

Combinatorial structure of rigid transformations in 2D digital images

Phuc Ngo^a, Yukiko Kenmochi^a, Nicolas Passat^b, Hugues Talbot^a

^a*Université Paris-Est, LIGM, UPEMLV-ESIEE-CNRS, France*

^b*Université de Strasbourg, LSIIT, UMR 7005 CNRS, France*

Abstract

Rigid transformations are involved in a wide range of digital image processing applications. When applied on discrete images, rigid transformations are usually performed in their associated continuous space, requiring a subsequent digitization of the result. In this article, we propose to study rigid transformations of digital images as fully discrete processes. In particular, we investigate a combinatorial structure modelling the whole space of digital rigid transformations on arbitrary subset of \mathbb{Z}^2 of size $N \times N$. We describe this combinatorial structure, which presents a space complexity $\mathcal{O}(N^9)$ and we propose an algorithm enabling to construct it in linear time with respect to its space complexity. This algorithm, which handles real (*i.e.*, non-rational) values related to the continuous transformations associated to the discrete ones, is however defined in a fully discrete form, leading to exact computation.

Keywords: digital rigid transformations, combinatorial structure, discrete algorithm.

1. Introduction

Rigid transformations, (*i.e.*, transformations based on translations and rotations) are frequently involved in the design of computer vision and image processing techniques (*e.g.*, object tracking [1], image registration [2]), and considered in applications related to 2D or 3D images (*e.g.*, remote sensing, medical imaging). Despite the digital nature of the processed images, such transformations are generally performed by considering the Euclidean space (\mathbb{R}^n) associated to the Eulerian space (\mathbb{Z}^n) of the data. Such “partially continuous” transformations then need to be interfaced with a subsequent digitization process to finally obtain a result in \mathbb{Z}^n .

The purpose of this article is to study rigid transformations of digital images as fully discrete processes. Discrete processes for the classes of rotations and affine transformations have been studied in the literature. One can cite the quasi-shear rotation [3] and quasi-affine [4] for instance. Their approach consists of decomposing transformations into three shears and then obtain the discrete transformations. From the decomposition, the transformations preserve their bijectivity. One major drawback of this approach is that the result obtained after composing the three shears is not always identical to the discretized result of the initial transform. Here, we study an approach which preserves such discretization rather than the bijectivity. Moreover, we consider several issues related to (i) the number of possible rigid transformations given a finite subspace of \mathbb{Z}^n , (ii) the way to generate all of them, and (iii) the relations between all such transformations. Some combinatorial and algorithmic answers are provided to these questions, in the case of \mathbb{Z}^2 . Recently, C. Hundt and M. Liśkiewicz have proposed in [5, 6, 7, 8]

a discretization technique under projective transformations and some of its subclasses for the problem of 2D pattern matching. Our approach is inspired by their discretization technique and adapted for the classes of rigid transformations.

The contributions of this article are twofold. The first –more theoretical– consists of the proposal of a combinatorial structure (namely a graph) modelling the whole space of digital rigid transformations on arbitrary subset of \mathbb{Z}^2 of size $N \times N$, and the links between these transformations. These links correspond to the discontinuities induced by the digitization of the continuous transformations in \mathbb{R}^2 associated to those defined in \mathbb{Z}^2 . This combinatorial structure presents a space complexity $\mathcal{O}(N^9)$. On the one hand, this first result provides a contribution to the field of combinatorial analysis of image transformations (where previous works, summarised in Table 1, have already been proposed for rotations [9, 10], scalings [11, 12], combined scalings and rotations [5], affine transformations [6, 7], projective and linear transformations [8]). On the other hand, a new concept related to the topology of digital transformations is introduced in this structure, namely a *neighborhood* relationship between transformations induced by their relations.

The second contribution –of a more methodological nature– is an efficient algorithm enabling the construction of this combinatorial structure, with a computational cost linear with respect to its space complexity. This algorithm, which handles real (*i.e.*, non-rational) values related to the continuous transformations associated to the discrete ones, is however defined in a fully discrete form, allowing for exact computation, and avoiding in particular any approximations related to floating point arithmetic.

Classes of transformations	Complexity
Rotations [9, 10]	$\mathcal{O}(N^3)$
Scalings [11, 12]	$\mathcal{O}(N^3)$
Rotations and scalings [5]	$\mathcal{O}(N^6)$
Linear transformations [8]	$\mathcal{O}(N^{12})$
Affine transformations [6, 7]	$\mathcal{O}(N^{18})$
Projective transformations [8]	$\mathcal{O}(N^{24})$

Table 1: Space complexity of different classes of transformations on a subspace of \mathbb{Z}^2 of size $N \times N$.

The article is organised as follows. Sections 2 and 3 restates background notions and useful concepts. Section 4 introduces two notions: those of *tipping surfaces* and *tipping curves*, which constitute the basis of the proposed combinatorial structure. Section 5 actually defines this structure, while Sections 6 and 7 propose an algorithm for its construction. Complexity and experiments are discussed in Section 8. Section 9 presents some applications of the proposed graph and Section 10 concludes the article.

2. Geometric transformations and digitization

2.1. 2D images and digitization

Let V be a set called *value space* containing at least two elements, including one, noted \perp , corresponding to the “background”. A *2D image* is a function $\mathcal{I} : \mathbb{R}^2 \rightarrow V$ such that $\mathcal{I}^{-1}(V \setminus \{\perp\})$ is finite. (As examples, if $|V| = 2$, we say that \mathcal{I} is a binary image; if $|V|$ is equipped with a total order, we say that \mathcal{I} is a grey-level image). Without loss of generality, we

can suppose that $\mathcal{I}^{-1}(V \setminus \{\perp\}) \subseteq E =]-\frac{1}{2}, N - \frac{1}{2}[^2$ for a given $N \in \mathbb{N}$. The set E is called the *support* of \mathcal{I} and N is called the *size* of \mathcal{I} .

A *2D digital image* is defined as a function $I : \mathbb{Z}^2 \rightarrow V$ such that $I^{-1}(V \setminus \{\perp\})$ is finite. Without loss of generality, we can suppose that $I^{-1}(V \setminus \{\perp\}) \subseteq S = [0, N - 1]^2$ for a given $N \in \mathbb{N}$. The set S is called the *support* of I and N is the *size* of I .

In the sequel, by abuse of notation, we will sometimes set $\mathcal{I} : E \rightarrow V$ and $I : S \rightarrow V$, instead of $\mathcal{I} : \mathbb{R}^2 \rightarrow V$ and $I : \mathbb{Z}^2 \rightarrow V$.

The 2D digital image I can be seen as a numerical representation of the 2D image \mathcal{I} obtained by a *digitization* process that associates $I : S \rightarrow V$ to $\mathcal{I} : E \rightarrow V$, such that $S = E \cap \mathbb{Z}^2$. It then maps each point of E to exactly one point of S according to a square grid structure. This digitization process is defined *via* the following function:

$$\left| \begin{array}{lll} D & : & E \quad \longrightarrow \quad S \\ & & \mathbf{x} = (x, y) \quad \longmapsto \quad \mathbf{p} = (p, q) = ([x], [y]) \end{array} \right. \quad (1)$$

where $[\cdot]$ is the rounding function. Note in particular that we have $I = \mathcal{I}|_S$, that is $I(\mathbf{x}) = \mathcal{I}(\mathbf{x})$ for any $\mathbf{x} \in S$. In other words, we have $D|_S = Id_S$.

Broadly speaking, the real points $\mathbf{x} \in E$ are mapped onto the integer point $\mathbf{p} \in S$ with respect to the Voronoi diagram of S in E . This mapping is deterministic everywhere except on the frontier of the Voronoi cells, that is, for the real points $\mathbf{x} \in E$ presenting at least one semi-integer coordinate. We call the subset \mathcal{H} of E that consist of all such points the *half-grid*:

$$\mathcal{H} = E \cap \left[(\mathbb{R} \times (\mathbb{Z} + \frac{1}{2})) \cup ((\mathbb{Z} + \frac{1}{2}) \times \mathbb{R}) \right]. \quad (2)$$

Broadly speaking, the digitization function D decomposes E into unit

(open) squares, namely the *pixels*, whose centres are points of \mathbb{Z}^2 . In other words, the half-grid \mathcal{H} represents the union of the boundaries of all pixels.

The correct handling of this half-grid, which can be seen as the “inter-pixel” space in digital imaging is sometimes of great importance, for instance in (digital) topology [13, 14]. However, in the digital geometry context of this article, the way the points of \mathcal{H} are digitized is not crucial, since they represent an infinitesimal part of E . Nevertheless, \mathcal{H} remains of first importance in the understanding of digitization, since the discontinuities induced by this process will occur on the points located in this set.

2.2. Geometric transformations for digital images

We call *geometric transformation* any bijective function $\mathcal{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. Such a geometric transformation applied on an image $\mathcal{I} : \mathbb{R}^2 \rightarrow V$ provides a new transformed image $\mathcal{I} \circ \mathcal{T}^{-1} : \mathbb{R}^2 \rightarrow V$.

However it is not possible to apply directly \mathcal{T} on a digital image I in order to define $I \circ \mathcal{T}^{-1}$, since there are no guarantees that $\mathcal{T}^{-1}(\mathbf{p}) \in \mathbb{Z}^2$ for all $\mathbf{p} \in \mathbb{Z}^2$. The handling of geometric transformations for digital images then requires the definition of a function noted $T^{-1} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$ (which is not necessarily bijective) such that we can define the digital transformed image $I \circ T^{-1} : \mathbb{Z}^2 \rightarrow V$ being correct with respect to \mathcal{T}^{-1} . This can be conveniently obtained by setting:

$$T^{-1} = D \circ \mathcal{T}^{-1} \tag{3}$$

as illustrated in Figure 1. Such a transformation T is called a *2D digital image transformation*.

Remark 1. In the “nearest neighbour” digitization paradigm considered in

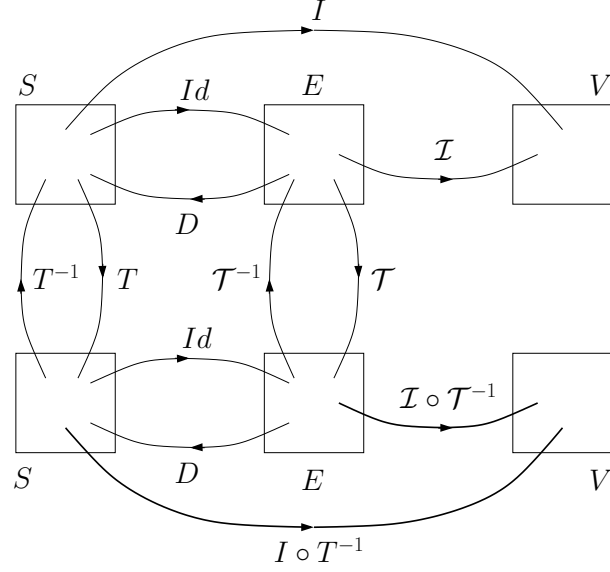


Figure 1: Geometric transformation and the associated digital image transformation (see text). First row: image support before transformation. Second row: image support after transformation.

Section 2.1 (see Equation (1)), we can consider two models for image transformations. Given a transformation \mathcal{T} , the first (the Lagrangian model) consists of determining $\mathcal{T}(\mathbf{x})$ for any point \mathbf{x} in the initial space, while the second (the Eulerian model) consists of determining $\mathcal{T}^{-1}(\mathbf{y})$ for any point \mathbf{y} in the deformed space. Both models are equal for a transformation $\mathcal{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, since \mathcal{T} is bijective. In the digital case, these models are actually distinct, since $D \circ \mathcal{T}^{-1}$ and $D \circ \mathcal{T}$ are not bijective, in general. In the current work, and without loss of generality, we focus on the Lagrangian model (see Figure 2). The results shown in this article remain however valid for both models. Still without loss of generality (due to symmetry considerations), we will consider in the sequel T and $D \circ \mathcal{T}$ instead of T^{-1} and $D \circ \mathcal{T}^{-1}$ (see Equation (3)).

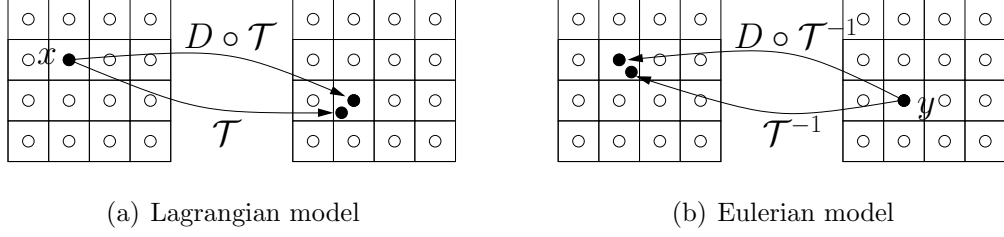


Figure 2: Image transformation models: (a) Lagrangian and (b) Eulerian. (a) and (b) left: image support before transformation, right: image support after transformation. The digital support $S \subset \mathbb{Z}^2$ is depicted by white dots. The Voronoi cells of E associated to the points of S are depicted by squares.

3. Rigid transformations for digital images

3.1. Digital rigid transformations

From this point on, we only consider *2D rigid transformations* which are composed of translations and rotations. Formally, a rigid transformation $\mathcal{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ transforms any point $\mathbf{p} = (p, q) \in \mathbb{R}^2$ into a point $\mathbf{p}' \in \mathbb{R}^2$ according to the following relation:

$$\mathcal{T}(\mathbf{p}) = \mathbf{p}' = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} \quad (4)$$

where $a, b, \theta \in \mathbb{R}$.

If we consider $T : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$, from Equations (1), (3) and (4), the transformed point $T(\mathbf{p}) = \mathbf{p}' = (p', q')$ associated to $\mathbf{p} \in \mathbb{Z}^2$ is then defined as:

$$T(\mathbf{p}) = \begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p \cos \theta - q \sin \theta + a] \\ [p \sin \theta + q \cos \theta + b] \end{pmatrix}. \quad (5)$$

This transformation is called a *2D digital rigid transformation*. Note that any rigid transformation (resp. digital rigid transformation) can be represented

by a triplet of parameters (a, b, θ) , noted $\mathcal{T}_{ab\theta}$ (resp. $T_{ab\theta}$). We note \mathbb{T} (resp. \mathbb{T}_d) the set of all rigid transformations (resp. digital rigid transformations):

$$\mathbb{T} = \{\mathcal{T}_{ab\theta} \mid (a, b, \theta) \in \mathbb{R}^3\}, \quad (6)$$

$$\mathbb{T}_d = \{T_{ab\theta} \mid (a, b, \theta) \in \mathbb{R}^3\}. \quad (7)$$

Remark 2. *A digital rigid transformation T (resp. a rigid transformation \mathcal{T}) maps the support S (resp. E) –of size N – of a digital image I (resp. an image \mathcal{I}) onto a set $T(S)$ (resp. $\mathcal{T}(E)$) that is included in a finite (resp. bounded) part of \mathbb{Z}^2 (resp. \mathbb{R}^2) corresponding to a disk of diameter $N\sqrt{2}$. In the following discussion, as the complexity is expressed as an order of N , the multiplicative constant factor $\sqrt{2}$ does not matter. Without loss of correctness, we can assume that both S and $T(S)$ (resp. E and $\mathcal{T}(E)$) are of size N such that N is “sufficiently large” with respect to S (resp. E) and the $\sqrt{2}$ factor.*

3.2. Discontinuities of digital rigid transformations

Let us define the function \mathfrak{T} (resp. \mathfrak{T}_d), which associates to each triplet of parameters (a, b, θ) the rigid transformation (resp. digital rigid transformation) modelled by these parameters (see (4) and (5)):

$$\left| \begin{array}{lll} \mathfrak{T} & : \mathbb{R}^3 & \longrightarrow \mathbb{T} \\ & (a, b, \theta) & \longmapsto \mathfrak{T}(a, b, \theta) = \mathcal{T}_{ab\theta}, \end{array} \right. \quad (8)$$

$$\left| \begin{array}{lll} \mathfrak{T}_d & : \mathbb{R}^3 & \longrightarrow \mathbb{T}_d \\ & (a, b, \theta) & \longmapsto \mathfrak{T}_d(a, b, \theta) = T_{ab\theta}. \end{array} \right. \quad (9)$$

The function \mathfrak{T} is continuous on \mathbb{R}^3 . In other words, for any value $\delta > 0$, there exists $\epsilon > 0$ such that for all $(a_1, b_1, \theta_1), (a_2, b_2, \theta_2) \in \mathbb{R}^3$

$$\|(a_1, b_1, \theta_1) - (a_2, b_2, \theta_2)\| < \epsilon \Rightarrow \|\mathfrak{T}(a_1, b_1, \theta_1) - \mathfrak{T}(a_2, b_2, \theta_2)\| < \delta \quad (10)$$

for given norms $\|\cdot\|$ on \mathbb{R}^3 and $(\mathbb{R}^3)^{\mathbb{R}^3} = \{F : \mathbb{R}^3 \rightarrow \mathbb{R}^3\}$ (for instance the N_∞ and pointwise N_∞ norms, respectively).

Due to the digitization involved in the definition of T in Equation (5), the function \mathfrak{T}_d is piecewise constant on \mathbb{R}^3 . For the same reason, \mathfrak{T}_d presents discontinuities. More precisely, there exist some points $\mathbf{c} \in \mathbb{R}^3$ such that for any ball $\mathcal{B}(\mathbf{c}, \epsilon)$ of center c and radius $\epsilon > 0$, there exist $(a_1, b_1, \theta_1), (a_2, b_2, \theta_2) \in \mathcal{B}(\mathbf{c}, \epsilon)$ such that for some values $\delta > 0$ (for instance $\delta = 1$), we have

$$\|(a_1, b_1, \theta_1) - (a_2, b_2, \theta_2)\| < \epsilon \wedge \|\mathfrak{T}_d(a_1, b_1, \theta_1) - \mathfrak{T}_d(a_2, b_2, \theta_2)\| \geq \delta. \quad (11)$$

Such points $\mathbf{c} \in \mathbb{R}^3$ are critical points in the parameter space of the digital rigid transforms. They characterize, in particular, some transformations which map at least one point onto the half grid (see Figure 3). These considerations lead to the following definition.

Definition 1. Let $(a, b, \theta) \in \mathbb{R}^3$, and $\mathcal{T}_{ab\theta}$ be the associated rigid transformation. We say that $\mathcal{T}_{ab\theta}$ is a critical transformation if

$$\exists \mathbf{p} \in \mathbb{Z}^2, \mathcal{T}_{ab\theta}(\mathbf{p}) \in \mathcal{H}. \quad (12)$$

We denote by \mathbb{T}_c the set of all critical transformations.

The set \mathbb{T}_c of the critical transformations obviously corresponds to the part of \mathbb{R}^3 inducing discontinuities in \mathfrak{T}_d .

3.3. Discrete rigid transformations

As a result of the discontinuity of digital rigid transformations, it is possible for the digitization D of two rigid transformations of a digital image to

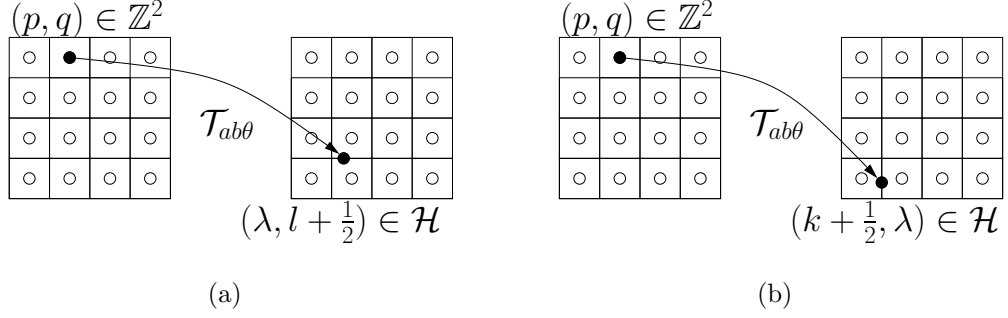


Figure 3: Examples of critical transformations $\mathcal{T}_{ab\theta}$, which map at least one integer value onto a “horizontal” (a) or “vertical” (b) half-grid point. The digital support $S \subset \mathbb{Z}^2$ is depicted by white dots. The Voronoi cells of E associated to the points of S are depicted by squares.

be identical. This leads to the following consideration about the equivalence between transformations.

Definition 2. Let $I : S \rightarrow V$. Two elements $\mathcal{T}, \mathcal{T}' \in \mathbb{T}$ are considered equivalent, and we write $\mathcal{T} \sim \mathcal{T}'$, iff $D \circ \mathcal{T}(\mathbf{p}) = D \circ \mathcal{T}'(\mathbf{p}), \forall \mathbf{p} \in S$.

Then an equivalence class in \mathbb{T} under \sim is a set of rigid transformations that provides the same digital transformed image.

Definition 3. For a given digital image $I : S \rightarrow V$, each equivalence class in \mathbb{T} (induced by I) is called a discrete rigid transformation (DRT).

Note that the term *discrete* thus refers to the non-continuous structure of transformations for digital images. In this way, the parameter space of the rigid transformations is partitioned into the disjoint sets of the equivalence classes, such that each of which associates exactly to one DRT.

4. Tipping surfaces and tipping curves

In this section, we introduce two notions, that allow us to study the decomposition of the parameter space into the DRTs. As stated in the preceding section, the digital rigid transformations present a discontinuous property, which is characterized by the critical transformations. The later transformations are actually related to the structure of DRTs in the parameter space, since they represent the boundaries of the equivalence classes. We propose the two following notions: *tipping surfaces* and *tipping curves* which are, in fact, analytical expressions of critical rigid transformations.

4.1. Tipping surfaces

According to Definition 1, a critical rigid transformation $\mathcal{T}_{ab\theta}$ moves at least one integer point $\mathbf{p} = (p, q) \in \mathbb{Z}^2$ into a half-grid point which can be either a vertical half-grid point $(k + \frac{1}{2}, \lambda)$ or a horizontal half-grid point $(\lambda, l + \frac{1}{2})$ for $k, l \in \mathbb{Z}$ and $\lambda \in \mathbb{R}$, or both. The set of values $(a, b, \theta) \in \mathbb{R}^3$ associated to this critical transformation forms a surface in the parameter space, called a *tipping surface*. Note that a tipping surface can be represented by an integer triplet associated to such a vertical or horizontal half-grid point.

Definition 4. Given an integer triplet (p, q, k) (resp. (p, q, l)), we define a vertical (resp. horizontal) tipping surface as the function Φ_{pqk} (resp. Ψ_{pql}) such that

$$\left| \begin{array}{lll} \Phi_{pqk} : & \mathbb{R}^2 & \longrightarrow \mathbb{R} \\ & (b, \theta) & \longmapsto a = k + \frac{1}{2} + q \sin \theta - p \cos \theta, \end{array} \right. \quad (13)$$

$$\left| \begin{array}{lll} \Psi_{pql} : & \mathbb{R}^2 & \longrightarrow \mathbb{R} \\ & (a, \theta) & \longmapsto b = l + \frac{1}{2} - p \sin \theta - q \cos \theta. \end{array} \right. \quad (14)$$

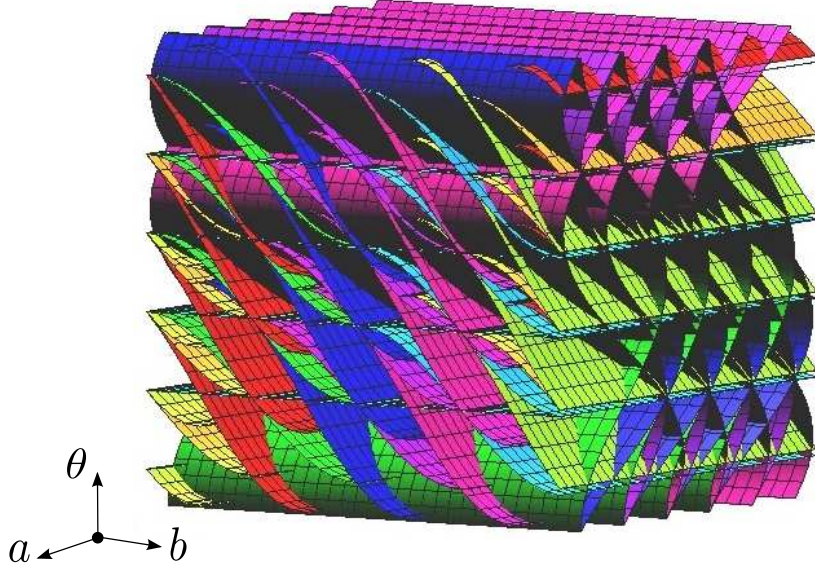


Figure 4: Discrete rigid transformations (DRTs) are observed as regions separated by tipping surfaces.

The sets of all tipping surfaces defined by Formulae (13) and (14) clearly correspond to the discontinuities of digital rigid transformations. They divide the parameter space (a, b, θ) into the equivalence classes. In each of these classes, any rigid transformation leads to an identical digital transformed image. We restate that each equivalence class is represented by a DRT. As the number of critical transformations is limited by the size of a given digital image I , the number of tipping surfaces and DRTs in the parameter space are both finite. Figure 4 visualizes the DRTs as regions separated by tipping surfaces Φ_{pqk} and Ψ_{pql} in the parameter space, for $p, q \in [0, 2]$ and $k, l \in [0, 3]$.

4.2. Tipping curves

We remark that the tipping surface Φ_{pqk} in Formula (13) is independent of b . Consequently, the cross-sections of Φ_{pqk} orthogonal to the b -axis orthogonal

are identical for any $b \in \mathbb{R}$, and each has the form of a trigonometric curve. We call these *tipping curves*, and we denote them by ϕ_{pqk} . This means that Formula (13) defines a tipping curve as well if it is observed in the plane (a, θ) . Similarly, Formula (14) is also considered as a tipping curve on the plane (b, θ) , denoted by ψ_{pql} .

Definition 5. *Given an integer triplet (p, q, k) (resp. (p, q, l)), we define a vertical (resp. horizontal) tipping curve as a function ϕ_{pqk} (resp. ψ_{pql}) such that*

$$\left| \begin{array}{l} \phi_{pqk} : \mathbb{R} \longrightarrow \mathbb{R} \\ \theta \longmapsto a = k + \frac{1}{2} + q \sin \theta - p \cos \theta, \end{array} \right. \quad (15)$$

$$\left| \begin{array}{l} \psi_{pql} : \mathbb{R} \longrightarrow \mathbb{R} \\ \theta \longmapsto b = l + \frac{1}{2} - p \sin \theta - q \cos \theta. \end{array} \right. \quad (16)$$

The sets of all tipping curves ϕ_{pqk} and ψ_{pql} provide two families of critical rigid transformations projected on the planes (a, θ) and (b, θ) respectively.

Definition 6. *Two families of tipping curves are defined for $\theta \in \mathbb{R}$ by*

$$F_\phi(\theta) = \{\phi_{pqk}(\theta) \mid p, q, k \in \mathbb{Z}\}, \quad (17)$$

$$F_\psi(\theta) = \{\psi_{pql}(\theta) \mid p, q, l \in \mathbb{Z}\}. \quad (18)$$

Note that an operation applied to a family of curves means that it is applied to all tipping curves belonging to the family. For instance, $-F_\phi(\theta) = \{-\phi_{pqk}(\theta) \mid \phi_{pqk}(\theta) \in F_\phi(\theta)\}$.

Figure 5 illustrates a part of F_ϕ for $p, q \in [0, 4]$ and $k \in [0, 5]$.

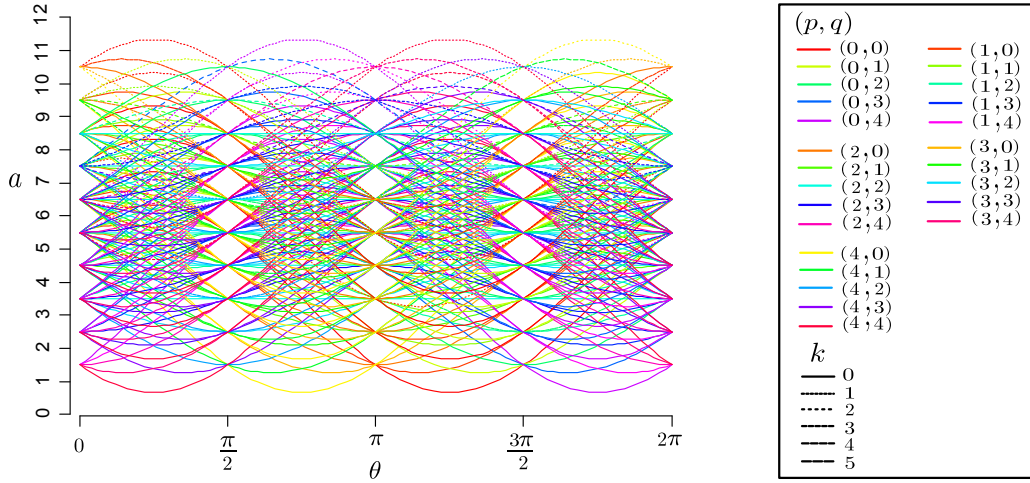


Figure 5: A part of the family of tipping curves F_ϕ .

4.3. Properties of tipping curves

So far we have studied the definitions of tipping surfaces and tipping curves. We now discuss the combinatorial structure of discrete rigid transformations by using the sets of tipping surfaces and curves. Before this, we state some properties about tipping curves. From Definition 5, we remark that F_ϕ and F_ψ have similar forms. Indeed, if we translate F_ψ by $\frac{\pi}{2}$ with respect to θ , then we obtain F_ϕ . For simplicity, we will thus show the properties for F_ϕ that are valid for F_ψ as well.

The first property concerns the unique representation of tipping curves. This is a direct result of the fact that the three functions $\{1, \cos, \sin\}$ involved in Formula (15) are linearly independent.

Property 1. *There exists a unique integer triplet $(p, q, k) \in \mathbb{Z}^3$ for each tipping curve.*

The next property shows the relationships between two tipping curves.

This will be used in the algorithm of construction of the combinatorial structure of discrete rigid transformations. Geometrically, two tipping curves can relate to each other in four different ways (see Figure 6): they can be identical, intersecting¹, tangent² or not intersecting. Analytical interpretation of these relationships is expressed as follows.

Property 2. *let ϕ_{pqk} and $\phi_{p'q'k'}$ be two tipping curves. Setting $K = k - k'$, $P = p - p'$, $Q = q - q'$, $\Delta_1 = P^2(P^2 + Q^2 - K^2)$, $\Delta_2 = Q^2(P^2 + Q^2 - K^2)$ and $\Delta = \Delta_1 + \Delta_2$, we have the following relations between ϕ_{pqk} and $\phi_{p'q'k'}$:*

- *if $P = Q = 0$,*
 - (i) *if $K = 0$ then ϕ_{pqk} and $\phi_{p'q'k'}$ are identical,*
 - (ii) *if $K \neq 0$ then they have no intersection,*
- *otherwise,*
 - (iii) *if $\Delta = 0$ and $|KP| \leq P^2 + Q^2$ and $|KQ| \leq P^2 + Q^2$ then they are tangent,*
 - (iv) *if $\Delta > 0$ and $|KP \pm \sqrt{\Delta_1}| \leq P^2 + Q^2$ and $|KQ \pm \sqrt{\Delta_2}| \leq P^2 + Q^2$ then they intersect, (note that \pm means “+ or -”)*
 - (v) *otherwise, they have no intersection.*

Proof (i) is easily induced by Property 1. Assume that two tipping curves ϕ_{pqk} and $\phi_{p'q'k'}$ are different. The solution set satisfying both equations ϕ_{pqk} and $\phi_{p'q'k'}$ determines the nature of the relationships between two curves; it provides either the tangent point (iii), the intersection points (iv) or the

¹Two tipping curves intersect if they cross each other at a point.

²Two tipping curves are tangent if both share the same tangent line at a point, *i.e.*, the curves touch but do not cross each others.

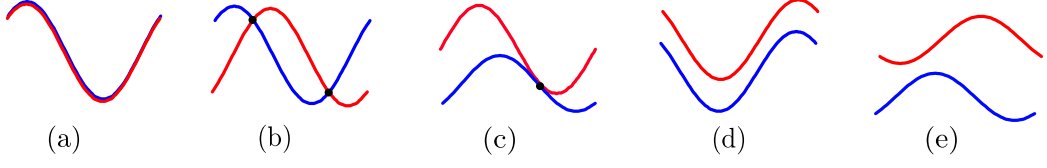


Figure 6: Relationships between two tipping curves, which are identical (a), intersecting (b), tangent (c) and not intersecting (d,e).

empty set (ii)(v). Supposing the equation system has a solution, then we can find a value θ satisfying the following:

$$k + \frac{1}{2} - p \cos \theta + q \sin \theta = k' + \frac{1}{2} - p' \cos \theta + q' \sin \theta.$$

Replacing $K = k - k'$, $P = p - p'$ and $Q = q - q'$, then we have

$$K - P \cos \theta = -Q \sin \theta. \quad (19)$$

Squaring both sides of Equation (19), we obtain

$$K^2 - 2KP \cos \theta + P^2 \cos^2 \theta = Q^2 \sin^2 \theta.$$

Because $\cos^2 \theta + \sin^2 \theta = 1$, we have

$$(P^2 + Q^2) \sin^2 \theta - 2KQ \sin \theta + K^2 - P^2 = 0, \quad (20)$$

$$(P^2 + Q^2) \cos^2 \theta + 2KP \cos \theta + K^2 - Q^2 = 0, \quad (21)$$

which are quadratic equations if $P^2 + Q^2 \neq 0$. As we assume that ϕ_{pqk} and $\phi_{p'q'k'}$ are different, either P , Q or K are not equal to 0. If $P = Q = 0$ and $K \neq 0$, then Equation (19) has no solution (ii). Otherwise, $P^2 + Q^2 \neq 0$ so that the discriminants Δ_1 and Δ_2 of Equations (20) and (21) determine the number and nature of the roots. There are three cases; the curves are:

- tangent (iii): each of Equations (20) and (21) has one solution, if $\Delta = 0$ (namely, $\Delta_1 = \Delta_2 = 0$) and $|KP| \leq P^2 + Q^2$ and $|KQ| \leq P^2 + Q^2$, since $-1 \leq \sin \theta \leq 1$ and $-1 \leq \cos \theta \leq 1$,
- intersecting (iv): at least one of Equations (20) and (21) has more than one solution, if $\Delta > 0$ (namely, either $\Delta_1 \geq 0 \wedge \Delta_2 > 0$ or $\Delta_1 > 0 \wedge \Delta_2 \geq 0$) and $|KP \pm \sqrt{\Delta_1}| \leq P^2 + Q^2$ and $|KQ \pm \sqrt{\Delta_2}| \leq P^2 + Q^2$, since $-1 \leq \sin \theta \leq 1$ and $-1 \leq \cos \theta \leq 1$,
- otherwise, there are no intersections (v).

■

Given two curves ϕ_{pqk} and $\phi_{p'q'k'}$, we can find out their relationship by evaluating the integer triplets (p, q, k) and (p', q', k') exactly, *i.e.*, by using only integers during computation. Indeed, the values $P, Q, K, \Delta_1, \Delta_2$ and Δ are all integers, while the inequalities $|KP \pm \sqrt{\Delta_1}| \leq P^2 + Q^2$ and $|KQ \pm \sqrt{\Delta_2}| \leq P^2 + Q^2$ can be verified by using squaring procedures. The proof of Property 2 implies the following corollary.

Corollary 1. *Let us consider the same notations as in Property 2. Given two tipping curves ϕ_{pqk} and $\phi_{p'q'k'}$, if they are*

- *tangent, the tangent point satisfies $\sin \theta = \frac{KQ}{P^2+Q^2}$ and $\cos \theta = \frac{KP}{P^2+Q^2}$,*
- *intersecting, the intersection points satisfy $\sin \theta = \frac{KQ \pm \sqrt{\Delta_2}}{P^2+Q^2}$ and $\cos \theta = \frac{KP \pm \sqrt{\Delta_1}}{P^2+Q^2}$. Note that there are four possible combinations for $\sin \theta$ and $\cos \theta$, but only two of them are valid for intersections (see Corollary 2).*

Since Equation (19) has at most two solutions in $[0, 2\pi[$, we also have the following corollary.

Corollary 2. *Two distinct tipping curves have at most two intersections in $[0, 2\pi[$.*

As the tipping curves have the form of trigonometric functions with $\sin \theta$ and $\cos \theta$, they inherit the properties of the trigonometric functions as follows.

Property 3. *The family of tipping curves F_ϕ is symmetric:*

$$\begin{aligned} F_\phi(\theta) &= -F_\phi(\theta), \\ F_\phi(\theta) &= F_\phi(-\theta + m\frac{\pi}{4}), \quad \forall m \in \mathbb{Z}. \end{aligned}$$

Property 4. *The family of tipping curves F_ϕ is periodic:*

$$\begin{aligned} F_\phi(\theta) &= F_\phi(\theta + m\frac{\pi}{2}), \quad \forall m \in \mathbb{Z}, \\ F_\phi(\theta) &= F_\phi(\theta) + m, \quad \forall m \in \mathbb{Z}. \end{aligned}$$

So far, we have considered a digital image support as a set of points of \mathbb{Z}^2 . Under this assumption, the subdivision of the parameter space into DRTs is infinite. However a digital image is finite in practice, and we can then consider S instead of \mathbb{Z}^2 . Given a digital image I of size $N \times N$, we define the set of tipping surfaces with respect to I as Φ_{pqk} and Ψ_{pql} for $0 \leq p, q \leq N-1$, $0 \leq k, l \leq N$ and $p, q, k, l \in \mathbb{Z}$. Thus we have $2N^2(N+1)$ tipping surfaces partitioning the parameter space. Tipping curves ϕ_{pqk} and ψ_{pql} are correlated to the tipping surfaces, since they are the projections of Φ_{pqk} and Ψ_{pql} respectively on the planes (a, θ) and (b, θ) of the parameter space.

5. Graph representation of discrete rigid transformations

As discussed in the previous section, the parameter space of rigid transformations is divided into DRTs, whose boundaries are the tipping surfaces.

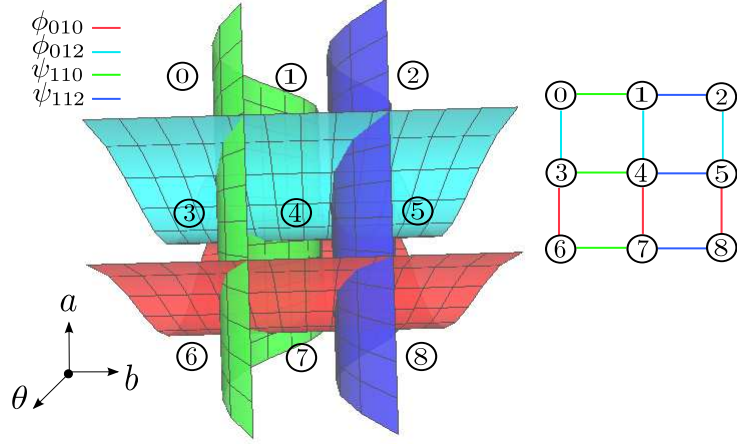


Figure 7: Four tipping surfaces in the parameter space (a, b, θ) (left) and its DRT graph (right).

This subdivision can be represented by a graph which is defined as follows.

Definition 7. Let V be a set of vertices and E be a set of labelled edges, such that

- a vertex $v \in V$ corresponds to a DRT, and
- an edge $e = (u, w, f) \in E$, where $f = \Phi_{pqk}$ or Ψ_{pql} , connects a pair of DRTs $\{u, w\} \in V$ separated by the tipping surface f , which is considered as the label of the edge.

The graph $G = (V, E)$ is called a discrete rigid transformation graph (DRT graph) (see Figure 7).

In order to distinguish between the integer triplets of vertical and horizontal tipping surfaces, we use an integer quadruple (p, q, k, i) for modelling

each tipping surface, where $i = 0$ or 1 indicates respectively the vertical or horizontal set.

Strictly speaking, each vertex in V represents one digital transformed image generated by the corresponding DRT. Each labelled edge in E that links two vertices represents a critical transformation. This transformation links a digital image to a neighbouring one, such that they differ by only one pixel. Note that the DRT graph $G = (V, E)$ does not contain any geometric information, such as the parameter (a, b, θ) of rigid transformations, but only the integer quadruples (p, q, k, i) representing the tipping surfaces.

The construction of a DRT graph can be performed with the help of surface arrangements [15]. However this implies a complexity of $\Omega(n^4)$ for general surfaces, where n is the number of surfaces. In this article, we propose an algorithm with a better complexity of $\mathcal{O}(n^3)$ by using the properties of tipping surfaces described in Section 4. We know that while projecting two families of tipping surfaces on the planes (a, θ) and (b, θ) , we obtain the corresponding families of tipping curves defined by Formulae (13) and (14). The combinatorial structure of DRTs in a 2D parameter space is called a *2D DRT graph*. This graph is built from tipping curves on the planes (a, θ) or (b, θ) . The DRT graph can be reconstructed by combining these two 2D DRT graphs. For this, we will first describe an algorithm for building a 2D DRT graph. Then we extend it to 3D to reconstruct the complete DRT graph.

6. Construction of 2D discrete rigid transformation graph

This section presents a method for constructing a 2D DRT graph of a set of tipping curves generated from a given digital image I of size $N \times N$.

6.1. Problem formalization

A finite set of n tipping curves C partitions the plane into three types of regions: a *vertex* is an intersection point of curves, an *arc* is a largest connected portion of a curve that is not intersected by any other curve, and a *face* is a largest connected region that is not intersected by any other curve in C . We define as follows an incident graph represented by the partition C .

Definition 8. *Given a set of tipping curves C , the 2D DRT graph $G^C = (V^C, E^C)$ of C consists of a set V^C of vertices and a set E^C of labelled edges, such that:*

- *each vertex $v \in V^C$ corresponds to a face, and*
- *each edge $e = (u, w, \phi) \in E^C$ corresponds to an arc that connects two faces $\{u, w\} \in V^C$ sharing a boundary tipping curve ϕ .*

Note that the tipping curve ϕ is considered as a label of the edge $e \in E^C$. Figure 8 illustrates a 2D DRT graph.

The problem of the construction of G^C is related to *curve arrangements* [16]. Various methods have been proposed including incremental constructions [16], sweeping lines [17] and others. A comprehensive discussion on arrangements can be found in [16, 18]. Since our curves are tipping curves, many degenerate cases can present themselves, such as tangent and multiple intersections. In addition, we are only interested in the information about faces and arcs in the arrangement. Therefore, instead of using the basic algorithm of curve arrangement, we propose a variation of the *sweeping line* method for constructing the graph G^C . The main idea of the algorithm is that a (*vertical*) *cut* is swept throughout tipping curves, and stops at some

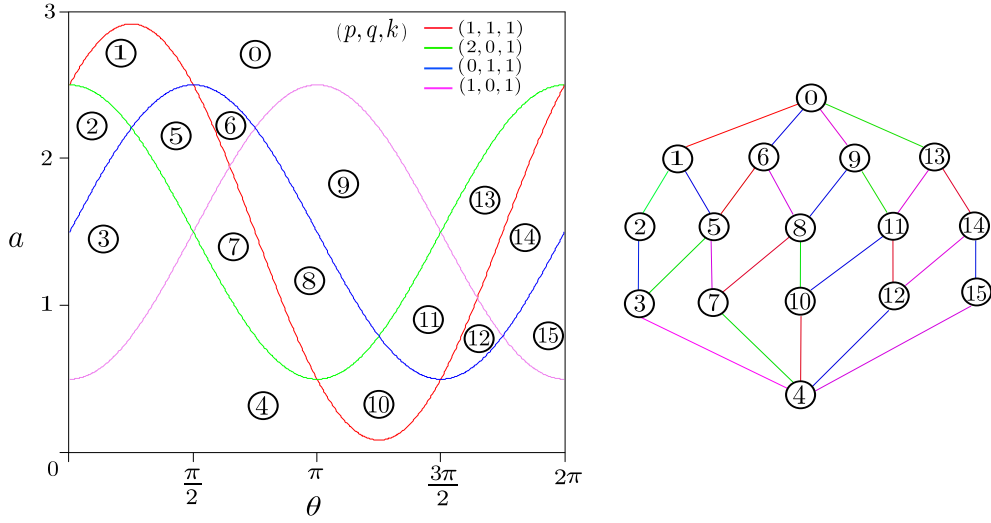


Figure 8: Four tipping curves on the plane (a, θ) (left) and the associated 2D DRT graph (right).

points to construct G^C incrementally. The details of algorithm and its implementation are explained in the sequel.

6.2. Principles of incremental 2D DRT graph reconstruction

Let C be a set of n tipping curves and G^C denotes the partition graph of C . We define a (*vertical*) *cut*, noted γ , as a monotonic line intersecting each tipping curve in C exactly once [17]. Note that γ is an unbounded simple curve and is represented by a sequence of tipping curves which γ intersects from top to bottom, as illustrated in Figure 9. We assume that γ starts at $\theta = 0$ and ends at $\theta = 2\pi$. While moving γ , its sequence of tipping curves changes, but not continuously. Indeed, γ changes only at intersection points, called *event points*. When γ reaches an event point, the algorithm updates γ and constructs a part of G^C . This is called an *elementary step* of the algorithm. As a set of event points forms a series of elementary steps,

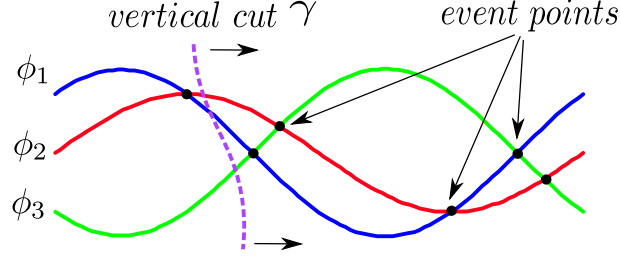


Figure 9: Event points and a vertical cut $\gamma = (\phi_2, \phi_1, \phi_3)$.

we need to maintain a sorted set of event points and make the curve γ go through them in their increasing order.

In fact, the cut γ can be also represented as a *directed graph* such that each edge corresponds to a tipping curve in γ and each vertex corresponds to a face bounded by two consecutive tipping curves in γ .

Definition 9. Let $\gamma = (\phi_1, \phi_2, \dots, \phi_n)$ be the cut. The partial graph $\delta G^C = (\delta V^C, \delta E^C)$ with respect to γ is defined as a directed graph, such that

- $\delta V^C = \{v_0, v_1, \dots, v_n\}$ is the set of vertices,
- $\delta E^C = ((v_0, v_1, \phi_1), (v_1, v_2, \phi_2), \dots, (v_{n-1}, v_n, \phi_n))$ is the ordered set of edges.

In practice, elements of δE^C are also ordered in the same way as γ .

At each elementary step, the partial graph δG^C is updated with respect to the change of γ . If such an operation is applied, then the sweep progresses such that δG_C is partially modified and integrated in the final graph G_C .

Proposition 1. Let G^C be the 2D DRT graph. Then we have

$$G^C = \bigcup_{i=0}^m \delta G_i^C \quad (22)$$

where δG_i^C is the partial graph at the i -th elementary step and m is the total number of ordered event points.

Note that a partial graph is a directed graph because we need edge directions during the update. However the final graph G^C is not directed, so that we do not keep directions while integrating δG_i^C into G^C .

6.3. Initial graph construction

The initialization step provides the graph δG_0^C with respect to the initial cut γ_0 . In fact, γ_0 is a sequence of tipping curves in C with the order defined by the following relation \prec_0 .

Definition 10. For any pair of tipping curves $\phi_{pqk}(\theta)$ and $\phi_{p'q'k'}(\theta)$, a relation \prec_0 is defined as $\phi_{pqk}(\theta) \prec_0 \phi_{p'q'k'}(\theta)$ iff

- $\phi_{pqk}(0) < \phi_{p'q'k'}(0)$, or
- $\phi_{pqk}(0) = \phi_{p'q'k'}(0)$ and the first derivatives verify $\phi'_{pqk}(0) < \phi'_{p'q'k'}(0)$, or
- $\phi_{pqk}(0) = \phi_{p'q'k'}(0)$ and $\phi'_{pqk}(0) = \phi'_{p'q'k'}(0)$ and $\phi_{pqk}(\pi) < \phi_{p'q'k'}(\pi)$.

From Definition 9, we can generate the initial graph δG_0^C of γ_0 . Note that δG_0^C corresponds to the cut at $\theta = \epsilon$, where ϵ is some very small positive value. The 2D DRT graph G^C is then initialized by δG_0^C .

6.4. Incremental 2D DRT graph construction for simple cases

We first present the construction algorithm in *simple* cases, *i.e.*, such that any intersection consists only of two crossing tipping curves (Figure 10(a)). We call the other cases *degenerate* or *non-simple* (see Figure 10(b)). We will discuss how to deal with such degeneracies in Section 6.5.

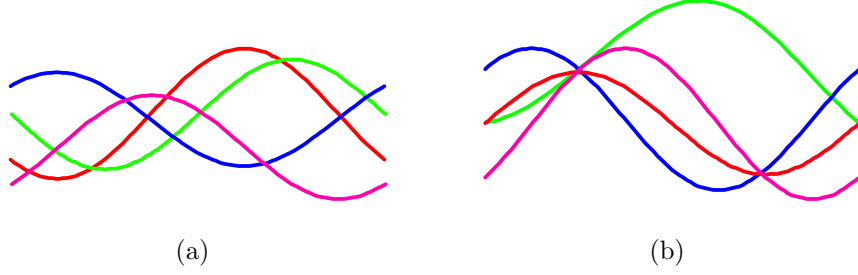


Figure 10: Illustration for simple (a) and non-simple (b) cases.

6.4.1. Detecting and ordering event points

The following question arises in the sweeping algorithm: how to detect event points, or how to discover when an elementary step is applied? Since event points are intersections of two tipping curves, the answer is linked to Property 2 (iv). More precisely, given two tipping curves ϕ_{pqk} and $\phi_{p'q'k'}$ modelled by the integer triplets (p, q, k) and (p', q', k') , they intersect if the following relation is satisfied: $\Delta_1 + \Delta_2 > 0$ and $|KP \pm \sqrt{\Delta_1}| \leq P^2 + Q^2$ and $|KQ \pm \sqrt{\Delta_2}| \leq P^2 + Q^2$, where $P = p - p'$, $Q = q - q'$, $K = k - k'$, $\Delta_1 = P^2(P^2 + Q^2 - K^2)$ and $\Delta_2 = Q^2(P^2 + Q^2 - K^2)$.

All event points need to be sorted and then stored in a *queue*, to be handled one by one. We call this queue of event points an *event queue*. It is defined as $\mathcal{Q} = (\mathcal{E}, \prec_{\mathcal{E}})$ where \mathcal{E} is a set of all event points and $\prec_{\mathcal{E}}$ is a binary relation defined on \mathcal{E} . The event points are sorted by $\prec_{\mathcal{E}}$ as follows.

Definition 11. For any pair of event points $\mathbf{u} = (u_x, u_y)$ and $\mathbf{v} = (v_x, v_y) \in \mathcal{E}$, a relation $\prec_{\mathcal{E}}$ is defined as $\mathbf{u} \prec_{\mathcal{E}} \mathbf{v}$ iff $u_x < v_x$ or $u_x = v_x$ and $u_y < v_y$.

Sorting event points can be performed with exact computation. We know that the coordinates (θ, a) of event points are typically irrational numbers and that we generally cannot compute an exact value for θ . Nevertheless, in

order to sort event points with respect to $\theta \in [0, 2\pi[$, we can use the values $\cos \theta$ and $\sin \theta$ which are calculated from Corollary 1. We can also easily obtain the value a from Formula (15) with $\cos \theta$ and $\sin \theta$. Note that a , $\cos \theta$ and $\sin \theta$ are all quadratic irrationals³. Therefore, sorting event points then relies on the capacity to compare quadratic irrationals. It is known in [19] that two quadratic irrationals can be compared by an exact method. In fact, a quadratic irrational can be represented exactly using a periodic continued fraction modelled by a sequence of integers, and this representation is unique. Moreover the comparison of periodic continued fractions can be performed in constant time [20] (see Appendix A for more details), so that sorting all event points requires $\mathcal{O}(|\mathcal{Q}| \log |\mathcal{Q}|)$ times of such a comparison.

Using integer arithmetic avoids the technical problems due to the use of floating point representations. Most importantly, this allows us to prevent detecting false event points that would arise from the limited precision of floating-point representation supported in computer implementations. This would be likely to occur in degenerate cases such as multiple tangent and/or intersecting points.

Each event point is represented by the intersecting tipping curves at that point, but not its coordinates, since they are more important for constructing G^C . In simple cases, any event point in \mathcal{Q} is thus stored as two tipping curves, represented by the integer triplets.

³A quadratic irrational is an irrational number that is a solution of some quadratic equations.

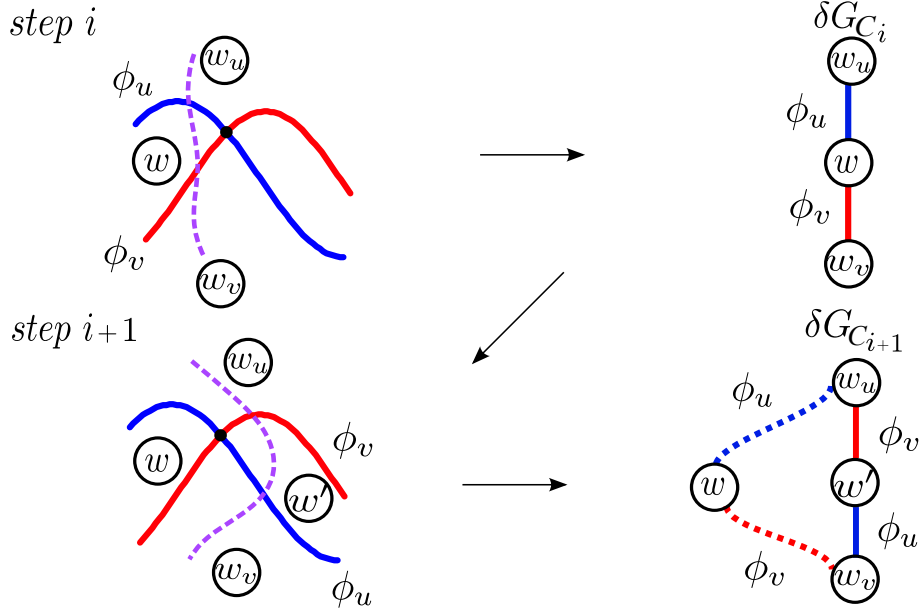


Figure 11: Illustration of a progress of a cut by which the partial graph δG_{i+1}^C is modified from δG_i^C in simple case.

6.4.2. Elementary step

An elementary step corresponds to a transposition of two curves in a cut γ around an event point, as illustrated in Figure 11. Following a change in γ , the partial graph δG^C is modified.

Formally, given an event point $\mathbf{q} = \{\phi_u, \phi_v\}$, if the cut on the left of \mathbf{q} is denoted by $\gamma_i = (\phi_{i_1}, \dots, \phi_u, \phi_v, \dots, \phi_{i_n})$, then after \mathbf{q} we have $\gamma_{i+1} = (\phi_{i_1}, \dots, \phi_v, \phi_u, \dots, \phi_{i_n})$. Let δG_i^C and δG_{i+1}^C denote the partial graph of γ_i and γ_{i+1} respectively. We can generate δG_{i+1}^C from δG_i^C according to the following four steps:

1. finding the current vertex w bounded by ϕ_u and ϕ_v ;
2. deleting two edges that are adjacent to w ;

Procedure 1: Elementary step for simple cases.

Input: A partial graph δG_i^C and an event point $\mathbf{q} = \{\phi_u, \phi_v\}$.

Output: A partial graph δG_{i+1}^C .

```

1  $e_u \leftarrow \varepsilon(\phi_u); e_v \leftarrow \varepsilon(\phi_v);$ 
2  $\{w\} \leftarrow \vartheta(e_u) \cap \vartheta(e_v);$ 
3  $\{w_u\} \leftarrow \vartheta(e_u) \setminus \{w\}; \{w_v\} \leftarrow \vartheta(e_v) \setminus \{w\};$ 
4  $\Delta V_-^C \leftarrow \{w\};$  //  $w$  is a removed vertex
5  $\Delta E_-^C \leftarrow \{(w_u, w, \phi_u), (w, w_v, \phi_v)\};$ 
6  $\Delta V_+^C \leftarrow \{w'\};$  //  $w'$  is a new vertex
7  $\Delta E_+^C \leftarrow \{(w_u, w', \phi_u), (w', w_v, \phi_v)\};$ 
8  $\delta G_{i+1}^C \leftarrow \delta G_i^C \setminus \Delta G_-^C \cup \Delta G_+^C;$ 
   /*  $\Delta G_-^C = (\Delta V_-^C, \Delta E_-^C)$  is the subtracting part from  $\delta G_i^C$ 
   and  $\Delta G_+^C = (\Delta V_+^C, \Delta E_+^C)$  is the adding part to  $\delta G_i^C$  */

```

3. replacing w by a new vertex w' ;
4. creating two new edges that are linked to w' .

This procedure is called an i -th elementary step, by which the partial graph of a cut is modified. The implementation is given in Procedure 1, which requires the following two functions:

- $\vartheta(e)$ returns two adjacent vertices of the edge e in δV_i^C ;
- $\varepsilon(\phi)$ returns the edge corresponding to the tipping curve ϕ in δE_i^C .

6.4.3. Algorithm

We now present an algorithm for incremental 2D DRT graph construction. The algorithm builds G^C by picking event points in \mathcal{Q} one by one. Each

iteration consists of modifying the partial graph δG^C according to the current cut γ (see Section 6.4.2), and then integrating δG^C into G^C .

Algorithm 1: Incremental construction of a 2D DRT graph in simple cases.

Input: A tipping curve set $C = \{\phi_1, \phi_2, \dots, \phi_n\}$.

Output: A 2D DRT graph $G^C = (V^C, E^C)$.

```

1 Initialize  $G^C$  and  $\delta G^C$ ; (see Section 6.3)
2 Generate an event queue  $\mathcal{Q}$ ; (see Section 6.4.1)
3 while  $\mathcal{Q} \neq \emptyset$  do
4      $\mathbf{q} \leftarrow \text{dequeue}(\mathcal{Q})$ ;
5      $\delta G^C \leftarrow \text{Procedure 1 } (\delta G^C, \mathbf{q})$ ;
6      $G^C \leftarrow G^C \cup \delta G^C$ ;
```

6.5. Incremental 2D DRT graph construction for degenerate cases

As real data with tipping curves have degeneracies, we now discuss how to deal with such cases. The algorithm for constructing a 2D DRT graph with degeneracies is similar to Algorithm 1 except for two modifications. The first of these –in step 2– consists of detecting and sorting degenerate event points to generate an event queue \mathcal{Q} (see Section 6.5.1). The second modification –in step 5– is, despite dealing with a pair of tipping curves in simple cases, dealing with a family of tipping curves; thus the elementary step needs to be modified (see Section 6.5.2).

6.5.1. Detecting and sorting event points

According to their natures, the degeneracies can be classified into the following three cases, illustrated in Figure 12. More than two tipping curves can be:

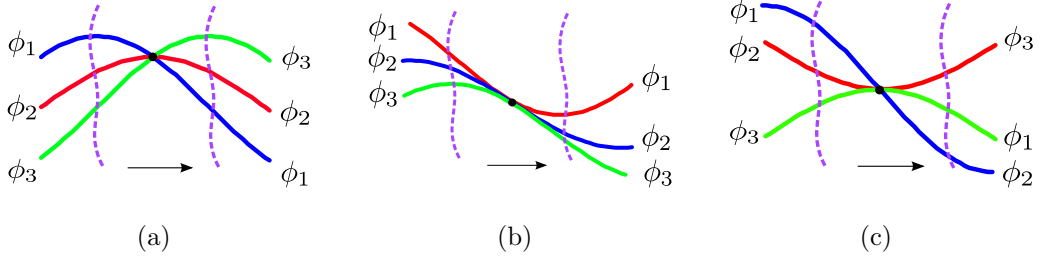


Figure 12: Degenerate cases: multiple intersection (a), multiple tangent point (b) and multiple mixed point (c).

- (i) intersecting at a single point (*multiple intersection*);
- (ii) tangent at a single point (*multiple tangent point*);
- (iii) tangent and/or intersecting at a single point (*multiple mixed point*).

In fact, those degeneracies can be detected by using Property 2 (iii)(iv) and Corollary 1 as mentioned in Section 6.4.1. More precisely, comparing the intersections allows us to detect degenerate event points since they have the same coordinates⁴ at the point. Such a degenerate event point is now represented by a family of tipping curves that go through the intersection point.

6.5.2. Elementary step for degenerate event points

Based on the classification of degenerate event points, we can handle each case explicitly. For a multiple intersection, we swap the order of all intersecting tipping curves in the cut γ before and after this event point. A multiple tangent is not considered as an event point because there is no change of γ around this point. The last case, a multiple mixed point, is

⁴Note that we do not know the exact value of θ but the values $\sin \theta$ and $\cos \theta$.

more complicated. Observing carefully Figure 12, we remark that tipping curves are decomposed into sets sorted by tangent values; each set contains tipping curves with the same tangent. When γ passes this point, only the order of these sets of curves are reversed while the order of curves in each set is preserved. In fact, this provides the general procedure for multiple intersection cases in which each tipping curve is seen as a set of equal tangent curves.

At each event point, the elementary step consists of modifying the partial graph δG^C according to the change of γ in a similar way to Procedure 1. Note that in degenerate cases, each event point contains a family of tipping curves instead of a pair as in simple cases. Let \mathbf{q} be an event point, then $\mathbf{q} = \{\tau_0, \tau_1, \dots, \tau_m\}$ is a family of tipping curves where each $\tau_j = \{\phi_{j1}, \phi_{j2}, \dots\}$ is a set of tipping curves with the same tangent at \mathbf{q} . Let δG_i^C and δG_{i+1}^C be respectively the partial graph with respect to γ_i and γ_{i+1} , which go through on the left and right of the event point \mathbf{q} . The construction of δG_{i+1}^C from δG_i^C proceeds following these steps:

1. Generating two lists of tipping curves which give the order of tipping curves before and after \mathbf{q} and storing them into two Last-In First-Out (LIFO) stacks \mathbf{S}_1 and \mathbf{S}_2 respectively. Details are given below.
2. Finding the initial and terminal vertices u and v for an event point \mathbf{q} , between which δG_i^C changes.
3. Finding the current vertices and edges between u and v .
4. Replacing the current vertices and edges by the new vertices and edges between u and v .

While creating new edges in the last step, each edge is given a label of a

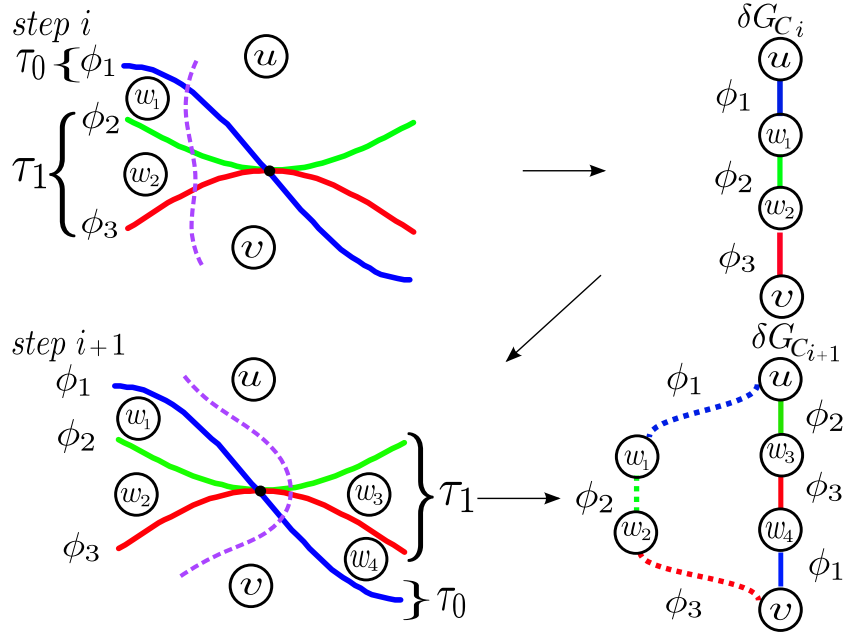


Figure 13: Illustrating the progress of a cut by which the partial graph δG_{i+1}^C is modified from δG_i^C in a degenerate case.

tipping curve taken from \mathbf{S}_2 .

We now explain how to fill two LIFO stacks \mathbf{S}_1 and \mathbf{S}_2 according to the order of tipping curves before and after the event point \mathbf{q} . For this, we first need to sort the sets τ_j and the tipping curves ϕ in each set τ_j of \mathbf{q} with the order obtained from δE_i^C . After sorting, we assume that $\mathbf{q} = (\tau_0, \tau_1, \dots, \tau_m)$ and each $\tau_j = (\phi_{j1}, \phi_{j2}, \dots)$ with respect to δE_i^C . The stacks \mathbf{S}_1 and \mathbf{S}_2 are generated by using the reverse sequences of τ_j , which is $\bar{\tau}_j = (\dots, \phi_{j2}, \phi_{j1})$, as follows.

```

for  $j = 0 \rightarrow m$  do
  for each  $\phi \in \bar{\tau}_{m-j}$  do
     $\mathbf{S}_1 \leftarrow push(\phi);$ 

```

```

end for
for each  $\phi \in \bar{\tau}_j$  do
     $\mathbf{S}_2 \leftarrow push(\phi);$ 
end for
end for

```

Note that in \mathbf{S}_1 , tipping curves have the same order as in δE_i^C , and in \mathbf{S}_2 the order of sets of tipping curves are reversed while the order of curves in each set is preserved. Procedure 2 generates δG_{i+1}^C from δG_i^C at the event point \mathbf{q} .

7. Construction of a discrete rigid transformation graph

In this section, we present an algorithm to construct a DRT graph from a set of tipping surfaces. The basic idea is quite similar to the algorithm for constructing a 2D DRT graph, which is based on sweeping a cut and updating the DRT graph G at event points. At each elementary step, we build an additional part ΔG and then integrate it into G .

7.1. Principles of incremental DRT graph construction

As explained before, the graph of the parameter space can be constructed from its projections into the planes (a, θ) and (b, θ) . We denote $G^a = (V^a, E^a)$ (resp. $G^b = (V^b, E^b)$) the projection DRT graph into the planes (a, θ) (resp. (b, θ)). From Formulae (13) and (14) of tipping curves, we derive the following proposition:

Proposition 2. *Let G^a (resp. G^b) be a 2D DRT graph constructed from a set of tipping curves F_ϕ (resp. F_ψ). G^a and G^b are isomorphic, denoted by*

Procedure 2: Elementary step for degenerate cases.

Input: A partial graph δG_i^C and an event point \mathbf{q} .

Output: A partial graph δG_{i+1}^C .

```

1  Fill  $\mathbf{S}_1$  and  $\mathbf{S}_2$  as explained above.
2   $\phi_u \leftarrow pop(\mathbf{S}_1)$ ;  $\phi_v \leftarrow pop(\mathbf{S}_1)$ ;
3   $e_u \leftarrow \varepsilon(\phi_u)$ ;  $e_v \leftarrow \varepsilon(\phi_v)$ ;
4   $\{u\} \leftarrow \vartheta(e_u) \setminus \vartheta(e_u) \cap \vartheta(e_v)$ ; //  $u$  is the initial vertex
5   $\mathbf{S}_1 \leftarrow push(\phi_v)$ ;  $\mathbf{S}_1 \leftarrow push(\phi_u)$ ;
6   $\phi_u \leftarrow pop(\mathbf{S}_2)$ ;  $\phi_v \leftarrow pop(\mathbf{S}_2)$ ;
7   $e_u \leftarrow \varepsilon(\phi_u)$ ;  $e_v \leftarrow \varepsilon(\phi_v)$ ;
8   $\{v\} \leftarrow \vartheta(e_u) \setminus \vartheta(e_v) \cap \vartheta(e_v)$ ; //  $v$  is the terminal vertex
9   $\mathbf{S}_2 \leftarrow push(\phi_v)$ ;  $\mathbf{S}_2 \leftarrow push(\phi_u)$ ;
10 while  $\mathbf{S}_1 \neq \emptyset$  do
11      $\phi_u \leftarrow pop(\mathbf{S}_1)$ ;  $\phi_v \leftarrow pop(\mathbf{S}_1)$ ;
12      $e_u \leftarrow \varepsilon(\phi_u)$ ;  $e_v \leftarrow \varepsilon(\phi_v)$ ;
13      $\{w\} \leftarrow \vartheta(e_u) \cap \vartheta(e_v)$ ;
14      $\{w_u\} \leftarrow \vartheta(e_u) \setminus \{w\}$ ;
15      $\Delta V_-^C \leftarrow \Delta V_-^C \cup \{w\}$ ; //  $w$  is a removed vertex
16      $\Delta E_-^C \leftarrow \Delta E_-^C \cup \{(w_u, w, \phi_u)\}$ ;
17     if  $\mathbf{S}_1 \neq \emptyset$  then  $\mathbf{S}_1 \leftarrow push(\phi_v)$ ;
18  $j \leftarrow 0$ ;
19  $\phi \leftarrow pop(\mathbf{S}_2)$ ;
20  $\Delta V_+^C \leftarrow \{w_j\}$ ; //  $w_j$  is a new vertex
21  $\Delta E_+^C \leftarrow \{(u, w_j, \phi)\}$ ;
22 while  $\mathbf{S}_2 \neq \emptyset$  do
23      $j \leftarrow j + 1$ ;
24      $\phi \leftarrow pop(\mathbf{S}_2)$ ;
25      $\Delta V_+^C \leftarrow \Delta V_+^C \cup \{w_j\}$ ; //  $w_j$  is a new vertex
26     if  $\mathbf{S}_2 \neq \emptyset$  then  $\Delta E_+^C \leftarrow \Delta E_+^C \cup \{(w_{j-1}, w_j, \phi)\}$ ;
27     else  $\Delta E_+^C \leftarrow \Delta E_+^C \cup \{(w_j, v, \phi)\}$ ;
28  $\delta G_{i+1}^C \leftarrow \delta G_i^C \setminus \Delta G_-^C \cup \Delta G_+^C$ ;

```

$$G^a \sim G^b.$$

Proof Translating Formula (14) by $\frac{\pi}{2}$ with respect to θ , we obtain a set of tipping curves that corresponds to the set of Formula (13), so that there exists exactly one correspondence between ϕ_{pqk} and ψ_{pql} :

$$\begin{aligned} b(\theta + \frac{\pi}{2}) &= l + \frac{1}{2} - p \sin(\theta + \frac{\pi}{2}) - q \cos(\theta + \frac{\pi}{2}) \\ &= l + \frac{1}{2} - p \cos \theta + q \sin \theta = a(\theta). \end{aligned}$$

As the sets F_ϕ and F_ψ are periodic with period $\frac{\pi}{2}$ (Property 4), the 2D DRT graphs in the two planes (a, θ) and (b, θ) are isomorphic. \blacksquare

Since $G^a \sim G^b$, we need to construct only one graph, G^a for example, and by the correspondence of tipping curves between ϕ_{pqk} and ψ_{pql} we can induce the other. The proof of Proposition 2 implies the following lemma.

Lemma 7.1. *Let $\mathcal{E}_a \subset \mathbb{R}^2$ (resp. $\mathcal{E}_b \subset \mathbb{R}^2$) be the set of event points for the set of tipping curves F_ϕ (resp. F_ψ). We have $\mathcal{E}_a = \mathcal{E}_b$.*

If \mathcal{Q}_a (resp. \mathcal{Q}_b) denotes the event queue corresponding to \mathcal{E}_a (resp. \mathcal{E}_b), then $|\mathcal{Q}_a| = |\mathcal{Q}_b|$. Note that we store an event point as a list of tipping curves which generates this event point, but not as its coordinates; thus $\mathcal{Q}_a \neq \mathcal{Q}_b$ even if their event points have the same coordinates.

In order to construct a DRT graph G we use two vertical cuts, each of which sweeps the plane either (a, θ) or (b, θ) respectively. Thus an elementary step is performed for each pair of event points on the two planes to generate the additional part ΔG of G by combining the partial graph δG^a and δG^b of G^a and G^b respectively. The construction of δG^a and δG^b was described in Section 6.

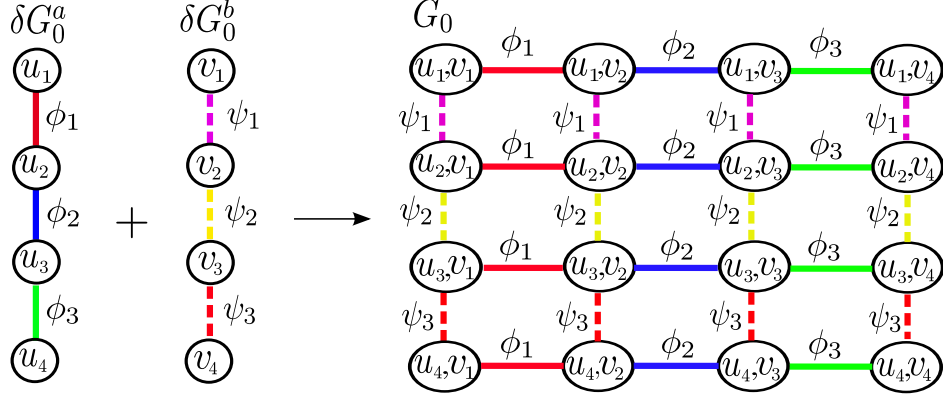


Figure 14: Illustration of construction of an initial DRT graph G_0 from two 2D DRT partial graphs δG_0^a and δG_0^b on the planes (a, θ) and (b, θ) .

7.2. Initial graph construction

The initial DRT graph $G_0 = (V_0, E_0)$ is generated from $\delta G_0^a = (\delta V_0^a, \delta E_0^a)$ and $\delta G_0^b = (\delta V_0^b, \delta E_0^b)$ as follows:

- $V_0 = \{(v_a, v_b) \mid v_a \in \delta V_0^a, v_b \in \delta V_0^b\};$
- $E_0 = \{((u_1, v), (u_2, v), \phi_u) \mid u_1, u_2 \in \delta V_0^a, v \in \delta V_0^b, (u_1, u_2, \phi_u) \in \delta E_0^a\} \cup \{((u, v_1), (u, v_2), \phi_v) \mid v_1, v_2 \in \delta V_0^b, u \in \delta V_0^a, (v_1, v_2, \phi_v) \in \delta E_0^b\}.$

Therefore G_0 contains $(n+1)^2$ vertices and $2n(n+1)$ edges, where n is the number of tipping curves of each of F_ϕ and F_ψ (see Figure 14).

7.3. Elementary step

The additional part ΔG_i at each elementary step is constructed by Procedure 3, from δG_i^a and δG_i^b in a similar way to the initialization. The difference being that a new vertex is generated from a pair of a new vertex in δV_i^a and one of the vertices in δV_i^b or vice-versa. Thus a new edge connects two vertices sharing an edge in δE_i^a and δE_i^b . Procedure 3 requires

Procedure 3: Elementary step for DRT graph construction.

Input: Two 2D partial graphs δG_i^a and δG_i^b and their additional vertex sets ΔV_+^a and ΔV_+^b .

Output: The additional part ΔG_i .

```

1 Initialize  $\Delta G_i = \emptyset$ , with  $\Delta V_i = \emptyset$  and  $\Delta E_i = \emptyset$ ;
2 foreach  $u \in \Delta V_+^a$  and  $v \in \delta V_i^b$  do
3    $\Delta V_i \leftarrow \Delta V_i \cup \{(u, v)\}$ ;
4 foreach  $u \in \delta V_i^a$  and  $v \in \Delta V_+^b$  do
5    $\Delta V_i \leftarrow \Delta V_i \cup \{(u, v)\}$ ;
6 foreach  $e_u = (u_1, u_2, \phi_u) \in \delta E_i^a$  do
7   foreach  $v \in \Delta V_+^b$  do
8      $\Delta E_i \leftarrow \Delta E_i \cup \{((u_1, v), (u_2, v), \phi_u)\}$ ;
9 foreach  $e_v = (v_1, v_2, \phi_v) \in \delta E_i^b$  do
10  foreach  $u \in \Delta V_+^a$  do
11     $\Delta E_i \leftarrow \Delta E_i \cup \{((u, v_1), (u, v_2), \phi_v)\}$ ;

```

the additional parts ΔV_+^a and ΔV_+^b of δG_i^a and δG_i^b respectively. We modify Procedure 2 and obtain Procedure 2', such that it returns not only a partial graph δG_{i+1}^C but also an adding vertex set ΔV_{+1}^C , which is already calculated in Procedure 2 as an intermediate parameter.

7.4. Algorithm

The final algorithm (Algorithm 2) builds a DRT graph G by taking two event points $\mathbf{q}_a \in \mathcal{Q}_a$, $\mathbf{q}_b \in \mathcal{Q}_b$, whose coordinates are identical. At each iteration, we generate ΔG simply by using Procedure 3 and then integrates

Algorithm 2: Construction of DRT graph.

Input: A set of tipping surfaces, *i.e.*, two sets of tipping curves.

Output: A DRT graph $G = (V, E)$.

```
1 Initialize  $\delta G^a$  and  $\delta G^b$ ; (see Section 6.3)
2 Initialize  $G = G_0$  from  $\delta G_0^a$  and  $\delta G_0^b$ ; (see Section 7.2)
3 Generate  $\mathcal{Q}_a$  and  $\mathcal{Q}_b$ ; (see Section 6.5.1)
4 while  $\mathcal{Q}_a \neq \emptyset$  and  $\mathcal{Q}_b \neq \emptyset$  do
5      $\mathbf{q}_a \leftarrow \text{dequeue}(\mathcal{Q}_a)$ ;
6      $\{\delta G^a, \Delta G_+^a\} \leftarrow \text{Procedure 2'} (\delta G^a, \mathbf{q}_a)$ ;
7      $\mathbf{q}_b \leftarrow \text{dequeue}(\mathcal{Q}_b)$ ;
8      $\{\delta G^b, \Delta G_+^b\} \leftarrow \text{Procedure 2'} (\delta G^b, \mathbf{q}_b)$ ;
9      $\Delta G \leftarrow \text{Procedure 3 } (\delta G^a, \delta G^b, \Delta G_+^a, \Delta G_+^b)$ ;
10     $G \leftarrow G \cup \Delta G$ ;
```

it into G .

8. Complexity analysis and experiments

8.1. Space complexity of DRT graph

The complexity of a 2D DRT graph, *i.e.*, the numbers of its vertices and edges, is obtained by counting the number of event points.

Proposition 3. *Given a set C of n tipping curves,*

- (i) *the number of event points is at most $n(n - 1)$;*

and the generated 2D DRT graph G^C has

- (ii) *at most $n^2 + 1$ vertices;*

(iii) *at most $2n^2 - n$ edges.*

Proof (i) The number of event points is the number of intersections of two curves in C . Since two tipping curves meet at most in two points (Corollary 2), the number of event points is less than or equal to $2\binom{n}{2} = n(n-1)$.
(ii) The vertices of G^C correspond to the faces of the arrangement of tipping curves. If $n = 1$, the number of faces is $2 = 1^2 + 1$, the curve forms the boundary between two faces. Let us now assume that there are $(n-1)^2 + 1$ faces for $n-1$ tipping curves. When adding the n -th curve, this curve will be divided into at most $2(n-1) + 1$ arcs by the $n-1$ other curves, and each of these arcs will split at most one face into two. Therefore, at most $2(n-1) + 1$ new faces will be created. Thus, the total number of faces (*i.e.*, that of vertices in the 2D DRT graph) is at most $(n-1)^2 + 1 + 2(n-1) + 1 = n^2 + 1$. The result follows by induction. (iii) If $n = 1$, there is one curve and thus we obtain $1 = 2 \cdot 1^2 - 1$ edge. Let us now assume that for $n-1$ curves there are at most $2(n-1)^2 - (n-1)$ edges. When adding the n -th curve, this curve will intersect at most the $n-1$ previous curves. Since there are at most two intersections for each one, this creates at most $2(n-1)$ edges. Moreover, the n -th curve itself will create at most $2(n-1) + 1$ new edges as it has intersected at most $2(n-1)$ points. Thus, the total number of created edges is at most $2(n-1)^2 - (n-1) + 2(n-1) + 2(n-1) + 1 = 2n^2 - n$. The result follows by induction. ■

The following property allows us to study the complexity of F_ϕ for a digital image I with finite size. This property induces the fact that the number of possible DRTs is finite and bounded by the size of I . We say two tipping curves are *vertically offset* if one can be obtained by translating the

other vertically. In fact, two vertically offset curves is a special case of two tipping curves with no intersection, which corresponds to (ii) in Property 2.

Property 5. *Given a digital image I of size $N \times N$, the family F_ϕ has*

- (i) $N^2(N + 1)$ tipping curves,
- (ii) N^2 sets of vertical offset tipping curves,
- (iii) $N + 1$ vertically offset tipping curves in each set, and
- (iv) $2N$ intersections at $\theta = \frac{\pi}{2}d$ for $d \in \mathbb{Z}$.

Proof (i) The number of tipping curves ϕ_{pqk} is simply the possible combinations of integer triplets (p, q, k) . Since $0 \leq p, q \leq N - 1$ and $0 \leq k \leq N$, there are $N^2(N + 1)$ tipping curves. (ii),(iii) From (ii) of Property 2, two trigonometric curves $\phi_{p_i q_i k_i}$ and $\phi_{p_j q_j k_j}$ are vertically offset if and only if they have $p_i = p_j$, $q_i = q_j$ and $k_i \neq k_j$. We thus obtain N^2 sets of vertically offset tipping curves, such that each set contains $N + 1$ curves with different values of k . (iv) Thanks to Property 4, we only need to evaluate Formula (15) at $\theta = 0$ and we have $a(0) = k + \frac{1}{2} + q$. The number of intersections is the number of different sums of q and k , which is $2N$. ■

In practice, the various numbers of event points, vertices and edges are all lower than their upper-bound $\mathcal{O}(n^2)$, due to the degenerated cases in the arrangement of n tipping curves. From Property 5, we know that $n = \mathcal{O}(N^3)$ for an image of size $N \times N$. Then, those complexities such as the numbers of event point, vertices and edges can be re-expressed as $\mathcal{O}(N^6)$.

As mentioned in Section 7, the construction of a DRT graph G is obtained from its projections on the planes (a, θ) and (b, θ) using two cuts. Then we notice that the initial graph has a complexity $\mathcal{O}(N^3) \times \mathcal{O}(N^3)$. We also

know that at each elementary step, there are $\mathcal{O}(N^3)$ vertices generated. As the number of event points is $\mathcal{O}(N^6)$, in total there are $\mathcal{O}(N^6) \times \mathcal{O}(N^3)$ vertices added in G . This justifies the following theorem.

Theorem 1. *The DRT graph G associated to an image of size $N \times N$ has a space complexity of $\mathcal{O}(N^9)$.*

8.2. Run-time complexity of construction algorithm

First, in Step 1 of Algorithm 2, the partial graphs δG_0^a and δG_0^b are initialized with a time complexity of $\mathcal{O}(N^3 \log N)$, when tipping curves are sorted as explained in Section 6.3. Note that the number of tipping curves is $\mathcal{O}(N^3)$. Then in Step 2, since each of δG_0^a and δG_0^b has $\mathcal{O}(N^3)$ vertices and edges, from section 7.2 we can generate G_0 in $\mathcal{O}(N^6)$. For Step 3, firstly a $\mathcal{O}(N^6)$ time complexity is needed to detect all the intersections of tipping curves, Secondly, from Proposition 3, we know that the total number of event points is $\mathcal{O}(N^6)$, thus sorting the event points takes $\mathcal{O}(N^6 \log N)$. As explained in Section 6.4.1, to compare any two event points we use their corresponding continued fractions. Practically, this comparison is executed in constant time [20] so that this does not increase the time complexity of sorting event points. The sweep of a cut, in Steps 4 to 10, has $\mathcal{O}(N^6)$ iterations, for each of which Procedure 2' and Procedure 3 are executed in $\mathcal{O}(N^3)$ operations respectively. Finally, the DRT graph is constructed in $\mathcal{O}(N^9)$ time.

8.3. Experiments

We have implemented our algorithm in C++. From the experiments, the numbers of vertices (and edges) of the DRT graphs have been computed

	2D DRT Graph		DRT Graph	
N	Vertices	Edges	Vertices	Edges
1	1	0	1	0
2	49	144	1 033	5 040
3	431	1 472	29 631	160 512
4	2 277	8 144	357 421	1 993 696
5	8 371	3 0304	2 487 053	13 978 176
6	25 033	92 176	12 550 225	71 310 320
7	62 199	229 184	48 604 267	276 284 416
8	139 661	518 096	160 554 101	916 648 928
9	282 731	1 049 344	457 270 393	2 612 082 816

Table 2: Numbers of DRT graph vertices and edges with respect to image sizes of $N \times N$.

for images of sizes varying from 1×1 to 9×9 . The experiments were carried out on a personal computer equipped with a processor 3.0GHz Intel® Core™ 2 Duo and 4GB of memory. The results, shown in Table 2 and Figure 15, validate the theoretical $\mathcal{O}(N^9)$ space complexity stated in the previous theorem.

9. Applications

This section illustrates the practical applicability of our proposed combinatorial structure for DRTs.

9.1. Generation of all digital transformed images

Given a digital image I of size $N \times N$, we firstly show how to generate all possible transformed images of I under rigid transformations. We know that

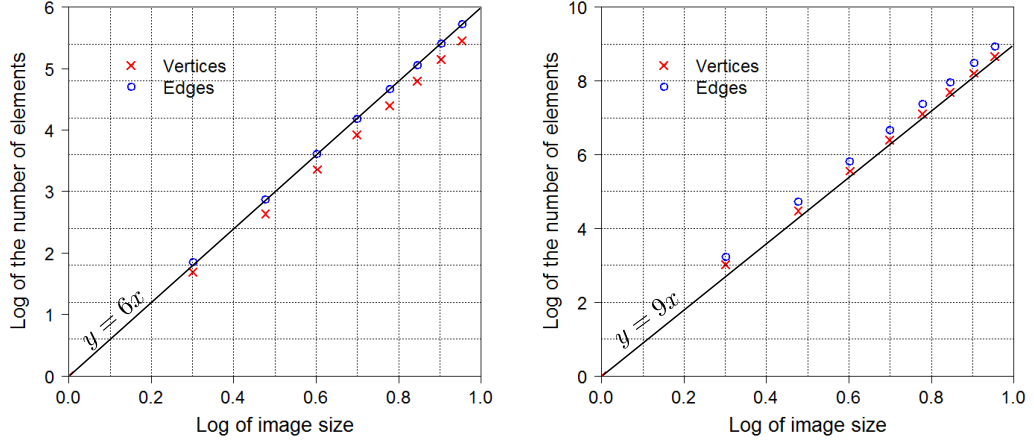


Figure 15: The relation between image size and number of elements, (*i.e.*, vertices or edges) in Table 2 for 2D DRT graphs (left) and DRT graphs (right).

the DRT graph models the whole space of digital rigid transformations on any subset of \mathbb{Z}^2 of size $N \times N$. It should be mentioned that this graph does not contain any geometric parameters (a, b, θ) for the rigid transformations but only the topological information, which gives the relationship between any neighbouring transformed images. Indeed, each edge of the DRT graph is labelled by an integer quadruple (p, q, k, i) indicating that the pixel $(p, q) \in \mathbb{Z}^2$ will cross the half-grid line, either $x = k + \frac{1}{2}$ if $i = 0$ or $y = k + \frac{1}{2}$ if $i = 1$. Using this information, when we move from one vertex—a transformed image—to its neighbour, only one pixel (p, q) changes. Thus, we can provide incrementally all possible images under any rigid transformations of I . Figure 16 shows all generated digital transformed images from a part of the DRT graph for an image of size 3×3 where $a \in]-\frac{1}{2}, \frac{1}{2}[$, $b \in]-\frac{1}{2}, \frac{1}{2}[$ and $\theta \in]0, \frac{\pi}{4}[$.

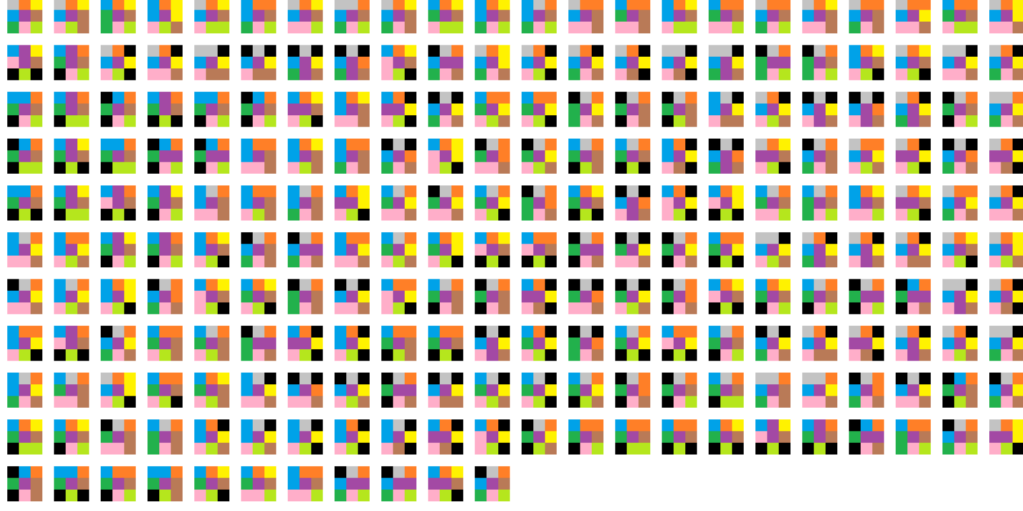


Figure 16: A set of 231 images generated from the part of a DRT graph for a given image of size 3×3 where $a \in]-\frac{1}{2}, \frac{1}{2}[$, $b \in]-\frac{1}{2}, \frac{1}{2}[$ and $\theta \in]0, \frac{\pi}{4}[$. The first image is the identity transformation.

9.2. Discrete representation of the transformation of an image: discrete transition path

Given an image I and its transformed image I' , we define a *transition path* between I and I' as the (possibly non unique) locus of rigid transformations transitioning I to I' in the parameter space (a, b, θ) . By using the DRT framework, such a transition path can be represented as a connected path in the DRT graph G , called a *discrete transition path*. Indeed, we can find a discrete transition path in G between I and I' using the topological information of G providing the neighbouring relations of DRTs. Figure 17 shows a discrete transition path between two vertices of G and the sequence of transformed images corresponding to the vertices through the path. This representation of such a sequence illustrates the fact that only one pixel is

changed between two incident transformed images. This allows us to define a metric on the DRT graph for measuring similarity between different transformed images, simply by counting the number of pixel changes, *i.e.*, the length of the path.

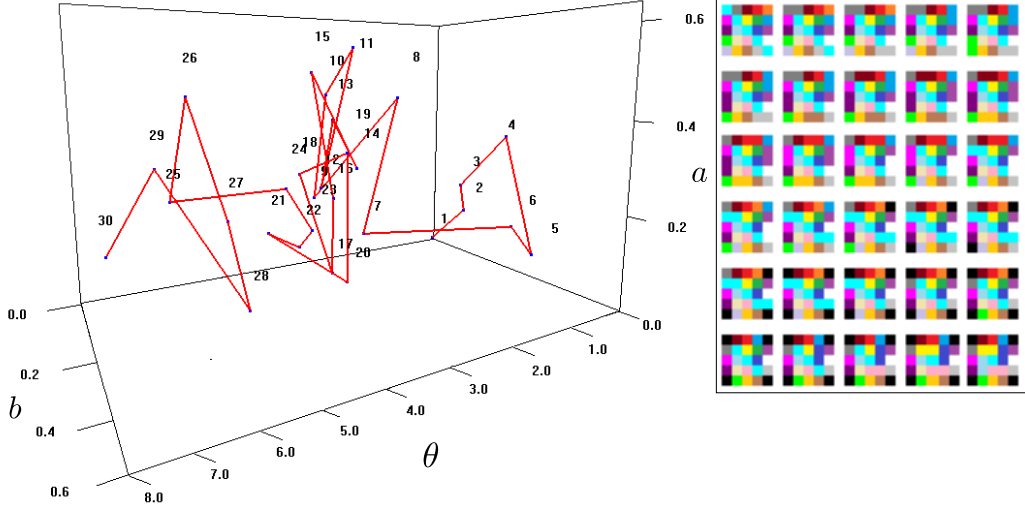


Figure 17: A discrete transition path of the DRT graph in the parameter space (a, b, θ) (left) and a sequence of transformed images (right). Each vertex of the path represents a DRT which gives a digital transformed image. The first image corresponds to the identity transformation and the images from left to right and from top to bottom correspond to the vertices ordered in the path.

9.3. Evaluation of discrete transition paths

Given two DRT graph vertices, it is observed in Figure 18 that their discrete transition paths are not unique. Assuming the existence of several different transition paths, we would like to find the best path which preserves some additional criteria such as topology and geometry of our object of interest in images. We know that for any two consecutive images in a discrete

transition path only one pixel changes and thus it may be easy to verify our criteria incrementally along the path by checking only this pixel. By using the DRT framework, it may be possible to propose a new evaluation strategy for discrete transition paths.

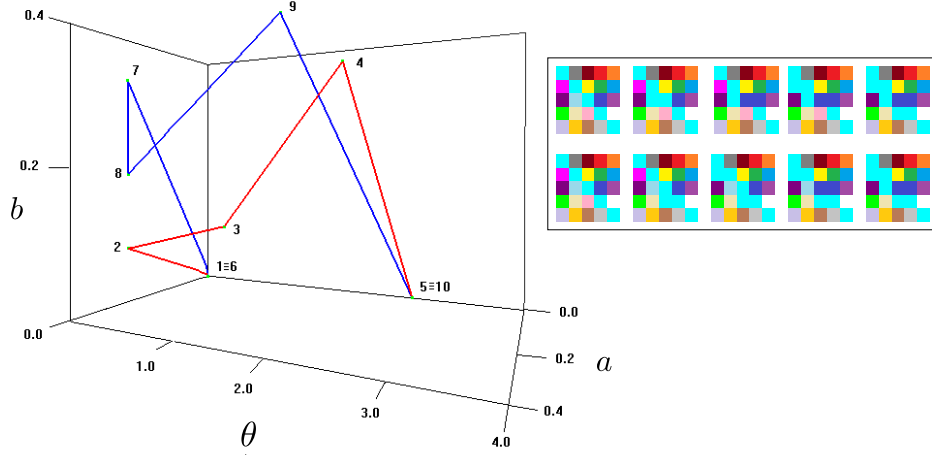


Figure 18: Illustration of two different discrete transition paths with the same starting and ending vertices in a DRT graph: The two paths (left) and their sequences of images (right). The first row images correspond to the red path and the bottom ones correspond to the blue path.

10. Conclusion

In this article, we have introduced a combinatorial structure represented as a graph for modelling the parameter space of digital rigid transformations. This graph consists of finite sets of vertices and edges. In this graph, each vertex represents a digital transformed image, and each edge linking two vertices represents a transition changing only one pixel between two transformed images. This structure presents a space complexity of $\mathcal{O}(N^9)$, where $N \times N$

is the size of any considered subspace of \mathbb{Z}^2 . An algorithm has also been proposed in order to define this structure in linear time with respect to this space complexity.

Experiments performed on a standard computer emphasize both the correctness of the algorithm, and the estimated time/space polynomial complexities. Due to these complexities, it remains however hardly tractable to compute the proposed combinatorial structure for large images.

However, this size limitation is not a crippling default in the case of several applications. Indeed, image processing techniques based on sub-image/sample analysis can take advantage of the proposed approach, *e.g.*, in the context of pattern matching, non-local image processing [21], or marker-based registration [22].

From a methodological point of view, further work will now involve studying ways to use the proposed combinatorial structure in multiscale strategies, most importantly in order to process large images without computing the whole data structure. Furthermore, for image registration, most existing methods [2] provide no guarantee to find a global optimal solution in general. With our approach, we may define a new graph based metric using neighbouring relations between discrete rigid transformations, which may lead to a global optimal solution. From a theoretical point of view, extensions of the presented results to 3D digital images (following some connected works related to 3D pattern matching [23]) will also be investigated.

Appendix A. Exact comparison of quadratic irrationals

This appendix describes an algorithmic process enabling to compare two quadratic irrationals without numerical approximation. In [19, 24], it was proved that a quadratic irrational can be rewritten as a periodic continued fraction. More formally, for any quadratic irrational $Q = \frac{p+\sqrt{q}}{r}$ where $p, q, r \in \mathbb{Z}$, $q > 0$, $r \neq 0$ we have a periodic continued fraction:

$$Q = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_n + \frac{1}{a_1 + \frac{1}{a_2 + \dots}}}}}} \quad , \quad (\text{A.1})$$

where $a_0 \in \mathbb{Z}$ and $a_i \in \mathbb{N}$ for $i = 1 \dots n$. Such a periodic continued fraction is unambiguously modelled by a finite sequence of integers (a_0, a_1, \dots, a_n) , denoted by $[a_0; a_1, a_2, \dots, a_n]$. Based on this formulation, the comparison between two quadratic irrationals can be performed as follows.

Given $Q_1 = \frac{p_1 + \sqrt{q_1}}{r_1} = [a_0; a_1, a_2, \dots]$ and $Q_2 = \frac{p_2 + \sqrt{q_2}}{r_2} = [b_0; b_1, b_2, \dots]$,

let $k \in \mathbb{N}$ be the smallest index for which $a_k \neq b_k$. If $Q_1 \neq Q_2$ (the equality can be easily checked by comparing the values p_i, q_i and r_i), the order between Q_1 and Q_2 is characterised by the sign of the value $E = (-1)^k(a_k - b_k)$. In particular, we have $Q_1 < Q_2$ (resp. $Q_1 > Q_2$) if $E < 0$ (resp. $E > 0$) (see Algorithm 3). Note that for a quadratic irrational $Q = \frac{p+\sqrt{q}}{r}$, it is proved

Algorithm 3: Comparison of two quadratic irrationals

Input: $(p_1, q_1, r_1), (p_2, q_2, r_2)$ representing two quadratic irrationals Q_1 and Q_2 .

Output: Value in $\{<, =, >\}$ denoting the relation between Q_1 and Q_2 .

if $(p_1, q_1, r_1) = (p_2, q_2, r_2)$ **then**
| return =;

else

| $E \leftarrow 0$;

| $(p_1^0, q_1^0, r_1^0, a_1^0) \leftarrow (p_1, q_1, r_1, \lfloor \frac{p_1 + \sqrt{q_1}}{r_1} \rfloor)$;

| $(p_2^0, q_2^0, r_2^0, a_2^0) \leftarrow (p_2, q_2, r_2, \lfloor \frac{p_2 + \sqrt{q_2}}{r_2} \rfloor)$;

| $i \leftarrow 1$;

| **while** $E = 0$ **do**

| | // calculate the term a_1^i of Q_1

| | $(p_1^i, r_1^i, a_1^i) \leftarrow (a_1^{i-1}r_1^{i-1} - p_1^{i-1}, \frac{q_1 - (p_1^i)^2}{r_1^{i-1}}, \lfloor \frac{p_1^i + \sqrt{q_1}}{r_1^i} \rfloor)$;

| | // calculate the term a_2^i of Q_2

| | $(p_2^i, r_2^i, a_2^i) \leftarrow (a_2^{i-1}r_2^{i-1} - p_2^{i-1}, \frac{q_2 - (p_2^i)^2}{r_2^{i-1}}, \lfloor \frac{p_2^i + \sqrt{q_2}}{r_2^i} \rfloor)$;

| | $E \leftarrow (-1)^{i-1}(a_1^i - a_2^i)$;

| | $i \leftarrow i + 1$;

| **if** $E > 0$ **then** return >;

| **else** return <;

in [25] that the length of repeating block for Q is $\mathcal{O}(\sqrt{q} \ln q)$. In the worst case, all terms of the repeating block are compared, *i.e.*, we have $\mathcal{O}(\sqrt{q} \ln q)$ comparisons. However, it is proved in [20] that the comparison of continued fractions has an average-case complexity of $\mathcal{O}(1)$.

References

- [1] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, *ACM Computing Surveys* 38 (4) (2006) 1–45.
- [2] B. Zitová, J. Flusser, Image registration methods: A survey, *Image and Vision Computing* 21 (11) (2003) 977–1000.
- [3] E. Andres, The quasi-shear rotation, in: *Discrete Geometry for Computer Imagery, DGCI, Proceedings*, Vol. 1176 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 307–314.
- [4] D. Coeurjolly, V. Blot, M.-A. Jacob-Da Col, Quasi-affine transformation in 3-D: Theory and algorithms, in: *International Workshop on Combinatorial Image Analysis, IWCIA, Proceedings*, *Lecture Notes in Computer Science*, Springer, 2009, pp. 68–81.
- [5] C. Hundt, M. Liśkiewicz, N. Ragnar, A combinatorial geometrical approach to two-dimensional robust pattern matching with scaling and rotation, *Theoretical Computer Science* 410 (51) (2009) 5317–5333.
- [6] C. Hundt, M. Liśkiewicz, On the complexity of affine image matching, in: *Symposium on Theoretical Aspects of Computer Science, STACS, Proceedings*, Vol. 4393 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 284–295.
- [7] C. Hundt, Affine image matching is uniform TC^0 -complete, in: *Combinatorial Pattern Matching, CPM, Proceedings*, Vol. 6129 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 13–25.

- [8] C. Hundt, M. Liśkiewicz, Combinatorial bounds and algorithmic aspects of image matching under projective transformations, in: Mathematical Foundations of Computer Science, MFCS, Proceedings, Vol. 5162 of Lecture Notes in Computer Science, Springer, 2008, pp. 395–406.
- [9] A. Amir, O. Kapah, D. Tsur, Faster two-dimensional pattern matching with rotations, *Theoretical Computer Science* 368 (3) (2006) 196–204.
- [10] Y. Thibault, Rotations in 2D and 3D discrete spaces, Ph.D. thesis, University Paris-Est (2010).
- [11] A. Amir, G. M. Landau, U. Vishkin, Efficient pattern matching with scaling, *Journal of Algorithms* 13 (1) (1992) 2–32.
- [12] A. Amir, A. Butman, M. Lewenstein, E. Porat, Real two dimensional scaled matching, *Algorithmica* 53 (3) (2009) 314–336.
- [13] V. A. Kovalevsky, Finite topology as applied to image analysis, *Computer Vision, Graphics & Image Processing* 46 (2) (1989) 141–161.
- [14] T. Y. Kong, A. Rosenfeld, Digital topology: Introduction and survey, *Computer Vision, Graphics & Image Processing* 48 (3) (1989) 357–393.
- [15] M. Sharir, Recent developments in the theory of arrangements of surfaces, in: Foundations of Software Technology and Theoretical Computer Science, FSTTCS, Proceedings, Vol. 1738 of Lecture Notes in Computer Science, Springer, 1999, pp. 1–21.
- [16] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel, M. Sharir,

Arrangements of curves in the plane-topology, combinatorics, and algorithms, Theoretical Computer Science 92 (2) (1992) 319–336.

- [17] J. Snoeyink, J. Hershberger, Sweeping arrangements of curves, in: Symposium on Computational Geometry, SOCG, Proceedings, ACM, 1989, pp. 354–363.
- [18] D. Halperin, Arrangements, in: J. E. Goodman, J. O’Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press LLC, 2004, Ch. 24, pp. 529–562.
- [19] K. H. Rosen, Elementary Number Theory and its Applications, 3rd Edition, Addison-Wesley, 1992.
- [20] P. Flajolet, B. Vallée, Continued fractions, comparison algorithms, and fine structure constants, Tech. Rep. 4072, INRIA (2000).
- [21] A. Buades, B. Coll, J. M. Morel, A review of image denoising algorithms with a new one, Multiscale Modeling & Simulation 4 (2) (2005) 490–530.
- [22] X. Pennec, N. Ayache, J.-P. Thirion, Landmark-based registration using features identified through differential geometry, in: I. N. Bankman (Ed.), Handbook of Medical Imaging, Academic Press, 2000, Ch. 31, pp. 499–513.
- [23] K. Fredriksson, E. Ukkonen, Combinatorial methods for approximate pattern matching under rotations and translations in 3D arrays, in: String Processing and Information Retrieval, SPIRE, Proceedings, IEEE, 2000, pp. 96–104.

- [24] A. Y. Khinchin, Continued Fractions, Dover Publications, 1964.
- [25] E. V. Podsypanin, Length of the period of a quadratic irrational, Journal of Mathematical Sciences 18 (6) (1982) 919–923.