats in Arbitrary Dimensions

IVAN STOJMEŃOVIĆ AND RATKO TOŠIĆ

**ABSTRACT.** New digitization schemes for line segments, hyperplanes and (in general) flats in a Euclidean space are introduced in the same way for arbitrary dimensions. Using these schemes, algorithms are presented that determine whether or not a finite set  $S$  of  $n$  digital points in a  $k$ -dimensional space is a digital line segment, digital hyperplane, or a digital  $m$ -flat ( $1 \leq m < k$ ). A digital line segment is recognized in  $O(kn)$  time by showing that  $S$  is a digital line segment if and only if  $k-1$  of its projections into specific two-dimensional coordinate planes are (two dimensional) line segments. We present two algorithms to determine whether  $S$  is a digital hyperplane. One runs in polynomial time (for  $k \leq 3$  the time is  $O(n \log n)$ ), and is based on the convex hull construction and polytope intersection detection, while the other runs in linear time (but with large constants) and is based on a linear programming technique. A general method for recognizing digital  $m$ -flats is also outlined. While the results for two-dimensional space are well known in the literature, the results for three- and higher-dimensional cases are improvements over known results in several aspects: consistency of definition of digital line segments and digital arcs, the time complexity for the digital plane recognition problem and simplicity, uniformity and generality of all results.

## 1. Introduction

Digital geometry has applications in image processing, pattern recognition, computer vision, and other areas. Before an image of an object is processed by a computer, it is digitized to yield a finite digital subset also called a digital image.

The following two problems are of interest in digital geometry. One is the problem of digitization, that is, how to represent continuous objects using

---

1980 *Mathematics Subject Classification* (1985 Revision). Primary 68U05, 68Q25.

Research of the first author was supported in part by the National Science and Engineering Research Council of Canada under grant OGPIN 007.

This paper is in final form, and no version of it will be submitted for publication elsewhere.

finite digital sets. The other is the problem of retrieving from a digital set information about the object represented by it.

The geometric properties of 2-D digital images have been extensively studied in the literature. Some extensions to 3-D digital geometry have also been investigated in [K1, K2, KR2, MR]. Convex digital solids are characterized in [KR2] while digital surfaces are studied in [MR]. Digitization schemes and the recognition algorithms for digital straight line segments and planes in 3-D space are analyzed in [K1] and [K2], respectively. The common notions of 2-D and 3-D digital geometry (grids, metrics, point products, grid point digitizations, digital straight lines and hyperplanes) are extended to the  $k$ -D case in [K].

In this paper we study basic objects in an arbitrary dimensional Euclidean space: line segments, hyperplanes, flats, arcs and surfaces. We give digitization schemes and characterize them in a way that permits their fast recognition. The general approach taken here also gives more insight into the 3-D case.

We show that the digitization scheme and the recognition algorithm for (digital) straight lines given in [K1] for 3-dimensional space and generalized in [K] for arbitrary dimensions are not consistent. The problem is that under their digitization schemes a digital line segment is not necessarily a digital arc; consequently the two-dimensional projections of digital straight lines are not always digital lines in the plane. We present a new digitization scheme and prove that it satisfies the property that a digital line segment is a digital arc. The digital line segment recognition algorithms announced in [K1, K] can be safely used under our new digitization scheme. [K2, K] also propose digitization schemes for hyperplanes. For the three-dimensional case [K2] suggests an algorithm (with time complexity  $O(n^2)$ , where  $n$  is the number of digital points on the digital plane) for the recognition of digital planes. We introduce two new schemes and show that they are equivalent with the known one. The advantages of new schemes are faster digitization and faster recognition of (digital) hyperplanes (in fact, we propose two faster recognition algorithms, for arbitrary dimensions). We shall also present a generalized digitization scheme and recognition algorithm for  $m$ -flat in  $k$ -dimensional space.

We now introduce some basic definitions, notations and terminology.

The set of all points in  $k$ -dimensional Euclidean space is denoted by  $E^k$ . By  $D^k$  we denote the set of all points in  $E^k$  with integer-valued coordinates. A point in  $D^k$  is called a digital point. Any subset of  $D^k$  is called a digital set.

A function that is often employed for the analysis of digital sets is referred to as the chess board distance. It is defined in the following way.

For any two points of  $E^k$ ,  $A(x_1, \dots, x_k)$  and  $B(x'_1, x'_2, \dots, x'_k)$ ,  $d(A, B) = \max_{1 \leq i \leq k} |x_i - x'_i|$ .

Two points  $A$  and  $B$  of  $E^k$  are said to be *near each other* if  $d(A, B) < 1$ .

If only points from  $D^k$  are considered then points  $A$  and  $B$  are called *neighbors* if  $d(A, B) = 1$ . Thus, each point of  $D^k$  has  $3^k - 1$  neighbors.

A finite digital set  $R$  of  $n$  points is called a *digital arc* if it can be represented by a sequence  $(d_1, \dots, d_n)$  such that two points  $d_i$  and  $d_j$  are neighbors if and only if  $|i - j| = 1$ .

The coordinate axes of  $E^k$  are denoted by  $x_1, \dots, x_k$  (in  $E^3$  they will be called the  $x$ ,  $y$ , and  $z$ -axes as usual). The coordinate hyperplane orthogonal to the  $x_j$ -axis is defined by the equation  $x_j = 0$ . The two dimensional coordinate plane  $x_i x_j$  is the plane containing the axes  $x_i$  and  $x_j$ ; this plane is given by equations  $x_m = 0$  for all  $m, 1 \leq m \leq k, m \neq i, m \neq j$ .

This paper is organized as follows. Section 2 presents a new digitization scheme for digital lines in  $E^k$  and proposes an optimal and simple line segment recognition procedure. The three-dimensional digital lines are studied in §3. Section 4 discusses digital hyperplane segments in  $E^k$  while §5 applies the results to three-dimensional Euclidean space. The results for lines and hyperplanes are generalized in §6 by presenting the digitization scheme and the recognition technique for digital flats (subspaces of arbitrary dimension). Finally, §7 proposes digitization schemes for arcs and surfaces.

## 2. Digital line segments

A straight line in  $E^k$  is a one-dimensional space in  $E^k$ . Let a line  $L$  in  $E^k$  be given by  $(x_1 - a_1)/l_1 = (x_2 - a_2)/l_2 = \dots = (x_k - a_k)/l_k$ , or, equivalently, by  $x_i = a_i + t * l_i, 1 \leq i \leq k$ .

Assume without loss of generality that  $l_1 \geq l_2 \geq \dots \geq l_k \geq 0$  (otherwise the line can be transformed by mirror symmetries and permuting axes to this case). We now define digital images of straight lines and line segments as follows.

**Digitization scheme.** Let the crossing point of the line (segment)  $L$  and the hyperplane  $x_1 = j$  be  $P_j = (j, p_2, p_3, \dots, p_k)$ . Then we say that the digital point  $P'_j = (j, r_2, r_3, \dots, r_k)$ , where  $p_m - 1/2 < r_m \leq p_m + 1/2$  ( $m = 2, \dots, k$ ) is the nearest digital point to  $P_j$ , and that  $P'_j$  is the digital image of  $P_j$ . The set consisting of all points  $P'_j$  is said to be the digital image of the line (segment)  $L$ , and we denote it by  $L'$ .

Our digitization scheme differs from that of others. The major difference is expressed as Corollary 1 below; the details will be discussed in the next section.

**COROLLARY 1.** *Each hyperplane  $x_1 = j$  contains exactly one digital point of the line (segment)  $L$ .*

Note that in case of ties, say  $l_1 = l_2$ , the exchange of axes  $x_1$  and  $x_2$

does not affect the image of the line (segment).

A (finite) subset  $Q$  of  $D^k$  is said to be a *digital line (segment)* if there exists a line (segment) in  $E^k$  such that  $Q = L'$ .

**THEOREM 1.** *The digital image of a line segment is a digital arc.*

**PROOF.** From Corollary 1, the unique digital point of the image of a line segment that belongs to the hyperplane  $x_1 = j$  may have at most two neighbors: those points from the image that belong to the hyperplanes  $x_1 = j - 1$  and  $x_1 = j + 1$ . These points are indeed neighbors since from  $l_1 \geq l_2 \geq \dots \geq l_k \geq 0$  it follows that  $1 = \Delta x_1 \geq \Delta x_2 \geq \dots \geq \Delta x_k$ , where  $\Delta x_i$  is the differential along the  $x_i$  axis between two neighboring intersections of the line segment with the hyperplanes  $x_1 = j$ .

Let  $S$  be a finite digital set in  $D^k$ . For each  $i = 1, 2, \dots, k$  we determine the points of  $S$  with the smallest and with the greatest  $x_i$ -coordinates and determine the difference between them. Let  $x_1$  be the axis for which this difference  $x_{1\max} - x_{1\min}$  is the greatest (in case of ties any axis can be taken). We call  $x_1$  the main axis for  $S$ .

**THEOREM 2.** *A finite digital set  $S$  is a digital line segment in  $D^k$  if and only if the projections of  $S$  onto  $k - 1$  coordinate two-dimensional planes  $x_1 x_i$  for  $i = 2, 3, \dots, k$  (where  $x_1$  is the main axis for  $S$ ) are all digital line segments in these planes.*

**PROOF.** Denote by  $S_i$  the projection of  $S$  onto the two-dimensional coordinate plane  $x_1 x_i$ ,  $i = 2, 3, \dots, k$ .

*Necessity.* Suppose  $S$  is a digital line segment in  $D^k$ . Then there exists a line segment  $L$  in  $E^k$  such that  $L' = S$ . According to the definition, each hyperplane  $x_1 = j$  ( $x_{1\max} \leq j \leq x_{1\min}$ ) contains exactly one point  $P'_j = (j, r_2, r_3, \dots, r_k)$  from  $S$ , which is the digital image of the point of intersection  $P_j = (j, p_2, p_3, \dots, p_k)$  of  $L$  with the hyperplane  $x_1 = j$  (thus  $p_m - 1/2 < r_m \leq p_m + 1/2$ , for each  $m$ ,  $2 \leq m \leq k$ , is satisfied). Let  $L_i$  be the projection of  $L$  onto the coordinate plane  $x_1 x_i$ . The intersection of  $L_i$  with the straight line  $x_1 = j$  in the plane  $x_1 x_i$  is the projection of the point  $P_j$  and is given by  $P_{j,i} = (j, 0, \dots, 0, p_i, 0, \dots, 0)$ . The digital image of the point  $P_{j,i}$  in plane  $x_1 x_i$  is point  $P'_{i,j} = (j, 0, \dots, 0, r'_i, 0, \dots, 0)$  where  $p_i - 1/2 < r'_i \leq p_i + 1/2$ . Obviously  $r'_i = r_i$ . Thus the digital image of  $L_i$  in the plane  $x_1 x_i$  is exactly  $S_i$ , i.e.,  $S_i$  (the projection of  $S$ ) is a digital line segment.

*Sufficiency.* Suppose that the projections  $S_2, S_3, \dots, S_k$  of  $S$  are all digital line segments. For each  $i$ ,  $2 \leq i \leq k$ , denote by  $L_i$  the line segment in the plane  $x_1 x_i$  such that  $S_i$  is the digital image of  $L_i$  ( $L'_i = S_i$ ). The intersection of  $L_i$  with the straight line  $x_1 = j$  in plane  $x_1 x_i$  is the unique point  $P_{j,i} = (j, 0, \dots, 0, p_i, 0, \dots, 0)$ . According to the defini-

tion, the digital image of point  $P_{j,i}$  in plane  $x_1x_i$  is the point  $P'_{j,i} = (j, 0, \dots, 0, r_j, 0, \dots, 0)$  of  $S_i$ , where  $p_i - 1/2 < r_i \leq p_i + 1/2$ . The line segments  $L_i$  uniquely determine a line segment  $L$  in  $E^k$  whose projection onto the coordinate plane  $x_1x_i$  is  $L_i$  for each  $i$ ,  $2 \leq i \leq k$ . Consider the points  $P'_j = (j, r_2, r_3, \dots, r_k)$  from  $D^k$ , and  $P_j = (j, p_2, p_3, \dots, p_k)$  from  $E^k$ . Clearly, the points  $P'_{j,i}(P_{j,i})$  are projections of the point  $P'_j(P_j)$  onto coordinate planes  $x_1x_i$ ; if only one point exists that gives all projections  $P'_{j,i}$  then it is the point  $P'_j$ . Does  $P'_j$  belong to  $S$ ? Suppose that it does not. Then there exist at least two other points from  $S$  that cover points  $P'_{j,i}$  by projections. These two points must differ in at least one coordinate, say coordinate  $i$ . But then these two points determine by their projections two different points on the plane  $x_1x_i$ . This contradicts Corollary 1, since  $S_i$  is a digital line segment. Therefore  $P'_j$  belongs to  $S$ . Is  $P'_j$  the digital image of  $P_j$ ? Yes, since  $p_m - 1/2 < r_m \leq p_m + 1/2$ , for each  $m$ ,  $2 \leq m \leq k$ . Therefore  $S$  is the digital image of  $L$ . In other words,  $S$  is a digital line segment. This completes the proof of Theorem 2.

Theorem 2 provides an optimal algorithm for the *recognition of digital line segments* problem: given a set  $S$  of digital points in  $D^k$ , determine whether or not  $S$  is a digital line segment. The algorithm goes as follows.

*Algorithm digital line segment (S)*

input: the set  $S$  of  $n$  points  $p_t = (p_{t,1}, p_{t,2}, \dots, p_{t,k})$ ,  $1 \leq t \leq n$ .

output: true/false if  $S$  is/is not a digital line segment.

Step 1. For each  $i = 1, 2, \dots, k$  determine the points from  $S$  with the smallest and with the greatest  $x_i$ -coordinates and determine the difference between them. Let  $x_1$  be the axis for which this difference  $x_{1\max} - x_{1\min}$  is the greatest. Denote by  $S_i$  the projection of  $S$  onto the two-dimensional coordinate plane  $x_1x_i$ ,  $i = 2, 3, \dots, k$ .  $S_i$  contains points  $p_{t,i} = (p_{t,1}, 0, \dots, 0, p_{t,i}, 0, \dots, 0)$ .

Step 2. Determine for each  $i$ ,  $2 \leq i \leq k$ , whether  $S_i$  is a digital line segment in the two dimensional plane  $x_1x_i$ . If any test fails then return false. Otherwise, i.e., if all  $S_i$  are digital line segments, return true.

Step 1 can be clearly performed in  $O(kn)$  time. Given a set of  $n$  points in the plane, in [KR1] an  $O(n)$  algorithm is given that determines whether or not the set is a digital line segment. This test is applied  $k - 1$  times, leading to  $O(kn)$  time for Step 2. Thus, the following theorem is valid.

**THEOREM 3.** *Given a set  $S$  of  $n$  digital points in  $D^k$ , the algorithm DIGITAL\_LINE\_SEGMENT determines whether or not  $S$  is a digital line segment and its time complexity is  $O(kn)$ , which is optimal.*

The algorithm is optimal since the input consists of  $kn$  data, and no algorithm can process the data in faster than  $O(kn)$  time.

### 3. Problems with former digitization schemes for line segments

Digital line segments in 3-D space are studied by Kim in [K1]. A generalization for  $k - D$  space was given by Klette in [K]. The results presented in this paper differ from those presented in [K1, K]. The major difference is in the digitization scheme. [K1, K] define the digital image of a line segment (or, in general, an arc)  $f$  in the following way.

Whenever  $f$  crosses a coordinate hyperplane (plane in 3-D, line in 2-D, respectively), the nearest digital point to the crossing becomes a point of the digital image of  $f$ .

Using this definition, in [K1, Theorem 2 and Corollary 3], and in [K, Theorem 15] it is proved that the digital image of a line segment is a digital arc. We show that this is not correct for an arbitrary line segment. Consider Figures 1 and 2. The line segment  $f$  crosses the coordinate planes  $x = 0$ ,  $z = 0$ ,  $x = 1$ ,  $y = 1$ , and  $x = 1$  at points  $T(0, 1/3, -1/3)$ ,  $Q(1/3, 2/3, 0)$ ,  $P(2/3, 1, 1/3)$ ,  $R(1, 4/3, 2/3)$ , respectively. Their images are digital points  $T'(0, 0, 0)$ ,  $Q'(0, 1, 0)$ ,  $P'(1, 1, 0)$ , and  $R'(1, 1, 1)$ , respectively. However, the point  $Q'$  has three neighbors ( $T'$ ,  $R'$  and  $P'$ ) which contradicts the definition of a digital arc (other points also have three neighbors each).

Therefore a digital line segment is not necessarily a digital arc under the digitization scheme introduced in [K1, K]. Also, the projections of digital line segments in 3-D onto coordinate planes are not necessarily digital line segments in 2-D (for instance, there is no digital line segment in 2-D containing points  $T'$ ,  $P'$  and  $R'$ ). The proofs given by Kim [K1] and Klette [K] rely on Corollary 1, for which they gave incorrect intuitive proofs.

The digitization scheme suggested in this paper yields fewer digital points in the image of a straight line than the definition given in [K1]. For example, if  $x$  is chosen for the main axis (in this case the choice is arbitrary, since  $l_1 = l_2 = l_3 = 1/3$ ; obviously any of the three coefficients can be slightly

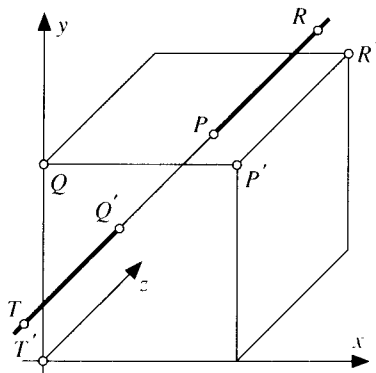


FIGURE 1

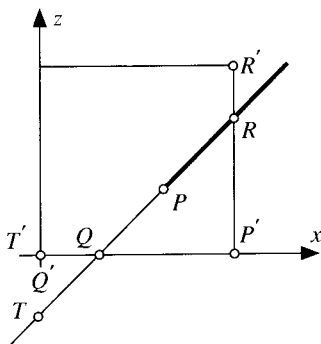


FIGURE 2

increased without affecting the image) the image of line segment  $f$  from Figures 1 and 2 will contain digital points  $T'$  and  $R'$  (the intersections of  $f$  with plane  $x = 0$ ) but not points  $Q'$  and  $R'$ .

It is interesting to note that, despite the inconsistency which we have explained, the digital line segment recognition algorithms proposed in [K1, K] turn out to be equivalent to our algorithms. However, their recognition algorithms fail to work properly under their digitization schemes.

#### 4. Digital hyperplanes

A hyperplane  $p$  in  $E^k$  is a  $(k - 1)$ -dimensional space in  $E^k$  and is given by the equation  $a_1x_1 + a_2x_2 + \dots + a_kx_k + a_{k+1} = 0$ . Without loss of generality we assume  $|a_i| \leq |a_k|$ , for each  $i$ ,  $1 \leq i \leq k - 1$ , and  $a_k > 0$  (otherwise the variable names can be changed); such an axis  $x_k$  is called the major axis for  $p$ . A hyperplane segment is a subset of a hyperplane that is bounded (say, by a hypercube). We define the digital image of a hyperplane (segment)  $p$  as follows.

**Digitization scheme.** Let the crossing point of  $p$  and a coordinate digital straight line parallel to the major axis  $x_k$  of  $p$  (the equation of the line is given by  $x_1 = r_1, x_2 = r_2, \dots, x_{k-1} = r_{k-1}$ , where  $r_1, r_2, \dots, r_{k-1}$  are integers) be the point  $P = (r_1, r_2, \dots, r_{k-1}, p_k)$ . Then we say that the digital point  $P' = (r_1, r_2, \dots, r_{k-1}, r_k)$ , where  $p_k - 1/2 < r_k \leq p_k + 1/2$ , is the nearest digital point to  $P$ , and that  $P'$  is the digital image of  $P$ . The set consisting of all points  $P'$  corresponding to the above crossings of  $p$  is said to be the digital image of  $p$ , and we denote it by  $p'$ .

The proposed scheme is a new scheme, and the relation to known schemes is discussed in the next section.

Note that the above definition may give meaningless digital images for some hyperplane segments (for instance, the image of a nonempty set can be empty). We will not discuss these anomalies since in applications hyperplane segments have some regularities (for examples, convexity) that are consistent with the digitization scheme.

A subset  $q$  of  $D^k$  is said to be a *digital hyperplane (segment)* if there is a hyperplane (segment)  $p$  such that  $q = p'$ . The new scheme provides an efficient solution to the digital hyperplane segment recognition problem: given a set  $q$  of digital points in  $D^k$ , determine whether or not  $q$  is a digital hyperplane segment.

The above digitization scheme will be updated to facilitate the description of the recognition technique. Given a hyperplane (segment)  $p$ , let  $p''$  be the hyperplane (segment) obtained by translating  $p$  along the major axis  $x_k$  of  $p$  by  $1/2$ . For  $p$  defined as above,  $p''$  has the equation  $a_1x_1 + a_2x_2 + \dots + a_k(x_k - 1/2) + a_{k+1} = 0$ . Let the crossing point of  $p''$  and a coordinate digital straight line parallel to the  $x_k$  axis be the point  $P'' = (r_1, r_2, \dots, r_{k-1}, p''_k)$ . Under the above assumptions and notations, it is clear that  $p''_k = p_k + 1/2$ , and  $p''_k - 1 < r_k \leq p''_k$ . Thus, the digital point  $P' = (r_1, r_2, \dots, r_{k-1}, r_k)$ , is the first digital point that is below the point  $P''$  on the line parallel to the  $x_k$  axis. By taking into account all the intersections of  $p$  with the digital  $x_k$  axes, one can obtain a new digitization scheme for hyperplane (segments).

**Another digitization scheme.** The digital image of a hyperplane (segment)  $p$  is the set of all digital points that are first points below (along the major axis  $x_k$  of  $p$ ) the intersections of the hyperplane  $p''$  (obtained by translating  $p$  by  $1/2$  along  $x_k$  axis) with coordinate digital lines that are parallel to  $x_k$  axes.

Although this scheme is new, it is equivalent to "marking all points at one side of a digital hyperplane and then taking the boundary", up to a translation; this is how such planes are sometimes obtained in practice.

For each point  $P' = (r_1, r_2, \dots, r_{k-1}, r_k)$  from the image  $p'$  of hyperplane (segment)  $p$ , consider the point  $P''' = (r_1, r_2, \dots, r_{k-1}, r_k + 1)$ . Since  $p''_k - 1 < r_k \leq p''_k < r_k + 1$ , the point  $P''$  separates points  $P'$  and  $P'''$ . The set of all such points  $P'''$  is denoted  $p'''$ . It is easy to verify that  $p'''$  is obtained from  $p'$  by translating it by 1 along the  $x_k$  axis. Using this, one can obtain the following lemma.

**LEMMA 1.** *The hyperplane  $p''$  separates the digital image  $p'$  of  $p$  from  $p'''$  so that all digital points from  $p'$  are on or below  $p''$ , while all points from  $p'''$  are above  $p''$ .*

The latter digitization scheme and Lemma 1 lead to criteria for recognizing digital hyperplane (segments), expressed in the following theorem.

**THEOREM 4.** *A digital set  $S$  from  $D^k$  is a digital hyperplane segment if and only if there exists a hyperplane  $p''$  that separates  $S$  from the set  $S'''$  obtained from  $S$  by translating it along the  $x_k$  axis by 1 (where  $x_k$  is the major axis for  $p''$ ).*

**PROOF.** Suppose  $S$  is a digital hyperplane. Then there is a hyperplane  $p$  such that  $S$  is the digital image of  $p$ , and the existence of a separating plane  $p''$  is shown above. Suppose now that  $S'''$  is constructed from  $S$  by translating as described, and that  $S$  and  $S'''$  can be separated by a hyperplane  $p''$ . Given a coordinate digital straight line parallel to the  $x_k$  axis,  $p''$  intersects it in a point  $(r_1, r_2, \dots, r_{k-1}, p_k'')$  such that  $r_k \leq p_k'' < r_k + 1$ , where  $r_k$  belongs to  $S$  and  $r_k + 1$  to  $S'''$ . Then  $r_k - 1/2 \leq p_k'' - 1/2 < r_k + 1/2$ , i.e.,  $r_k$  is the digital image of  $p_k'' - 1/2 = p_k$ . Therefore  $S$  is the digital image of a plane  $p$  obtained from  $p''$  by translating by  $-1/2$  along the  $x_k$  axis.

Theorem 4 is a basis for two different hyperplane segment recognition algorithms. A high level description of both of them can be given in the same way.

#### Algorithm digital hyperplane segments (S)

input: the set  $S$  of  $n$  points  $p_t = (p_{t,1}, p_{t,2}, \dots, p_{t,k})$ ,  $1 \leq t \leq n$ .

output: true/false if  $S$  is/is not a digital hyperplane segment.

Step 1. Find the axes that are candidates for the major axis of  $S$ ; for each of these axes perform the steps below until a successful axis is found, or all axes are checked. Let  $x_k$  be a candidate axis (otherwise exchange axes names).

Step 2. Translate  $S$  by 1 along the  $x_k$  axis to obtain the set  $S'''$ .

Step 3. Determine whether or not  $S$  and  $S'''$  are separable by a hyperplane. If there exists such a hyperplane  $p''$  then  $S$  is the digital image of the hyperplane  $p$  obtained from  $p''$  by translating by  $-1/2$  along the  $x_k$  axis, and return true. Otherwise return false.

The details of each step follow.

For hyperplanes in general position, the major axis is unique, and can be found (step 1) by several techniques, the efficiency of which depends on the point distribution. The simplest technique would be to choose randomly  $k$  points from  $S$ , find the hyperplane passing through these points, and determine the largest slope parallel to coordinate axes (slopes correspond to coefficients  $a_i$ ,  $1 \leq i \leq k$ ). If the answer is unique then it is correct. Otherwise points that are not on the hyperplane must be involved. For hyperplane segments that have some regularities in their distribution (for example, convexity or connectivity) the desired coordinate line is any that has at most one point from  $S$  at any digital coordinate line parallel to that axis (the projection on the coordinate hyperplane that is orthogonal to the axis is one-to-one). The checking can be done by sorting with respect to the first  $k - 1$  coordinates (this would require  $O(kn \log n)$  time) or faster, in  $O(kn)$  time, since the uniqueness problem for bounded integers can be solved in linear time. The time to find the desired axis using this method is therefore  $O(k^2 n)$ . We also suggest a third method, find the axis by comparing the slopes in pairs: the comparison of slopes with respect to the axes  $x_i$  and  $x_j$  can be carried

out by intersecting  $S$  with the coordinate plane  $x_i x_j$  and determining the larger slope. By  $k-1$  such comparisons the desired slope will be found. This also can take  $O(k^2 n)$  time in the worst case. This method will be justified below by Lemma 3. The elimination of axes other than major ones can be also done by applying Lemma 2.

**LEMMA 2.** Let  $T' = (t'_1, \dots, t'_k)$  and  $T'' = (t''_1, \dots, t''_k)$  be two points from a digital hyperplane  $S$ , and let  $\Delta t_i = |t'_i - t''_i| + 1$ ,  $1 \leq i \leq k$ . Then for major axis  $x_k$  the following inequality is satisfied:  $\Delta t_k \leq \Delta t_1 + \dots + \Delta t_{k-1} + 2$ .

**PROOF.** Let  $p$  be a hyperplane such that  $p' = S$ , and let the equation of  $p$  be  $a_1 x_1 + \dots + a_k x_k + a_{k+1} = 0$ . If  $Q' = (q'_1, \dots, q'_k)$  and  $Q'' = (q''_1, \dots, q''_k)$  are two points of  $p$  then one can easily obtain  $a_1 \Delta q_1 + \dots + a_k \Delta q_k = 0$ , where  $\Delta q_i = q'_i - q''_i$ . Since  $|a_i/a_k| \leq 1$  ( $1 \leq i \leq k-1$ ), from  $\Delta q_k = -a_1/a_k q'_k - \dots - a_{k-1}/a_k q'_{k-1}$  we obtain  $|\Delta q_k| \leq |\Delta q_1| + |\Delta q_2| + \dots + |\Delta q_{k-1}|$ . For each  $i$ , the error in digitizing that resulted in  $t'_i$  or  $t''_i$  is at most  $1/2$ . Thus  $\Delta t_i - 2 = |t'_i - t''_i| - 1 \leq |\Delta q_i| \leq |t'_i - t''_i| + 1 = \Delta t_i$ ,  $1 \leq i \leq k$ . This gives  $\Delta t_k - 2 \leq \Delta t_1 + \dots + \Delta t_{k-1}$ , as required.

**LEMMA 3.** Let  $T' = (t'_1, \dots, t'_k)$  and  $T'' = (t''_1, \dots, t''_k)$  be two points from a digital hyperplane  $S$  that belong to a hyperplane that is parallel to the coordinate hyperplane  $x_i x_j$  (thus  $t'_m = t''_m$  for  $m \neq i$ ,  $m \neq j$ ), and let  $\Delta t_i = |t'_i - t''_i| + 1$ ,  $\Delta t_j = |t'_j - t''_j| + 1$ . Also, let  $p' = S$  where  $p$  has equation  $a_1 x_1 + \dots + a_k x_k + a_{k+1} = 0$ . Under these assumptions, if  $|a_j| \leq |a_i|$  then  $\Delta t_i \leq \Delta t_j + 2$ .

**PROOF.** Similar to the proof of Lemma 2.

In step 2, the translation of  $S$  can be done easily in  $O(kn)$  time.

Step 3 can be solved in two different ways. They are described as two different procedures.

Procedure *separability\_ch*( $S, S'''$ )

Step 3.1. Find the convex hulls  $\text{CH}(S)$  and  $\text{CH}(S''')$  of  $S$  and  $S'''$ , respectively.

Step 3.2. Determine whether  $\text{CH}(S)$  and  $\text{CH}(S''')$  are separable by a hyperplane by checking whether or not they intersect.  $S$  and  $S'''$  are separable if and only if  $\text{CH}(S)$  and  $\text{CH}(S''')$  do not intersect.

The convex hull of a set of  $n$  points in  $E^k$  can be constructed in  $O(n \log n + n^{(k+1)/2})$  time and  $O(n^{k/2})$  storage [Sei], where integer division is applied in exponent (e.g.,  $5/2 = 2$ ). For  $k = 3$  there exists an  $O(n \log n)$  convex hull construction algorithm [PH].

Detecting the intersection of two convex polyhedra in  $E^k$  can be done in  $O(2^k n^{2^{k-2}k} \log n)$  time [GW]. For  $k = 3$  there is an  $O(n \log n)$  algorithm [DM].

Thus the running time of the procedure *separability\_ch* is determined by the time for detecting intersection of two convex polyhedra in  $E^k$ ; this is also the running time for the entire hyperplane recognition algorithm that uses steps 3.1 and 3.2. We therefore have the following theorem.

**THEOREM 5.** *Given a set  $S$  of  $n$  digital points in  $D^k$ , algorithm DIGITAL\_HYPERPLANE\_SEGMENT based on the procedure SEPARABILITY\_CH determines whether or not  $S$  is a digital hyperplane segment and its time complexity is  $O(C(k, n) + I(k, n))$ , where  $C(k, n)$  and  $I(k, n)$  are the complexities of the convex hull construction and detecting intersection of two polyhedra, respectively.*

Another separability algorithm uses a linear programming technique [D, M, M1].

Procedure *separability\_lp* ( $S, S'''$ )

Step 3'. Apply linear programming in an arbitrary dimension to detect separability.

Suppose that the desired hyperplane  $p''$  has equation  $a_1x_1 + \dots + a_kx_k + a_{k+1} = 0$ , where  $a_1, \dots, a_{k+1}$  are to be determined. The requirement that points  $(r_1, \dots, r_{k-1}, r_k)$  from  $S$  are below hyperplane  $p''$  is equivalent to (1)

$$a_1r_1 + a_2r_2 + \dots + a_kr_k + a_{k+1} \leq 0 \quad \text{for all } n \text{ points } (r_1, r_2, \dots, r_k) \text{ from } S.$$

On the other hand, points  $(r_1, r_2, \dots, r_{k-1}, r_k + 1)$  are above  $p''$ , i.e.,  $a_1r_1 + \dots + a_{k-1}r_{k-1} + a_k(r_k + 1) + a_{k+1} > 0$  for all points  $(r_1, r_2, \dots, r_{k-1}, r_k + 1)$  from  $S'''$ .

The latter condition can be written in its weaker form:

$$(2) \quad -a_1r_1 - \dots - a_{k-1}r_{k-1} - a_k(r_k + 1) - a_{k+1} \leq 0$$

for all  $n$  points  $(r_1, r_2, \dots, r_{k-1}, r_k + 1)$  from  $S'''$ .

The  $2n$  conditions (1) and (2) can be considered as a special form of the  $k$ -variable linear programming problem, which is generally stated in the following form: given constants  $b_1, \dots, b_k$  and variables  $a_1, \dots, a_k, a_{k+1}$ , maximize  $b_1a_1 + b_2a_2 + \dots + b_ka_k$  subject to (1) and (2).

In our case we, in fact, do not have to optimize any function; we merely need to find out whether there is any feasible solution, this is a part of solving the linear programming problem. Meggido [M1] presented an  $O(2^{2^k} n)$  solution to the problem, i.e., linear time solution but with an extremely large doubly exponential constant. This time complexity then applies to the digital hyperplane recognition algorithm that uses Step 3'.

To assure the correctness of the algorithm, an additional step is required, because of our weaker conditions (2). The weaker condition can be overcome

by merely decreasing the values  $r_k$  for a small real number  $e$  ( $r_k$  replaced by  $r_k - e$ ). This completes the proof of the following theorem.

**THEOREM 6.** *Given a set  $S$  of  $n$  digital points in  $D^k$ , algorithm DIGITAL\_HYPERPLANE\_SEGMENT which uses the procedure SEPARABILITY\_LP determines whether or not  $S$  is a digital hyperplane segment; its time complexity is  $O(LP(k, n))$ , where  $LP(k, n)$  is the complexity of the  $k$ -variable linear programming problem with  $n$  constraints.*

### 5. Two-dimensional digital planes in 3-D space

[K2] and [K] studied 2-D digital planes in 3-D and  $k$ -D spaces, respectively. They apply the following digitization scheme.

Whenever a plane (segment)  $p$  intersects a coordinate line, the nearest point to the intersection becomes a point of the digital image of  $p$ .

It can be shown that this digitization scheme is equivalent to the one introduced in this paper (for 3-D space it is also proved in [K2, Lemma 1]).

In [K2] several properties of digital planes are studied, and a digital plane recognition algorithm is suggested. Given a set  $S$  of  $n$  points in  $E^3$ , this algorithm determines in  $O(n^2)$  time whether or not  $S$  is a convex digital plane segment. A digital plane is convex if its projections onto coordinate planes are convex [K2]. The convexity can be easily tested in  $O(n)$  time [K2].

In §4 we presented two faster solutions to the digital plane segment recognition problem. They are obtained when dimension  $k$  is set equal to 3.

The first algorithm is based on the convex hull construction [PH] and polyhedra detection intersection algorithm [DM]. Both can be solved in  $O(n \log n)$  time, which is the running time of the first improvement.

The second improvement is based on the three-variable linear programming problem. The three-variable linear programming problem is solved in [M, D] in linear time. Therefore the three-dimensional digital convex plane segment recognition algorithm can be solved in optimal  $O(n)$  time. The algorithm is asymptotically faster than the first one, but has a larger constant and is expected to run slower than the first one.

Let the algorithm `Digital_convex_plane_segment(S)` be defined as the algorithm `Digital_hyperplane_segment(S)` when  $k$  is fixed to 3, and has, in addition, the simple convexity test. Then we have the following result.

**THEOREM 7.** *Given a set  $S$  of  $n$  digital points in  $D^3$ , the algorithm DIGITAL\_CONVEX\_PLANE\_SEGMENT determines whether or not  $S$  is a convex digital plane segment; its time complexity is  $O(n \log n)$  or  $O(n)$ , depending on whether the procedure SEPARABILITY\_CH or the procedure SEPARABILITY\_LP is used, respectively.*

Note that our digital hyperplane algorithms are the first such algorithms for arbitrary dimensions.

**6. Digital  $m$ -flats in  $k$ -dimensional space**

An  $m$ -flat in  $E^k$  is an  $m$ -dimensional space in  $E^k$ . It can be represented as the set of all linear combinations of  $m + 1$  independent points from  $E^k$ . A 2-flat is a straight line in  $E^k$ , while a  $(k - 1)$ -flat is a hyperplane. In  $E^3$ , 1-flats and 2-flats are straight lines and (hyper)planes, respectively. The first  $m$ -flat which is not a straight line or a hyperplane is a 2-flat in  $E^4$ .

In this section we generalize the digitization schemes for lines and hyperplanes so as to apply to arbitrary  $m$ -flats in  $E^k$ .

Consider the angle  $\beta_i$  (or its slope  $tg\beta_i$ ) that a coordinate axis  $x_i$  forms with a given  $m$ -flat. This angle can be determined from the length  $s_i$  of the projection of a unit vector with direction along the  $x_i$  coordinate axis onto the given  $m$ -flat. In fact,  $\cos\beta_i = s_i$ . Thus, the angle decreases with increasing length.

Let  $x_1x_2 \cdots x_m$  denote the coordinate  $m$ -flat that contains coordinate axes  $x_1, x_2, \dots, x_m$ . The coordinate  $m$ -flat is given by equations  $x_{m+1} = 0, x_{m+2} = 0, \dots, x_k = 0$ .

Let  $L$  be an  $m$ -flat in  $E^k$ . Suppose the slopes of  $L$  with coordinate axes are ordered by  $\beta_1 \leq \beta_2 \leq \dots \leq \beta_k$ . Thus, the  $m$  smallest slopes are formed with the axes  $x_1, x_2, \dots, x_m$ . Consider the  $(k - m)$ -flat that is parallel to coordinate  $(k - m)$ -flat  $x_1x_2 \cdots x_{k-m}$  and passes through digital point  $(j_1, j_2, \dots, j_m)$ . We apply the following digitization scheme.

**Digitization scheme.** Whenever  $L$  crosses a  $(k - m)$ -flat that is parallel to the coordinate  $(k - m)$ -flat  $x_1x_2 \cdots x_{k-m}$  and passes through the digital point  $(j_1, \dots, j_m)$ , the nearest digital point to the intersection is the image of the crossing point. The image of the  $m$ -flat is the set of all images of all such intersection points. A set  $S$  of digital points in  $E^k$  is a digital  $m$ -flat if there exists an  $m$ -flat for which  $S$  is its image.

It is easy to verify that the above digitization schemes for lines and hyperplanes are special cases of this digitization scheme.

Now we suggest a criteria for recognizing digital  $m$ -flats in  $E^k$ .

**THEOREM 8.** *Let  $S$  be a set of digital points in  $D^k$ .  $S$  is a digital  $m$ -flat if and only if all  $k - m$  of its projections onto the coordinate  $(m + 1)$ -flats  $x_1x_2 \cdots x_mx_{m+1}, x_1x_2 \cdots x_mx_{m+2}, \dots, x_1x_2 \cdots x_mx_k$  are digital  $m$ -flats in these  $(m + 1)$ -flats (i.e., hyperplanes in these  $(m + 1)$ -flats).*

The proof is omitted since it is a tedious generalization of the proof of Theorem 2.

This provides an algorithm for recognizing digital  $m$ -flats in  $D^k$ . The algorithm basically consists of  $k - m$  calls to the digital hypercube recognition problem in  $D^{m+1}$ , which was studied in §4. The time complexity of the digital  $m$ -flat recognition algorithm in  $D^k$  is therefore  $O((k - m)T(m))$ ,

where  $T(m)$  is the time complexity of a digital hypercube recognition method in  $E^{m+1}$  (discussed in previous sections).

### 7. Digital arcs and surfaces

The digitization scheme for arcs suggested in [K1, K] suffers from the same problems which were present with the digitization scheme for line segments. We here propose two different schemes.

**Arc digitization scheme 1.** Approximate a given arc piecewise by line segments (using any known analytic method) and apply the digitization scheme for line segments for each piece separately.

**Arc digitization scheme 2.** Whenever a given arc crosses a coordinate hyperplane, the nearest point to the intersection becomes a point of digital image of the arc if the axis normal to the coordinate hyperplane is the main axis (defined in §2) for the tangent line of the arc at the point of intersection.

Note that under both schemes the digital image of an arc is not necessarily a digital arc; some conditions apply, and this subject falls beyond the scope of the paper.

These schemes can be generalized for surfaces of any dimensions (“hyper-surfaces”). Either “triangulation” of the surface is done to replace surfaces by hyperplanes, or the intersection points of the surface with digital lines parallel to the corresponding axis are found (the axis varies with the slope of the surface).

### Conclusion

In this paper we presented new digitization schemes and recognition algorithms for lines, planes and spaces of arbitrary dimensions that apply to the new scheme. While the 2-D case has been extensively studied in the literature, the known results for three- and higher-dimensional objects were not consistent or efficient.

This paper concentrated on defining basic notions and properties that would be sufficient to propose a digitization scheme and an algorithm for recognizing digital objects in arbitrary dimensional space. There are some properties (chord property, chordal triangle property, volume property, convexity, etc.) that were studied for objects in two and three dimensions and are not covered in this paper. Extending these properties to higher dimensions can be a topic for further research.

All known digitization schemes (including those suggested in this paper) suffer from the following problem: the digital image of intersection of two planes is different from the intersection of digital images of these planes (even the digital image of an intersection line may not belong to any of the plane images). This problem is illustrated in Figure 3. The same figure gives a contradiction for arbitrary dimensions (starting from 2-D space). For

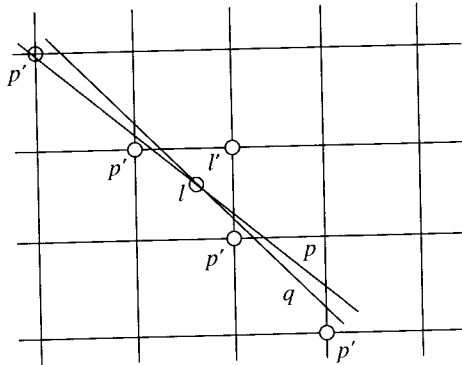


FIGURE 3

instance, let  $p$  and  $q$  be orthogonal projections of two planes and  $l$  be the orthogonal projection of their intersection. Points denoted by  $p'$  belong to images of both planes but not to the image of their intersection  $l$ . On the other hand, the image of line  $l$ , denoted  $l'$ , does not belong to any of plane images.

Therefore, the digital geometry of flats in arbitrary dimensions does not satisfy the usual properties of real geometry under all known digitization schemes. Designing a scheme that will reflect these properties to a greater extent is an open problem for further study.

#### Acknowledgment

The authors appreciate several valuable comments made by the referee which improved the presentation of the paper.

#### REFERENCES

- [D] M. E. Dyer, *Linear time algorithms for two- and three-variable linear programs*, SIAM J. Comput. **13-1** (1984), 31-45.
- [DM] D. P. Dobkin and J. I. Munro, *Efficient uses of the past*, Proc. 21st Annual Sympos. on Found. of Comput. Sci., 1980, pp. 200-206.
- [GW] O. Gunther and E. Wong, *A dual approach to detecting polyhedral intersections in arbitrary dimensions*, Proc. Allerton Conference, University of Illinois, 1987, pp. 859-868.
- [K1] C. E. Kim, *Three-dimensional digital line segments*, IEEE Trans. PAMI **5** (1983), 231-234.
- [K2] —, *Three-dimensional digital planes*, IEEE Trans. PAMI **6** (1984), 639-645.
- [K3] —, *Digital convexity, straightness and convex polygons*, IEEE Trans. PAMI **4** (1982), 618-626.
- [KR1] C. E. Kim and A. Rosenfeld, *Digital straight lines and convexity of digital regions*, IEEE Trans. PAMI **4** (1982), 149-153.
- [KR2] —, *Convex digital solids*, IEEE Trans. PAMI **4** (1982), 612-618.
- [Ki] D. G. Kirkpatrick, *Optimal search in planar subdivision*, SIAM J. Comput. **12** (1979), 18-27.
- [K] R. Klette, *The  $m$ -dimensional grid point space*, Comput. Vision, Graphics, and Image Processing **30** (1985), 1-12.
- [M] N. Megiddo, *Linear time algorithm for linear programming in  $R^3$  and related problems*, SIAM J. Comput. **12** (1983), 759-776.

- [M1] —, *Linear programming in linear time when the dimension is fixed*, J. Assoc. Comput. Mach. **31** (1984), 114–127.
- [MR] D. G. Morgenthaler and A. Rosenfeld, *Surfaces in three-dimensional digital images*, Inform. and Control **51** (1981), 227–247.
- [PH] F. P. Preparata and S. J. Hong, *Convex hulls of finite sets of points in two and three dimensions*, Comm. ACM **20** (1977), 87–93.
- [PS] F. P. Preparata and M. I. Shamos, *Computational Geometry, An Introduction*, Springer-Verlag, New York, 1985.
- [Sei] R. Seidel, *A convex hull algorithm optimal for point sets in even dimensions*, Tech. Rep. 81-14, Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, 1981.

COMPUTER SCIENCE DEPARTMENT, UNIVERSITY OF OTTAWA, OTTAWA, ONTARIO, CANADA  
K1N 9B4

INSTITUTE OF MATHEMATICS, UNIVERSITY OF NOVI SAD, 21000 NOVI SAD, YUGOSLAVIA