Combinatorial structure of rigid transformations in 2D digital images

Phuc Ngo, Yukiko Kenmochi, Nicolas Passat, Hugues Talbot

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 such discrete images, rigid transformations are however usually performed in their associated continuous space, then requiring a subsequent digitization of the result. In this article, we propose to study rigid transformations of digital images as a fully discrete process. In particular, we investigate a combinatorial structure modelling the whole space of digital rigid transformations on any 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 build it in linear time with respect to this 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 a fully discrete process. In particular, several issues are considered, related to the number of possible rigid transformations given a finite subspace of $\mathbb{Z}^n$, the way to generate all of them, and the relations between all such transformations. Some combinatorial and algorithmic answers are provided to these questions, in the case of $\mathbb{Z}^2$.

The contributions of this article are twofold. The first –theoretical– one consists of the proposal of a combinatorial structure (namely a graph) modelling the whole space of digital rigid transformations on any 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 $\mathbb{Z}^2$. The 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 [3, 4], scalings [5, 6], combined scalings and rotations [7], affine transformations [8, 9], projective and linear transformations [10]). On the other hand, a new concept related to the topology of digital transformations is introduced in this structure. This is *neighbour* of transformations induced by the relations between the transformations.

Classes of transformations	Complexity
Rotations	$\mathcal{O}(N^3)$
Scalings	$\mathcal{O}(N^3)$
Rotations and scalings	$\mathcal{O}(N^6)$
Linear transformations	$\mathcal{O}(N^{12})$
Affine transformations	$\mathcal{O}(N^{18})$
Projective transformations	$\mathcal{O}(N^{24})$

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

The second –methodological– contribution is an efficient algorithm enabling to build 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, leading to ex-

act computation, and avoiding in particular any approximations related to floating point arithmetic.

The article is organised as follows. Sections 2 and 3 recall background notions and useful concepts. Section 4 introduces two notions: *tipping surfaces*, and *tipping curves*, which constitute the basis of the proposed combinatorial structure. Section 5 actually defines this structure, while Section 6 and 7 propose an algorithm for its construction. Complexity and experiments are discussed in Section 8 . Section 9 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, and in particular one of them is background and noted $\perp$. A *2D image* is a function $\mathcal{I} : \mathbb{R}^2 \to V$ such that $\mathcal{I}^{-1}(V \setminus \{\perp\})$ is finite. (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; etc.). We can suppose that $\mathcal{I}^{-1}(V \setminus \{\perp\}) \subseteq E = [-\frac{N}{2}, \frac{N}{2}]^2$ for a given $N \in \mathbb{N}$. The set E is called a *support* of $\mathcal{I}$ and N is the size of $\mathcal{I}$.

A *2D digital image* is defined as a function $I : \mathbb{Z}^2 \to V$ such that $I^{-1}(V \setminus \{\perp\})$ is finite. We can suppose that $I^{-1}(V \setminus \{\perp\}) \subseteq S = [-\frac{N}{2}, \frac{N}{2}]^2$ for a given $N \in \mathbb{N}$. The set S is called a *support* of I and N is the size of I.

In the sequel, by abusing of notation, we set $\mathcal{I} : E \to V$ and $I : S \to V$, in stead of $\mathcal{I} : \mathbb{R}^2 \to V$ and $I : \mathbb{Z}^2 \to V$.

The 2D digital image I is a numeric presentation of the 2D image $\mathcal{I}$ followed by a digitization process. In other words, the *digitization* consists of defining $I : S \to V$ associated to $\mathcal{I} : E \to V$, such that $S = E \cap \mathbb{Z}^2$. It relies on the partition of E into a regular grid (such as square, triangular, hexagonal grid, *etc.*) induced by the points of S.

The digitization process then associates each point of E to exactly one point of S. In this paper, we define the partition of E on a square grid. In such a case, the digitization process is defined *via* the following function

$$\left| \begin{array}{rcccc} D & : & E & \to & S \\ & & \boldsymbol{x} = (x, y) & \mapsto & \boldsymbol{p} = (p, q) \end{array} \right. \tag{1}$$

such that $\boldsymbol{x}$ and $\boldsymbol{p}$ verify

$$p = [x + \frac{1}{2}] \quad \text{and} \quad q = [y + \frac{1}{2}] \tag{2}$$

where [.] is a round function, which can be either the floor, ceiling, *etc.* All these round functions are equal for all the points which do not have semi-integer coordinates, and (possibly) differ on these last points.

We call the subset of E made of such points the *half-grid*, and we note this subset $\mathcal{H}$. More formally, we have

$$\mathcal{H} = E \cap \big(\mathbb{R} \times (\mathbb{Z} + \frac{1}{2})\big) \cup \big((\mathbb{Z} + \frac{1}{2}) \times \mathbb{R}\big). \tag{3}$$

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 set of boundaries of 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 [11, 12]. However, in the present context of digital geometry, the way the points of $\mathcal{H}$ are digitised 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 injective function $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$. Such a transformation induces, of course, a bijection between E and $\mathcal{T}(E) = \{\mathcal{T}(x) \mid x \in E\}$. A geometric transformation applied on an image $\mathcal{I} : E \to V$ will provide a new transformed image $\mathcal{T} \circ \mathcal{I} : E \to V$.

However it is not possible to apply directly $\mathcal{T}$ on a digital image I, since there is no guarantee that $\mathcal{T}(x) \in \mathbb{Z}^2$, for any $x \in S$. The handling of geometric transformations for digital images requires to define a digital function $T : \mathbb{Z}^2 \to \mathbb{Z}^2$, such that by applying T on $I : S \to V$ we obtain a new digital transformed image $T \circ I : S \to V$. This can be conveniently done by setting

$$T = D \circ \mathcal{T}, \tag{4}$$

i.e. T is obtained by embedding S in E, then applying $\mathcal{T}$, and digitizing the result by D (see Figure 1). Such a transformation T is called *2D digital image transformation*.

Remark 1. *There exist two models for image transformation. Given a transformation $\mathcal{T}$, the first one (Lagrangian model) consists of determining $\mathcal{T}(\boldsymbol{x})$ for any point $\boldsymbol{x}$ in the initial space, while the second one (Eulerian model)*

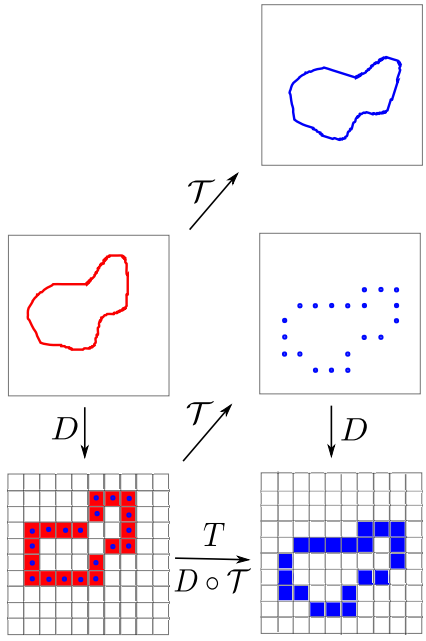

Figure 1: Geometric transformation and its digitalization.

consists of determining $\mathcal{T}^{-1}(\boldsymbol{y})$ *for any point* $\boldsymbol{y}$ *in the deformed space. Both models are equal for a transformation* $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$*, since* $\mathcal{T}$ *is bijective. In the digital case, these model are actually distinct, since* $D \circ \mathcal{T}$ *is not bijective, in general. In the current work, and without loss of generality, we focus on the Lagrangian model (see Figure 2). Note that the results shown in this paper are valid for any of the two models.*

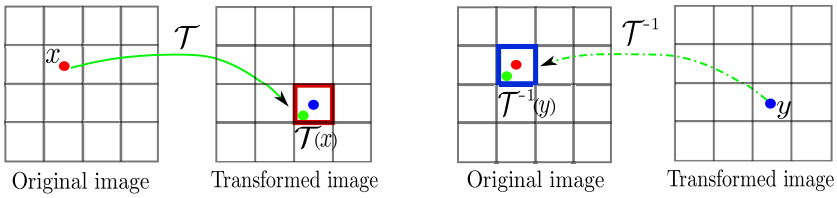

Figure 2: Two digital image transformation models: Lagrangian (left) and Eulerian (right) models.

3. Rigid transformations for digital images

3.1. Digital rigid transformation

From now on, we only consider *2D rigid transformations* which compose of translations and rotations. Formally, a rigid transformation $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$ transforms any point $\boldsymbol{p} = (p, q) \in \mathbb{R}^2$ into a point $\boldsymbol{r} = (r, s) \in \mathbb{R}^2$ according to the following relation

$$\mathcal{T}(\boldsymbol{p}) = \boldsymbol{r} = \begin{pmatrix} r \\ s \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} \tag{5}$$

where $a, b, \theta \in \mathbb{R}$ and $\theta \in [0, 2\pi[$.

In the case where $T : \mathbb{Z}^2 \to \mathbb{Z}^2$, from (1), (4) and (5), the transformed point $T(\boldsymbol{p}) = (p', q')$ associated to $\boldsymbol{p} \in \mathbb{Z}^2$ is then defined as

$$T(\boldsymbol{p}) = \begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p\cos\theta - q\sin\theta + a + \frac{1}{2}] \\ [p\sin\theta + q\cos\theta + b + \frac{1}{2}] \end{pmatrix}. \tag{6}$$

This transformation is called *2D digital rigid transformation*. Note that any rigid transformation (resp. digital rigid transformation) can be modelled 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 the rigid transformations (resp. digital rigid transformations)

$$\mathbb{T} = \{\mathcal{T}_{ab\theta} \mid (a, b, \theta) \in \mathbb{R}^3\} \tag{7}$$

$$\mathbb{T}_d = \{T_{ab\theta} \mid (a, b, \theta) \in \mathbb{R}^3\} \tag{8}$$

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 (5) and (6)):

$$\left| \begin{array}{rcccl} \mathfrak{T} & : & \mathbb{R}^3 & \to & \mathbb{T} \\ & & (a, b, \theta) & \mapsto & \mathfrak{T}\ (a, b, \theta) = \mathcal{T}_{ab\theta}, \end{array} \right. \tag{9}$$

$$\left| \begin{array}{rcccl} \mathfrak{T}_d & : & \mathbb{R}^3 & \to & \mathbb{T}_d \\ & & (a, b, \theta) & \mapsto & \mathfrak{T}_d(a, b, \theta) = T_{ab\theta}. \end{array} \right. \tag{10}$$

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 $(a_1, b_1, \theta_1), (a_2, b_2, \theta_2) \in \mathbb{R}^3$

$$\|(a_1, b_1, \theta_1) - (a_2, b_2, \theta_2)\| < \epsilon \Rightarrow \|\mathfrak{T}(a_1, b_1, \theta_1) - \mathfrak{T}(a_2, b_2, \theta_2)\| < \delta \tag{11}$$

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

Due to the digitization involved in the definition of T in (6), 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 $\boldsymbol{c} \in \mathbb{R}^3$ such that for any ball $\mathcal{B}(\boldsymbol{c}, \epsilon)$ of center c and radius $\epsilon > 0$, there exist $(a_1, b_1, \theta_1), (a_2, b_2, \theta_2) \in \mathcal{B}(\boldsymbol{c}, \epsilon)$ such that for $\delta = 1$,

$$\|(a_1, b_1, \theta_1) - (a_2, b_2, \theta_2)\| < \epsilon \Rightarrow \|\mathfrak{T}_d(a_1, b_1, \theta_1) - \mathfrak{T}_d(a_2, b_2, \theta_2)\| \geq \delta. \quad (12)$$

Such points $\boldsymbol{c} \in \mathbb{R}^3$ are critical points in the parameter space of the digital rigid transform. They characterise, 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*

$$\exists \boldsymbol{p} = (p, q) \in \mathbb{Z}^2, \mathcal{T}_{ab\theta}(\boldsymbol{p}) \in \mathcal{H}. \quad (13)$$

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

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

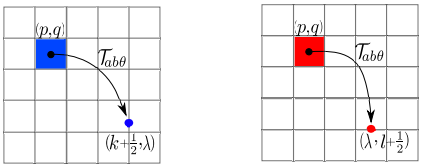

Figure 3: A critical transformation is a transformation of an integer point to either a vertical (left) and horizontal (right) half-grid point.

3.3. Discrete rigid transformations

As a result of the discontinuity of digital rigid transformation, it is possible that the digitalizations D for two rigid transformations of a digital image are identical. This leads to the following consideration about the equivalence transformations.

Definition 2. *Let $\mathbb{T}$ be the set of all rigid transformations for a given digital image I. Two elements $\mathcal{T}, \mathcal{T}' \in \mathbb{T}$ are considered* equivalent*, and we write $\mathcal{T} \sim \mathcal{T}'$, iff $D \circ \mathcal{T}(\boldsymbol{p}) = D \circ \mathcal{T}'(\boldsymbol{p})$ for $\forall \boldsymbol{p} \in I$.*

We define the *equivalence class* of $\mathcal{T} \in \mathbb{T}$ under the equivalence relation $\sim$, denoted by $[\mathcal{T}]_\sim$, as the set of all transformations in $\mathbb{T}$ that are equivalent to $\mathcal{T}$

$$[\mathcal{T}]_\sim = \{\mathcal{T}' \in \mathbb{T} : \mathcal{T}' \sim \mathcal{T}\}.$$

In this context, each equivalence class is a set of rigid transformations that provides the same digital transformed image.

Definition 3. *For a given digital image I, each equivalence class in $\mathbb{T}$ of I under $\sim$ is called a* discrete rigid transformation*(DRT).*

Please note that the term *discrete* thus refers to the non-continuous structure of transformations for digital images. By 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 necessary notions to study the decomposition of the parameter space into the DRTs. As we have explained in precedent section, the digital rigid transformations possess a discontinuous property, which is characterized by the critical transformations. The last transformations are actually related to the structure of DRTs in the parameter space, since they represent the boundaries of the equivalence classes. We will propose as follows two notions: *tipping surfaces* and *tipping curves* which in fact are analytical expression of critical rigid transformations.

4.1. Tipping surface

According to Definition 1, a critical rigid transformation $\mathcal{T}_{ab\theta}$ moves at least one integer point $\boldsymbol{p} = (p, q) \in \mathbb{Z}^2$ into a half-grid point which can be either 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 $\lambda \in \mathbb{R}$. The set of (a, b, θ) associated to the critical transformation forms in the parameter space a surface, called *tipping surfaces*. Note that any tipping surface can be represented by an integer triplet (p, q, k) or (p, q, l) corresponding to 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 a function Φ_{pqk} (resp. Ψ_{pql}) such that*

$$\left| \begin{array}{rccl} \Phi_{pqk}: & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ & (b,\theta) & \longmapsto & a = k + \frac{1}{2} + q\sin\theta - p\cos\theta, \end{array} \right. \tag{14}$$

$$\left| \begin{array}{rccl} \Psi_{pql}: & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ & (a,\theta) & \longmapsto & b = l + \frac{1}{2} - p\sin\theta - q\cos\theta. \end{array} \right. \tag{15}$$

The sets of all tipping surfaces defined by (14) and (15) correspond obviously to the discontinuity of digital rigid transformations. They divide then the parameter space (a, b, θ) into the equivalence classes, in each of which any rigid transformation gives the 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 finite. Figure 4 illustrates the parameter space of an image of size 3×3.

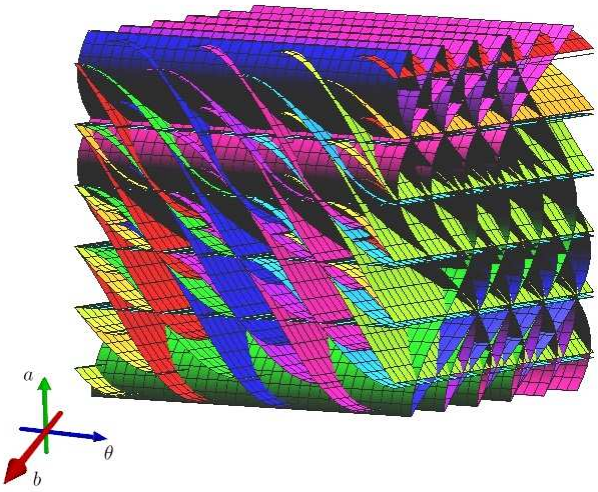

Figure 4: Discrete rigid transformations (DRTs) are observed as regions separated by tipping surfaces.

4.2. Tipping curve

We remark that the tipping surface Φ_{pqk} in Equation (14) is independent of b. Consequently, the b-axis orthogonal cross-sections of Φ_{pqk} at any $b \in$

$\mathbb{R}$ are identical, and each has the form of a trigonometric curve, called a *tipping curve*, denoted by ϕ_{pqk}. This means that (14) is the equation of a *tipping curve* as well if it is observed in the plane (a, θ). Similarly, (15) 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*

$$\left| \begin{array}{rccl} \phi_{pqk}: & \mathbb{R} & \longrightarrow & \mathbb{R} \\ & \theta & \longmapsto & a = k + \frac{1}{2} + q\sin\theta - p\cos\theta, \end{array} \right. \tag{16}$$

$$\left| \begin{array}{rccl} \psi_{pql}: & \mathbb{R} & \longrightarrow & \mathbb{R} \\ & \theta & \longmapsto & b = l + \frac{1}{2} - p\sin\theta - q\cos\theta. \end{array} \right. \tag{17}$$

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 as:*

$$\left| \begin{array}{rccl} F_\phi: & \mathbb{R} & \longrightarrow & \mathbb{R}^{|\mathbb{Z}^3|} \\ & \theta & \longmapsto & \bigcup\limits_{p,q,k\in\mathbb{Z}} \phi_{pqk}(\theta), \end{array} \right.$$

$$\left| \begin{array}{rccl} F_\psi: & \mathbb{R} & \longrightarrow & \mathbb{R}^{|\mathbb{Z}^3|} \\ & \theta & \longmapsto & \bigcup\limits_{p,q,l\in\mathbb{Z}} \psi_{pql}(\theta). \end{array} \right.$$

Figure 5 illustrates a part of F_ϕ for an digital image I of size 5×5; we set here $p, q \in [-2, 2]$ and $k \in [-3, 2]$.

4.3. Properties of tipping curves

So far we have studied the definitions of *tipping surfaces* and *tipping curves*. We will discuss more about combinatorial structures of discrete rigid transformations by using the sets of tipping surfaces and curves. Before this, we will state in this part some properties about tipping curves. From Definition 5, we remarked that F_ϕ and F_ψ have the 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 talks about the unique representation of tipping curves. This is a direct result of the fact that the basis $\{1, cos\theta, sin\theta\}$ of function (16) is linearly independent.

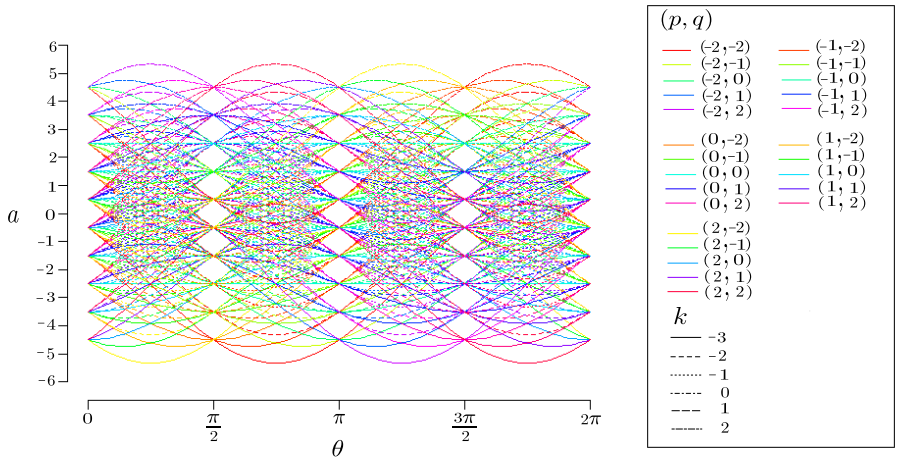

Figure 5: A part of the family of tipping curves F_ϕ for an digital image of size 5×5.

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 for the algorithm of constructing the combinatorial structure of discrete rigid transformations. Geometrically, two tipping curves can relate to each other in five different ways; they are identical, intersecting[1], vertical offset[2], tangent[3] or not intersecting (see Figure 6). Analytical interpretation of these relationships is expressed as follows.

Property 2. *Let ϕ_{pqk} and $\phi_{p'q'k'}$ be two tipping curves. Setting $K = k - k'$, $P = p - p'$, $Q = q - q'$ and $\Delta = 4Q^2(P^2 - K^2 + Q^2)$, we have the following relations between ϕ_{pqk} and $\phi_{p'q'k'}$:*

- *if $P = 0$, $Q = 0$, and*
  - (i) *if $K = 0$ then ϕ_{pqk} and $\phi_{p'q'k'}$ are identical,*
  - (ii) *otherwise they are vertical offset,*
- *otherwise,*

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

[2]Two tipping curves are vertical offset if one can be obtained by translating the other vertically, thus they do not intersect.

[3]Two 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.

(iii) *if* $\Delta < 0$ *then they have no intersection,*
(iv) *if* $\Delta = 0$ *then they are tangent,*
(v) *if* $\Delta > 0$ *then they intersect.*

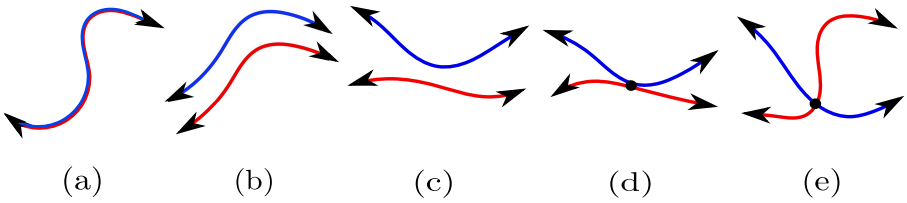

Figure 6: Relationships between two tipping curves, which are identical (a), vertical offset (b), not intersecting (c), tangent (e) and intersecting (d).

Proof (i) is easily induced by Property 1. Now, assuming that two tipping curves ϕ_{pqk} and $\phi_{p'q'k'}$ are different. The solution set of the equations ϕ_{pqk} and $\phi_{p'q'k'}$ determinates the nature of the relationships between two curves; it gives the either intersection points (v), tangent point (iv), or empty set (ii)(iii). Supposing the equation system has a solution, then we can find a 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

$$K - P\cos\theta \;\; = \;\; -Q\sin\theta.$$

Squaring both sides of the equation, we obtain

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

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

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

which is a quadratic equation 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 $P = 0$, $Q = 0$ and $K \neq 0$, then two curves are vertical offset (ii). Otherwise $P^2 + Q^2 \neq 0$, the discriminant Δ of the above quadratic equation determines the number and nature of the roots. There are three cases:

- $\Delta < 0$: no-intersection (iii),
- $\Delta = 0$: tangent curves (iv),
- $\Delta > 0$: intersecting curves (v).

■

Therefore, 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') with exact computing[4]. The proof of Property 2 implies the following corollaries.

Corollary 1. *Given two tipping curves ϕ_{pqk} and $\phi_{p'q'k'}$, if they are*

- *tangent, the tangent point satisfies cos $\theta = \frac{KP}{P^2+Q^2}$,*
- *intersect, the intersection points satisfy cos $\theta = \frac{-2KP \pm \sqrt{\Delta}}{2(P^2+Q^2)}$,*

where $P = p - p'$, $Q = q - q'$, $K = k - k'$ and $\Delta = 4Q^2(P^2 - K^2 + Q^2)$.

Corollary 2. *Two distinct tipping curves have at most two intersections.*

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, such that*

$$\begin{aligned} F_\phi(\theta) &= -F_\phi(\theta), \\ F_\phi(\theta) &= F_\phi(-\theta + m_1\frac{\pi}{4}) \quad for\ \ m_1 \in \mathbb{Z}. \end{aligned}$$

Property 4. *The family of tipping curves F_ϕ is periodic, such that*

$$\begin{aligned} F_\phi(\theta) &= F_\phi(\theta + m_2\frac{\pi}{2}) \quad for\ \ m_2 \in \mathbb{Z}, \\ F_\phi(\theta) &= F_\phi(\theta) + m_3 \quad for\ \ m_3 \in \mathbb{Z}. \end{aligned}$$

The following property studies about the complexity of F_ϕ for a digital image I of a finite size. This property induces the fact that the number of possible DRTs is finite and bounded by the size of I.

[4]Exact computing means using only integers during computation.

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) $N+1$ *vertical 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 triplet (p, q, k). Since $0 \leq p \leq N-1$, $0 \leq q \leq N-1$ and $0 \leq k \leq N$, there are $N^2(N+1)$ tipping curves. (ii) and (iii): From Property 2, two trigonometric curves $\phi_{p_iq_ik_i}$ and $\phi_{p_jq_jk_j}$ are vertical 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 vertical offset tipping curves, such that each set contains $N+1$ curves having different values of k. (iv): Thanks to the property 4, we need only evaluating (16) at $\theta = 0$, we have $a(0) = k + \frac{1}{2} + q$. The number of intersections is the number of combinations of q and k, which is $2N$. ■

5. Graph representation of discrete rigid transformations

As we discussed in the previous section, the parameter space of rigid transformation is divided into DRTs, whose the boundaries are the tipping surfaces. 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 a label of the edge.*

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

As was mentioned, each tipping surface is modelled by an integer triplet, such as for the vertical (resp. horizontal) tipping surface Φ_{pqk} (resp. Ψ_{pql}), we have (p, q, k) (resp. (p, q, l)). Reversely, given an integer triplet (p, q, k), it can be either Φ_{pqk} or Ψ_{pqk}. In order to distinguish between the integer triplets

of vertical and horizontal tipping surfaces, we will use an integer quadruple (p, q, k, i) for modelling each tipping surface, where i indicates either vertical or horizontal ones.

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 ones, such that they differ by only one pixel. By that way, the DRT graph $G = (V, E)$ does not contain any geometric information, such as the parameter (a, b, θ) of rigid transformations, but integer quadruples (p, q, k, i) representing the tipping surfaces.

The construction of a DRT graph can be solved with the help of surface arrangements [13, 14], with a complexity of $\mathcal{O}(n^3)$ for general surfaces, where n is the number of surfaces. In this article, we propose an algorithm for the problem whose complexity is $\mathcal{O}(n^2)$ by using 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 whose equations are given in (14) and (15). 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 plane either (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 for 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 tipping curves C partition the plane into three types of maximal connected regions: *a vertex* is an intersection point of curves, *an arc* is a maximal connected portion of a curve that is not intersected by any other curve, and *a face* is a maximal connected region that is not intersected by any other curve in C.

Our problem is formalized as follows: given a set of n tipping curves C, we would like to construct a 2D DRT graph of C, denoted by G^C. Figure 7 illustrates the 2D DRT graph for a set of 4 tipping curves on the plane (a, θ).

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 $e \in E^C$.

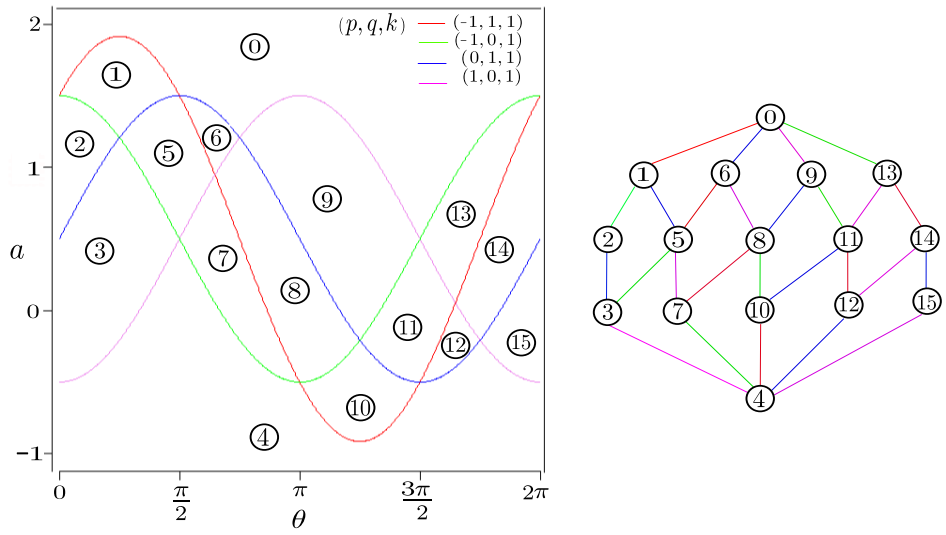

Figure 7: Four tipping curves on plane (a, θ) (left) and its 2D DRT graph (right).

This problem is related to *curve arrangements* [15, 16]. Various methods have proposed such as: incremental construction [17], zone theorem [18], sweeping line [16], etc. A comprehensive discussion on arrangement can be found in [19]. Since our curves are tipping curves, there thus exist many degenerate cases such as tangent and multiple intersections. In additions, we are only interested in the informations about faces and arcs in the arrangement. Therefore, instead of using the basic algorithm of curve arrangement, we will propose a variation of *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 points to construct G^C incrementally. The detail of algorithm and its implementation will be explained in the following.

6.2. Principles of incremental 2D DRT graph reconstruction

We restate that C denotes a set of n tipping curves. We define a *(vertical) cut*, noted γ, as monotonic line intersecting exactly once for each tipping curve in C. Please note that γ is an unbounded simple curve and represented by a sequence of tipping curves which γ intersects from the top to bottom, as illustrated in Figure 8. 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 the intersection points, called *event points*. The moment at which γ reaches an event point, the algorithm updates γ and constructs a part of the graph G^C. This is called an *elementary step* of the algorithm. As a set of event points forms a series of elementary steps, we need to maintain a sorted set of event points and progress γ in their increasing order.

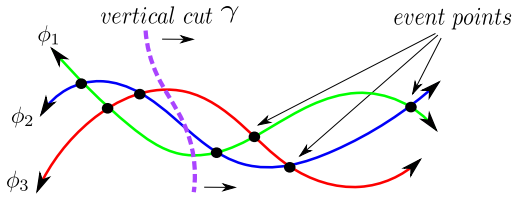

Figure 8: The event points, the vertical cut $\gamma = (\phi_3, \phi_2, \phi_1)$.

In fact, the cut *gamma* 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 $\gamma = (\phi_1, \phi_2, ..., \phi_n)$ 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*

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

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

At each element step, the partial graph δG^C is updated with respect to the change of γ. This corresponds to modify a part of δG^C. If such an operation is applied, then the sweep progresses as such δG_C is modified and integrated in G_C for constructing the final graph.

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

$$G^C = \cup \sum_{i=0}^{m} \delta G_i^C$$

where δG_i^C is the partial graph at the i-th element 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 for its 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

Our initialization step provides the initial 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* $\phi'_{pqk}(0) < \phi'_{p'q'k'}(0)$, *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 δG_0^C of γ_0. Note that δG_0^C corresponds to the initial 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 will first present the construction algorithm in *simple* cases, such that any intersection is made by only two crossing tipping curves (Figure 9a). Otherwise, we have *degenerate* or *non-simple* cases (Figure 9b). We will discuss how to deal with such degeneracies in the next part.

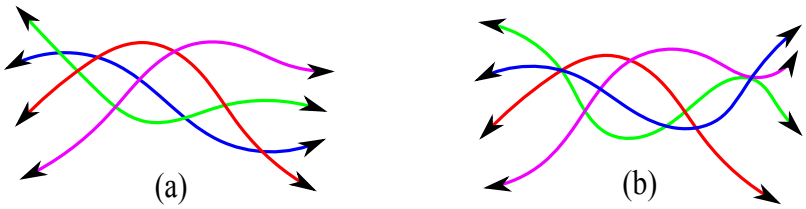

Figure 9: Illustration for simple (a) and non-simple (b) cases.

6.4.1. Detecting and ordering event points

The following question arises in 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. 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: $Q^2(P^2 - K^2 + Q^2) > 0$, where $P = p - p'$, $Q = q - q'$ and $K = k - k'$. Then the coordinates of their intersections are also obtained by Corollary 1.

All event points need to be sorted and then stored in a *queue*, to be handled one by one. Such a queue of event points is called an *event queue* and 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 to define the order on $\mathcal{E}$. The event points are sorted by $\prec_{\mathcal{E}}$ as follows.

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

Sorting event points can be done with exact computing. Indeed, from Corollary 1 we know that the coordinates of any intersection of two tipping curves can be expressed by quadratic irrationals[5]; they have the form $\frac{A+\sqrt{B}}{C}$, where $A, B, C \in \mathbb{Z}$. Sorting event points then becomes the problem of comparing quadratic irrationals. It is known in [20] 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, and this representation is unique. Then the comparison of their corresponding periodic continued fractions are made by comparing two sequences of integers

[5]A quadratic irrational is an irrational number that is a solution of some quadratic equations.

(see Appendix A). Comparison between two quadratic irrationals is realized in $\mathcal{O}(\sqrt{A}\ln A)$ and sorting event points requires $\mathcal{O}(|\mathcal{Q}|\log|\mathcal{Q}|)$ time, so that there is no influence to the global complexity of algorithm.

The fact of using integer arithmetic is first to avoid the technical problems due to the use of floating points. Also the more important point is to prevent faux event points which are obtained from the limited precision of floating-point representation supported in computer implementation for degenerate cases such as multiple tangent and/or intersecting points.

Rather than the coordinates of event points, a pair of their intersecting tipping curves is 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. This representation however must be modified in degenerate cases (see Section 6.5.1).

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 10. According to such a change of γ, the partial graph δG^C is modified.

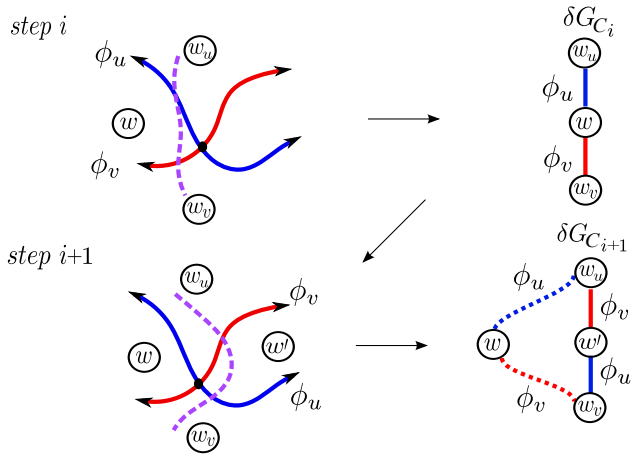

Figure 10: Illustration of a progress of a cut by which the partial graph δG^C_{i+1} is modified from δG^C_i in simple case.

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

1. finding the removed vertex w bounded by ϕ_u and ϕ_v.
2. deleting two edges that are adjacent to w.
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. This implementation is given in Procedure 1.

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.

$e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$;
$\{w\} \leftarrow \vartheta(e_u) \cap \vartheta(e_v)$;
$\{w_u\} \leftarrow \vartheta(e_u) \setminus \{w\}$; $w_v \leftarrow \vartheta(e_v) \setminus \{w\}$;
$\Delta V_-^C \leftarrow \{w\}$; // w is a removed vertex
$\Delta E_-^C \leftarrow \{(w_u, w, \phi_u), (w, w_v, \phi_v)\}$;
$\Delta V_+^C \leftarrow \{w'\}$; // w' is a new vertex
$\Delta E_+^C \leftarrow \{(w_u, w', \phi_v), (w', w_v, \phi_u)\}$;
$\delta G_{i+1}^C \leftarrow \delta G_i^C \setminus \Delta G_-^C \cup \Delta G_+^C$;
/* $\Delta G_-^C = (\Delta V_-^C, \Delta E_-^C)$ is a subtracting part of δG_i^C and $\Delta G_+^C = (\Delta V_+^C, \Delta E_+^C)$ is a adding part of δG_i^C */

The procedure requires 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 present hereafter an algorithm of our incremental 2D DRT graph construction. The algorithm builds G^C by picking event points in $\mathcal{Q}$ one by one. Each iteration consists in modifying the partial graph δG^C according to the current cut γ (see Section 6.4.2), and then integrate δG^C into G^C.

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

Input: A tipping curve set $C = \{\phi_1, \phi_2, ..., \phi_n\}$.
Output: A 2D DRT graph $G^C = (V^C, E^C)$.

```
1 Initialize G^C and δG^C as explained in Section 6.3 ;
2 Generate an event queue Q as explained in Section 6.4.1 ;
3 while Q ≠ ∅ do
4 |   q ← dequeue(Q) ;
5 |   δG^C ← Procedure 1 (δG^C, q) ;
6 |   G^C ← G^C ∪ δG^C ;
```

6.5. Incremental 2D DRT graph construction for degenerate cases

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

6.5.1. Detecting and sorting event points

According to the nature of the degeneracies, they can be classified into the following three cases, as shown in Figure 11: more than two tipping curves are

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

In fact, those degeneracies can be detected by using Property 2 and Corollary 1 as mentioned in Section 6.4.1. Comparing the coordinates of intersections enable us to detect degenerate event points; they have the same coordinates. Such an event point is now represented by a family of tipping curves that go through the intersection point.

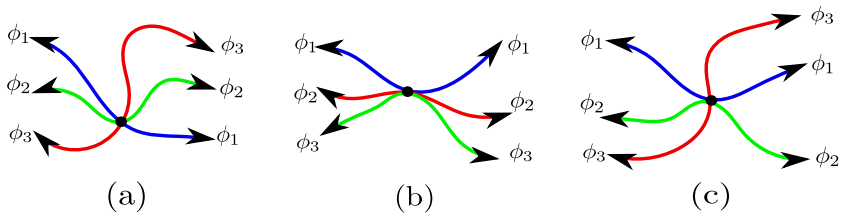

Figure 11: Degenerate cases: multiple intersection (a), multiple tangent point (b) and multiple mixed point (c).

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 more complicated. Observing carefully Figure 11, we remark that tipping curves are decomposed into sets sorted by tangent values and each sets contains tipping curves have the same tangent. While 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 gives the general procedure for multiple intersection cases in which each tipping curve is seen as a tangent set.

At each event point, the elementary step consists in 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 in stead of a pair as in simple cases. Let $\mathbf{q}$ be an event point, then $\mathbf{q} = \{\tau_0, \tau_1, ..., \tau_m\}$ is a family of tipping curves where each $\tau_j = \{\phi_u, \phi_v, ...\}$ is a set of tipping curves having 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 in the following steps:

1. generating two lists of tipping curves which give the order of tipping curves before and after event point $\mathbf{q}$ and storing them in two queues $\mathbf{Q_1}$ and $\mathbf{Q_2}$ respectively. The detail is given below.
2. finding initial and terminal vertices, u and v, for an event point $\mathbf{q}$ between which δG_i^C changes.
3. finding the removed vertices between u and v.
4. deleting the edges which are tied to those removed vertices.

5. replacing the removed vertices by the new ones.
6. linking the new vertices, u and v by the new edges.

While creating new edges in the last step, each edge is given a label of a tipping curve taken from $\mathbf{Q_2}$.

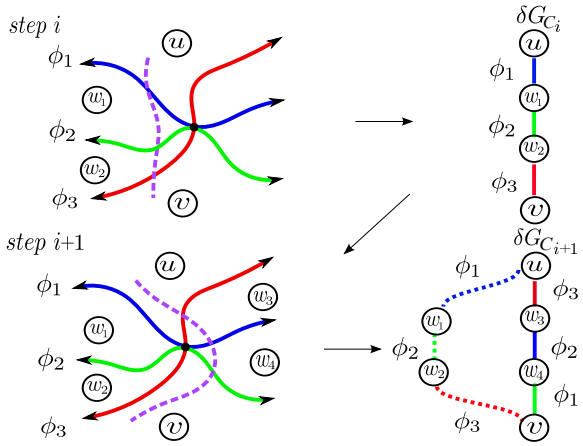

Figure 12: Illustration of a progress of a cut by which the partial graph δG_{i+1}^C is modified from δG_i^C in a degenerate case.

We will explain how to generate two queues $\mathbf{Q_1}$ and $\mathbf{Q_2}$ 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 of tipping curves 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 $\mathbf{q} = (\tau_0, \tau_1, ..., \tau_m)$ and each $\tau_j = (\phi_u, \phi_u, ...)$ with respect to δE_i^C. The queues $\mathbf{Q_1}$ and $\mathbf{Q_2}$ are generated as follows.

for $j = 0 \to m$ **do**
  for each $\phi \in \tau_j$ **do**
    $\mathbf{Q_1} \leftarrow enqueue(\phi)$;
  end for
  for each $\phi \in \tau_{m-j}$ **do**
    $\mathbf{Q_2} \leftarrow enqueue(\phi)$;
  end for
end for

Note that in $\mathbf{Q_1}$ tipping curves have the same order as in δE_i^C, and in $\mathbf{Q_2}$ 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 $\mathbf{q}$.

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.

Generate $\mathbf{Q_1}$ and $\mathbf{Q_2}$ as explained above.
$\phi_u \leftarrow dequeue(\mathbf{Q_1})$; $\phi_v \leftarrow dequeue(\mathbf{Q_1})$;
$e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$;
$\{u\} \leftarrow \vartheta(e_u) \setminus \vartheta(e_u) \cap \vartheta(e_v)$; // u is the initial vertex
$\mathbf{Q_1} \leftarrow enqueue(\phi_v)$; $\mathbf{Q_1} \leftarrow enqueue(\phi_u)$;
$\phi_u \leftarrow dequeue(\mathbf{Q_2})$; $\phi_v \leftarrow dequeue(\mathbf{Q_2})$;
$e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$;
$\{v\} \leftarrow \vartheta(e_u) \setminus \vartheta(e_v) \cap \vartheta(e_v)$; // v is the terminal vertex
$\mathbf{Q_2} \leftarrow enqueue(\phi_v)$; $\mathbf{Q_2} \leftarrow enqueue(\phi_u)$;
while $\mathbf{Q_1} \neq \emptyset$ **do**
    $\phi_u \leftarrow dequeue(\mathbf{Q_1})$; $\phi_v \leftarrow dequeue(\mathbf{Q_1})$;
    $e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$;
    $\{w\} \leftarrow \vartheta(e_u) \cap \vartheta(e_v)$;
    $\{u\} \leftarrow \vartheta(e_u) \setminus \{w\}$;
    $\Delta V_-^C \leftarrow \Delta V_-^C \cup \{w\}$; // w is a removed vertex
    $\Delta E_-^C \leftarrow \Delta E_-^C \cup \{(w_u, w, \phi_u)\}$;
    if $\mathbf{Q_1} \neq \emptyset$ **then** $\mathbf{Q_1} \leftarrow enqueue(\phi_v)$;
$j \leftarrow 0$;
$\phi \leftarrow dequeue(\mathbf{Q_2})$;
$\Delta V_+^C \leftarrow \{w_j\}$; // w_j is a new vertex
$\Delta E_+^C \leftarrow \{(u, w_j, \phi)\}$;
while $\mathbf{Q_2} \neq \emptyset$ **do**
    $j \leftarrow j + 1$;
    $\phi \leftarrow dequeue(\mathbf{Q_2})$;
    $\Delta V_+^C \leftarrow \Delta V_+^C \cup \{w_j\}$; // w_j is a new vertex
    if $\mathbf{Q_2} \neq \emptyset$ **then** $\Delta E_+^C \leftarrow \Delta E_+^C \cup \{(w_{j-1}, w_j, \phi)\}$;
    else $\Delta E_+^C \leftarrow \Delta E_+^C \cup \{(w_j, v, \phi)\}$;
$\delta G_{i+1}^C \leftarrow \delta G_i^C \setminus \Delta G_-^C \cup \Delta G_+^C$;

7. Construction of discrete rigid transformation graph

In this section, we will present an algorithm to construct a DRT graph G from a set of tipping surfaces. The basic idea is *quite similar* to the algorithm

for construction a 2D DRT graph, which are based on sweeping a cut and modifying it at event points. At each elementary step, we modify a partial graph δG and then integrate it into G.

7.1. Principles of incremental DRT graph construction

As explained, the graph of 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)). Observing (14) and (15) of tipping curves, we find out 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 equivalent, denoted $G^a \sim G^b$.*

Proof Translating (15) by $\frac{\pi}{2}$ with respect to θ, we obtain a set of tipping curves that corresponds to the set of (14), 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 equivalent. ■

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 another. 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 generate this event point, but not as their coordinates; thus $\mathcal{Q}_a \neq \mathcal{Q}_b$ event if their event point have the same coordinates.

Our algorithm uses two cuts, each of which sweeps a plane that is orthogonal to the θ-axis and considered as a vertical cut in both planes (a, θ) and (b, θ) to construct G. Thus the elementary steps are performed for each pair of event points on the two planes to generate δG of G by combining δG^a and δG^b. The construction of δG^a and δG^b is shown in Section 6.

7.2. Initial graph construction

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

- a vertex $v \in \delta V_0$ is an association of two vertices $v_a \in \delta V_0^a$ and $v_b \in \delta V_0^b$,
- an edge $e \in \delta E_0$ connects two vertices sharing an edge in either δE_0^a or δE_0^b.

Therefore δG_0 contains $(n+1)^2$ vertices and n^2 edges, where n is the number of given tipping surfaces. Formally, we then have:

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

7.3. Elementary step

The modification of the partial graph δG_{i+1} at each elementary step is made by Procedure 3 from δG_i^a and δG_i^b in a similar way as the initialization. Except, a new vertex is generated from a pair of a new vertex in δG_{i+1}^a and a vertex in δG_{i+1}^b or vice-versa. So a new edge connects two vertices sharing an edge in δE_i^a and δE_i^b. The function is given as follows.

Procedure 3: Elementary step for DRT graph construction.

Input: A partial graph δG_i and two modified 2D partial graphs δG^a_{i+1} and δG^b_{i+1}.

Output: The modified partial graph δG_{i+1}.

Initialize $\delta G_{i+1} = \emptyset$, with $\delta V_{i+1} = \emptyset$ and $\delta E_{i+1} = \emptyset$;
foreach $u \in \delta V^a_{i+1}$ *and* $v \in \delta V^b_{i+1}$ **do**
    if $(u, v) \notin \delta V_i$ **then**
        $\delta V_{i+1} \leftarrow \delta V_{i+1} \cup \{(u, v)\}$; // $u \in \Delta V_{a+}$ and $v \in \Delta V_{b+}$
foreach $e_u = (u_1, u_2, \phi_u) \in \delta E^a_{i+1}$ **do**
    foreach $v \in \delta V^b_{i+1}$ **do**
        $e \leftarrow ((u_1, v), (u_2, v), \phi_u)$;
        if $e \notin \delta E_i$ **then**
            $\delta E_{i+1} \leftarrow \delta E_{i+1} \cup \{e\}$;
foreach $e_v = (v_1, v_2, \phi_v) \in \delta E^b_{i+1}$ **do**
    foreach $u \in \delta V^a_{i+1}$ **do**
        $e \leftarrow ((u, v_1), (u, v_2), \phi_v)$;
        if $e \notin \delta E_i$ **then**
            $\delta E_{i+1} \leftarrow \delta E_{i+1} \cup \{e\}$;

7.4. Algorithm

Our algorithm 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 coordinate are identical. For each iteration, we generate δG and then integrate it into G.

Algorithm 2: Construction of DRT graph.

Input: A set of tipping surfaces, *i.e.* two sets of tipping curves.
Output: A DRT graph $G = (V, E)$.

Initialize δG^a and δG^b as explained in Section 6.3
Initialize G, δG from δG^a and δG^b as explained in Section 7.2 ;
Generate $\mathcal{Q}_a$ and $\mathcal{Q}_b$ as explained in Section 6.5.1
while $\mathcal{Q}_a \neq \emptyset$ *and* $\mathcal{Q}_b \neq \emptyset$ **do**
    $\mathbf{q}_a \leftarrow dequeue(\mathcal{Q}_a)$;
    $\delta G^a \leftarrow$ Procedure 2 $(\delta G^a, \mathbf{q}_a)$;
    $\mathbf{q}_b \leftarrow dequeue(\mathcal{Q}_b)$;
    $\delta G^b \leftarrow$ Procedure 2 $(\delta G^b, \mathbf{q}_b)$;
    $\delta G \leftarrow$ Procedure 3 $(\delta G, \delta G^a, \delta G^b)$;
    $G \leftarrow G \cup \delta G$;

8. Complexity analysis and experiments

8.1. Complexity of DRT graph

The complexity of a 2D DRT graph, *i.e.* the numbers of its vertices and edges, is obtained by 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 $2.\binom{2}{n} = n(n-1)$.

(ii) The vertices of G^C correspond to the faces of the arrangement of tipping curves. If $n = 1$, the number of faces is $2 = 1^2+1$, since there are two faces in two sides of the curve. Let us now assume that there are $(n-1)^2+1$ faces for $n-1$ tipping curves. When adding an n-th curve, this curve will be divided into at most $2(n-1)+1$ arcs by the $n-1$ other curves, and each one of these arcs will split at most one face into two faces. 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 $(n-1)^2+1+2(n-1)+1 = n^2+1$. The result follows by induction.

(iii) If $n = 1$, there is one curve and thus we obtain $1 = 2.1^2 - 1$ edge. Let us now assume that for $n-1$ curves; there are at most $2(n-1)^2 - (n-1)$ edges. When adding an 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: $2(n-1)^2-(n-1)+2(n-1)+2(n-1)+1 = 2n^2 - n$. The result follows by induction. ■

Practically, the number of event points, vertices and edges, will be generally lower than these upper-bounds, due to degenerated cases in the tipping curves arrangement. From Property 5, we know that $n = \mathcal{O}(N^3)$ for an image of size $N \times N$. Then, these complexities can be re-expressed as $\mathcal{O}(N^6)$.

As mentioned in Section 7 the construction of a DRT graph G is done 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 are 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. Experiments

We have implemented our algorithm in C++ which computes the DRT graph for a given size of 2D digital image. From the experiments, the numbers of vertices (and edges) of the DRT graphs have been computed 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 13, validate the $\mathcal{O}(N^9)$ space complexity stated in the previous theorem.

9. Conclusion

In this article, we have introduced a combinatorial structure as a graph for modelling the parameter space of digital rigid transformations. Note that this graph does not contain any parameter (a, b, θ) of rigid transformations

	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: Number of vertices and edges of the graphs, with respect to the image size ($N \times N$).

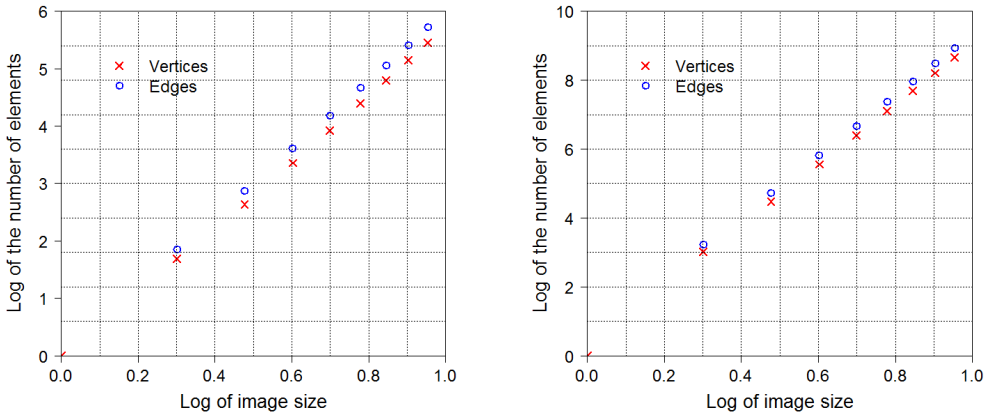

Figure 13: The relation between image size and number of elements, (*i.e.* vertices or edges) in Table 2 for 2D DRT graphs (left) and DRT graphs (right).

but the integer quadruples (p, q, k, i) modelling the discontinuities of rigid transformations. Such a graph consists of finite sets of vertices and edges, in which each vertex represents a digital transformed image and each edge joins/links two vertices by a transformation moving only one pixel between two transformed images. This structure presents a space complexity $\mathcal{O}(N^9)$, 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 emphasise both the correctness of the algorithm, and the estimated time/space polynomial complexities. Due to these complexities, it remains hardly tractable to compute the proposed combinatorial structure for images of big size.

However, this size limitation does not actually constitute a deadlock for 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 works will now involve studying the way to use the proposed combinatorial structure in multiscale strategies (enabling to process large images without computing the whole data structure). Furthermore, for image registration most of the existing methods [23] have no guarantee to find a global optimal solution in general. With our approach, we may define a new metric for the graph based on neighbouring relations between discrete rigid transformations, and this way leads 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 [24]) 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 [20, 25], it has been proved that a quadratic irrational can be rewritten as a periodic continued fraction. More formally, for any quadratic irrational $Q = \frac{p+\sqrt{q}}{r}$ ($p, q, r \in \mathbb{Z}$, $q > 0$, $r \neq 0$) we have:

$$Q = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ldots + \cfrac{1}{a_n + \cfrac{1}{\ldots}}}}} \quad . \tag{A.1}$$

Such a periodic continued fraction can be unambiguously expressed as the finite sequence of integers $[a_0; a_1, a_2, \ldots a_n]$ modelling its period.

Based on this formulation, the comparison between two quadratic irrationals:

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

with $p_i, q_i, r_i, a_j, b_j \in \mathbb{Z}$, $q_i > 0$, and $r_i \neq 0$ for $i = 1, 2$ and $j = 0, 1, 2, \ldots$, can be performed as follows.

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. $E > 0$). This leads to Algorithm 3, which presents a mean complexity $\mathcal{O}(1)$.

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
  $E \leftarrow 0$
  $i \leftarrow 0$
  $(p_1^i, q_1^i, r_1^i) \leftarrow (p_1, q_1, r_1)$;
  $(p_2^i, q_2^i, r_2^i) \leftarrow (p_2, q_2, r_2)$;
  while $E = 0$ **do**
    $(q_1^{i+1}, r_1^{i+1}, a_1^{i+1}) \leftarrow$ Function $(q_1, p_1^i, r_1^i, a_1^i)$; `// calculate the term` a_1^{i+1} `of the continued fraction of` Q_1
    $(q_2^{i+1}, r_2^{i+1}, a_2^{i+1}) \leftarrow$ Function $(q_2, p_2^i, r_2^i, a_2^i)$; `// calculate the term` a_1^{i+1} `of the continued fraction of` Q_2
    $E \leftarrow (-1)^i(a_1^{i+1} - a_2^{i+1})$;
    $i \leftarrow i + 1$;
  if $E > 0$ **then** return $>$;
  else return $<$;

The function for calculating the term a^{i+1} is given as follows.

Function – Calculates the i+1-th term of a continued fraction

Input: The triplet (q, p^i, r^i) and the i-th term a^i.
Output: The triplet $(p^{i+1}, r^{i+1}, a^{i+1})$.
$p^{i+1} \leftarrow a^i.r^i - p^i$;
$r^{i+1} \leftarrow \frac{q-(p^{i+1})^2}{r^i}$;
$a^{i+1} \leftarrow \lfloor \frac{p^{i+1}+\sqrt{q}}{r^{i+1}} \rfloor$; `//` $\lfloor \ . \ \rfloor$ `is the floor function`

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Computing Surveys 38 (4) (2006) 1–45.

[2] L. G. Brown, A survey of image registration techniques, ACM Computing Surveys 24 (4) (1992) 326–376.

[3] A. Amir, O. Kapah, D. Tsur, Faster two-dimensional pattern matching with rotations, Theoretical Computer Science 368 (3) (2006) 196–204.

[4] Y. Thibault, Rotations in 2D and 3D discrete spaces, Ph.D. thesis, University Paris-Est (2010).

[5] A. Amir, G. M. Landau, U. Vishkin, Efficient pattern matching with scaling, Journal of Algorithms 13 (1) (1992) 2–32.

[6] A. Amir, A. Butman, M. Lewenstein, E. Porat, Real two dimensional scaled matching, Algorithmica 53 (3) (2009) 314–336.

[7] 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.

[8] C. Hundt, M. Liśkiewicz, On the complexity of affine image matching, in: Symposium on Theoretical Aspects of Computer Science, Proceedings, Vol. 4393 of Lecture Notes in Computer Science, Springer, 2007, pp. 284–295.

[9] C. Hundt, Affine image matching is uniform TC^0-complete, in: Combinatorial Pattern Matching, Proceedings, Vol. 6129 of Lecture Notes in Computer Science, Springer, 2010, pp. 13–25.

[10] C. Hundt, M. Liśkiewicz, Combinatorial bounds and algorithmic aspects of image matching under projective transformations, in: Mathematical Foundations of Computer Science, Proceedings, Vol. 5162 of Lecture Notes in Computer Science, Springer, 2008, pp. 395–406.

[11] V. A. Kovalevsky, Finite topology as applied to image analysis, Computer Vision, Graphics & Image Processing 46 (2) (1989) 141–161.

[12] T. Y. Kong, A. Rosenfeld, Digital topology: Introduction and survey, Computer Vision, Graphics & Image Processing 48 (3) (1989) 357–393.

[13] M. Sharir, Recent developments in the theory of arrangements of surfaces, in: Foundations of Software Technology and Theoretical Computer Science, Proceedings, Vol. 1738 of Lecture Notes in Computer Science, Springer, 1999, pp. 1–21.

[14] T. M. Chan, On levels in arrangements of surfaces in three dimensions, in: Symposium on Discrete Algorithms, Proceedings, ACM-SIAM, 2005, pp. 232–240.

[15] J. Snoeyink, J. Hershberger, Sweeping arrangements of curves, in: Symposium on Computational Geometry, Proceedings, ACM, 1989, pp. 354–363.

[16] H. Edelsbrunner, L. J. Guibas, Topologically sweeping an arrangement, Journal of Computer and System Sciences 38 (1) (1991) 165–194.

[17] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-Verlag, 1987.

[18] H. Edelsbrunner, R. Seidel, M. Sharir, On the zone theorem for hyperplane arrangements, in: New Results and New Trends in Computer Science, Proceedings, Vol. 555 of Lecture Notes in Computer Science, Springer, 1991, pp. 108–123.

[19] 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.

[20] K. H. Rosen, Elementary Number Theory and its Applications, 3rd Edition, Addison-Wesley, 1992.

[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] B. Zitov, J. Flusser, Image registration methods: a survey, Image and Vision Computing 21 (2003) 977–1000.

[24] K. Fredriksson, E. Ukkonen, Combinatorial methods for approximate pattern matching under rotations and translations in 3D arrays, in: String Processing and Information Retrieval, Proceedings, IEEE, 2000, pp. 96–104.

[25] A. Y. Khinchin, Continued Fractions, Dover Publications, 1964.