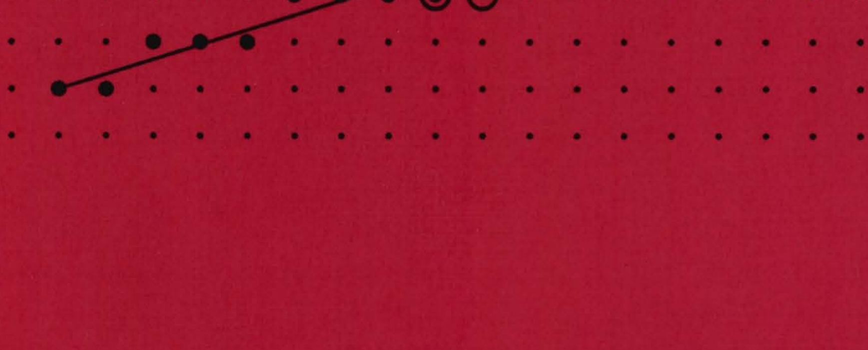
Digitisation functions in computer graphics

DOI:
10.6100/IR265934

Document status and date:
Published: 01/01/1987

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Digitisation Functions in Computer Graphics



Marloes van Lierop

Digitisation Functions
in
Computer Graphics

Digitisation Functions in Computer Graphics

PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR
AAN DE TECHNISCHE UNIVERSITEIT EINDHOVEN,
OP GEZAG VAN DE RECTOR MAGNIFICUS,
PROF. DR. F.N. HOOGE, VOOR EEN
COMMISSIE AANGEWEZEN DOOR HET COLLEGE VAN
DEKANEN IN HET OPENBAAR TE VERDEDIGEN OP
VRIJDAG 3 JULI 1987 TE 16.00 UUR

DOOR

MARIA LUCIA PETRONELLA VAN LIEROP

GEBOREN TE HELMOND

Druk: Dissertatiedrukkerij Wibro, Helmond.

Dit proefschrift is goedgekeurd
door de promotoren

Prof. dr. F.J. Peters

en

Prof. dr. M. Rem

Aan mijn ouders

These investigations were supported by the Netherlands Technology Foundation (STW).

Contents

0

Introduction

0.0 Motivation

Computer Graphics is the discipline concerned with the generation of images by means of computers. Nowadays, the most common type of devices on which these images are shown is a raster device, like raster monitors and (laser) printers. In this thesis we consider problems associated with the generation of images on raster monitors and printing devices. For an overview text on Computer Graphics we refer to [Foley & van Dam 1982] and [Newman & Sproull 1979].

The images displayed on raster monitors and printing devices are *digital* images: they consist of a finite number of discrete elements, called *pixels*, regularly arranged in a square grid. A pixel may be considered as an element of $\mathbf{Z}^2$, hence a digital image is a subset of $\mathbf{Z}^2$. Usually, with each pixel a colour or gray level value is associated.

Since the objects that are to be displayed are mostly defined in $\mathbf{R}^3$ or $\mathbf{R}^2$, the display process includes a digitisation mapping from $\mathbf{R}^3$ or $\mathbf{R}^2$ to $\mathbf{Z}^2$. In this thesis we concentrate on digitisation mappings from $\mathbf{R}^2$ to $\mathbf{Z}^2$.

Thus far, in Computer Graphics literature, digitisation is dealt with by presenting algorithms. The specification, if any, of these algorithms mostly deals with *closeness*: the resulting set of pixels should correspond to the original object in $\mathbf{R}^2$ as good as possible. However, a formal specification of the correspondence is seldom presented; the reader is assumed to have an intuitive idea what is meant. As an illustration we shall quote from three well-known books on Computer Graphics; in these books, digitisation algorithms are called *scan conversion algorithms*.

[Newman & Sproull 1979]

- On p.21, a number of criteria for computer-generated lines are discussed, namely

  Lines should appear straight ("... we must approximate the line by choosing addressable points close to it. If we choose well, the line will appear straight; if not, we shall produce crooked lines...").
  Lines should terminate accurately.
  Lines should have constant density.
  Line density should be independent of line length and angle.
  Lines should be drawn rapidly.

Note that the last criterion concerns the algorithm to generate lines, not the lines themselves.

The algorithms that are subsequently presented are compared to the first and last criteria only. A more explicit description of straightness of computer generated lines can be found in the discussion of one of these algorithms, called the symmetrical DDA (p.23):

> The symmetrical DDA generates accurate lines, since the displacement of a displayed dot from the true line is never greater than one-half a screen unit.

However, they fail to define the notion displacement.

- On p.215 the definition of *scan conversion* is given:

  > ...computing the pattern of dots that most closely matches a stored definition of the image.

  "Closely" is not worked out any further.
- On p.229 the definition of *solid-area scan conversion* is given:

  > The task of computing an area's mask from a geometrical description of the shape of the object...

  where the *mask of an area* is

  > ...a representation that defines which pixels lie within the solid area.
- On p.232 is given:

  > The scan conversion of a polygon involves finding all pixels that lie inside the polygon boundaries...

[Foley & van Dam 1982]

- On p.133 the definition of *scan conversion* is given:

  > The process of converting a line, point, and area representation to the pixel array of the image storage is called *scan conversion*.
- On criteria for line algorithms only the following is said (p.432):

  > The basic task of a scan-conversion algorithm for lines is to compute the coordinates of the pixels which lie near the line on a two-dimensional raster grid.

  A measure for this nearness is not explicitly presented.
- In the discussion of scan conversion algorithms for circles, the following sentence occurs (p.443):

  > On each step, the algorithm selects the point $P_i(x_i, y_i)$ which is closest to the true circle...

  and this appears to be the whole specification.
- The specification of a scan conversion algorithm for polygons is dismissed in the following words (p.457):

  > We must determine which pixels on the scan line are within the polygon, and set the corresponding pixels (...) to the appropriate values. By repeating this for each scan line which intersects the polygon, we scan-convert the entire polygon.

[Rogers 1985]

Here *rasterisation* ("the process of determining which pixels will provide the best approximation to the desired line...", p.29) and *scan conversion* ("the process of converting the rasterized picture stored in a frame buffer to the rigid display pattern of video...", p.17) are properly distinguished.

- For line drawing algorithms, the same criteria as in [Newman & Sproull 1979] are presented, and discussed likewise (p.29). Closeness is not explained.
- On solid area scan conversion the following is said (p.69):

  Scan conversion techniques attempt to determine, in scan line order, whether or not a point is inside a polygon or contour.

In each of these books, precise specifications of digitisation algorithms for lines are missing. Note that in all three books the scan conversion for polygons is considered to generate the set of all pixels within the polygon, that is, the digitisation of a polygon $V \subset \mathbf{R}^2$ is $V \cap \mathbf{Z}^2$.

Problems with digitisation algorithms are noticed by, for example, [Forrest 1985], [Franklin 1986], [Corthout & Jonkers 1986a], [Bresenham 1986], [Cook 1986], [Crow 1977]. We think that there are two principal causes for these problems:

- negligent use of floating point arithmetic, and
- the deficiency of a formal specification of the algorithm.

Problems due to floating point arithmetic may be avoided by using integer arithmetic only. A well known digitisation algorithm that is based on integer arithmetic is the Bresenham algorithm for line segments ([Bresenham 1965], also presented in, for instance [Foley & van Dam 1982]), to be discussed in Chapter 2.

A precise specification of what the algorithm is expected to do not only allows verification of the correctness of the algorithm (see [Dijkstra 1976], for example), but may also be of help in the design of the algorithm. A nice example hereof is presented by [van Overveld 1986].

For the specification of an algorithm one needs a formal framework, in which one can define requirements of digitisation mappings. This thesis aims at providing such a framework, based on the concept that any partial function from $\mathbf{R}^2$ into $\mathbf{Z}^2$ is a digitisation function. Several desirable properties of digitisation functions are formulated, thus leading to a classification. We concentrate on two properties, closeness and *convexity*. Convexity is a property only recently formulated ([Franklin 1986]). The importance of closeness is obvious. Convexity is desirable in applications where windowing is used, see for instance [Luby 1986]. Note that in our view on digitisation, closeness is not longer a necessary property of digitisation functions, though, in most applications, still a highly desirable one. For digitisation functions that are not close, a measure to express their quality with regard to closeness is introduced.

To show the implications of this new view on digitisation, we extensively discuss digitisation functions for line segments, shortly called line functions. We first give an axiomatic definition of line functions, and then distinguish several classes, each associated with a particular property. We shall show that closeness and convexity are difficult to combine.

In the field of Image Analysis, where digitisation is an important issue too, one is interested in the following question. Given a (close) digitisation function, what criteria characterise pixel sets as digitisations of that function. For line functions, examples of criteria can be found in [Freeman 1970], [Rosenfeld 1974], [Brons 1974], [Wu 1982], [Hung 1985], and [Dorst 1986].

In the Computer Graphics field, the reverse question is relevant: given certain properties of digitisations or digitisation functions (which may differ for various applications), what functions satisfy these properties.

0.1 Overview of this thesis

In Chapter 1, several desirable properties of digitisation functions are introduced, including closeness. The vicinity measure will be defined, which expresses the quality of a digitisation function with respect to closeness. Furthermore, some examples of digitisation functions will be presented.

In Chapter 2, digitisation functions for straight line segments are discussed. Several examples are presented, and classified with respect to the properties introduced in Chapter 1. Furthermore, it will be proven that for some combinations of properties no digitisation functions exist that satisfy all these properties.

Chapter 3 deals with a special class of digitisation functions for straight line elements, namely the class of recursive functions. Three examples are presented and classified.

In Chapter 4 the property convexity is treated in more detail. It will be shown that each permutation induces a convex line function on a limited domain.

In Chapter 5 the results are discussed, as well as implications for future research.

The following section contains some remarks on the notational conventions used in this thesis.

0.2 Notational conventions

- For denoting variables, we shall use the characters

i,j,k,l,m,n	for elements of $\mathbf{N}$
a,b,c	for elements of $\mathbf{Z}$
x,y,z	for elements of $\mathbf{R}$
p,q,r,s	for elements of $\mathbf{Z}^2$
v,w	for elements of $\mathbf{R}^2$
P,Q,R,S	for subsets of $\mathbf{Z}^2$
V,W	for subsets of $\mathbf{R}^2$

- Elements of $\mathbf{R}^2$ will be called *points*; a point will be denoted as an ordered pair of real numbers (x, y). For a point $v = (x, y)$, we define $v.x := x$ and $v.y := y$.
  Elements of $\mathbf{Z}^2$ will be called *pixels*. The pixel $(0,0)$ will be denoted as $\underline{0}$.
- We extend the use of the common arithmetic operators to points in a straightforward way:

$$v + w := (v.x + w.x, v.y + w.y),$$

$$\lfloor v \rfloor := (\lfloor v.x \rfloor, \lfloor v.y \rfloor),$$

  and so forth.
- If H is a set of characters, then H^* denotes the set of all strings, including the empty string ϵ, whose elements are contained in H. If σ is a string, and h a character, then $N_\sigma(h)$ denotes the number of occurrences of h in σ.
- The power set of a set V will be denoted as $\boldsymbol{P}(V)$.
- The set of integers between the values m and n will be denoted as $[m..n]$, i.e.,

$$[m..n] := \{ i \in \mathbf{Z} \mid m \leqslant i \leqslant n \}.$$

- Universal quantification is denoted as

$$(\underline{A}\, x : \mathrm{R}(x) : \mathrm{P}(x));$$

  it expresses that for all x satisfying restriction $\mathrm{R}(x)$, predicate $\mathrm{P}(x)$ holds. Instead of x, a sequence of variables may be used. In the same way

$$(\underline{E}\, x : \mathrm{R}(x) : \mathrm{P}(x));$$

  is used to denote existential quantification.
- A similar notation will be used for operations on sets. For instance,

$$(\cup\, i : \mathrm{R}(i) : V(i))$$

  denotes the union of all sets $V(i)$, where i satisfies restriction $\mathrm{R}(i)$, and

$$(\underline{\mathrm{sum}}\ i : \mathrm{R}(i) : x(i)) \quad \text{and} \quad (\underline{\mathrm{max}}\ i : \mathrm{R}(i) : x(i))$$

  denote the sum and maximum respectively of all numbers $x(i)$, where i satisfies restriction $\mathrm{R}(i)$. (Apart from $\underline{\mathrm{max}}$ and $\underline{\mathrm{min}}$, we shall also use the notations $\max(x, y)$

and $\min(x, y)$ for the maximum and minimum values of the numbers x and y.)

- In proofs, equalities or inequalities are often derived in a sequence of steps. For example, to prove that $V_0 \subset V_2$, one may derive that $V_0 = V_1$ and $V_1 \subset V_2$. To avoid the annoyance of writing down the formula of V_1 twice, we shall often use the following layout.

$$
\begin{array}{ll}
 & V_0 \\
= & \quad \{\text{ hint why } V_0 = V_1 \} \\
 & V_1 \\
\subset & \quad \{\text{ hint why } V_1 = V_2 \} \\
 & V_2
\end{array}
$$

This convention is taken from [Dijkstra & Feijen 1984].

1

Digitisation functions

1.0 Introduction

We consider digitisation as a mapping from $\mathcal{P}(\mathbf{R}^2)$ into $\mathcal{P}(\mathbf{Z}^2)$. In this chapter we provide a framework in which it is possible to specify a broad range of digitisation functions, corresponding to the different needs of different computer graphics applications. Each application may need its own set of desirable properties, with its own priority distribution.

An evident desirable property is closeness; its formal definition is based on a *distance function*. As said in Chapter 0, digital images consist of a finite number of discrete elements, called pixels, arranged in a square grid. This square grid may be represented by $\mathbf{Z}^2$. To measure distances of pixels, we do not use the Euclidean notion of distance, but define a distance function in which the distance of two pixels is expressed in the number of pixels that separates them. This distance function will be formally introduced in Section 1.

Section 2 contains the general definition of digitisation functions, together with some examples. Sections 3, 4, and 5 deal with the properties *translation invariance*, *closeness*, and *convexity* respectively. Section 4 includes the introduction of a *vicinity measure*, which expresses the quality of a digitisation function with respect to closeness. Section 6 contains some concluding remarks.

1.1 Metric space

The digital images we consider are subsets of the metric space $(\mathbf{Z}^2, d)$, where the distance function $d : \mathbf{R}^2 \times \mathbf{R}^2 \to \mathbf{R}$ is defined by

$$d(v, w) := \max(\,|v.x - w.x\,|,\ |v.y - w.y\,|\,).$$

For a treatise on metric spaces in general we refer to [Shreider 1974], and on $(\mathbf{Z}^2, d)$ to [Rosenfeld & Pfaltz 1968].

The following property follows directly from the definition of the distance function d; it expresses that the distance of two points is invariant under translation.

Property 1.0:

For all $v_0, v_1, w \in \mathbf{R}^2$,

$$d(w+v_0, w+v_1) = d(v_0, v_1).$$

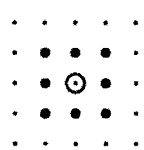
□

The distance of a point to a nonempty set of points is defined by

$$\tilde{d}(v, W) := (\underline{\inf}\, w : w \in W : d(v, w)),$$

where <u>inf</u> stands for infimum.

A *neighbour* of a pixel p is a pixel at distance 1 of p, that is, a pixel q such that $d(p,q)=1$. In the figure below all neighbours (indicated by •) of pixel p (indicated by ○) are shown.



In many papers on this subject, the name d_8 or $d8$ is used instead of d, referring to the number of neighbours each pixel has in this metric space.

A *path* π *from* p *to* q is a sequence of pixels $r_0, r_1, \ldots, r_n$ such that $r_0 = p$, $r_n = q$, and for all $i : 0 < i \leqslant n$, r_i is a neighbour of r_{i-1}. We shall call n the *length* of the path. The set of all elements of the path is denoted as $<\pi>$, i.e.,

$$<\pi> = \{r_i \mid 0 \leqslant i \leqslant n\}.$$

Note that $\pi = r_0, r_1, \ldots, r_n$ is a path from r_0 to r_n if and only if $r_n, r_{n-1}, \ldots, r_0$ is a path from r_n to r_0. This second path is referred to as π^{-1}.

For given p and q, the lengths of the paths from p to q have a lower bound, as is expressed by the following property.

Property 1.1: (See [Rosenfeld 1978], for instance)

The length of a path from p to q is at least $d(p,q)$.

□

If P is a subset of $\mathbf{Z}^2$ containing the pixels p and q, then p and q are said to be *connected in* P if a path from p to q exists whose elements are all contained in P. Note that the relation "connected in P" is both reflexive, symmetric, and transitive, and therefore yields an equivalence relation; its classes are called the *connected components of* P. P itself is called *connected* if it has only one connected component, that is, if any pair of points in P is connected in P.

1.2 Digitisation function

Digitisation functions are partial functions from $\mathbf{P}(\mathbf{R}^2)$ into $\mathbf{P}(\mathbf{Z}^2)$. Domain and range of a digitisation function f are denoted as D_f and R_f respectively. The domain of most digitisation functions used in Computer Graphics nowadays, is a restricted class of subsets of $\mathbf{R}^2$: the class of all line segments, for example, or the class of all polygons. Such functions are referred to as *line segment digitisation functions* and *convex polygon digitisation functions* respectively.

Two simple digitisation functions with domain $\mathbf{P}(\mathbf{R}^2)$ are now presented.

Example 1.2:

For all $V \subseteq \mathbf{R}^2$,

$$f(V) = \varnothing.$$

□

This digitisation function has no practical use; the following one, however, is commonly used for polygons, as noted in Chapter 0.

Example 1.3:

For all $V \subseteq \mathbf{R}^2$,

$$f(V) = V \cap \mathbf{Z}^2.$$

□

In the following sections we shall introduce a classification of digitisation functions, based on various criteria.

1.3 Translation invariance

The first criterion concerns the invariance of a digitisation function under translation. To express this formally, we introduce an operator that translates subsets of $\mathbf{R}^2$.

For w an element of $\mathbf{R}^2$ and V a subset of $\mathbf{R}^2$, the translation operator $\oplus$ is defined by

$$w \oplus V \ := \ \{ w + v \mid v \in V \}.$$

For this operator, the following property holds.

Property 1.4:

For all $w, w_0, w_1 \in \mathbf{R}^2$ and all $V, V_0, V_1 \subseteq \mathbf{R}^2$,

$$w \oplus (V_0 \cup V_1) = (w \oplus V_0) \cup (w \oplus V_1) \qquad \text{(a)}$$

$$(w_0 + w_1) \oplus V = w_0 \oplus (w_1 \oplus V) \qquad \text{(b)}$$

$$w \oplus (V_0 \cap V_1) = (w \oplus V_0) \cap (w \oplus V_1). \qquad \text{(c)}$$

□

A digitisation function f is called ***translation invariant*** iff for all $V \in D_f$ and all $p \in \mathbf{Z}^2$ such that $p \oplus V \in D_f$,

$$f(p \oplus V) = p \oplus f(V).$$

Although this property might seem to be very natural for digitisation functions, we shall see in the succeeding chapters that it is incompatible with some combinations of other desirable properties.

Example 1.5:

The digitisation function of example 1.2 is translation invariant, since for all $w \in \mathbf{R}^2$,

$$w \oplus \varnothing = \varnothing.$$

□

Example 1.6:

The digitisation function of example 1.3 is translation invariant, since for all $V \subseteq \mathbf{R}^2$ and $p \in \mathbf{Z}^2$ holds,

$f(p \oplus V)$

$=$ { definition f }

$(p \oplus V) \cap \mathbf{Z}^2$

$=$ { definition $\oplus$ }

$$
\begin{aligned}
& \{ p+v \mid v \in V \} \cap \mathbf{Z}^2 \\
= \quad & \{ \text{calculus} \} \\
& \{ p+v \mid v \in V \wedge p+v \in \mathbf{Z}^2 \} \\
= \quad & \{ p \in \mathbf{Z}^2 \} \\
& \{ p+v \mid v \in V \wedge v \in \mathbf{Z}^2 \} \\
= \quad & \{ \text{calculus} \} \\
& \{ p+v \mid v \in V \cap \mathbf{Z}^2 \} \\
= \quad & \{ \text{definition of } \oplus \} \\
& p \oplus (V \cap \mathbf{Z}^2) \\
= \quad & \{ \text{definition of } f \} \\
& p \oplus f(V).
\end{aligned}
$$

□

On the analogy of Property 1.0, the following property expresses that the distance of a point to a set of points is invariant under translation.

Property 1.7:

For all $v_0, v_1 \in \mathbf{R}^2$ and $W \subseteq \mathbf{R}^2$,

$$\tilde{d}(v_0+v_1, v_0 \oplus W) = \tilde{d}(v_1, W).$$

□

1.4 Closeness

In most Computer Graphics applications, digitisation functions are required whose images resemble the originals as much as possible. In this section we shall formalise the word 'resemble'.

1.4.0 Close digitisation

A set $P \subseteq \mathbf{Z}^2$ is called a *close digitisation* of a set $V \subseteq \mathbf{R}^2$ iff

$$(\underline{A}p : p \in P : (\underline{E}v : v \in V : d(p,v) < 1)) \qquad \text{(cd0)}$$

$$\wedge \; (\underline{A}v : v \in V : (\underline{E}p : p \in P : d(p,v) < 1)). \qquad \text{(cd1)}$$

The notation $P \vdash V$ is used to denote that P is a close digitisation of V.

There are three remarkable aspects in this definition. Firstly, the symmetry between conditions (cd0) and (cd1). Secondly, the use of the distance function d instead of the Euclidean distance function. And thirdly, the use of value 1 in the predicates of the existential quantifications.

Concerning the symmetry, we would like to remark that the second condition is hardly ever mentioned explicitly. However, this condition guarantees that close digitisations are 'large' enough. Consider, for instance, the sets

$$P = \{(0,10), (0,-10)\}, \quad V = \{(x,y) \in \mathbf{R}^2 \mid x^2 + y^2 \leqslant 100\}.$$

P and V do satisfy (cd0), but hardly anyone would consider P a good digitisation of V.

The use of the distance function d is motivated as follows. Consider for $p \in \mathbf{Z}^2$ the set

$$\mathrm{CRS}(p) := \{v \in \mathbf{R}^2 \mid d(v,p) < 1\} \qquad \text{(a)}$$

which is called the *close region of sensitivity* of p. See Figure 1.0(a). Substituting the Euclidean distance function for d in (a) would result in a circular region of sensitivity, as is illustrated in Figure 1. 0(b).

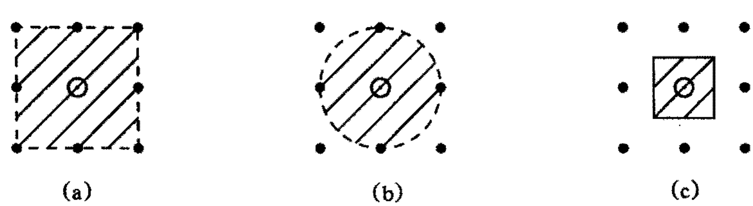


Figure 1. 0

Regions of sensitivity of pixel p.
p is indicated by ○, its neighbours are indicated by •.

a) close region of sensitivity
b) Euclidean region of sensitivity
c) optimal region of sensitivity

We think that the regions of sensitivity associated with d, go better with the square grid in which we consider our digital images, than the circular regions do. The notion 'region of sensitivity' is introduced by [Dorst 1986], though in a slightly different meaning.

The third remark concerns the value 1 in conditions (cd0) and (cd1). Since $z = ½$ is the smallest value such that for any pixel p a point v exists such that $d(p,v) \leqslant z$, the following class of close digitisations may be seen as 'optimal' digitisations.

P is called an *optimal digitisation* of V iff

$$(\underline{A}p : p \in P : (\underline{E}v : v \in V : d(p,v) \leqslant ½)) \qquad \text{(od0)}$$

$$\wedge\ (\underline{A}v : v \in V : (\underline{E}p : p \in P : d(p,v) \leqslant ½)). \qquad \text{(od1)}$$

The notation $P \models V$ is used to denote that P is an optimal digitisation of V.

The set

$$\mathrm{ORS}(p) := \{v \in \mathbf{R}^2 \mid d(v,p) \leqslant ½\}$$

is called the *optimal region of sensitivity* of p. In Figure 1.0(c) the optimal region of sensitivity of pixel p is shown.

Conditions (cd0) and (cd1) may be rewritten as

$$(\underline{A}p : p \in P : \mathrm{CRS}(p) \cap V \neq \varnothing) \qquad \text{(cd'0)}$$

$$\wedge\ (\underline{A}v : v \in V : \mathrm{CRS}(v) \cap P \neq \varnothing), \qquad \text{(cd'1)}$$

where the definition of CRS has been generalised for points.

In a similar way, conditions (od0) and (od1) may be rewritten using ORS(p).

Obviously, any optimal digitisation is also a close digitisation. In Figure 1.1 a set V is shown, together with some close digitisations. The digitisation in Figure 1. 1(d) is optimal. The one in (c) is a subset hereof, but is not optimal since condition (od1) is not satisfied.

The following properties are stated without proof. The first one expresses that the union of two close (optimal) digitisations is also a close (optimal) digitisation.

Property 1.8:

For all $P_0, P_1 \subseteq \mathbf{Z}^2$ and $V \subseteq \mathbf{R}^2$,

$$P_0 \vdash V \wedge P_1 \vdash V \Rightarrow (P_0 \cup P_1) \vdash V$$

and

$$P_0 \models V \wedge P_1 \models V \Rightarrow (P_0 \cup P_1) \models V.$$

□

The following property expresses that a close digitisation of V contains all pixels of V.

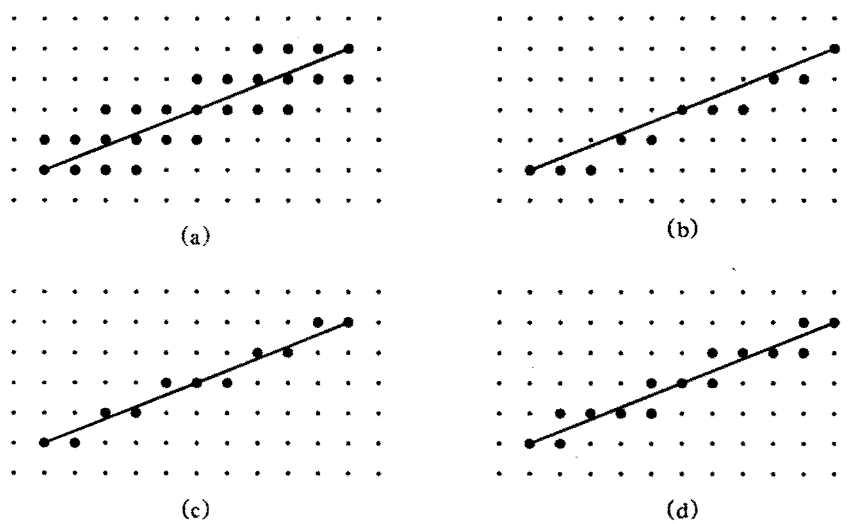


Figure 1. 1

A line segment in $\mathbf{R}^2$, together with examples of close digitisations; the one of (d) is also optimal.

Property 1.9:

For all $P \subseteq \mathbf{Z}^2$ and $V \subseteq \mathbf{R}^2$,

$$P \vdash V \Rightarrow V \cap \mathbf{Z}^2 \subseteq P.$$

□

The following property expresses that any close digitisation of a set containing pixels only, is the set itself.

Property 1.10:

For all $P_0, P_1 \subseteq \mathbf{Z}^2$,

$$P_0 \vdash P_1 \Rightarrow P_0 = P_1.$$

□

1.4.1 Close digitisation function

The notions 'close' and 'optimal' are extended to digitisation functions in a straightforward manner.

A digitisation function f is called *close* iff $f(V)$ is a close digitisation of V, for all $V \in D_f$, that is, iff

$$(\underline{\mathrm{A}}\, V : V \in D_f : f(V) \vdash V\ ).$$

Similarly, f is called *optimal* iff

$$(\underline{\mathrm{A}}\, V : V \in D_f : f(V) \models V\ ).$$

Example 1.11:

The digitisation function of example 1.2 is not close, since condition (cd1) is not satisfied for $V \neq \varnothing$.

□

Example 1.12:

The digitisation function of example 1.3 is not close: for $V = \{ v \}$, where $v \in \mathbf{R}^2 \backslash \mathbf{Z}^2$, $f(V) = \varnothing$, and hence condition (cd1) is not satisfied.

□

We proceed with the introduction of two digitisation functions that are close and optimal respectively.

The digitisation functions $f_{\vdash} : \boldsymbol{P}(\mathbf{R}^2) \rightarrow \boldsymbol{P}(\mathbf{Z}^2)$ and $f_{\models} : \boldsymbol{P}(\mathbf{R}^2) \rightarrow \boldsymbol{P}(\mathbf{Z}^2)$ are defined by

$$f_{\vdash}(V) := \{ p \in \mathbf{Z}^2 \mid (\underline{\mathrm{E}}\, v : v \in V : d(p,v) < 1) \}$$

$$f_{\models}(V) := \{ p \in \mathbf{Z}^2 \mid (\underline{\mathrm{E}}\, v : v \in V : d(p,v) \leqslant ½) \}$$

$f_{\vdash}(V)$ is called the *Close Embedding of V*, and $f_{\models}(V)$ the *Optimal Embedding of V*. Similarly, $f_{\vdash}$ and $f_{\models}$ are called the *Close Embedding function* and *Optimal Embedding function* respectively.

These functions have the following properties.

Property 1.13:

The digitisation function $f_{\vdash}$ is close, and $f_{\models}$ is optimal.

□

Property 1.14:

For all $V \subseteq \mathbf{R}^2$,

$$V \cap \mathbf{Z}^2 \subseteq f_{\vdash}(V).$$

Proof:

From Property 1.9.

□

Property 1.15:

$f_{\vdash}$ and $f_{\models}$ are translation invariant.

Proof:

Let $r \in \mathbf{Z}^2$. Then for any $V \subseteq \mathbf{R}^2$ holds,

$$
\begin{aligned}
& f_{\vdash}(r \oplus V) \\
= \quad & \{\text{ definition of } f_{\vdash} \ \} \\
& \{p \in \mathbf{Z}^2 \mid (\underline{E}v : v \in r \oplus V : d(p,v) < 1)\} \\
= \quad & \{\text{ definition of } \oplus \text{ and renaming dummy variable } v \ \} \\
& \{p \in \mathbf{Z}^2 \mid (\underline{E}v : v \in V : d(p, r+v) < 1)\} \\
= \quad & \{\text{ renaming dummy variable } p \ \} \\
& \{r+p \in \mathbf{Z}^2 \mid (\underline{E}v : v \in V : d(r+p, r+v) < 1)\} \\
= \quad & \{\text{ Property 1.0 and definition of } \oplus \ \} \\
& r \oplus \{p \in \mathbf{Z}^2 \mid (\underline{E}v : v \in V : d(p,v) < 1)\} \\
= \quad & \{\text{ definition of } f_{\vdash} \ \} \\
& r \oplus f_{\vdash}(V).
\end{aligned}
$$

For $f_{\models}$ a similar derivation may be used.

□

Property 1.16:

For all $V, V_0, V_1 \subseteq \mathbf{R}^2$,

$f_\vdash$	$f_\models$
$f_\vdash(V) = (\cup P : P \subseteq \mathbf{Z}^2 \wedge P \vdash V : P)$	$f_\models(V) = (\cup P : P \subseteq \mathbf{Z}^2 \wedge P \models V : P)$
$f_\vdash(f_\vdash(V)) = f_\vdash(V)$	$f_\models(f_\models(V)) = f_\models(V)$

($f_\vdash$ and $f_\models$ are idempotent)

$$f_\models(V) \subseteq f_\vdash(V)$$

$V_0 \subseteq V_1 \;\Rightarrow\; f_\vdash(V_0) \subseteq f_\vdash(V_1)$	$V_0 \subseteq V_1 \;\Rightarrow\; f_\models(V_0) \subseteq f_\models(V_1)$

($f_\vdash$ and $f_\models$ are monotone)

$f_\vdash(V_0 \cup V_1) = f_\vdash(V_0) \cup f_\vdash(V_1)$	$f_\models(V_0 \cup V_1) = f_\models(V_0) \cup f_\models(V_1)$
($f_\vdash$ and $\cup$ are distributive)	($f_\models$ and $\cup$ are distributive)

□

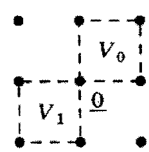
The digitisation functions $f_\vdash$ and $f_\models$ are not distributive with regard to intersection, as is shown in the following example.

Example 1.17:

Let $w_0 = (½, ½)$, $w_1 = (-½, -½)$, and let $V_0, V_1 \subseteq \mathbf{R}^2$ be defined as follows.

$$V_0 = \{ v \in \mathbf{R}^2 \mid d(v, w_0) < ½ \},$$

$$V_1 = \{ v \in \mathbf{R}^2 \mid d(v, w_1) < ½ \}.$$



Then

$$V_0 \cap V_1 = \varnothing$$

and hence

$$f_\vdash(V_0 \cap V_1) = \varnothing, \quad f_\models(V_0 \cap V_1) = \varnothing.$$

Since

$$f_\vdash(V_0) = f_\models(V_0) = \{(0,0), (1,0), (1,1), (0,1)\}$$

and

$$f_\vdash(V_1) = f_\models(V_1) = \{(0,0), (-1,0), (-1,-1), (0,-1)\},$$

it follows that

$$f_\vdash(V_0) \cap f_\vdash(V_1) = f_\models(V_0) \cap f_\models(V_1) = \{(0,0)\}.$$

Therefore

$$f_{\vdash}(V_0 \cap V_1) \neq f_{\vdash}(V_0) \cap f_{\vdash}(V_1)$$
$$f_{\models}(V_0 \cap V_1) \neq f_{\models}(V_0) \cap f_{\models}(V_1).$$

□

1.4.2 Vicinity measure

For digitisation functions that are not close, one would like to have the possibility to express how 'unclose' these functions are. For this purpose we introduce the vicinity measure $e_f : D_f \rightarrow \mathbf{R}^+$(including 0), which yields the maximum distance of any element of $f(V)$ to V and of any element of V to $f(V)$. Its definition is based on $\tilde{d}$, as introduced in section 1.1.

For a digitisation function f the *vicinity function* $e_f : D_f \rightarrow \mathbf{R}^+ \cup \{\infty\}$ is defined by

$$e_f(V) := \begin{cases} 0 & \text{if } V = \varnothing \wedge f(V) = \varnothing \\ \max(e0_f(V), e1_f(V)) & \text{if } V \neq \varnothing \wedge f(V) \neq \varnothing \\ \infty & \text{otherwise,} \end{cases}$$

where the functions $e0_f, e1_f : D_f \rightarrow \mathbf{R}^+ \cup \{\infty\}$ are defined by

$$e0_f(V) := (\underline{\sup}\, p : p \in f(V) : \tilde{d}(p, V))$$
$$e1_f(V) := (\underline{\sup}\, v : v \in V : \tilde{d}(v, f(V))),$$

and where sup stands for supremum. Note the symmetry in the definition of e_f.

Example 1.18:

Let $V = \{(x, y) \in \mathbf{R}^2 \mid x^2 + y^2 \leqslant 100\}$, and let f be a digitisation function such that $e_f(V) = \{(0,10),(0,-10)\}$.
Then $e0_f(V) = 0$, whereas $e1_f(V) = 10$. Hence $e_f(V) = 10$.

□

Example 1.19:

For the digitisation function of Example 1.2,

$$e_f(V) = \begin{cases} 0 & \text{if } V = \varnothing \\ \infty & \text{if } V \neq \varnothing. \end{cases}$$

□

Example 1.20:

For the digitisation function of Example 1.3,

$$e_f(V) \begin{cases} = 0 & \text{if } V \cap \mathbf{Z}^2 = V \\ = \infty & \text{if } V \cap \mathbf{Z}^2 = \varnothing \wedge V \neq \varnothing \\ \in (0, \infty) & \text{otherwise.} \end{cases}$$

□

Property 1.21:

For any close digitisation function f holds,

$$(\underline{\mathrm{A}}\, V : V \in D_f : e_f(V) \leqslant 1). \qquad \text{(a)}$$

For any optimal digitisation function f holds,

$$(\underline{\mathrm{A}}\, V : V \in D_f : e_f(V) \leqslant \tfrac{1}{2}). \qquad \text{(b)}$$

□

Note the occurrence of $\leqslant$ instead of $<$ in (a); this is because the supremum of an infinite set whose elements are all smaller than 1, might equal 1.

In Chapter 2, the vicinity measure will be discussed in more detail.

For translation invariant digitisation functions, the vicinity function is invariant under translation, as is expressed in the following property.

Property 1.22:

If f is a translation invariant digitisation function, then for all $V \in D_f$ and all $p \in \mathbf{Z}^2$ such that $p \oplus V \in D_f$ holds,

$$e_f(p \oplus V) = e_f(V).$$

Proof:

Let $V \in D_f$, such that $V \neq \varnothing$ and $f(V) \neq \varnothing$. Let $p \in \mathbf{Z}^2$ such that $p \oplus V \in D_f$. Then

$e_f(p \oplus V)$

$=$ $\{$ definition of e_f and $V \neq \varnothing$ and $f(p \oplus V) \neq \varnothing$ $\}$

$\max(\, (\underline{\sup}\, v : v \in p \oplus V : \tilde{d}(v, f(p \oplus V)),$
$\quad (\underline{\sup}\, q : q \in f(p \oplus V) : \tilde{d}(q, p \oplus V))\,)$

$=$ $\{$ f is translation invariant $\}$

$\max(\, (\underline{\sup}\, v : v \in p \oplus V : \tilde{d}(v, p \oplus f(V)),$

$$
\begin{aligned}
& \qquad (\underline{\sup}\, q : q \in p \oplus f(V) : \tilde{d}(q, p \oplus V))\ ) \\
= \quad & \{ \text{ renaming dummy variables } \} \\
& \max(\ (\underline{\sup}\, v : v \in V : \tilde{d}(p+v, p \oplus f(V)), \\
& \qquad (\underline{\sup}\, q : q \in f(V) : \tilde{d}(p+q, p \oplus V))\ ) \\
= \quad & \{ \text{ Property 1.7 } \} \\
& \max(\ (\underline{\sup}\, v : v \in V : \tilde{d}(v, f(V)), \\
& \qquad (\underline{\sup}\, q : q \in f(V) : \tilde{d}(q, V))\ ) \\
= \quad & \{ \text{ definition of } e_f \ \} \\
& e_f(V).
\end{aligned}
$$

□

1.5 Convexity

Subsets of $\mathbf{R}^2$ are often specified by a finite number of elements of $\mathbf{R}^2$ and an implicit function that maps these elements into a subset of $\mathbf{R}^2$. For example, line segments are usually specified by their two end points, triangles by their three vertices, and n-gons by their n vertices. Hence, if f is a digitisation function whose domain is the set of all line segments in $\mathbf{R}^2$, and g is the function that maps a pair of end points to a line segment, then $f \circ g$ is a function from $\mathbf{R}^2 \times \mathbf{R}^2$ into $\boldsymbol{P}(\mathbf{Z}^2)$.

The following definition concerns digitisation functions whose domain depends on such a function g.

Let $n > 0$, g be a function from $(\mathbf{R}^2)^n$ into $\boldsymbol{P}(\mathbf{R}^2)$, and let f be a digitisation function such that $D_f = R_g$.
Then f is called *convex with respect to* g iff for all $\underline{v} \in (\mathbf{R}^2)^n$ and all $\underline{w} \in (f \circ g(\underline{v}))^n$:

$$f \circ g(\underline{w}) \subseteq f \circ g(\underline{v}) \tag{a}$$

For instance, if f is a line segment digitisation function, and $g : \mathbf{R}^2 \times \mathbf{R}^2 \rightarrow \boldsymbol{P}(\mathbf{R}^2)$ is a function that maps any pair of points to the line segment in $\mathbf{R}^2$ that connects these points, then condition (a) means that $f \circ g$, applied to any pair of points from a set $f \circ g(v, w)$ should generate a subset of $f \circ g(v, w)$. More formally expressed (where $[v, w]$ denotes the line segment in $\mathbf{R}^2$ connecting v and w and $f[v, w]$ denotes $f([v, w])$):

$$(\underline{\mathrm{A}}\, v, w : v, w \in \mathbf{R}^2 : (\underline{\mathrm{A}}\, p, q : p, q \in f[v, w] : f[p, q] \subseteq f[v, w])).$$

A similar condition is firstly mentioned by [Franklin 1986], who called it the 'subset property'. In [Luby 1986] it is called the 'subpath property'. Franklin states that the line segment digitisation functions commonly used do not satisfy this property. In the following chapter we shall show that convexity hardly combines with closeness.

Convexity is a desirable property for line segment digitisation functions in interactive applications where parts of lines on the screen have to be erased and displayed again. See [Luby 1986] for examples of applications. If the digitisation function is convex, then erasing may be performed by digitising the part that is to be erased using the same digitisation function and setting these pixels to background colour. If the digitisation function would not be convex, then pixels not belonging to the original set might be set to background colour, and, reversely, pixels that do belong to the original set, may not have been set to background colour. Also, if the erased part is displayed again afterwards, a convex digitisation function guarantees that the resulting pixel set is exactly the same as it was before erasing the part in question.

Chapter 4 deals with the construction of convex digitisation functions for line segments within a limited domain.

1.6 Concluding remarks

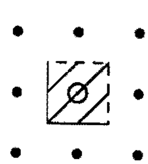
In this chapter we have introduced the notion *digitisation function*, and formulated some properties that may be desirable in particular applications. These properties are *translation invariance*, *closeness*, and *convexity*.

Of these properties, *closeness* is the most important one. A vicinity measure has been introduced to express the quality of a digitisation function with respect to closeness. The Close Embedding and Optimal Embedding functions are examples of digitisation functions that are close. In [van Overveld 1987a] an algorithm is presented for the generation of the Embedding digitisations.

The notion *region of sensitivity* is introduced based on the definition of *close digitisation*. The other way around is also possible: starting from some definition of *region of sensitivity*, the notion *close digitisation* may then be defined by conditions (cd0) and (cd1). This may be useful, for instance, in cases where a partition of $\mathbf{R}^2$ (instead of a covering) in regions of sensitivity is required. In this case, each element of $\mathbf{R}^2$ is contained in exactly one region of sensitivity. An example of such definition of region of sensitivity is

$$\mathrm{RS}(p) := \{ v \in \mathbf{R}^2 \mid -\tfrac{1}{2} \leqslant v.x - p.x < \tfrac{1}{2} \wedge -\tfrac{1}{2} \leqslant v.y - p.y < \tfrac{1}{2} \},$$

as is illustrated below, where p is indicated by $\circ$.
It might be interesting to investigate the implications of such definition.



2

Line functions

2.0 Introduction

In this chapter we discuss line segment digitisation functions (in short: line functions). Like all digitisation functions, their range is a subset of $\boldsymbol{P}(\mathbf{Z}^2)$. The line functions we consider have as domain the set of all line segments in $\mathbf{R}^2$ connecting two pixels. This is the set of line segments that is usually considered in Computer Graphics.

If f is a line function, then the elements of R_f will be called digitised line segments of f. We shall denote the line segment in $\mathbf{R}^2$ that connects the pixels p and q as $[p,q]$. Note that if $p=q$, then $[p,q]$ degenerates to $\{p\}$. For clarity, we shall often omit the parentheses from expressions like $f([p,q])$.

In Computer Graphics literature, line functions are seldom discussed explicitly; instead, algorithms are presented, and one is supposed to know intuitively what pixels should be generated by these algorithms.

The algorithm commonly used for digitising line segments is the Bresenham algorithm ([Bresenham 1965], also presented in [van Berckel & Mailloux 1965]), which uses integer arithmetic only. For a line segment $[p,q]$ with $p.x<q.x$, $p.y \leqslant q.y$, and $q.y-p.y \leqslant q.x-p.x$, it generates the pixel set

$$\{(x, \lceil g(x)-\tfrac{1}{2}\rceil) \in \mathbf{Z}^2 \mid p.x \leqslant x \leqslant q.x\},$$

where the function $g:\mathbf{R}\rightarrow\mathbf{R}$ is the function that prescribes the y-value of points contained in the line segment $[p,q]$ as a function of their x-value. The Bresenham algorithm selects for each $x \in [p.x..q.x]$ a pixel p, where $p.x=x$, that minimises $d(p,(x,g(x)))$.

The set of pixels generated by this algorithm is connected, and contains $q.x-p.x+1$ pixels. It is also a close digitisation of $[p,q]$, as will be shown in Section 2.4.5. Besides the line function associated with the Bresenham algorithm, there are many more line functions.

In section 1 we present the general definition of a line function, and we relate the properties of Chapter 1, translation invariance, closeness, and convexity, to this definition. In section 2 we shall introduce a new kind of vicinity measure for line functions. In section 3 we introduce a new property for line functions, namely *minimality*. This property refers to the number of pixels in the digitised line segments. In this section we also introduce the notion *minimal path*, and its representation by means of *chain codes*. In section 4 we present seven examples of line functions, classifying them with regard to the properties mentioned above. In section 5 we present some important theorems on the incompatibility of some combinations of properties. Finally, we conclude with some remarks in section 6.

2.1 Definition

The following definition of line functions is a very general one: we only require that the endpoints of a line segment are contained in a finite, connected pixel set.

A function f that maps line segments in $\mathbf{R}^2$ with pixel endpoints onto subsets of $\mathbf{Z}^2$ is called a *line function* iff f satisfies the following properties.

lf0) $(\underline{\mathbf{A}}\, p,q : p,q \in \mathbf{Z}^2 : \{p,q\} \subseteq f[p,q])$

lf1) $(\underline{\mathbf{A}}\, p,q : p,q \in \mathbf{Z}^2 : f[p,q]$ is finite and connected$)$

Notice that, since $[p,q] = [q,p]$, any line function is consistent under the exchange of p and q. To meet this requirement in the examples of section 4, we shall use the following proposition, which expresses what the position of two pixels is with respect to each other.

$$\mathrm{NF}(p,q) :\equiv \ (p.x < q.x \ \vee \ (p.x = q.x \wedge p.y \leqslant q.y)).$$

In words, $\mathrm{NF}(p,q)$ expresses that p is the leftmost pixel of $\{p,q\}$, or, if $p.x = q.x$, it is the downmost one. Note that for all pixels p,q holds

$$\mathrm{NF}(p,q) \vee \mathrm{NF}(q,p),$$

and

$$(\mathrm{NF}(p,q) \wedge \mathrm{NF}(q,p)) \ \equiv \ (p = q).$$

For pixels p,q we shall use the notation $\hat{p},\hat{q}$ for the permuted pair of pixels that satisfies

$$\{p,q\} = \{\hat{p},\hat{q}\} \ \wedge \ \mathrm{NF}(\hat{p},\hat{q}).$$

The following property expresses that if p is the leftmost pixel of $\{p,q\}$ (or the downmost one), then $r+p$ is the leftmost of $\{r+p, r+q\}$ (or the downmost one respectively).

Property 2.0:

For all $p,q,r \in \mathbf{Z}^2$,

$$\mathrm{NF}(p,q) \ \Rightarrow \ \mathrm{NF}(r+p, r+q).$$

□

In the previous chapter some properties of digitisation functions have been introduced, namely translation invariance, closeness, and convexity. We shall apply these general definitions to line functions.

A line function is translation invariant iff

$$(\underline{A}\, p,q,r : p,q,r \in \mathbf{Z}^2 : f(r \oplus [p,q]) = r \oplus f[p,q]).$$

Since $r \oplus [p,q] = [r+p, r+q]$, this is the same as

$$(\underline{A}\, p,q,r : p,q,r \in \mathbf{Z}^2 : f[r+p, r+q] = r \oplus f[p,q]).$$

In Section 1.4 we have defined what close digitisation functions are; applying this to line functions we get the following characterisation: a line function f is close iff for all $p,q \in \mathbf{Z}^2$ holds,

$$(\underline{A}\, r : r \in f[p,q] : (\underline{E}\, v : v \in [p,q] : d(r,v) < 1))$$
$$\wedge\ (\underline{A}\, v : v \in [p,q] : (\underline{E}\, r : r \in f[p,q] : d(r,v) < 1)).$$

In Section 1.5 the notion convex digitisation function has been introduced, as well as its application to line functions: a line function f is convex iff

$$(\underline{A}\, v,w : v,w \in \mathbf{R}^2 : (\underline{A}\, p,q : p,q \in f[v,w] : f[p,q] \subseteq f[v,w])).$$

Since we only deal with line segments whose endpoints are pixels, this may be rewritten as

$$(\underline{A}\, r,s : r,s \in \mathbf{Z}^2 : (\underline{A}\, p,q : p,q \in f[r,s]) : f[p,q] \subseteq f[r,s])).$$

In section 2.4 we present various examples of line functions, where each function will be investigated with respect to the above properties.

2.2 Vicinity for line functions

In Section 1.4 the vicinity measure for digitisation functions has been introduced. Applying this definition to line functions, it follows that the vicinity measure e_f for a line function f is defined by

$$e_f([p,q]) = \max(e0_f([p,q]), e1_f([p,q])),$$

where

$$e0_f([p,q]) = (\underline{\max}\, r : r \in f[p,q] : \tilde{d}(r,[p,q])).$$
$$e1_f([p,q]) = (\underline{\sup}\, v : v \in [p,q] : \tilde{d}(v, f[p,q])).$$

From now on, the parentheses in expressions of e_f, $e0_f$, and $e1_f$ will be omitted.

We shall demonstrate that, due to the connectedness of $f[p,q]$,

$$e1_f[p,q] \leqslant 1 + e0_f[p,q]. \qquad \text{(a)}$$

Because of this inequality, we shall introduce for line functions a new kind of vicinity measure, one that depends solely on $e\,0_f$. Note that, if (a) holds, then for all line functions f, and all $p,q \in \mathbf{Z}^2$,

$$e\,0_f[p,q] \leqslant e_f([p,q]) \leqslant 1 + e\,0_f[p,q].$$

First some notational remarks.

For $p,q \in \mathbf{Z}^2$, $p \neq q$, $[\![p,q]\!]$ will denote the infinite line in $\mathbf{R}^2$ that contains both p and q. For $p=q$, $[\![p,q]\!] := \{p\}$.

Next we present some properties that will be needed to prove (a).

Property 2.1:

For all $p,q \in \mathbf{Z}^2$, and all $v \in [p,q]$,

$$d(p,v)+d(v,q) = d(p,q).$$

□

The following property expresses that the distance of a pixel r to an infinite line equals the distance of r to the intersection point of that line with either the line through r of slope 1 or of slope -1.

Property 2.2:

For any $p,q,r \in \mathbf{Z}^2$, $p \neq q$,

$$\tilde{d}(r,[\![p,q]\!]) = d(r,v),$$

where $v \in \mathbf{R}^2$ is the intersection point of $[\![p,q]\!]$ with

$$\begin{cases} y+x = r.y+r.x & \text{if } (q.y-p.y)*(q.x-p.x) \geqslant 0 \\ y-x = r.y-r.x & \text{otherwise.} \end{cases}$$

If $v \in [p,q]$, then also

$$\tilde{d}(r,[p,q]) = d(r,v).$$

Hint:

We present an intuitive argument only.
The set of all points that have distance z to pixel r form a square with center r. If z increases, starting at 0, the first square to hit the line $[\![p,q]\!]$ determines $\tilde{d}(r,[\![p,q]\!])$. If the line is horizontal or vertical, the square will be hitted along one of its edges, otherwise in one of its corners. The corners are elements of the lines through r with slopes 1 and -1.

□

In the following lemma (a) is proven.

Lemma 2.3:

For any line function f and pixels $p, q \in \mathbf{Z}^2$,

$$e1_f[p,q] \leqslant 1 + e0_f[p,q].$$

Proof:

Let f be a line function and $p, q \in \mathbf{Z}^2$.
If $p = q$, then

$$\begin{aligned}(\underline{\sup}\, v : v \in [p,q] : \tilde{d}(v, f[p,q])) &= d(p, f[p,q]) \\ &= 0 \\ &\leqslant 1 + e0_f[p,q].\end{aligned}$$

Let $p \neq q$.
Suppose $p.x \leqslant q.x$ and $p.y \leqslant q.y$.
Let $r_0, r_1, \ldots, r_n$ be a path in $f[p,q]$ from p to q.
For $i \in \mathbf{N}, 0 \leqslant i \leqslant n$, we define $P(r_i)$ to be the intersection point of $[\![p,q]\!]$ with the line through r_i defined by

$$y + x = r_i.y + r_i.x.$$

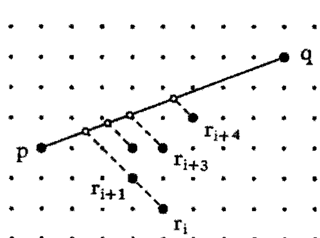
Then, according to the above property,

$$\tilde{d}(r_i, [\![p,q]\!]) = d(r_i, P(r_i)),$$

and, for $P(r_i) \in [p,q]$,

$$\tilde{d}(r_i, [p,q]) = d(r_i, P(r_i)). \tag{b}$$

Because $r_0, r_1, \ldots, r_n$ is a path from p to q, $[p,q]$ is partitioned into segments $[s_0, s_1], \cdots, [s_{k-1}, s_k]$, where $s_0 = p$, $s_k = q$, and for all i, $0 \leqslant i < k$, $s_i = P(r_j)$ for some $j : 0 \leqslant j \leqslant n$. Note that for $i \neq j$, $P(r_i)$ may equal $P(r_j)$, hence k may be less than n.
The distance of $P(r_j)$ and $P(r_{j+1})$ is maximal if $r_{j+1} - r_j = (1,1)$ or if $r_{j+1} - r_j = (-1,-1)$, as is illustrated in the figure below, where all possible relative positions of two neighbours occur.



Define $\Delta x := q.x - p.x$ and $\Delta y := q.y - p.y$, and assume, without loss of generality, that $\Delta y \leqslant \Delta x$.
If $r_{j+1} - r_j = (1,1)$ or $r_{j+1} - r_j = (-1,-1)$, then we may derive the following (see also the figure below).

$d(P(r_j), P(r_{j+1}))$

$=$ $\{ s = (P(r_{j+1}).x, P(r_j).y) \}$

$d(P(r_j), s)$

$=$ $\{ [\![r_j, P(r_j)]\!]$ and $[\![r_{j+1}, P(r_{j+1})]\!]$ both have slope -1, and $[\![p, q]\!]$ has slope α, and definition of s, and geometry $\}$

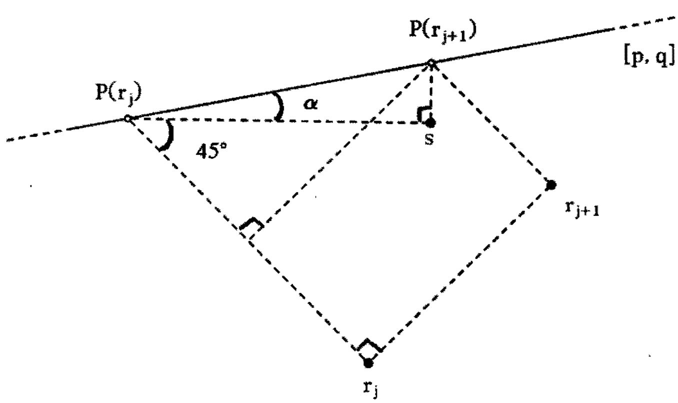
$\cos(\alpha) * \dfrac{\sqrt{2}}{\sin(\frac{1}{4}\pi + \alpha)}$

$=$ $\{ \Delta x * \sin(\alpha) = \Delta y * \cos(\alpha) \}$

$\dfrac{2\Delta x}{\Delta x + \Delta y}$

$\leqslant$ $\{ 0 \leqslant \Delta y \leqslant \Delta x \}$

2.



Hence, for all i, $0 \leqslant i < k$,

$$d(s_i, s_{i+1}) \leqslant 2. \qquad \text{(c)}$$

Let $v \in [p, q]$. Then $v \in [s_i, s_{i+1}]$ for some i, $0 \leqslant i < k$. From (c) and Property 2.1, it follows that

$$d(v, s_i) \leqslant 1 \vee d(v, s_{i+1}) <= 1.$$

Suppose, without loss of generality, that

$$d(v, s_i) \leqslant 1. \tag{d}$$

Let r_j be the pixel that satisfies

$$s_i = P(r_j). \tag{e}$$

Then

$\tilde{d}(v, f[p, q])$

$\leqslant$ { definition $\tilde{d}$ and $r_j \in f[p, q]$ }

$d(v, r_j)$

$\leqslant$ { triangle inequality for d }

$d(v, s_i) + d(s_i, r_j)$

$\leqslant$ { (d) and (e) }

$1 + d(r_j, P(r_j))$

$=$ { (b) }

$1 + \tilde{d}(r_j, [p, q])$

$\leqslant$ { $e0_f[p, q] \geqslant \tilde{d}(r_j, [p, q])$ }

$1 + e0_f[p, q]$.

Since v was an arbitrary point of $[p, q]$, we have proven that for p, q such that $p.x \leqslant q.x$ and $p.y \leqslant q.y$,

$$e1_f[p, q] \leqslant 1 + e0_f[p, q].$$

The other cases may be dealt with in a similar way.

□

From the above lemma it follows that $e0_f[p, q]$ suffices to express the quality of a digitised line segment with regard to closeness.

It will turn out that for several line functions that are not close, $e0_f[p, q]$ depends on $d(p, q)$. Therefore we introduce the function $E_f : \mathbf{N} \to \mathbf{R}^+ \cup \{\infty\}$, which measures the maximal deviation if the distance of the endpoints is fixed. E_f is defined by

$$E_f(n) := (\underline{\sup}\, p, q : p, q \in \mathbf{Z}^2 \wedge d(p, q) = n : e0_f[p, q]).$$

E_f is called the *deviation function of* f.

Property 2.4:

If f is a close line function, then for all $n \in \mathbf{N}$,

$$E_f(n) \leqslant 1.$$

Proof:

Let f be a close digitisation function, $p, q \in \mathbf{Z}^2$, and $r \in f[p,q]$.
From condition (cd0) of close digitisations, it follows that $v \in [p,q]$ exists such that $d(r,v) < 1$. Hence, $\tilde{d}(r,[p,q]) < 1$, for any $r \in f[p,q]$. Since

$$e0_f[p,q] = (\underline{\max} r : r \in f[p,q] : \tilde{d}(r,[p,q])),$$

it follows that $e0_f[p,q] < 1$, for all $p,q \in \mathbf{Z}^2$. Hence, $E_f(n) \leqslant 1$.

□

In Section 2.4, expressions for the deviation of the example line functions that are not close will be derived.

Note that for translation invariant line functions f,

$$e0_f(r+p, r+q) = e0_f[p,q],$$

for all $p,q,r \in \mathbf{Z}^2$. For these functions $E_f(n)$ may be rephrased as

$$E_f(n) = (\underline{\max} q : q \in \mathbf{Z}^2 \wedge d(\underline{0}, q) = n : e0_f[\underline{0}, q]).$$

2.3 Minimality

In this section we introduce a new property of line functions, one that has to do with the number of elements in the digitisations.

2.3.0 Definition

Let f be a line function. Because of condition lf1 in the definition of line function, for all $p,q \in \mathbf{Z}^2$, $f[p,q]$ contains a path connecting p and q. From Property 1.1 we know that the length of this path is at least $d(p,q)$. Hence the following lemma holds.

Lemma 2.5:

For any line function f,

$$(\underline{\mathrm{A}} p,q : p,q \in \mathbf{Z}^2 : \# f[p,q] \geqslant d(p,q) + 1).$$

□

Line functions for which the above inequality may be replaced by an equality will be called *minimal*. Hence, a line function f is minimal iff

$$(\underline{A}\, p, q : p, q \in \mathbb{Z}^2 : \# f[p,q] = d(p,q) + 1).$$

For any minimal line function f, the elements of $f[p,q]$ may be uniquely arranged in a sequence π such that π is a path from $\hat{p}$ to $\hat{q}$. The length of this path is $d(p,q)$. Such paths are called *minimal paths*. In the following subsection we shall concentrate on minimal paths and their representation by means of chain codes.

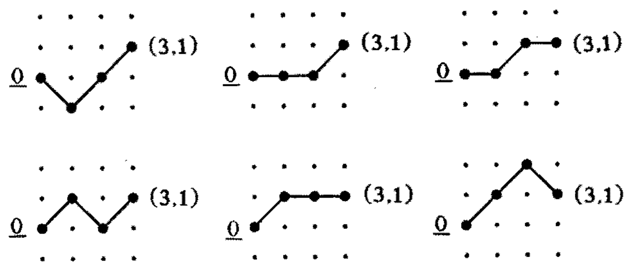
Line functions commonly used in Computer Graphics and Image Analysis are minimal. However, non-minimal line functions are also worth investigating: to support line functions in which anti-aliasing is incorporated, for instance, or to solve the decreasing intensity problem.

In Computer Graphics, anti-aliasing is a technique to reduce the staircase appearance (due to the discreteness of the grid) of digitised line segments. If pixels may be assigned an intensity value, this staircase effect may be reduced by assigning small intensity values to pixels 'in the neighbourhood' of the digitised line segment. (See [Foley & van Dam 1982], for instance.) Hence, assuming that line functions may be combined with functions that generate intensity values, anti-aliasing requires nonminimal line functions.

The decreasing intensity problem has to do with the following phenomenon. Consider for $n \in \mathbf{N}$ the $n+1$ line segments $[p,(p.x+n, p.y+m)]$, where $0 \leqslant m \leqslant n$. Any minimal line function will map these line segments in pixel sets containing $n+1$ elements each, whereas the length of the real segments increases from n to $\sqrt{2}n$ if m increases from 0 to n. For solving this decreasing intensity problem, one needs a line function for which the cardinality of these digitised line segments increases from n to $\sqrt{2}n$ if m increases from 0 to n. Such a function cannot be minimal.

2.3.1 Minimal paths and chain codes

A *minimal* path from p to q is a path of length $d(p,q)$. For any pair of pixels, at least one minimal path exists that connects these pixels. Below, all minimal paths from $p = \underline{0}$ to $q = (3,1)$ are shown. For the sake of clearness we have also drawn the connections between neighbours.



[Rosenfeld 1978] uses the name *geodesics* for minimal paths. He proves the following property.

Property 2.6:

For any minimal path $r_0, r_1, \ldots, r_n$, each subsequence $r_i, r_{i+1}, \ldots, r_j$, where $0 \leqslant i \leqslant j \leqslant n$, is a minimal path from r_i to r_j.

□

For given p and q, the value of $f[p,q]$, where f is a minimal line function, will be a subset of the set of all pixels that are elements of minimal paths from p to q. Therefore we introduce the notion *Minimal Path Set*, in short MPS, which is defined by

$$\mathrm{MPS}(p,q) := (\cup\, \pi : \pi \textit{ is a minimal path from } p \textit{ to } q : <\pi>).$$

The following properties of MPS hold.

Property 2.7:

$$\mathrm{MPS}(p,q) = \mathrm{MPS}(q,p)$$

Proof:

This follows from $<\pi> = <\pi^{-1}>$.

□

Property 2.8:

$$\mathrm{MPS}(p,q) = \{r \in \mathbf{Z}^2 \mid d(p,r) + d(r,q) = d(p,q)\}.$$

Proof:

The proof consists of two parts.

- First, we prove that

$$\text{MPS}(p,q) \subseteq \{r \in \mathbf{Z}^2 \mid d(p,r)+d(r,q)=d(p,q)\}.$$

Suppose $r \in \text{MPS}(p,q)$.
Then a minimal path $\pi = r_0, r_1, \ldots, r_n$ exists such that $r_0 = p$, $r_n = q$, $n = d(p,q)$, and $r \in <\pi>$. Let $r = r_i$. From Property 2.6 it follows that $r_0, r_1, \ldots, r_i$ is a minimal path from p to r, and $r_i, r_{i+1}, \ldots, r_n$ a minimal path from r to q. Hence

$$d(p,r) = i \text{ and } d(r,q) = n-i.$$

Therefore

$$d(p,r)+d(r,q) = i+(n-i) = n = d(p,q).$$

and consequently,

$$r \in \{r \in \mathbf{Z}^2 \mid d(p,r)+d(r,q)=d(p,q)\}.$$

- Next, we prove that

$$\text{MPS}(p,q) \supseteq \{r \in \mathbf{Z}^2 \mid d(p,r)+d(r,q)=d(p,q)\}.$$

Suppose $r \in \mathbf{Z}^2$ and $d(p,r)+d(r,q) = d(p,q)$.
Let $p_0, p_1, \ldots, p_i$ be a minimal path from p to r and $q_0, q_1, \ldots, q_j$ a minimal path from r to q. Then $i = d(p,r)$ and $j = d(r,q)$. Define π as the concatenation of the two paths:

$$\pi = p_0, p_1, \ldots, p_{i-1}, r, q_1, q_2, \ldots, q_j.$$

Then π is a path from p to q of length

$$i+j = d(p,r)+d(r,q) = d(p,q).$$

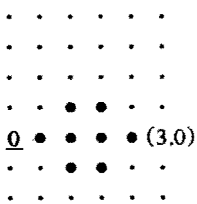
Hence π is a minimal path. Since $r \in <\pi>$, it follows that $r \in \text{MPS}(p,q)$.

We have now proven that

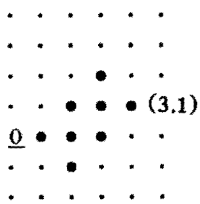
$$\text{MPS}(p,q) = \{r \in \mathbf{Z}^2 \mid d(p,r)+d(r,q)=d(p,q)\}.$$

□

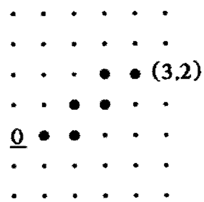
Some examples of Minimal Path Sets are given below.



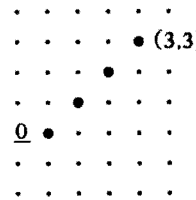
MPS($\underline{0}$,(3,0))



MPS($\underline{0}$,(3,1))



MPS($\underline{0}$,(3,2))

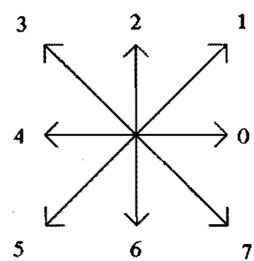


MPS($\underline{0}$,(3,3))

According to [Freeman 1970], each path may be associated with a chain code, which is an element of $\{0,1,2,3,4,5,6,7\}^*$, in the following way.

A path $\pi = r_0, r_1, \ldots, r_n$ has chain code $\gamma = c_1 c_2 \cdots c_n$, where

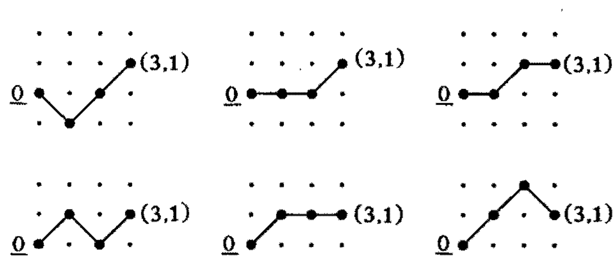
$$c_i = \begin{cases} 0 & \text{if } r_i - r_{i-1} = (1,0) \\ 1 & \text{if } r_i - r_{i-1} = (1,1) \\ 2 & \text{if } r_i - r_{i-1} = (0,1) \\ 3 & \text{if } r_i - r_{i-1} = (-1,1) \\ 4 & \text{if } r_i - r_{i-1} = (-1,0) \\ 5 & \text{if } r_i - r_{i-1} = (-1,1) \\ 6 & \text{if } r_i - r_{i-1} = (0,-1) \\ 7 & \text{if } r_i - r_{i-1} = (1,-1). \end{cases}$$



For the relation between a chain code element i, $i \in \{0,1,2,3,4,5,6,7\}$, and the vector $r_i - r_{i-1}$ (see the figure above) we will use the function ν. Hence $\nu(0) = (1,0)$, $\nu(1) = (1,1)$, and so forth.

As an example, the minimal paths from $p = \underline{0}$ to $q = (3,1)$ are shown again, now accompanied by their chain codes.

711 001 010



171 100 117

A chain code $\gamma = c_1 c_2 \cdots c_n$ is said to have length n. We shall write $\gamma[i]$ for c_i $(0 < i \leqslant n)$, and $\gamma[i:j]$ for $c_i c_{i+1} \cdots c_j$. On the analogy of paths, $<\gamma>$ will denote the set of elements of γ, i.e., $<\gamma> = \{c_i \mid 0 < i \leqslant n\}$. $<\gamma>$ is called the *alphabet* of γ. If a path π has chain code γ, then the chain code of π^{-1} will be referred to as γ^{-1}.

From the above definitions the following property may be derived.

Property 2.9:

For any path $r_0, r_1, \ldots, r_n$ with chain code γ, and all $i : 0 \leqslant i \leqslant n$,

$$r_i = r_0 + \sum_{j=1}^{i} v(\gamma[j]).$$

□

[Rosenfeld 1978] proves that a path $\pi = r_0, r_1, \ldots, r_n$ is minimal if and only if

$$r_0.x < r_1.x < \ldots < r_n.x \ \vee \ r_0.x > r_1.x > \ldots > r_n.x \ \vee$$

$$r_0.y < r_1.y < \ldots < r_n.y \ \vee \ r_0.y > r_1.y > \ldots > r_n.y.$$

Since for any path $\pi = r_0, r_1, \ldots, r_n$ holds that

$$r_i.x - r_{i-1}.x \in \{-1,0,1\} \ \wedge \ r_i.y - r_{i-1}.y \in \{-1,0,1\},$$

and that π is minimal iff $d(r_0, r_n) = n$, the following property holds.

Property 2.10:

For any minimal path $\pi = r_0, r_1, \ldots, r_n$ holds

$$(r_0.x < r_1.x < \ldots < r_n.x \iff r_n.x - r_0.x = n) \ \wedge$$

$$(r_0.x > r_1.x > \ldots > r_n.x \iff r_0.x - r_n.x = n) \ \wedge$$

$$(r_0.y < r_1.y < \ldots < r_n.y \iff r_n.y - r_0.y = n) \ \wedge$$

$$(r_0.y > r_1.y > \ldots > r_n.y \iff r_0.y - r_n.y = n).$$

□

This property may be expressed in terms of chain codes:

Property 2.11:

For the chain code γ of any minimal path $r_0, r_1, \ldots, r_n$ holds

$$(<\gamma> \subseteq \{7,0,1\} \iff r_n.x - r_0.x = n) \ \wedge$$

$$(<\gamma> \subseteq \{3,4,5\} \iff r_0.x - r_n.x = n) \ \wedge$$

$$(<\gamma> \subseteq \{1,2,3\} \Leftrightarrow r_n.y - r_0.y = n) \wedge$$
$$(<\gamma> \subseteq \{5,6,7\} \Leftrightarrow r_0.y - r_n.y = n).$$

□

Note that the conjuncts of Property 2.11 are not mutual exclusive. If, for instance,

$$r_n.x - r_0.x = r_n.y - r_0.y = n,$$

then

$$<\gamma> \subseteq \{7,0,1\} \wedge <\gamma> \subseteq \{1,2,3\},$$

hence $<\gamma> \subseteq \{1\}$.

We now know from Property 2.11 that the alphabet of the chain code of a minimal path contains at most 3 elements. For p and q with $|q.y - p.y| \leqslant q.x - p.x = d(p,q)$ for example, each minimal path from p to q has a chain code whose alphabet is a subset of $\{7,0,1\}$. The following lemmas express the necessary and sufficient conditions for an element of $\{0,1,2,3,4,5,6,7\}^*$ to be the chain code of a minimal path.

Lemma 2.12:

Let p and q be pixels such that $d(p,q) = n = q.x - p.x$.
Let σ be an element of $\{7,0,1\}^*$ of length n.
Then σ is the chain code of a minimal path from p to q if and only if

$$N_\sigma(1) - N_\sigma(7) = q.y - p.y. \qquad \text{(a)}$$

Proof:

- First we prove that condition (a) implies that σ is the chain code of a minimal path from p to q.
  Let σ be an element of $\{7,0,1\}^*$ of length n, and let $N_\sigma(1) - N_\sigma(7) = q.y - p.y$.
  Define the path $\pi = r_0, r_1, \ldots, r_n$ by

  $$r_0 := p,$$
  $$r_i := r_{i-1} + v(\sigma[i]), \quad 0 < i \leqslant n.$$

  Then π is a path from p to r_n of length n and with chain code σ. We shall prove that $r_n = q$.

  r_n

  $=$ { definition of r_i }

  $p + \sum_{i=1}^{n} v(\sigma[i])$

  $=$ { $\sigma \in \{7,0,1\}^*$ and $v(7) = (1,-1)$, $v(0) = (1,0)$, $v(1) = (1,1)$ }

$$(p.x + n\ ,\ p.y + N_\sigma(1) - N_\sigma(7))$$

$= \qquad \{\ n = q.x - p.x \text{ and (a) }\}$

q.

Thus π is a minimal path from p to q, and its chain code is σ.

- Now we prove that the chain code of any minimal path from p to q satisfies condition (a).
  Suppose $\pi = r_0, r_1, \ldots, r_n$ is a minimal path from p to q. Let γ be the chain code associated with this path.
  From Property 2.11 it follows that $\gamma \in \{7,0,1\}^*$. Then

$q.y - p.y$

$= \qquad \{\ q = p + \sum_{i=1}^{n} v(\gamma[i])\ \}$

$\sum_{i=1}^{n} v(\gamma[i]).y$

$= \qquad \{\ \gamma \in \{7,0,1\}^* \text{ and } v(7).y = -1,\ v(0).y = 0,\ v(1).y = 1\ \}$

$N_\gamma(1) - N_\gamma(7)$.

This completes the proof for p and q such that $d(p,q) = n = q.x - p.x$.

□

The next three lemmas may be proven in a similar way.

Lemma 2.13:

Let p and q be pixels such that $d(p,q) = n = p.x - q.x$.
Let σ be an element of $\{3,4,5\}^*$ of length n.
Then σ is the chain code of a minimal path from p to q if and only if

$$N_\sigma(3) - N_\sigma(5) = q.y - p.y.$$

□

Lemma 2.14:

Let p and q be pixels such that $d(p,q) = n = q.y - p.y$.
Let σ be an element of $\{1,2,3\}^*$ of length n.
Then σ is the chain code of a minimal path from p to q if and only if

$$N_\sigma(1) - N_\sigma(3) = q.x - p.x.$$

□

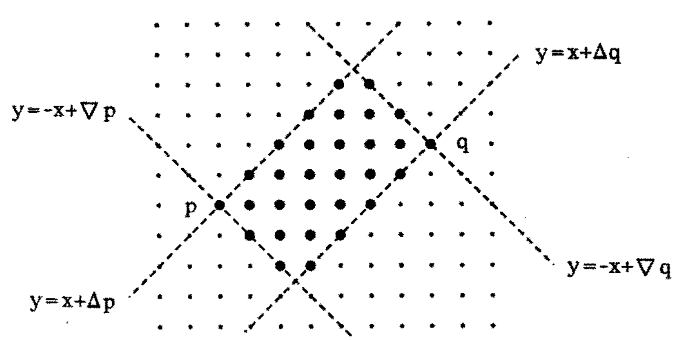
Lemma 2.15:

Let p and q be pixels such that $d(p,q) = n = p.y - q.y$.
Let σ be an element of $\{5,6,7\}^*$ of length n.
Then σ is the chain code of a minimal path from p to q if and only if

$$N_\sigma(7) - N_\sigma(5) = q.x - p.x.$$

□

The above lemmas can be used to state something about the shape of the Minimal Path Set MPS(p,q). It turns out that MPS(p,q) consists of all pixels within the box that has p and q as opposite vertices and whose edges are parallel to the diagonal lines $y = x$ and $y = -x$. This is expressed in the following lemma. For convenience we firstly introduce some shorthands: if p is a pixel, then Δp will stand for $p.y - p.x$, and ∇p for $p.y + p.x$. In the figure below the meaning of these notations is illustrated.



Lemma 2.16:

$$\mathrm{MPS}(p,q) = \{ r \in \mathbf{Z}^2 \mid \min(\Delta p, \Delta q) \leqslant \Delta r \leqslant \max(\Delta p, \Delta q) \wedge \min(\nabla p, \nabla q) \leqslant \nabla r \leqslant \max(\nabla p, \nabla q) \}.$$

Proof:

Let p and q be pixels, and let $n = d(p,q)$.
Suppose $q.x - p.x = n$.
Then $|q.y - p.y| \leqslant q.x - p.x$, and consequently

$$\Delta q \leqslant \Delta p \wedge \nabla p \leqslant \nabla q.$$

Therefore, the set

$$\{r \in \mathbf{Z}^2 \mid \min(\Delta p, \Delta q) \leqslant \Delta r \leqslant \max(\Delta p, \Delta q) \wedge$$
$$\min(\nabla p, \nabla q) \leqslant \nabla r \leqslant \max(\nabla p, \nabla q)\}$$

can be expressed more simply as

$$\{r \in \mathbf{Z}^2 \mid \Delta q \leqslant \Delta r \leqslant \Delta p \wedge \nabla p \leqslant \nabla r \leqslant \nabla q\},$$

for which we will use the abbreviation DB (of Diagonal Box).

- First, we shall prove that $\mathrm{MPS}(p,q) \subseteq \mathrm{DB}$.
  Suppose $r \in \mathrm{MPS}(p,q)$.
  Then a minimal path from p to q exists that contains r. Let π be such a path, and let γ denote its chain code. Then

$$q = p + \sum_{i=1}^{n} v(\gamma[i]) \tag{1}$$

From Property 2.6 we know that $\gamma[1{:}d(p,r)]$ is the chain code of a minimal path from p to r, hence

$$r = p + \sum_{i=1}^{d(p,r)} v(\gamma[i]), \tag{2}$$

and combined with (1) this leads to

$$r = q - \sum_{i=d(p,r)+1}^{n} v(\gamma[i]). \tag{3}$$

Since, according to Property 2.11, $\gamma[i] \in \{7,0,1\}$, and $v(7)=(1,-1)$, $v(0)=(1,0)$, and $v(1)=(1,1)$,

$$\sum_{i=1}^{d(p,r)} v(\gamma[i]).x = d(p,r)$$

and

$$-d(p,r) \leqslant \sum_{i=1}^{d(p,r)} v(\gamma[i]).y \leqslant d(p,r),$$

hence,

$$(\Delta r - \Delta p).y = \sum_{i=1}^{d(p,r)} v(\gamma[i]).y - \sum_{i=1}^{d(p,r)} v(\gamma[i]).x \leqslant 0 \tag{4}$$

and

$$(\nabla r - \nabla p).y = \sum_{i=1}^{d(p,r)} v(\gamma[i]).y + \sum_{i=1}^{d(p,r)} v(\gamma[i]).x \geqslant 0. \tag{5}$$

(4) and (5) combined with (2) lead to

$$\Delta r \leqslant \Delta p, \quad \nabla r \geqslant \nabla p. \tag{6}$$

For

$$\sum_{i=d(p,r)+1}^{n} v(\gamma[i]).$$

inequalities similar to (4) and (5) may be derived, and these combined with (3) lead to the inequalities

$$\Delta r \geqslant \Delta q, \quad \nabla r \leqslant \nabla q. \tag{7}$$

From (6) and (7) it follows that $r \in \mathrm{DB}$, and hence $\mathrm{MPS}(p,q) \subseteq \mathrm{DB}$.

- Now we shall prove that $\mathrm{DB} \subseteq \mathrm{MPS}(p,q)$.
  Suppose $r \in \mathrm{DB}$.
  From the definition of DB it follows that

  $$r.x - p.x \geqslant |p.y - r.y| \ \wedge \ q.x - r.x \geqslant |q.y - r.y|.$$

  This implies

  $$d(p,r) + d(r,q) = (r.x - p.x) + (q.x - r.x) = q.x - p.x = d(p,q).$$

  With Property 2.8 it then follows that $r \in \mathrm{MPS}(p,q)$.

The lemma is now proven for points p and q with $d(p,q) = q.x - p.x$. The cases in which $d(p,q) = q.y - p.y$, $d(p,q) = p.y - q.y$, and $d(p,q) = p.x - q.x$ may be proven in a similar way.

□

As said before, minimal line functions generate pixel sets that are associated with minimal paths. If f is a minimal line function, then $c(f[p,q])$ (or $c \circ f[p,q]$) will denote the chain code associated with the minimal path from $\hat{p}$ to $\hat{q}$. From the definition of chain codes the following property may be derived.

Property 2.17:

For all minimal line functions f and all $p,q,r \in \mathbf{Z}^2$ holds

$$c(r \oplus f[p,q]) = c \circ f[p,q].$$

□

Corollary 2.18:

For all minimal line functions f that are translation invariant, and all $p,q,r \in \mathbf{Z}^2$ holds

$$c \circ f[r+p, r+q] = c \circ f[p, q].$$

□

2.4 Examples of line functions

2.4.0 Introduction

In this section we present several examples of line functions. All line functions will be discussed with respect to the properties translation invariance, minimality, convexity and closeness.

In order to prove that the functions presented are indeed line functions, we have to prove that they satisfy conditions lf0 and lf1 of the definition of line functions. These conditions are

lf0) $(\underline{\mathrm{A}}\, p, q : p, q \in \mathbf{Z}^2 : \{p, q\} \subseteq f[p, q])$

lf1) $(\underline{\mathrm{A}}\, p, q : p, q \in \mathbf{Z}^2 : f[p, q]$ is finite and connected $)$.

We shall use the following auxiliary sets and functions.
The set $O_0 \subseteq \mathbf{R}^2$ is defined by

$$O_0 := \{(x, y) \in \mathbf{R}^2 \mid 0 \leqslant y \leqslant x\}.$$

The functions $f_i : O_0 \rightarrow \boldsymbol{P}(\mathbf{R}^2)$, where $i \in \{0, \ldots, 7\}$, are defined by

$$f_0(p) := p, \qquad f_1(p) := (p.y, p.x),$$

$$f_2(p) := (-p.y, p.x), \qquad f_3(p) := (-p.x, p.y),$$

$$f_4(p) := (-p.x, -p.y), \qquad f_5(p) := (-p.y, -p.x),$$

$$f_6(p) := (p.y, -p.x), \qquad f_7(p) := (p.x, -p.y).$$

The functions f_i are generalised to subsets of O_0 in the usual way: for $P \subseteq O_0$,

$$f_i(P) := \{f_i(p) \mid p \in P\}.$$

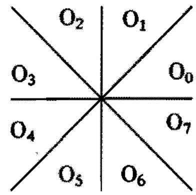
Note that the functions f_i are combinations of reflections in the $y=x$, $y=0$, and $x=0$ axes, and hence each f_i preserves distance and straightness, that is,

$$d(f_i(p), f_i(q)) = d(p, q),$$

$$f_i[p, q] = [f_i(p), f_i(q)].$$

The octants O_i, where $i \in \{0, \ldots, 7\}$, are defined by

$$O_i \ := \ f_i(O_0).$$



Then

$$\mathbf{R}^2 = (\cup i : i \in \{0,\dots,7\} : O_i).$$

Alongside, the sets O_i are illustrated.

To investigate the suitability of the nonminimal line functions, presented in the following subsections, to solve the decreasing intensity problem, we shall give expressions for the cardinality of $f[p,q]$.

2.4.1 The Bounding Box function

The Bounding Box function $f: \mathbf{Z}^2 \times \mathbf{Z}^2 \rightarrow \boldsymbol{P}(\mathbf{Z}^2)$ is defined by

$$f[p,q] \ := \ \{(x,y) \in \mathbf{Z}^2 \mid \min(p.x, q.x) \leqslant x \leqslant \max(p.x, q.x) \ \wedge \\ \min(p.y, q.y) \leqslant y \leqslant \max(p.y, q.y)\}.$$

In the sequel, the set

$\{(x,y) \in \mathbf{Z}^2 \mid \min(p.x, q.x) \leqslant x \leqslant \max(p.x, q.x) \ \wedge \ \min(p.y, q.y) \leqslant y \leqslant \max(p.y, q.y)\}$

will be referred to as BB(p,q). In Figure 2.0, the function value $f[\underline{0}, (7,5)]$ is shown.

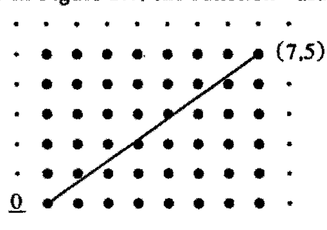


Figure 2. 0

$f[\underline{0}, (7,5)]$, *where f is the Bounding Box function.*

From the definition of the Bounding Box function the following properties may be derived.

Property 2.19:

The Bounding Box function is a line function.

□

Property 2.20:

For the Bounding Box function f,

$$\# f[p,q] = (|q.x - p.x|+1)*(|q.y-p.y|+1).$$

□

Property 2.21:

The Bounding Box function is translation invariant.

□

Property 2.22:

The Bounding Box function is convex.

□

From the second property it follows immediately that f is not a minimal line function; however, it does not solve the decreasing intensity problem, since for fixed p and $q.x = p.x + n$, the number of elements in $f[p,q]$ increases from $n+1$ to $(n+1)^2$ if $q.y$ increases from $p.y$ to $p.y+n$.

From Figure 2.0 it can be seen that the Bounding Box function is not close. In the following property we give an expression for its deviation; in the proof we use that f is translation invariant.

Property 2.23:

For the Bounding Box function f,

$$E_f(n) = \frac{n}{2}.$$

Proof:

Recall that

$$E_f(n) = (\underline{\max} q : q \in \mathbf{Z}^2 \wedge d(\underline{0}, q) = n : eO_f[\underline{0}, q]),$$

where

$$eO_f[\underline{0}, q] = (\underline{\max} r : r \in f[\underline{0}, q] : \tilde{d}(r, [\underline{0}, q])).$$

In Figure 2.1(a) it can be see that for $q \in O_0 \cup O_7$,

$$eO_f[\underline{0}, q] = (\underline{\max} r : r \in f[\underline{0}, q] : \tilde{d}(r, [\underline{0}, q])) = \tilde{d}(s, [\underline{0}, q]),$$

where $s = (q.x, 0)$. With $a = \tilde{d}(s, [\underline{0}, q])$, a can be computed, using Property 2.2, from the equation (see Figure 2.1(b))

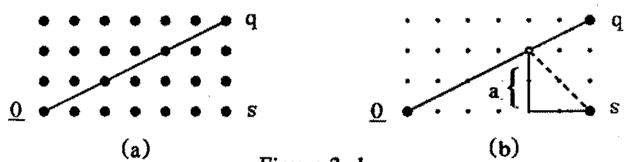


Figure 2. 1

a) pixel $s \in f[\underline{0}, q]$ has maximal distance to $[\underline{0}, q]$
b) $a = \tilde{d}(s, [p, q])$.

$$a : (q.x - a) = |q.y| : q.x.$$

It follows that

$$\tilde{d}(s, [\underline{0}, q]) = \frac{|q.y| * q.x}{|q.y| + q.x}.$$

Similar equations may be derived if $q \notin O_0 \cup O_7$.
Thus, the deviation function satisfies

$$E_f(n) = (\underline{\max} k : k \in [0..n] : \frac{kn}{k+n})$$
$$= \frac{n}{2}.$$

□

Consequently, the deviation of the Bounding Box function is linear in $d(p, q)$.

2.4.2 The Minimal Path Set function

The Minimal Path Set function $f : \mathbf{Z}^2 \times \mathbf{Z}^2 \rightarrow \mathcal{P}(\mathbf{Z}^2)$ is defined by

$$f[p, q] := \text{MPS}(p, q).$$

In Figure 2.2 the function value $f[\underline{0}, (7,5)]$ is shown.

Using Lemma 2.16, the following property on the numbers of pixels in $f[p, q]$ may be proven.

Property 2.24:

For the Minimal Path Set function f,

$$\# f[p, q] = \lceil \tfrac{1}{2}(|\nabla q - \nabla p| + 1) * (|\Delta q - \Delta p| + 1) \rceil.$$

□

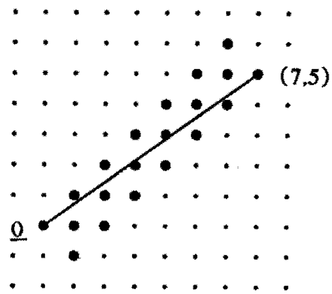


Figure 2. 2

$f[\underline{0},(7,5)]$, *where* f *is the Minimal Path Set function.*

Property 2.25:

The Minimal Path Set function is a line function.

Proof:

lf0) Since any path connecting p and q contains p and q, it follows that p and q are contained in MPS(p,q).

lf1) From the above property it follows that MPS(p,q) is finite.
Let $r,s \in$ MPS(p,q). From the definition of MPS(p,q) it follows that a minimal path π_0 exists that contains r. Since $<\pi_0> \subseteq$ MPS(p,q), a path in MPS(p,q) exists from p to r. In the same way a path π_1 in MPS(p,q) exists from p to s. The concatenation of π_0 and π_1^{-1} is a path in MPS(p,q) from r to s.
Hence, MPS(p,q) is connected.

□

We shall now prove that the Minimal Path Set function is translation invariant and convex.

Property 2.26:

The Minimal Path Set function is translation invariant.

Proof:

$f[r+p,r+q]$

$=$ { Property 2.8 }

$\{s \in \mathbf{Z}^2 \mid d(r+p,s)+d(s,r+q) = d(r+p,r+q)\}$

$=\quad$ { Property 1.0 }

$\{s \in \mathbf{Z}^2 \mid d(p, s-r) + d(s-r, q) = d(p, q)\}$

$=\quad$ { renaming dummy variable s }

$\{r+t \in \mathbf{Z}^2 \mid d(p, t) + d(t, q) = d(p, q)\}$

$=\quad$ { definition of $\oplus$ and Property 2.8 and definition of f }

$r \oplus f[p, q]$.

□

Property 2.27:

The Minimal Path Set is convex.

Proof:

Suppose $r, s \in \mathrm{MPS}(p, q)$. Then, according to Lemma 2.16,

$$\min(\Delta p, \Delta q) \leqslant \Delta r \leqslant \max(\Delta p, \Delta q) \ \wedge \tag{1}$$

$$\min(\Delta p, \Delta q) \leqslant \Delta s \leqslant \max(\Delta p, \Delta q).$$

Similarly, for any $t \in \mathrm{MPS}(r, s)$,

$$\min(\Delta r, \Delta s) \leqslant \Delta t \leqslant \max(\Delta r, \Delta s). \tag{2}$$

Combining (1) and (2) gives us

$$\min(\Delta p, \Delta q) \leqslant \Delta t \leqslant \max(\Delta p, \Delta q).$$

In the same way it may be shown that

$$\min(\nabla p, \nabla q) \leqslant \nabla t \leqslant \max(\nabla p, \nabla q).$$

Thus, $t \in \mathrm{MPS}(p, q)$, and we have proven that

$$\mathrm{MPS}(r, s) \subseteq \mathrm{MPS}(p, q).$$

Hence, the Minimal Path Set function is convex.

□

From Property 2.24 it follows that the Minimal Path Set function is not minimal. In fact, for $p = \underline{0}$ and $q.x = n$,

$$\# f[p, q] = \lceil \tfrac{1}{2}(|n + q.y| + 1) * (|n - q.y| + 1) \rceil. \tag{a}$$

The right hand side of (a) may be rewritten as

$\lceil \frac{1}{2}(n^2 + 2n - q.y^2 + 1) \rceil$.

Hence, the Minimal Path Set function offers no solution for the decreasing intensity problem, since if $q.y$ increases from 0 to n, then $\# f[p,q]$ decreases from $\lceil \frac{1}{2}(q.x+1)^2 \rceil$ to $q.x+1$, which is almost the opposite of what is required for an decreasing intensity solver.

From Figure 2.2 it can be seen that the Minimal Path Set function is not close. In the derivation of its deviation function we shall use that the Minimal Path Set function is translation invariant.

Property 2.28:

For the Minimal Path Set function f,

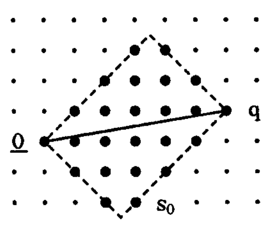
$$E_f(n) = \left\lfloor \frac{n}{2} \right\rfloor .$$

Proof:

Let $q \in O_0 \cup O_7$. Then it can be seen that

$$\begin{aligned} e\,0_f[\underline{0},q] &= (\underline{\max}\, r : r \in f[\underline{0},q] : \tilde{d}(r,[\underline{0},q])) \\ &= \tilde{d}(s,[\underline{0},q]), \end{aligned}$$

where s is one of the 'corners' of MPS(p,q) other than p and q. However, if $q.x + q.y$ is odd, these corners are not contained in $\mathbf{Z}^2$, as shown in the figure below, and in this case s_0 has maximal distance to $[p,q]$.



The following expression for $e\,0_f[\underline{0},q]$ may be derived.

$$e\,0_f[\underline{0},q] = \begin{cases} \dfrac{q.x - |q.y|}{2} & \text{if } q.x + |q.y| \text{ even} \\[2ex] \dfrac{q.x - |q.y| - 1}{2} + \dfrac{|q.y|}{q.x + |q.y|} & \text{otherwise.} \end{cases}$$

For cases in which $q \notin O_0 \cup O_7$, similar expressions may be derived. For the deviation function, which is defined by

$$E_f(n) := (\underline{\max}\, q : q \in \mathbf{Z}^2 \wedge d(\underline{0},q) = n : e\,0_f[\underline{0},q]),$$

it then follows that

$$E_f(n) = \left\lfloor \frac{n}{2} \right\rfloor .$$

□

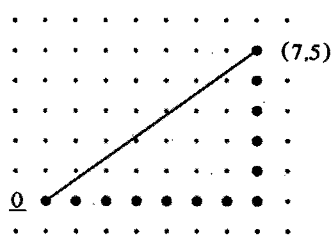
Thus, the deviation of the Minimal Path Set function is linear in $d(p,q)$, and equals, for even n, the deviation of the Bounding Box function.

2.4.3 The Franklin function

The function $g: O_0 \cup O_1 \cup O_6 \cup O_7 \rightarrow \boldsymbol{P}(O_0 \cup O_1 \cup O_6 \cup O_7)$ is defined by

$$g(p) := \{(i,0) \mid i \in [0..p.x]\} \cup \begin{cases} \{(p.x, j) \mid j \in [0..p.y]\} & \text{if } p.y \geqslant 0 \\ \{(p.x, j) \mid j \in [p.y..0]\} & \text{if } p.y < 0. \end{cases}$$

Below, the function value $g((7,5))$ is illustrated.



The function g may be considered as a digitisation function for line segments with one endpoint in the origin and the other endpoint contained in $O_0 \cup O_1 \cup O_6 \cup O_7$. The Franklin function $f: \mathbf{Z}^2 \times \mathbf{Z}^2 \rightarrow \boldsymbol{P}(\mathbf{Z}^2)$ is a translation invariant generalisation of g; it is defined by

$$f[p,q] := \hat{p} \oplus g(\hat{q} - \hat{p}).$$

Note that for any p and q,

$$\hat{q} - \hat{p} \in O_0 \cup O_1 \cup O_6 \cup O_7.$$

This function is suggested in [Franklin 1986] as an example of a convex line function. In [Luby 1986], this function is called the "square grid geometry".
The following properties of the Franklin function may be derived.

Property 2.29:

The Franklin function is a line function.

□

Property 2.30:

The Franklin function is translation invariant.

□

Property 2.31:

The Franklin function is convex.

□

See Figure 2.3 for some examples of digitised line segments of f.

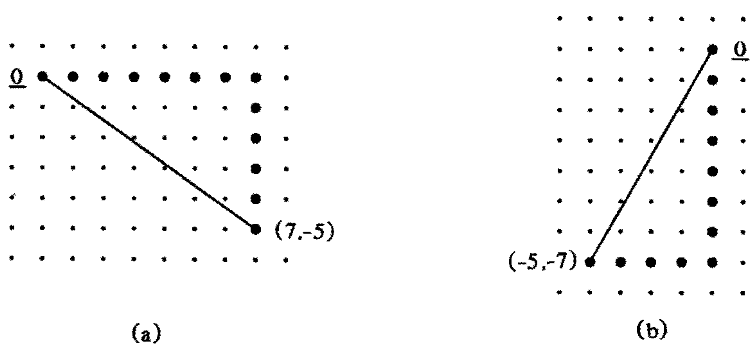


Figure 2. 3

Two digitised line segments of the Franklin function:

a) $f[\underline{0}, (7,-5)]$
b) $f[\underline{0}, (-5,-7)]$

As may be seen in Figure 2.3, the Franklin function is not minimal. From the definition of the Franklin function it follows that

$$\# f[\underline{0}, q] = q.x + |q.y| + 1.$$

Hence, if $q.y$ increases from 0 to $q.x$, then $\# f[\underline{0}, q]$ increases from $q.x + 1$ to $2q.x + 1$.

In Figure 2.3 it can be seen that the Franklin function is not close. The following can be stated on its deviation.

Property 2.32:

For the Franklin function f,

$$E_f(n) = \frac{n}{2}.$$

Proof:

Similar to the Bounding Box function, in Property 2.23.

□

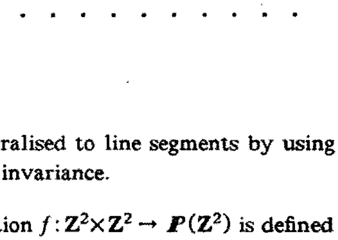
In the following section we shall introduce a line function that resembles the Franklin function, except that this new function is minimal, and has a somewhat better deviation.

2.4.4 The Adapted Franklin function

Again, we introduce an auxiliary function; the notation Δp is used for $p.x - p.y$. The function $g : O_0 \to \boldsymbol{P}(O_0)$ is defined by

$$g(p) := \{(i,0) \mid i \in [0..\Delta p]\} \cup \{(\Delta p + j, j) \mid j \in [0..p.y]\}.$$

Below, the function value $g((7,5))$ is shown.



The function g is now generalised to line segments by using the transformation functions f_i and imposing translation invariance.

The Adapted Franklin function $f : \mathbf{Z}^2 \times \mathbf{Z}^2 \to \boldsymbol{P}(\mathbf{Z}^2)$ is defined by

$$f[p,q] := \hat{p} \oplus f_i \circ g \circ f_i^{-1}(\hat{q} - \hat{p}).$$

where i is such that $\hat{q} - \hat{p} \in O_i$.

Property 2.33:

The Adapted Franklin function is a line function.

□

In Figure 2.4 some examples of digitised Adapted Franklin line segments are shown.

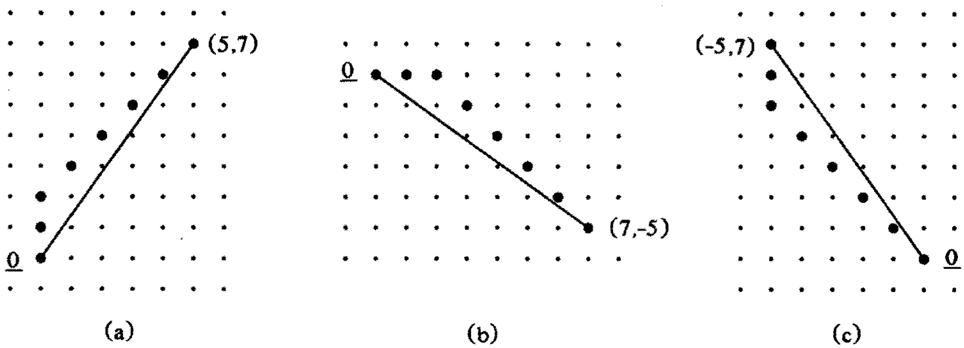


Figure 2. 4

Digitised line segments of the Adapted Franklin function:

a) $f[\underline{0},(5,7)]$
b) $f[\underline{0},(7,-5)]$
c) $f[\underline{0},(-5,7)]$

Like the Franklin function, the Adapted Franklin function is translation invariant and convex.

Property 2.34:

The Adapted Franklin function is translation invariant.

□

The proof that the Adapted Franklin function is convex is somewhat complicated, because of the occurrence of f_i in its definition.

Property 2.35:

The Adapted Franklin function is convex.

Proof:

Let f be the Adapted Franklin function, and suppose $p, q \in \mathbf{Z}^2$. Since f is translation invariant, we may assume that $p=\underline{0}$.
Furthermore, we assume, without loss of generality, NF($\underline{0}, q$). This implies that $q \in O_0 \cup O_1 \cup O_6 \cup O_7$. The cases $q \in O_0$ (case I) and $q \in O_1 \cup O_6 \cup O_7$ (case II) are dealt with separately.

I. Let $q \in O_0$. Then $f[\underline{0}, q] = g(q)$.
For $r, s \in f[\underline{0}, q]$, such that NF($r, s$), the following cases may be distinguished.

A) $r = (m, 0) \wedge s = (n, 0)$ with $0 \leqslant m \leqslant n \leqslant q.x - q.y$

B) $r = (m, 0) \wedge s = (q.x - q.y + n, n)$ with $0 \leqslant m < q.x - q.y \wedge 0 < n \leqslant q.y$

C) $r = (q.x - q.y + m, m) \wedge s = (q.x - q.y + n, n)$ with $0 \leqslant m \leqslant n \leqslant q.y$.

For each case it may be derived in a straightforward way that $f[r, s] \subseteq f[\underline{0}, q]$.

II. Suppose $q \in O_1 \cup O_6 \cup O\,7$. Since NF($\underline{0}, q$), for $q \in O_i$,

$$f[\underline{0}, q] = f_i \circ g \circ f_i^{-1}(q).$$

Using I, it may be proven that for $r, s \in f[\underline{0}, q]$ such that NF($f_i^{-1}(r), f_i^{-1}(s)$),

$$r \oplus f_i \circ g \circ f_i^{-1}(r - s) \subseteq f[\underline{0}, q]. \qquad \text{(a)}$$

By distinguishing the same cases as in I, it may then be derived that for $i = 1$ and $i = 7$,

$$\text{NF}(f_i^{-1}(r), f_i^{-1}(s)) \wedge f_i^{-1}(r), f_i^{-1}(s) \in g \circ f_i^{-1}(q) \Rightarrow \text{NF}(r, s). \qquad \text{(b)}$$

Hence, for $i = 1$ and $i = 7$,

$f[r, s]$

= { NF(r, s) and definition of f }

$r \oplus f_i \circ g \circ f_i^{-1}(r - s)$

$\subseteq$ { (a) }

$f[\underline{0}, q]$.

If $i = 6$, (b) holds for r, s such that $f_i^{-1}(r), f_i^{-1}(s)$ satisfy cases B and C (see I). Hence, in these cases

$$f[r, s] \subseteq f[\underline{0}, q].$$

For case A,

$$\text{NF}(f_i^{-1}(r), f_i^{-1}(s)) \wedge f_i^{-1}(r), f_i^{-1}(s) \in g \circ f_i^{-1}(q) \Rightarrow \text{NF}(s, r),$$

but in this case it may be proven that

$$r \oplus f_i \circ g \circ f_i^{-1}(r - s) = s \oplus f_i \circ g \circ f_i^{-1}(s - r),$$

which means that in this case too

$$f[r, s] \subseteq f[\underline{0}, q].$$

□

Unlike the Franklin function, the Adapted Franklin function is minimal, as is proven in the following property.

Property 2.36:

The Adapted Franklin function is minimal.

Proof:

For $p \in O_0$, $g(p)$ is defined by

$$g(p) := \{(i,0) \mid 0 \leqslant i \leqslant \Delta p\} \cup \{(\Delta p + j, j) \mid 0 \leqslant j \leqslant p.y\},$$

where $\Delta p = p.x - p.y$. Since

$$\{(i,0) \mid 0 \leqslant i \leqslant \Delta p\} \cap \{(\Delta p + j, j) \mid 0 \leqslant j \leqslant p.y\} = \{(\Delta p, 0)\},$$

the number of elements in $g(p)$ is $(\Delta p + 1) + (p.y + 1) - 1 = p.x + 1$. Since $p \in O_0$, $d(\underline{0}, p) = p.x$, hence

$$\#g(p) = d(\underline{0}, p) + 1.$$

Since the number of elements of a set is not changed by a translation, nor by a transformation f_i, it follows that for all p and q,

$$\# f[p,q] = d(p,q) + 1.$$

□

From Section 3 we know that a minimal line function generates minimal paths. These minimal paths may be represented by chain codes. The following lemma expresses what the chain codes of Adapted Franklin digitisations look like.

Lemma 2.37:

Let $\sigma \in \{0, \ldots, 7\}^*$.
Then σ is the chain code of an Adapted Franklin line segment iff

$$\sigma = 0^i 1^j \vee \sigma = 2^i 1^j \vee \sigma = 0^i 7^j \vee \sigma = 6^i 7^j \tag{a}$$

for some $i, j \in \mathbb{N}$.

Proof:

- First we shall prove that the chain code of any Adapted Franklin line satisfies (a). Suppose $p \in O_0$. The auxiliary function g for the Adapted Franklin line is defined by

  $$g(p) := \{(i,0) \mid 0 \leqslant i \leqslant \Delta p\} \cup \{(\Delta p + j, j) \mid 0 \leqslant j \leqslant p.y\},$$

  where $\Delta p = p.x - p.y$. From this it can be seen that the unique path from $\underline{0}$ to p in $g(p)$ has chain code

  $$0^{\Delta p} 1^{p.y}.$$

  The functions f_1, f_6, f_7 transform the set $g(p)$ into

$$\{(0,i) \mid 0 \leqslant i \leqslant \Delta p\} \cup \{(j, \Delta p + j) \mid 0 \leqslant j \leqslant p.y\},$$
$$\{(0,-i) \mid 0 \leqslant i \leqslant \Delta p\} \cup \{(j, -(\Delta p + j)) \mid 0 \leqslant j \leqslant p.y\},$$
$$\{(i,0) \mid 0 \leqslant i \leqslant \Delta p\} \cup \{(\Delta p + j, -j) \mid 0 \leqslant j \leqslant p.y\}$$

respectively, which results in chain codes

$$2^{\Delta p}\, 1^{p.y},\quad 6^{\Delta p}\, 7^{p.y},\quad 0^{\Delta p}\, 7^{p.y}$$

respectively. Since for all $p, q \in \mathbf{Z}^2$,

$$\hat{q} - \hat{p} \in O_0 \cup O_1 \cup O_6 \cup O_7,$$

it follows that the chain code of any Adapted Franklin line satisfies (a).

- Conversely, we shall prove that for any σ satisfying condition (a), p and q exist such that $\sigma = c \circ f[p,q]$.
  Suppose $\sigma \in \{0, \ldots, 7\}^*$ satisfies condition (a).
  If $\sigma = 0^i 1^j$, then the sequence $r_0, r_1, \ldots, r_{i+j}$ is defined by

$$r_k := \begin{cases} \underline{0} & k = 0 \\ \sum_{l=1}^{k} v(\sigma[l]) & 0 < k \leqslant i+j, \end{cases} \tag{b}$$

forms a path from r_0 to r_{i+j} with chain code σ. We shall prove that

$$\{r_k \mid 0 \leqslant k \leqslant i+j\} = f[\underline{0}, (i+j, j)],$$

which implies that σ is the chain code of an Adapted Franklin Line.

Since $v(0) = (1,0)$, $v(1) = (1,1)$, and $\sigma = 0^i 1^j$, it follows from (b) that

$$r_k = \begin{cases} (k, 0) & 0 \leqslant k \leqslant i \\ (k, k-i) & i \leqslant k \leqslant i+j. \end{cases} \tag{c}$$

Hence,

$$\{r_k \mid 0 \leqslant k \leqslant i+j\}$$

$=$ { (c) and definition of g for the Adapted Franklin function }

$$g((i+j, j))$$

$=$ { $(i+j, j) \in O_0$ and $f_0 = I$ }

$$f[\underline{0}, (i+j, j)].$$

From the first part of the proof we know that the functions f_1, f_6, f_7 transform $c \circ f[\underline{0}, (i+j, j)]$ into $2^i 1^j$, $6^i 7^j$, $0^i 7^j$ respectively. Hence, with $p := \underline{0}$ and

$$q := \begin{cases} (i+j,j) & \text{if } \sigma = 0^i 1^j \\ f_1(i+j,j) & \text{if } \sigma = 2^i 1^j \\ f_6(i+j,j) & \text{if } \sigma = 6^i 7^j \\ f_7(i+j,j) & \text{if } \sigma = 0^i 7^j, \end{cases}$$

$$c \circ f[p,q] = \sigma.$$

□

The Franklin function, though not minimal, also generates pixel sets of which the elements may be arranged in unique paths from $\hat{p}$ to $\hat{q}$. For the chain codes of these paths we state the following lemma, without proof.

Lemma 2.38:

Let $\sigma \in \{0,\dots,7\}^*$.
Then σ is the chain code of a Franklin line segment iff

$$\sigma = 0^i 2^j \ \vee \ \sigma = 0^i 6^j$$

for some $i, j \in \mathbf{N}$.

□

As can be seen in Figure 2.4, the Adapted Franklin function is not close. The following can be said on its deviation function.

Property 2.39:

For the Adapted Franklin function f,

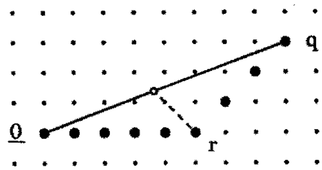
$$\frac{n}{6} \leqslant E_f(n) \leqslant (3-2\sqrt{2})n, \text{ for all } n \geqslant 3.$$

Proof:

Let f be the Adapted Franklin function, and $p, q \in \mathbf{Z}^2$. Since f is translation invariant, we may assume that $p = \underline{0}$.
Let $q \in O_0$. In the figure below it can be seen that

$$e\,0_f[\underline{0}, q] = \tilde{d}(r, [\underline{0}, q]),$$

where $r = (q.x - q.y, 0)$.



$a = \tilde{d}(r, [\underline{0}, q])$ can be computed from the equation

$$a : (q.x - q.y - a) = q.y : q.x;$$

it follows that

$$\tilde{d}(r, [\underline{0}, q]) = \frac{q.y\,(q.x - q.y)}{q.y + q.x}.$$

The deviation function E_f satisfies

$$E_f(n) = (\underline{\max}\, k : k \in [0..N] : \frac{k\,(n-k)}{k+n}).$$

Considered as a function on $\{ k \in \mathbf{R} \mid 0 \leqslant k \leqslant n \}$, E_f has maximum value $(3-2\sqrt{2})n$, for $k = (\sqrt{2}-1)n$. Furthermore, for $n \geqslant 3$ and $k = \lfloor \frac{1}{2} n \rfloor$, it may be shown that

$$\frac{k\,(n-k)}{k+n} \geqslant \frac{n}{6}.$$

□

Consequently, the deviation of the Adapted Franklin function is linear in $d(p,q)$, but smaller than the deviation of the Minimal Path Set, Bounding Box, and Franklin functions.

2.4.5 The Bresenham function

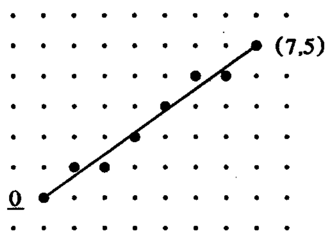
Before we introduce the following line function, we mention that for any $p \in O_0$, the line segment that connects $\underline{0}$ and p satisfies

$$[\underline{0}, p] = \{ (x, (^{p.y}/_{p.x})x) \in \mathbf{R}^2 \mid 0 \leqslant x \leqslant p.x \}.$$

Based on the above equation, we introduce the function $g : O_0 \rightarrow \boldsymbol{P}(O_0)$ defined by

$$g(p) := \{ (x, \lceil (^{p.y}/_{p.x})x - \tfrac{1}{2} \rceil) \in \mathbf{Z}^2 \mid 0 \leqslant x \leqslant p.x \}.$$

Hence, $g(p)$ contains from each column between 0 and $p.x$ the pixel that is most close to the intersection point of the line segment and that column. Below, the pixel set $g((7,5))$ is shown.



The function g is generalised in the same way as with the Adapted Franklin function. The Bresenham function $f : \mathbf{Z}^2 \times \mathbf{Z}^2 \to \mathbf{P}(\mathbf{Z}^2)$ is defined by

$$f[p,q] := \hat{p} \oplus f_i \circ g \circ f_i^{-1}(\hat{q}-\hat{p}),$$

where i is such that $\hat{q}-\hat{p} \in O_i$.
In [Bresenham 1965], as well as in [van Berckel & Mailloux 1965], algorithms are presented that generate the above pixel sets; for a proof see, for instance, [Bresenham 1985].

Property 2.40:

The Bresenham function is a line function.

Proof:

Let $p,q \in \mathbf{Z}^2$.

1f0) From the definition of g it follows that for all $r \in O_0$,

$$\{(0, \lceil -\tfrac{1}{2} \rceil), (r.x, \lceil ({}^{r.y}/_{r.x}) r.x - \tfrac{1}{2} \rceil)\} \subseteq g(r),$$

hence,

$$\{\underline{0}, r\} \subseteq g(r).$$

Analogously to the Adapted Franklin function, that is, the corresponding part of the proof of Property 2.33, it then follows that

$$\{p,q\} \subseteq f[p,q].$$

1f1) From the definition of g it follows that $g(p)$ is a finite set.
We shall now demonstrate that $g(p)$ is connected for any $p \in O_0$.

$$\lceil {}^{p.y}/_{p.x}(x+1) - \tfrac{1}{2} \rceil - \lceil ({}^{p.y}/_{p.x})x - \tfrac{1}{2} \rceil$$

$<$ $\{ (\underline{A} z : z \in \mathbf{R} : z - \tfrac{1}{2} \leqslant \lceil z - \tfrac{1}{2} \rceil < z + \tfrac{1}{2}) \}$

$${}^{p.y}/_{p.x}(x+1) + \tfrac{1}{2} - (({}^{p.y}/_{p.x})x - \tfrac{1}{2})$$

$=$ $\{$ arithmetic $\}$

$${}^{p.y}/_{p.x} + 1$$

$\leqslant \quad \{p \in O_0\}$

$2.$

Consequently,

$$\lceil {}^{p.y}\!/_{p.x}(x+1)-\tfrac{1}{2}\rceil - \lceil({}^{p.y}\!/_{p.x})x-\tfrac{1}{2}\rceil < 2,$$

and since both operands are elements of $\mathbf{Z}$,

$$\lceil {}^{p.y}\!/_{p.x}(x+1)-\tfrac{1}{2}\rceil - \lceil({}^{p.y}\!/_{p.x})x-\tfrac{1}{2}\rceil \leqslant 1.$$

Then, for any $x \in \mathbf{Z}$, $0 \leqslant x < p.x$,

$$d\left((x+1, \lceil {}^{p.y}\!/_{p.x}(x+1)-\tfrac{1}{2}\rceil), (x, \lceil({}^{p.y}\!/_{p.x})x-\tfrac{1}{2}\rceil)\right) \leqslant 1,$$

and this implies that $g(p)$ is connected.
Connectedness and finiteness are not violated by a transformation f_i, nor by a translation.

□

Like the Adapted Franklin function, the Bresenham function is translation invariant and minimal.

Property 2.41:

The Bresenham function is translation invariant.

□

Property 2.42:

The Bresenham function is minimal.

Proof:

For the Bresenham function, $g(p)$ is defined by

$$g(p) := \{(x, \lceil({}^{p.y}\!/_{p.x})x-\tfrac{1}{2}\rceil) \in \mathbf{Z}^2 \mid 0 \leqslant x \leqslant p.x\}.$$

Therefore, $\#g(p) = p.x+1 = d(\underline{0}, p)+1$, and consequently

$$\#f[p,q] = d(p,q)+1.$$

□

The Bresenham function is not convex: consider, for example, in Figure 2.5 the digitised line segments $f[\underline{0},(16,5)]$ and $f[(8,2),(14,4)]$. The latter one is not a subset of the former one, although $(8,2)$ and $(14,4)$ are both elements of the former one.

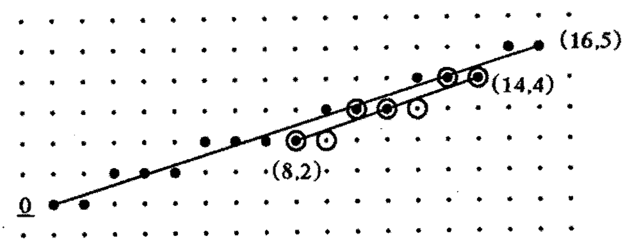


Figure 2. 5

Illustration that the Bresenham function is not convex. The elements of $f[\underline{0},(16,5)]$ and $f[(8,2),(14,4)]$ are indicated by • and ○ respectively.

Unlike the previously introduced line functions, the Bresenham function is close.

Property 2.43:

The Bresenham function is close.

Proof:

Let f be the Bresenham line function. We shall demonstrate that for $p,q \in \mathbf{Z}^2$, such that $p=\underline{0}$ and $q \in O_0$, $f[p,q]$ is a close digitisation of $[p,q]$, that is,

$$(\underline{A}r : r \in f[p,q] : (\underline{E}v : v \in [p,q] : d(r,v)<1)) \qquad \text{(a)}$$

$$(\underline{A}v : v \in [p,q] : (\underline{E}r : r \in f[p,q] : d(r,v)<1)). \qquad \text{(b)}$$

Recall that

$[\underline{0},q] = \{(x,(^{q.y}/_{q.x})x) \in \mathbf{R}^2 \mid 0 \leqslant x \leqslant q.x\}$ and

$f[\underline{0},q] = \{(x,\lceil(^{q.y}/_{q.x})x - \frac{1}{2}\rceil) \in \mathbf{Z}^2 \mid 0 \leqslant x \leqslant q.x\}$.

(a) Suppose $r \in f[p,q]$.
Then $r \in \mathbf{Z}^2$, $0 \leqslant r.x \leqslant q.x$, and

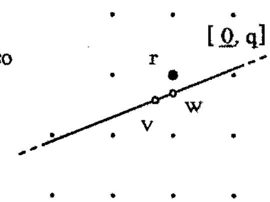
$$(^{q.y}/_{q.x})r.x - \tfrac{1}{2} \leqslant r.y < (^{q.y}/_{q.x})r.x + \tfrac{1}{2}.$$

Since $(r.x,(^{q.y}/_{q.x})r.x) \in [\underline{0},q]$, and $|\,(^{q.y}/_{q.x})r.x - r.y\,| \leqslant \frac{1}{2}$,

$$d(r,(r.x,(^{q.y}/_{q.x})r.x)) \leqslant \tfrac{1}{2} < 1,$$

hence condition (a) is satisfied.

(b) Suppose $v \in [\underline{0},q]$.
Define $x := \lceil v.x - \frac{1}{2}\rceil$, $w := (x,(^{q.y}/_{q.x})x)$, and $r := (x,\lceil(^{q.y}/_{q.x})x - \frac{1}{2}\rceil)$. Then

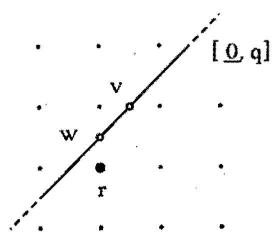


$w \in [\underline{0}, q]$, and $r \in f[\underline{0}, q]$.
Since $|x - v.x| \leqslant ½$, and $q.x \geqslant q.y$, also $|({}^{q.y}/_{q.x})x - v.y| \leqslant ½$, hence $d(v, w) \leqslant ½$.
Also $d(r, w) \leqslant ½$, hence

$$d(v, r) \leqslant d(v, w) + d(w, r) \leqslant ½ + ½ = 1.$$

If $d(v, r) < 1$, then condition (b) is satisfied.

Suppose $d(v, r) = 1$. Consequently, in this case $d(r, w) = d(w, v) = ½$. This is only possible in the following situation (recall that $a \leqslant \lceil a \rceil < a$),



where $v.y - w.y = v.x - w.x = ½$. Then

$${}^{q.y}/_{q.x}(x+1) = ({}^{q.y}/_{q.x})x + 1 \quad \wedge$$

$$\lceil {}^{q.y}/_{q.x}(x+1) - ½ \rceil = \lceil ({}^{q.y}/_{q.x})x - ½ \rceil + 1 = r.y + 1 = v.y.$$

Since for $s := (x+1, \lceil {}^{q.y}/_{q.x}(x+1) - ½ \rceil)$,

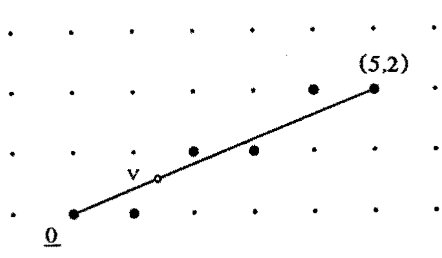
$$s \in f[\underline{0}, q] \quad \wedge \quad d(v, s) = ½,$$

condition (b) is satisfied in this case too.

From this it follows that $f[\underline{0}, q]$ is close.

□

f is not optimal, as is illustrated below: for $p = \underline{0}$ and $q = (5, 2)$, the point v, defined by $v = (1.4, 0.56)$, is an element of $[p, q]$, whereas no $r \in f[p, q]$ exists such that $d(r, v) \leqslant ½$.



2.4.6 The Close Embedding function

We now present a line function based on the digitisation function $f_{\vdash}$, which was defined in Chapter 1 by

$$f_{\vdash}(V) = \{p \in \mathbf{Z}^2 \mid (\underline{E}v : v \in V : d(p,v)<1)\}.$$

The Close Embedding function $f : \mathbf{Z}^2 \times \mathbf{Z}^2 \to \boldsymbol{P}(\mathbf{Z}^2)$ is defined by

$$f[p,q] := f_{\vdash}[p,q].$$

In Figure 2.6, the function value $f[\underline{0}, (7,5)]$ is shown.

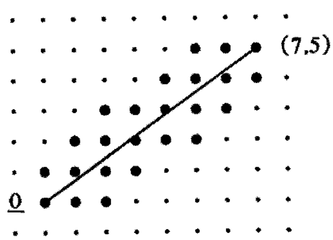


Figure 2. 6

$f[\underline{0}, (7,5)]$, *where f is the Close Embedding function.*

From Chapter 1 we know already that f is close (Property 1.13) and translation invariant (Property 1.15). We shall use that f is translation invariant in the proof of the following property.

Property 2.44:

The Close Embedding function is a line function.

Proof:

Let $p,q \in \mathbf{Z}^2$.
From the definition of $f_{\vdash}$ it follows that

$$f[p,q] = \{r \in \mathbf{Z}^2 \mid (\underline{E}v : v \in [p,q] : d(r,v)<1)\}.$$

lf0) Since $p,q \in [p,q]$, and $d(p,p)=d(q,q)=0$, $\{p,q\} \subseteq f[p,q]$.

lf1) Since $[p,q] \subseteq \mathrm{BB}(p,q)$, it follows that any $r \in f[p,q]$ is contained in $\mathrm{BB}(p,q)$. Consequently, $f[p,q]$ is finite.

We shall now prove that $f[p,q]$ is connected. Since f is translation invariant, we may assume that $p=\underline{0}$.
Let $r \in O_0$.
Then it can be seen that the Bresenham digitisation is a subset of the Close

Embedding digitisation:

$$\{(x, \lceil ({}^{r.y}/_{r.x})x - \tfrac{1}{2} \rceil) \in \mathbf{Z}^2 \mid 0 \leqslant x \leqslant r.x\} \subseteq f[\underline{0}, r].$$

We denote the intersection point of the Bresenham digitisation with column $x = i$ as r_i, and the intersection set of $f[\underline{0}, r]$ with $x = i$ as R_i. More formally, for $i \in \mathbf{N}$, $0 \leqslant i \leqslant r.x$,

$$r_i := (i, \lceil ({}^{r.y}/_{r.x})i - \tfrac{1}{2} \rceil), \text{ and}$$

$$R_i := \{s \in f[\underline{0}, r] \mid s.x = i\}.$$

Then it can be seen that

$$r_i \in R_i, \tag{a}$$

$$f[\underline{0}, r] = (\cup\ i : 0 \leqslant i \leqslant r.x : R_i), \tag{b}$$

and, from the proof that the Bresenham function is a line function, Property 2.40,

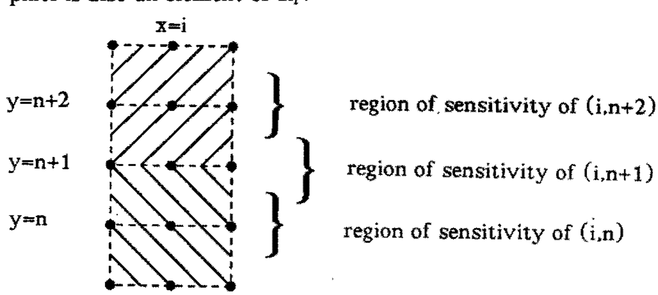
$$d(r_i, r_{i-1}) = 1 \quad \text{for all } i : 0 < i \leqslant r.x. \tag{c}$$

Furthermore, for all $i \in \mathbf{N}$, $0 \leqslant i \leqslant r.x$,

$$R_i \text{ is connected}, \tag{d}$$

as can be seen by the following argument.

Suppose $(i, n) \in R_i$ and $(i, n+2) \in R_i$. Then $v, w \in [p, q]$ exist such that $d(v, (i, n)) < 1$ and $d(w, (i, n+2)) < 1$. Since any line segment containing v and w passes through the region of sensitivity of the pixel $(i, n+1)$, this pixel is also an element of R_i.



This may be generalised to pixels (i, n) and $(i, n+j)$ where $j > 2$.

Hence, R_i is connected.

From (a), (b), (c), and (d) it follows that $f[\underline{0}, r]$ is connected. This holds for all $r \in O_0$, also for $f_i^{-1}(\hat{q} - \hat{p})$.

Let $r \in O_1 \cup O_6 \cup O_7$, and let i be such that $f_i^{-1}(r) \in O_0$. Then, according to the above part, $f[\underline{0}, f_i^{-1}(r)]$ is connected. Since it may be proven that

$$f[\underline{0},r] = f_i \circ f[\underline{0},f_i^{-1}(r)],$$

and connectedness is not violated by a transformation f_i, it follows that $f[\underline{0},r]$ is connected.

□

As can be seen in Figure 2.6, the Close Embedding function is not minimal. We shall derive an expression for the number of pixels contained in $f[p,q]$. We first introduce some new notions.

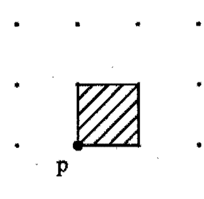
With each $p \in \mathbf{Z}^2$ we associate a subset of $\mathbf{R}^2$, notated as $S(p)$, and defined by

$$S(p) := \{(x,y) \in \mathbf{R}^2 \mid p.x \leqslant x \leqslant p.x+1 \wedge p.y \leqslant y \leqslant p.y+1\}.$$

Note that

$$\mathbf{R}^2 = (\cup p : p \in \mathbf{Z}^2 : S(p)).$$

A set $S(p)$ is called a unit square. Below, this set is illustrated.



Furthermore, we define the following subsets of $S(p)$.

$$Corner(S(p)) := \{p, p+(1,0), p+(1,1), p+(0,1)\},$$

and

$$\hat{S}(p) := \{(x,y) \in S(p) \mid p.x < x < p.x+1 \wedge p.y < y < p.y+1\}.$$

The elements of $\hat{S}(p)$ are called internal points of $S(p)$. Now it can be seen that for $p \neq q$,

$$f_{\vdash}[p,q] = (\cup r : r \in \mathbf{Z}^2 \wedge (\underline{E} v : v \in \hat{S}(r) : v \in [p,q]) : Corner(S(r))).$$

Property 2.45:

For the Close Embedding function f,

$$\# f[\underline{0},q] = 2(q.x + |q.y|) - \gcd(q.x, |q.y|) + 1, \qquad \text{(a)}$$

where $\gcd(m,n)$ denotes the greatest common divisor of m and n, and $\gcd(n,0)$ is supposed to equal n.

Proof:

The proof consists of three parts.

- Suppose $q.x = 0$.
  Then
  $$f[\underline{0}, q] = [\underline{0}, q] \cap \mathbf{Z}^2,$$
  and consequently
  $$\# f[\underline{0}, q] = |q.y| + 1.$$
  Since
  $$\begin{aligned} & 2(q.x + |q.y|) - \gcd(q.x, |q.y|) + 1 \\ = \; & 2|q.y| - |q.y| + 1 \\ = \; & |q.y| + 1, \end{aligned}$$
  equation (a) is satisfied.
  For reasons of symmetry, this is also true in case that $q.y = 0$.
- Suppose $q.x > 0$ and $|q.y| > 0$ and $\gcd(q.x, |q.y|) = 1$.
  Define V as the set of points in $[\underline{0}, q]$ that have an integer x- or y-coordinate:
  $$V \;:=\; \{(x, y) \in [\underline{0}, q] \mid x \in \mathbf{Z} \vee y \in \mathbf{Z}\}.$$
  Because $\gcd(q.x, |q.y|) = 1$, the only elements in V with both integer x- and y-coordinate are $\underline{0}$ and q, hence
  $$\# V = q.x + 1 + |q.y| + 1 - 2 = q.x + |q.y|.$$
  Let $r_0, r_1, \ldots, r_{q.x + |q.y| - 1}$ be the sequence of all elements of V, in increasing x-order, that is,
  $$r_0 = \underline{0} \wedge r_{q.x+|q.y|-1} = q \wedge (\underline{\mathbf{A}}\, i : 0 \leqslant i < q.x + |q.y| - 1 : r_i.x < r_{i+1}.x).$$
  The unit square that contains r_0 and r_1, contributes its four corners to the set $f[\underline{0}, q]$. If the line segment $[\underline{0}, q]$ is followed from $\underline{0}$ to q, then with each r_i, $0 < i < q.x + |q.y| - 1$, a new unit square is entered, which shares two of its corners with the previous unit square. Hence,
  $$\# f[\underline{0}, q] = 4 + 2(q.x + |q.y| - 2).$$
  Since $\gcd(q.x, |q.y|) = 1$, equation (a) is satisfied.
- Suppose $q.x > 0$ and $|q.y| > 0$ and $\gcd(q.x, |q.y|) > 1$.
  Let $n = \gcd(q.x, |q.y|)$. Define
  $$\begin{aligned} p_0 &:= \underline{0}, \\ p_i &:= i * ({}^{q.x}/_n, {}^{q.y}/_n), \text{ for } i : 0 < i \leqslant n. \end{aligned}$$
  Then $p_i \in [\underline{0}, q]$, and

$$(\underline{\mathbf{A}}\, i : 0 \leqslant i < n : \gcd(|p_{i+1}.x - p_i.x|, |p_{i+1}.y - p_i.y|) = 1), \text{ and} \tag{b}$$

$$[\underline{0}, q] = (\cup\, i : 0 \leqslant i < n : [p_i, p_{i+1}]). \tag{c}$$

Combining (c) with Property 1.8 implies

$$f[\underline{0}, q] = (\cup\, i : 0 \leqslant i < n : f[p_i, p_{i+1}]). \tag{d}$$

From (b) and the second part of this proof it follows that

$$\# f[p_i, p_{i+1}] = 2(\tfrac{q.x}{n} + \tfrac{|q.y|}{n}). \tag{e}$$

Furthermore,

$$f[p_{i-1}, p_i] \cap f[p_i, p_{i+1}] = \{p_i\}. \tag{f}$$

Combining (d), (e), and (f) results in

$$\begin{aligned} \# f[\underline{0}, q] &= n\left(\frac{2(q.x + |q.y|)}{n}\right) - (n-1) \\ &= 2(q.x + |q.y|) - n + 1. \end{aligned}$$

Hence, equation (a) is satisfied in this case too.

□

Note that if $q.x$ is fixed and prime, and $0 \leqslant q.y \leqslant q.x$,

$$\# f[\underline{0}, q] = \begin{cases} q.x + 1 & \text{if } q.y = 0 \\ 2q.x + 2q.y & \text{if } 0 < q.y < q.x \\ 3q.x + 1 & \text{if } q.y = q.x, \end{cases}$$

hence, if $q.y$ increases from 1 to $q.x$, then $\# f[\underline{0}, q]$ increases linearly in $q.y$, but discrepancies occur for $q.y = 0$ and $q.y = q.x$. This is due to the following property of $f_{\vdash}$.

$$\# f_{\vdash}(\{r\}) = \begin{cases} 1 & \text{if } r \in \mathbf{Z}^2 \\ 4 & \text{if } r \in \mathbf{R}^2 \setminus \mathbf{Z}^2 \\ 2 & \text{otherwise.} \end{cases}$$

For $q.x$ not prime, the number of elements in $f[\underline{0}, q]$ fluctuates even more. Consequently, the Close Embedding function does not solve the decreasing intensity problem.

The Close Embedding line function is not convex, as is illustrated in Figure 2.7. Here, $f[\underline{0}, (10,4)]$ contains $(0,1)$ and $(5,3)$, but $f[(0,1), (5,3)]$ is not a subset of $f[\underline{0}, (10,4)]$.

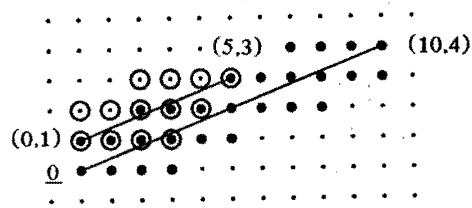


Figure 2. 7

Illustration that the Close Embedding line function is not convex. The elements of $f[\underline{0},(10,4)]$ and $f[(0,1),(5,3)]$ are indicated by • and O respectively.

2.4.7 The Optimal Embedding function

The line function introduced in this section is based on the digitisation function $f_{\models}$, which was defined in Chapter 1 by

$$f_{\models}(V) = \{p \in \mathbf{Z}^2 \mid (\underline{E}v : v \in V : d(p,v) \leqslant ½)\}.$$

The Optimal Embedding function $f : \mathbf{Z}^2 \times \mathbf{Z}^2 \rightarrow \mathbf{P}(\mathbf{Z}^2)$ is defined by

$$f[p,q] := f_{\models}[p,q].$$

In Figure 2.8, the function value $f[\underline{0},(7,5)]$ is shown.

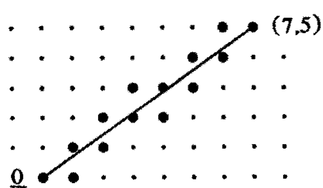


Figure 2. 8

$f[\underline{0},(7,5)]$, *where f is the Optimal Embedding function.*

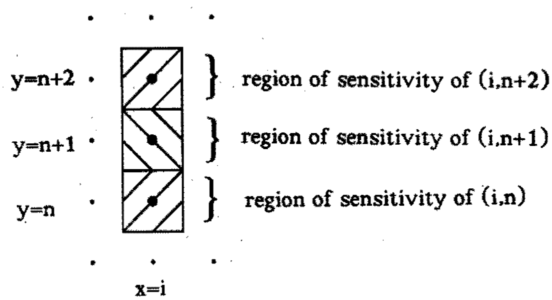
In Chapter 1 it has been shown that f is optimal (Property 1.13) and translation invariant (Property 1.15).

Property 2.46:

The Optimal Embedding function is a line function.

Proof:

This may be proven in the same way as for the Close Embedding function, though in the part where R_i is proven to be connected, the picture should be replaced by the one below.



□

In Figure 2.8 it can be seen that f is not minimal. The following property deals with the number of pixels in $f[p,q]$.

Property 2.47:

For the Optimal Embedding function f,

$$\#f[\underline{0},q] = \begin{cases} q.x + |q.y| + 1 & \text{if } \gcd(q.x, |q.y|) > 0 \wedge \\ & \quad ({}^{q.x}/_{\gcd(q.x,|q.y|)} \text{ even} \vee {}^{|q.y|}/_{\gcd(q.x,|q.y|)} \text{ even}) \\ q.x + |q.y| + 1 + \gcd(q.x, |q.y|) & \text{otherwise,} \end{cases} \qquad \text{(a)}$$

where $\gcd(n, 0)$ is supposed to equal n.

Proof:

Recall that for $f_{\models}$ the region of sensitivity of a pixel is

$$R(p) = \{(x,y) \in \mathbf{R}^2 \mid |x - p.x| \leqslant \tfrac{1}{2} \wedge |y - p.y| \leqslant \tfrac{1}{2}\},$$

and that

$$f[\underline{0},q] = (\cup r : r \in \mathbf{Z}^2 \wedge R(r) \cap [p,q] \neq \emptyset : \{r\}).$$

The construction of the proof is similar to the one above.

- Suppose $q.x = 0$.
  Then

  $$f[\underline{0},q] = [\underline{0},q] \cap \mathbf{Z}^2,$$

  and consequently

  $$\#f[\underline{0},q] = |q.y| + 1.$$

  Since

$$|q.y| = 0 \Rightarrow \gcd(q.x\,,\, |q.y|) = 0$$

$$|q.y| > 0 \Rightarrow \frac{q.x}{\gcd(q.x\,,\, |q.y|)} \text{ even,}$$

equation (a) is satisfied.
For reasons of symmetry, this is also true in case that $|q.y| = 0$.

- Suppose $q.x > 0$ and $|q.y| > 0$ and $\gcd(q.x\,,\, |q.y|) = 1$.
  Define V as the set of points of $[\underline{0}, q]$ with x- or y-coordinates that are odd multiples of ½:

  $$V := \{(x\,,y) \in [\underline{0}, q] \mid x - ½ \in \mathbf{Z} \vee y - ½ \in \mathbf{Z}\}.$$

  Since

  $$\#\{(x\,,y) \in [\underline{0}, q] \mid x - ½ \in \mathbf{Z}\} = q.x \quad \wedge$$
  $$\#\{(x\,,y) \in [\underline{0}, q] \mid y - ½ \in \mathbf{Z}\} = |q.y|,$$

  it follows that

  $$\#V = q.x + |q.y| - \#\{(x\,,y) \in V \mid x - ½ \in \mathbf{Z} \wedge y - ½ \in \mathbf{Z}\}.$$

  We shall show that

  $$\#\{(x\,,y) \in V \mid x - ½ \in \mathbf{Z} \wedge y - ½ \in \mathbf{Z}\} = \begin{cases} 0 & \text{if } q.x \text{ even} \vee |q.y| \text{ even} \\ 1 & \text{otherwise.} \end{cases} \tag{b}$$

  - Suppose $q.x$ even or $|q.y|$ even.
    Furthermore, suppose that $(n + ½, m + ½) \in [\underline{0}, q]$, for some $n\,, m \in \mathbf{Z}$.
    Then

    $$\frac{|2m+1|}{|2n+1|} = \frac{|q.y|}{q.x}.$$

    Since $\gcd(q.x\,,\, |q.y|) = 1$, and $|n + ½| < q.x$, it follows that $|2m+1| = |q.y|$ and $|2n+1| = q.x$. However, because at least one of $q.x$ and $|q.y|$ is even, this leads to a contradiction. Thus,

    $$q.x \text{ even} \vee |q.y| \text{ even} \Rightarrow \#\{(x\,,y) \in V \mid x - ½ \in \mathbf{Z} \wedge y - ½ \in \mathbf{Z}\} = 0.$$

  - Suppose $q.x$ odd $\wedge$ $q.y$ odd.
    Then the point r halfway the line segment $[\underline{0}, q]$,

    $$r = ({}^{q.x}\!/_2, {}^{q.y}\!/_2),$$

    is contained in $\{(x\,,y) \in V \mid x - ½ \in \mathbf{Z} \wedge y - ½ \in \mathbf{Z}\}$.

    Suppose an other point $s \in [\underline{0}, q]$ exists such that $s.x - ½ \in \mathbf{Z}$ and $s.y - ½ \in \mathbf{Z}$. Then

    $$\frac{|r.y - s.y|}{|r.x - s.x|} = \frac{|q.y|}{q.x} \wedge |r.y - s.y| \in \mathbf{N} \wedge |r.x - s.x| \in \mathbf{N} \wedge |r.y - s.y| < |q.y|.$$

    Because $\gcd(q.x\,,\, |q.y|) = 1$, this is impossible, and therefore r is the only element of

$$\{(x,y)\in V \mid x-\tfrac{1}{2}\in\mathbf{Z} \wedge y-\tfrac{1}{2}\in\mathbf{Z}\}.$$

Thus,

$$q.x \text{ odd } \wedge\ q.y \text{ odd } \Rightarrow \#\{(x,y)\in V \mid x-\tfrac{1}{2}\in\mathbf{Z} \wedge y-\tfrac{1}{2}\in\mathbf{Z}\} = 1.$$

Now define $n := \#V$.
Let $r_0, r_1, \ldots, r_{n-1}$ be the sequence of all elements of V ordered in increasing x-value, that is,

$$(\underline{\mathrm{A}}\, i : 0<i<n : r_{i-1}.x < r_i.x).$$

$\underline{0}$ itself contributes one element to $f[\underline{0},q]$. If the line segment $[\underline{0},q]$ is followed from $\underline{0}$ to q, then with each r_i not contained in

$$\{(x,y)\in V \mid x-\tfrac{1}{2}\in\mathbf{Z} \wedge y-\tfrac{1}{2}\in\mathbf{Z}\}$$

one new region of sensitivity is entered, contributing one element to $f[\underline{0},q]$, and with each r_i, if any, contained in

$$\{(x,y)\in V \mid x-\tfrac{1}{2}\in\mathbf{Z} \wedge y-\tfrac{1}{2}\in\mathbf{Z}\}$$

three new regions of sensitivity are entered, contributing three elements to $f[\underline{0},q]$. Consequently,

$$q.x \text{ even } \vee\ |q.y| \text{ even } \Rightarrow \#f[\underline{0},q] = 1+(q.x+|q.y|)$$

$$q.x \text{ odd } \wedge\ |q.y| \text{ odd } \Rightarrow \#f[\underline{0},q] = 1+(q.x+|q.y|-2)+3.$$

Since $\gcd(q.x, |q.y|) = 1$, this results in

$$\#f[\underline{0},q] = \begin{cases} 1+q.x+|q.y| & \text{if } {}^{q.x}/_{\gcd(q.x,|q.y|)} \text{ even } \vee\ {}^{|q.y|}/_{\gcd(q.x,|q.y|)} \text{ even} \\ 1+q.x+|q.y|+\gcd(q.x,|q.y|) & \text{otherwise.} \end{cases}$$

Hence, equation (a) is satisfied.

- Suppose $q.x>0$ and $|q.y|>0$ and $\gcd(q.x,|q.y|)>1$.
Similar to the reasoning in the third part of the Proof of the previous Lemma, the line segment $[\underline{0},q]$ may be divided into n parts $[p_i,p_{i-1}]$, where $n=\gcd(q.x,|q.y|)$, $p_i\in\mathbf{Z}^2$,

$$|p_{i+1}.x-p_i.x| = {}^{q.x}/_{n},\quad |p_{i+1}.y-p_i.y| = {}^{|q.y|}/_{n},$$

and

$$f[p_{i-1},p_i]\cap f[p_i,p_{i+1}] = \{p_i\}.$$

Then

$$\#f[\underline{0},q] = n*(\#f[p_i,p_{i+1}])-(n-1)$$

$$= \begin{cases} n*(1 + {}^{q.x}/_n + {}^{|q.y|}/_n) - (n-1) & \text{if } {}^{q.x}/_n \text{ even } \vee {}^{|q.y|}/_n \text{ even} \\ n*(1 + {}^{q.x}/_n + {}^{|q.y|}/_n + 1) - (n-1) & \text{otherwise} \end{cases}$$

$$= \begin{cases} 1 + {}^{q.x}/_n + {}^{|q.y|}/_n & \text{if } {}^{q.x}/_n \text{ even } \vee {}^{|q.y|}/_n \text{ even} \\ 1 + {}^{q.x}/_n + {}^{|q.y|}/_n + n & \text{otherwise.} \end{cases}$$

Thus, equation (a) has been proved.

□

Note that for $q.x$ a fixed prime and $0 \leqslant |q.y| \leqslant q.x$,

$$\# f[\underline{0}, q] = \begin{cases} 1 + q.x + |q.y| & \text{if } |q.y| \text{ even} \\ 2 + q.x + |q.y| & \text{if } |q.y| \text{ odd} \wedge |q.y| < q.x \\ 1 + 2q.x + |q.y| & |q.y| = q.x. \end{cases}$$

Hence, a discrepancy occurs at $|q.y| = q.x$ when $|q.y|$ increases from 0 to $q.x$. This is due to the following property of $f_{\models}$.

$$\# f_{\models}(\{r\}) = \begin{cases} 1 & \text{if } r - (\tfrac{1}{2}, \tfrac{1}{2}) \notin \mathbf{Z}^2 \\ 4 & \text{if } r - (\tfrac{1}{2}, \tfrac{1}{2}) \in \mathbf{Z}^2 \\ 2 & \text{otherwise.} \end{cases}$$

For $q.x$ not prime, the number of elements in $f[\underline{0}, q]$ fluctuates even more. Consequently, the Optimal Embedding function does not solve the decreasing intensity problem.

The Optimal Embedding function is not convex: $f[\underline{0}, (10,4)]$, for example, contains (1,1) and (6,3), but $f[(1,1), (6,3)]$ is not a subset of $f[\underline{0}, (10,4)]$. See Figure 2.9.

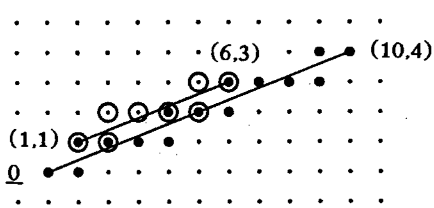


Figure 2. 9

Illustration that the Optimal Embedding line function is not convex. The elements of $f[\underline{0}, (10,4)]$ and $f[(1,1), (6,3)]$ are indicated by • and ○ respectively.

2.5 Combinations of properties

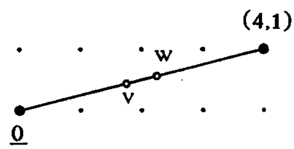
In this section we shall prove that some combinations of properties are conflicting. The first one concerns optimality and minimality.

Lemma 2.48:

There are no minimal, optimal line functions.

Proof:

Suppose f is an optimal line function. Consider the line segment $[\underline{0}, (4,1)]$. See the figure alongside. This line segment contains the points $v=(7/4, 7/16)$ and $w=(9/4, 9/16)$. According to the definition of optimal line function, $f[\underline{0}, (4,1)]$ must contain pixels p and q such that $d(p,v) \leqslant 1/2$ and $d(q,w) \leqslant 1/2$. Hence $f[\underline{0}, (4,1)]$ contains both $(2,0)$ and $(2,1)$, which implies that f is not minimal.



□

Next, we derive some properties concerning the chain codes of the paths generated by minimal, convex line functions.

Property 2.49:

Let f be a minimal, convex line function, and $p, q \in \mathbf{Z}^2$.
Let $\pi = r_0, r_1, \ldots, r_n$, where $n = d(p,q)$, be the minimal path from p to q associated with $f[p,q]$. Then for all $i, j \in \mathrm{N}$ with $0 \leqslant i \leqslant j \leqslant n$,

$$c \circ f[r_i, r_j] = c \circ f[p,q][i+1:j].$$

Proof:

Because f is convex,

$$f[r_i, r_j] \subseteq f[p,q].$$

Because f is minimal,

$$f[r_i, r_j] = \{r_i, r_{i+1}, \ldots, r_j\},$$

and hence

$$c \circ f[r_i, r_j] = c \circ f[p,q][i+1:j].$$

□

Although there are various convex line functions, the additional requirement for minimality and translation invariance is rather restrictive; the following lemma expresses that the chain codes of such line functions have a simple structure.

Lemma 2.50:

Let f be a translation invariant, minimal, convex line function, $p, q \in \mathbf{Z}^2$, $a \in \{0,\dots,7\}$, and $\sigma \in \{0,\dots,7\}^*$. Then

$$c \circ f[p,q] = a\sigma a \quad \Rightarrow \quad \sigma \in \{a\}^*.$$

Proof:

We shall use induction on the length of σ.

- If σ has length 0, then $\sigma \in \{a\}^*$.
- Suppose for all $\sigma \in \{0,\dots,7\}^*$ with length at most n, where $n \geqslant 0$,

$$c \circ f[p,q] = a\sigma a \quad \Rightarrow \quad \sigma \in \{a\}^*.$$

Let $\sigma \in \{0,\dots,7\}^*$ have length $n+1$. Then

$$c \circ f[p,q] = a\sigma a$$

$\Rightarrow$ { Property 2.11 }

$$c \circ f[p+v(a),q] = \sigma a \quad \wedge \quad c \circ f[p,q-v(a)] = a\sigma$$

$\Rightarrow$ { translation invariance }

$$\sigma a = a\sigma$$

$\Rightarrow$ { length of σ is at least 1 }

$$\sigma = a \quad \vee \quad (\underline{E}\gamma : \gamma \in \{0,\dots,7\}^* : \sigma = a\gamma a)$$

$\Rightarrow$ { induction assumption for γ }

$$\sigma \in \{a\}^*.$$

The lemma has now been proven.

□

Corollary 2.51:

For any translation invariant, minimal, convex, line function f, and any $p, q \in \mathbf{Z}^2$,

$$c \circ f[p,q] = a_0^{\,i(a_0)} a_1^{\,i(a_1)} \cdots a_7^{\,i(a_7)},$$

where $a_0, a_1, \dots, a_7$ is a permutation of $0,1,\dots,7$.

□

From Property 2.11 we know that the alphabet of the chain code of any minimal path has at most three elements. For $\hat{q}-\hat{p} \in O_0 \cup O_7$, for example, this alphabet is a subset of $\{0,1,7\}$. In combination with the above Corollary and Lemma 2.12, the chain code of $c \circ f[p,q]$ has structure

$$a^{i_a} b^{i_b} c^{i_c},$$

where a,b,c is a permutation of 7,0,1, and $i_7+i_0+i_1 = |q.x-p.x|$, and $i_1-i_7 = \hat{q}.y-\hat{p}.y$. However, the range of possible chain codes can be restricted even more, as is stated in the following lemma.

Lemma 2.52:

Let f be a translation invariant, minimal, convex line function, and let $p,q \in \mathbf{Z}^2$, such that $\hat{q}-\hat{p} \in O_0 \cup O_7$. Then

$$c \circ f[p,q] = a^{i_a} b^{i_b} c^{i_c},$$

where a,b,c is a permutation of 7,0,1, and $i_7+i_0+i_1 = |q.x-p.x|$, and $i_1-i_7 = \hat{q}.y-\hat{p}.y$, and

$$i_0>1 \wedge i_1>0 \wedge i_7>0 \implies b=0.$$

Proof:

From Lemma 2.12 and Corollary 2.51 it follows that

$$c \circ f[p,q] = a^{i_a} b^{i_b} c^{i_c},$$

where a,b,c is a permutation of 7,0,1, and $i_7+i_0+i_1 = |q.x-p.x|$, and $i_1-i_7 = \hat{q}.y-\hat{p}.y$.
Suppose $i_0>1 \wedge i_1>0 \wedge i_7>0$.
If $a=0$ or $c=0$ then $c \circ f[p,q]$ contains either the substring 17 or 71. From Property 2.11 it follows that $f[p,q]$ contains r and s such that $c \circ f[r,s] = 17$ or $c \circ f[r,s] = 71$. This means that $s = r+v(1)+v(7) = r+(2,0)$. Since f is translation invariant, it follows from Corollary 2.18 that

$$c \circ f[\underline{0},(2,0)] = 17 \vee c \circ f[\underline{0},(2,0)] = 71. \tag{a}$$

However, since $i_0 \geq 2$, $c \circ f[p,q]$ also contains the substring 00, which would imply, following a similar reasoning, that $c \circ f[\underline{0},(2,0)] = 00$. This contradicts (a), and therefore $b=0$.

□

Similar lemmas may be proven for the cases $\hat{q}-\hat{p} \in O_1$ and $\hat{q}-\hat{p} \in O_6$. We will present them without proof.

Lemma 2.53:

Let f be a translation invariant, minimal, convex line function, and let $p,q \in \mathbf{Z}^2$, such that $\hat{q}-\hat{p} \in O_1$. Then

$$c \circ f[p,q] = a^{i_a} b^{i_b} c^{i_c},$$

where a,b,c is a permutation of 1,2,3, and $i_1+i_2+i_3 = |q.y-p.y|$, and $i_1-i_3 = \hat{q}.x-\hat{p}.x$, and

$$i_2>1 \wedge i_1>0 \wedge i_3>0 \Rightarrow b=2.$$

□

Lemma 2.54:

Let f be a translation invariant, minimal, convex line function, and let $p,q \in \mathbf{Z}^2$, such that $\hat{q}-\hat{p} \in O_6$. Then

$$c \circ f[p,q] = a^{i_a} b^{i_b} c^{i_c},$$

where a,b,c is a permutation of 5,6,7, and $i_5+i_6+i_7 = |q.y-p.y|$, and $i_7-i_5 = \hat{q}.x-\hat{p}.x$, and

$$i_6>1 \wedge i_7>0 \wedge i_5>0 \Rightarrow b=6.$$

□

From the line functions presented thus far, the Adapted Franklin function is the only one which is minimal, translation invariant, and convex. The chain code structure of its digitised lines is previously described by Lemma 2.37; that structure is a special case of the structure described by the above lemmas. We also know that the Adapted Franklin function is not close. The following theorem states that any translation invariant, minimal, convex line function has a deviation function that is at least linear in $d(p,q)$.

Theorem 2.55:

For all translation invariant, minimal, convex line functions f, and all $n \in \mathbf{N}$, $n \geqslant 6$,

$$E_f(n) \geqslant \frac{n}{6}. \qquad \text{(a)}$$

Proof:

Let f be a translation invariant, minimal, convex line function.
Let $n \in \mathbf{N}$, $n \geqslant 6$.
Define $q := (n,k)$, where $k = \lfloor \frac{1}{2} n \rfloor$. We shall show that

$$e\,0_f[\underline{0},q] \geqslant \frac{n}{6}.$$

Then, from the definition of E_f, (a) follows.
Recall that

$$e\,0_f\,[\underline{0},q\,] = (\underline{\max}\, r : r \in f[\underline{0},q\,] : \tilde{d}\,(r\,,[\underline{0},q\,])\,).$$

We extend $e\,0$ to pixel sets in the following way. For $P \subseteq \mathbf{Z}^2$,

$$e\,0(P) := (\underline{\max}\, r : r \in P : \tilde{d}\,(r\,,[\underline{0},q\,])\,).$$

Note that with the above definition,

$$e\,0_f\,[\underline{0},q\,] = e\,0(f\,[p\,,q\,]).$$

From Lemma 2.52 it follows that the chain code γ associated with $f\,[\underline{0},q\,]$ satisfies

$$\gamma = a^{i_a} b^{i_b} c^{i_c} \tag{b}$$

where $a\,,b\,,c$ is a permutation of 7,0,1, and

$$i_7 + i_0 + i_1 = n \tag{c}$$

$$i_1 - i_7 = k \tag{d}$$

$$i_0 > 1 \;\wedge\; i_1 > 0 \;\wedge\; i_7 > 0 \quad \Rightarrow \quad b = 0. \tag{e}$$

We shall show that for any path π from $\underline{0}$ to q whose chain code satisfies (b–e),

$$e\,0(<\pi>) \geqslant \frac{n}{6}.$$

Then it follows that

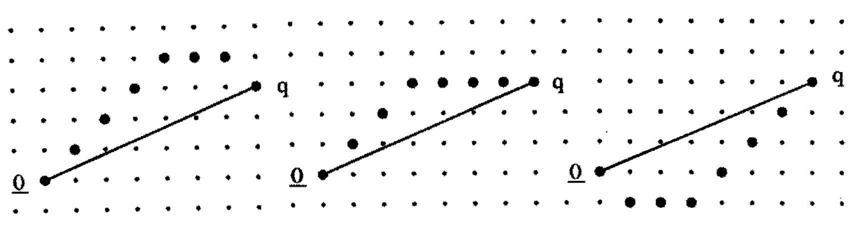
$$e\,0_f\,[\underline{0},q\,] \geqslant \frac{n}{6}.$$

Let π be a path from $\underline{0}$ to q whose chain code σ satisfies (b–e).

- Suppose $i_0 > 1$. Then (e) implies that

$$\sigma = 1^{i_1} 0^{i_0} 7^{i_7} \;\vee\; \sigma = 7^{i_7} 0^{i_0} 1^{i_1}. \tag{f}$$

Examples of π are shown below.



It can be seen that of all paths whose chain codes satisfy (b),(c),(d), and (f), the ones associated with $1^{i_1}0^{i_0}$ and $0^{i_0}1^{i_1}$ have the smallest $e\,0$ value. From Property 2.37 we know that $0^{i_0}1^{i_1}$ is the chain code of an Adapted Franklin line, and from the proof of Property 2.39 we know that the $e\,0$ value of its associated path is at least $^n\!/\!_6$.

Since $1^{i_1}0^{i_0}$ and $0^{i_0}1^{i_1}$ are symmetrical with regard to $[\underline{0}, q]$, we may conclude that if $i_0 > 0$, then for any path π whose chain code satisfies (b–e),

$$e\,0(<\pi>) \geqslant \frac{n}{6}.$$

- Suppose $i_0 \leqslant 1$.
  From (b) and (c) it follows that $2i_1 = n + k - i_0$, hence,

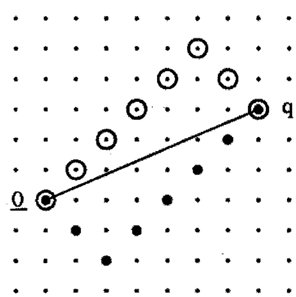
$$i_0 = \begin{cases} 0 & \text{if } n+k \text{ even} \\ 1 & \text{if } n+k \text{ odd.} \end{cases}$$

- If $i_0 = 0$ (hence $n + k$ is even), it follows from (b), (c), and (d) that

$$\sigma = 1^{i_1}7^{i_7} \;\vee\; \sigma = 7^{i_7}1^{i_1},$$

where

$$i_1 + i_7 = n \;\wedge\; i_1 - i_7 = k.$$



The associated paths are shown alongside. It can be seen that in these cases $e\,0(<\pi>) = i_7$. Since

$$i_7 = \tfrac{1}{2}(n-k) = \tfrac{1}{4}n \geqslant \tfrac{1}{6}n,$$

it follows that if $i_0 = 0$, then for any path π whose chain code satisfies (b), (c), and (d),

$$e\,0(<\pi>) \geqslant \frac{n}{6}.$$

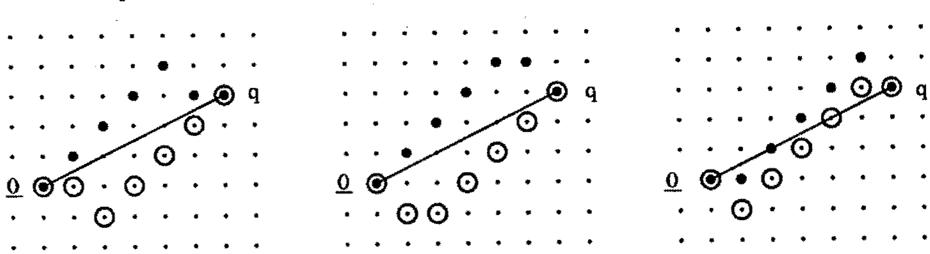
- If $i_0 = 1$ (hence $n + k$ is odd), it follows from (b), (c), and (d) that

$$\sigma = 07^{i_7}1^{i_1} \;\vee\; \sigma = 7^{i_7}01^{i_1} \;\vee\; \sigma = 7^{i_7}1^{i_1}0 \;\vee$$
$$\sigma = 1^{i_1}7^{i_7}0 \;\vee\; \sigma = 1^{i_1}07^{i_7} \;\vee\; \sigma = 01^{i_1}7^{i_7},$$

where

$$i_1 + i_7 = n - 1 \;\wedge\; i_1 - i_7 = k.$$

The paths associated with these chain codes are shown below.



It can be seen that for these paths π, $e\,0(<\pi>) \geqslant i_7$. Since

$$i_7 = \tfrac{1}{2}(n-1-k) = \tfrac{1}{2}(n-1-\lfloor\tfrac{1}{2}n\rfloor) \geqslant \tfrac{1}{4}n - \tfrac{1}{2} \geqslant \frac{n}{6}, \quad \text{for } n \geqslant 6,$$

it follows that if $i_0 = 1$, then for any path π whose chain code satisfies (b), (c), and (d),

$$e\,0(<\pi>) \geqslant \frac{n}{6}.$$

Now we have proven that for any path π whose chain code satisfies (b–e),

$$e\,0(<\pi>) \geqslant \frac{n}{6}, \tag{g}$$

and thus the path π associated with $f[\underline{0}, q]$ also satisfies (g), which completes the proof.

□

Corollary 2.56:

No minimal, convex, translation invariant line function exists that is close.

□

2.6 Concluding remarks

In this chapter we have introduced a new concept of line digitisation function. This definition is rather general. However, several classes of line functions are distinguished, based on various properties of line functions; restricting oneself to one of these classes is like incorporating the associated property into the definition of line function. The properties considered are translation invariance, minimality, convexity, and closeness.

The main theorem of this chapter (Theorem 2.55), implies that no translation invariant, minimal, convex line function exists that is close. We even conjecture that convexity and closeness are mutual exclusive properties for line functions.

From Theorem 2.55 we know that the deviation of any translation invariant, minimal, convex line function is at least linear in $d(p,q)$. An important consequence hereof is that one has to drop minimality or translation invariance if one wants to search for convex line functions that have smaller deviation functions. This idea will be worked out in Chapter 4.

In Section 2.4, seven examples of line functions have been presented. In Table 2.0 these functions are classified with regard to the above properties. The digitisations of the line segment $[\underline{0},(11,4)]$ for these functions are shown in Figure 2.10.

Line Function	*Property*			
	minimal	*tr. invar.*	*convex*	*close*
Bounding Box	-	+	+	-
Minimal Path Set	-	+	+	-
Franklin	-	+	+	-
Adapted Franklin	+	+	+	-
Bresenham	+	+	-	+
Close Embedding	-	+	-	+
Optimal Embedding	-	+	-	+

Table 2.0

Classification of the line functions of Section 2.4

The Bounding Box and Minimal Path set functions are merely presented to show that the notion of line function as introduced in this chapter, allows for rather exotic digitisation functions. They are not considered to be very useful. The same holds for the Franklin and Adapted Franklin function. Nevertheless, if minimality, translation invariance, and convexity is required, the Adapted Franklin line function is about the best we can get.

The Bresenham function turns out to be minimal, translation invariant, and close; its practical use is beyond dispute.

The Close Embedding and Optimal Embedding functions are close and optimal respectively, and they are both translation invariant. When combined with a suitable intensity function, we expect these functions to be candidates for line digitisation functions incorporating anti-aliasing.

We shall now spend a few words on algorithms for these line functions. Any algorithm that computes $f\,[p\,;q\,]$ for a particular line function f and arbitrary pixels p and q, should generate the same set of pixels when the input parameters p and q are

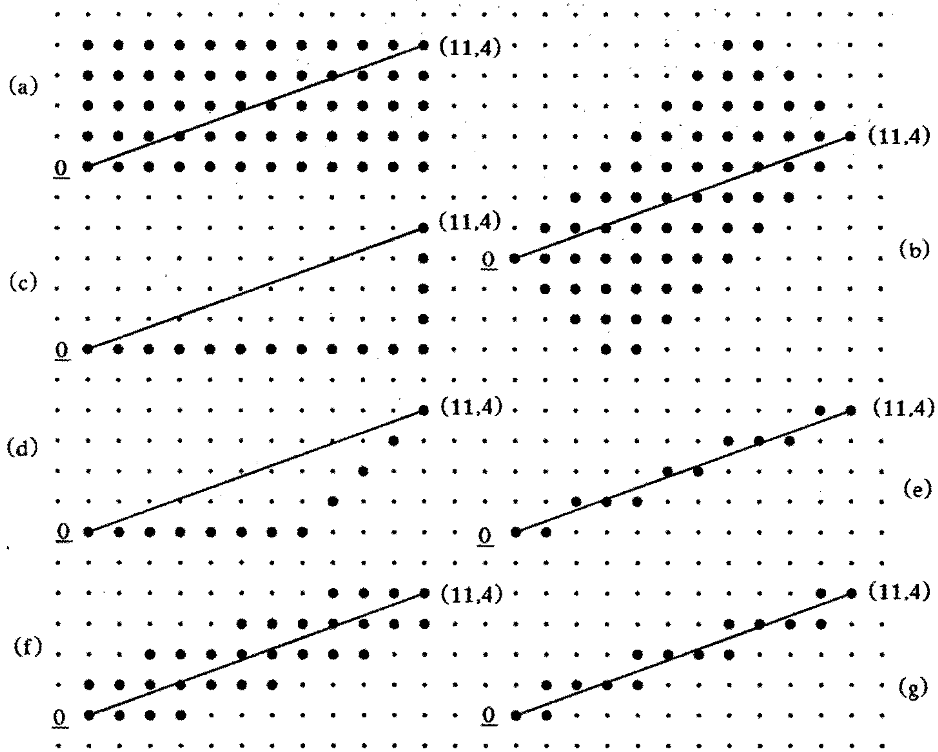


Figure 2. 10

Various digitisations of the line segment [0,(11,4)].

a) Bounding Box function
b) Minimal Path Set function
c) Franklin function
d) Adapted Franklin function
e) Bresenham function
f) Close Embedding function
g) Optimal Embedding function.

interchanged. Although this may seem a trivial remark, most existing algorithms for line functions do not meet this requirement. An extensive treatment on this subject can be found in [Bresenham 1986].

The definition of the Franklin and the Adapted Franklin line functions are such that (integer) algorithms for these functions may easily be derived. For the Bresenham function, we refer to the literature, [Bresenham 1965], [van Berckel & Mailloux 1965], or [Foley & van Dam 1982], for instance. These algorithms, however, need some adjustments in order to guarantee consistency with regard to endpoint interchange.

For the Close Embedding and Optimal Embedding functions we refer to [van Overveld 1987a], where algorithms are presented which use integer arithmetic only.

3

Recursive line functions

3.0 Introduction

In this chapter we present three other line functions, which will be defined recursively. For a general treatment of recursiveness we refer to [Yasuhara 1971] or [Eilenberg & Elgot 1970].

The line functions presented in this chapter, are based on the following principle: define a pixel r somewhere halfway the line segment $[p,q]$, and subsequently define

$$f[p,q] = f[p,r] \cup f[r,q].$$

r is called the *split point* of $f[p,q]$. Different definitions of r induce different line functions.

There are two reasons why we consider this class of recursive line functions. Firstly, the algorithms to generate recursive line functions are easily derived from the definition. If in these definitions only integer values occur, the algorithms will generate the correct images for any permitted input on any machine. In order to make these functions attractive for hardware implementations, the operations used in the definition should be as simple as possible.

Secondly, the above split point technique is a divide-and-conquer technique: the task of computing $f[p,q]$ is split up into two smaller tasks. (See [Aho et al 1974] for a general treatment of divide-and-conquer techniques.) If the subtasks have about the same size, several operations may be performed in logarithmic instead of linear time complexity. A well-known example hereof is finding a number in a sorted sequence by binary search ([Knuth 1973]). For a split point line function f, the point-containment test (finding out whether a given pixel r is contained in $f[p,q]$) would have logarithmic time complexity.

In order to guarantee that $f[p,r]$ and $f[r,q]$ have about the same size, r should be chosen near the point $v = \frac{1}{2}(p+q)$ halfway the line segment $[p,q]$. If v is not a pixel, that is, if $p.x+q.x$ or $p.y+q.y$ is odd, one has to choose one of the pixels nearby. Of course, it is possible to compute which of these pixels has the smallest distance to the line segment, but this requires quite a few operations. Since we wish, with an eye to fast hardware implementations, as few and as simple operations as possible, it is more convenient to choose a pixel by rounding the values $\frac{1}{2}(p.x+q.x)$ and $\frac{1}{2}(p.y+q.y)$ in a definite way. A natural operation for this is integer division by two, or, if *shift* operations are supported, a shift to the right. Since the effect of a right shift depends on the way integers are represented in computers (whether the "two's complement notation" is used or not, see [Tanenbaum 1984]), we shall abstract from this operation.

In Section 1 we present two functions that generate split points. Based on these functions, in Sections 2, 3, and 4, three different line functions are introduced. These line functions will be discussed with respect to translation invariance, minimality, convexity, and closeness. Section 5 contains some concluding remarks.

3.1 Arithmetic

In this section various properties and definitions have been assembled that will be needed in the following sections. Most of the properties are presented without proof.

To denote the distance of a point to $\underline{0}$, we introduce the length function $l : \mathbf{R}^2 \rightarrow \mathbf{R}^+$, defined by

$$l(v) := d(v, \underline{0}).$$

In the previous chapter the notion of octants has been introduced. The following lemma expresses that within each octant the length of the sum of two points equals the sum of the lengths of the points.

Lemma 3.0:

For any octant O_i, and all points $v, w \in O_i$,

$$l(v+w) = l(v) + l(w).$$

Proof:

Let $v, w \in O_0$. Then

$l(v+w)$

$=$ { definition l and d }

$\max(|v.x + w.x|, |v.y + w.y|)$

$=$ { $v, w \in O_0$ }

$v.x + w.x$

$=$ { definition l and d and $v, w \in O_0$ }

$l(v) + l(w).$

If v and w are contained in one of the other octants, a similar reasoning holds.

□

For the maximum function the following holds.

Property 3.1:

For all $x, y, z \in \mathbf{R}$,

a) $x \leqslant y \Rightarrow \max(x, z) \leqslant \max(y, z)$

b) $\max(x+y, z+y) = \max(x, z) + y$.

□

Now two so-called bitwise operators are introduced; for their definition we need the binary representation of natural numbers. If $a \in \mathbf{N}$, and

$$a = \sum_{i=0}^{n} a_i 2^i, \quad \text{where } a_i \in \{0, 1\} \text{ and } a_n = \begin{cases} 0 & \text{if } a = 0 \\ \lfloor {}^2\log a \rfloor & \text{if } a > 0. \end{cases}$$

then $a_n, a_{n-1}, \ldots, a_0$ is called the *binary representation of a*.

If $a_n, a_{n-1}, \ldots, a_0$ and $b_m, b_{m-1}, \ldots, b_0$ are the binary notations of a and b respectively, and $l = \min(n, m)$, then we define the complement of a by

$$\neg a := \sum_{i=0}^{n} (1-a_i) 2^i.$$

and the bitwise conjunction of a and b by

$$a \,\&\, b := \sum_{i=0}^{l} c_i 2^i \quad \text{where } c_i = \begin{cases} 1 & \text{if } a_i = b_i = 1 \\ 0 & \text{otherwise.} \end{cases}$$

We shall frequently use integer division by 2, denoted as $\underline{\mathrm{div}}\,2$, and defined by

$$a\,\underline{\mathrm{div}}\,2 := \begin{cases} \left\lfloor \frac{a}{2} \right\rfloor & \text{if } a \geqslant 0 \\ \left\lceil \frac{a}{2} \right\rceil & \text{if } a < 0. \end{cases}$$

For $a - a\,\underline{\mathrm{div}}\,2$ we introduce the notation $a\,\underline{\mathrm{vid}}\,2$, that is,

$$a\,\underline{\mathrm{vid}}\,2 := a - a\,\underline{\mathrm{div}}\,2.$$

Associated with integer division by 2 is the operator $\underline{\mathrm{mod}}\,2$, defined by

$$a\,\underline{\mathrm{mod}}\,2 := a - 2(a\,\underline{\mathrm{div}}\,2).$$

We shall use the following properties.

Property 3.2:

For all $a, b \in \mathbf{Z}$ and $n, m \in \mathbf{N}$ holds.

a) $(-a)\,\underline{\text{div}}\,2 = -(a\,\underline{\text{div}}\,2)$

b) $|a|\,\underline{\text{div}}\,2 = |a\,\underline{\text{div}}\,2|$

c) $a \leqslant b \;\Rightarrow\; a\,\underline{\text{div}}\,2 \leqslant b\,\underline{\text{div}}\,2 \;\wedge\; a\,\underline{\text{vid}}\,2 \leqslant b\,\underline{\text{vid}}\,2$

d) $(\max(a, b))\underline{\text{div}}\,2 = \max(a\,\underline{\text{div}}\,2, b\,\underline{\text{div}}\,2)$

e) $a > 1 \;\Rightarrow\; 0 < a\,\underline{\text{div}}\,2 < a$

f) $a\,\underline{\text{mod}}\,2 = \begin{cases} 0 & \text{if } a \text{ even} \\ 1 & \text{if } a \text{ odd} \wedge a > 0 \\ -1 & \text{if } a \text{ odd} \wedge a < 0 \end{cases}$

g) $(\neg n)\,\underline{\text{div}}\,2 = \neg(n\,\underline{\text{div}}\,2)$

h) $(\neg n)\,\underline{\text{mod}}\,2 = \neg(n\,\underline{\text{mod}}\,2)$

i) $(n \,\&\, m)\,\underline{\text{div}}\,2 = (n\,\underline{\text{div}}\,2) \,\&\, (m\,\underline{\text{div}}\,2)$

j) $(n \,\&\, m)\,\underline{\text{mod}}\,2 = (n\,\underline{\text{mod}}\,2) \,\&\, (m\,\underline{\text{mod}}\,2)$

□

The operators introduced above are extended to pixels in a straightforward way. For example, for $p \in \mathbf{Z}^2$,

$$p\,\underline{\text{div}}\,2 \;:=\; (p.x\,\underline{\text{div}}\,2, p.y\,\underline{\text{div}}\,2).$$

Property 3.3:

For all octants O_i and all p holds.

$$p \in O_i \;\Rightarrow\; p\,\underline{\text{div}}\,2 \in O_i \;\wedge\; p\,\underline{\text{vid}}\,2 \in O_i\,.$$

Proof:

We shall prove the property for the octants O_0 and O_2 only; the other octants may be treated in a similar way.

- $p \in O_0$

$\Rightarrow$ { definition O_0 }

$0 \leqslant p.y \leqslant p.x$

$\Rightarrow$ { Property 3.2(c) and $0\,\underline{\text{div}}\,2 = 0$ }

$0 \leqslant p.y\,\underline{\text{div}}\,2 \leqslant p.x\,\underline{\text{div}}\,2 \;\wedge\; 0 \leqslant p.y\,\underline{\text{vid}}\,2 \leqslant p.x\,\underline{\text{vid}}\,2$

$\Rightarrow$ { definition O_0 }

$p\,\underline{\mathrm{div}}\,2 \in O_0 \wedge p\,\underline{\mathrm{vid}}\,2 \in O_0$.

- $p \in O_2$

$\Rightarrow$ { definition O_2 }

$0 \leqslant -p.x \leqslant p.y$

$\Rightarrow$ { Property 3.2(c) and $0\,\underline{\mathrm{div}}\,2 = 0$ }

$0 \leqslant (-p.x)\underline{\mathrm{div}}\,2 \leqslant p.y\,\underline{\mathrm{div}}\,2 \;\wedge\; 0 \leqslant (-p.x)\underline{\mathrm{vid}}\,2 \leqslant p.y\,\underline{\mathrm{vid}}\,2$

$\Rightarrow$ { Property 3.2(a) }

$0 \leqslant -(p.x\,\underline{\mathrm{div}}\,2) \leqslant p.y\,\underline{\mathrm{div}}\,2 \;\wedge\; 0 \leqslant -(p.x\,\underline{\mathrm{vid}}\,2) \leqslant p.y\,\underline{\mathrm{vid}}\,2$

$\Rightarrow$ { definition O_2 }

$p\,\underline{\mathrm{div}}\,2 \in O_2 \wedge p\,\underline{\mathrm{vid}}\,2 \in O_2$.

□

To compute split points we introduce two functions, *sdv* and *s fl*.

The first function, $sdv : \mathbf{Z}^2 \times \mathbf{Z}^2 \to \mathbf{Z}^2$, is defined by

$$sdv(p,q) \;:=\; p + (q-p)\underline{\mathrm{div}}\,2.$$

Since $(q-p)\underline{\mathrm{div}}\,2$ may be seen as an integer approximation of $v = \frac{1}{2}(q-p)$, $sdv(p,q)$ may be seen as an integer approximation of v, where v is rounded in the direction of p. Note that, in general, $sdv(p,q) \neq sdv(q,p)$. More precisely, the following properties hold.

Property 3.4:

For all $p, q \in \mathbf{Z}^2$,

$$(sdv(p,q) = sdv(q,p)) \quad \equiv \quad (q.x - p.x \text{ even} \wedge q.y - p.y \text{ even}).$$

□

Property 3.5:

$$d(p, sdv(p,q)) = d(p,q)\underline{\mathrm{div}}\,2.$$

Proof:

$d(p, sdv(p,q))$

$=$ { definition sdv }

$d(p, p + (q-p)\underline{\mathrm{div}}\,2)$

$$
\begin{aligned}
= &\quad \{ \text{ Property 1.0 } \} \\
& d(\underline{0}, (q-p)\,\underline{\text{div}}\,2) \\
= &\quad \{ \text{ definition of } d \ \} \\
& \max(\,|(q.x-p.x)\,\underline{\text{div}}\,2\,|,\ |(q.y-p.y)\,\underline{\text{div}}\,2\,|) \\
= &\quad \{ \text{ Property 3.2(b,d) } \} \\
& \max(\,|q.x-p.x\,|,\ |q.y-p.y\,|)\,\underline{\text{div}}\,2 \\
= &\quad \{ \text{ definition of } d \ \} \\
& d(p,q)\,\underline{\text{div}}\,2.
\end{aligned}
$$

□

Property 3.6:

$$d(p, sdv(p,q)) + d(sdv(p,q), q) = d(p,q).$$

Proof:

$$
\begin{aligned}
& d(p, sdv(p,q)) + d(sdv(p,q), q) \\
= &\quad \{ \text{ definition } sdv \text{ and Property 1.0 } \} \\
& d(\underline{0}, (q-p)\,\underline{\text{div}}\,2) + d(\underline{0}, (q-p)\,\underline{\text{vid}}\,2) \\
= &\quad \{ \text{ definition of } l \ \} \\
& l((q-p)\,\underline{\text{div}}\,2) + l((q-p)\,\underline{\text{vid}}\,2) \\
= &\quad \{ \text{ Property 3.3 and Lemma 3.0 } \} \\
& l(q-p) \\
= &\quad \{ \text{ definition of } l \text{ and Property 1.0 } \} \\
& d(p,q).
\end{aligned}
$$

□

In the following sections we shall prove properties by induction on $d(p,q)$. The following corollary will then be frequently used.

Corollary 3.7:

If $d(p,q) > 1$ and $r = sdv(p,q)$, then

$$0 < d(p,r) < d(p,q) \;\wedge\; 0 < d(r,q) < d(p,q).$$

□

Corollary 3.8:

If $r = sdv(p,q)$, then

$$d(p,q) \text{ even} \Rightarrow d(p,r) = d(r,q) = \tfrac{1}{2}d(p,q)$$

$$d(p,q) \text{ odd} \Rightarrow (d(p,r) = \tfrac{1}{2}(d(p,q)-1) \;\wedge\; d(r,q) = \tfrac{1}{2}(d(p,q)+1)).$$

□

The second function, $sfl : \mathbf{Z}^2 \times \mathbf{Z}^2 \to \mathbf{Z}^2$, is defined by

$$sfl(p,q) := \lfloor \tfrac{1}{2}(p+q) \rfloor.$$

$sfl(p,q)$ may be seen as an integer approximation of $v = \tfrac{1}{2}(p+q)$, where v is coordinatewise rounded downwards. Note that for all p and q

$$sfl(p,q) = sfl(q,p) = p + \lfloor \tfrac{1}{2}(q-p) \rfloor = q + \lfloor \tfrac{1}{2}(p-q) \rfloor.$$

Furthermore, for p and q such that $q-p \in O_0 \cup O_1$,

$$sdv(p,q) = sfl(p,q).$$

In Figure 3.0 the differences between $sdv(p,q)$, $sdv(q,p)$, and $sfl(p,q)$ are illustrated.

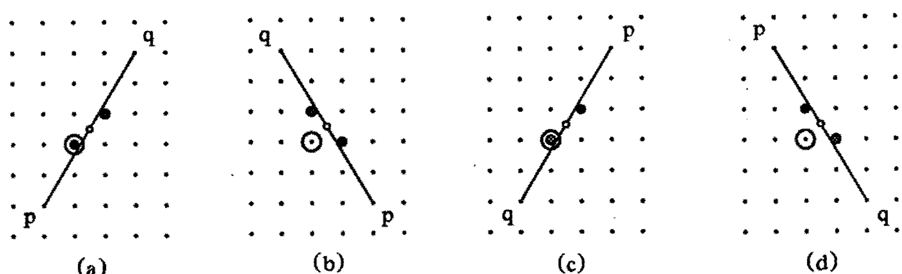


Figure 3.0

Differences between $v = \tfrac{1}{2}(p+q)$, $sdv(p,q)$, $sdv(q,p)$, $sfl(p,q)$, indicated by ∘, •, •, and ○ respectively.

$p = \underline{0}$

$a)\ q = (3,5)$ $b)\ q = (-3,5)$ $c)\ q = (-3,-5)$ $d)\ q = (3,-5)$

The following properties of sfl are the analogies of Properties 3.5 and 3.6.

Property 3.9:

$$d(p,q)-1 \leqslant 2*d(p,sfl(p,q)) \leqslant d(p,q)+1.$$

Proof:

$$2*d(p,sfl(p,q))$$
$$= \quad \{\text{ definition } sfl \}$$
$$2*d(p,\lfloor \tfrac{1}{2}(p+q)\rfloor)$$
$$= \quad \{\text{ definition of } d \}$$
$$2*\max(\,|p.x-\lfloor \tfrac{1}{2}(p.x+q.x)\rfloor|,\,|p.y-\lfloor \tfrac{1}{2}(p.y+q.y)\rfloor|\,)$$
$$= \quad \{\text{ properties of max and } |\,| \}$$
$$\max(\,|2p.x-2\lfloor \tfrac{1}{2}(p.x+q.x)\rfloor|,\,|2p.y-2\lfloor \tfrac{1}{2}(p.y+q.y)\rfloor|\,)$$
$$= \quad \{\, r := (p+q)-2\lfloor \tfrac{1}{2}(p+q)\rfloor \,\}$$
$$\max(\,|2p.x-(p.x+q.x)+r.x|,\,|2p.y-(p.y+q.y)+r.y|\,)$$
$$\leqslant \quad \{\, |a+b| \leqslant |a|+|b| \text{ and Property 3.1(a) } \}$$
$$\max(\,|p.x-q.x|+|r.x|,\,|p.y-q.y|+|r.y|\,)$$
$$\leqslant \quad \{\, (\underline{\mathbf{A}}\, a : a \in \mathbf{Z} : a-2\left\lfloor \frac{a}{2} \right\rfloor \in \{0,1\}) \text{ and definition of } r \text{ and Property 3.1(a) } \}$$
$$\max(\,|p.x-q.x|+1,\,|p.y-q.y|+1\,)$$
$$= \quad \{\text{ Property 3.1(b) } \}$$
$$\max(\,|p.x-q.x|,\,|p.y-q.y|\,)+1$$
$$= \quad \{\text{ definition of } d \}$$
$$d(p,q)+1.$$

The other inequality may be proven in a similar way.

□

Property 3.10:

$$d(p,q) \leqslant d(p,sfl(p,q))+d(sfl(p,q),q) \leqslant d(p,q)+1.$$

Proof:

Since $sfl(p,q)=sfl(q,p)$, Property 3.9 also holds for $d(sfl(p,q),q)$, hence

$$2*d(sfl(p,q),q) \leqslant d(p,q)+1.$$

Then

$$d(p,sfl(p,q))+d(sfl(p,q),q)$$

$\leqslant$ { Property 3.9 and $d(v,w) = d(w,v)$ }

$\frac{1}{2}(d(p,q)+1) + \frac{1}{2}(d(p,q)+1)$

$=$ $d(p,q) + 1.$

The other inequality follows from the triangle inequality of d.

□

Corollary 3.11:

If $d(p,q) > 1$ and $r = sfl(p,q)$, then

$$0 < d(p,r) < d(p,q) \ \wedge \ 0 < d(r,q) < d(p,q).$$

□

Corollary 3.12:

If $r = sfl(p,q)$, then

$$d(p,q) \text{ even} \Rightarrow d(p,r) = d(r,q) = \tfrac{1}{2}d(p,q)$$

$$d(p,q) \text{ odd} \Rightarrow (d(p,r) = \tfrac{1}{2}(d(p,q)+1) \ \vee \ d(r,q) = \tfrac{1}{2}(d(p,q)+1)).$$

□

3.2 The Corthout-Jonkers function

3.2.0 Definition

As explained before, the line functions presented in this chapter differ in the definition of the split point. The first function is based on the function sfl.

The Corthout-Jonkers function $f : \mathbf{Z}^2 \times \mathbf{Z}^2 \rightarrow \boldsymbol{P}(\mathbf{Z}^2)$ is defined by

$$f[p,q] := \begin{cases} \{p,q\} & \text{if } d(p,q) \leqslant 1 \\ f[p, sfl(p,q)] \cup f[sfl(p,q), q] & \text{if } d(p,q) > 1. \end{cases}$$

Some examples of pixel sets generated by this function are shown in Figure 3.1.

In [Corthout and Jonkers 1986a] a point containment algorithm is presented for Bezier shapes on a discrete grid. Their argument for using discrete shapes only is the wish for robustness; when continuous shapes are used one is at the mercy of the restricted machine

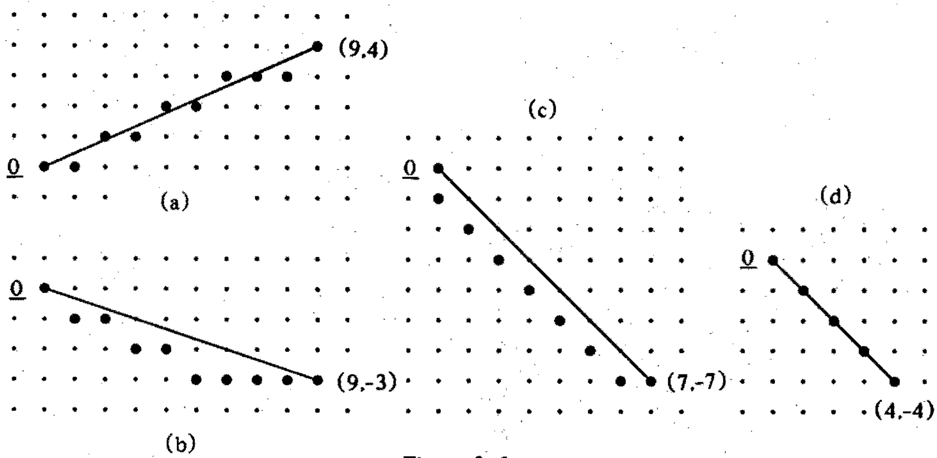


Figure 3. 1

Some examples of Corthout-Jonkers lines.

a) $f[\underline{0},(9,4)]$ *b)* $f[\underline{0},(9,-3)]$
c) $f[\underline{0},(7,-7)]$ *d)* $f[\underline{0},(4,-4)]$

precision in real-arithmetic. They define Bezier curves of arbitrary order in integer space by means of recursion, based on properties of Bezier curves in continuous space. The line function presented above is their Bezier curve of order 1.

We shall prove that the Corthout-Jonkers function is a translation invariant line function, which is neither minimal, nor convex, nor close. An upper bound for the deviation will be derived that is logarithmic in $d(p,q)$; this upper bound will be shown to be strict for particular values of $d(p,q)$.

Because of the recursive nature of the definition, all proofs concerning the Corthout-Jonkers function are based on induction on $d(p,q)$. As an introduction to these kinds of proofs, the following property, though seemingly trivial, is proved formally.

Property 3.13:

f is a line function.

Proof:

We have to prove that f satisfies both lf0 and lf1, that is, lf0) $\{p,q\} \subseteq f[p,q]$

lf1) $f[p,q]$ is finite and connected.

Proof of lf0)

- If $d(p,q) \leqslant 1$, then, by definition of f, $\{p,q\} \subseteq f[p,q]$.
- Let $n \geqslant 1$, and assume

  $$(\underline{\mathrm{A}}\, p,q : d(p,q) \leqslant n : \{p,q\} \subseteq f[p,q]).$$

  Let p and q be such that $d(p,q) = n+1$.
  By definition of f,

  $$f[p,q] = f[p, s\mathit{fl}(p,q)] \cup f[s\mathit{fl}(p,q), q].$$

  Because of Corollary 3.11, the induction assumption may be applied to both $f[p, s\mathit{fl}(p,q)]$ and $f[s\mathit{fl}(p,q), q]$, hence $p \in f[p, s\mathit{fl}(p,q)]$ and $q \in f[s\mathit{fl}(p,q), q]$. Consequently,

  $$\{p,q\} \subseteq f[p,q].$$

Proof of lf1)

- If $d(p,q) \leqslant 1$, then, by definition of f, $f[p,q] = \{p,q\}$. Since $\{p,q\}$ is both finite and connected, $f[p,q]$ is finite and connected.
- Let $n \geqslant 1$, and assume

  $$(\underline{\mathrm{A}}\, p,q : d(p,q) \leqslant n : f[p,q] \text{ is finite and connected}).$$

  Let p and q be such that $d(p,q) = n+1$.
  By definition of f,

  $$f[p,q] = f[p, s\mathit{fl}(p,q)] \cup f[s\mathit{fl}(p,q), q].$$

  Because of Corollary 3.11, the induction assumption may be applied to both $f[p, s\mathit{fl}(p,q)]$ and $f[s\mathit{fl}(p;q), q]$, hence $f[p, s\mathit{fl}(p,q)]$ is finite and connected and $f[s\mathit{fl}(p,q), q]$ is finite and connected. Furthermore, because of lf0, $s\mathit{fl}(p,q)$ is contained in both $f[p, s\mathit{fl}(p,q)]$ and $f[s\mathit{fl}(p,q), q]$. Consequently, $f[p,q]$ is finite and connected.

□

Since

$$s\mathit{fl}((r+p),(r+q)) = r + s\mathit{fl}(p,q),$$

the function $s\mathit{fl}$ is invariant under translation, and thus it can be seen that the following property holds.

Property 3.14:

f is translation invariant.

□

f is not minimal, as can be seen in Figure 3.1(c): here $\#f[p,q] = 9$, whereas $d(p,q) = 7$. f is not convex, as is illustrated in Figure 3.2: (2,0) and (6,2) are both elements of $f[\underline{0},(7,3)]$, whereas $f[(2,0),(6,2)]$ is not a subset of $f[\underline{0},(7,3)]$.

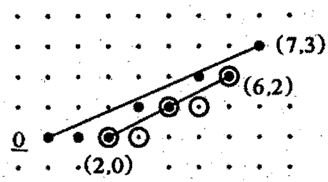


Figure 3. 2

Illustration that the Corthout-Jonkers function is not convex. The elements of $f[\underline{0},(7,3)]$ and $f[(2,0),(6,2)]$ are indicated by • and ○ respectively.

f is not close, as can be seen, for example, in Figure 3.1(c): for $v = (1,-1)$, which is an element of $[\underline{0},(7,-7)]$, no $r \in f[\underline{0},(7,-7)]$ exists such that $d(r,v) < 1$.

3.2.1 Deviation

As said previously, the Corthout-Jonkers function is not close. We shall first derive an upper bound for its deviation, and subsequently show that for some values of n this upper bound is strict.

Recall that the deviation function E_f is defined by

$$E_f(n) = (\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q) = n : e0_f[p,q]),$$

where

$$e0_f[p,q] = (\underline{\max}\, r : r \in f[p,q] : \tilde{d}(r,[p,q])).$$

3.2.1.0 Upper bound

To start with, we associate with each element of $f[p,q]$ a number, the so-called *level* number. Consider, for this purpose, the following derivation (in the sequel referred to as Derivation CJ), obtained by repeatedly applying the definition of f.

$$
\begin{array}{ll}
 & f[p,q] \\
= & \quad \{ r_1 := \lfloor ½(p+q) \rfloor \} \\
 & f[p,r_1] \cup f[r_1,q] \\
= & \quad \{ r_2 := \lfloor ½(p+r_1) \rfloor \text{ and } r_3 := \lfloor ½(r_1+q) \rfloor \} \\
 & f[p,r_2] \cup f[r_2,r_1] \cup f[r_1,r_3] \cup f[r_3,q] \\
= & \quad \{ r_4 := \lfloor ½(p+r_2) \rfloor \text{ and } r_5 := \lfloor ½(r_2+r_1) \rfloor \text{ and} \\
 & \quad\; r_6 := \lfloor ½(r_1+r_3) \rfloor \text{ and } r_7 := \lfloor ½(r_3+q) \rfloor \} \\
 & f[p,r_4] \cup f[r_4,r_2] \cup f[r_2,r_5] \cup f[r_5,r_1] \cup \\
 & f[r_1,r_6] \cup f[r_6,r_3] \cup f[r_3,r_7] \cup f[r_7,q] \\
= & \quad \ldots\ldots \\
 & \ldots\ldots
\end{array}
$$

For this line function and this p and q, p and q are said to have level number 0, r_1 level number 1, r_2 and r_3 level number 2, r_4, r_5, r_6, and r_7 level number 3, and so forth. This is illustrated below, where the line segment $[p,q]$ is supposed to be horizontal, i.e., $p.y = q.y$.

level number:	0	3	2	3	1	3	2	3	0
pixel:	p	r_4	r_2	r_5	r_1	r_6	r_3	r_7	q

Note that the level of a pixel r_i, where r_i is the split point of $[r_j, r_k]$, equals

$1 + \max(\text{level of } r_j, \text{level of } r_k)$,

and that r_j and r_k have different levels for $r_i \neq r_1$.

For $p,q \in \mathbf{Z}^2$, $p \neq q$, let $lv(p,q)$ denote the maximum level of any pixel in $f[p,q]$; $lv(p,q)$ depends on $d(p,q)$, as is expressed in the following property.

Property 3.15:

For any $p,q \in \mathbf{Z}^2$, $p \neq q$,

$$lv(p,q) = \left\lceil {}^2\log d(p,q) \right\rceil . \qquad \text{(a)}$$

Proof:

We shall use induction on $d(p,q)$.

- Suppose $d(p,q)=1$. Then, ${}^2\log d(p,q)=0$.
  Also, by definition of f,

  $$f[p,q] = \{p,q\}.$$

  Hence, the two pixels of $f[p,q]$ have level 0, and thus (a) is satisfied.
- Suppose $n \in \mathbf{N}, n \geqslant 1$, and assume that for all p and q with $1 \leqslant d(p,q) \leqslant n$, (a) holds.
  Let $p,q \in \mathbf{Z}^2$, such that $d(p,q)=n+1$. Then

$$
\begin{array}{ll}
 & lv(p,q) \\
= & \{ d(p,q)>1 \text{ and definition of } f \text{ and definition of level number} \} \\
 & 1+\max(lv(p,sfl(p,q)), lv(sfl(p,q),q)) \\
= & \{ d(p,q)>1 \text{ and Corollary 3.11 and the induction assumption} \} \\
 & 1+\max(\lceil {}^2\log d(p,sfl(p,q)) \rceil, 2logdsflq) \\
= & \{ \text{Corollary 3.12} \} \\
 & \begin{cases} 1+\lceil {}^2\log ½ d(p,q) \rceil & \text{if } d(p,q) \text{ even} \\ 1+\lceil {}^2\log ½ (d(p,q)+1) \rceil & \text{if } d(p,q) \text{ odd} \end{cases} \\
= & \{ \text{properties of } {}^2\log \text{ and } \lceil\ \rceil \} \\
 & \begin{cases} \lceil {}^2\log d(p,q) \rceil & \text{if } d(p,q) \text{ even} \\ \lceil {}^2\log (d(p,q)+1) \rceil & \text{if } d(p,q) \text{ odd} \end{cases} \\
= & \{ d(p,q)>1 \text{ and } (i>1 \wedge i \text{ odd}) \Rightarrow \lceil {}^2\log(i+1) \rceil = \lceil {}^2\log i \rceil \} \\
 & \lceil {}^2\log d(p,q) \rceil .
\end{array}
$$

□

Now we introduce a notion for the maximum distance of any pixel of $f[p,q]$ of level number k, to $[p,q]$.
Formally, for $p,q \in \mathbf{Z}^2, p \neq q$, and $k \in \mathbf{N}, k \leqslant lv(p,q)$,

$$a_k(p,q) \ := \ (\underline{\max}\, r : r \in f[p,q] \wedge r \text{ has level } k : \tilde{d}(r,[p,q])).$$

We shall introduce an increasing sequence $(b_k)_{k \in \mathbf{N}}$, and show that for all $k \in \mathbf{N}$, and all $p,q \in \mathbf{Z}^2$, where $p \neq q$ and $lv(p,q) \geqslant k$,

$$a_k(p,q) \leqslant b_k . \tag{b}$$

In this case an upper bound for the deviation function E_f can be expressed in terms of $(b_k)_{k \in \mathbf{N}}$, as the following derivation shows.
Let $n \in \mathbf{N}$.

$E_f(n)$

$=$ { definition of E_f }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : e\,0_f[p,q])$

$=$ { definition of $e\,0_f[p,q]$ }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : (\underline{\max}\, r : r \in f[p,q] : \tilde{d}(r,[p,q])))$

$=$ { definition of $a_k(p,q)$ }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : (\underline{\max}\, k : k \in \mathbf{N} \wedge k \leqslant lv(p,q) : a_k(p,q)))$

$\leqslant$ { assuming (b) }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : (\underline{\max}\, k : k \in \mathbf{N} \wedge k \leqslant lv(p,q) : b_k))$

$\leqslant$ { assuming $(b_k)_{k \in \mathbf{N}}$ is increasing }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : b_{lv(p,q)})$

$=$ { Property 3.15 }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : b_{\lceil {}^2\log d(p,q) \rceil})$

$=$ { substitution }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : b_{\lceil {}^2\log n \rceil})$

$=$ { $b_{\lceil {}^2\log n \rceil}$ is independent of p and q }

$b_{\lceil {}^2\log n \rceil}$.

The sequence $(b_k)_{k \in \mathbf{N}}$ is defined by

$$b_0 := 0$$

$$b_1 := ½$$

$$b_{l+2} := ½(b_{l+1}+b_l) + ½, \text{ for } l \geqslant 0.$$

Before we prove that $a_k(p,q) \leqslant b_k$ for any $p,q \in \mathbf{Z}^2$ such that $lv(p,q) \geqslant k$, we first show that $(b_k)_{k \in \mathbf{N}}$ is an increasing sequence indeed.

Property 3.16:

For all $l \in \mathbf{N}$ holds,

$$b_l = \frac{l}{3} + \frac{(1-(-½)^l)}{9}. \qquad \text{(a)}$$

Proof:

By induction on l.

- For $l=0$ and $l=1$, it follows from the definition of b_0 and b_1 that (a) is satisfied.
- Let $l \geqslant 0$, and assume that for all $k \in \mathbf{N}$, $k \leqslant l+1$, b_k satisfies (a). Then

$$\begin{aligned}
& b_{l+2} \\
= & \quad \{ \text{definition of } (b_k)_{k \in \mathrm{N}} \} \\
& ½(b_{l+1}+b_l) + ½ \\
= & \quad \{ \text{induction assumption} \} \\
& ½\left(\frac{l+1}{3} + \frac{(1-(-½)^{l+1})}{9} + \frac{l}{3} + \frac{(1-(-½)^l)}{9}\right) + ½ \\
= & \quad \{ \text{arithmetic} \} \\
& \frac{l+2}{3} + \frac{(1-(-½)^{l+2})}{9}.
\end{aligned}$$

Hence b_{l+2} satisfies (a).

□

Note that the above property implies that $(b_k)_{k \in \mathrm{N}}$ is indeed increasing. This will be used in the proof of the following property.

Property 3.17:

For all $k \in \mathbf{N}$, and all $p, q \in \mathbf{Z}^2$, where $lv(p,q) \geqslant k$, holds

$$a_k(p,q) \leqslant b_k. \qquad \text{(b)}$$

Proof:

We shall distinguish three cases: $k=0$, $k=1$, and $k>1$.

- Let $k=0$.
  For any $p, q \in \mathbf{Z}^2$, the only elements of $f[p,q]$ with level 0 are p and q itself. Hence, $a_0(p,q) = 0$ for any p and q, and consequently, $a_0(p,q) \leqslant b_0$.
- Let $k=1$.
  Let $p, q \in \mathbf{Z}^2$, such that $d(p,q) > 1$.
  $f[p,q]$ contains only one element of level 1, namely the split point $r = sfl(p,q) = \lfloor v \rfloor$, where $v = ½(p+q)$. Because of the definition of $\tilde{d}$,

$$\tilde{d}(r,[p,q]) \leqslant d(r,v),$$

and since

$$d(r,v) = \max(\,|r.x-v.x\,|,\,|r.y-v.y\,|\,) \leqslant ½,$$

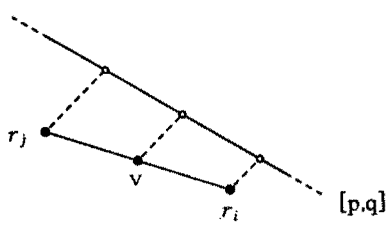
it follows that

$$a_1(p,q) = \tilde{d}(r,pq) \leqslant d(r,v) \leqslant ½ = b_1.$$

- Let $k>1$, and assume that for all $l \in \mathbf{N}$, $l<k$, and for all p,q such that $lv(p,q) \geqslant l$, (b) is satisfied.
  Let $p,q \in \mathbf{Z}^2$, such that $lv(p,q) \geqslant k$, and let $r \in f[p,q]$ have level k.
  Suppose r is the split point of $f[r_i,r_j]$, where split point r_i has level i and r_j has level j. Then $r = \lfloor ½(r_i+r_j)\rfloor$. Note that, with Derivation CJ, r_i and r_j have different levels numbers, which are both smaller than k. Let $v = ½(r_i+r_j)$. Then, using Property 2.2,

$$\tilde{d}(v,[p,q]) = ½(\tilde{d}(r_i,[p,q]) + \tilde{d}(r_j,[p,q])), \qquad \text{(c)}$$

as is illustrated below.



Since

$½(\tilde{d}(r_i,[p,q]) + \tilde{d}(r_j,[p,q]))$

$\leqslant$ { definition of $a_k(p,q)$ }

$½(a_i(p,q) + a_j(p,q))$

$\leqslant$ { induction assumption }

$½(b_i + b_j)$

$\leqslant$ { $(b_k)_{k\in\mathbf{N}}$ is increasing and $i \neq j$ and $i<k$ and $j<k$ }

$½(b_{k-1} + b_{k-2})$.

it follows from (c) that

$$\tilde{d}(v,[p,q]) \leqslant ½(b_{k-1} + b_{k-2}). \qquad \text{(d)}$$

Since

$$\tilde{d}(r,[p,q]) \leqslant \tilde{d}(v,[p,q]) + d(v,r),$$

and $d(v,r) \leqslant \frac{1}{2}$, it follows that

$$\tilde{d}(r,[p,q]) \leqslant \tilde{d}(v,[p,q]) + \tfrac{1}{2}. \tag{e}$$

Then we may derive the following.

$\tilde{d}(r,[p,q])$

$\leqslant$ $\{$ (e) $\}$

$\tilde{d}(v,[p,q]) + \frac{1}{2}$

$\leqslant$ $\{$ (d) $\}$

$\frac{1}{2}(b_{k-1} + b_{k-2}) + \frac{1}{2}$

$=$ $\{$ definition of b_k $\}$

b_k. (Derivation A.)

Since this holds for any $r \in f[p,q]$ that has level k, we have proven that

$$a_k(p,q) \leqslant b_k.$$

□

Since for all $n \in \mathbf{N}$,

$$E_f(n) \leqslant b_{\lceil {}^2\log n \rceil},$$

and

$$b_l = \frac{l+2}{3} + \frac{(1-(-\frac{1}{2})^{l+2})}{9},$$

we may conclude that we have found an upper bound for $E_f(n)$ which is logarithmic in n. Compared to the non-close functions presented in the previous chapter, this is a substantial improvement. In the following subsection we shall show that this upper bound is strict for some values of n.

3.2.1.1 Upper bound is strict

In this section we shall show that for all $k \in \mathbf{N}$ a pair of pixels p,q exists such that

$$a_l(p,q) = b_l \quad \text{for all } l: 0 \leq l \leq k.$$

For such p and q,

$$E(d(p,q)) = b_k.$$

We assume that $p.x \leq q.x$, and define $\Delta x(p,q) := q.x - p.x$ and $\Delta y(p,q) := q.y - p.y$. Suppose $r \in f[p,q]$ has level number k and is such that

$$a_k(p,q) = \tilde{d}(r,[p,q]).$$

Let r be the split point of $f[r_i,r_j]$, where split point r_i has level i and r_j has level j. Then $r = \lfloor v \rfloor$, where $v = ½(r_i + r_j)$.
Now consider Derivation A of the previous subsection. In order to arrive at

$$\tilde{d}(r,[p,q]) = b_k,$$

the following two conditions must hold.

$$\tilde{d}(r,[p,q]) = \tilde{d}(v,[p,q]) + ½ \tag{a.0}$$

and

$$\tilde{d}(v,[p,q]) = ½(b_{k-1} + b_{k-2}). \tag{a.1}$$

Analysing all possible positions of r relative to v (see Figure 3.3), it follows that (a.0) is satisfied if the slope of line segment $[p,q]$ is negative, and if $r = v - (½,½)$, hence, p and q must satisfy

$$\frac{\Delta y(p,q)}{\Delta x(p,q)} < 0 \ \wedge \ p.x + q.x \text{ odd} \ \wedge \ p.y + q.y \text{ odd}. \tag{b.0}$$

Condition (a.1) implies that r_i and r_j should be of level $k-1$ and $k-2$, and

$$\tilde{d}(r_i,[p,q]) = a_{k-1}(p,q) = b_{k-1} \ \wedge \ \tilde{d}(r_j,[p,q]) = a_{k-2}(p,q) = b_{k-2}. \tag{b.1}$$

By repeating these considerations for r_i and r_j, it follows that two different sequences of split points may maximise the distance to $[p,q]$, namely the sequence $r_1, r_2, r_5, \ldots$ (see Derivation CJ), and the sequence $r_1, r_3, r_6, \ldots$ We shall investigate the sequence $r_1, r_2, r_5, \ldots$ first.

Since for all integers a and b,

$$a + b \text{ even} \iff a - b \text{ even},$$

it follows from (b.0) that p and q must satisfy

$$\Delta x(p,q) = 2k_x + 1 \ \wedge \ \Delta y(p,q) = -(2k_y + 1), \text{ for some } k_x, k_y \in \mathbf{N}. \tag{c}$$

Furthermore, since $r_1 = \lfloor ½(p+q) \rfloor = \lfloor ½(p + (\Delta x(p,q), \Delta y(p,q))) \rfloor$,

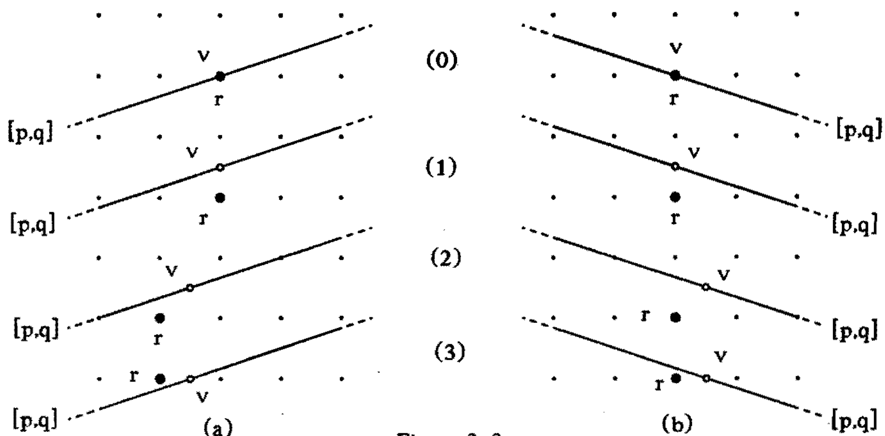


Figure 3. 3

The admitted positions of $r = \lfloor v \rfloor$ *relative to* $v = \frac{1}{2}(p+q)$.

$$a)\ \frac{q.y-p.y}{q.x-p.x} > 0 \qquad b)\ \frac{q.y-p.y}{q.x-p.x} < 0$$

$$0)\ r = v \qquad 1)\ r = v-(\tfrac{1}{2},0) \qquad 2)\ r = v-(\tfrac{1}{2},\tfrac{1}{2}) \qquad 3)\ r = v-(0,\tfrac{1}{2})$$

$$r_1 = \lfloor \tfrac{1}{2}(p.x+p.x+2k_x+1,\ p.y+p.y-(2k_y+1)) \rfloor = (p.x+k_x,\ p.y-(k_y+1)).$$

Similarly, in order to satisfy condition (b.0) for $r_2 = \lfloor \frac{1}{2}(p,r_1) \rfloor$, both $p.x+r_1.x$ and $p.y+r_1.y$ should be odd, which is equivalent to k_x and k_y+1 are odd. Hence,

$$k_x = 2l_x+1 \ \wedge\ k_y = 2l_y, \ \text{ for some } l_x, l_y \in \mathbf{N}. \tag{d}$$

and

$$r_2 = \lfloor \tfrac{1}{2}(p.x+p.x+2l_x+1,\ p.y+p.y-(2l_y+1)) \rfloor = (p.x+l_x,\ p.y-(l_y+1)).$$

Since

$$r_5 = \lfloor \tfrac{1}{2}(r_2+r_1) \rfloor,$$

a necessary condition for r_5 to satisfy condition (b.0) is that both $r_2.x+r_1.x$ and $r_2.y+r_1.y$ are odd, which is equivalent to l_x+1 and l_y are odd. Hence,

$$l_x = 2m_x \ \wedge\ l_y = 2m_y+1, \ \text{ for some } m_x, m_y \in \mathbf{N}. \tag{e}$$

Consequently,

$\Delta x\,(p,q),\ \Delta y\,(p,q)$

= { equation (c) }

$2k_x+1,\ -(2k_y+1)$

$=$ { equation (d) }

$1+2(2l_x+1), -(1+2(2l_y))$

$=$ { equation (e) }

$1+2+8m_x, -(1+4+8m_y)$.

Repeating the above reasoning, we arrive at the following statement.

Property 3.18:

Let $k \in \mathbf{N}$, and

$$\Delta x_k = 1 + (\underline{\text{sum}}\, i : 1 \leqslant i \leqslant \left\lfloor \frac{n+1}{2} \right\rfloor : 2^{2i-1}),$$

$$\Delta y_k = -(\underline{\text{sum}}\, i : 0 \leqslant i \leqslant \left\lfloor \frac{n}{2} \right\rfloor : 2^{2^i}).$$

Then for all p and q such that $q-p = (\Delta x_k, \Delta y_k)$, and all $l \in \mathbf{N}, 0 \leqslant l \leqslant k$,

$$a_l(p,q) = b_l.$$

Proof:

Define the sequence s_i by

$$s_i = \begin{cases} p & \text{if } i=0 \\ \lfloor \frac{1}{2}(p+q) \rfloor & \text{if } i=1 \\ \lfloor \frac{1}{2}(s_{i-1}+s_{i-2}) \rfloor & \text{if } 1<i \leqslant k. \end{cases}$$

Then it may be proven by induction on i that for all $i \in \mathbf{N}, 0 \leqslant i \leqslant n$,

$$\tilde{d}(s_i, [p,q]) = b_i.$$

□

In the table below b_k is shown for several values of k, together with the associated Δx_k and Δy_k values.

k	b_k			Δx_k	$-\Delta y_k$
1	$\frac{1}{2}$	=	0.5	3	1
2	$\frac{3}{4}$	=	0.75	3	5
3	$\frac{9}{8}$	≈	1.13	11	5
4	$\frac{23}{16}$	≈	1.44	11	21
5	$\frac{57}{32}$	≈	1.78	43	21
6	$\frac{135}{64}$	≈	2.11	43	85
7	$\frac{313}{128}$	≈	2.45	171	85
8	$\frac{711}{256}$	≈	2.78	171	341
9	$\frac{1593}{512}$	≈	3.11	683	341
10	$\frac{3527}{1024}$	≈	3.44	683	1365

As an example $f[\underline{0}, (11,-5)]$ is shown in Figure 3.4, together with s_1, s_2, and s_3.

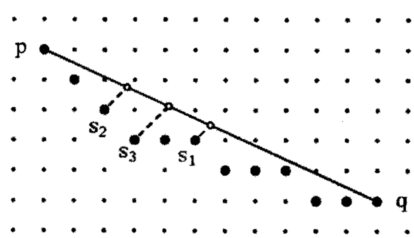


Figure 3. 4

Illustration of maximum distances at level 1, 2, *and* 3.

$p = \underline{0}$, $q = (11,-5)$, $s_1 = (5,-3)$, $s_2 = (2,-2)$, $s_3 = (3,-3)$.

Note that the digitised line segment also contains pixels of level 4 ((4,-3) for example), but that the distances of these pixels to $[p,q]$ are smaller than b_3.

A derivation similar to the one preceding Property 3.18 may be used to obtain expressions for $\Delta x(p,q)$ and $\Delta y(p,q)$ such that $r_1, r_3, r_6, \ldots$ have maximal distances to $[p,q]$. It turns out that in this case the absolute values of Δx_k and Δy_k are interchanged compared with the first case, as is stated in the following property.

Property 3.19:

Let $k \in \mathbf{N}$, and

$$\Delta x_k = (\underline{\text{sum}}\, i : 0 \leqslant i \leqslant \left\lfloor \frac{n}{2} \right\rfloor : 2^{2i}),$$

$$\Delta y_k = -(1 + (\underline{\text{sum}}\, i : 1 \leqslant i \leqslant \left\lfloor \frac{n+1}{2} \right\rfloor : 2^{2i-1}))$$

Then for all p and q such that $q - p = (\Delta x_k, \Delta y_k)$, and all $l \in \mathbf{N}, 0 \leqslant l \leqslant k$,

$$a_l(p, q) = b_l.$$

□

In Figure 3.5 $f[\underline{0}, (21,-11)]$ is shown, together with s_1, s_2, s_3, and s_4.

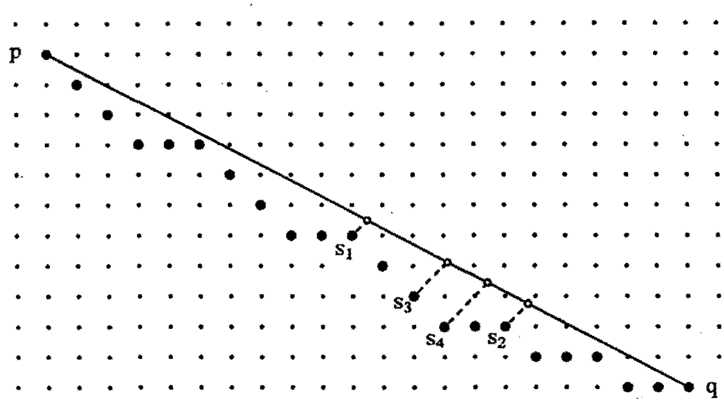


Figure 3. 5

Illustration of maximum distances at level 1, 2, 3, *and* 4.

$p = \underline{0}$, $q = (21, -11)$.
$s_1 = (10, -6)$, $s_2 = (15, -9)$, $s_3 = (12, -8)$, $s_4 = (13, -9)$.

Note that the digitised line segment also contains pixels of level 5 ((14,-9) for instance), but that these have a smaller distance to $[p, q]$ than b_4.

3.3 The Adapted Corthout-Jonkers function

Unlike the Corthout-Jonkers function, the function presented in this section is minimal. The definition of the split point is based on *sdv*, as introduced in section 1.

3.3.0 Definition

The Adapted Corthout-Jonkers (in short: Adapted CJ) function $f:\mathbf{Z}^2\times\mathbf{Z}^2\rightarrow \boldsymbol{P}(\mathbf{Z}^2)$ is defined by

$$f[p,q] := \begin{cases} \{p,q\} & \text{if } d(p,q) \leqslant 1 \\ f[p,sdv(\hat{p},\hat{q})] \cup f[sdv(\hat{p},\hat{q}),q] & \text{if } d(p,q) > 1 \wedge \hat{p}.y \leqslant \hat{q}.y \\ f[p,sdv(\hat{q},\hat{p})] \cup f[sdv(\hat{q},\hat{p}),q] & \text{if } d(p,q) > 1 \wedge \hat{p}.y > \hat{q}.y. \end{cases}$$

Informally, the split point of $f[p,q]$, where NF(p,q), is rounded in the direction of the endpoint with the smallest y-coordinate. Some examples of pixel sets generated by this function are shown in Figure 3.6. Note that for $p,q\in\mathbf{Z}^2$, where NF(p,q) and $q-p\in O_0\cup O_1$, the Adapted CJ function generates the same pixel sets as the Corthout-Jonkers function.

Again, all proofs are based on induction on $d(p,q)$.

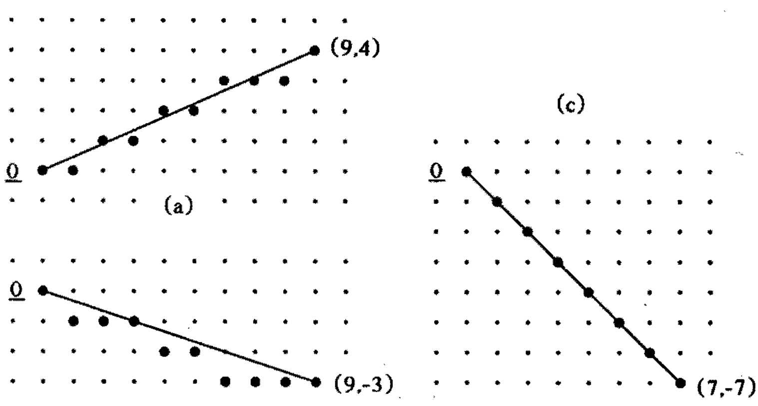


Figure 3.6

Some examples of Adapted CJ lines.

a) $f[\underline{0},(9,4)]$ *b)* $f[\underline{0},(9,-3)]$ *c)* $f[\underline{0},(7,-7)]$

Property 3.20:

f is a line function.

Proof:

The proof is the same as in Property 3.13, except that Corollary 3.7 should be used instead of Corollary 3.11.

□

The following property may be proven using

$$sdv((r+p),(r+q)) = r + sdv(p,q).$$

Property 3.21:

f is translation invariant.

□

f is not convex, as is illustrated by Figure 3.7: $f[(3,0),(7,2)]$ is not a subset of $f[\underline{0},(9,3)]$, although $(3,0)$ and $(7,2)$ are both elements of $f[\underline{0},(9,3)]$.

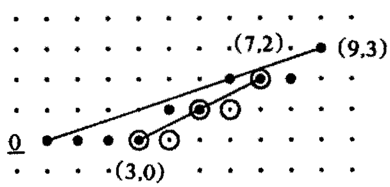


Figure 3.7

Illustration that the Adapted CJ function is not convex. The elements of $f[\underline{0},(9,3)]$ and $f[(3,0),(7,2)]$ are indicated by • and ○ respectively.

f is not close, as can be seen in Figure 3.6(a): for $v=(3,1)$, which is an element of $[\underline{0},(9,3)]$, no $r \in f[\underline{0},(9,3)]$ exists such that $d(r,v) < 1$.

3.3.1 Minimality

In this section we shall prove that the Adapted CJ function is minimal. We use the following properties, which are based on the definition of the Bounding Box from Chapter 2.

Property 3.22:

For all $p, q \in \mathbf{Z}^2$,

$$f[p,q] \subseteq \mathrm{BB}(p,q).$$

Proof:

- If $d(p,q) \leqslant 1$, then $f[p,q] = \{p,q\} \subseteq \mathrm{BB}(p,q)$.
- Let $n \geqslant 1$, and assume

$$(\underline{\mathbf{A}}\, p,q : d(p,q) \leqslant n : f[p,q] \subseteq \mathrm{BB}(p,q)).$$

Let p and q be such that $d(p,q) = n+1$, and $\hat{p}.y \leqslant \hat{q}.y$. Then

$$\begin{array}{ll}
 & f[p,q] \\
= & \{\, d(p,q) > 1 \text{ and definition of } f \,\} \\
 & f[p, sdv(\hat{p},\hat{q})] \cup f[sdv(\hat{p},\hat{q}), q] \\
= & \{\, \text{definition of } \hat{p},\hat{q} \text{ and } (\underline{\mathbf{A}}\, r,s : r,s \in \mathbf{Z}^2 : [r,s] = [s,r]) \,\} \\
 & f[\hat{p}, sdv(\hat{p},\hat{q})] \cup f[sdv(\hat{p},\hat{q}), \hat{q}] \\
\subseteq & \{\, d(p,q) > 1 \text{ and Corollary 3.11 and the induction assumption} \,\} \\
 & \mathrm{BB}(\hat{p}, sdv(\hat{p},\hat{q})) \cup \mathrm{BB}(sdv(\hat{p},\hat{q}), \hat{q}) \\
\subseteq & \{\, \hat{p}.x \leqslant sdv(\hat{p},\hat{q}).x \leqslant \hat{q}.x \text{ and } \hat{p}.y \leqslant sdv(\hat{p},\hat{q}).y \leqslant \hat{q}.y \text{ and definition of BB} \,\} \\
 & \mathrm{BB}(\hat{p},\hat{q}) \\
= & \{\, \text{definition of } \hat{p},\hat{q} \text{ and definition of BB} \,\} \\
 & \mathrm{BB}(p,q).
\end{array}$$

If $\hat{p}.y > \hat{q}.y$, then $f[p,q] \subseteq \mathrm{BB}(p,q)$ may be proven in a similar way.

□

Corollary 3.23:

For all $p, q \in \mathbf{Z}^2$ holds,

$$\hat{p}.y \leqslant \hat{q}.y \;\Rightarrow\; sdv(\hat{p},\hat{q}) \in \mathrm{BB}(p,q)$$

$$\hat{p}.y < \hat{q}.y \;\Rightarrow\; sdv(\hat{q}, \hat{p}) \in \mathrm{BB}(p, q)$$

□

Property 3.24:

For all $p, q \in \mathbf{Z}^2$ and all $r \in \mathrm{BB}(p, q)$ holds

$$\mathrm{BB}(p, r) \cap \mathrm{BB}(r, q) = \{r\}. \qquad \text{(a)}$$

Proof:

From Section 2.4.1 we know that BB is translation invariant. Therefore we may assume that $p = \underline{0}$ and $q.x \geqslant 0$.
Let $q \in O_0 \cup O_1$.
Let $r \in \mathrm{BB}(\underline{0}, q)$. From the definition of BB it then follows that

$$0 \leqslant r.x \leqslant q.x.$$

Suppose $s \in \mathrm{BB}(\underline{0}, r) \cap \mathrm{BB}(r, q)$. Then

$$0 \leqslant s.x \leqslant r.x \;\wedge\; r.x \leqslant s.x \leqslant q.x,$$

hence $r.x = s.x$.
Similarly, $r.y = s.y$, thus $s = r$, and (a) is proven for $q \in O_0 \cup O_1$.

For $q \in O_6 \cup O_7$, (a) may be proven similarly.

□

Property 3.25:

For all $p, q \in \mathbf{Z}^2$ holds

$$\hat{p}.y \geqslant \hat{q}.y \;\Rightarrow\; f[p, sdv(\hat{p}, \hat{q})] \cap f[sdv(\hat{p}, \hat{q}), q] = \{sdv(\hat{p}, \hat{q})\}$$

$$\hat{p}.y < \hat{q}.y \;\Rightarrow\; f[p, sdv(\hat{q}, \hat{p})] \cap f[sdv(\hat{q}, \hat{p}), q] = \{sdv(\hat{q}, \hat{p})\}$$

Proof:

Suppose $\hat{p}.y \geqslant \hat{q}.y$.
Then,

$$f[p, sdv(\hat{p}, \hat{q})] \cap f[sdv(\hat{p}, \hat{q}), q]$$

$\subseteq$ { Corollary 3.23 and Property 3.24 }

$$\mathrm{BB}(p, sdv(\hat{p}, \hat{q})) \cap \mathrm{BB}(sdv(\hat{p}, \hat{q}), q)$$

$\subseteq$ { Property 3.24 }

$$\{sdv(\hat{p}, \hat{q})\}.$$

Since, according to condition lf0 of line functions,

$$sdv(\hat{p},\hat{q}) \in f[p, sdv(\hat{p},\hat{q})] \wedge sdv(\hat{p},\hat{q}) \in f[sdv(\hat{p},\hat{q}), q],$$

we have proven that

$$f[p, sdv(\hat{p},\hat{q})] \cap f[sdv(\hat{p},\hat{q}), q] = \{ sdv(\hat{p},\hat{q}) \}.$$

The second part of the property may be proven similarly.

□

Now we are ready to prove that f is minimal.

Property 3.26:

The Adapted CJ function is minimal.

Proof:

We have to prove that $\# f[p,q] = d(p,q) + 1$, for all p and q.

- If $d(p,q) \leqslant 1$, then $\# f[p,q] = \# \{p,q\} = d(p,q) + 1$.
- Let $n \geqslant 1$, and assume

$$(\underline{A}\, p,q : d(p,q) \leqslant n : \# f[p,q] = d(p,q) + 1).$$

Let p and q be such that $d(p,q) = n+1$, and $\hat{p}.y \leqslant \hat{q}.y$. Then

$$\begin{aligned}
& \# f[p,q] \\
= \quad & \{ d(p,q) > 1 \text{ and definition of } f \} \\
& \#(f[p, sdv(\hat{p},\hat{q})] \cup f[sdv(\hat{p},\hat{q}), q]) \\
= \quad & \{ \text{Property 3.25} \} \\
& \# f[p, sdv(\hat{p},\hat{q})] + \# f[sdv(\hat{p},\hat{q}), q] - 1 \\
= \quad & \{ d(p,q) > 1 \text{ and Corollary 3.7 and the induction assumption} \} \\
& (d(p, sdv(\hat{p},\hat{q})) + 1) + (d(sdv(\hat{p},\hat{q}), q) + 1) - 1 \\
= \quad & \{ \text{Property 3.6 and definition of } \hat{p}, \hat{q} \text{ and } d(r,s) = d(s,r) \} \\
& d(p,q) + 1.
\end{aligned}$$

If $\hat{p}.y > \hat{q}.y$, then it may be proven in a similar way that

$$f[p,q] = d(p,q) + 1.$$

□

From the previous chapter we know that digitisations of minimal line functions may be represented by chain codes. In the following chapter, where a convex line function will be constructed based on the Adapted CJ function, more will be said on the structure of the chain codes associated with the Adapted CJ function.

3.3.2 Deviation

As has been indicated already in section 3.3.0, the Adapted CJ function is not close. In this section we shall investigate its deviation function. It turns out that the same upper bound as for the Corthout Jonkers function may be derived, but unlike the Corthout Jonkers function, this upper bound is not strict.

Again, we associate with each element of $f[p,q]$ a level number, in the same way as we have done with the Corthout Jonkers function. Also, if $lv(p,q)$ denotes the maximum level of any pixel in $f[p,q]$, then $lv(p,q)$ depends on $d(p,q)$ in a way similar as with the Corthout Jonkers function.

Property 3.27:

For any $p,q \in \mathbf{Z}^2$, $p \neq q$,

$$lv(p,q) = \left\lceil {}^2\log d(p,q) \right\rceil .$$

Proof:

Similar to the proof of Property 3.15, except that Corollaries 3.7 and 3.8 must be used instead of 3.11 and 3.12.

□

Consider a line segment $[p,q]$, and its Adapted CJ digitisation $f[p,q]$. Since f is translation invariant, we may assume that $p=\underline{0}$ and $q.x \geqslant 0$.

Consider the point $v = \frac{1}{2}q$ halfway this segment.
If $q \in O_0 \cup O_1$, then for the split point r of $f[\underline{0},q]$, defined by $r = sdv(\underline{0},q)$, the following four positions with regard to v are possible:

$$r = \begin{cases} v & \text{if } q.x \text{ even} \wedge q.y \text{ even} \\ v-(\frac{1}{2},0) & \text{if } q.x \text{ odd} \wedge q.y \text{ even} \\ v-(\frac{1}{2},\frac{1}{2}) & \text{if } q.x \text{ odd} \wedge q.y \text{ odd} \\ v-(0,\frac{1}{2}) & \text{if } q.x \text{ even} \wedge q.y \text{ odd.} \end{cases}$$

If $q \in O_6 \cup O_7$, then for the split point $r = sdv(q,\underline{0})$ the following four relations with v are possible:

$$r = \begin{cases} v & \text{if } q.x \text{ even} \wedge q.y \text{ even} \\ v + (½, 0) & \text{if } q.x \text{ odd} \wedge q.y \text{ even} \\ v + (½, -½) & \text{if } q.x \text{ odd} \wedge q.y \text{ odd} \\ v + (0, -½) & \text{if } q.x \text{ even} \wedge q.y \text{ odd.} \end{cases}$$

In Figure 3.8(a) and (b) these possibilities are shown for $q \in O_0$ and $q \in O_7$ respectively.

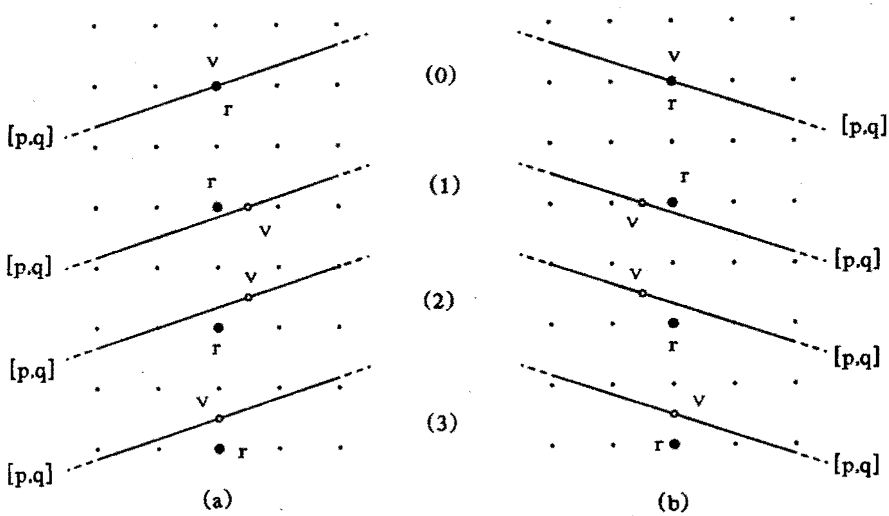


Figure 3. 8

The admitted positions of r relative to $v = \frac{1}{2}q$.

a) $r = sdv(\underline{0}, q)$ *b)* $r = sdv(q, \underline{0})$

0) $r = v$

1*a*) $r = v-(½, 0)$ 1*b*) $r = v+(½, 0)$

2*a*) $r = v-(½, ½)$ 2*b*) $r = v+(½, -½)$

3*a*) $r = v-(0, ½)$ 3*b*) $r = v+(0, -½)$

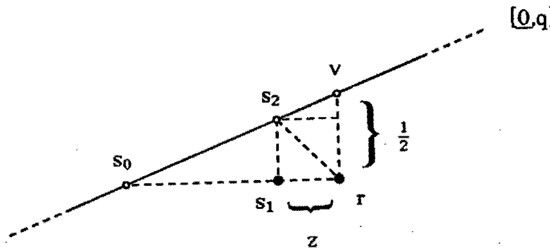
It can be seen that $\tilde{d}(r, [\underline{0}, q])$ is maximal for

- $r = v - (0, ½)$ if $q \in O_0$
- $r = v + (0, -½)$ if $q \in O_7$.

Then

$$\tilde{d}(r, [\underline{0}, q]) = \frac{1}{2(1+\alpha(q))}, \text{ where } \alpha(q) = \frac{|q.y|}{q.x}. \tag{a}$$

This will be shown for the case $q \in O_0$. See the figure below.



Here, applying Property 2.2, $\tilde{d}(r, [\underline{0}, q]) = z$. Since $v.y - r.y = ½$, and triangles $[s_0, r, v]$ and $[\underline{0}, (q.x, 0), q]$ are congruent, it follows that

$$r.x - s_0.x = \frac{q.y}{2*q.x}.$$

From the congruency of triangles $[s_0, s_1, s_2]$ and $[\underline{0}, (q.x, 0), q]$ the following equation may be derived.

$$z : (\frac{q.y}{2*q.x} - z) = q.y : q.x.$$

Hence,

$$z = \frac{q.x}{2(q.x + q.y)} = \frac{1}{2(1+\alpha(q))}.$$

and thus

$$\tilde{d}(r, [\underline{0}, q]) = \frac{1}{2(1+\alpha(q))}. \qquad \text{(b)}$$

Equation (b) also holds for $q \in O_7$. If $q \in O_1 \cup O_6$, then (b) holds for

$$\alpha(q) = \frac{q.x}{|q.y|}.$$

Since the maximal distance of the split point r to $[\underline{0}, q]$ depends on q, there is a complication in comparison with the Corthout Jonkers function, where $\tilde{d}(r, [\underline{0}, q])$ was maximally ½. However, since $0 \leqslant \alpha(q) \leqslant 1$, and hence

$$\frac{1}{4} \leqslant \frac{1}{2(1+\alpha(q))} \leqslant \frac{1}{2},$$

it follows immediately that the upper bound that we have derived in Section 3.2.1 for the deviation of the Corthout Jonkers function, is also an upper bound for the deviation of the Adapted CJ function.

Although we are able to prove, in a way quite similar as in Section 3.2.1 that for the Adapted CJ function

$$a_k(p,q) \leqslant c_k(p,q) \tag{c}$$

where $c_k(p,q)$ is defined by

$$c_0(p,q) := 0 \qquad c_1(p,q) := \frac{1}{2(1+\alpha(p,q))}$$

$$c_{k+2}(p,q) := ½(c_{k+1}+c_k) + \frac{1}{2(1+\alpha(p,q))}.$$

and satisfies

$$c_k(p,q) = \frac{1}{(1+\alpha(p,q))}\left(\frac{k}{3}+\frac{1-(-½)^k}{9}\right), \tag{d}$$

where $\alpha(p,q)$ is defined by

$$\alpha(p,q) := \frac{\min(\,|q.y-p.y\,|,\,|q.x-p.x\,|\,)}{\max(\,|q.y-p.y\,|,\,|q.x-p.x\,|\,)},$$

this does not help us in finding a smaller upper bound for the deviation $E_f(n)$:

$E_f(n)$

$=$ { definition of E_f }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : e0_f[p,q])$

$=$ { definition of $e0_f[p,q]$ }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : (\underline{\max}\, r : r \in f[p,q] : \tilde{d}(r,[p,q])))$

$=$ { definition of $a_k(p,q)$ }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : (\underline{\max}\, k : k \in \mathbf{N} \wedge k \leqslant lv(p,q) : a_k(p,q)))$

$\leqslant$ { equation (c) }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : (\underline{\max}\, k : k \in \mathbf{N} \wedge k \leqslant lv(p,q) : c_k(p,q)))$

$\leqslant$ { $(c_k(p,q))_{k \in [0..lv(p,q)]}$ is increasing }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : c_{lv(p,q)}(p,q))$

$=$ { Property 3.27 and substitution }

$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : c_{\lceil {}^2\log n \rceil}(p,q))$

$=$ { equation (d) }

$$(\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : \frac{1}{(1+\alpha(p,q))}\left(\frac{\lceil {}^2\log n \rceil}{3}+\frac{1-(-½)^{\lceil {}^2\log n \rceil}}{9}\right))$$

$=$ { property of $\underline{\sup}$ }

$$\left(\frac{\lceil {}^2\log n \rceil}{3}+\frac{1-(-½)^{\lceil {}^2\log n \rceil}}{9}\right) * (\underline{\sup}\, p,q : p,q \in \mathbf{Z}^2 \wedge d(p,q)=n : \frac{1}{1+\alpha(p,q)})$$

$=$ { definition of $\alpha(p,q)$ }

$$\left(\frac{\left\lceil {}^2\log n \right\rceil}{3} + \frac{1-(-\frac{1}{2})^{\left\lceil {}^2\log n \right\rceil}}{9}\right) * 1$$

$=$ { definition of $(b_k)_{k \in \mathbb{N}}$ }

$$b_{\left\lceil {}^2\log n \right\rceil}.$$

Still, if we compare the Adapted CJ and Corthout-Jonkers digitisations of the line segment $[\underline{0}, (11,-5)]$ (a segment for which the upper bound was strict in case of the Corthout-Jonkers function), then the Adapted CJ function is obviously better than the Corthout-Jonkers function. See Figure 3.9.

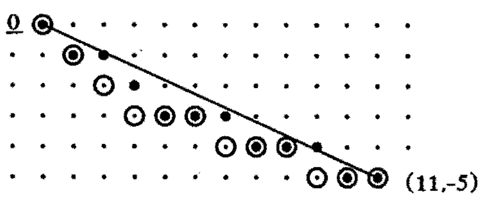


Figure 3.9

Digitisations of the line segment $[\underline{0}, (11,-5)]$.
The Corthout-Jonkers digitisation is indicated by ○ *and the Adapted CJ digitisation by* •.

3.4 The Symmetric function

The definition of the two previous functions was based on the principle

$$f[p,q] = \begin{cases} \{p,q\} & \text{if } d(p,q) \leqslant 1 \\ f[p,r] \cup f[r,q] & \text{if } d(p,q) > 1, \end{cases}$$

where for each function a different definition of the split point r was used. For both the Corthout-Jonkers and Adapted CJ function, $d(p,r) \neq d(r,q)$ if $d(p,q)$ is odd, in which case asymmetry is introduced. In the following function symmetry is preserved by the use of two split points, if necessary.

3.4.0 Definition

The Symmetric function f is defined by

$$f[p,q] := \begin{cases} \{p,q\} & \text{if } d(p,q) \leqslant 1 \\ f[p,sdv(p,q)] \cup f[sdv(q,p),q] & \text{if } d(p,q) > 1. \end{cases}$$

Note that $sdv(p,q)$ differs from $sdv(q,p)$ when $q.x+p.x$ or $q.y+p.y$ is odd. Note also that if in the above definition sdv is replaced by sfl, then f would be the Corthout-Jonkers function, since for all p and q holds that $sfl(p,q) = sfl(q,p)$.

In Figure 3.10 some examples of pixel sets generated by this function are shown . We shall prove that f is a translation invariant line function, which is not minimal, nor convex, nor close. Apart from the difference indicated above, it differs from the other two functions in its deviation: we shall prove that the Symmetric function has a deviation upper bound which is still logarithmic in $d(p,q)$, but smaller than the one of the Adapted CJ function.

Again, all proofs are based on induction on $d(p,q)$.

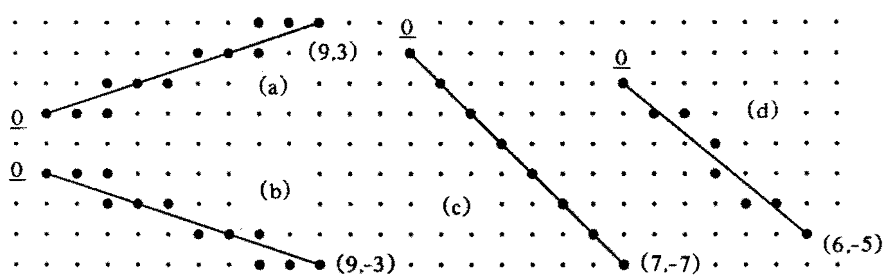


Figure 3. 10

Some examples of Symmetric lines.

a) $f[\underline{0},(9,3)]$ *b)* $f[\underline{0},(9,-3)]$
c) $f[\underline{0},(7,-7)]$ *d)* $f[\underline{0},(6,-5)]$

In order to prove that f is a line function, we shall use the following property.

Property 3.28:

For all $p,q \in \mathbf{Z}^2$.

$$d(sdv(p,q), sdv(q,p)) \leqslant 1.$$

Proof:

$d(sdv(p,q), sdv(q,p))$

$=$ { definition of sdv }

$d(p+(q-p)\underline{\text{div}}\,2, q+(p-q)\underline{\text{div}}\,2)$

$=$ { Property 1.0 }

$d(p-q, (p-q)\underline{\text{div}}\,2-(q-p)\underline{\text{div}}\,2)$

$=$ { Property 3.2(a) }

$d(p-q, 2*((p-q)\underline{\text{div}}\,2))$

$=$ { definition of $\underline{\text{mod}}\,2$ }

$d(p-q, (p-q)-((p-q)\underline{\text{mod}}\,2))$

$=$ { Property 1.0 }

$d((p-q)\underline{\text{mod}}\,2, \underline{0}))$

$\leqslant$ { Property 3.2(f) and definition of d }

1.

□

Property 3.29:

f is a line function.

Proof:

lf0) This part is exactly the same as the corresponding part in the proof of Property 3.29, except that Corollary 3.7 should be used instead of 3.11.

lf1) • If $d(p,q) \leqslant 1$, then $\{p,q\}$ is finite and connected, and hence, $f[p,q]$ is finite and connected.

• Let $n \geqslant 1$, and assume

$$(\underline{\mathbf{A}}\, p,q : d(p,q) \leqslant n : f[p,q] \text{ is finite and connected}).$$

Let p and q be such that $d(p,q) = n+1$.
By definition of f,

$$f[p,q] = f[p, sdv(p,q)] \cup f[sdv(q,p), q].$$

Because of Corollary 3.7, the induction assumption may be applied to both $f[p, sdv(p,q)]$ and $f[sdv(q,p), q]$, hence

$f[p, sdv(p,q)]$ is finite and connected $\wedge$

$f[sdv(q,p),q]$ is finite and connected.

Then, because of Property 3.28,

$f[p,sdv(p,q)] \cup f[sdv(q,p),q]$ is finite and connected.

Consequently,

$f[p,q]$ is finite and connected.

□

Since sdv is invariant under translation, i.e.,

$$sdv((r+p),(r+q)) = r + sdv(p,q),$$

the following property holds.

Property 3.30:

f is translation invariant.

□

f is not minimal, as can be seen in Figure 3.10: in (a) and (b) $\#f[p,q] = 12$, whereas $d(p,q) = 9$, and in (d) $\#f[p,q] = 8$, whereas $d(p,q) = 6$. In the following subsection an expression for the cardinality of $f[p,q]$ will be derived.

f is not convex, as is illustrated in Figure 3.11: (2,0) and (7,1) are both elements of $f[\underline{0},(9,2)]$, whereas $f[(2,0),(7,1)]$ is not a subset of $f[\underline{0},(9,2)]$.

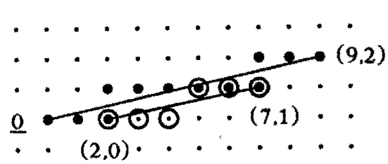


Figure 3. 11

Illustration that the Symmetric function is not convex. The elements of $f[\underline{0},(9,2)]$ and $f[(2,0),(7,1)]$ are indicated by • and ○ respectively.

In Figure 3.12 it can be seen that f is not close either: for $v = (4,-{}^{80}/_{27})$, which is an element of $[\underline{0},(27,-20)]$, no $r \in f[\underline{0},(27,-20)]$ exists such that $d(r,v) < 1$.

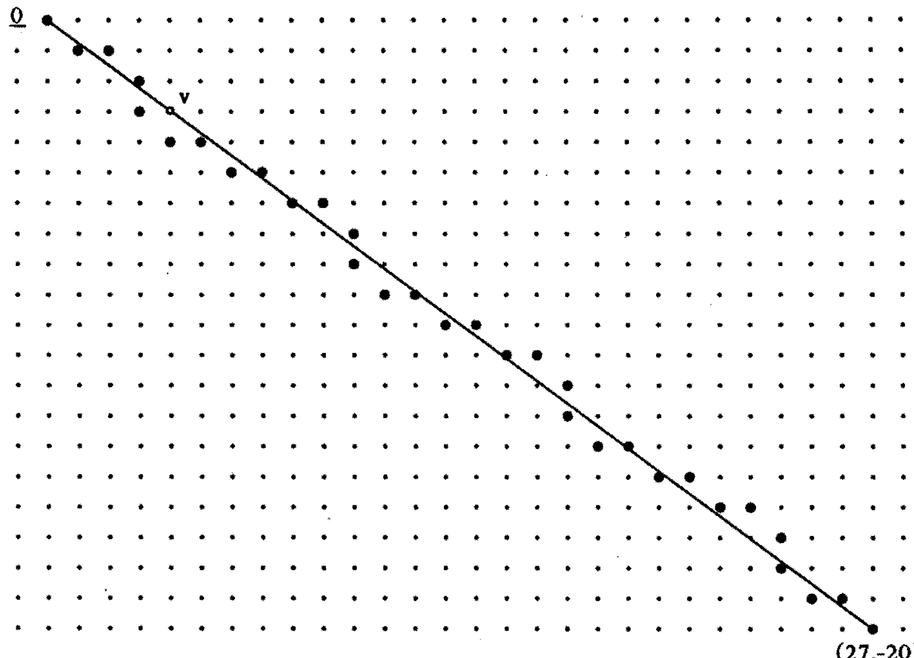


Figure 3. 12

Illustration that the Symmetric function is not close:
$v = (4,-^{80}/_{27}) \in [\underline{0}, (27,-20)]$, *whereas* $\tilde{d}(v, f([\underline{0}, (27,-20)]) > 1$.

3.4.1 Cardinality

As indicated in Section 4.0, the Symmetric function is not minimal. In this section we shall present an expression for the number of elements in $f[p,q]$. The following properties may be proven in a way similar to Property 3.22 and 3.24.

Property 3.31:

For all $p,q \in \mathbf{Z}^2$,

$$f[p,q] \subseteq \mathrm{BB}(p,q).$$

□

Property 3.32:

For all $p,q \in \mathbf{Z}^2$,

$$\mathrm{BB}[p\,, sdv(p\,,q)] \cap \mathrm{BB}[sdv(q\,,p), q] = \begin{cases} \varnothing & \text{if } sdv(p\,,q) \neq sdv(q\,,p) \\ \{ sdv(p\,,q) \} & \text{if } sdv(p\,,q) = sdv(q\,,p). \end{cases}$$

□

The operators $\neg$ and &, which occur in the properties below, have been introduced in Section 3.1.

Property 3.33:

For all $p\,,q \in \mathbf{Z}^2$,

$$\#(f[p\,, sdv(p\,,q)] \cap f[sdv(q\,,p), q]) = ((\neg\, |q.x - p.x\,|) \,\&\, (\neg\, |q.y - p.y\,|))\,\underline{\mathrm{mod}}\,2.$$

Proof:

From the previous two properties we may derive that

$$f[p\,, sdv(p\,,q)] \cap f[sdv(q\,,p), q]) \subseteq \begin{cases} \varnothing & \text{if } sdv(p\,,q) \neq sdv(q\,,p) \\ \{ sdv(p\,,q) \} & \text{if } sdv(p\,,q) = sdv(q\,,p). \end{cases}$$

According to condition lf0 of line functions, $sdv(p\,,q) \in f[p\,, sdv(p\,,q)]$, and thus, if $sdv(q\,,p) = sdv(p\,,q)$, then

$$sdv(p\,,q) \in f[p\,, sdv(p\,,q)] \cap f[sdv(q\,,p), q]).$$

Then we may derive that

$$\#(f[p\,, sdv(p\,,q)] \cap f[sdv(q\,,p), q])$$

$=$ { above reasoning }

$$\begin{cases} 0 & \text{if } sdv(p\,,q) \neq sdv(q\,,p) \\ 1 & \text{if } sdv(p\,,q) = sdv(q\,,p) \end{cases}$$

$=$ { Property 3.4 }

$$\begin{cases} 0 & \text{if } q.x - p.x \text{ even} \wedge q.y - p.y \text{ even} \\ 1 & \text{otherwise} \end{cases}$$

$=$ { definition of & and $\neg$ and case analysis }

$$((\neg\, |q.x - p.x\,|) \,\&\, (\neg\, |q.y - p.y\,|))\,\underline{\mathrm{mod}}\,2.$$

□

Now we arrive at the actual property about the cardinality of $f[p\,,q]$.

Property 3.34:

For all $p\,,q \in \mathbf{Z}^2$,

$$\# f[p,q] = d(p,q) + 1 + (\neg d(p,q)) \& k , \tag{a}$$

where

$$k = \min(|q.x - p.x|, |q.y - p.y|).$$

Proof:

- Suppose $d(p,q)=0$.
  Then $p=q$, and $\# f[p,q] = \#\{p\} = 1$.
  Also, since $\min(|q.x-p.x|, |q.y-p.y|) = 0$,

  $$d(p,q) + 1 + (\neg d(p,q)) \& k = 1,$$

  hence (a) is satisfied.
- Suppose $d(p,q)=1$.
  Then $\# f[p,q] = \#\{p,q\} = 2$.
  Also, since $\neg 1 = 0$,

  $$d(p,q) + 1 + (\neg d(p,q)) \& k = 1+1 = 2,$$

  hence (a) is satisfied.
- Let $n \in \mathbf{N}, n \geqslant 1$, and assume that for all $p,q \in \mathbf{Z}^2$ such that $d(p,q) \leqslant n$, (a) holds.
  Suppose $p,q \in \mathbf{Z}^2$, $d(p,q) = n+1$.
  Because f is translation invariant, we may assume that $p = \underline{0}$ and $q.x \geqslant 0$. Furthermore, without loss of generality, we assume that $|q.y| \leqslant q.x$. We shall prove that

  $$\# f[\underline{0},q] = q.x + 1 + (\neg q.x) \& |q.y| ,$$

  in which case (a) is satisfied.

  $\# f[\underline{0},q]$

  $=$ $\{ d(\underline{0},q) > 1$ and definition of f $\}$

  $\# (f[\underline{0}, sdv(\underline{0},q)] \cup f[sdv(q,\underline{0}),q])$

  $=$ $\{$ definition sdv and Property 3.2(a) $\}$

  $\# (f[\underline{0}, q \,\underline{\mathrm{div}}\, 2] \cup f[q - q \,\underline{\mathrm{div}}\, 2, q])$

  $=$ $\{$ Property 3.33 $\}$

  $\# f[\underline{0}, q \,\underline{\mathrm{div}}\, 2] + \# f[q - q \,\underline{\mathrm{div}}\, 2, q] - ((\neg q.x) \& (\neg |q.y|)) \,\underline{\mathrm{mod}}\, 2$

  $=$ $\{$ translation invariance of f $\}$

  $2 * \# f[\underline{0}, q \,\underline{\mathrm{div}}\, 2] - ((\neg q.x) \& (\neg |q.y|)) \,\underline{\mathrm{mod}}\, 2$

  $=$ $\{ d(\underline{0},q) > 1$ and Corollary 3.7 and the induction assumption $\}$

  $2(q.x \,\underline{\mathrm{div}}\, 2 + 1 + (\neg (q.x \,\underline{\mathrm{div}}\, 2)) \& (|q.y| \,\underline{\mathrm{div}}\, 2)) - ((\neg q.x) \& (\neg |q.y|)) \,\underline{\mathrm{mod}}\, 2$

  $=$ $\{$ Properties 3.2(g) and 3.2(i) $\}$

$$2(q.x\ \underline{\text{div}}\ 2)+2+2(((\neg q.x)\&|q.y|)\ \underline{\text{div}}\ 2)-((\neg q.x)\&(\neg|q.y|))\underline{\text{mod}}\ 2$$

$=$ { definition of $\underline{\text{mod}}\ 2$ }

$$q.x - q.x\ \underline{\text{mod}}\ 2 + 2 + (\neg q.x)\&|q.y| - ((\neg q.x)\&|q.y|)\underline{\text{mod}}\ 2 - i\ ((\neg q.x)\&(\neg|q.y|))\underline{\text{mod}}\ 2$$

$=$ { case analysis }

$$q.x + 1 + (\neg q.x)\&|q.y|.$$

Hence, (a) has been proven.

□

From the above property it follows that for $d(p,q)$ a power of 2, $\#f[p,q]$ is maximal for $k = d(p,q)-1$. Namely, for $d(p,q) = 2^n$,

$$\neg(d(p,q)) = 2^n - 1,$$

and hence $(\neg d(p,q))\&k$ is maximal for $k = 2^n - 1$. Furthermore, for $d(p,q) = 2^n - 1$, the number of elements in $f[p,q]$ equals $d(p,q)+1$, regardless of the value of k, because

$$\neg d(p,q) = 2^n,$$

and for any $k < 2^n$,

$$2^n \& k = 0.$$

3.4.2 Deviation

Again, we associate with each element of $f[p,q]$ a level number; if $f[p,q]$ has two split points, then these split points have both level number 1. The split point(s) of $f[p,sdv(p,q)]$ and $f[sdv(q,p),q]$ have level number 2, and so forth. Again, $lv(p,q)$ denotes the maximum level of any pixel in $f[p,q]$. $lv(p,q)$ depends on $z = {}^2\log d(p,q)$, but equals the floor of z, whereas for the Corthout-Jonkers and Adapted CJ functions, $lv(p,q)$ equals the ceiling of z.

Property 3.35:

For any $p,q \in \mathbf{Z}^2$, $p \neq q$,

$$lv(p,q) = \left\lfloor {}^2\log d(p,q) \right\rfloor. \tag{a}$$

Proof:

We shall use induction on $d(p,q)$.

- Suppose $d(p,q)=1$. Then, ${}^2\log d(p,q)=0$.
  Also, by definition of f,

  $$f[p,q]=\{p,q\}.$$

  Hence, the two pixels of $f[p,q]$ have level 0, and thus (a) is satisfied.
- Suppose $n \in \mathbf{N}, n \geqslant 1$, and assume that for all p and q with $1 \leqslant d(p,q) \leqslant n$, (a) holds.
  Let $p,q \in \mathbf{Z}^2$, such that $d(p,q)=n+1$. Then

$$
\begin{aligned}
& lv(p,q) \\
= \quad & \{\, d(p,q)>1 \text{ and definition of } f \text{ and definition of level number} \,\} \\
& 1+\max(lv(p,sdv(p,q)),lv(sdv(q,p),q)) \\
= \quad & \{\, d(p,q)>1 \text{ and Corollary 3.7 and the induction assumption} \,\} \\
& 1+\max(\lfloor {}^2\log d(p,sdv(p,q))\rfloor, \lfloor {}^2\log d(sdv(q,p),q)\rfloor) \\
= \quad & \{\, \text{Property 3.5} \,\} \\
& 1+\lfloor {}^2\log(d(p,q)\,\underline{\text{div}}\,2)\rfloor \\
= \quad & \{\, \text{definition of } \underline{\text{div}} \,\} \\
& \begin{cases} 1+\lfloor {}^2\log(\tfrac{1}{2}d(p,q))\rfloor & \text{if } d(p,q) \text{ even} \\ 1+\lfloor {}^2\log(\tfrac{1}{2}(d(p,q)-1))\rfloor & \text{if } d(p,q) \text{ odd} \end{cases} \\
= \quad & \{\, \text{properties of } {}^2\log \text{ and } \lceil\ \rceil \,\} \\
& \begin{cases} \lfloor {}^2\log d(p,q)\rfloor & \text{if } d(p,q) \text{ even} \\ \lfloor {}^2\log(d(p,q)-1)\rfloor & \text{if } d(p,q) \text{ odd} \end{cases} \\
= \quad & \{\, d(p,q)>1 \text{ and } (i>1 \wedge i \text{ odd}) \Rightarrow \lfloor {}^2\log(i-1)\rfloor = \lfloor {}^2\log i\rfloor \,\} \\
& \lfloor {}^2\log d(p,q)\rfloor .
\end{aligned}
$$

□

In a way quite similar as at the end of Section 3.3.2, an upper bound for $E_f(n)$ may be derived, namely

$$E_f(n) \leqslant b_{\lfloor {}^2\log n \rfloor}.$$

Because of the floor function, this is indeed a small improvement compared to the deviation function of the Corthout-Jonkers function. However, if we compare the Symmetric and Corthout-Jonkers digitisations of the line segment $[\underline{0}, (11,-5)]$ (a segment for which the upper bound was strict in case of the Corthout-Jonkers function), then the Symmetric function is obviously better than the Corthout-Jonkers function. See Figure 3.13.

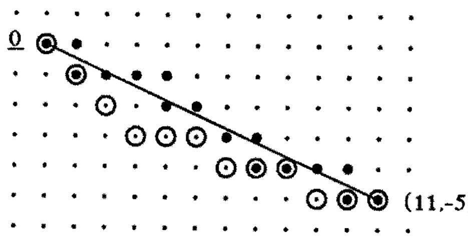


Figure 3. 13

Digitisations of the line segments $[\underline{0}, (11,-5)]$.
The Corthout-Jonkers digitisation is indicated by ○ *and the Symmetric digitisation by* •.

We conjecture that the above upper bound for the deviation of the Symmetric function may be sharpened substantially.

3.5 Concluding remarks

In this chapter we have introduced three recursive line functions. Because they are based on simple integer operations, they are candidates for fast hardware implementations. Furthermore, the recursive nature may be exploited by the use of parallel processors, which would speed up the generation time substantially.

A drawback of these functions is the absence of closeness. This is due to the accumulation of rounding errors. For the deviation function of the Corthout-Jonkers function it is proven that

$$E_f(n) \leqslant \frac{l}{3} + \frac{(1-(-\tfrac{1}{2})^l)}{9}, \tag{a}$$

where $l = \lceil {}^2\log n \rceil$. For particular values of n, this upper bound is proven to be strict. For screens of size 1024×1024 pixels, (a) implies a maximal deviation of approximately $^{10}/_3 + ^1/_9$. In [Corthout & Jonkers 1986b] a refinement of the definition of Bezier shapes in discrete space is presented, in which the grid is upscaled on each recursion step. Thus, the accumulation of rounding errors is avoided, and the pixel sets that are generated are close digitisations. Furthermore, the authors claim that for line segments, this upscale is not at the expense of extra bits. This seems to be a promising development.

The deviation functions of the Adapted CJ and the Symmetric function also satisfy (a), where for the Symmetric function

$$l = \left\lfloor {}^2\log n \right\rfloor .$$

Thus, for the Symmetric function, the deviation upper bound given by (a) is smaller then for the Adapted CJ function. In a way similar as in Section 3.2.2 it may be proven that for the Adapted CJ function, the <u>vertical</u> distance of any pixel $r \in f[p,q]$ to $[p,q]$ (which is at most $\tilde{d}(r,[p,q])$), does not exceed

$$\frac{n}{3} + \frac{(1-(-\frac{1}{2})^n)}{9},$$

where $n = \left\lceil {}^2\log d(p,q) \right\rceil$.

We have not been able to find strict upper bounds for the deviation functions of the Adapted CJ and Symmetric function.

For comparison, the value of $f[\underline{0},(21,-11)]$ is shown in Figure 3.14, for all three functions. Remember that this line segment is an example where the upper bound of the Adapted Corthout-Jonkers function is strict.

To give an expression of the appearance of the digitised lines on a high resolution device, we have also included some figures of digitised line segments generated on a Sun workstation (resolution 1152×900 pixels) and subsequently printed on a laser printer. Figure 3.15 concerns a line segment for which the deviation of the Corthout-Jonkers function equals $b_9 \approx 3.11$. In the second example (Figure 3.16), the differences between the Bresenham digitisation on the one hand, and the recursively defined digitisations on the other hand, are smaller than in the previous one.

From these figures we may conclude that for the qualification of line functions an other measure might be needed, one that expresses the smoothness of digitised line segments.

Finally we would like to mention that in [de Roo et al 1980] an interesting algorithm is presented that generates values of the Symmetric function. This algorithm is not the straight forward translation of the definition, but is based on "rolling out" previously generated line parts. They even present a schematic hardware representation of this algorithm.

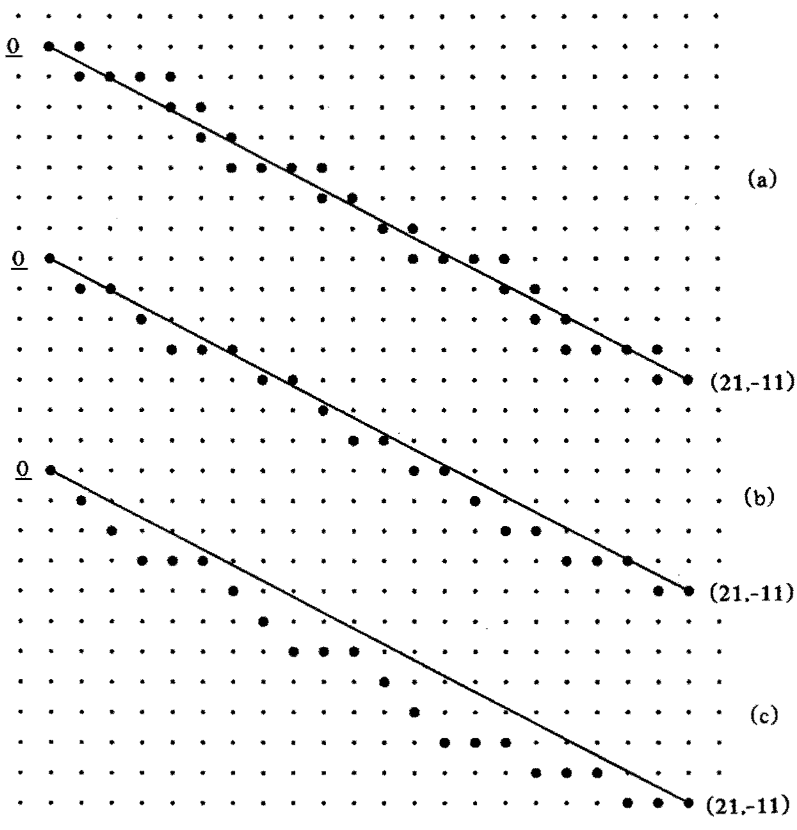


Figure 3. 14

f[0, (21, -11)], *where f is*

a) the Symmetric function.
b) the Adapted CJ function
c) the Corthout-Jonkers function

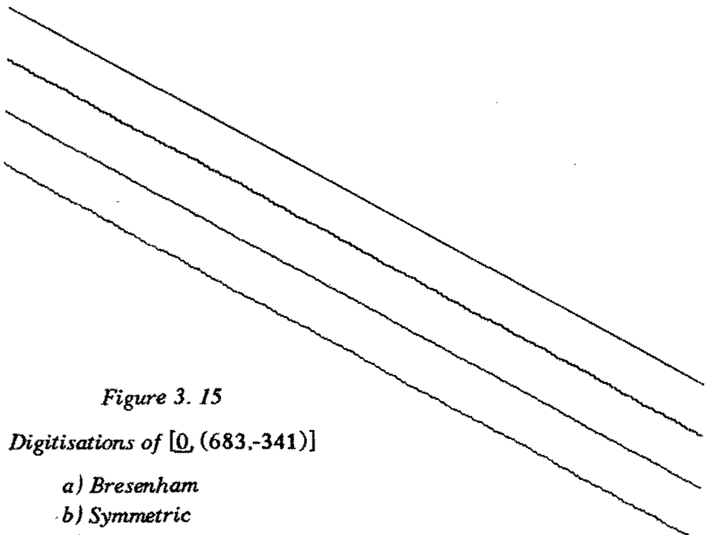


Figure 3. 15

Digitisations of $[\underline{0}, (683,-341)]$

a) Bresenham
b) Symmetric
c) Adapted CJ
d) Corthout-Jonkers

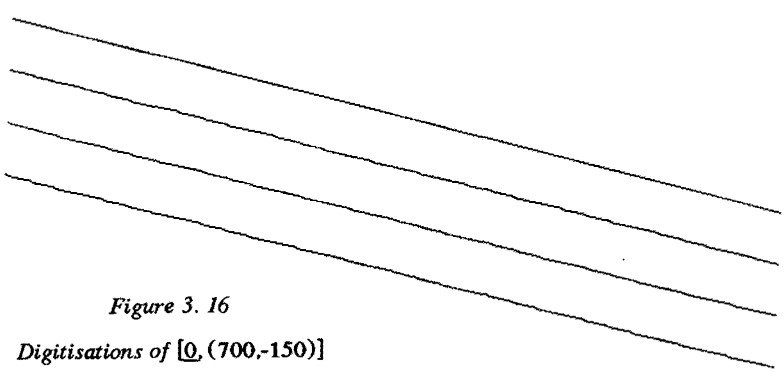


Figure 3. 16

Digitisations of $[\underline{0}, (700,-150)]$

a) Bresenham
b) Symmetric
c) Adapted CJ
d) Corthout-Jonkers

4

Construction of convex line functions

4.0 Introduction

This chapter deals with the construction of a class of convex line functions. Recall that a line function f is convex if and only if for all pixels $r, s \in f[p,q]$ holds that $f[r,s]$ is a subset of $f[p,q]$.

In Section 2 we have proved that no translation invariant, minimal, convex line functions exist that are close. Moreover, we have demonstrated that the deviation function of any translation invariant, minimal, convex line function has a lower bound which is linear in $d(p,q)$. Hence, to find convex line functions with smaller deviation functions, one has to give up translation invariance or minimality. In this section we shall present a class of minimal, convex line functions on domain $D = [0..N]^2 \times [0..N]^2$. These functions are not close; for each function f from this class, the deviation value $E_f(N)$ has a lower bound that is logarithmic in N.

Although the restricted domain of these functions might seem a serious limitation, we would like to remark that convexity is a property that is important for applications where computations on graphical objects are performed in image space. This space is finite, since it depends on the resolution of the graphical device that is used.

The following strategy will be used, where we restrict ourselves to line segments $[p,q]$ for which $0 \leqslant \hat{p}.x \leqslant \hat{q}.x \leqslant N$, where N is a fixed natural number, and $\hat{q} - \hat{p} \in O_0$.
Suppose that g is a translation invariant, minimal line function, which itself is not convex, and suppose that g satisfies the following two conditions.

For all $p, q \in \mathbf{Z}^2$, where $0 \leqslant p.x \leqslant q.x \leqslant N$ and $q - p \in O_0$,

0) a pair of pixels t_0, t_1 exists with x-coordinate 0 and N respectively, such that p and q are contained in $g[t_0, t_1]$, that is, t_0, t_1 must satisfy

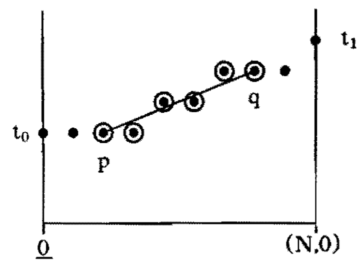
$$t_0.x = 0 \ \wedge \ t_1.x = N \ \wedge \ t_1 - t_0 \in O_0 \ \wedge \ p, q \in g[t_0, t_1]. \tag{a}$$

1) if both t_0, t_1 and t_2, t_3 satisfy (a), then the subsets of $g[t_0, t_1]$ and $g[t_2, t_3]$ associated with the path between p and q, are exactly the same.

Then we may define the function f by

$$f[p,q] := \{r \in g[t_0,t_1] \mid r.x \in [\hat{p}.x..\hat{q}.x]\},$$

where t_0,t_1 satisfy (a). In other words, $f[p,q]$ is the subset of $g[t_0,t_1]$ corresponding to the path between p and q in $g[t_0,t_1]$. This is illustrated in the figure below, where the elements of $g[t_0,t_1]$ are indicated by • and the elements of $f[p,q]$ by ○.



Note that, because of condition 0, for any p and q, a pair t_0,t_1 exists that satisfies (a), and because of condition 1 it does not matter which t_0,t_1 is chosen, hence the above definition of f is consistent.

From the above definition it follows immediately that f is convex since $r,s \in f[p,q]$ implies that

$$r,s \in \{t \in g[t_0,t_1] \mid t.x \in [\hat{p}.x..\hat{q}.x]\},$$

where t_0,t_1 satisfy (a). Then t_0,t_1 also satisfy

$$t_0.x = 0 \wedge t_1.x = N \wedge t_1 - t_0 \in O_0 \wedge r,s \in g[t_0,t_1],$$

which implies that

$$f[r,s] \subseteq f[p,q].$$

Hence, any translation invariant, minimal line function g (which itself need not be convex) that satisfies conditions 0 and 1, induces a line function f that does have the convexity property. In this chapter we shall investigate what line functions g satisfy conditions 0 and 1.

In Section 1, conditions 0 and 1 are rephrased in terms of chain codes; these rephrased conditions are called the *all height condition* and the *equal height condition*. Next it is proven that these two conditions are equivalent to two other, simpler ones, called the *once-a-one condition* and the *non-decreasing condition*. In Section 2 it is shown that the Adapted CJ function satisfies the once-a-one condition and the non-decreasing condition. Hence, the Adapted CJ function induces a convex line function f_{ACJ} on domain D. In Section 3 f_{ACJ} is generalised to line segments $[p,q]$ where $\hat{q}-\hat{p}$ is contained in other octants than O_0. Furthermore, in Section 3 the deviation function of f_{ACJ} will be investigated.

In Section 4, it is shown that the class of translation invariant, minimal line functions that satisfy the once-a-one condition and the non-decreasing condition, is characterised by the class of permutations of the numbers $1,\ldots,N$. Hence, each permutation of the numbers $1,\ldots,N$ induces a convex line function on domain D, and the number of convex line functions that may be constructed this way is $N!$.

Section 5 contains the concluding remarks.

4.1 Equivalent conditions

Let g be a translation invariant, minimal line function. From Chapter 2 we know that with each pixel set $g[p,q]$ a chain code $c \circ g[p,q]$ is associated. Here, we are interested in the values and chain codes of $g[t_0,t_1]$, where

$$t_0.x = 0 \wedge t_1.x = N \wedge t_1 - t_0 \in O_0. \qquad \text{(b)}$$

Since g is translation invariant, we consider the case $t_0.y = 0$ only. Furthermore, since $t_1 - t_0 \in O_0$, we only need to investigate the values and chain codes of $g[\underline{0},(N,n)]$, where $n \in [0..N]$. Therefore we define for any minimal, translation invariant line function g, the sequence of chain codes $(\sigma_n(g))_{n \in [0..N]}$ by

$$\sigma_n(g) := c \circ g[\underline{0},(N,n)].$$

In words, $\sigma_n(g)$ is the chain code of the path corresponding to the g digitisation of the line segment connecting $\underline{0}$ and (N,n). In Figure 4.0, the function values $g[\underline{0},(N,n)]$ and their associated chain codes are shown for the Bresenham function (Section 2.4.5) and $N=5$.

From Property 2.11 it follows that if g is a minimal line function, then for any t_0,t_1 satisfying (b), the chain code of $g[t_0,t_1]$ is an element of $\{7,0,1\}^*$. From Lemma 2.12 it follows that $\sigma \in \{7,0,1\}^*$ of length l is the chain code of a minimal path from p to q if and only if

$$q.x - p.x = l \wedge q.y - p.y = \mathrm{N}_\sigma(1) - \mathrm{N}_\sigma(7).$$

We therefore introduce for such chain codes the *height function* $h : \{7,0,1\}^* \to \mathbf{Z}$, defined by

$$h(\sigma) := \mathrm{N}_\sigma(1) - \mathrm{N}_\sigma(7).$$

The height of a string in $\{7,0,1\}^*$ corresponds to the difference of y-coordinates of the end-points of any minimal path which has this string as chain code. For instance, $h(00100) = 1$ and $h(07110) = 1$. From the definition of $\sigma_n(g)$ the following property may be derived.

Property 4.0:

If g is a minimal line function, then for all $n \in [0..N]$,

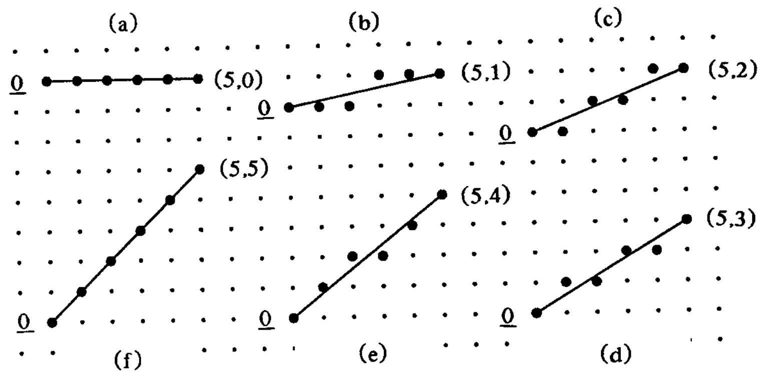


Figure 4. 0

The digitisations $g[\underline{0}, (5, n)]$, where g is the Bresenham function.

$a)\ \sigma_0(g) = 00000$ $d)\ \sigma_3(g) = 10101$
$b)\ \sigma_1(g) = 00100$ $e)\ \sigma_4(g) = 11011$
$c)\ \sigma_2(g) = 01010$ $f)\ \sigma_5(g) = 11111$

$$h(\sigma_n(g)) = n.$$

□

The height function satisfies the following property, expressing that if a minimal path is partitioned into 2 subpaths, then the sum of the height of the chain codes associated with the subpaths equals the height of the chain code associated with the original path. The property is presented without proof.

Property 4.1:

For all $l \in \mathbf{N}$ and all chain codes $\sigma \in \{7,0,1\}^*$ of length l,

$$(\underline{A}\, i : i \in [1..l] : h(\sigma) = h(\sigma[1:i]) + h(\sigma[i+1:l])).$$

□

For any translation invariant, minimal line function g it is possible to express in terms of chain codes a necessary and sufficient condition for a pixel r to be contained in $g[\underline{0}, (N, n)]$. This condition, presented in the following property, will be used to translate the conditions of Section 4.0 into the all height condition and equal height condition.

Property 4.2:

For any translation invariant, minimal line function g, $n \in [0..N]$, and $r \in \mathbf{Z}^2$, where

$0 \leq r.x \leq N$,

$$r \in g[\underline{0}, (N, n)] \Leftrightarrow h(\sigma_n(g)[1{:}r.x]) = r.y.$$

Proof:

Let g be a translation invariant, minimal line function, and $n \in [0..N]$.
Let $r_0, r_1, \ldots, r_N$ be the minimal path from $\underline{0}$ to (N, n) associated with $g[\underline{0}, (N, n)]$. From Property 2.11 it follows that the chain code of this path, $\sigma_n(g)$, is an element of $\{7,0,1\}^*$.

- Suppose $r \in g[\underline{0}, (N, n)]$.
  Property 2.10 implies that $r = r_i$, where $i = r.x$. Property 2.6 implies that $r_0, r_1, \ldots, r_i$ is a minimal path from $\underline{0}$ to r. The chain code of $r_0, r_1, \ldots, r_i$ is $\sigma_n(g)[1{:}i]$. Lemma 2.12 then implies that

  $$h(\sigma_n(g)[1{:}i]) = r.y.$$

- Suppose pixel r, where $0 \leq r.x \leq N$, satisfies

  $$h(\sigma_n(g)[1{:}i]) = r.y.$$

  Then, according to Lemma 2.12, $\sigma_n(g)[1{:}i]$ is the chain code of a minimal path from $\underline{0}$ to r, hence, applying Property 2.9,

  $$r = \sum_{j=1}^{r.x} v(\sigma_n(g)[j]).$$

  From Property 2.9 it follows that r is an element of the path $r_0, r_1, \ldots, r_N$, and hence $r \in g[\underline{0}, (N, n)]$.

□

Using the above property, we shall now rephrase the conditions of Section 4.1. Consider Condition 0, which expresses that for all $p, q \in \mathbf{Z}^2$ such that $0 \leq p.x \leq q.x \leq N$ and $q - p \in O_0$, a pair of pixels t_0, t_1 exists such that

$$t_0.x = 0 \wedge t_1.x = N \wedge t_1 - t_0 \in O_0 \wedge p, q \in g[t_0, t_1]. \qquad \text{(a)}$$

This condition may be rewritten in the following way, where p, q, t_0, t_1 are considered to be elements of $\mathbf{Z}^2$, and i, j, k, l, m of $\mathbf{Z}$.

$(\underline{A}p, q : 0 \leq p.x \leq q.x \leq N \wedge q - p \in O_0 :$
$\quad (\underline{E}t_0, t_1 : t_0.x = 0 \wedge t_1.x = N \wedge t_1 - t_0 \in O_0 : p, q \in g[t_0, t_1]))$

$\equiv$ { renaming $p.x$ as i, $q.x$ as j, $q.y - p.y$ as k, $p.y$ as l, $t_0.y$ as m, and $t_1.y$ as n }

$(\underline{A}i, j, k, l : 0 \leq i \leq j \leq N \wedge 0 \leq k \leq j - i :$
$\quad (\underline{E}m, n : 0 \leq n - m \leq N : (i, l), (j, k + l) \in g[(0, m), (N, n)]))$

$\equiv$ { g is translation invariant }

$(\underline{A} i, j, k, l : 0 \leqslant i \leqslant j \leqslant N \wedge 0 \leqslant k \leqslant j-i :$
$\quad (\underline{E} m, n : 0 \leqslant n-m \leqslant N : (i, l-m), (j, k+l-m) \in g[\underline{0}, (N, n-m)]))$

$\equiv$ { Property 4.2 }

$(\underline{A} i, j, k, l : 0 \leqslant i \leqslant j \leqslant N \wedge 0 \leqslant k \leqslant j-i :$
$\quad (\underline{E} m, n : 0 \leqslant n-m \leqslant N : h(\sigma_{n-m}(g)[1{:}i]) = l-m$
$\qquad \wedge\ h(\sigma_{n-m}(g)[1{:}j]) = k+l-m\ ))$

$\equiv$ { Property 4.1 }

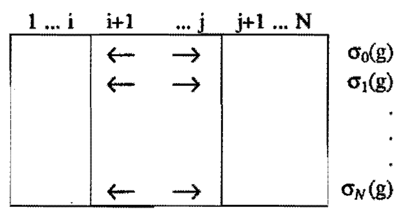
$(\underline{A} i, j, k, l : 0 \leqslant i \leqslant j \leqslant N \wedge 0 \leqslant k \leqslant j-i :$
$\quad (\underline{E} m, n : 0 \leqslant n-m \leqslant N : h(\sigma_{n-m}(g)[1{:}i]) = l-m$
$\qquad \wedge\ h(\sigma_{n-m}(g)[i+1{:}j]) = k\ ))$

$\equiv$ { renaming $n-m$ as n }

$(\underline{A} i, j, k, l : 0 \leqslant i \leqslant j \leqslant N \wedge 0 \leqslant k \leqslant j-i :$
$\quad (\underline{E} m, n : n \in [0..N] : h(\sigma_n(g)[1{:}i]) = l-m\ \wedge\ h(\sigma_n(g)[i+1{:}j]) = k\ ))$

$\equiv$ { proposition calculus }

0') $(\underline{A} i, j, k : 0 \leqslant i \leqslant j \leqslant N \wedge 0 \leqslant k \leqslant j-i :$
$\quad (\underline{E} n : n \in [0..N] : h(\sigma_n(g)[i+1{:}j]) = k\ ))$



Condition 0') will be referred to as the *all height condition*. Its meaning may be illustrated as follows. If the sequence $(\sigma_n(g))_{n \in [0..N]}$ is considered as an array (see the figure alongside), then the all height condition expresses that the chain codes in each column-range $[i+1..j]$ should cover the height values $0, 1, \ldots, j-i$.

Now we shall rephrase Condition 1 of the previous section. This condition expresses that, if for given p and q, the pairs t_0, t_1 and t_2, t_3 both satisfy (a), then the parts of $g[t_0, t_1]$ and $g[t_2, t_3]$ between p and q are the same. This is equivalent to

1') for all $i, j, k \in \mathbf{N}$, where $0 \leqslant i \leqslant j \leqslant N$ and $0 \leqslant k \leqslant i-j$, and all $n, m \in [0..N]$,

$$h(\sigma_n(g)[i+1{:}j]) = k \wedge h(\sigma_m(g)[i+1{:}j]) = k \ \Rightarrow\ \sigma_n(g)[i+1{:}j] = \sigma_m(g)[i+1{:}j].$$

This condition will be referred to as the *equal height condition*.

We shall now formulate two other conditions, and subsequently prove that these two new conditions are equivalent to conditions 0' and 1'. Recall that the chain code associated with $g[\underline{0}, (N, n)]$ is an element of $\{7,0,1\}^*$.

The first condition, called the *non-decreasing condition*, expresses that the y-values of the subsequent pixels in the path from $\underline{0}$ to (N, n) associated with $g[\underline{0}, (N, n)]$, do not decrease. In terms of chain codes this is formulated as follows.

$$(\underline{A}n : n \in [0..N] : \sigma_n(g) \in \{0,1\}^*).$$

The second condition expresses that if a 1 occurs at index i in chain code $\sigma_n(g)$, than in all chain codes $\sigma_m(g)$, where $m > n$, also a 1 occurs at index i:

$$(\underline{A}i, n : i \in [1..N] \wedge n \in [0..N-1] :$$
$$\sigma_n(g)[i] = 1 \Rightarrow \sigma_{n+1}(g)[i] = 1).$$

This condition will be referred to as the *once-a-one condition*.

In the remaining part of this section the following will be proven. Given a sequence of chain codes $(\sigma_n)_{n \in [0..N]}$, each of length N, where for all $n \in [0..N]$,

$$\sigma_n \in \{7,0,1\}^* \;\wedge\; h(\sigma_n) = n,$$

then the conjunction of the all height condition and the equal height condition is equivalent to the conjunction of the non-decreasing condition and the once-a-one condition. This is shown by subsequently proving:

- (non-decreasing condition $\wedge$ once-a-one condition)
  $\Rightarrow$ all height condition (Lemma 4.3)
- (non-decreasing condition $\wedge$ once-a-one condition)
  $\Rightarrow$ equal height condition (Lemma 4.4)
- (all height condition $\wedge$ equal height condition)
  $\Rightarrow$ non-decreasing condition (Lemma 4.7)
- (all height condition $\wedge$ equal height condition)
  $\Rightarrow$ once-a-one condition (Lemma 4.8)

Lemma 4.3:

(non-decreasing condition $\wedge$ once-a-one condition) $\Rightarrow$ all height condition.

Proof:

Assume that the non-decreasing condition and once-a-one condition hold.
Let $i, j, k \in [0..N]$, $i \leqslant j$, $k \leqslant j-i$. We have to prove that for some $n \in [0..N]$,

$$h(\sigma_n[i+1:j]) = k.$$

Define the function $H : [0..N] \to \mathbf{N}$ by

$$H(n) := h(\sigma_n[i+1:j]).$$

Since $h(\sigma_n) = n$, the non-decreasing condition implies that $h(\sigma_n)$ equals the number of ones occurring in σ_n. Hence, each element of σ_0 equals 0, and, since the length of σ_N equals N, each element of σ_N equals 1. Consequently,

$$H(0) = 0 \wedge H(N) = j - i. \tag{a}$$

Furthermore, because of the once-a-one condition and the non-decreasing condition,

$$(\underline{A}n : n \in [0..N-1] : H(n) \leqslant H(n+1)), \tag{b}$$

which means that H is an increasing function. Because $h(\sigma_n) = n$, σ_{n+1} contains one 1 more than σ_n : in combination with the once-a-one condition this implies that

$$(\underline{A}n : n \in [0..N-1] : H(n+1) \leqslant H(n)+1), \tag{c}$$

which means that H increases with steps of 1. Hence, combining (a), (b), (c), it follows that the range of H equals $[0..i-j]$. Since $k \leqslant i-j$, it follows that $n \in [0..N]$ exists such that $H(n) = k$, which is equivalent to

$$h(\sigma_n[i+1:j]) = k.$$

The all height condition has thus been proven.

□

Lemma 4.4:

(non-decreasing condition $\wedge$ once-a-one condition) $\Rightarrow$ equal height condition.

Proof:

Assume that the non-decreasing condition and once-a-one condition hold.
Let $i, j, k, n, m \in [0..N]$, $i \leqslant j$, $k \leqslant j-i$, such that

$$h(\sigma_n[i+1:j]) = k \ \wedge \ h(\sigma_m[i+1:j]) = k.$$

We have to prove that $\sigma_n[i+1:j] = \sigma_m[i+1:j]$.
Suppose, without loss of generality, that $n < m$.
Because of the non-decreasing condition, $h(\sigma_n[i+1:j]) = k$ implies that $\sigma_n[i+1:j]$ contains k ones. Then, due to the once-a-one condition, $\sigma_m[i+1:j]$ also contains ones at these indices. Since $h(\sigma_m[i+1:j]) = k$, and $\sigma_m \in \{0,1\}^*$, these are the only ones that $\sigma_m[i+1:j]$ contains, hence

$$\sigma_n[i+1:j] = \sigma_m[i+1:j].$$

□

We have now proven that the all height condition and equal height condition are necessary conditions for the non-decreasing condition in combination with the once-a-one condition. The proof that they are also sufficient conditions is based on induction on N. In order to apply the induction assumptions in a correct way, we need the following two lemmas.

Lemma 4.5:

Let $N > 1$. If for $(\sigma_n)_{n \in [0..N]}$, where each σ_n has length N and $\sigma_n \in \{7,0,1\}^*$ and

$h(\sigma_n) = n$, the all height condition and equal height condition hold, then the sequence $(\sigma_n)_{n \in [0..N]}$ may be rearranged in a sequence $(\sigma'_n)_{n \in [0..N]}$ such that

(a) $(\underline{A}n : n \in [0..N-1] : h(\sigma'_n[1{:}N-1]) = n)$

(b) $(\underline{E}i : i \in [0..N-1] : \sigma'_i[1{:}N-1] = \sigma'_N[1{:}N-1])$

(c) the all height condition and equal height condition hold for $(\sigma'_n[1{:}N-1])_{n \in [0..N-1]}$

Proof:

Let $N > 1$.
Let $(\sigma_n)_{n \in [0..N]}$ be a sequence of $N+1$ chain codes, each of length N, such that for all $n \in [0..N]$,

$$\sigma_n \in \{7,0,1\}^* \ \wedge \ h(\sigma_n) = n.$$

Then

$$\sum_{n=0}^{N} h(\sigma_n) = \sum_{n=0}^{N} n = ½N(N+1). \tag{d}$$

Suppose that $(\sigma_n)_{n \in [0..N]}$ satisfies the all height and equal height conditions.

Consider the $N+1$ substrings $\sigma_n[1{:}N-1]$ of length $N-1$. From the all height condition it follows that for all $k \in [0..N-1]$, an $n \in [0..N]$ exists such that $h(\sigma_n[1{:}N-1]) = k$. Hence, the sequence $(\sigma_n)_{n \in [0..N]}$ may be rearranged into a sequence $(\sigma'_n)_{n \in [0..N]}$ such that for all $n \in [0..N-1]$,

$$h(\sigma'_n[1{:}N-1]) = n.$$

(This is illustrated in Figure 4.1)
Hence, a) is satisfied.

Furthermore,

$$\sum_{n=0}^{N-1} h(\sigma'_n[1{:}N-1]) = \sum_{n=0}^{N-1} n = ½(N-1)N. \tag{e}$$

Let γ be defined by $\gamma := \sigma'_N[1{:}N-1]$. Since $\sigma'_n[N] \in \{7,1,0\}$ and $h(\sigma'_N) \in [0..N]$, it follows that $h(\sigma'_N[1{:}N-1]) \in [-1..N-1]$, hence $h(\gamma) \in [-1..N-1]$.
We shall now prove that $h(\gamma) \in [0..N-1]$.

Suppose

$$h(\gamma) = -1. \tag{f}$$

We then may derive the following.

$½N(N+1)$

$=$ { equation (d) }

$$\sum_{n=0}^{N} h(\sigma_n)$$

$=$ { $(\sigma'_n)_{n \in [0..N]}$ is a rearrangement of $(\sigma_n)_{n \in [0..N]}$ }

$$\sum_{n=0}^{N} h(\sigma'_n)$$

$=$ { Property 4.1 }

$$\sum_{n=0}^{N} h(\sigma'_n[1:N-1]) + \sum_{n=0}^{N} h(\sigma'_n[N])$$

$=$ { definition of γ }

$$h(\gamma) + \sum_{n=0}^{N-1} h(\sigma'_n[1:N-1]) + \sum_{n=0}^{N} h(\sigma'_n[N])$$

$=$ { equations (e) and (f) }

$$-1 + \tfrac{1}{2}(N-1)N + \sum_{n=0}^{N} h(\sigma'_n[N]).$$

From this it follows that

$$\sum_{n=0}^{N} h(\sigma'_n[N]) = N+1.$$

This implies that for all $n \in [0..N]$, $\sigma'_n[N]=1$, and thus $\sigma_n[N]=1$. This contradicts with the all height condition, which prescribes that $n \in [0..N]$ exists such that $\sigma_n[N]=0$.

Hence, we may conclude that $h(\gamma) \in [0..N-1]$.
Consequently, according to (a), $i \in [0..N-1]$ exists such that $h(\sigma'_i[1:N-1]) = h(\gamma)$.
Applying the equal height condition then leads to

$$\sigma'_i[1:N-1] = \gamma. \qquad \text{(g)}$$

Thus, b) is satisfied.

Furthermore, because of b), the all height and equal height conditions also hold for the sequence of chain codes $(\sigma'_n[1:N-1])_{n \in [0..N-1]}$, since the exclusion of γ does not change anything.

This completes the proof.

□

In quite a similar way the following lemma may be proven.

Lemma 4.6:

Let $N>1$. If for $(\sigma_n)_{n \in [0..N]}$, where each σ_n has length N and $\sigma_n \in \{7,0,1\}^*$ and $h(\sigma_n) = n$, the all height condition and equal height condition hold, then the sequence $(\sigma_n)_{n \in [0..N]}$ may be rearranged in a sequence $(\sigma'_n)_{n \in [0..N]}$ such that

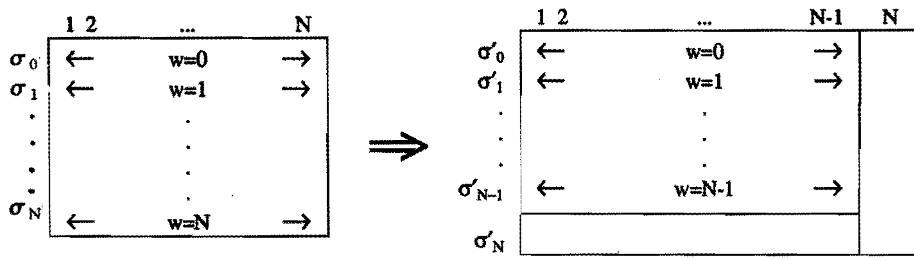


Figure 4. 1

Illustration of rearrangement of $(\sigma_n)_{n \in [0..N]}$ *into* $(\sigma'_n)_{n \in [0..N]}$.

(a) $(\underline{A}n : n \in [0..N-1] : h(\sigma'_n[2{:}N] = n)$

(b) $(\underline{E}i : i \in [0..N-1] : \sigma'_i[2{:}N] = \sigma'_N[2{:}N])$

(c) the all height condition and equal height condition hold for $(\sigma'_n[2{:}N])_{n \in [0..N-1]}$

□

We shall now prove that the all height condition and the equal height condition together imply the non-decreasing condition and the once-a-one condition.

Lemma 4.7:

(all height condition $\wedge$ equal height condition) $\Rightarrow$ non-decreasing condition .

Proof:

We shall use induction on N.

- Let $N=0$.
  Then $\sigma_0 = \epsilon$, which implies that $\sigma_0 \in \{0,1\}^*$, and hence the non-decreasing condition holds for $(\sigma_n)_{n \in [0..0]}$.
- Let $N=1$.
  Then, since σ_0 and σ_1 have length 1, and $h(\sigma_0) = 0$, $h(\sigma_1) = 1$,

  $\sigma_0 = 0$, and $\sigma_1 = 1$.

  Hence, $\sigma_n \in \{0,1\}^*$, and thus the non-decreasing condition holds for $(\sigma_n)_{n \in [0..1]}$.
- Let $N>1$, and assume that for all $M<N$ it has been proven that for $(\sigma_n)_{n \in [0..M]}$, where each σ_n has length M and $\sigma_n \in \{7,0,1\}^*$ and $h(\sigma_n) = n$, the all height condition and the equal height condition together imply the non-decreasing condition.

  Let $(\sigma_n)_{n \in [0..N]}$ be a sequence of $N+1$ chain codes, each of length N, such that for all $n \in [0..N]$,

$\sigma_n \in \{7,0,1\}^* \;\wedge\; h(\sigma_n) = n.$

Suppose the all height condition and equal height condition hold for $(\sigma_n)_{n \in [0..N]}$.

From Lemma 4.5 it follows that the sequence $(\sigma_n)_{n \in [0..N]}$ may be rearranged in a sequence $(\sigma'_n)_{n \in [0..N]}$ such that the all height condition and equal height condition hold for $(\sigma'_n[1:N-1])_{n \in [0..N-1]}$, and

$$(\underline{E}i : i \in [0..N-1] : \sigma'_i[1:N-1] = \sigma'_N[1:N-1]). \tag{a}$$

We then may apply the induction assumption on the sequence $(\sigma'_n[1:N-1])_{n \in [0..N-1]}$, from which it follows that for all $n \in [0..N-1]$,

$$\sigma'_n[1:N-1] \in \{0,1\}^*.$$

In combination with (a) this leads to

$$(\underline{A}n : n \in [0..N] : \sigma'_n[1:N-1] \in \{0,1\}^*),$$

and thus, since $(\sigma'_n)_{n \in [0..N]}$ is a rearrangement of $(\sigma_n)_{n \in [0..N]}$,

$$(\underline{A}n : n \in [0..N] : \sigma_n[1:N-1] \in \{0,1\}^*). \tag{b}$$

In a quite similar way, using Lemma 4.6, we may prove that

$$(\underline{A}n : n \in [0..N] : \sigma_n[2:N] \in \{0,1\}^*). \tag{c}$$

Combining (b) and (c) results in

$$(\underline{A}n : n \in [0..N] : \sigma_n \in \{0,1\}^*),$$

and thus the non-decreasing condition holds for $(\sigma_n)_{n \in [0..N]}$.

□

Lemma 4.8:

(all height condition $\wedge$ equal height condition) $\Longrightarrow$ once-a-one condition .

Proof:

We shall use induction on N.

- Let $N = 0$.
  Then $\sigma_0 = \epsilon$, which implies that the once-a-one condition holds for $(\sigma_n)_{n \in [0..0]}$.
- Let $N = 1$.
  Then, since σ_0 and σ_1 have length 1, and $h(\sigma_0) = 0$, $h(\sigma_1) = 1$,

  $$\sigma_0 = 0, \text{ and } \sigma_1 = 1.$$

  Hence, the once-a-one condition holds for $(\sigma_n)_{n \in [0..1]}$.
- Let $N > 1$, and assume that for all $M < N$ it has been proven that for $(\sigma_n)_{n \in [0..M]}$, where each σ_n has length M and $\sigma_n \in \{7,0,1\}^*$ and $h(\sigma_n) = n$, the all height

condition and the equal height condition together imply the once-a-one condition.

Let $(\sigma_n)_{n \in [0..N]}$ be a sequence of $N+1$ chain codes, each of length N, such that for all $n \in [0..N]$,

$$\sigma_n \in \{7,0,1\}^* \ \wedge \ h(\sigma_n) = n.$$

Suppose the all height condition and equal height condition hold for $(\sigma_n)_{n \in [0..N]}$.

From Lemma 4.5 it follows that the sequence $(\sigma_n)_{n \in [0..N]}$ may be rearranged in a sequence $(\sigma'_n)_{n \in [0..N]}$ such that $h(\sigma'_n[1{:}N-1]) = n$ for all $n \in [0..N-1]$, and the all height condition and equal height condition hold for $(\sigma'_n[1{:}N-1])_{n \in [0..N-1]}$, and an $i \in [0..N-1]$ exists such that

$$\sigma'_i[1{:}N-1] = \sigma'_N[1{:}N-1]. \tag{a}$$

We then may apply the induction assumption on the sequence $(\sigma'_n[1{:}N-1])_{n \in [0..N-1]}$, from which it follows that the once-a-one condition holds for $(\sigma'_n[1{:}N-1])_{n \in [0..N-1]}$.
From (a) it follows that σ'_i and σ'_N differ only in their last element. Since $(\sigma'_n)_{n \in [0..N]}$ is a rearrangement of $(\sigma_n)_{n \in [0..N]}$, Lemma 4.7 implies that $\sigma'_n \in \{0,1\}^*$, for all $n \in [0..N]$, and therefore either

$$\sigma'_i[N] = 0 \ \wedge \ \sigma'_N[N] = 1 \tag{b}$$

or

$$\sigma'_i[N] = 1 \ \wedge \ \sigma'_N[N] = 0. \tag{c}$$

Assume without loss of generality that (b) holds.
Now we rearrange the sequence $(\sigma'_n)_{n \in [0..N]}$ into $(\gamma_n)_{n \in [0..N]}$ by inserting σ'_N between σ'_i and σ'_{i+1}. (See Figure 4.2 for an illustration hereof.) Since $\sigma'_i[1{:}N-1] = \sigma'_N[1{:}N-1]$, and the once-a-one condition holds for the sequence $(\sigma'_n[1{:}N-1])_{n \in [0..N-1]}$, it then follows that the once-a-one condition holds for $(\gamma_n[0..N-1])_{n \in [0..N]}$, that is,

$$(\underline{\mathrm{A}}\, i, n : i \in [1..N-1] \wedge n \in [0..N-1] : \gamma_n[i] = 1 \Rightarrow \gamma_{n+1}[i] = 1). \tag{d}$$

We shall show that the once-a-one condition also holds for $(\gamma_n[N])_{n \in [0..N]}$, that is,

$$(\underline{\mathrm{A}}\, n : n \in [0..N-1] : \gamma_n[N] = 1 \Rightarrow \gamma_{n+1}[N] = 1). \tag{e}$$

Since $(\gamma_n)_{n \in [0..N]}$ is a rearrangement of $(\sigma_n)_{n \in [0..N]}$, two different elements of $(\gamma_n)_{n \in [0..N]}$ must have different heights. Then it can be seen that $\gamma_{i-1}[N] \neq 1$, because in that case $h(\gamma_{i-1})$ would equal $h(\gamma_i)$. Hence, since $\gamma_n \in \{0,1\}^*$, it follows that $\gamma_{i-1}[N] = 0$. By induction on j it can then be proven that

$$(\underline{\mathrm{A}}\, j : j \in [0..i] : \gamma_i[N] = 0). \tag{f}$$

Similarly, it may be proven that

$(\underline{A}\, j : j \in [i+1..N] : \gamma_i[N] = 1).$ (g)

From (f) and (g) it follows that (e) is satisfied, and furthermore, that

$(\underline{A}\, n : n \in [0..N] : h(\gamma_n) = n\ ).$ (h)

From (d) and (e) it follows that the once-a-one condition holds for the sequence $(\gamma_n)_{n\,\in[0..N]}$. From (h) it follows that $(\gamma_n)_{n\,\in[0..N]}$ is the same sequence as $(\sigma_n)_{n\,\in[0..N]}$, and therefore the once-a-one condition holds for $(\sigma_n)_{n\,\in[0..N]}$.

□

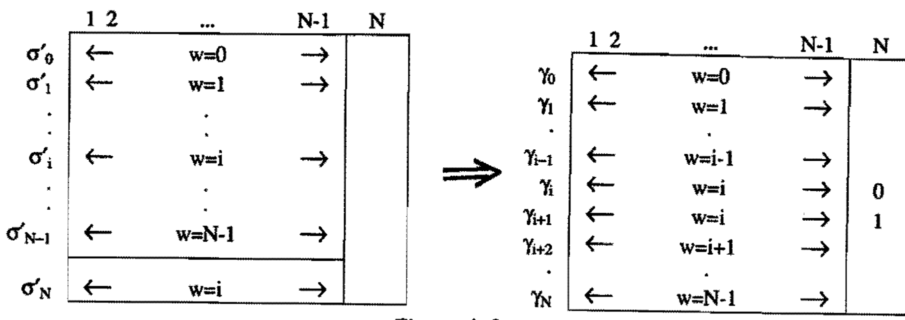


Figure 4. 2

Illustration of the rearrangement of $(\sigma'_n)_{n\,\in[0..N]}$ *into* $(\gamma_n)_{n\,\in[0..N]}$.

Theorem 4.9:

The conjunction of the all height condition and the equal height condition is equivalent to the conjunction of the non-decreasing condition and the once-a-one condition.

Proof:

From Lemmas 4.3, 4.4, 4.7, and 4.8.

□

In Section 4.0 we have shown that any translation invariant, minimal line function g satisfying certain conditions, induces a convex line function on domain D. These conditions were rephrased as the all height condition and the equal height condition. Because of the above theorem, any translation invariant, minimal line function g for which the chain code sequence $(\sigma_n(g))_{n\,\in[0..N]}$ satisfies the non-decreasing condition and the once-a-one condition, induces a convex line function. In the following section we shall investigate which of the translation invariant, minimal line functions presented thus far, generate chain code sequences $(\sigma_n(g))_{n\,\in[0..N]}$ that do satisfy the non-decreasing and once-a-one conditions.

4.2 Functions satisfying the non-decreasing and once-a-one condition

In this section we shall investigate which of the translation invariant, minimal line functions presented thus far, generate chain code sequences $(\sigma_n(g))_{n \in [0..N]}$, for which the non-decreasing condition and once-a-one condition hold.

Of the line functions presented in Section 2.4, only the Adapted Franklin and Bresenham functions are translation invariant and minimal. However, the Adapted Franklin function itself is convex, so the construction of a new function based on the Adapted Franklin function, as proposed in Section 4.0 has no use at all.

If g is the Bresenham function, then, according to its definition in Section 2.4.5,

$$g[\underline{0}, (N, n)] = \{(x, \lceil (n/N)x - 1/2 \rceil) \mid x \in [0..N]\}.$$

Since

$$\lceil (n/N)(x+1) - 1/2 \rceil \geq \lceil (n/N)x - 1/2 \rceil,$$

it follows that $(\sigma_n(g))_{n \in [0..N]}$ does indeed satisfy the non-decreasing condition. However, it does not satisfy the once-a-one condition, as can be seen in Figure 4.0. For example, $\sigma_1(g)[3] = (c \circ g[\underline{0}, (5,1)])[3] = 1$, whereas $\sigma_2(g)[3] = (c \circ g[\underline{0}, (5,2)])[3]) = 0$.

Of the recursive line functions presented in Chapter 3, only the Adapted CJ function is minimal. In Figure 4.3, the sets $g[\underline{0}, (5, n)]$ are shown, where g is the Adapted CJ function, together with their associated chain codes. It can be seen that for this sequence of chain codes the non-decreasing condition and once-a-one condition hold.

In the remaining of this section we shall prove that the Adapted CJ function g indeed satisfies the required conditions. For this purpose we shall introduce the function Y that gives an expression for the y-values of the pixels in $g[\underline{0}, (N, n)]$ as a function of both n and x. The non-decreasing condition and once-a-one condition will then be proven by using various properties of Y.

Since the Adapted CJ function g is minimal, the chain code associated with $g[\underline{0}, (N, n)]$ is an element of $\{7,0,1\}^*$. Furthermore, for each $i \in [0..N]$, $g[\underline{0}, (N, n)]$ contains exactly one pixel p such that $p.x = i$. This observation underlies the following definition.

The function $Y_N : [0..N]^2 \rightarrow \mathbf{N}$ is implicitly defined by the following equation.

$$\{(i, Y_N(n, i)) \mid i \in [0..N]\} = g[\underline{0}, (N, n)].$$

In words, $Y_N(n, i)$ is the y-value of the pixel in $g[\underline{0}, (N, n)]$ whose x-value equals i.

For notational ease, $Y_N(n, i)$ will be denoted in the sequel as $Y(N; n, i)$.

Property 4.10:

For all $N \in \mathbf{N}$, and all $n \in [0..N]$,

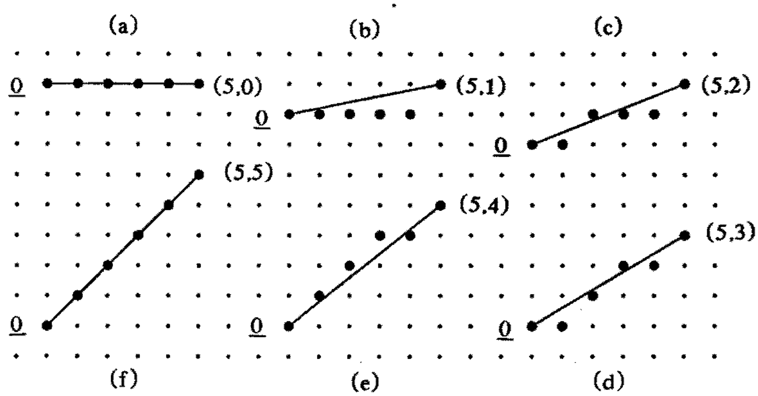


Figure 4. 3

The digitisations $g[\underline{0},(5,n)]$, where g is the Adapted CJ function.

$a)\ \sigma_0(g) = 00000$ $\quad d)\ \sigma_3(g) = 01101$
$b)\ \sigma_1(g) = 00001$ $\quad e)\ \sigma_4(g) = 11101$
$c)\ \sigma_2(g) = 01001$ $\quad f)\ \sigma_5(g) = 11111$

$$Y(N;n,0) = 0 \wedge Y(N;n,N) = n.$$

Proof:

This follows from the lf0 condition of line functions, which implies that $\underline{0} \in g[\underline{0},(N,n)]$ and $(N,n) \in g[\underline{0},(N,n)]$.

□

Let γ_n be the chain code associated with $\{(i,Y(N;n,i)) \mid i \in [0..N]\}$, that is, with $g[\underline{0},(N,n)]$. Since $\gamma_n \in \{7,0,1\}^*$, it can be seen that for all $i \in [1..N]$,

$$\gamma_n[i] = \begin{cases} 0 \text{ iff } Y(N;n,i)-Y(N;n,i-1)=0 \\ 1 \text{ iff } Y(N;n,i)-Y(N;n,i-1)=1 \\ 7 \text{ iff } Y(N;n,i)-Y(N;n,i-1)=\text{-}1. \end{cases}$$

Hence, to prove that $(\gamma_n)_{n\in[0..N]}$ satisfies the non-decreasing condition, it suffices to prove that

$$(\underline{A}i : i \in [1..N] : 0 \leqslant Y(N;n,i)-Y(N;n,i-1) \leqslant 1), \tag{a}$$

and for the once-a-one condition it suffices to prove that

$$(\underline{A}n,i : n \in [0..N-1] \wedge i \in [1..N] : \tag{b}$$

$$Y(N:n,i)-Y(N:n,i-1)=1 \Rightarrow Y(N;n+1,i)-Y(N:n+1,i-1)=1).$$

We shall use the following property to prove that the Adapted CJ function generates chain codes that satisfy both (a) and (b).

Property 4.11:

For all $N \in \mathbf{N}$, $N > 1$, and all $n, i \in [0..N]$,

$$Y(N;n,i) = \begin{cases} Y(N\,\underline{\text{div}}\,2;n\,\underline{\text{div}}\,2,i) & \text{if } i \in [0..N\,\underline{\text{div}}\,2] \\ n\,\underline{\text{div}}\,2 + Y(N-N\,\underline{\text{div}}\,2;n-n\,\underline{\text{div}}\,2,i-N\,\underline{\text{div}}\,2) & \text{if } i \in [N\,\underline{\text{div}}\,2..N]. \end{cases}$$

Proof:

First note that, since $N\,\underline{\text{div}}\,2$ is contained in both ranges, $Y(N\,\underline{\text{div}}\,2;n\,\underline{\text{div}}\,2,N\,\underline{\text{div}}\,2)$ must equal $n\,\underline{\text{div}}\,2+Y(N-N\,\underline{\text{div}}\,2;n-n\,\underline{\text{div}}\,2,0)$. Indeed, from Property 4.10 it can be seen that both expressions equal $n\,\underline{\text{div}}\,2$.

Let $N \in \mathbf{N}$, $N > 1$, and $n \in [0..N]$.

Then

$$\{(i,Y(N;n,i)) \mid i \in [0..N]\}$$

$=$ { definition of Y }

$$g[\underline{0},(N,n)]$$

$=$ { definition of the Adapted CJ function }

$$g[\underline{0},(N\,\underline{\text{div}}\,2,n\,\underline{\text{div}}\,2)] \cup g[(N\,\underline{\text{div}}\,2,n\,\underline{\text{div}}\,2),(N,n)]$$

$=$ { translation invariance of the Adapted CJ function }

$$g[\underline{0},(N\,\underline{\text{div}}\,2,n\,\underline{\text{div}}\,2)] \cup ((N\,\underline{\text{div}}\,2,n\,\underline{\text{div}}\,2) \oplus g[\underline{0},(N-N\,\underline{\text{div}}\,2,n-n\,\underline{\text{div}}\,2)])$$

$=$ { definition of Y }

$$\{(i,Y(N\,\underline{\text{div}}\,2;n\,\underline{\text{div}}\,2,i)) \mid i \in [0..N\,\underline{\text{div}}\,2]\} \cup$$

$$((N\,\underline{\text{div}}\,2,n\,\underline{\text{div}}\,2) \oplus \{(i,Y(N-N\,\underline{\text{div}}\,2;n-n\,\underline{\text{div}}\,2,i)) \mid i \in [0..N-N\,\underline{\text{div}}\,2]\})$$

$=$ { definition $\oplus$ }

$$\{(i,Y(N\,\underline{\text{div}}\,2;n\,\underline{\text{div}}\,2,i)) \mid i \in [0..N\,\underline{\text{div}}\,2]\} \cup$$

$$\{(N\,\underline{\text{div}}\,2+i,n\,\underline{\text{div}}\,2+Y(N-N\,\underline{\text{div}}\,2;n-n\,\underline{\text{div}}\,2,i)) \mid i \in [0..N-N\,\underline{\text{div}}\,2]\}$$

$=$ { renaming dummy variable i in the second set }

$$\{(i,Y(N\,\underline{\text{div}}\,2;n\,\underline{\text{div}}\,2,i)) \mid i \in [0..N\,\underline{\text{div}}\,2]\} \cup$$

$$\{(i,n\,\underline{\text{div}}\,2+Y(N-N\,\underline{\text{div}}\,2;n-n\,\underline{\text{div}}\,2,i-N\,\underline{\text{div}}\,2)) \mid i \in [N\,\underline{\text{div}}\,2..N]\}.$$

Hence, the property has been proven.

□

We shall now prove that the Adapted CJ function satisfies (a) and (b).

Property 4.12:

For all $N \in \mathbf{N}$, $N > 0$, and all $n \in [0..N]$ and $i \in [1..N]$,

$$0 \leqslant Y(N; n, i) - Y(N; n, i-1) \leqslant 1.$$

Proof:

This will be proven by induction on N.

- Let $N = 1$.
  Then, using Property 4.10,

  $$Y(1; 0,1) - Y(1; 0,0) = 0 \wedge Y(1; 1,1) - Y(1; 1,0) = 1,$$

  hence, for all $n \in [0..1]$ and all $i \in [1..1]$,

  $$0 \leqslant Y(1; n, i) - Y(1; n, i-1) \leqslant 1.$$

- Let $N > 1$, and assume that for all $M < N$, and all $n \in [0..M]$, $i \in [1..M]$,

  $$0 \leqslant Y(M; n, i) - Y(M; n, i-1) \leqslant 1.$$

  Let $n \in [0..N]$ and $i \in [1..N]$.
  If $i \leqslant N \underline{\text{div}}\, 2$, then

  $$Y(N; n, i) - Y(N; n, i-1) = Y(N \underline{\text{div}}\, 2; n \underline{\text{div}}\, 2, i) - Y(N \underline{\text{div}}\, 2; n \underline{\text{div}}\, 2, i-1),$$

  and thus, applying the induction assumption,

  $$0 \leqslant Y(N; n, i) - Y(N; n, i-1) \leqslant 1.$$

  For $i > N \underline{\text{div}}\, 2$, it may be proven in the same way.

□

The following property may be proven in a similar way.

Property 4.13:

For all $N \in \mathbf{N}$, $N > 0$, and all $n \in [0..N-1]$, $i \in [1..N]$,

$$Y(N; n, i) - Y(N; n, i-1) = 1 \Rightarrow Y(N; n+1, i) - Y(N; n+1, i-1) = 1.$$

□

Corollary 4.14:

For the Adapted CJ function g, $(\sigma_n(g))_{n \in [0..N]}$ satisfies the non-decreasing condition and the once-a-one condition.

□

Hence, the Adapted CJ function induces a convex line function on domain D. In the following section this convex function will be defined formally, and its deviation will be investigated.

4.3 A convex line function based on the Adapted CJ function

In this section we shall define a convex line function f, which is based on the Adapted CJ function g in a way as described in Section 4.0.

Let $N \in \mathbb{N}, N > 0$. For $p, q \in [0..N]^2$, such that $\hat{q} - \hat{p} \in O_0$, we shall define $f[p,q]$ as the subset of $g[t_0, t_1]$ between p and q, where t_0, t_1 satisfy

$$t_0.x = 0 \wedge t_1.x = N \wedge t_1 - t_0 \in O_0 \wedge p, q \in g[t_0, t_1]. \tag{a}$$

From the previous three sections we know that this is a consistent definition.

For $p, q \in [0..N]^2$ such that $\hat{q} - \hat{p} \notin O_0$, we shall use the functions f_i again, which were introduced in Chapter 2.

For $p, q \in [0..N]^2$, such that $\hat{q} - \hat{p} \in O_1$, we can simply apply the above definition to $f_1^{-1}(p)$ and $f_1^{-1}(q)$, since $f_1^{-1}(\hat{p}) - f_1^{-1}(\hat{q}) \in O_0$, and then transform the resulting set back with f_1.

For $p, q \in [0..N]^2$, such that $\hat{q} - \hat{p} \in O_6 \cup O_7$, we first translate p and q downwards such that $\hat{p}$ is situated at the x-axis, then apply transformation f_7^{-1}, and next the definition is applied to the thus transformed endpoints. The resulting set is then transformed and translated back again.

More formally, for $p, q \in [0..N]^2$, we define $f[p,q]$ as

$$f[p,q] := \begin{cases} \{ r \in g[t_0, t_1] \mid r.x \in [\hat{p}.x .. \hat{q}.x] \} & \text{if } \hat{q} - \hat{p} \in O_0 \\ f_1 \circ f[f_1^{-1}(p), f_1^{-1}(q)] & \text{if } \hat{q} - \hat{p} \in O_1 \\ (0, \hat{p}.y) \oplus f_7 \circ f[f_7^{-1}(p - (0, \hat{p}.y)), f_7^{-1}(q - (0, \hat{p}.y))] & \text{if } \hat{q} - \hat{p} \in O_6 \cup O_7. \end{cases}$$

where t_0, t_1 satisfy (a).

Based on the above definition, and on the discussions in the previous sections, the following property may be formulated.

Property 4.15:

On the domain $[0..N]^2 \times [0..N]^2$, the function f satisfies the line function conditions lf0 and lf1. Furthermore, f is minimal and convex.

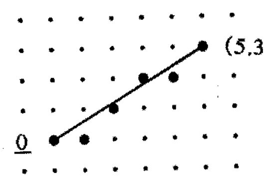
□

f is not translation invariant. For instance, if $N=5$, then for $p_0=\underline{0}$ and $q_0=(3,2)$,

$$f[p_0,q_0] = \{\underline{0},(1,0),(2,1),(3,2)\}$$

and for $p_1=(1,0)$ and $q_1=(4,2)$,

$$f[p_1,q_1] = \{(1,0),(2,1),(3,2),(4,2)\}.$$



See the figure alongside, where $g[(\underline{0},(5,3)]$ is shown. Hence,

$$[p_1,q_1] = (1,0) \oplus [p_0,q_0],$$

whereas

$$f[p_1,q_1] \neq (1,0) \oplus f[p_0,q_0].$$

However, f may be considered to be partly translation invariant, namely in the y-direction for p,q such that $\hat{q}-\hat{p} \in O_0 \cup O_7$, and in the x-direction for p,q such that $\hat{q}-\hat{p} \in O_1 \cup O_6$. This is expressed more formally in the following property.

Property 4.16:

For all $p,q \in [0..N]^2$ such that $\hat{q}-\hat{p} \in O_0 \cup O_7$, and all $r \in \mathbf{Z}^2$ such that $r.x=0$ and $r+p, r+q \in [0..N]^2$,

$$f[r+p,r+q] = r \oplus f[p,q].$$

Similarly, for all $p,q \in [0..N]^2$ such that $\hat{q}-\hat{p} \in O_1 \cup O_6$, and all $r \in \mathbf{Z}^2$ such that $r.y=0$ and $r+p, r+q \in [0..N]^2$,

$$f[r+p,r+q] = r \oplus f[p,q].$$

Proof:

Let $p,q \in [0..N]^2$ such that $\hat{q}-\hat{p} \in O_0$, and $r \in \mathbf{Z}^2$ such that $r.x=0$ and $r+p, r+q \in [0..N]^2$. Then,

$f[r+p,r+q]$

$=$ { definition f and $r.x = 0$ }

$\{s \in g[t_0,t_1] \mid s.x \in [\hat{p}.x..\hat{q}.x]\}$,
where $t_0.x=0 \wedge t_1.x=N \wedge t_1-t_0 \in O_0 \wedge r+p, r+q \in g[t_0,t_1]$

$=$ { renaming t_0-r as t_0 and t_1-r as t_1 }

$\{s \in g[r+t_0,r+t_1] \mid s.x \in [\hat{p}.x..\hat{q}.x]\}$,
where $t_0.x=0 \wedge t_1.x=N \wedge t_1-t_0 \in O_0 \wedge r+p, r+q \in g[r+t_0,r+t_1]$

$=$ { translation invariance of g and definition of $\oplus$ }

$r \oplus f[p,q]$.

The cases in which $\hat{q}-\hat{p} \in O_1 \cup O_6 \cup O_7$, may be reduced to the case $\hat{q}-\hat{p} \in O_0$.

□

From the previous chapter we know that an upper bound for the deviation of the Adapted CJ function g is given by

$$E_g(d) \leqslant \tfrac{1}{3}\left\lceil {}^2\log d \right\rceil + \tfrac{1}{9}(1-(-\tfrac{1}{2})^{\left\lceil {}^2\log d \right\rceil}).$$

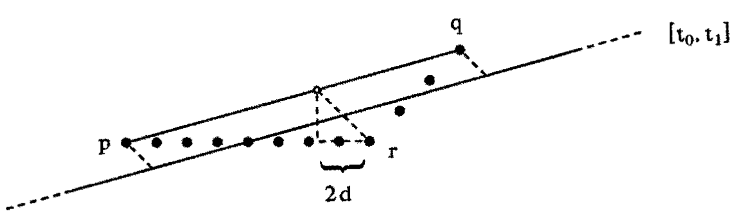
A general upper bound for $E_f(d)$, where $d \in [0..N]$, is given by

$$E_f(d) \leqslant 2E_g(N) \leqslant 2*(\frac{l}{3} + \frac{1-(-\frac{1}{2})^l}{9}), \quad \text{where } l = \left\lceil {}^2\log N \right\rceil. \qquad \text{(b)}$$

This is illustrated in the figure below, where p, q, and r (where $r \in f[p,q]$) are supposed to have maximum distance

$$d = \frac{l}{3} + \frac{1-(-\frac{1}{2})^l}{9}$$

to the line segment $[t_0, t_1]$. In this case, $\tilde{d}(r, [p,q])$ may equal $2d$.



However, by making a good choice for N, this upper bound may be decreased. For instance, if $N = 2^l$, then the pixels of $g[\underline{0}, (N,n)]$ are all beneath the line segment $[\underline{0}, (N,n)]$, namely, for all pixels $(x, Y(2^l;n,x)) \in g[\underline{0}, (N,n)]$,

$$Y(2^l;n,r.x) \leqslant \frac{n*r.x}{2^l},$$

which may be proven by induction on l. Hence, the factor 2 in (b) may be eliminated, and we may state that if N is a power of 2, for all $d \in [0..N]$,

$$E_f(d) \leqslant E_g(N) \leqslant \frac{l}{3} + \frac{1-(-\frac{1}{2})^l}{9}.$$

Instead of using the upper bound from Chapter 3 for the deviation $E_g(N)$, we may also try to find an exact expression for $E_g(N)$, using the function Y.

$E_g(N)$

$=$ { definition E_g and translation invariance of g }

$(\underline{\max} q : q \in O_0 \wedge d(\underline{0}, q) = N : e0_g[\underline{0}, q])$

$=$ { definition of O_0 and d }

$(\underline{\max} n : n \in [0..N] : e0_g[\underline{0}, (N, n)])$

$=$ { definition of $e0_g$ }

$(\underline{\max} n : n \in [0..N] : (\underline{\max} i : i \in [0..N] : \tilde{d}((i, Y(N;n,i)), [\underline{0}, (N,n)])))$

$=$ { using Property 2.2 }

$$(\underline{\max} n : n \in [0..N] : (\underline{\max} i : i \in [0..N] : \frac{N}{n+N} \mid Y(N;n,i) - \frac{i*n}{N} \mid)$$

$=$ { arithmetic }

$$(\underline{\max} n : n \in [0..N] : \frac{1}{n+N}(\underline{\max} i : i \in [0..N] : \mid N*Y(N;n,i) - i*n \mid)).$$

If N is restricted to have a certain level, that is, if $\lceil {}^2\log N \rceil = l$, then one can search for the N that minimises $E_g(N)$. For instance, for $l = 4$, it turns out that $E_g(16) = 1.105$, whereas $E_g(9) = 0.8$. We have not worked this out any further, but we expect that $E_g(N)$, and hence a general upper bound for $E_f(d)$, may decrease if N is chosen appropriately.

Since f is not translation invariant, it is difficult to find an exact expression for $E_f(d)$, where $d \in [0..N]$.

4.4 Permutations

In this section we shall show that each permutation of the numbers $1, \ldots, N$ induces a sequence of chain codes $(\gamma_n)_{n \in [0..N]}$, each of length N, that satisfies both the non-decreasing condition and the once-a-one condition, and for which $h(\gamma_n) = n$. Consequently, each permutation induces a convex, minimal line function on domain $D = [0..N]^2 \times [0..N]^2$.

We shall also show that any sequence of chain codes $(\gamma_n)_{n \in [0..N]}$, each of length N, that satisfies both the non-decreasing condition and the once-a-one condition, and for which $h(\gamma_n) = n$, induces a permutation. Hence, the class of convex line functions constructed as described in Section 4.0 is fully characterised by the class of permutations of N numbers. Consequently, $N!$ different convex line functions may be constructed that way.

A function $G : [1..N] \rightarrow [1..N]$ is a permutation if and only if for all $n \in [1..N]$,

$$(\underline{E} i : i \in [1..N] : G(i) = n).$$

Permutations will be denoted as $(G(1)\ G(2) \cdots G(N))$.

Suppose that the function $G:[1..N]\to[1..N]$ is a permutation. We shall associate a sequence of chain codes with G that satisfies the once-a-one condition. Informally spoken, we may interpret $G(i)=m$ as: the first 1 at index i occurs in γ_m. Formally, we define the sequence of chain codes $(\gamma_n(G))_{n\in[0..N]}$, each of length N, as follows.

$$\gamma_n(G)[i] := \begin{cases} 0 & \text{if } n < G(i) \\ 1 & \text{if } n \geqslant G(i). \end{cases} \tag{a}$$

It then follows immediately that $(\gamma_n(G))_{n\in[0..N]}$ satisfies the non-decreasing condition and once-a-one condition.

Example 4.17:

The permutations $G_0=(1\,2\,3\,4\,5)$ and $G_1=(3\,5\,2\,1\,4)$ result in the chain code sequences

$\gamma_0(G_0) = 00000$	$\gamma_0(G_1) = 00000$
$\gamma_1(G_0) = 10000$	$\gamma_1(G_1) = 00010$
$\gamma_2(G_0) = 11000$	$\gamma_2(G_1) = 00110$
$\gamma_3(G_0) = 11100$	$\gamma_3(G_1) = 10110$
$\gamma_4(G_0) = 11110$	$\gamma_4(G_1) = 10111$
$\gamma_5(G_0) = 11111$	$\gamma_5(G_1) = 11111$

respectively, which correspond to the pixel sets shown in Figure 4.4 and Figure 4.5.

□

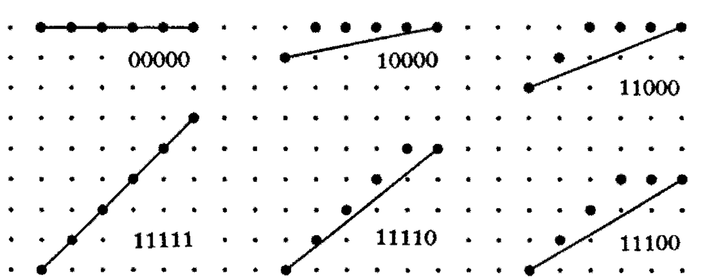


Figure 4.4

The chain codes and corresponding pixel sets, associated with the permutation $(1\,2\,3\,4\,5)$.

The sequence of chain codes $(\gamma_n(G))_{n\in[0..N]}$ as defined by (a) also satisfies $h(\gamma_n)=n$, as will now be proven.

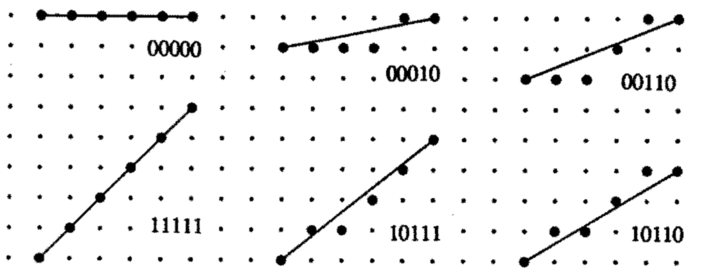


Figure 4. 5

The chain codes and corresponding pixel sets, associated with the permutation (3 5 2 1 4).

Property 4.18:

For all permutations $G:[1..N]\to[1..N]$, the sequence of chain codes $(\gamma_n(G))_{n\in[0..N]}$ defined by (a) satisfies, for all $n\in[0..N]$,

$$h(\gamma_n(G)) = n.$$

Proof:

$$
\begin{aligned}
& h(\gamma_n(G)) \\
= \quad & \{\text{ definition of } w \text{ and } \gamma_n(G)\in\{0,1\}^* \ \} \\
& \#\{i\in[1..N] \mid \gamma_n(G)[i]=1\} \\
= \quad & \{\text{ definition } \gamma_n(G)\ \} \\
& \#\{i\in[1..N] \mid G(i)\leqslant n\} \\
= \quad & \{\ G \text{ is a permutation }\} \\
& \#\{i\in[1..N] \mid i\leqslant n\} \\
= \quad & n.
\end{aligned}
$$

□

Thus, we have shown that each permutation induces a sequence of chain codes $(\gamma_n)_{n\in[0..N]}$ that satisfies the non-decreasing condition, the once-a-one condition, and $h(\gamma_n)=n$, for all $n\in[0..N]$.

Reversely, any sequence of chain codes $(\gamma_n)_{n\in[0..N]}$ satisfying these conditions induces a permutation, as will now be shown.

Let $(\gamma_n)_{n\in[0..N]}$ be a sequence of chain codes, each of length N, that satisfies the non-decreasing condition, the once-a-one condition, and $h(\gamma_n)=n$ for all $n\in[0..N]$.
Define the function $G:[1..N]\to[1..N]$ as follows.

$$G(i) := (\underline{\min}\ m : m \in [1..N] \wedge \gamma_m[i] = 1 \wedge \gamma_{m-1}[i] = 0 : m\ ).$$

Note that for any $i \in [1..N]$ this minimum exists, since $\gamma_0[i] = 0$ and $\gamma_N[i] = 1$.

Property 4.19:

G is a permutation.

Proof:

Let $i, j \in [1..N]$, $i \neq j$.
Suppose $G(i) = G(j) = n$.
Then $\gamma_n[i] = 1$, $\gamma_{n-1}[i] = 0$, $\gamma_j[i] = 1$, and $\gamma_{j-1}[i] = 0$. Hence, $h(\gamma_n) \geq h(\gamma_{n-1}) + 2$. This contradicts with $h(\gamma_n) = n$ and $h(\gamma_{n-1}) = n-1$.
Hence, $G(i) \neq G(j)$, and thus G is a permutation.

□

Example 4.20:

For $N=5$, we have seen in Figure 4.3 that the sequence of chain codes $(\sigma_n(g))_{n \in [0..N]}$, where g is the Adapted CJ function, is

$\sigma_0(g) = 00000$
$\sigma_1(g) = 00001$
$\sigma_2(g) = 01001$
$\sigma_3(g) = 01101$
$\sigma_4(g) = 11101$
$\sigma_5(g) = 11111.$

This corresponds to the permutation $G = (4\,2\,3\,5\,1)$.

□

Now the question arises whether a permutation exists, for given N, that induces a convex line function that is close.

In [Luby 1986], a similar construction of convex line functions is presented, based on so-called N-trees, and it is shown that there is a one-to-one correspondence between N-trees and permutations of N numbers. Hence, a one-to-one correspondence exists between N-trees and chain code sequences $(\sigma_n)_{n \in [0..N]}$ that satisfy the non-decreasing condition, once-a-one condition, and $h(\sigma_n) = n$ for all $n \in [0..N]$. Such chain codes provide a one-to-one correspondence between 8-connected paths (the paths we consider) and 4-connected paths (the paths Luby considers). Since Luby measures distances of paths with the same distance function d as we use, his results are applicable to our situation. Theorem 19 of his paper expresses that any N-tree induces a path of length N that contains a pixel with distance at least $c \log(N) - 1$ (where $c = 1/200$) to the line segment connecting the end-points of the path. Applying this to our situation, this means that for a convex line function f on domain $[0..N]^2 \times [0..N]^2$, induced by a permutation in a way described in this chapter, a lower bound exists for $E_f(N)$ that is logarithmic in N. Hence, such a function cannot be close for N large enough.

4.5 Concluding remarks

In this chapter we have investigated the construction of a class of convex line functions on domain $D = [0..N]^2 \times [0..N]^2$.

The construction is based on a translation invariant, minimal line function that must satisfy two conditions, namely the all height condition and the equal height condition. These conditions are proven to be equivalent with the non-decreasing condition and the once-a-one condition. We have proved that any permutation of N numbers induces a sequence of chain codes that satisfies the non-decreasing and once-a-one conditions. Thus, any permutation of N numbers induces a convex line function. Furthermore, any convex line function that is based on a definition scheme as used in this chapter, is shown to induce a permutation.

We have also shown that the Adapted CJ function induces a convex line function f whose deviation value $E_f(N)$ has an upper bound which is logarithmic in N. [Luby 1986] shows that for any convex line function f that is based on a permutation, $E_f(N)$ is at least logarithmic in N. The question remains what permutation minimises the value $E_f(N)$.

Although the line functions presented in this chapter are not close, it remains an open question whether a close, convex line function on domain $\mathbf{Z}^2 \times \mathbf{Z}^2$ exists. We conjecture that such a function does not exist.

For those who are interested in algorithms for the generation of the line functions presented in this chapter, we refer to [van Lierop et al 1986], where a recursive and a non-recursive algorithm are presented that generate the values of the convex line function associated with the Adapted CJ function. The recursive algorithm has time complexity $O(\log^2 N)+O(d(p,q))$. The non-recursive algorithm, which requires preprocessing time $O(N)$, has time complexity $O(d(p,q))$. In that same paper, also algorithms can be found for convex digitisation functions for line segments with endpoints in $\mathbf{Q}^2$. The recursive one has time complexity $O(\log N)+O(d(p,q))$, and the non-recursive one $O(d(p,q))$.

5

Final remarks

5.0 Results

In this thesis digitisation is dealt with in a formal way, uncoupled from algorithms. Any function from $\mathbf{P}(\mathrm{R}^2)$ to $\mathbf{P}(\mathrm{Z}^2)$ is considered to be a digitisation function. The prescription of a digitisation function may be used as a formal specification for an algorithm that has to generate values of that digitisation function.

Desirable properties of digitisation functions have been formulated, and several examples of line functions, illustrating the broad range of possibilities, have been discussed with respect to these properties.

An important result is that any translation invariant, minimal, convex line function has a deviation which is at least linear in the distance of the endpoints (Theorem 2.55), hence such a function cannot be close. This might explain why convexity is hardly paid attention to in Computer Graphics literature, since most common line functions are translation invariant, minimal, and close. Nevertheless, apart from its theoretical interest, convexity might be a desirable property in applications where windowing is used ([Luby 1986]).

Because of the above result, one has to abandon translation invariance or minimality if one wants to find convex line functions with smaller deviation values. We have described a construction method for minimal, convex line functions on a limited domain $D = [0..N]^2 \times [0..N]^2$; this method is based on a translation invariant, minimal line function that satisfies certain conditions, called the once-a-one condition and the non-decreasing condition. The Adaptive CJ function, which is one of the recursive line functions discussed in Chapter 3, happens to satisfy these conditions. The deviation of the convex line function that is derived from the Adapted CJ function has a general upper bound that is logarithmic in N.

Furthermore, it has been proven that a one-to-one correspondence exists between, on the one hand, translation invariant, minimal line functions on domain D that satisfy the non-decreasing and once-a-one conditions, and on the other hand, permutations of N numbers. Consequently, each permutation of N numbers induces a minimal, convex line function on domain D. A general upper bound for the deviation of these functions is at least logarithmic in N.

The consideration that closeness is a desirable property instead of a necessary one, opens the way to new kinds of digitisation functions, like the recursive ones presented in Chapter 3. The operations needed for straightforward implementations are additions, shifts, and comparisons, all easily implementable in hardware. Furthermore, since the recursive nature of these functions may be exploited by the use of parallel processors, these functions seem to be good candidates for future hardware implementations. The results of the work of Corthout et al support this conjecture. See [Corthout & Jonkers 1986b].

We have the following recommendations for the choice of a line function.

- If closeness is required, the Bresenham, Close Embedding, or Optimal Embedding functions are appropriate.
- If convexity is required, the function derived from the Adapted CJ function is a candidate, or any other function that may be derived from a permutation, as described in Chapter 4.
- If fast hardware implementations have the highest priority, the recursive functions from Chapter 3 should be considered.

5.1 Remaining questions

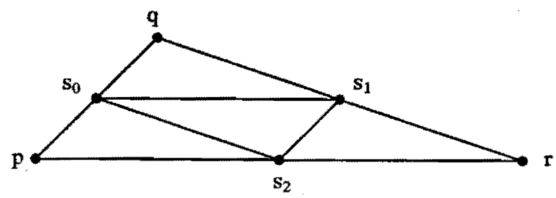
In this thesis we have focussed on the properties closeness and convexity. Of course, other properties may be conceivable. [Serra 1982] imposes four criteria on morphological transformations (of which digitisation functions may be considered to be a special class), namely translation invariance, scale invariance, local knowledge, and semi-continuity. It should be investigated how functions can be defined that have these properties, and how these properties combine with the ones mentioned in this thesis.

As remarked in Section 1.6, the definition of closeness might also be based on the definition of a region of sensitivity. The implications of such a definition should be investigated.

An interesting question that remains is whether closeness and convexity are conflicting properties or not. We conjecture that they are conflicting indeed.

An other subject that deserves further investigations is the search for the permutation that induces the convex line function with the smallest deviation.

The main part of this thesis deals with line functions. To what extent may the theory presented in this thesis be applied to other kinds of objects, like polygons, for example? If the idea of splitting line segments into two more or less equal parts is generalised to triangles, (see the figure below) we come to the following definition of a triangle digitisation function. The triangle with vertices p, q, r is denoted as $[p, q, r]$.



$$f[p,q,r] := \begin{cases} \{p,q,r\} & \text{if } d(p,q)\leqslant 1 \wedge d(q,r)\leqslant 1 \wedge d(r,p)\leqslant 1 \\ f[s_0,s_1,s_2] \cup f[p,s_0,s_1] & \text{otherwise} \\ \quad \cup\, f[q,s_0,s_2] \cup f[r,s_1,s_2] & \end{cases}$$

where s_0, s_1, s_2 are split points of $[p,q]$, $[q,r]$, and $[r,p]$ respectively.

In [van Overveld & van Lierop 1986] algorithms can be found for the digitisation of triangles and triangular patches, based on the above definition. In [Corthout & Jonkers 1986a] point containment tests are presented for similarly defined Bezier shapes.

Digitisation is only one aspect of the display process. Other aspects inherent to the rendering of objects, such as anti-aliasing and shading, have not been addressed. Both anti-aliasing and shading impose an intensity or colour value on each pixel.

A common technique for anti-aliasing is to incorporate it into the digitisation function, for example by making the intensity value of a pixel depend on the area of the part of the pixels region of sensitivity covered by the object. This requires a lot of computations.

Fortunately, there is another anti-aliasing technique, one that is independent of the digitisation. In this case, anti-aliasing is considered as a mapping from the frame buffer FB (where the colour of each pixel is stored) into FB; the anti-aliasing algorithm postprocesses the result of the digitisation algorithm.

It is almost impossible to separate shading from digitisation; furthermore, it is difficult to avoid real arithmetic. However, in [van Overveld & van Lierop 1986] a start is made by the incorporation of shading into the digitisation algorithm using integer arithmetic only. [van Overveld 1987b] is a continuation hereof.

References

Aho, A.V., J.E. Hopcroft, and J.D. Ullman, 1974
The Design and Analysis of Computer Algorithms.
London: Addison-Wesley, 1974.

Berckel, J.A.Th.M. van, and B.J. Mailloux, 1965
Some Algol Plotting Procedures.
Amsterdam: Stichting Mathematisch Centrum, 1965.
(Report MR 73)

Bresenham, J.E., 1965
Algorithm for Computer Control of a Digital Plotter.
IBM Systems Journal **4** (1965), 25-30.

Bresenham, J.E., 1985
Run Length Slice Algorithm for Incremental Lines.
in: Fundamental Algorithms for Computer Graphics.
(Proceedings of NATO ASI Fundamental Algorithms for Computer Graphics); ed. by R.A. Earnshaw
Berlin: Springer-Verlag, 1985, 59-104.

Bresenham, J.E., 1986
Ambiguities in Incremental Line Rastering.
Submitted for publication.

Brons, R., 1974
Linguistic Methods for the Description of a Straight Line on a Grid.
Computer Graphics and Image Processing **2** (1974), 48-62.

Cook, R.L., 1986
Picture Debugging.
in: ACM Siggraph Tutorial Lectures;
16: Image Rendering Tricks, 1986.

Corthout, M.E.A., and H.B.M. Jonkers, 1986a
The Transformational Development of a New Point Containment Algorithm.
Philips Journal of Research **41** (1986), 83-174.

Corthout, M.E.A., and H.B.M. Jonkers, 1986b
A New Point Containment Algorithm for B_regionsin the Discrete Plane.
to appear in: Proceedings of NATO ASI Geometrical Algorithms for Computer Graphics and CAD.

Crow, F.C., 1977
The Aliasing Problem in Computer Shaded Images.
Communications of the ACM **20** (1977), 799-805.

Dijkstra, E.W., 1976
A Discipline of Programming.
Englewood Cliffs: Prentice-Hall, 1976.

Dijkstra, E.W., and W.H.J. Feijen, 1984
Een Methode van Programmeren (in Dutch).
Den Haag: Academic Service, 1984.

Dorst, L., 1986
Discrete Straight Line Segments: Parameters, Primitives and Properties.
Ph.D. Thesis, Delft: University of Technology, 1986.

Eilenberg, S., and C.C. Elgot, 1970
Recursiveness.
New York: Academic Press, 1970.
(ACM Monograph Series: 7001)

Foley, J.D., and A. van Dam, 1982
Fundamentals of Interactive Computer Graphics.
Reading: Addison-Wesley, 1982.

Forrest, A.R., 1985
Computational Geometry in Practice.
in: Fundamental Algorithms for Computer Graphics,
(Proceedings of NATO ASI Fundamental Algorithms for Computer Graphics); ed. by R.E. Earnshaw
Berlin: Springer-Verlag, 1985, 707-724.

Franklin, Wm. R., 1986
Problems with Raster Graphics Algorithms.
in: Data Structures for Raster Graphics,
(Proceedings of a Workshop held at Steensel, The Netherlands, June 24-28, 1985);
ed. by L.R.A. Kessener, F.J. Peters, and M.L.P. van Lierop.
Berlin: Springer-Verlag, 1986, 1-7.

Freeman, H., 1961
On the Encoding of Arbitrary Geometric Configurations.
IRE Transactions on Electronic Computers **10** (1961), 260-268.

Freeman, H., 1970
Boundary Encoding and Processing.
in: Picture Processing and Psychopictorics; ed. by B.S. Lipkin and A. Rosenfeld.
New York: Academic Press, 1970, 241-266.

Hung, S.H.Y., 1985
On the Straightness of Digital Arcs.
IEEE Transactions on Pattern Analysis and Machine Intelligence 7 (1985), 203-215.

Knuth, D.E., 1973
The Art of Computer Programming Vol.3: Sorting and Searching.
London: Addison-Wesley, 1973.

Lierop, M.L.P. van, C.W.A.M. van Overveld, and H.M.M. van de Wetering, 1986
Line Rasterisation Algorithms that Satisfy the Subset Line Property.
to appear in: *Computer Vision, Graphics, and Image Processing.*

Luby, M.G., 1986
Grid Geometries which Preserve Properties of Euclidean Geometry: A Study of Graphics Line Drawing Algorithms.
Toronto: University of Toronto, 1986.
(Technical Report 190/86)

Newman, M., and R.F. Sproull, 1979
Principles of Interactive Computer Graphics.
London: McGraw-Hill, 1979.

Overveld, C.W.A.M. van, 1986
Applications of the Method of Invariants in Computer Graphics.
in: Data Structures for Raster Graphics,
(Proceedings of a Workshop held at Steensel, The Netherlands, June 24-28, 1985);
ed. by L.R.A. Kessener, F.J. Peters, and M.L.P. van Lierop.
Berlin: Springer-Verlag, 1986, 173-180.

Overveld, C.W.A.M. van, and M.L.P. van Lierop, 1986
A Consistent Algorithm to Fill Triangles and Triangular Patches.
in: Eurographics'86 Proceedings; ed. by A.A.G. Requicha.
Amsterdam: North-Holland, 1986, 153-160.

Overveld, C.W.A.M. van, 1987a
A Family of Algorithms for Generating Discrete Embeddings of Continuous Objects.
to appear in: Proceedings of NATO ASI Geometrical Algorithms for Computer Graphics and CAD.

Overveld, C.W.A.M. van, 1987b
Integer Implementation and Application of Bilinear Blended Surfaces.
submitted for publication.

Rogers, D.F., 1985
Procedural Elements of Computer Graphics.
London: McGraw-Hill, 1985.

Roo, J. de, J. van Daele, A. Oosterlinck, and H. Van den Berghe, 1980
A Digital Signal Generating and Encoding Algorithm.
in: Signal Processing: Theories and Applications; ed. by M. Kunt and F. de Coulon.
Amsterdam: North-Holland Publishing Company, 1980, 377-381.

Rosenfeld, A., 1974
Digital Straight Line Segments.
IEEE Transactions on Computing **23** (1974), 1264-1269.

Rosenfeld, A., 1978
Geodesics in Digital Pictures.
Information and Control **36** (1978), 74-84.

Rosenfeld, A., and J.L. Pfaltz, 1968
Distance Functions on Digital Pictures.
Pattern Recognition **1** (1968), 33-61.

Serra, J., 1982
Image Analysis and Mathematical Morphology.
London: Academic Press, 1982.

Shreider, Yu.A., 1974
What is Distance?
Chicago: The University of Chicago Press, 1974.

Tanenbaum, A.S., 1984
Structured Computer Organization.
Englewood Cliffs: Prentice-Hall, 1984.

Wu, L.-D., 1982
On the Chain Code of a Line.
IEEE Transactions on Pattern Analysis and Machine Intelligence **4** (3) (1982), 347-353.

Yasuhara, A., 1971
Recursive Function Theory & Logic.
New York: Academic Press, New York, 1971.
(Computer Science and Applied Mathematics; 7102)

Index

Summary

For the generation of digital images from continuous objects, a digitisation mapping from $\mathbf{R}^2$ (or $\mathbf{R}^3$) to $\mathbf{Z}^2$ is required. Thus far, in Computer Graphics literature, digitisation is dealt with by presenting algorithms; formal specifications of these algorithms are seldom presented. In this thesis we abstract from algorithms; we present a framework in which desirable properties of digitisation functions are formulated formally (Chapter 1).

One of these properties is *closeness*, which expresses that the digitisation of an object should correspond as good as possible to the original object. Other properties are *translation invariance* and *convexity*. Thus far, closeness has been considered to be a necessary requirement for digitisation functions. In this thesis, however, we also deal with digitisation functions that are not close. A measure is introduced to express the quality of these functions with regard to closeness; this measure is called the *vicinity function* (Chapter 1).

To show the implications of this new view on digitisation, we concentrate on digitisation functions for line segments, shortly called line functions. For these functions a special kind of vicinity measure is introduced, called the *deviation function*. Apart from the properties mentioned above, a new property for line functions is formulated, namely *minimality*. Several examples of line functions are discussed (Chapter 2).

It is proved that translation invariant, minimal, close line functions are not convex. It is even conjectured that closeness and convexity are conflicting properties (Chapter 2).

In Chapter 3, three recursive line functions are discussed. These line functions are not close; an upperbound for their deviation function is derived that is logarithmic in the distance of the endpoints.

In Chapter 4 we focus on convexity. We show that on a restricted domain D, a convex line function can be constructed from any translation invariant, minimal line function that satisfies certain conditions. It is shown that one of the recursive functions of Chapter 3 satisfies these conditions. Furthermore, it is proved that there is a one-to-one correspondence between line functions satisfying these conditions, and permutations. Consequently, each permutation induces a convex line function on domain D. None of these functions is close.

In short, this thesis deals with digitisation in a formal way; if properties like recursiveness or convexity are given priority over closeness, one must resort to other than the traditional digitisation functions.

Samenvatting

Om digitale beelden te kunnen maken van continue objecten is een digitalisatiefunctie nodig van $\mathbb{R}^2$ (of $\mathbb{R}^3$) naar $\mathbb{Z}^2$. In de computergrafische vakliteratuur werd digitalisatie tot nu toe afgedaan met het presenteren van algoritmen; van deze algoritmen worden zelden formele specificaties gegeven. In dit proefschrift abstraheren we van algoritmen; we presenteren een raamwerk waarin gewenste eigenschappen van digitalisatiefuncties formeel beschreven kunnen worden (Hoofdstuk 1).

Een van deze eigenschappen is *nabijheid*, die uitdrukt dat de digitalisatie van een object zo goed mogelijk moet corresponderen met het object zelf. Andere eigenschappen zijn *translatie-invariantie* en *convexiteit*. Tot nu toe werd nabijheid altijd als een nodige eigenschap van digitalisatiefuncties beschouwd. In dit proefschrift bestuderen we echter ook functies die niet nabij zijn. We introduceren een maat om de kwaliteit van deze functies met betrekking tot nabijheid in uit te drukken (Hoofdstuk 1).

Om te laten zien wat deze nieuwe kijk op digitalisatie voor gevolgen heeft, concentreren we ons op digitalisatiefuncties voor lijnstukken, in het kort lijnfuncties genoemd. Voor deze functies introduceren we een nieuwe nabijheidsmaat, namelijk de afwijkingsfunctie. Behalve de hierboven genoemde eigenschappen, formuleren we nog een andere gewenste eigenschap voor lijnfuncties, genaamd *minimaliteit*. Verscheidene voorbeelden van lijnfuncties worden behandeld (Hoofdstuk 2).

Er wordt bewezen dat translatie-invariante, minimale, nabije functies niet convex zijn. We vermoeden zelfs dat nabijheid en convexiteit elkaar uitsluiten (Hoofdstuk 2).

In Hoofdstuk 3 worden drie recursieve lijnfuncties besproken. Deze functies zijn niet nabij; er wordt een bovengrens voor hun afwijkingsfunctie afgeleid die logaritmisch is in de afstand van de eindpunten.

In Hoofdstuk 4 gaat het om convexiteit. We laten zien dat op een begrensd domein D, van elke translatie-invariante, minimale lijnfunctie die aan bepaalde voorwaarden voldoet, een convexe lijnfunctie afgeleid kan worden. Een van de recursieve lijnfuncties uit Hoofdstuk 3 voldoet aan deze eigenschappen. Bovendien bewijzen we dat er een een-eenduidige relatie bestaat tussen lijnfuncties die aan die eigenschappen voldoen, en permutaties. Daardoor induceert elke permutatie een convexe lijnfunctie op domein D. Deze lijnfuncties zijn niet nabij.

Kort samengevat: dit proefschrift behandelt op formele wijze het onderwerp digitalisatie; als aan eigenschappen zoals recursiviteit of convexiteit de voorkeur wordt gegeven boven nabijheid, dan moet men zijn toevlucht nemen tot andere dan de gebruikelijke digitalisatiefuncties.

Woorden van dank

Veel dank ben ik verschuldigd aan Kees van Overveld, wiens inspirerende ideeën geleid hebben tot het schrijven van dit proefschrift.

Rens Kessener en Frans Peters hebben als projectleiders een onderzoekssfeer geschapen waarin niet alleen het proefschrift, maar ook ikzelf kon groeien.

Huub van de Wetering, Jan Kalisvaart, Arnold Smeulders en Leo Dorst worden bedankt voor hun commentaar op eerdere versies.

Tom Verhoeff heeft mij ingewijd in de beginselen van PostScript, waarmee de plaatjes in dit proefschrift vervaardigd zijn.

Tenslotte bedank ik Ronald voor de altijd aanwezige steun op de achtergrond.

Curriculum vitae

Marloes van Lierop werd geboren 5 januari 1955 te Helmond. In 1973 haalde zij het einddiploma Atheneum-b aan het Peellandcollege te Deurne. Daarna studeerde zij wiskunde aan de Technische Universiteit Eindhoven, alwaar zij in februari 1983 het ingenieursexamen aflegde. Het afstudeerwerk werd verricht onder leiding van Prof.dr. M. Rem, en had betrekking op het genereren van topologische layouts van chips. Sinds 1 mei 1983 is zij, als wetenschappelijk ambtenaar in dienst van de Stichting Technische Wetenschappen, werkzaam binnen het STW-project Datastructuren voor Rastergrafiek, onder leiding van ing. L.R.A. Kessener en prof.dr. F.J. Peters, aan de Technische Universiteit Eindhoven.

STELLINGEN

behorend bij het proefschrift

Digitisation Functions
in
Computer Graphics

van

Marloes van Lierop

Eindhoven,
3 juli 1987

0. Het is gewenst dat digitalisatiefuncties een zo klein mogelijke afwijking hebben, maar dit hoeft, met name bij interaktieve grafische toepassingen, niet altijd de hoogste prioriteit te hebben.

1. Het ondoordacht gebruik van "floating point" arithmetiek heeft de populariteit van retoucheerprogramma's doen toenemen.

2. Van alle translatie invariante, minimale, convexe lijnfuncties, heeft de Adapted Franklin functie de kleinste afwijking.

3. Minimale lijnfuncties waarvan de "chain codes" aan de lineariteitscondities uit de Beeldbewerking voldoen, zijn niet convex.

   [L.-D. Wu, On the Chain Code of a Line, *IEEE* **PAMI 4** (3), 1982, pp. 347-353]

4. Zoals het gebruik van Z-buffers voor het berekenen van de zichtbare oppervlakken toenam naarmate computergeheugens compacter en goedkoper werden, zal in de toekomst het gebruik van Item Buffers voor het snel identificeren van op het scherm aangewezen objecten steeds gangbaarder worden.

   [M.L.P. van Lierop, Intermediate data structures for display algorithms, in: Data Structures for Raster Graphics, F.J. Peters, L.R.A. Kessener, M.L.P. van Lierop (eds.), Springer Verlag, 1986, pp. 39-55]

   [H. Weghorst, G. Hooper, D.P. Greenberg, Improved computational methods for ray tracing, *ACM Trans. on Graphics* **3** (1), 1984, pp. 52-69]

5. Bij het transformeren van een in "leaf codes" gecodeerd computerbeeld, kan de sorteerslag vermeden worden indien gebruik gemaakt wordt van de intrinsieke boomstructuur van deze codering.

[I. Gargantini, Translation, rotation, and superposition of linear quadtrees, *Int. J. of Man-Mach. Stud.* **18,** 1983, pp. 253-263]

[M.L.P. van Lierop, Geometrical Transformations on Pictures Represented by Leafcodes, *Computer Vision, Graphics, and Image Processing* **33,** 1986, pp. 81-98]

6. Door een "bottom-up" methode te combineren met het selectief uitstellen van sommige verbindingen, wordt de complexiteit van het ontwerpen van topologische chip layouts drastisch verlaagd, terwijl toch een grote mate van flexibiliteit gegarandeerd wordt.

[1 M.L.P. van Lierop, A flexible bottom-up approach for layout generation, *Integration, the VLSI journal* **3,** 1985, pp. 49-59]

7. Het bewijs van Stelling 3.3 in het boek "Computational Geometry" van Preparata en Shamos is fout.

[F.P. Preparata and M.I. Shamos, Computational Geometry: an Introduction, Springer-Verlag, 1985]

8. De gelden uit het Informatica Stimulerings Plan voor het LBO en MBO, zijn tot nu toe meer ten goede gekomen aan jongens dan aan meisjes.

9. De voorgedrukte kaarten die gebruikt worden om bij auteurs afdrukken van artikelen aan te vragen, met daarop de aanhef "Dear Sir", getuigen van stereotiep denken bij wetenschappers.