Pattern Recognition Letters 14 (1993) 83–88
North-Holland

February 1993

PATREC 1037

A new characterization of digital lines by least square fits

Robert A. Melter

Department of Mathematics, Long Island University, Southampton, NY 11968, USA

Ivan Stojmenović

Computer Science Department, University of Ottawa, Ottawa, Ontario K1N 9B4, Canada

Joviša Žunić

Institute of Basic Appl. Disc., Faculty of Engineering, University of Novi Sad, 21000 Novi Sad, Yugoslavia

Received 14 August 1992

Abstract

Melter, R.A., I. Stojmenović and J. Žunić, A new characterization of digital lines by least square fits, Pattern Recognition Letters 14 (1993) 83–88.

In this paper we prove that digital line segments and their least square line fits are in one-to-one correspondence and give a new simple representation (x_1,n,b_0,b_1) of a digital line segment, where x_1 and n are the x-coordinate of the left endpoint and the number of digital points, respectively, while b_0 and b_1 are the coefficients of the least square line fit $Y=b_0+b_1X$ for the given digital line segment. An O(n log n) time algorithm for obtaining a digital line segment from its least square line fit is described.

1. Introduction

Consider a line υ in the Euclidean plane with equation $y=\alpha x+\beta$. The line υ will be digitized by rounding. Its associated set of lattice points, called a digital line, is defined by

$$L(\upsilon) = \{(j, r(\alpha j+\beta)) \mid j \text{ is an integer}\}.$$

Here r is the rounding function with $r(n+0.5)=n$ if n is an integer. For convenience, we use a modified digitizing method, in which the first points below a given line are taken instead of rounding. Obviously it is equivalent to translating the line by -0.5 in the vertical direction and rounding. In general we will be dealing with finite subsets of $L(\upsilon)$. In particular, a digital (straight) line segment is obtained by digitizing a line segment by the above method.

The concept of a digital straight line has been formulated by Kim and Rosenfeld (1982). Given a set of n points in the plane, they gave an O(n) algorithm that determines whether or not the set is a digital line segment. Digital straight lines in

---

Correspondence to: R.A. Melter, Department of Mathematics, Long Island University, Southampton, NY 11968, USA.

This research is supported by NATO Collaborative Research Grant CRG 900840 and NSERC operating grant OGPIN007.

higher dimensions were studied by Kim (1983) and Stojmenović and Tosic (1991).

A finite set of points in the plane is sometimes called a scatter diagram. For convenience we assume that the points have distinct abscissas. The least square line for the scatter diagram is a line which minimizes the sum of the squares of the vertical distances to the data points. The formula for such a line can be found in textbooks on statistics, e.g., Burr (1974).

If the scatter diagram is given by $\{(x_i, y_i), i=1,2,\ldots,n\}$ and the equation of the least square line is $Y=b_0+b_1X$, then

$$b_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}$$

and

$$b_0 = \bar{y} - b_1\bar{x},$$

where

$$\bar{y} = \frac{y_1+\cdots+y_n}{n} \quad \text{and} \quad \bar{x} = \frac{x_1+\cdots+x_n}{n}.$$

In the case of the least square line for a digital line segment, we assume that $0 \leqslant b_1 \leqslant 1$ (or, equivalently, that the digital line segment maps one-to-one to the x-coordinate), because the other cases can be easily obtained by symmetry.

Melter and Rosenfeld (1989) reformulated the concept of a digital line. They introduced the concept of a noisy straight line segment, based on least square line fitting and defined in terms of bounds on correlation coefficients, and show that it is a generalization of a digital straight line.

Melter and Rosenfeld (1989) posed the following question:

Question 1. If a continuous line is digitized and least squares is applied to the scatter diagram, can the original line be recovered?

They gave only a partial answer, for some special case. In this paper we answer the question positively for any line segment. More precisely, we prove that the least square line fit uniquely determines the digital line on a segment. This statement is proved as Theorem 1 of Section 2. The following corollary is a consequence of the result.

Corollary 1. *Any digital line segment can be uniquely coded by four numbers* (x_1, n, b_0, b_1) *where* x_1 *and* n *are the x-coordinate of the left endpoint and the number of digital points, respectively, while* b_0 *and* b_1 *are the coefficients of the least square line fit* $Y=b_0+b_1X$ *for the given digital line segment.*

The corollary matches digital line segments with their least square line segments. The question that arises is whether a digital line segment (and, ultimately, the line segment which is digitized) can be recovered efficiently if our new representation is used. In Section 3 we give an efficient recovery procedure. The time complexity of the procedure is $O(n \log n)$.

Therefore we obtain a new representation of digital line segments by four numbers. This representation is an alternative to the known representation of digital lines by adjacent pairs given by Lindenbaum and Koplowitz (1991) and to the one suggested by Dorst and Smeulders (1984).

2. One-to-one correspondence between least square line fits and digital lines

In this section we prove the following theorem.

Theorem 1. *Let* $L(p)=\{(x_i, y_i), i=1,2,\ldots,n\}$ *be a digital line segment (the digitization of a line seg-*

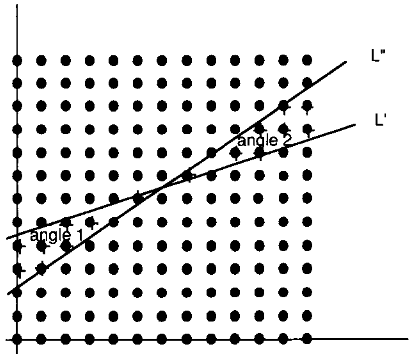

Figure 1.

ment p), where we can assume that $x_i = x_{i-1}+1$ *for* $1 < i \leqslant n$. *Let* $R(p)$ *be the least square line segment for the set* $L(p)$, *i.e.,* $R(P)$ *is limited to the interval* $[x_1, x_n]$. *Then* $R(p)=R(q)$ *implies* $L(P)=L(Q)$.

Proof. For digital lines we assume that $x_i = x_{i-1}+1$, $1 < i \leqslant n$. For simplicity (and without loss of generality), let $(x_1, \dots, x_n) = (0, 1, \dots, n-1)$. Then

$$\sum_{i=1}^{n} x_i = \frac{n(n-1)}{2}, \qquad \sum_{i=1}^{n} x_i^2 = \frac{(n-1)n(2n-1)}{6}.$$

Hence

$$\begin{aligned} b_1 &= \left[\frac{(n-1)n(2n-1)}{6} - \frac{n^2(n-1)^2}{4n}\right]^{-1} \\ &\quad \times \left[\sum_{i=1}^{n} x_i y_i - \frac{n-1}{2}\sum_{i=1}^{n} y_i\right] \\ &= \frac{12}{n^3-n}\left[\sum_{i=1}^{n} x_i y_i - \frac{n-1}{2}\sum_{i=1}^{n} y_i\right] \\ &= \frac{12}{n^3-n}\,(y_1, y_2, \dots, y_n) \\ &\qquad \cdot \left(-\frac{n-1}{2}, -\frac{n-1}{2}+1, \dots, \frac{n-1}{2}\right), \end{aligned}$$

where the dot product of last two vectors is taken.

The above is an alternative expression for b_1 which will be useful in the sequel. It is easy to find an example to show that different non-parallel digital lines may have the same coefficient b_1 in their least square line fits. One such example consists of the digital straight line segments $L_1 = \{(0,0), (1,0), (2,0), (3,0), (4,0), (5,1), (6,1), (7,1)\}$ and $L_2 = \{(0,0), (1,0), (2,0), (3,1), (4,1), (5,1), (6,1), (7,1)\}$ for which $b_1 = 5/28$.

The coefficient b_0 can be calculated as follows:

$$b_0 = \bar{y} - b_1\bar{x} = \frac{y_1 + \cdots + y_n}{n} - \frac{n(n-1)}{2n}\,b_1.$$

Suppose that two digital line segments L' and L'' are both defined on the interval $[0, n-1]$ and have the same coefficients in their least square line fits, i.e., $b_1' = b_1''$ and $b_0' = b_0''$. From $b_0' = b_0''$ it follows that

$$\frac{y_1' + \cdots + y_n'}{n} - \frac{n-1}{2}\,b_1' = \frac{y_1'' + \cdots + y_n''}{n} - \frac{n-1}{2}\,b_1''.$$

Since $b_1' = b_1''$ we obtain the following condition:

$$y_1' + \cdots + y_n' = y_1'' + \cdots + y_n''$$

or, equivalently,

$$(y_1' - y_1'') + (y_2' - y_2'') + \cdots + (y_n' - y_n'') = 0.$$

Figure 1 shows the digital line segments L' and L''. The differences $y_i' - y_i''$ correspond to the number of digital points between L' and L'' lying between these two lines and having the x-coordinate $x_i = i-1$. The positive differences correspond to digital points in angle 1 while negative differences correspond to digital points in angle 2. Clearly there exists an i such that $y_j' - y_j''$ is positive for $j \leqslant i$ and zero or negative for $j > i$. Thus

$$\begin{aligned} &(y_1' - y_1'') + \cdots + (y_i' - y_i'') \\ &\quad = (y_{i+1}'' - y_{i+1}') + \cdots + (y_n'' - y_n') \end{aligned} \tag{1}$$

where all numbers in parentheses are nonnegative. This means that the numbers of digital points (points with integer coordinates) in angle 1 and angle 2 (these angles are formed by L' and L'') are the same. If digital line segments corresponding to L' and L'' are different then each of the two angles contains some digital points.

From (1) we also have

$$\begin{aligned} &(y_1' - y_1'') + 2(y_2' - y_2'') + \cdots + i(y_i' - y_i'') \\ &\quad \leqslant i((y_1' - y_1'') + \cdots + (y_i' - y_i'')) \\ &\quad = i((y_{i+1}'' - y_{i+1}') + \cdots + (y_n'' - y_n')) \\ &\quad \leqslant (i+1)(y_{i+1}'' - y_{i+1}') + \cdots + n(y_n'' - y_n'). \end{aligned}$$

The later inequality $\leqslant$ is clearly strictly $<$ whenever angle 2 (and hence angle 1) contains at least one digital point. Therefore if L' and L'' are different digital line segments then

$$\begin{aligned} &(y_1' - y_1'') + 2(y_2' - y_2'') + \cdots + i(y_i' - y_i'') \\ &\quad < (i+1)(y_{i+1}'' - y_{i+1}') + \cdots + n(y_n'' - y_n'). \end{aligned} \tag{2}$$

Let $a = -(n-1)/2$. From $b_1' = b_1''$ it follows that

$$\begin{aligned} &(y_1', \dots, y_n')(a, a+1, \dots, a+n-1) \\ &\quad = (y_1'', \dots, y_n'')(a, a+1, \dots, a+n-1), \end{aligned}$$

i.e.

$$\begin{aligned} &(a-1)(y_1' + \cdots + y_n') + y_1' + 2y_2' + \cdots + ny_n' \\ &\quad = (a-1)(y_1'' + \cdots + y_n'') + y_1'' + 2y_2'' + \cdots + ny_n''. \end{aligned}$$

By applying $y_1' + \cdots + y_n' = y_1'' + \cdots + y_n''$ to the last equality one obtains

$$y_1' + 2y_2' + \cdots + ny_n' = y_1' + 2y_2'' + \cdots + ny_n'',$$

i.e.,

$$(y_1' - y_2') + 2(y_2' - y_2'') + \cdots + n(y_n' - y_n'') = 0. \quad (3)$$

But (2) and (3) contradict each other. Thus angles 1 and 2 do not contain any digital points, and L' and L'' represent the same digital line segment. This completes the proof.

3. Digital line segment construction from least square line fit

In this section we describe an efficient recovery procedure for digital line segments if they are represented by their endpoints (without loss of generality, we may assume the endpoints are 0 and $n-1$) and the coefficients b_0 and b_1 of the corresponding least square line fits.

First we describe a procedure line (n, α, A, L) which, for a given interval $[0, n-1]$, slope α and sum A finds a straight line L with slope α such that $A = y_1 + \cdots + y_n$, where $y_1, \ldots, y_n$ are the y-coordinates of the digital line obtained by digitizing L.

Procedure line(n, α, A, L)

Input. Interval $[0, n-1]$, slope α, and sum A.

Output. Straight line L having slope α and the sum A of the y-coordinates of the digital line corresponding to L on the interval $[0, n-1]$.

Step 1. Draw a line segment p with slope α such that p contains $(0,0)$.

Step 2. Find the number of digital points $s(p)$ bounded by the triangle determined by p, the x-axis and vertical line $x = n-1$ ($s(p) = y_1 + \cdots + y_n$ for the line p).

Step 3. Translate p by $\lfloor (A - s(p))/n \rfloor$ in the vertical direction (upward if $A - s(p) > 0$ and downward otherwise) to a new position p'. Obviously $s(p') = s(p) + n\lfloor (A - s(p))/n \rfloor$ and $|s(p') - s(p)| < n$. If $s(p) - s(p') > 0$ then p' should be translated upwards by at most 1; otherwise it should be translated downwards by at most 1.

Step 4. Let $(i-1, y_i)$ be a digital point of p'. Find the distances d_i from $(i-1, y_i+1)$ if the translation goes upwards, and to $(i-1, y_i-1)$ otherwise.

Step 5. Find the $(s(p) - s(p'))$th smallest distance among the distances d_i. Any algorithm that finds the kth median of a set can be used (for example, Blum et al. (1973) which is a linear time algorithm).

Step 6. Translate p' to its final position L by the $(s(p) - s(p'))$th smallest distance, in the appropriate vertical direction.

The above procedure clearly runs in linear time.

Theorem 2. *A digital line segment can be recovered from its given code* (x_1, n, b_0, b_1) *in time* $O(n \log n)$.

Proof. Suppose, for simplicity, that $x_1 = 0$, $x_n = n-1$ and $x_i = i-1$ for $1 < i < n$. From

$$b_0 = \frac{y_1 + \cdots + y_n}{n} - b_1 \frac{x_1 + \cdots + x_n}{n}$$

it follows that

$$y_1 + \cdots + y_n = n(b_0 + n(n-1)b_1/2) = A. \quad (4)$$

In other words, the sum of the ordinates of all digital points on the required digital line segment is fixed.

From

$$b_1 = \frac{12}{n^3 - n}(y_1, y_2, \ldots, y_n) \cdot \left(-\frac{n-1}{2}, -\frac{n-1}{2} + 1, \ldots, \frac{n-1}{2} \right)$$

we obtain

$$\frac{b_1(n^3 - n)}{12} = -\frac{n+1}{2}(y_1 + \cdots + y_n) + y_1 + 2y_2 + \cdots + ny_n,$$

i.e.

$$y_1 + 2y_2 + \cdots + ny_n = \frac{b_1(n^3 - n)}{12} + \frac{n+1}{2} n\left(b_0 + \frac{n(n-1)b_1}{2}\right) = B. \quad (5)$$

Therefore, the sum $y_1 + 2y_2 + \cdots + ny_n$ is also fixed ($= B$). The fixed sums (4) and (5) form the basis of our algorithm.

The algorithm consists of several iterations. In each iteration we construct a straight line L with slope α such that (4) is satisfied. This is done by the procedure line(n, α, A, L) where $A = n(b_0 + n(n-1) b_1/2)$. Once L is constructed, find $\mathrm{sum}(L) = y_1 + 2y_2 + \cdots + ny_n$ for the digital line segment defined by L and compare it with the right side of (5), i.e., with B.

Consider Figure 1. Let L' and L'' be lines (corresponding to two different digital line segments) obtained by calling line(n, α', A, L') and line(n, α'', A, L''), respectively. The numbers of digital points in angle 1 and angle 2 are the same because the sums $y_1 + \cdots + y_n$ are the same for both L' and L''. Now compare the sums $y_1 + 2y_2 + \cdots + ny_n$ for L' and L''. Inequality (2) yields

$$\mathrm{sum}(L') = y_1' + \cdots + ny_n' < y_1'' + \cdots + ny_n'' = \mathrm{sum}(L''),$$

which implies that if $\alpha' < \alpha''$ then $\mathrm{sum}(L') < \mathrm{sum}(L'')$ (when L' and L'' represent different digital line segments; otherwise $\mathrm{sum}(L') = \mathrm{sum}(L'')$). This property permits a binary search over slopes α to find a line L with slope α for which $\mathrm{sum}(L)$ is equal to B.

The slopes are initially within the interval $[0, 1]$. Binary search for the desired slope will, in each iteration, reduce the size of the remaining interval to half of the original size. If an accuracy of, say, 10^{-K} (for the slope of the digital line) is desired then the number of iterations m that is necessary to achieve this is given by $2^{-m} < 10^{-K}$ which is satisfied for $m > K \log_2 10$. In other words, $M = O(K)$ iterations suffice to find the line with a given accuracy in its slope. Since each iteration requires linear time, the algorithm requires $O(nK)$ time overall, which is linear if the number K of output bits of accuracy is a constant (which is an usual assumption). For example, if we desire accuracy of 10^{-3} in the slope of the line segment that generates the digital line segment then we require the first $K = 4$ accurate digits of the slope (between 0 and 1). However, the digital line obtained is not necessarily the desired solution; the sum $y_1 + 2y_2 + \cdots + ny_n$ for this line does not need to be equal to B, i.e., the least square fit line for the solution obtained may not coincide with the given least square fit line.

Therefore we perform a binary search until the line obtained satisfies the second criterion $y_1 + 2y_2 + \cdots + ny_n = B$, which is sufficient to ensure a line with the given least square line fit. How many iterations are necessary?

It is well known that the minimal difference between the slopes of two different lines, each passing through two integer points of a grid of size $n \times n$, is at least $1/n^2$ (these slopes are rational numbers, say $a/b \neq c/d$ and

$$\left|\frac{a}{b} - \frac{c}{d}\right| = \left|\frac{ad - bc}{bd}\right| \geqslant \frac{1}{bd} \geqslant \frac{1}{n^2}\Bigg).$$

For the desired digital line segment there exists a corresponding interval containing all possible slopes of lines which share the same digital line segment. This interval has length $\geqslant 1/n^2$.

Our binary search procedure starts with the interval $[0, 1]$ and will certainly terminate when the interval remaining to be searched is $\leqslant 1/n^2$. The number of iterations needed to obtain an interval that is n^2 times smaller than the original one is $\log n^2 = 2 \log n$. This is the maximal possible number of iterations that may be needed. Therefore the time complexity is $O(n \log n)$. □

This algorithm is sumarized as follows:

Algorithm decode(b_0, b_1, n, L)

Input. b_0 and b_1, coefficients of the least square fit line $Y = b_0 + b_1 X$ and n, the size of interval.

Output. A straight line L for which the corresponding digital line segment on the interval $[0, n-1]$ has least square fit line $Y = b_0 + b_1 X$.

```
Choose a slope $\alpha$ from $[0, n-1]$.
    $A := n(b_0 + n(n-1)b_1/2)$;
    $B := \frac{b_1(n^3 - n)}{12} + \frac{n+1}{2} n\left(b_0 + \frac{n(n-1)b_1}{2}\right)$;
line$(n, \alpha, A, L)$;
while $\mathrm{sum}(L) \neq B$ do
    {choose the next $\alpha$ by using a binary search
    strategy; line$(n, \alpha, A, L)$;}
```

Conclusions

We have shown that digital line segments and their least square line fits are in one-to-one correspondence. A new simple representation by four numbers is given; the numbers are the endpoints of the digital line and coefficients of its least square fit line. An $O(n \log n)$ time procedure to find a digital line from its least square line fit is described. In practice, since slopes are represented with a certain accuracy, the algorithm is linear (more precisely, $O(nK)$, where slopes are represented with K digits of accuracy).

Several open problems remain for further research. One could apply similar representations to other types of digital curves (circles, ellipses etc.) It would also be of interest to investigate representations by other types of line fits (for example, by the line that minimizes the total sum of distances to it).

Another problem is to find an exact linear time 'decoding' algorithm (rather than $O(n \log n)$). It is clearly possible if the least square fit line is parallel to one of the coordinate axes, or at least is sufficiently 'close' to the interval of possible slopes of the generators of the corresponding digital line segment. It is not known whether any of these conjectures, which seem to be intuitively correct, holds for an arbitrary digital line.

It would be also interesting to apply the new representation to real-world images. For example, suppose we wish to represent a part of such a real-world image by a line segment (call this segment of the image a 'scatter line'). A possible prescription for representation of this line would be:

(1) compute a continuous approximation to the scatter line by least squares;

(2) compute a digital line by rounding;

(3) do a least squares to this digital line to find the 4-parameter representation.

Then the digital line (but, of course, not the 'scatter line') can be reconstructed from the 4-parameter representation. The slope/intersect parameters of the second least squares fit are generally different from those of the first, and it would be of some interest to estimate the magnitude of these errors.

A related question is to investigate the range of the 4-number parameter vectors, since an arbitrary vector cannot be expected to represent some digital line.

References

Burr, I.W. (1974). *Applied Statistical Methods.* Academic Press, New York.

Blum, M., R.W. Floyd, V. Pratt, R.L. Rivest and R.E. Tarjan (1973). Time bounds for selection. *J. Comput. System Sci.* 7 (4) 448-461.

Dorst, L. and A.W.M. Smeulders (1984). Discrete representation of straight lines. *IEEE Trans. Pattern Anal. Machine Intell.* 6, 450-463.

Kim, C.E. (1983). Three-dimensional digital line segments, *IEEE Trans. Pattern Anal. Machine Intell.* 5, 231-234.

Kim, C.E. and A. Rosenfeld (1982). Digital straight lines and convexity of digital regions, *IEEE Trans. Pattern Anal. Machine Intell.* 4, 149-153.

Lindenbaum, M. and J. Koplowitz (1991). A new parametrization of digital straight lines. *IEEE Trans. Pattern Anal. Machine Intell.* 13 (4).

Melter, R.A. and A. Rosenfeld (1989). New views of linearity and connectedness in digital geometry. *Pattern Recognition Lett.* 10, 9-16.

Stojmenović, I. and R. Tosic (1991). Digitization schemes and the recognition of digital lines, hyperplanes, and flats in arbitrary dimensions. In: R.A. Melter, A. Rosenfeld, and P. Bhattacharya, Eds., *Vision Geometry*, Contemporary Math. 119. Amer. Math. Soc. Providence, RI, 197-212.