

Towards a Geometric Interpretation of Double-Cross Matrix-based Similarity of Polylines

Bart Kuijpers and Bart Moelans

Theoretical Computer Science
Hasselt University & Transnational University of Limburg, Belgium
{bart.kuijpers, bart.moelans}@uhasselt.be

ABSTRACT

One of the formalisms to qualitatively describe polylines in the plane are double-cross matrices. In a double-cross matrix the relative position of any two line segments in a polyline is described with respect to a double cross based on their start points. Two polylines are called DC-similar if their double-cross matrices are identical. Although double-cross matrices have been widely applied, a geometric interpretation of the similarity they express is still lacking. In this paper, we provide a first step in the geometric interpretation of this qualitative definition of similarity. In particular, we give an effective characterization of what DC-similarity means for polylines that are drawn on a grid. We also provide algorithms that, given a DC-matrix, check whether it is realizable by a polyline on a grid and that construct, if possible, in quadratic time example polylines that satisfy this matrix. We also describe algorithms to reconstruct polylines, satisfying a given double-cross matrix, in the two-dimensional plane, that is, not necessarily on a grid.

Categories and Subject Descriptors

F.2 [Analysis of algorithms and problem complexity]: Miscellaneous; H.2.8 [Database applications]: Datamining, Spatial databases and GIS

General Terms

Algorithms, Theory

Keywords

Double-cross calculus, polylines, similarity

1. INTRODUCTION AND SUMMARY

The *double-cross calculus* [10, 19] was introduced as a formalism to qualitatively describe configurations of several vectors in the plane. In this language, two vectors are encoded as a 4-tuple that expresses their relative orientation

with respect to each other. The double-cross formalism is used, for instance, in the *qualitative trajectory calculus* [6, 7]. In turn, the qualitative trajectory calculus has been used to test polyline similarity with applications to query-by-sketch, indexing and classification [13].

In this paper, we consider polylines in the two-dimensional plane and model them as finite sequences of vertices, or equivalently, as finite sequences of contiguous vectors. The *double-cross matrix (DC-matrix)* of a polyline contains for each pair of vectors in the polyline the 4-tuple that expresses their relative orientation. For a polyline consisting of N vectors, the DC-matrix has N^2 entries but only the $N(N-1)/2$ above the diagonal matter. It is well-known that not every N by N matrix of such 4-tuples is the DC-matrix of a polyline [6]. Two polylines with the same DC-matrix are called *DC-similar*. We are interested in a geometric interpretation of the concept of DC-similarity and the literature on this topic has not yet provided a satisfying answer to this difficult question. The main contribution of this paper are some first steps in the geometric interpretation of this qualitative definition of similarity. In each direction of our investigation, we limit the full difficulty of the problem by removing some degrees of freedom.

In a first direction, we give an effective characterization of what DC-similarity means for polylines that are drawn on a grid. For a polyline on a grid, we call the vertical and horizontal straight lines through its vertices its *vertical and horizontal carriers* and we call the order in which they appear as we go through the polyline from start to end the V- and H-order of the polyline. We call two polylines *VH-equivalent* if they have the same V-order and the same H-order.

To a polyline on a grid, we associate a *canonical polyline*, which is VH-equivalent to the original polyline and which has the same DC-matrix as the original polyline. In fact, the canonical polyline can be seen as the smallest realization of a DC-matrix. And it turns out that VH-equivalence is the notion that captures the geometric information contained in the DC-matrix: two polylines have the same DC-matrix if and only if, after alignment, they are VH-equivalent (their last vectors may differ in length, though). We also give an algorithm that on input a DC-matrix of size N by N , checks in $O(N^2)$ time whether it is realizable by a polyline on a grid. If so, it also produces an example polyline on an exponentially large grid in $O(N^2)$ time. From this example polyline, the V- and H-order can be derived and once these are found, many example polylines satisfying the given DC-matrix can be generated, among which the canonical polyline. They all

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM GIS '08, November 5-7, 2008, Irvine, CA, USA
Copyright 2008 ACM 978-1-60558-323-5/08/11 ...\$5.00.

differ in horizontal and vertical compressions and dilatations from the canonical polyline.

For polylines whose vertices are snapped to the grid, the above results can be improved on. Here, once the realizability of a DC-matrix has been checked (in $O(N^2)$ time again), it can be realized in $O(N)$ time.

In a second direction, we look at polylines in the two dimensional plane, not necessarily on a grid. A geometric characterization of DC-similarity is still failing here. As a first step in understanding DC-similarity, we look at the realization of DC-matrices. By algebraically interpreting the entries in a DC-matrix, algorithms from computational algebra, such as cylindrical algebraic decomposition [4], can produce in exponential time example polylines satisfying a given DC-matrix (and can also be used to verify their realizability). This approach is only practical for very small matrices. We propose a heuristic algorithm that produces, for a given DC-matrix, example polylines with vectors of equal length.

This paper is organized as follows. In Section 2, we define polylines and the double-cross formalism. In Section 3, polylines on grids are studied. Section 4 addresses the reconstruction of polylines in the plane.

Related work. In recent years, the growing number of location-aware devices has given rise to a increasing availability of mobility data. There is a big interest in the analysis of data produced by moving objects [11]. The traces left by moving objects are typically interpolated between measured points and we can roughly say that they are polylines in the plane [12, 18]. While trying to group or cluster moving objects, similarity measures between trajectories [16] have gained in importance. Also in other areas of spatio-temporal datamining there is an interest in similarity measures [14]. Although there are some authors that address the temporal dimension of trajectories [15, 17], the majority of the work only looks at the geometric shape of the trajectories [5, 6, 7, 11, 13, 19]. So, actually, we consider in this context, polylines with a walk-through direction. Most similarity measures are quantitative and they do not sufficiently support intuitive or common-sense oriented human-computer interaction [5]. Therefore there is increasing interest in *naive geography* [8] and qualitative representations. We also refer to [5, 6] and references therein.

2. DEFINITIONS AND PRELIMINARIES

In this section, we define polylines in the plane and explain the formalism of double-cross matrices and how it is used to measure similarity of polylines.

2.1 Polylines

Let \mathbb{R} denote the set of real numbers and \mathbb{R}^2 the real plane. We now define how a polyline, that is, a piecewise linear curve in \mathbb{R}^2 , is represented.

DEFINITION 1. A *polyline* P in \mathbb{R}^2 is given by the ordered list of coordinate pairs of its **vertices**, that is, $P = \langle (x_0, y_0), (x_1, y_1), \dots, (x_N, y_N) \rangle$. The vertices (x_0, y_0) and (x_N, y_N) are respectively called the **start** and **end vertex** of P . We assume that consecutive vertices are different. For $i = 1, \dots, N$, we denote the closed line segment between (x_{i-1}, y_{i-1}) and (x_i, y_i) by $L_i(P)$ and the the vector from (x_{i-1}, y_{i-1}) to (x_i, y_i) by $\vec{\ell}_i(P)$. We say that N is the **size** of P .

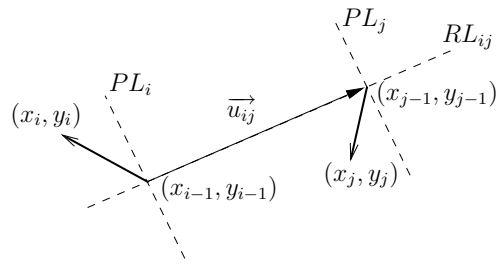


Figure 1: Double-cross design: the lines RL_{ij}, PL_i, PL_j .

The *semantics of the polyline* P is the subset of \mathbb{R}^2 consisting of all line segments between consecutive vertices of P , or $\text{sem}(P) := \bigcup_{1 \leq i \leq N} L_i(P)$. \square

If P is clear from the context, we will omit (P) from the above notations. Further, we remark that polylines may be self intersecting. It is reasonable to assume that polylines coming from GIS applications have vertices with rational coordinates.

Below, we often have to align the start vectors of two polylines. Let P and Q be two polylines and let α_{PQ} be the unique affinity of \mathbb{R}^2 that maps $\vec{\ell}_1(P)$ to $\vec{\ell}_1(Q)$. The transformation α_{PQ} can be decomposed into a translation τ_{PQ} that maps the start vertex of P to the start vertex of Q ; a rotation ρ_{PQ} that aligns $\tau_{PQ}(\vec{\ell}_1(P))$ with $\vec{\ell}_1(Q)$; and finally a point-scaling (or homotety) σ_{PQ} that maps $\rho_{PQ}(\tau_{PQ}(\vec{\ell}_1(P)))$ onto $\vec{\ell}_1(Q)$.

2.2 Double-Cross Matrices

The double-cross calculus [10, 19] was introduced as a formalism to qualitatively represent a configuration of several vectors in the plane. Specifically, in this formalism, two vectors are encoded by means of a 4-tuple that expresses the relative orientation of both vectors with respect to each other. Above, we have associated to a polyline P the vectors $\vec{\ell}_1, \dots, \vec{\ell}_N$, representing oriented line segments between consecutive vertices. We use the double-cross formalism to qualitatively present the orientation between $\vec{\ell}_i$ and $\vec{\ell}_j$ by means of a 4-tuple

$$DC(\vec{\ell}_i, \vec{\ell}_j) = (C_1 \ C_2 \ C_3 \ C_4) \in \{-, 0, +\}^4.$$

To determine C_1, C_2, C_3 and C_4 , first of all, we define a *double cross* for the vectors $\vec{\ell}_i$ and $\vec{\ell}_j$, determined by three lines: the *reference line* RL_{ij} connecting (x_{i-1}, y_{i-1}) and (x_{j-1}, y_{j-1}) ; the *perpendicular lines* PL_i on RL_{ij} through (x_{i-1}, y_{i-1}) ; and PL_j on RL_{ij} through (x_{j-1}, y_{j-1}) . We refer to the vector between (x_{i-1}, y_{i-1}) and (x_j, y_j) as \vec{u}_{ij} . The lines RL_{ij}, PL_i, PL_j and the vector \vec{u}_{ij} are illustrated in Figure 1. The entries C_1, C_2, C_3 and C_4 express how $\vec{\ell}_i$ and $\vec{\ell}_j$ are located towards RL_{ij}, PL_i and PL_j . We now define this more formally [6, 7, 10, 19].

DEFINITION 2. For $\vec{\ell}_i, \vec{\ell}_j$ with $(x_{i-1}, y_{i-1}) \neq (x_{j-1}, y_{j-1})$, $DC(\vec{\ell}_i, \vec{\ell}_j) = (C_1 \ C_2 \ C_3 \ C_4)$ is defined as follows:

- $C_1 = -$ iff (x_i, y_i) lies on the same side of PL_i as (x_{j-1}, y_{j-1}) and $(x_i, y_i) \notin PL_i$;
 $C_1 = 0$ iff $(x_i, y_i) \in PL_i$;
 $C_1 = +$ iff else;
 $C_2 = -$ iff (x_j, y_j) lies on the same side of PL_j as (x_{i-1}, y_{i-1}) and $(x_j, y_j) \notin PL_j$;
 $C_2 = 0$ iff $(x_j, y_j) \in PL_j$;
 $C_2 = +$ iff else;
 $C_3 = -$ iff (x_i, y_i) lies on the left of \vec{u}_{ij} ;
 $C_3 = 0$ iff $(x_i, y_i) \in RL_{ij}$;
 $C_3 = +$ iff else;
 $C_4 = -$ iff (x_j, y_j) lies on the right of \vec{u}_{ij} ;
 $C_4 = 0$ iff $(x_j, y_j) \in RL_{ij}$; and
 $C_4 = +$ iff else.

For $\vec{\ell}_i, \vec{\ell}_j$ with $(x_{i-1}, y_{i-1}) = (x_{j-1}, y_{j-1})$, we define, for reasons of continuity [9], $DC(\vec{\ell}_i, \vec{\ell}_j) = (0 \ 0 \ 0 \ 0)$. \square

For example, the 4-tuple $DC(\vec{\ell}_i, \vec{\ell}_j)$ for the vectors $\vec{\ell}_i$ and $\vec{\ell}_j$, shown in Figure 1, is $(+ \ - \ - \ -)$.

We repeat the following property from [13], because it gives the algebraic expression to calculate the value $DC(\vec{\ell}_i, \vec{\ell}_j)$, which will be used extensively to prove some properties further on. By $\text{sign} : \mathbb{R} \rightarrow \{-, 0, +\}$ we denote the function that maps strictly negative numbers to $-$, 0 to 0 and strictly positive numbers to $+$.

PROPERTY 1. Let $\vec{\ell}_i$ and $\vec{\ell}_j$ have coordinates as described in Figure 1. Then we have

$$\begin{aligned}
C_1 &= -\text{sign}((x_{j-1} - x_{i-1}) \cdot (x_i - x_{i-1}) \\
&\quad + (y_{j-1} - y_{i-1}) \cdot (y_i - y_{i-1})); \\
C_2 &= \text{sign}((x_{j-1} - x_{i-1}) \cdot (x_j - x_{j-1}) \\
&\quad + (y_{j-1} - y_{i-1}) \cdot (y_j - y_{j-1})); \\
C_3 &= -\text{sign}((x_{j-1} - x_{i-1}) \cdot (y_i - y_{i-1}) \\
&\quad - (y_{j-1} - y_{i-1}) \cdot (x_i - x_{i-1})); \\
\text{and} \\
C_4 &= \text{sign}((x_{j-1} - x_{i-1}) \cdot (y_j - y_{j-1}) \\
&\quad - (y_{j-1} - y_{i-1}) \cdot (x_j - x_{j-1})). \quad \square
\end{aligned}$$

We remark that for any $\vec{\ell}_i$ and $\vec{\ell}_j$, arbitrary changing their lengths, does not influence the value of $DC(\vec{\ell}_i, \vec{\ell}_j)$.

DEFINITION 3. A **double-cross matrix (DC-matrix)** of a polyline $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$, denoted $DCM(P)$, is a $N \times N$ matrix with $DCM(P)[i, j] = DC(\vec{\ell}_i, \vec{\ell}_j)$. Two polylines P and Q are called **DC-similar** if $DCM(P) = DCM(Q)$. \square

We remark that it suffices to consider only the upper triangle of the matrix $DCM(P)$, i.e., the $(N^2 - N)/2$ elements $DC(\vec{\ell}_i, \vec{\ell}_j)$, with $i < j$. The other elements do not give any extra information [13].

We also remark that for a polyline $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$, changing the length of the last vector $\vec{\ell}_N$ will not change $DCM(P)$. Changing the length of any other vector can change $DCM(P)$, as can be seen in Figure 2. Indeed, increasing the length of $\vec{\ell}_{j-1}$ changes $DC(\vec{\ell}_i, \vec{\ell}_j)$.

The double-cross matrix is invariant under a number of transformations of the plane [13].

PROPERTY 2. Let P be a polyline and let α be a translation, an isometry or a point-scaling (homotecy) of \mathbb{R}^2 , then $DCM(P) = DCM(\alpha(P))$. \square

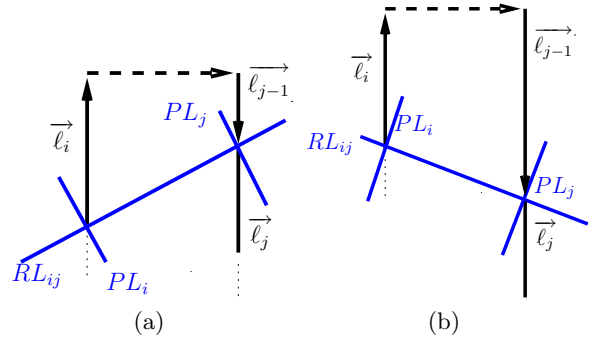


Figure 2: An example of the influence on the DC-matrix of a polyline on a grid when changing the length of one of its vectors.

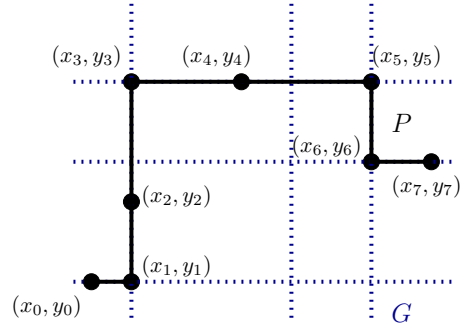


Figure 3: An example of a polyline on a grid.

If we compare vectors only with the next k vectors in a polyline, we get the following definition.

DEFINITION 4. The **k -partial double-cross matrix** of P is denoted and defined as $DCM^k(P) = \{DCM(P)[i, j] \mid i < j \leq \max(i+k, N)\}$. Two polylines P and Q are called **k -similar** if $DCM^k(P) = DCM^k(Q)$. \square

Clearly, for polylines consisting of N line segments, DC-similarity and N -similarity coincide.

3. POLYLINES ON A GRID

We study polylines that are on a grid. First, we define some types of grid. Let \mathbb{Z} denote the set of the integers.

DEFINITION 5. The **complete infinite grid** is defined to be the set $(\mathbb{Z} \times \mathbb{R}) \cup (\mathbb{R} \times \mathbb{Z})$. We call the elements of $\mathbb{Z} \times \mathbb{Z}$ its **crossings**. A **grid** is a subset of the complete infinite grid of the form $(A \times \mathbb{R}) \cup (\mathbb{R} \times B)$, with A, B finite subsets of \mathbb{Z} .

A **polyline P is on a grid G** , if $\text{sem}(P) \subset G$. \square

In Figure 3, a grid G and polyline P on G are shown. In this example, not all vertices of P are located on crossings of G . For the case where all the vertices are crossings, we have the following definition.

DEFINITION 6. Let G be a grid and let $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ be a polyline on G . We say that P is **snapped to G** if all vertices of P are crossings of G . Furthermore, if all crossings of G that belong to $\text{sem}(P)$ are vertices of P , we say that P is **completely snapped to G** . \square

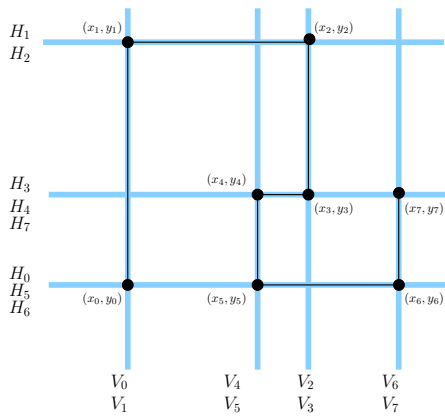


Figure 4: A polyline and its horizontal and vertical carriers.

We remark that (completely) snapped polylines have vertices with integer-valued coordinates. But, since we assume rational coordinates, we can always find a suitable point-scaling that maps an arbitrary polyline on a grid to a snapped polyline on a sufficiently large grid, such that the resulting polyline has vertices with integer-valued coordinates. We refer to Property 7 for details.

Later on, for an arbitrary polyline P on a grid, we define a canonical polyline $\text{can}(P)$, which has the same DC-matrix as P , that is snapped to the complete infinite grid and that is minimal in some sense. To prove the existence of $\text{can}(P)$, we will first introduce some definitions.

DEFINITION 7. Given a polyline $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ on a grid, the lines V_i , given by the equation $x = x_i$, and H_i , given by the equation $y = y_i$ ($i = 0, \dots, N$), are called the **vertical**, respectively **horizontal carriers** of P . \square

Figure 4 shows a polyline $P = \langle (x_0, y_0), \dots, (x_7, y_7) \rangle$ on a grid (not shown) and its vertical and horizontal carriers. We remark that some of these carriers may coincide.

We now define two lists of lists that capture the order of the carriers of a polyline on a grid.

DEFINITION 8. Let $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ be a polyline on a grid. The **V-order** of P , denoted as $V(P)$, is a list (A_1, \dots, A_K) of lists $A_i = (a_{i1}, \dots, a_{ik_i})$ such that each a_{ij} is an element of $\{0, 1, \dots, N\}$ and appears exactly once in $V(P)$. Within each A_i , the elements appear in increasing order, and for $a, a' \in A_i$, we have $x_a = x_{a'}$. For $a \in A_i$ and $a' \in A_j$, with $i < j$, we have $x_a < x_{a'}$.

The **H-order** of P , denoted as $H(P)$, is a list (B_1, \dots, B_L) of lists $B_i = (b_{i1}, \dots, b_{il_i})$ such that each b_{ij} is an element of $\{0, 1, \dots, N\}$ and appears exactly once in $H(P)$. Within each B_i , the elements appear in increasing order, and for $b, b' \in B_i$, we have $y_b = y_{b'}$. For $b \in B_i$ and $b' \in B_j$, with $i < j$, we have $y_b < y_{b'}$. \square

For the polyline P in Figure 4, $V(P) = ((0, 1), (4, 5), (2, 3), (6, 7))$ and $H(P) = ((0, 5, 6), (3, 4, 7), (1, 2))$. We remark that $V(P)$ and $H(P)$ are invariant under translations and scalings but not under rotations of the plane \mathbb{R}^2 . For instance, if ρ is a rotation over 180° , then $V(\rho(P)) = ((6, 7), (2, 3), (4, 5), (0, 1))$ and $H(\rho(P)) = ((1, 2), (3, 4, 7), (0, 5, 6))$.

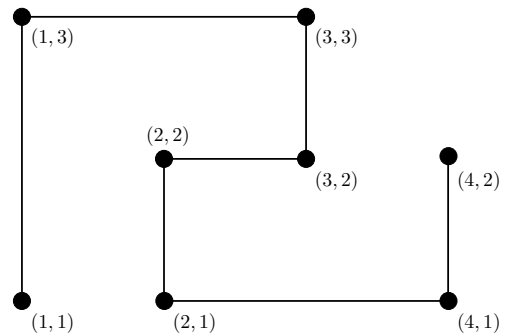


Figure 5: The canonical polyline of the polyline of Figure 4.

The following property follows immediately from the definition.

PROPERTY 3. Let $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ be a polyline on a grid. Given only $V(P)$, we can decide whether $x_i < x_j$, $x_i = x_j$ or $x_j < x_i$ for any $1 \leq i < j \leq N$. Given only $H(P)$, we can decide whether $y_i < y_j$, $y_i = y_j$ or $y_j < y_i$ for any $1 \leq i < j \leq N$. \square

We now define equivalence of polylines on a grid in terms of V - and H -order.

DEFINITION 9. Let P and Q be two polylines of the same size on a grid. We say that P and Q are **VH-equivalent**, denoted $P \equiv_{VH} Q$, if $V(P) = V(Q)$ and $H(P) = H(Q)$. \square

Given a polyline on a grid, we now associate a *canonical polyline* on the complete infinite grid to it.

DEFINITION 10. Given a polyline $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ on a grid, with $V(P) = (A_1, \dots, A_K)$ and $H(P) = (B_1, \dots, B_L)$, we define the **canonical polyline** of P , denoted $\text{can}(P)$, to be the polyline $\langle (x'_0, y'_0), \dots, (x'_N, y'_N) \rangle$, that is snapped to the complete infinite grid, such that for all $i = 0, \dots, N$, we have $i \in A_{x'_i}$ and $i \in B_{y'_i}$. \square

For the polyline P of Figure 4, $\text{can}(P) = \langle (1, 1), (1, 3), (3, 3), (3, 2), (2, 2), (2, 1), (4, 1), (4, 2) \rangle$ is shown in Figure 5. We remark that if P is a polyline on a grid, its canonical polyline, $\text{can}(P)$ is a snapped polyline on the complete infinite grid. In fact, $\text{can}(P)$ can be viewed as a polyline on the “small” grid $(\{1, \dots, K\} \times \mathbb{R}) \cup (\mathbb{R} \times \{1, \dots, L\})$.

The following property is straightforward.

PROPERTY 4. Let P and Q be of the same size on a grid. We have $P \equiv_{VH} Q$ if and only if $\text{can}(P) = \text{can}(Q)$. \square

3.1 Properties of DC-matrices of polylines on a grid

In this section, we discuss some properties of the DC-matrix of polylines on a grid. First, we remark that only 33 4-tuples of $-$, 0 and $+$ can appear in the DC-matrix of a polyline on a grid, namely, those of the form $(C_1 C_2 C_3 C_4)$, $(C_1 C_2 0 0)$, $(C_1 0 0 C_4)$, $(0 0 C_3 C_4)$, $(0 C_2 C_3 0)$ and $(0 0 0 0)$, with $C_1, C_2, C_3, C_4 \in \{+, -\}$.

PROPERTY 5. Let $k \geq 1$. For polylines on a grid, the $(k + 1)$ -partial double-cross matrix $DCM^{k+1}(P)$ cannot be derived from the k -partial double-cross matrix $DCM^k(P)$.

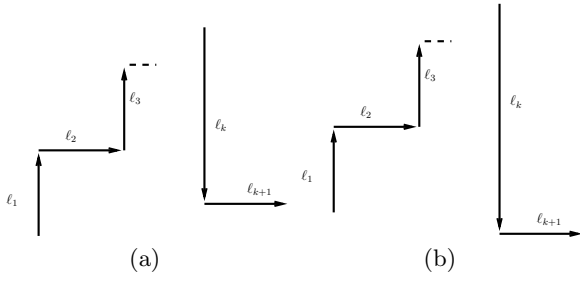


Figure 6: Example polylines for the proof of Prop. 5.

PROOF. The polylines in (a) and (b) of Figure 6 have the same DCM^k but they do not have the same DCM^{k+1} . A polyline like in Figure 6 can be generated for each $k \geq 2$. For the case $k = 1$, just consider $P_1 = \langle (0, 0), (0, 2), (0, 1), (1, 1) \rangle$ and $P_2 = \langle (0, 0), (0, 2), (0, -1), (1, -1) \rangle$. We have $DCM^1(P_1) = DCM^1(P_2)$ but $DCM^2(P_1) \neq DCM^2(P_2)$. Since polyline on a grid are a subset of polylines on the real plane, this also proves the general case. \square

This property implies that, given a DC-matrix M , we need to look at all cells (above the diagonal) of M to be able to reconstruct a polyline P such that $DCM(P) = M$.

PROPERTY 6. *If P is a polyline on a grid, then $DCM(P) = DCM(\text{can}(P))$.*

PROOF. Let $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ and $\text{can}(P) = \langle (x'_0, y'_0), \dots, (x'_N, y'_N) \rangle$. Suppose $DCM(P)[i, j] = DC(\vec{\ell}_i, \vec{\ell}_j) = (C_1 \ C_2 \ C_3 \ C_4)$ and $DCM(\text{can}(P))[i, j] = (D_1 \ D_2 \ D_3 \ D_4)$, with $i, j \in \{1, \dots, N\}$. We have to show that $C_\ell = D_\ell$, for $\ell = 1, 2, 3, 4$. If we look at the algebraic expression for C_1, C_2, C_3 and C_4 in Property 1, we can see they are all of the form $\pm \text{sign}((t_{11} - t_{12}) \cdot (t_{21} - t_{22}) \pm (t_{31} - t_{32}) \cdot (t_{41} - t_{42}))$ and that t_{k1} and t_{k2} are x -coordinates or y -coordinates of vertices, with $k \in \{1, 2, 3, 4\}$. Let us focus on C_1 and D_1 : $C_1 = -\text{sign}((x_{j-1} - x_{i-1}) \cdot (x_i - x_{i-1}) - (y_{j-1} - y_{i-1}) \cdot (y_i - y_{i-1}))$ and $D_1 = -\text{sign}((x'_{j-1} - x'_{i-1}) \cdot (x'_i - x'_{i-1}) - (y'_{j-1} - y'_{i-1}) \cdot (y'_i - y'_{i-1}))$. Since P is a polyline on a grid, we have that $(x_i = x_{i-1} \text{ and } y_i \neq y_{i-1})$ or $(x_i \neq x_{i-1} \text{ and } y_i = y_{i-1})$. Assume $x_i = x_{i-1}$ and $y_i \neq y_{i-1}$ (the other case is similar). By definition of $V(P)$ and $\text{can}(P)$ we know when $x_i = x_{i-1}$ then $x'_i = x'_{i-1}$, thus we can rewrite the expression for C_1 and D_1 as follows: $C_1 = \text{sign}((y_{j-1} - y_{i-1}) \cdot (y_i - y_{i-1}))$ and $D_1 = \text{sign}((y'_{j-1} - y'_{i-1}) \cdot (y'_i - y'_{i-1}))$. By definition of $\text{can}(P)$, we can see that if $y_i < y_j$, then also $y'_i < y'_j$ and thus $C_1 = D_1$. We can show in a similar way that $C_2 = D_2$, $C_3 = D_3$ and $C_4 = D_4$. \square

In the next theorem, we use the following notation. If $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ is a polyline, then we denote by P_i the polyline $\langle (x_0, y_0), \dots, (x_i, y_i) \rangle$ ($0 \leq i \leq N$). So, in particular $P = P_N$.

THEOREM 1. *Let P and Q be two polylines of size N on a grid. Then P and Q are DC-similar if and only if*

$$\alpha_{PQ}(P_{N-1}) \equiv_{VH} Q_{N-1}$$

and the last vector of $\alpha_{PQ}(P)$ and Q have the same direction with respect to their one but last vector, i.e., $DC(\vec{\ell}_{N-1}(P), \vec{\ell}_N(P)) = DC(\vec{\ell}_{N-1}(Q), \vec{\ell}_N(Q))$.

PROOF. From Property 6, we know that $DCM(P) = DCM(Q)$ if and only if $DCM(\text{can}(P)) = DCM(\text{can}(Q))$ and from Property 4 that $\alpha_{PQ}(P_{N-1}) \equiv_{VH} Q_{N-1}$ if and only if $\text{can}(\alpha_{PQ}(P_{N-1})) = \text{can}(Q_{N-1})$. Therefore, it is enough to prove that $DCM(\text{can}(P)) = DCM(\text{can}(Q))$ if and only if $\text{can}(\alpha_{PQ}(P_{N-1})) = \text{can}(Q_{N-1})$ and the last vector of $\text{can}(\alpha_{PQ}(P_N))$ and $\text{can}(Q_N)$ have the same direction with respect to their one but last vector. Since $\text{can}(\alpha_{PQ}(P_{N-1})) = \text{can}(Q_{N-1}) = \alpha_{\text{can}(P)\text{can}(Q)}(\text{can}(P_{N-1}))$ (with $\alpha_{\text{can}(P)\text{can}(Q)}$ basically a rotation over 0, 90, 180, or 270 degrees), it suffices to prove the theorem, with VH-equivalence replaced by equality, for canonical polylines. So, from now, we simplify the notation and assume that P and Q are canonical.

For $N = 2$, we have to show that $DCM(\text{can}(P_2)) = DCM(\text{can}(Q_2))$ if and only if $\text{can}(\alpha_{PQ}(P_1)) = \text{can}(Q_1)$ and the last vector of $\text{can}(\alpha_{PQ}(P_2))$ and $\text{can}(Q_2)$ have the same direction with respect to their one but last vector (in this case the first vector). This is trivial since α_{PQ} maps the first segment of P , which is given by P_1 , on the first segment of Q which is given by Q_1 and since the direction of the second segment with respect to the first is uniquely determined by the right upper element in the DC-matrix. Let us now assume $N > 2$.

For the if direction, let us assume that $Q_{N-1} = \alpha_{PQ}(P_{N-1})$ and $DC(\vec{\ell}_{N-1}(P), \vec{\ell}_N(P)) = DC(\vec{\ell}_{N-1}(Q), \vec{\ell}_N(Q))$. The latter implies that $DCM(P)[N-1, N] = DCM(Q)[N-1, N]$. The former certainly implies that $DCM(P)[1..N-1, 1..N-1] = DCM(Q)[1..N-1, 1..N-1]$. But since the end vertex of Q_{N-1} is equal to the end vertex of $\alpha_{PQ}(P_{N-1})$, we also have $DCM(P)[i, N] = DCM(Q)[i, N]$ for $i = 1, \dots, N-2$.

For the only-if direction, let us assume that $DCM(P) = DCM(Q)$. From this assumption $DCM(P)[N-1, N] = DCM(Q)[N-1, N]$ follows and therefore the last vector of $\alpha_{PQ}(P)$ and Q have the same direction with respect to their one but last vector. Let $P = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$. For each pair of vertices (x_i, y_i) and (x_j, y_j) , with $1 \leq i < j < N$, it can be determined, from $DCM(P)$, whether $x_i < x_j$, $x_i = x_j$ or $x_i > x_j$ and also $y_i < y_j$, $y_i = y_j$ or $y_i > y_j$. Indeed, suppose $DCM(P)[i, j] = (C_1 \ C_2 \ C_3 \ C_4)$. By looking at Property 1 one can see that if $C_3 = 0$ that (x_{j-1}, y_{j-1}) is located on $\vec{\ell}_i(P)$. If $C_3 = -$, (x_{j-1}, y_{j-1}) is located right of $\vec{\ell}_i(P)$, otherwise (x_{j-1}, y_{j-1}) is located left of $\vec{\ell}_i(P)$. In other words, the V - and H -order of P_{N-1} can be derived from $DCM(P)$. Since $DCM(P) = DCM(Q)$, these canonical polylines of P_{N-1} and Q_{N-1} are therefore equal. \square

PROPERTY 7. *If P is a polyline on a grid G and $G' = (A \times \mathbb{R}) \cup (\mathbb{R} \times B)$ is a grid with $|A| \geq |V(P)|$ and $|B| \geq |H(P)|$, then there exists a snapped polyline Q on G' such that $DCM(P) = DCM(Q)$.*

PROOF. We construct Q on G' as we constructed the canonical polyline of P on the complete infinite grid (see Definition 10). Then by construction $P \equiv_{VH} Q$ and the last vector from P and Q have the same direction with respect to the one but last vector. By Theorem 1, it follows that $DCM(P) = DCM(Q)$. \square

Using the previous properties, we can see that two DC-similar polylines differ in horizontal and vertical compressions and dilatations from the canonical polyline (apart from their last vector, whose length is arbitrary).

snapped polyline generated using the algorithm `Reconstruct_DCM` with as input the DC-matrix in Table 1.

3.3 Properties of DC matrices of completely snapped polylines

Now, we prove a property of the DC-matrix of a completely snapped polyline, that does not hold for arbitrary polylines on a grid.

PROPERTY 9. *Given the 1-partial double-cross matrix $M = DCM^1(P)$ of a completely snapped polyline P , we can reconstruct $DCM(P)$.*

PROOF. Suppose $M = DCM^1(P)$ has $N - 2$ elements, that is, P has N vertices. Let $[-N, N]$ denote the set of integers $\{-N, -N + 1, \dots, 0, 1, \dots, N\}$. Take $G = ([-N, N] \times \mathbb{R}) \cup (\mathbb{R} \times [-N, N])$ as grid. We now show how we can reconstruct a polyline Q that satisfies M . We start with setting $Q = \langle (0, 0), (0, 1) \rangle$. Then Q is a completely snapped polyline and G is big enough to reconstruct any polyline with N line segments that has as start vertex $(0, 0)$. If we want to add a vertex $(x_2, y_2) \in G$ to Q such that Q is still a completely snapped polyline and $M[1, 2] = DC(\overrightarrow{\ell_1(Q)}, \overrightarrow{\ell_2(Q)})$, we only have one possibility for the position of (x_2, y_2) . Based on the fact that Q has to be a completely snapped polyline and $(x_2, y_2) \in G$ there are only four possible coordinates for (x_2, y_2) . The condition $M[1, 2] = DC(\overrightarrow{\ell_1(Q)}, \overrightarrow{\ell_2(Q)})$ will only be true for one of these four points and so we have a unique solution. For the same reasons we can see that if we want to extend Q to N line segments such that $M[i, i + 1] = DC(\overrightarrow{\ell_i(Q)}, \overrightarrow{\ell_{i+1}(Q)})$ for all $1 \leq i \leq N - 1$, all (x_j, y_j) , for $2 \leq j \leq N$ are uniquely predefined by $\overrightarrow{\ell_1(Q)}$, G and M . Once we have obtained $Q = \langle (x_0, y_0), \dots, (x_N, y_N) \rangle$ this way, we can calculate $DCM(Q)$. From Property 7 we know that the choice of G has no influence on $DCM(Q)$. Because a DC-matrix is invariant under a composition of a rotation, translation and point-scaling, taking another begin line segment for Q would also not change $DCM(Q)$ therefore for every Q constructed as described above $DCM(Q) = DCM(P)$. \square

The last proof actually describes an algorithm that given a (partial) DC-matrix M of a completely snapped polyline, generates a completely snapped polyline P with $DCM(P) = M$. The following corollary tells something about the time complexity.

COROLLARY 2. *Given a DC-matrix M of size $N \times N$ of a completely snapped polyline, a completely snapped polyline P of size N with $M = DCM(P)$ can be generated in $O(N)$ time. Checking whether a DC-matrix of size N by N , can be of a completely snapped polyline takes $O(N^2)$ time. \square*

4. RECONSTRUCT POLYLINES WITH EQUAL LENGTH LINE SEGMENTS IN THE TWO-DIMENSIONAL PLANE

From the previous section, we have gained geometric insight into what DC-similarity means for polylines on a grid. Although there are situations where traces of moving object are polylines on a grid (e.g., traffic in Manhattan), there are even more situations where traces are in the two-dimensional plane (e.g., football players on a field). Here, a geometric characterization of DC-similarity is still failing.

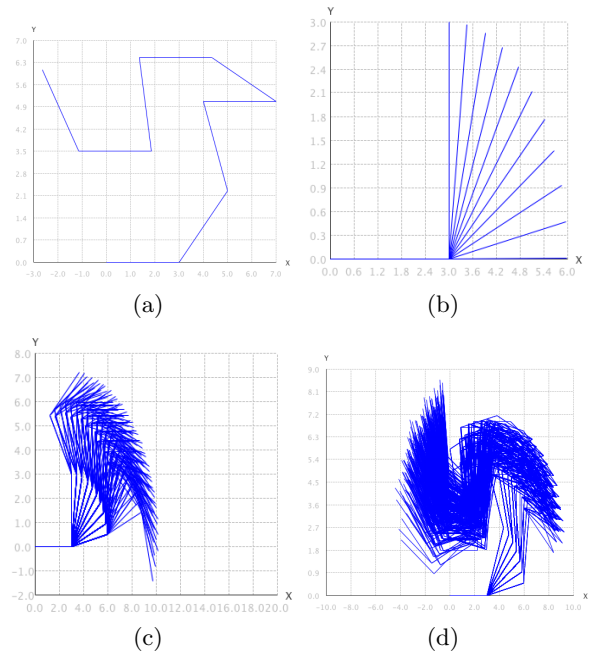


Figure 8: (a)The original polyline. (b)The output after the second step of the algorithm using the DC-matrix in Table 2. (c)The considered solutions in step 4 of the algorithm. (d)A set of complete possible solutions.

Table 2: The DC-matrix of the polyline in Figure 8(a).

- + 0+	- + + +	- + + -	- - + +	- - + +	- - + -	- - + +	+ + + +
	- + 0+	- + + -	- - + +	- - + +	- - + -	+ + + +	+ + + +
		- - 0-	- - + -	- - + -	- - + -	+ + + +	+ + + -
			- - 0+	- - + +	+ + + +	+ + - -	+ + - -
				- + 0+	- - + +	+ + - -	+ + - -
					- - 0+	+ + - -	+ + - -
						- - 0-	- - - -
							- + 0-

By algebraically interpreting the entries in a DC-matrix, as given in Property 1, algorithms from computational algebra, such as cylindrical algebraic decomposition (CAD) [4], can produce in exponential time example polylines satisfying a given DC-matrix (and can also be used to verify their realizability). In particular, for a polyline of size N , we have $4^{\frac{N(N-1)}{2}}$ equalities and inequalities, each of which involves at most 8 variables. In total there are $2(N + 1)$ variables. The CAD algorithm spends exponential (in N) time on producing a sample polyline. Experiments in systems like QEPCAD, Redlog and Mathematica [1, 2, 3], show that only polylines of up to 5 segments can be reconstructed this way.

In this section, we simplify the reconstruction task by looking for polylines with segments of *equal length*. This is not a big constraint, given an arbitrary polyline P , the present authors [13], describe a generalization algorithm that generate a polyline P^{2^N} with $\text{sem}(P) \approx \text{sem}(P^{2^N})$ and where the standard deviation of the lengths of the line segments of P^{2^N} tends to 0.

For a given DC-matrix M of size N by N ; the desired lengths of the line segments L ; and the granularity (the number of iteration steps) s , the reconstruction algorithm goes as follows. In the first step, the algorithm creates a line of length L with as start vertex $(0, 0)$ and end vertex $(L, 0)$.

In the second step, the algorithm creates a set of candidate second line segments. By looking at $M[1, 2]$, we know what the minimum and maximum angle is between the first and the second line segment. There are two possibilities, the minimum angle is equal to maximum angle or not. In the first case, this means that the second line is on the reference line RL_{12} or the perpendicular line PL_2 and thus there is a unique solution to position the second line segment. In the other case, we create s candidate solutions, equally distributed between the minimum and maximum angle. Given the DC-matrix in Table 2, which is the DC-matrix of the polyline in Figure 8(a), the output of the algorithm with $L = 3$ and $s = 10$, in the second step looks like the polyline in Figure 8(b).

In the i -th step of the algorithm ($i \leq N$), we take one by one the polylines created in the $(i - 1)$ -th step, and create new polylines as we did in the second step. The only difference now is that when we have a candidate polyline P_{new} we first check if $DCM(P_{new}) = M[1..i, 1..i]$. Only when this is the case P_{new} is further considered as possible solution for M . Given the DC-matrix in Table 2, the output of the algorithm after the fourth step of the algorithm looks like the polyline in Figure 8(c). Figure 8(d) gives polylines that all have the DC-matrix of Table 2. We remark that Figure 8(d) contains not all polylines that were considered as solution in Figure 8(c), especially the polylines of which the end vertex was below the X -axis. In this experiment, 632 polylines satisfying the DC-matrix of Table 2 were created in approximately 3 seconds on a Apple Macbook with 2.16 GHz Intel Core 2 Duo processor and 1 GB RAM.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have looked for a geometric interpretation of DC-similarity. Since it is difficult to find an interpretation for arbitrary polylines in the plane (e.g., given an arbitrary DC-matrix M of size N by N , exponential time (in N) is needed to generate a polyline P with $DCM(P) = M$), we studied two easier cases.

As a first case, we considered polylines on a grid. here, we have characterized geometrically DC-similarity of polylines on a grid. As a second case, where we limit the full difficulty of the problem by assuming polylines with equal-length line segments, we still lack a geometric interpretation, but we give a heuristic algorithm that produces for a given DC-matrix, example polylines with equal length line segments. In the future, we hope to find a general and effective geometric interpretation of DC-similarity.

Acknowledgments

This research has been partially funded by the European Union under the FP6-IST-FET programme, Project n. FP6-14915, GeoPKDD: Geographic Privacy-Aware Knowledge Discovery and Delivery”, (www.geopkdd.eu) and by the Research Foundation Flanders (FWO-Vlaanderen), Research Project G.0344.05.

6. REFERENCES

- [1] Mathematica 6. <http://www.wolfram.com>.
- [2] Qepcad. <http://www.cs.usna.edu/~qepcad>.
- [3] Redlog. <http://www.fmi.uni-passau.de/~redlog>.
- [4] B. Caviness and J. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer, 1998.
- [5] A. G. Cohn and J. Renz. Qualitative spatial representation and reasoning. In *Handbook of Knowledge Representation*, chapter 13, pages 551–596. Elsevier, 2007.
- [6] N. V. de Weghe. *Representing and Reasoning about Moving Objects: A Qualitative Approach*. PhD thesis, Ghent University, Faculty of Sciences, Department of Geography, 2004.
- [7] N. V. de Weghe, B. Kuijpers, P. Bogaert, and P. D. Maeyer. A qualitative trajectory calculus and the composition of its relations. In *GeoS*, volume 3799 of *Lecture Notes in Computer Science*, pages 60–76. Springer, 2005.
- [8] M. J. Egenhofer and D. M. Mark. Naive geography. In *COSIT*, pages 1–15, 1995.
- [9] K. D. Forbus. Qualitative physics: Past, present, and future. In *Readings in Qualitative Reasoning about Physical Systems*, pages 11–39. Kaufmann, 1990.
- [10] C. Freksa. Using orientation information for qualitative spatial reasoning. In A. F. et al., editor, *Spatio-Temporal Reasoning*, volume 639 of *Lecture Notes in Computer Science*, pages 162–178. Springer, 1992.
- [11] F. Giannotti and D. Pedreschi, editors. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008.
- [12] R. H. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005.
- [13] B. Kuijpers, B. Moelans, and N. V. de Weghe. Qualitative polyline similarity testing with applications to query-by-sketch, indexing and classification. In *Proceedings ACM-GIS '06*, pages 11–18, 2006.
- [14] M. Nanni, B. Kuijpers, C. Körner, M. May, and D. Pedreschi. Spatiotemporal data mining. In *Mobility, Data Mining and Privacy*, chapter 10, pages 267–296. Springer, 2008.
- [15] G. Trajcevski, H. Ding, P. Scheuermann, R. Tamassia, and D. Vaccaro. Dynamics-aware similarity of moving objects trajectories. In *Proceedings ACM-GIS'07*, pages 75–82, 2007.
- [16] M. J. van Kreveld and J. Luo. The definition and computation of trajectory and subtrajectory similarity. In *Proceedings ACM-GIS*, 2007.
- [17] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. J. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings ACM SIGKDD*, pages 216–225, 2003.
- [18] O. Wolfson. Moving objects information management: The database challenge. In *Proceedings of NGITS*, pages 75–89, 2002.
- [19] K. Zimmermann and C. Freksa. Qualitative spatial reasoning using orientation, distance, and path knowledge. *Appl. Intell.*, 6(1):49–58, 1996.