

On 2D constrained discrete rigid transformations

Phuc Ngo, Yukiko Kenmochi, Nicolas Passat, Hugues Talbot

▶ **To cite this version:**

Phuc Ngo, Yukiko Kenmochi, Nicolas Passat, Hugues Talbot. On 2D constrained discrete rigid transformations. Annals of Mathematics and Artificial Intelligence, 2015, Annals of Mathematics and Artificial Intelligence, 75 (1-2), pp.163-193. 10.1007/s10472-014-9406-x . hal-00838184v2

HAL Id: hal-00838184
https://hal.science/hal-00838184v2

Submitted on 16 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

On 2D Constrained Discrete Rigid Transformations

Phuc Ngo · Yukiko Kenmochi · Nicolas Passat · Hugues Talbot

Received: date / Accepted: date

Abstract Rigid transformations are involved in a wide range of digital image processing applications. In such a context, they are generally considered as continuous processes, followed by a digitization of the results. Recently, rigid transformations on $\mathbb{Z}^2$ have been alternatively formulated as a fully discrete process. Following this paradigm, we investigate – from a combinatorial point of view – the effects of pixel-invariance constraints on such transformations. In particular we describe the impact of these constraints on both the combinatorial structure of the transformation space and the algorithm leading to its generation.

Keywords Rigid transformation · discrete geometry · combinatorial structure · image processing · pixel-invariance constraints

1 Introduction

Rigid transformations are frequently involved in applications of computer vision and image processing (*e.g.*, motion tracking [10,28], image registration [15,29] or pattern recognition [4,8]). In such applications, images are generally digital, and thus defined on finite sets of points in the Eulerian space $\mathbb{Z}^n$. However, rigid transformations applied on such digital images are usually performed on the Euclidean space ($\mathbb{R}^n$). Their results then need to be followed by a subsequent digitization process to finally produce transformed images in $\mathbb{Z}^n$.

Phuc Ngo (corresponding author), Yukiko Kenmochi, Hugues Talbot
Université Paris-Est, LIGM, UPEMLV-ESIEE-CNRS, France
Tel.: +33-145926737
Fax: +33-145926699
E-mail: ngo.diemphuc@gmail.com

Nicolas Passat
Université de Reims Champagne-Ardenne, CReSTIC, France

In a recent work [18], we have proposed to alternatively study rigid transformations[1] on $\mathbb{Z}^2$ as a *fully discrete process*, similarly to previous contributions related, *e.g.*, to rotations [3,14,20,21,23,24] or quasi-affine transformations [6]. In this context, two main questions were considered: (*i*) How many rigid transformations can be defined on a finite subspace of $\mathbb{Z}^2$? (*ii*) How to generate all these transformations? The difficulty of these questions derives from the infinite number of rigid transformations in $\mathbb{R}^2$. Recently, some combinatorial studies have been devoted to 2D pattern matching under different classes of transformations such as rotations, scaling, affine and projective transformations. In particular, some discretization techniques were developed by Hundt *et al.* [11–13]. Inspired by these works, we provided in [18] some combinatorial and algorithmic answers to the above two questions, and then contributed to the state of the art in this research area [2,11–13,27].

More precisely, in [18], a combinatorial structure, namely a *discrete rigid transformation graph* (or *DRT graph*), was introduced to model the parameter space of 2D rigid transformations on $\mathbb{Z}^2$. This DRT graph describes all the possible rigid transformations on a digital image. We showed that there exist in the order of N^9 such transformations, if $N \times N$ is the number of pixels in the image. In addition, the DRT graph explicitly models the "topological links" between such digital transformations, and thus allows the incremental construction of discrete rigid transformations via elementary image modifications pixel by pixel. The DRT graph can be used in a local fashion, *e.g.*, in pattern-based strategies, as proposed in [19] for analysing the topological invariance of digital images under arbitrary rigid transformations. Beyond the theoretical aspects of the DRT graph, its high-order polynomial complexity makes it difficult to generate the whole graph for large images, and to use it directly in imaging applications such as registration or warping [1,9,22,29].

To reduce the complexity of this graph, we propose to provide spatial constraints in order to guide the computation of such transformations. Indeed these constraints introduce prior knowledge that contribute to reducing the search space. In this article – that is an extended and improved version of the conference paper [17] – we investigate such constrained search paradigms from the combinatorial and algorithmic points of view. We focus in particular on the effects of geometric constraints on discrete rigid transformations, via the analysis of the DRT graph. More precisely, we investigate *pixel-invariance constraints*, which consist of enforcing the correspondence between points in an initial subspace of $\mathbb{Z}^2$ and points (or more generally regions) in a transformed space.

This article is organised as follows. Section 2 briefly introduces some basic notions of rigid transformations on digital images. Section 3 describes pixel-

[1] In fact, rigid transformations are composed of reflections, rotations and translations, while combinations of only translations and rotations are called *proper* rigid transformations. In this article, as in [18], we focus on the latter, since reflections are not as generally useful in matching and tracking applications, and may needlessly complicate the search space. By an abuse of language, we will continue to refer to proper rigid transformations as rigid transformations.

invariance constraints in the associated parameter space of rigid transformations. In Section 4, we develop an algorithmic process for generating a combinatorial structure modeling all the discrete rigid transformations and their relationships under given constraints. This section is completed by an algorithmic appendix, provided at the end of the manuscript. Complexity analyses of the proposed algorithm and the induced structures are described in Section 5. A concluding discussion is finally provided in Section 6.

2 Background notions

2.1 Digital images and digital rigid transformations

In a 2D continuous space, an *image* can be defined as a function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{V}$, where $\mathbb{V}$ is a given value space. In computer imaging, such images are represented as discrete functions obtained through a sampling process, and then called *digital images*. In general, the sampling process relies on partitioning $\mathbb{R}^2$ into Voronoi cells induced by a square grid structure. It associates every point in $\mathbb{R}^2$ to a unit grid square (namely, a *pixel*), and equivalently to a point in $\mathbb{Z}^2$. Such a sampling process, also referred as *digitization*, is often carried out by the following function

$$\left| \begin{array}{ll} D : \mathbb{R}^2 & \longrightarrow \mathbb{Z}^2 \\ \quad \boldsymbol{x} = (x, y) & \longmapsto \boldsymbol{p} = (p, q) = ([x], [y]) \end{array} \right. \tag{1}$$

where $[\,.\,]$ is a rounding operator. Consequently, a digital image associated to $\mathcal{I}$ can be formalized as $I : \mathbb{Z}^2 \to \mathbb{V}$. In other words, we have $I = \mathcal{I}_{|\mathbb{Z}^2}$, and for each $\boldsymbol{p} \in \mathbb{Z}^2$, the value $I(\boldsymbol{p})$ models the value of $\mathcal{I}$ on the associated *pixel* $\boldsymbol{p} + [-\frac{1}{2}, \frac{1}{2}[^2$, namely the Voronoi cell of $\mathbb{R}^2$ induced by $\mathbb{Z}^2$ around $\boldsymbol{p}$.

A 2D rigid transformation is defined as a rotation followed by a translation. In the continuous framework, such a transformation can be formally expressed as a bijective function $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$ such that for any $\boldsymbol{x} = (x, y) \in \mathbb{R}^2$, the transformed point $\mathcal{T}(\boldsymbol{x})$ has the form

$$\mathcal{T}(\boldsymbol{x}) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} \tag{2}$$

where the parameters $a, b \in \mathbb{R}$ represent the translation, while $\theta \in [0, 2\pi[$ is the rotation angle. In particular, such a transformation is unambiguously modeled by a triplet of parameters (a, b, θ), and will be often denoted by $\mathcal{T}_{ab\theta}$. When applied to an image $\mathcal{I} : \mathbb{R}^2 \to \mathbb{V}$, it provides a new transformed image $\mathcal{I} \circ \mathcal{T} : \mathbb{R}^2 \to \mathbb{V}$.

It is not possible to apply directly $\mathcal{T}$ to a digital image $I : \mathbb{Z}^2 \to \mathbb{V}$, since there is no guarantee that $\mathcal{T}(\boldsymbol{x}) \in \mathbb{Z}^2$ for $\boldsymbol{x} \in \mathbb{Z}^2$. In the discrete framework, the handling of *digital* rigid transformations requires to define a function $T_{ab\theta} : \mathbb{Z}^2 \to \mathbb{Z}^2$, which is a discrete analogue of $\mathcal{T}_{ab\theta}$. Following the digitization paradigm D proposed above, a *digital rigid transformation* T associated to

$\mathcal{T}$ can be conveniently performed by setting $T = D \circ \mathcal{T}$, as illustrated in the following diagram.

$$\begin{array}{ccc} \mathbb{Z}^2 & \xrightarrow{T=D\circ\mathcal{T}} & \mathbb{Z}^2 \\ \Big\downarrow Id & & \Big\uparrow D \\ \mathbb{R}^2 & \xrightarrow{\mathcal{T}} & \mathbb{R}^2 \end{array} \tag{3}$$

The function $T : \mathbb{Z}^2 \to \mathbb{Z}^2$ is then explicitly defined for $\boldsymbol{p} = (p, q) \in \mathbb{Z}^2$ by

$$T(\boldsymbol{p}) = D \circ \mathcal{T}(\boldsymbol{p}) = \begin{pmatrix} [p\cos\theta - q\sin\theta + a] \\ [p\sin\theta + q\cos\theta + b] \end{pmatrix} \tag{4}$$

In general, this function is not bijective. However, by setting $T^{-1} : \mathbb{Z}^2 \to \mathbb{Z}^2$ as $T^{-1} = D \circ \mathcal{T}^{-1}$, *i.e.*, by considering the standard backward mapping, it is possible to define the digital transformed image $I \circ T^{-1} : \mathbb{Z}^2 \to \mathbb{V}$ with respect to T. In the sequel of this article, we focus on such digital rigid transformations. From this point on – for the sake of readability and without loss of correctness – we will note T instead of T^{-1}, due to the bijectivity of $\mathcal{T}$ and $\mathcal{T}^{-1}$.

From a theoretical point of view, the above notions (images, rigid transformations) are defined on $\mathbb{Z}^2$ and $\mathbb{R}^2$. Practically, our purpose is however to study rigid transformations on images of *finite* size. Under this hypothesis, only some digital rigid transformations are relevant, namely those that actually have an effect on such finite images. We focus on this finite case, and we assume that digital images are defined on subsets of $\mathbb{Z}^2$ of size $N \times N$. Without loss of generality, a digital image I is then written as $I : \mathbb{S} \to \mathbb{V}$ for $\mathbb{S} = [\![0, N]\!]^2 \subset \mathbb{Z}^2$.

2.2 Discontinuities of digital rigid transformations

In $\mathbb{R}^2$, any rigid transformation $\mathcal{T}_{ab\theta}$ is a continuous function (see Equation (2)). However, this notion of continuity is lost once the function is digitized. Indeed, due to the digitization process involved in the definition of digital rigid transformations (see Equation (4)), the parameter space (a, b, θ) of rigid transformations is divided into 3D open cells, in each of which the function $(a, b, \theta) \mapsto T_{ab\theta} = D \circ \mathcal{T}_{ab\theta}$ is constant. In particular, these 3D open cells are separated by 2D closed cells corresponding to rigid transformations that map at least one integer coordinate point onto a *half-grid* point (see Fig. 1). Such transformations, which lead to discontinuities within the parameter space, are called *critical transformations*.

Definition 1 (Critical transformation [18]) *Let $(a, b, \theta) \in \mathbb{R}^2 \times [0, 2\pi[$, and $\mathcal{T}_{ab\theta} : \mathbb{R}^2 \to \mathbb{R}^2$ be its associated rigid transformation. We say that $\mathcal{T}_{ab\theta}$ is a* critical transformation *if there exists $\boldsymbol{p} \in \mathbb{Z}^2$ such that $\mathcal{T}_{ab\theta}(\boldsymbol{p}) \in \mathcal{H}$, where $\mathcal{H}$ is the half-grid defined by*

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

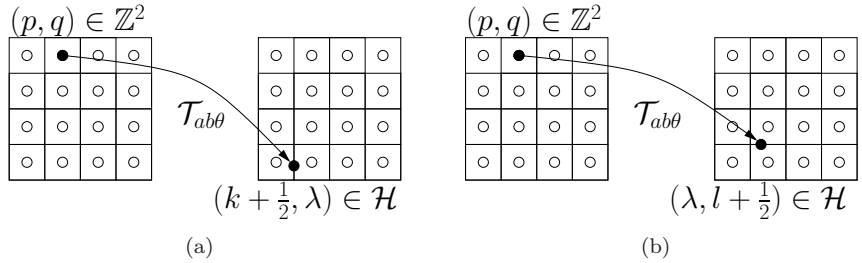


Fig. 1 Examples of critical transformations $\mathcal{T}_{ab\theta}$, each of which maps at least one integer-coordinate point onto a "vertical" (a) or "horizontal" (b) half-grid point. The integer-coordinate points in $\mathbb{Z}^2$ are depicted by dots, while the half-grid points are depicted by lines.

Note that the half-grid $\mathcal{H}$ corresponds to the boundaries of the Voronoi cells of $\mathbb{R}^2$ induced by $\mathbb{Z}^2$.

More precisely, for each $\boldsymbol{p} = (p,q) \in \mathbb{Z}^2$ that is mapped onto a half-grid point which can be either vertical $(k+\frac{1}{2},\lambda) \in \mathcal{H}$ or horizontal $(\lambda,l+\frac{1}{2}) \in \mathcal{H}$, we have a set of critical transformations, denoted either Φ_{pqk} or Ψ_{pql}, defined by the set of (a,b,θ) that satisfies one of the following formulas :

$$\left| \begin{array}{ll} \Phi_{pqk} : \mathbb{R} \times [0,2\pi[& \longrightarrow \mathbb{R} \\ \quad (b,\theta) & \longmapsto a = \phi_{pqk}(\theta) = k + \frac{1}{2} + q\sin\theta - p\cos\theta \end{array} \right. \tag{5}$$

$$\left| \begin{array}{ll} \Psi_{pql} : \mathbb{R} \times [0,2\pi[& \longrightarrow \mathbb{R} \\ \quad (a,\theta) & \longmapsto b = \psi_{pql}(\theta) = l + \frac{1}{2} - p\sin\theta - q\cos\theta \end{array} \right. \tag{6}$$

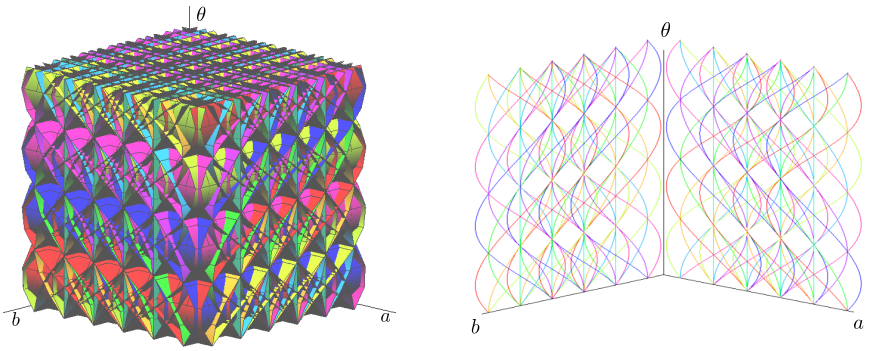
The 2D surfaces Φ_{pqk} (resp. Ψ_{pql}) defined in the parameter space (a,b,θ) are called *tipping surfaces* [18]. Their respective cross-sections ϕ_{pqk} (resp. ψ_{pql}) on the 2D plane (a,θ) (resp. (b,θ)) are called *tipping curves*. These tipping surfaces/curves, which correspond to the discontinuities of the digital rigid transformations, expressed in the parameter space (a,b,θ), are illustrated in Fig. 2. It is important to remark that the tipping surfaces Φ_{pqk} and Ψ_{pql} can be straightforwardly recovered by extruding the tipping curves ϕ_{pqk} and ψ_{pql}, respectively.

2.3 Partition of the parameter space and DRT graph

As a result of the discontinuity of digital rigid transformations induced by the digitization process (Equation (1)), it is possible that some distinct rigid transformations (Equation (2)) be mapped onto a same digital rigid transformation (Equation (4)). This leads to considering equivalence classes between transformations, which are defined by the following relation

$$\left(\mathcal{T}_{ab\theta} \sim \mathcal{T}_{a'b'\theta'}\right) \Longleftrightarrow \left(T_{ab\theta} = T_{a'b'\theta'}\right) \tag{7}$$

It has to be noticed that this equivalence relation is only defined between *non-critical* rigid transformations. As stated above, it is possible to identify



(a) (b)

Fig. 2 (a) Tipping surfaces in the 3D parameter space (a, b, θ), and (b) their cross-sections, namely tipping curves, in the 2D planes (a, θ) and (b, θ).

a rigid transformation with its triplet of parameters (a, b, θ). In this context, the equivalence classes of transformations, called the *discrete rigid transformations*[2] (*DRTs*), can be modeled by 3D open cells in this parameter space, whose boundaries are 2D tipping surfaces defined above (see Fig. 2(a)). In other words, the parameter space (a, b, θ) of rigid transformations is partitioned into disjoint sets of *non-critical transformations*, each of which is associated to exactly one DRT, and bounded by the surfaces modeling critical transformations.

We have shown in [18] that the subdivision of this parameter space could be modeled by using a dual combinatorial structure, that maps each 3D cell (*i.e.*, each DRT) onto a 0D point and each 2D tipping-surface segment (linked to a set of critical transformations) onto a 1D edge. The resulting structure is called a *DRT graph* (see Fig. 3).

Definition 2 (DRT graph [18]) *Given a set of tipping surfaces, Φ_{pqk} and Ψ_{pql}, the graph $G = (V, E)$ associated to DRTs induced by those Φ_{pqk} and Ψ_{pql} is defined in the following way:*

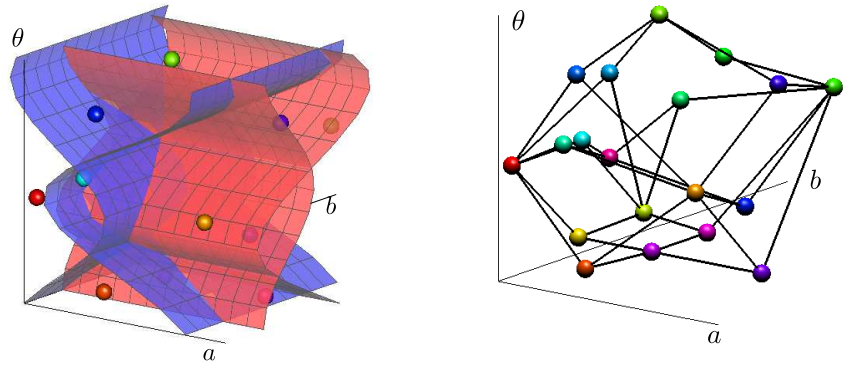
- *each vertex $v \in V$ models a 3D open cell associated to a DRT;*
- *each labelled edge $e = (u, w, f) \in E$ (where f is either Φ_{pqk} or Ψ_{pql}) models the tipping surface f between two adjacent vertices $v, w \in V$.*

This graph G is called a DRT graph*.*

In [18], we have proved that the space complexity of the DRT graph for any set $\mathbb{S}$ of size $N \times N$ is polynomial. An exact computation algorithm is proposed to build this graph in linear time with respect to the size of the graph.

Property 3 ([18]) *The DRT graph associated to a digital image of size $N \times N$ has a space complexity of $\mathcal{O}(N^9)$.*

[2] Contrarily to the terminology frequently used in the literature, the term *digital* refers here to the digitization process D defined in Equation (4), while the term *discrete* refers to the combinatorial structure induced by this operator D.



(a) (b)

Fig. 3 (a) The parameter space of rigid transformations subdivided by four tipping surfaces, and (b) the associated DRT graph.

The DRT graph models "neighbouring" relationships between DRTs. Indeed, by associating a resulting digital transformed image to each 3D open cell/DRT, the existence of a 2D surface between two cells indicates that the associated transformed images differ in exactly one pixel among the N^2 ones. More precisely, let us consider an edge $e = (v, w, f) \in E$ between two distinct vertices $v, w \in V$. The function f (that is either equal to Φ_{pqk} or Ψ_{pql}) indicates that exactly one point $\boldsymbol{p} = (p, q) \in \mathbb{S}$ differs between the transformed images corresponding to the DRTs v and w. Practically, let $\boldsymbol{q}$ be the point with coordinates (k, l) with respect to f. Let I_v and I_w be the transformed images corresponding to the vertices v and w respectively. The value of $\boldsymbol{p}$ at the vertex v is defined by $I_v(\boldsymbol{p}) = I(\boldsymbol{q})$ where $I : \mathbb{S} \to \mathbb{V}$ is the original image. After the elementary change along edge e, we obtain a new transformed image I_w by simply setting the pixel value at $\boldsymbol{p}$ to $I_w(\boldsymbol{p}) = I(\boldsymbol{q}+\boldsymbol{\delta})$ where $\boldsymbol{\delta} = (\pm 1, 0)$ or $(0, \pm 1)$ with respect to f. In this way, one can generate all the transformed images of I by incrementally and exhaustively scanning the associated DRT graph. This property, exemplified in Fig. 4, was used in [19] for verifying the topological invariance of digital images under rigid transformations.

3 Constraints and feasible rigid transformation sets

The DRT graph is highly complex in space and time, which makes its practical construction and handling challenging for large images. In the sequel, we investigate how the use of constraints may reduce these complexities. More precisely, we focus on *pixel-invariance* constraints which consist of enforcing correspondence between points in the initial and transformed image. In particular, we expect these constraints to reduce the size of the parameter space

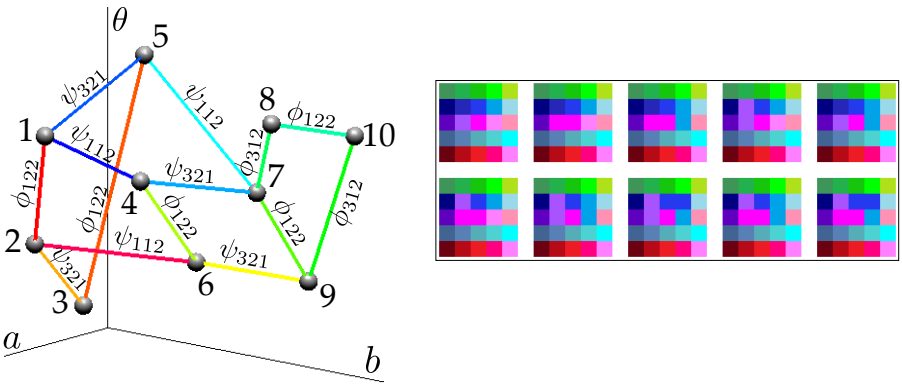


Fig. 4 Left: a part of a DRT graph in which a vertex/DRT represents a digital transformed image and an edge $e = (v, w, f)$ between two vertices v and w indicates that one pixel value is different between the associated transformed images I_v and I_w (see text). Right: the transformed images associated to the vertices of the DRT graph (in left). The images from upper-left to bottom-right correspond to the vertices ordered from 1 to 10 in the graph.

partition, in terms of the number of 3D cells, and therefore the size of the associated DRT graph.

3.1 Pixel-invariance constraints and interpretation in the parameter space

In the context of rigid transformations in $\mathbb{R}^2$, enforcing the correspondence between two points $\boldsymbol{p}$ and $\boldsymbol{p}'$ in the initial and in the transformed spaces respectively, leads to restricting the number of authorised transformations. More precisely, from Equation (2) we obtain, for a given pair of corresponding points, two equations representing 2D trigonometric surfaces, that intersect to provide a 1D trigonometric curve, which models an (affine) space of rotations (see Fig. 5(a)).

A unique constraint then leads to an infinite space of transformations, that we call *feasible transformations*. In order to obtain a finite space of feasible transformations, we then need two (distinct) constraints, *i.e.*, two pairs of corresponding points $(\boldsymbol{p}, \boldsymbol{p}')$ and $(\boldsymbol{q}, \boldsymbol{q}')$. If these pairs are well chosen, *i.e.*, they satisfy the isometric properties of rigid transformations ($\|\boldsymbol{p}-\boldsymbol{p}'\|_2 = \|\boldsymbol{q}-\boldsymbol{q}'\|_2$), then the space of feasible transformations is restricted to a unique transformation (see Fig. 5(b)) that corresponds to the intersection of the two 1D curves induced by these constraints. Otherwise, the space of feasible transformations is empty. More generally, setting the correspondence between $k \geq 2$ distinct couples of points $(\boldsymbol{p}_i, \boldsymbol{p}'_i)$, for $i \in [\![1, k]\!]$, restricts the authorised transformations to at most a single feasible one.

In contrast, in the context of digital rigid transformations (see Equation (4)), the way to restrict transformations under similar constraints is more permissive. Indeed, when setting the correspondence between one or several pairs of points $(\boldsymbol{p}_i, \boldsymbol{p}'_i)$ of $\mathbb{Z}^2$, a larger space of rigid transformations remains

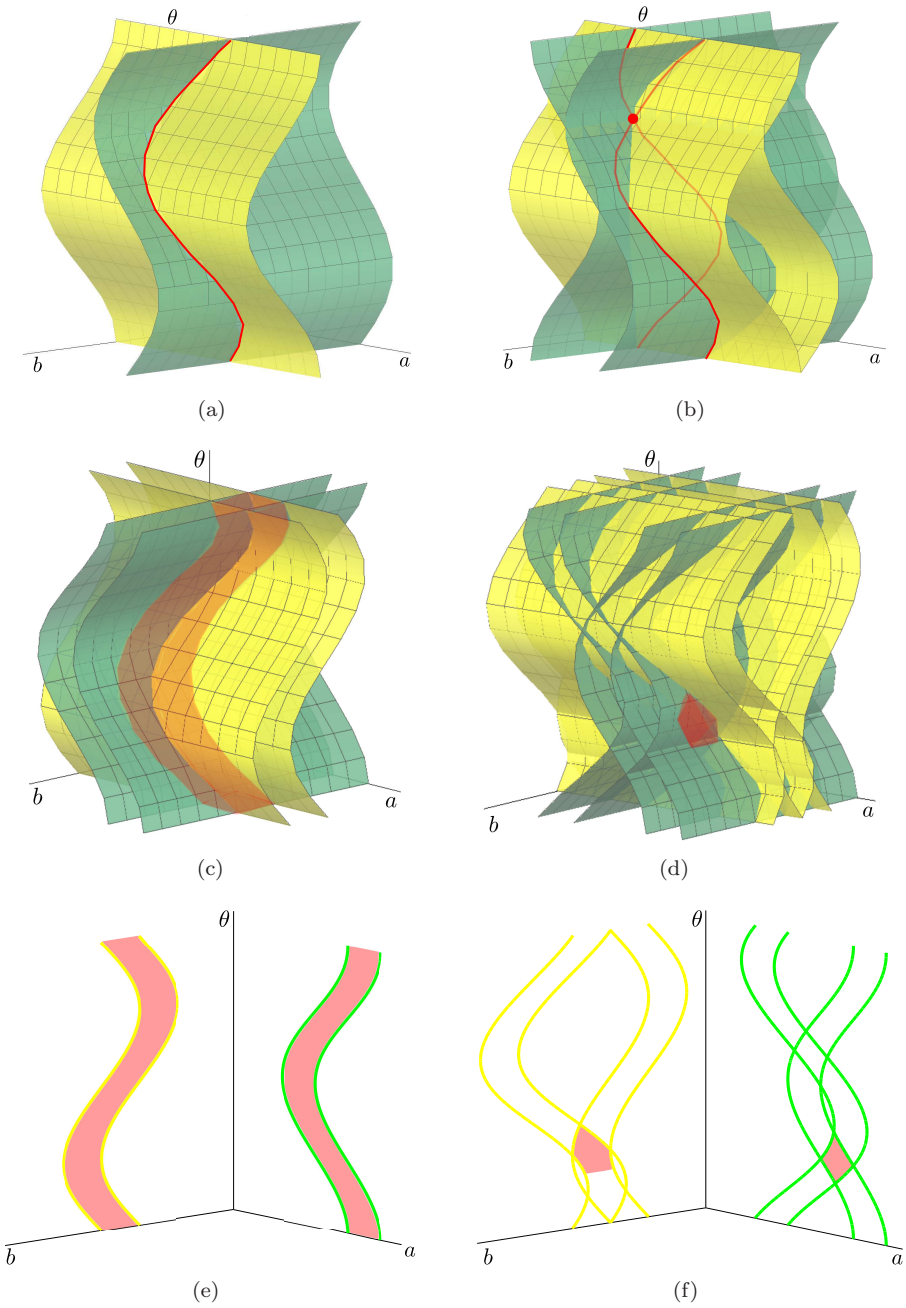


Fig. 5 Feasible rigid transformations induced by geometric constraints in the continuous (a,b) and discrete (c–f) frameworks. (a) Transformations with one point correspondence (red curve). (b) Transformation with two point correspondences (red dot at the intersection of the two red curves). (c) Transformations with one pixel correspondence (red tube-like volume). (d) Transformations with two pixel correspondences (red volume). (e,f) Projection/intersection of red volume parts of (c,d) respectively to the planes (a, θ) and (b, θ) with the associated tipping curves.

valid (see Fig. 5(c,d)). Such constraints, which rely on the pixel decomposition of the image, are called *pixel-invariance constraints* and are defined as follows.

Definition 4 (Pixel invariance constraints) *Let* $\boldsymbol{p} = (p, q)$ *and* $\boldsymbol{p}' = (p', q')$ *in* $\mathbb{S} \subset \mathbb{Z}^2$*, where* $\mathbb{S}$ *is of size* $N \times N$*. There exists a* pixel-invariance constraint *between* $\boldsymbol{p}$ *and* $\boldsymbol{p}'$ *if the authorised digital rigid transformations* T *between* $\boldsymbol{p}$ *and* $\boldsymbol{p}'$ *satisfy the equality* $T(\boldsymbol{p}) = \boldsymbol{p}'$*,* i.e.*, if*

$$p' - \frac{1}{2} < p\cos\theta - q\sin\theta + a < p' + \frac{1}{2} \tag{8}$$

$$q' - \frac{1}{2} < p\sin\theta + q\cos\theta + b < q' + \frac{1}{2} \tag{9}$$

More generally, there exist pixel-invariance constraints between two sets $\{\boldsymbol{p}_i\}_{i=1}^m$ *and* $\{\boldsymbol{p}'_i\}_{i=1}^m$ *(*$m \geq 1$*) if* $T(\boldsymbol{p}_i) = \boldsymbol{p}'_i$ *(*i.e.*, if Inequalities* (8)–(9) *are satisfied) for every* $i \in [\![1, m]\!]$*.*

In the absence of constraints, the 3D parameter space (a, b, θ) induced by the subset of size $N \times N$ where the image is defined, is divided into cells whose boundaries are the tipping surfaces Φ_{pqk} and Ψ_{pql}, with $p, q \in [\![0, N-1]\!]$ and $k, l \in [\![0, N]\!]$. In this case, the *whole* parameter space models adequate rigid transformations.

Under a pixel-invariance constraint, some of the digital rigid transformations may become unfeasible. In other words, only a part of the parameter space – namely the subspace of the parameters (a, b, θ) that satisfy this constraint – remains valid. From the definition of pixel-invariance constraint, provided by Inequalities (8)–(9), this parameter subspace is defined by the intersection of four half-spaces associated to four tipping surfaces. This is visually illustrated in Fig. 5(c,e). The graph modelling the subdivision of such subspace is in particular a part of the whole DRT graph, induced by the pixel-invariance constraint.

3.2 Feasible rigid transformation sets

More generally, if a set $\mathcal{P}$ of m pixel-invariance constraints is provided, the parameter subspace of relevant transformations is defined as the intersection of m 3D regions induced by these constraints, *i.e.*, as the intersection of $4m$ half-spaces defined by Inequalities (8)–(9).

Let $p, q \in [\![0, N-1]\!]$ and $k, l \in [\![0, N]\!]$. Let us consider the functions $V_{pqk}, H_{pql} : \mathbb{R} \times [0, 2\pi[\to \mathbb{R}$ respectively defined by

$$V_{pqk}(a, \theta) = a - \phi_{pqk}(\theta) \tag{10}$$

$$H_{pql}(b, \theta) = b - \psi_{pql}(\theta) \tag{11}$$

where ϕ_{pqk} and ψ_{pql} are two tipping curves (see Equations (5)–(6)). We then define the half-spaces induced by the associated tipping surfaces Φ_{pqk} and Ψ_{pql}

with respect to inequalities (8)–(9) as

$$V^+_{pqk} = \{(a,b,\theta) \mid V_{pqk}(a,\theta) > 0\} \tag{12}$$

$$V^-_{pqk} = \{(a,b,\theta) \mid V_{pqk}(a,\theta) < 0\} \tag{13}$$

and

$$H^+_{pql} = \{(a,b,\theta) \mid H_{pql}(b,\theta) > 0\} \tag{14}$$

$$H^-_{pql} = \{(a,b,\theta) \mid H_{pql}(b,\theta) < 0\} \tag{15}$$

The notion of a *feasible rigid transformation set* is then defined as follows.

Definition 5 (Feasible rigid transformation set) *Let $\mathcal{P} = \{(\boldsymbol{p}_i, \boldsymbol{p}'_i)\}_{i=1}^m$ ($m \geq 1$) be a set of pixel-invariance constraints with $\boldsymbol{p}_i = (p_i, q_i)$ and $\boldsymbol{p}'_i = (p'_i, q'_i)$ in $\mathbb{S} \subset \mathbb{Z}^2$. The* feasible rigid transformation set *(*FRTS*) associated to $\mathcal{P}$ is the subspace $\mathcal{R} \subset \mathbb{R}^2 \times [0, 2\pi[$ of the parameter space (a,b,θ) defined by*

$$\mathcal{R} = \bigcap_{i \in [\![1,m]\!]} \left(V^+_{p_i q_i p'_i} \cap V^-_{p_i q_i p'_i+1} \cap H^+_{p_i q_i q'_i} \cap H^-_{p_i q_i q'_i+1} \right) \tag{16}$$

For a single pixel-invariance constraint (*i.e.*, for $m = 1$), the FRTS forms a "tube" in the parameter space (a,b,θ) (see Fig. 5(c)). For two – or more – pixel-invariance constraints (*i.e.*, for $m \geq 2$), the FRTS forms a bounded and connected set (see Fig. 5(d)), or possibly becomes empty.

The FRTS is generated by m pixel-invariance constraints, and divided into 3D cells whose boundaries are the tipping surfaces induced by at most (N^2-m) unconstrained pixels of the given image of size $N \times N$. It has to be recalled that each cell contains a set of rigid transformations that provide the same digital transformation, namely a discrete rigid transformation (DRT). In particular, the combinatorial structure modeling the subdivision of an FRTS into DRTs is represented by a part of the DRT graph, as defined in Section 2.3, and is called a *feasible discrete rigid transformation graph* (*FDRT graph*, for short). More explanation as well as the construction of this graph is given in Section 4.

We now recall the notion of directional convexity, and show that any FRTS is directionally convex. This property will be used in the next section to study the combinatorial structure of DRTs under pixel-invariance constraints.

Definition 6 (Directional convexity [16]) *A region $R \subseteq \mathbb{R}^n$ in an n-variable space $(x_1, \ldots, x_n)$ is* x_k-convex *(with $1 \leq k \leq n$) if, for any two points $\boldsymbol{p_1}, \boldsymbol{p_2} \in R$ such that the segment $[\boldsymbol{p_1}\boldsymbol{p_2}] = \{\alpha \boldsymbol{p_1} + (1-\alpha)\boldsymbol{p_2} \mid \alpha \in [0,1]\}$ is parallel to the x_k-axis, $[\boldsymbol{p_1}\boldsymbol{p_2}]$ is included in R.*

Property 7 *Any FRTS is both a- and b-convex in the space (a,b,θ).*

Proof This is a direct consequence of the fact that any FRTS is the intersection of half-spaces which are both a- and b-convex (see Equation (16)). ■

Based on the relations that link tipping surfaces and tipping curves (see Equations (5)–(6) and Figs. 2 and 5(c–f)), it is plain that an FRTS, defined

in the parameter space (a, b, θ) by Equation (16), can be fully described from its two projections $\mathcal{R}_V$ and $\mathcal{R}_H$ on the planes (a, θ) and (b, θ) respectively, defined as

$$\mathcal{R}_V = \bigcap_{i \in [\![1,m]\!]} \left(v^+_{p_i q_i p'_i} \quad \cap \quad v^-_{p_i q_i p'_i + 1} \right) \tag{17}$$

$$\mathcal{R}_H = \bigcap_{i \in [\![1,m]\!]} \left(h^+_{p_i q_i q'_i} \quad \cap \quad h^-_{p_i q_i q'_i + 1} \right) \tag{18}$$

where v^*_{pqk} (resp. h^*_{pql}) is the cross-section of v^*_{pqk} (resp. h^*_{pql}) with the plane (a, θ) (resp. (b, θ)). In this context, v^+_*, h^+_* are called *upper half-planes*[3] and v^-_*, h^-_* *lower half-planes*.

From Property 7, it is obvious that $\mathcal{R}_V$ (resp. $\mathcal{R}_H$) has a pair of upper and lower half-planes (v^+, v^-) (resp. (h^+, h^-)) as the upper and lower parts of the boundary for each θ. Thus, the boundary of $\mathcal{R}_V$ (resp. $\mathcal{R}_H$) consists of two sets of half-planes:

- the *upper boundary set* **U** containing only upper half-planes;
- the *lower boundary set* **L** containing only lower half-planes.

From Property 7, we can also derive the following corollary for $\mathcal{R}_V$; a similar corollary is established for $\mathcal{R}_H$ as well.

Corollary 8 *Let $\mathcal{R}_V$ be the projection of an FRTS $\mathcal{R}$ (as defined in Equation* (17)*), and* **U** *(resp.* **L***) be the upper (resp. lower) boundary of $\mathcal{R}_V$. Then* **U** *(resp.* **L***) always contains at least* one *upper (resp. lower) half-plane.*

We now derive the following result related to the connectedness of an FRTS, which will be useful in the following section. Here, instead of the parameter space $\mathbb{R}^2 \times [0, 2\pi[$ of (a, b, θ), we consider – without loss of correctness – the quotient space $\mathbb{R}^3/\!\sim$ where $(a, b, \theta) \sim (a, b, \theta + 2\pi)$.

Property 9 *An FRTS is connected in the quotient space $\mathbb{R}^3/\!\sim$ of the parameter space (a, b, θ).*

Proof If $m = 1$, Inequalities (8)–(9), imply that the FRTS is the Minkowski addition between a 1D (connected) trigonometric curve – defined as a function from the (a, b) space to the θ one – and a (connected) square pattern $]-\frac{1}{2}, \frac{1}{2}[^2 \subset \mathbb{R}^2$ defined in the (a, b) space. The FRTS is then necessarily connected.

Let us now suppose that $m = 2$. Let $(\boldsymbol{p}_i, \boldsymbol{p}'_i)$, for $i = 1, 2$, be the two pixel-invariance constraints that generate the FRTS $\mathcal{R}$. As explained above, $\mathcal{R}$ can be described from its two projections $\mathcal{R}_V$ and $\mathcal{R}_H$ on the planes (a, θ) and (b, θ) respectively by tipping curves. Let us first consider $\mathcal{R}_V$; from Equation (17), $\mathcal{R}_V$ has two upper half-planes $\mathbf{U} = \{v^+_{p_1 q_1 q'_1}, v^+_{p_2 q_2 q'_2}\}$ and two lower half-planes

[3] Note that the term *half-space* is used for 3D regions induced by the tipping surfaces in the parameter space (a, b, θ), while the term *half-plane* is used for 2D regions induced by the tipping curves in the plane either (a, θ) or (b, θ).

$\mathbf{L} = \{v^-_{p_1q_1q'_1+1}, v^-_{p_2q_2q'_2+1}\}$. Then, any pair of upper and lower half-planes of $\mathcal{R}_V$, $(v^+, v^-) \in \mathbf{U} \times \mathbf{L}$, has the intersection $v^+ \cap v^-$ that is connected in $\mathbb{R}^2/\!\sim$ of the parameter space (a, θ), where $(a, \theta) \sim (a, \theta + 2\pi)$, as the associated tipping curves have at most two intersections for $\theta \in [0, 2\pi[$ (see [18, Prop. 2, Cor. 2]). For instance, each of $v^+_{p_1q_1q'_1} \cap v^-_{p_2q_2q'_2+1}$ and $v^+_{p_2q_2q'_2} \cap v^-_{p_1q_1q'_1+1}$ is a connected region in the quotient space $\mathbb{R}^2/\sim$. However, the intersection of these two regions, *i.e.* $\mathcal{R}_V$, may give at most two connected regions in the quotient space $\mathbb{R}^2/\!\sim$. A similar result is obtained for $\mathcal{R}_H$. We now show that the 3D intersection of (the extrusion of) $\mathcal{R}_V$ and $\mathcal{R}_H$ gives only one admissible connected region in the quotient space $\mathbb{R}^3/\sim$. As we know that the intersection of $\mathcal{R}_H$ and $\mathcal{R}_V$, *i.e.* $\mathcal{R}$, is never empty, there always exist $(r_i, s_i) \in]p'_i - \frac{1}{2}, p'_i + \frac{1}{2}[\times]q'_i - \frac{1}{2}, q'_i + \frac{1}{2}[$ for $i = 1, 2$ such that

$$r_1 = p_1 \cos\theta - q_1 \sin\theta + a \tag{19}$$
$$s_1 = p_1 \sin\theta + q_1 \cos\theta + b \tag{20}$$
$$r_2 = p_2 \cos\theta - q_2 \sin\theta + a \tag{21}$$
$$s_2 = p_2 \sin\theta + q_2 \cos\theta + b \tag{22}$$

At any intersection between (19) and (21) in the plane (a, θ), the following equation must be satisfied:

$$K - P\cos\theta + Q\sin\theta = 0 \tag{23}$$

by setting $P = p_1 - p_2$, $Q = q_1 - q_2$ and $K = r_1 - r_2$. Similarly, at any intersection between (20) and (22) in the plane (b, θ), the following equation must be satisfied:

$$L - P\sin\theta - Q\cos\theta = 0 \tag{24}$$

by additionally setting $L = s_1 - s_2$. The system of linear equations (23),(24) has a determinant equal to $P^2 + Q^2 \neq 0$, since the two pixel-invariance constraints are distinct. It then admits exactly one solution for the pair of unknowns $\cos\theta$ and $\sin\theta$, and thus at most one solution for θ in $[0, 2\pi[$. Since the FRTS $\mathcal{R}$ contains such a solution, $\mathcal{R}$ must be connected. The same result for $m > 2$ follows by induction. ■

4 Combinatorial structure of feasible discrete rigid transformations

An FRTS contains the rigid transformations that satisfy some given pixel-invariance constraints. It can then be subdivided into DRTs (see Section 2.3). This section presents a method for constructing the combinatorial structure of DRTs in an FRTS (namely, the FDRT graph) based on the idea of the sweeping algorithm [18] recalled in Section 4.1. This algorithm is used for building a graph modeling a subdivision of the parameter space from a given set of tipping surfaces without considering the FRTS. Then, the construction of a FDRT graph is performed by following these three successive steps:

(i) finding the boundaries of the FRTS in the parameter space (Section 4.2);
(ii) finding the tipping surfaces passing through this FRTS (Section 4.3) and their intersecting points (Section 4.4);
(iii) constructing the associated DRT graph (Section 4.5).

In this FDRT graph construction algorithm, the sweeping method (Section 4.1) is required for the step (*i*) and (*iii*) with some modifications.

4.1 Incremental construction of a discrete rigid transformation graph based on a sweeping method

In [18], we showed how to build a DRT graph using an incremental algorithm, that mainly relies on the algorithmic notion of *surface arrangement* [5,26]. A surface arrangement is defined as a decomposition of the space $\mathbb{R}^3$ into cells by a finite set of surfaces. Such a decomposition generates four types of cells: 0D vertices, 1D arcs, 2D surfaces and 3D regions. Surface arrangement algorithms present a polynomial complexity $\Omega(n^4)$ [26], where n is the number of surfaces. However, we are only interested in the information of regions (3D cells) and faces (2D cells) in the arrangement. In this specific case – when the surfaces are tipping surfaces – a better complexity for building the DRT graph in $\mathcal{O}(n^3)$ can be achieved [18]. We now first describe a DRT graph construction method, in which the input and output are as follows:

- **Input**: a set of tipping surfaces S;
- **Output**: the DRT graph modeling G, the subdivision of the parameter space (a, b, θ) induced by S.

As described in Section 2.2, 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 (5)–(6) (see Fig. 2). Relying on this property, the subdivision of the parameter space (a, b, θ) by these surfaces can be fully described from their two cross-sections in the planes (a, θ) and (b, θ), respectively expressed by two sets of tipping curves [18]. This leads to a constructive algorithm with a better complexity. More precisely, we first consider the structure of the graphs in the 2D planes (a, θ) and (b, θ), and then combine them to build the complete DRT graph G. In the sequel, we first recall the principles of this algorithm and its implementation for building G in the 2D planes. Then, we extend it to the 3D parameter space to compute the complete DRT graph.

4.1.1 2D sweeping method

The sweeping method in 2D then consists of sweeping a *cut* across all tipping curves in the plane – either (a, θ) or (b, θ) – from $\theta = 0$ to 2π. Such a cut is denoted by γ and defined as a monotonic line [7] intersecting exactly once each tipping curve in the plane. The monotonicity with respect to θ of the cut is a result of a- and b-convexity in the space (a, b, θ) (see Property 7). A cut is

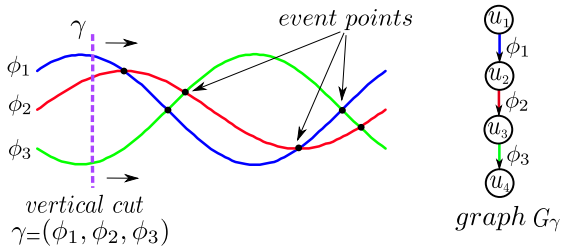


Fig. 6 Example of a cut γ and its associated graph G_γ.

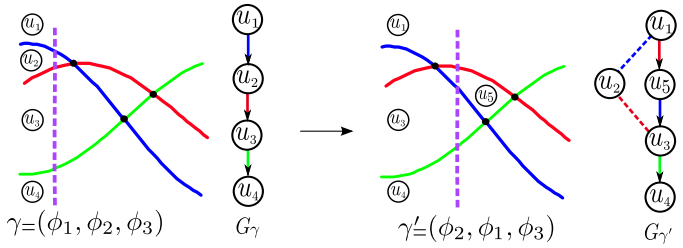


Fig. 7 Progress of the cut at an event point by which the cut is updated and the corresponding graph is modified.

then modeled by its sequence of intersecting tipping curves (see Fig. 6). This sequence can be conveniently represented as a directed graph as follows.

Definition 10 *Let $\gamma = (\psi_1, \psi_2, \ldots)$ be a sequence corresponding to a cut, where $\psi_1, \psi_2, \ldots$ are tipping curves. A graph $G_\gamma = (V_\gamma, E_\gamma)$ with respect to γ consists of:*

- *a set of vertices $V_\gamma = \{v_0, v_1, \ldots\}$ each of which corresponds to an interval on γ separated by tipping curves; and*
- *a set of labelled edges $E_\gamma = \{(v_0, v_1, \psi_1), (v_1, v_2, \psi_2), \ldots\}$, for which each edge $(u, w, f) \in E_\gamma$ connects two vertices $u, w \in V_\gamma$ separated by the tipping curve f, which is considered as an edge label.*

From an algorithmic point of view, the graph G_γ is directed, since the edge direction information is necessary for updating the sequence of γ during its progression with respect to the θ-axis from left to right (*i.e.*, form 0 to 2π). In this context, the edge direction in G_γ is specified by the sequence order of γ (see Definition 10); *e.g.*, the elements of E_γ are ordered in the same way as γ as illustrated in Fig. 6.

While moving the cut γ, its sequence changes only at intersections of tipping curves, called *event points*. When a cut reaches an event point, the algorithm performs an update of its sequence, and generates new vertices and edges in the graph (see Fig. 7). This constitutes an *elementary step* of the algorithm. Practically, it is only required to maintain a set of sorted event points with respect to θ, and to progress the cut in their increasing order to build the graph incrementally.

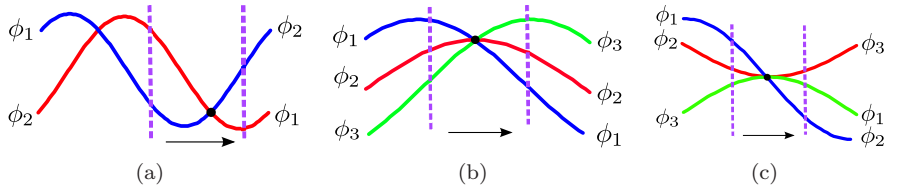


Fig. 8 Examples of a *simple* event point generated by only two tipping curves (a), and *degeneracies* (b,c) generated by more than two tipping curves.

In the context of tipping curves, the event points can be classified into

- *simple* cases: only two tipping curves intersect at an event point (Fig. 8(a));
- *degenerate* or *non-simple* cases: more than two tipping curves intersect (or, are intersecting and/or tangent) at an event point (Fig. 8(b,c)).

In the sequel, we only deal with the simple cases. A way to deal with the degenerate cases and other details can be found in [18].

4.1.2 3D sweeping method

For building a DRT graph G in the 3D parameter space (a, b, θ), two *cuts* are used such that each cut sweeps in either the plane (a, θ) or (b, θ). We denote those cuts by γ_a and γ_b respectively. For each update of the cuts, γ_a and γ_b, the associated graphs, G_{γ_a} and G_{γ_b}, are respectively modified, so that a part of G is generated. We call such a part of G a *partial graph*, denoted by δG. In fact, δG is a combination of the two graphs G_{γ_a} and G_{γ_b} as follows (and also see Fig. 9).

Definition 11 *The partial graph $\delta G = (\delta V, \delta E)$ is generated from $G_{\gamma_a} = (V_{\gamma_a}, E_{\gamma_a})$ and $G_{\gamma_b} = (V_{\gamma_b}, E_{\gamma_b})$, such that:*

- $\delta V = \{(v_a, v_b) \mid v_a \in V_{\gamma_a}, v_b \in V_{\gamma_b}\}$, *and*
- $\delta E = \{((u_1, v), (u_2, v), \psi_u) \mid u_1, u_2 \in V_{\gamma_a}, v \in V_{\gamma_b}, (u_1, u_2, \psi_u) \in E_{\gamma_a}\} \cup \{((u, v_1), (u, v_2), \psi_v) \mid v_1, v_2 \in V_{\gamma_b}, u \in V_{\gamma_a}, (v_1, v_2, \psi_v) \in E_{\gamma_b}\}$.

When the i-th elementary step is applied to G_{γ_a} or G_{γ_b}, the sweep progresses as the partial graph δG_i is generated and integrated in G for constructing the final graph as well. The following proposition was originally proposed in [18].

Proposition 12 ([18]) *Let S be a set of tipping surfaces, e be the total number of ordered event points, and G be a DRT graph modeling the subdivision of the parameter space by S. We have*

$$G = \bigcup_{i \in [\![1, e]\!]} \delta G_i \tag{25}$$

where δG_i is a partial graph at the i-th elementary step.

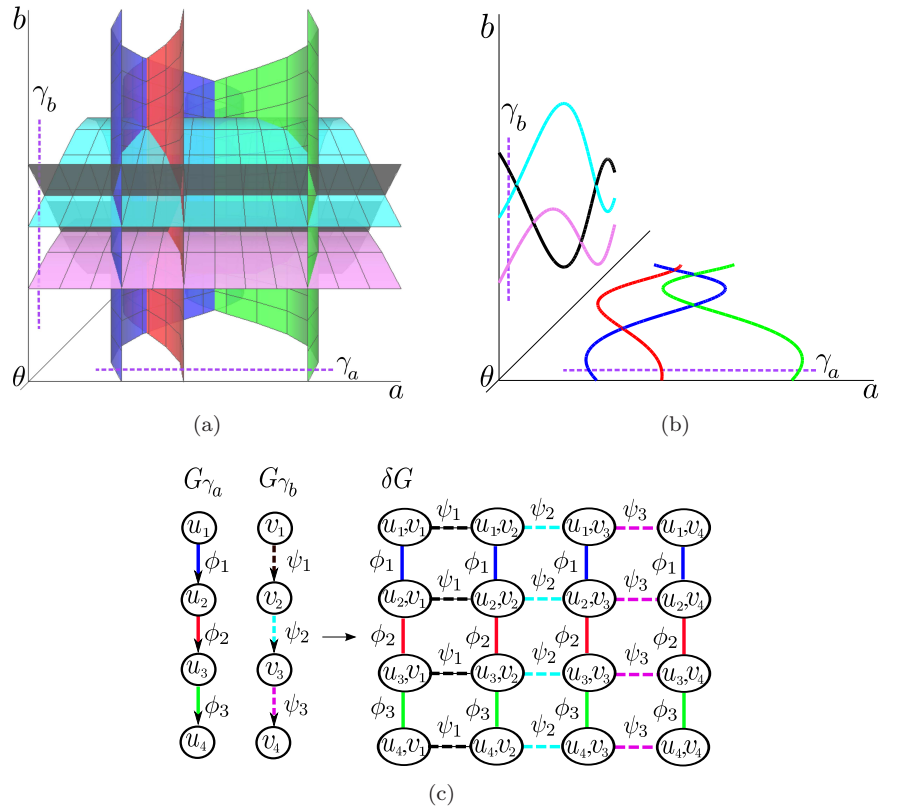


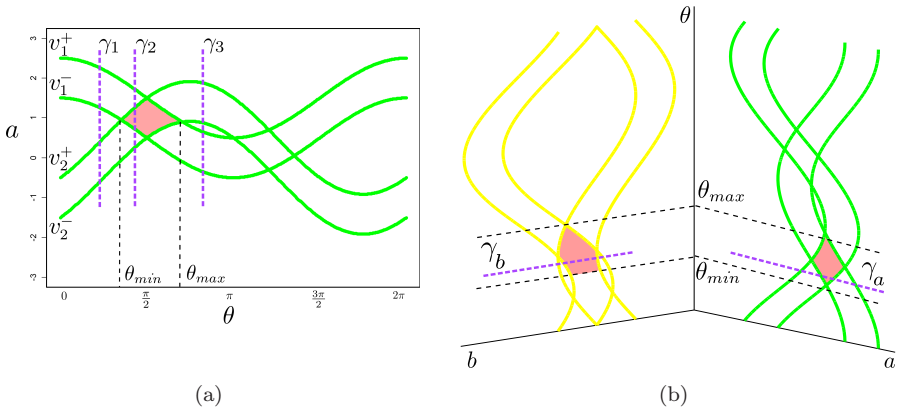
Fig. 9 Generation of a partial graph δG from two graphs G_{γ_a} and G_{γ_b} associated to cuts γ_a and γ_b respectively.

Note that a partial graph δG_i is a directed graph by construction. However the final graph G is not directed, so that we do not keep directions while integrating δG_i into G. More algorithmic details on the sweeping algorithm for tipping surfaces can be found in [18].

4.2 Finding the boundary of a feasible rigid transformation set

It is possible to describe an FRTS $\mathcal{R}$ defined from a set $\mathcal{P}$ of pixel-invariance constraints using a set of half-spaces constituting only the boundary of $\mathcal{R}$, instead of using all the half-spaces from $\mathcal{P}$ in Definition 5. This section explains how to find such a set of half-spaces, by using the above sweeping algorithm. We have here the input and output as follows:

- **Input**: A set of m pixel-invariance constraints $\mathcal{P} = \{(\boldsymbol{p}_i, \boldsymbol{p}'_i)\}_{i=1}^{m}$.
- **Output**: The boundary set $\mathbf{B} = \mathbf{U} \cup \mathbf{L}$ of the FRTS $\mathcal{R}$ induced by $\mathcal{P}$.



(a) (b)

Fig. 10 (a) Progression of a cut γ in the cross-section $\mathcal{R}_V$ of an FRTS $\mathcal{R}$ (in red) on the plane (a, θ). The initial cut is $\gamma_1 = (v_1^+, v_1^-, v_2^+, v_2^-)$. When it crosses $\mathcal{R}_V$, it becomes $\gamma_2 = (v_1^+, v_2^+, v_1^-, v_2^-)$, and finally $\gamma_3 = (v_2^+, v_2^-, v_1^+, v_1^-)$ after leaving $\mathcal{R}_V$. (b) Cross-sections of the constraints on the planes (a, θ) and (b, θ) via the use of tipping curves.

As explained in Section 3.2, an FRTS $\mathcal{R}$ induced by $\mathcal{P}$ in the parameter space (a, b, θ) can be fully described from its two cross-sections $\mathcal{R}_V$ and $\mathcal{R}_H$ on the planes (a, θ) and (b, θ) as defined in Equations (17)–(18), and illustrated in Fig. 5. Relying on the similarity of $\mathcal{R}_V$ and $\mathcal{R}_H$, hereafter we consider only $\mathcal{R}_V$ (the same argument stands for $\mathcal{R}_H$). Our problem is then specified as follows: given a constraint set $\mathcal{P}$ of half-planes of $\mathcal{R}_V$, report the set of half-planes constituting the boundary of $\mathcal{R}_V$. From Corollary 8, Properties 7 and 9, we recall that $\mathcal{R}_V$ contains two non-empty sequences of half-planes:

- a upper boundary sequence $U_V = (v^+_{p_i q_i q'_i}, \ldots)$ that contains only the upper half-planes; and
- a lower boundary sequence $L_V = (v^-_{p_i q_i q'_i+1}, \ldots)$ that contains only the lower half-planes.

The 2D sweeping algorithm[4], presented in Section 4.1.1, is used to find the U and L sequences of $\mathcal{R}_V$ in which the input set of tipping curves is obtained from the constraints in $\mathcal{P}$. Note that no FDRT graph is built at this stage; we only need to observe the sequence of the cut γ during its update in order to obtain all the elements of U_V and L_V. Indeed, while sweeping γ, its sequence changes at event points. We remark that γ intersects with $\mathcal{R}_V$ when its sequence of half-planes is separated into two subsequences, γ^+ and γ^-, such that $\gamma = \gamma^+\gamma^-$ where γ^+ contains only the upper half-planes and γ^- contains only the lower half-planes. Moreover, we see that the last element of γ^+ and the first element of γ^- correspond respectively to the upper and lower half-planes of U_V and L_V constituting the boundary of $\mathcal{R}_V$. The cut is moved out of $\mathcal{R}_V$ when there is no longer any such separation. Under the change of γ in $\mathcal{R}_V$, an upper or lower half-plane is progressively added in U_V and L_V at

[4] This algorithm can easily be modified to deal with the quotient space $\mathbb{R}^3/\sim$.

each event point, as illustrated in Fig. 10(a). Similarly, we can obtain the sets U_H and L_H constituting the boundary of $\mathcal{R}_H$ for the plane (b, θ).

By using two cuts γ_a and γ_b sweeping in the two planes (a, θ) and (b, θ), we can find the boundary of an FRTS $\mathcal{R}$, $\mathbf{B} = \mathbf{U} \cup \mathbf{L}$ where $\mathbf{U} = U_V \cup U_H$ and $\mathbf{L} = L_V \cup L_H$. Indeed, at each event point either on (a, θ) or (b, θ), the algorithm updates and checks the sequence of the corresponding cuts. We start getting the boundary segments of $\mathcal{R}$ from the first θ at which both sequences of γ_a and γ_b are separated in two parts. Similarly, we stop collecting the boundary segments of $\mathcal{R}$ at the first θ at which there is no longer any such separation in neither γ_a nor γ_b, as illustrated in Fig. 10(b). Moreover, from this procedure we can also obtain the lowest and greatest values of θ of $\mathcal{R}$, denoted respectively by θ_{min} and θ_{max}, which are needed in the next stage of the algorithm. From Properties 7 and 9, we know that these values θ_{min} and θ_{max} of $\mathcal{R}$ are unique.

4.3 Finding tipping surfaces passing through a feasible rigid transformation set

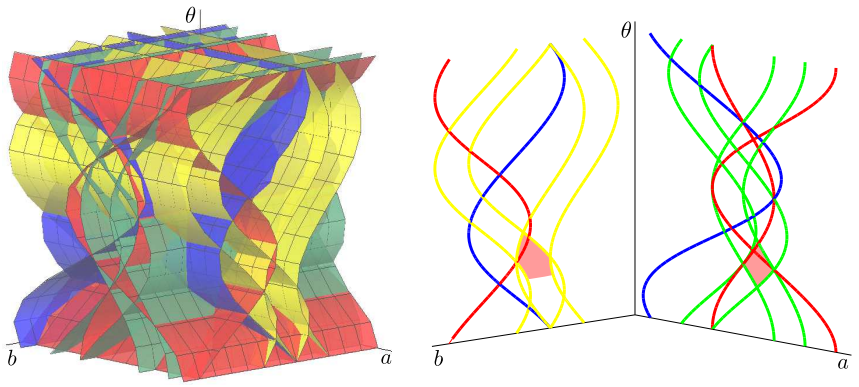
In order to compute an FDRT graph modeling the subdivision of an FRTS $\mathcal{R}$ into DRTs, we require not only the tipping surfaces constituting the boundary of $\mathcal{R}$ but also those passing through $\mathcal{R}$. In this section, we explain how to find these tipping surfaces.

So far, we know that an FRTS $\mathcal{R}$ contains all the rigid transformations satisfying given pixel-invariance constraints. $\mathcal{R}$ is partitioned into DRTs as well as the whole parameter space of rigid transformations, as explained in Section 2.3; the subdivision of $\mathcal{R}$ is induced by the tipping surfaces existing in $\mathcal{R}$ (see Fig. 11(a)). Therefore, we need to determine these surfaces among all the tipping surfaces in the parameter spaces (a, b, θ) of the rigid transformations associated to the image of size $N \times N$, *i.e.*, all vertical and horizontal tipping surfaces Φ_{pqk} and Ψ_{pql}, respectively, for $p, q \in [\![0, N-1]\!]$ and $k, l \in [\![0, N]\!]$.

- **Input**: The boundary set $\mathbf{B} = U_V \cup U_H \cup L_V \cup L_H$ of $\mathcal{R}$.
- **Output**: The set of tipping curves $\mathbf{P}$ passing through $\mathcal{R}$.

This problem is equivalent to finding the set $\mathbf{P}_V$ (resp. $\mathbf{P}_H$) of tipping curves ϕ_{pqk} (resp. ψ_{pql}) passing through $\mathcal{R}_V$ (resp. $\mathcal{R}_H$), the cross-sections of $\mathcal{R}$ on the plane (a, θ) (resp. (b, θ)) (see Fig. 11(b)). Then, we have $\mathbf{P} = \mathbf{P}_V \cup \mathbf{P}_H$.

Let us consider the cross-section $\mathcal{R}_V$ of $\mathcal{R}$. We call a segment of tipping curves that constitutes the boundary of $\mathcal{R}_V$ a *boundary* segment. Any tipping curve ϕ_{pqk} passes through $\mathcal{R}_V$ if it intersects one of the boundary segments of $\mathcal{R}_V$; without loss of generality, we denote such a boundary segment $\phi_{p'q'k'}$. This is easily detected by verifying the relationship between ϕ_{pqk} and $\phi_{p'q'k'}$, and the intersection is on a boundary segment of $\mathcal{R}_V$ as follows:



(a) (b)

Fig. 11 (a) Example of tipping surfaces passing through $\mathcal{R}$ (in red) and not passing through $\mathcal{R}$ (in blue) in the parameter space (a, b, θ). (b) Its cross-sections $\mathcal{R}_V$ and $\mathcal{R}_H$ of (a) on the planes (a, θ) and (b, θ) respectively.

(i) verify if ϕ_{pqk} and $\phi_{p'q'k'}$ intersect; this is true iff the following relations are satisfied [18, Property 2]:

$$\Delta_1 + \Delta_2 > 0 \tag{26}$$

$$|KP \pm \sqrt{\Delta_1}| \leq P^2 + Q^2 \tag{27}$$

$$|KQ \pm \sqrt{\Delta_2}| \leq P^2 + Q^2 \tag{28}$$

where $P = p - p'$, $Q = q - q'$, $K = k - k'$, $\Delta_1 = P^2(P^2 + Q^2 - K^2)$ and $\Delta_2 = Q^2(P^2 + Q^2 - K^2)$;

(ii) if they intersect, then calculate the following values at the intersection [18, Corollary 1]:

$$\sin\theta = \frac{KQ \pm \sqrt{\Delta_1}}{P^2 + Q^2} \tag{29}$$

$$\cos\theta = \frac{KP \pm \sqrt{\Delta_2}}{P^2 + Q^2} \tag{30}$$

and verify if $\theta_{min} \leq \theta \leq \theta_{max}$[5], where θ_{min} and θ_{max} are obtained from Section 4.2;

(iii) if (ii) is verified, then calculate

$$a_{upper} = \max_{V^+_{pqk} \in U} \{\phi_{pqk}(\theta)\} \tag{31}$$

$$a_{lower} = \min_{V^-_{pqk} \in L} \{\phi_{pqk}(\theta)\} \tag{32}$$

and verify if $a_{upper} \leq a \leq a_{lower}$, where the value a at the above intersection θ is calculated from Equation (5).

[5] Note that there are four possible combinations for $\sin\theta$ and $\cos\theta$. However, from [18, Corollary 2] and Property 9, only one of them is valid for the value of θ at the intersection.

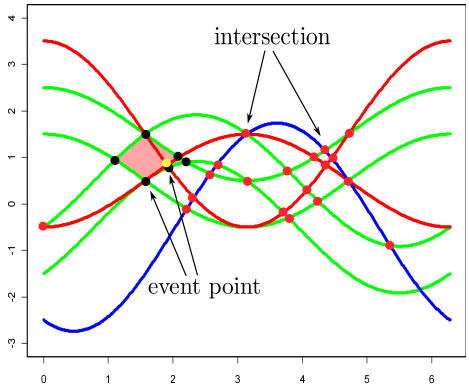


Fig. 12 Examples of boundary event points (in black), interior event points (in yellow) and intersections of tipping curves but not event points (in red) for computing a FRTS graph.

Note that the values $\cos\theta$ and $\sin\theta$ are used to represent θ. Since all $\cos\theta$, $\sin\theta$, $\cos\theta_{min}$, $\sin\theta_{min}$, $\cos\theta_{max}$, $\sin\theta_{max}$, a, a_{min} and a_{max} are quadratic irrationals[6], they can be compared exactly in average constant time [25].

We have ϕ_{pqk} passes through $\mathcal{R}_H$, *i.e.*, $\phi_{pqk} \in \mathbf{P}_H$, if all (i), (ii) and (iii) are satisfied. Then, the verification is performed for all tipping curves on the plane (a, θ) for computing $\mathbf{P}_H$. A similar approach is applied for the plane (b, θ) to compute $\mathbf{P}_H$.

4.4 Determination of event points in a feasible rigid transformation set

When carrying out the sweeping algorithm, it is mandatory to know how to detect event points in $\mathcal{R}$ or, equivalently, when to perform an elementary step. Due to the similarity of $\mathcal{R}_V$ and $\mathcal{R}_H$, in the following we consider only the cross-section $\mathcal{R}_V$ of $\mathcal{R}$. In the sweeping method for building a FDRT graph, *event points* are intersections of tipping curves at which the sequence of the cut γ in $\mathcal{R}_V$ is changed, as illustrated in Fig. 12. More precisely, event points in $\mathcal{R}_V$ are intersections of the tipping-curve segments which constitute the boundary of $\mathcal{R}_V$, *i.e.*, $\mathbf{B}_V = U_V \cup L_V$ (see Section 4.2), or pass through $\mathcal{R}_V$, *i.e.*, P_V (see Section 4.3). Note that event points generated by these tipping curves can be either on a boundary segment or in the interior of $\mathcal{R}_V$, and are called *boundary event points* or *interior event points*, respectively (see Fig. 12). We denote the set of boundary event points as $\mathbf{E}_b$, and this of interior event points as $\mathbf{E}_i$. This distinction of event points is necessary in the next step for computing a FTRS graph. The procedure for handling event points in $\mathcal{R}_H$ is explained in Section 4.5. We here focus on how to detect these event points.

- **Input**: A set of tipping curves $\mathcal{C} = \mathbf{B}_V \cup \mathbf{P}_V$.
- **Output**: The set $\mathbf{E} = \mathbf{E}_b \cup \mathbf{E}_i$ of event points of $\mathcal{R}_H$ generated by $\mathcal{C}$.

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

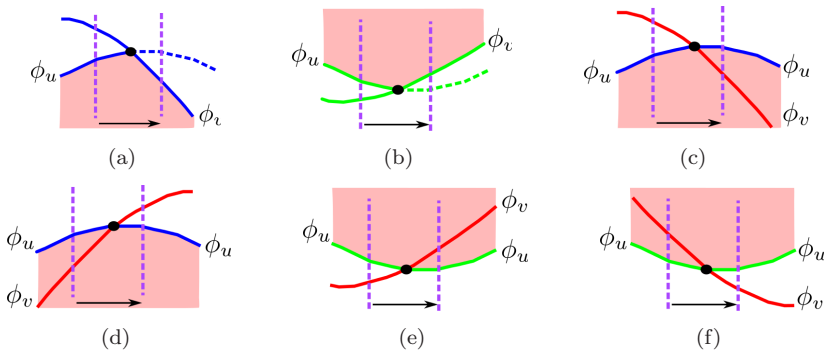


Fig. 13 Classification of simple boundary event points. An event point generated by two tipping curves ϕ_u and ϕ_v: (a,b) changes the boundary from ϕ_u to ϕ_v, (c,d) ϕ_v goes in and out by crossing the upper boundary ϕ_u, and (e,f) ϕ_v goes in and out by crossing the lower boundary ϕ_u.

Similarly to the previous method in Section 4.3, if an intersection coordinate (θ, a) satisfies $\theta_{min} \leq \theta \leq \theta_{max}$ and $a_{lower}(\theta) \leq a \leq a_{upper}(\theta)$, then it is an event point in $\mathcal{R}_H$. The algorithm described in Section 4.1.1 deals with interior event points in $\mathbf{E}_i$. In contrast, boundary event points in $\mathbf{E}_b$ must be treated separately as follows.

According to their nature, the boundary event points in $\mathbf{E}_b$ can be classified into the following six types. Let $\mathbf{q} \in \mathbf{E}_b$ be presented as the set of tipping curves intersecting at $\mathbf{q}$ [18], *i.e.,* in simple case $\mathbf{q} = \{\phi_u, \phi_v\}$. As illustrated in Fig. 13, $\mathbf{q}$ is a boundary event point in the following cases:

- the boundary segment, which is either upper (type (a)) or lower (type (b)), changes from ϕ_u to ϕ_v;
- the boundary segment does not change, such that the tipping curves ϕ_v
  - goes into (resp. out) $\mathcal{R}$ by crossing the upper boundary segment ϕ_u (type (c) (resp. type (d)));
  - goes into (resp. out) $\mathcal{R}$ by crossing the lower boundary segment ϕ_u (type (e) (resp. type (f))).

Following this classification, the type of a boundary event point can be easily detected during the sweeping of the cut γ as follows. Let $\mathbf{q} = \{\phi_u, \phi_v\} \in \mathbf{E}_b$. Let $\gamma(\phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_n)$ be the cut on the left of $\mathbf{q}$. Let U and L be respectively the upper and lower boundary sequences, and $\phi_1 \in U$ and $\phi_n \in L$ be respectively the current *upper* and *lower* boundaries. Then a type of the boundary event point $\mathbf{q}$ is detected by verifying:

- if $\phi_u, \phi_v \in U \cup L$, then $\mathbf{q}$ is in:
  - type (a), if ($\phi_u = \phi_1$ and $\phi_v \neq \phi_2$) or ($\phi_v = \phi_1$ and $\phi_u \neq \phi_2$);
  - type (b), if ($\phi_u = \phi_n$ and $\phi_v \neq \phi_{n-1}$) or ($\phi_v = \phi_n$ and $\phi_u \neq \phi_{n-1}$);
- otherwise, if either $\phi_u \in U \cup L$ or $\phi_v \in U \cup L$, then $\mathbf{q}$ is in:
  - type (c), if $\phi_u = \phi_1$ and $\phi_v \neq \phi_2$;
  - type (d), if $\phi_u = \phi_1$ and $\phi_v = \phi_2$;
  - type (e), if $\phi_u = \phi_n$ and $\phi_u \neq \phi_{n-1}$;
  - type (f), if $\phi_u = \phi_n$ and $\phi_u = \phi_{n-1}$.

4.5 Feasible discrete rigid transformation graph construction

In order to build the FDRT graph in an FRTS, we use the sweeping algorithm described in Section 4.1. This algorithm can be extended to deal with supplementary constraints. The resulting graph G actually extends the notion of DRT graph initially introduced in [18].

- **Input**: The boundary set $\mathbf{B} = \mathbf{U} \cup \mathbf{L}$ of the FRTS $\mathcal{R}$, the set of event points $\mathbf{E} = \mathbf{E}_i \cup \mathbf{E}_b$, θ_{min} and θ_{max} of $\mathcal{R}$.
- **Output**: The FDRT graph G in $\mathcal{R}$.

In this part, the cut γ sweeps from θ_{min} to θ_{max} instead of $[0, 2\pi[$, and contains only the tipping surfaces belonging to $\mathcal{R}$. As described in Section 4.1, an elementary step at each event point consists of the two following steps:

(i) update the graphs G_{γ_a} and G_{γ_b} according to the change of the cuts γ_a and γ_b respectively, and
(ii) build the partial graph δG from G_{γ_a} and G_{γ_b}.

As Step (ii) directly derives from Definition 10 and Section 4.1.2, we here explain how to perform Step (i). For the sake of concision, we restrict ourselves to the handling of event points for simple cases; the degenerate cases are obtained by modifying the procedure of this simple case, as detailed in [18]. Given the similarity between $\mathcal{R}_V$ and $\mathcal{R}_H$, with no loss of generality, we only show the cases for G_{γ_a} of $\mathcal{R}_V$ in the sequel.

We recall that at each elementary step for an event point, the sequence of the cut γ_a is changed. According to this change, the associated graph G_{γ_a} is modified. Note that the procedure for handling with interior event points is similar to that given in [18, Procedure 1]. However the procedure for boundary event points requires some modifications.

We here explain only how to update the cut at boundary event points. We first dealt with the types (a) and (b). Without loss of generality, let $\mathbf{q} = \{\phi_u, \phi_v\}$ be a boundary event point generated by two tipping curves ϕ_u, ϕ_v where $\phi_u \in U_H$ or L_H, and γ, γ' be the cuts before and after $\mathbf{q}$ respectively. Assuming $\gamma = (\phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_n)$, if $\mathbf{q}$ is on:

- the upper boundary of $\mathcal{R}_H$, *i.e.,* $\phi_u = \phi_1$ and $\phi_v \neq \phi_2$, then $\gamma' = (\phi_v, \phi_2, \ldots, \phi_{n-1}, \phi_n)$ (type (a));
- the lower boundary of $\mathcal{R}_H$, *i.e.,* $\phi_u = \phi_n$ and $\phi_v \neq \phi_{n-1}$, then $\gamma' = (\phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_v)$ (type (b)).

Similarly, the procedures for updating the cut for types (c) and (d) are given as follows. Let $\mathbf{q} = \{\phi_u, \phi_v\}$ be an event point on the upper boundary, such that $\phi_u = \phi_1$. We have two cases:

- when ϕ_v goes into $\mathcal{R}_H$, such that $\phi_v \neq \phi_2$, then $\gamma' = (\phi_1, \phi_v, \phi_2, \ldots, \phi_n)$;
- when ϕ_v goes out from $\mathcal{R}_H$, such that $\phi_v = \phi_2$, then $\gamma' = (\phi_1, \phi_3, \ldots, \phi_n)$.

The procedures for types (e) and (f) can be considered in the same way. Fig. 14 illustrates the elementary steps for these boundary event points.

These procedures are more formally detailed in Appendix A.

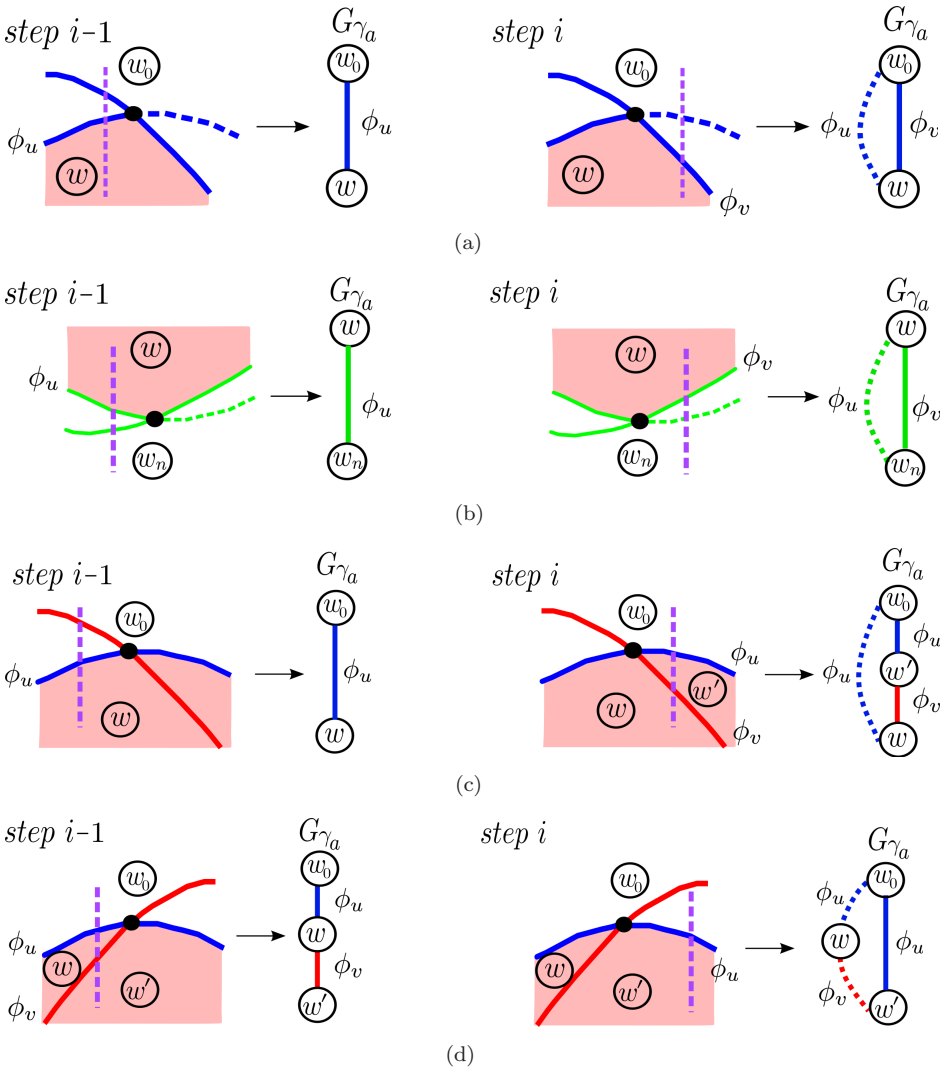


Fig. 14 Illustrations of elementary steps – update γ and generate its graph G_γ – for a tipping curve changing (a) an upper or (b) a lower boundary, and going (c) in or (d) out of an upper boundary.

5 Complexity analysis

5.1 Space complexity of feasible discrete rigid transformation graphs

5.1.1 Theoretical results

The space complexity of an FDRT graph corresponds to the numbers of its vertices and edges. These values directly depend on the number of event points involved in its construction, and the number of vertices generated at each

event point. Using a similar approach to [18, Proposition 3], we obtain that the number of edges is in the same order as the number of vertices. The following discussions, dealing only with vertices, then provide results for the space complexity of FDRT graphs including those for edges.

As mentioned in Section 4.1, the construction of a DRT graph G is obtained from its projections on the planes (a, θ) and (b, θ). In the absence of constraints, it was shown in [18] that there are $\mathcal{O}(N^3)$ tipping curves in each plane. Since any two tipping curves intersect in at most two points for $\theta \in [0, 2\pi[$, the number of event points is at most in $\mathcal{O}(N^6)$. Moreover, at each elementary step, *i.e.*, at each event point, the number of generated vertices is in $\mathcal{O}(N^3)$. The number of vertices in G is then in $\mathcal{O}(N^9)$, and thus this justifies the result already stated in Property 3. For one pixel-invariance constraint, we have the following property:

Property 13 *The FDRT graph G associated to a digital image of size $N \times N$ under one pixel-invariance constraint has a space complexity $\mathcal{O}(N^7)$.*

Proof The complexity analysis scheme remains similar to the non-constrained case. However, for one pixel-invariance constraint, some of the DRTs (*i.e.*, some of the vertices of the DRT graph) become infeasible, and the number of event points decreases from $\mathcal{O}(N^6)$ to $\mathcal{O}(N^5)$, due to periodicity properties (see [18, Property 4]). Moreover, we derive from Property 5 in [18] that the number of tipping curves in the projection on the plane either (a, θ) or (b, θ) of the FRTS associated to the given constraint, is $\mathcal{O}(N^2)$ instead of $\mathcal{O}(N^3)$. Thus at each elementary step, associated to each event point, the number of generated vertices is $\mathcal{O}(N^2)$. In total, there are $\mathcal{O}(N^5) \times \mathcal{O}(N^2) = \mathcal{O}(N^7)$ vertices added to the FDRT graph. ■

Geometrically, the associated FRTS correspond to a tube-like volume as illustrated in Fig. 5(c).

For more than one pixel-invariance constraints, the space complexity of the DRT graph does not only depend on the number of constraints, but also on the geometric configuration of the points involved in these constraints. This implies that the space complexity may not necessarily decrease, and so remains the same as with the one pixel-invariance constraint in the worst case.

Property 14 *The FDRT graph G associated to a digital image of size $N \times N$ under two pixel-invariance constraints has a space complexity of $\mathcal{O}(N^7)$ in the worst case.*

Proof Let us consider the two pixel-invariance constraints induced by the set of points $\{\boldsymbol{p}_1, \boldsymbol{p}_2\}$ and $\{\boldsymbol{p}'_1, \boldsymbol{p}'_2\}$, with $\boldsymbol{p}_1 = \boldsymbol{p}_2 + (0, 1)$ and $\boldsymbol{p}'_1 = \boldsymbol{p}'_2 + (0, 1)$. While the first constraint reduces the space complexity of G from $\mathcal{O}(N^9)$ to $\mathcal{O}(N^7)$ as stated in Property 13 (see also Fig. 5(c)), the second reduces the θ part of the FRTS from $[0, 2\pi[$ to $]0, \pi[$. With these constraints, the size of G is only divided by a constant factor of 2. As a consequence, the complexity remains $\mathcal{O}(N^7)$. ■

N	FDRT Graph	
	Vertices	Edges
1	1	0
2	147	700
3	2 256	11 616
4	14 651	78 140
5	80 289	430 752
6	265 899	1 445 986
7	842 137	5 314 904
8	2 076 029	13 190 632
9	5 103 633	32 291 768
10	10 244 909	65 204 024

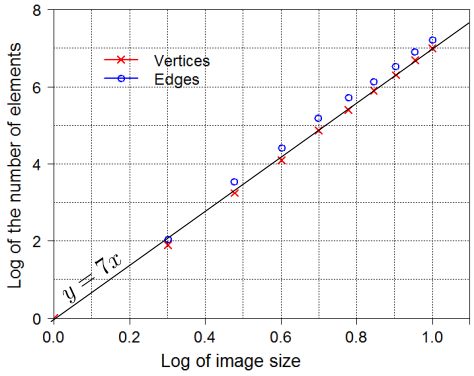


Fig. 15 Space complexities of FDRT graphs expressed as the numbers of vertices and edges in FDRT graphs under one pixel-invariance constraint, for images of size $N \times N$.

However, we show in Section 5.1.2, that the actual space complexities of FDRT graphs under multiple pixel-invariance constraints are generally lower than this worst case.

5.1.2 Experimental results

We now describe some experiments and results obtained with the proposed algorithm for FDRT graph construction under pixel-invariance constraints. This algorithm was implemented in C++. Experiments were carried out on a personal computer equipped with a 3.0GHz Intel® Core™ 2 Duo processor and 4GB of memory. The aim of these experiments is to validate the proposed algorithm with respect to the theoretical complexity results established above, but also to investigate practical complexities for tighter constraints.

The first experiments, illustrated in Fig. 15, deal with FDRT graphs for one pixel-invariance constraint. They confirm the theoretical results established in Property 13. One may notice that in previous works dealing with discrete rotations [2,27], a complexity of only $\mathcal{O}(N^3)$ was established. In those works, no translation was considered. In the current case, we allow translations due to the pixel-invariance formulation. As a result, for a given constraint, we obtain a set of feasible transformations forming a tube which contains arbitrary rotations and their associated translations whose regions forms a pixel, as illustrated in Fig. 5(c).

As discussed in Section 5.1.1, the space complexity of G under two pixel-invariance constraints is also $\mathcal{O}(N^7)$ (see Property 14) if we consider a pair of pixels separated by a distance of 1. This constitutes however an extreme case. In practice, the complexity of an FDRT graph is generally lower, since the distance between two constraint points is likely to exceed 1 (see Fig. 18(a)). It is reasonable to infer that the longer the distance between these points, the more constrained the feasible transformations, and therefore the lower the

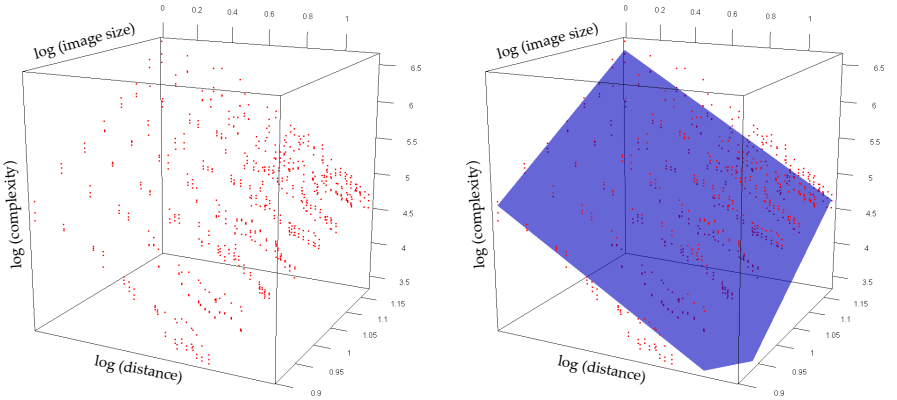


Fig. 16 Experimental space complexities of FDRT graphs in the two pixel-invariance constraints case. The complexities are measured over 676 experiments as a function of image size, varying from 8×8 to 15×15, and varying distances between two given constraints limited by the image sizes. Left: 3D log plot of distance, image size and space complexity. Right: least-squares best plane fit (colored in blue).

complexity of G. Following this intuition, we propose the following conjecture which links the complexity of G with the distance between points of given constraints.

Conjecture 15 *The FDRT graph G associated to a digital image of size $N \times N$ under two pixel-invariance constraints has a space complexity of $\Theta(N^{\alpha} d^{-\beta})$, with $\alpha, \beta > 0$, where d denotes the Euclidean distance between two pixel-invariance constraints.*

The α and β values are assumed to be constant, and we propose to estimate them experimentally (see Fig. 16). To this end, we synthesized images of sizes varying from 8×8 to 15×15. For each image size, we set several distances, and randomly chose several two pixel-invariance constraints for each distance. The estimation of α and β is interpreted as a plane fitting estimation in the 3D space induced by N, d and the space complexity c by taking log for both side of $c = sN^{\alpha}d^{-\beta}$, *i.e.*, $\log c = \log s + \alpha \log N - \beta \log d$, where s is a constant. By using the least-squares method, we obtained $\alpha = 5.5$ and $\beta = 1.6$ with a residual standard error for derived parameters of 0.1244 on 676 experiments performed and the adjusted multiple correlation coefficient of 0.9993, which corresponds to an accurate fit.

We now consider more than two constraints. We may have expected the FDRT graph G associated to these constraints to be reduced. Nevertheless, this is not always true. Indeed, the space complexity of G then depends on the geometric configuration of the pixel-invariance constraints, as illustrated by the following examples. Let us consider two pixel-invariance constraints (in red and blue in Fig. 17(a)). When a supplementary constraint is added (for instance the yellow one in Fig. 17(a)) we can see in Fig. 17(c) that the FRTS is strictly reduced, and so is the FDRT graph G. On the contrary, let us consider

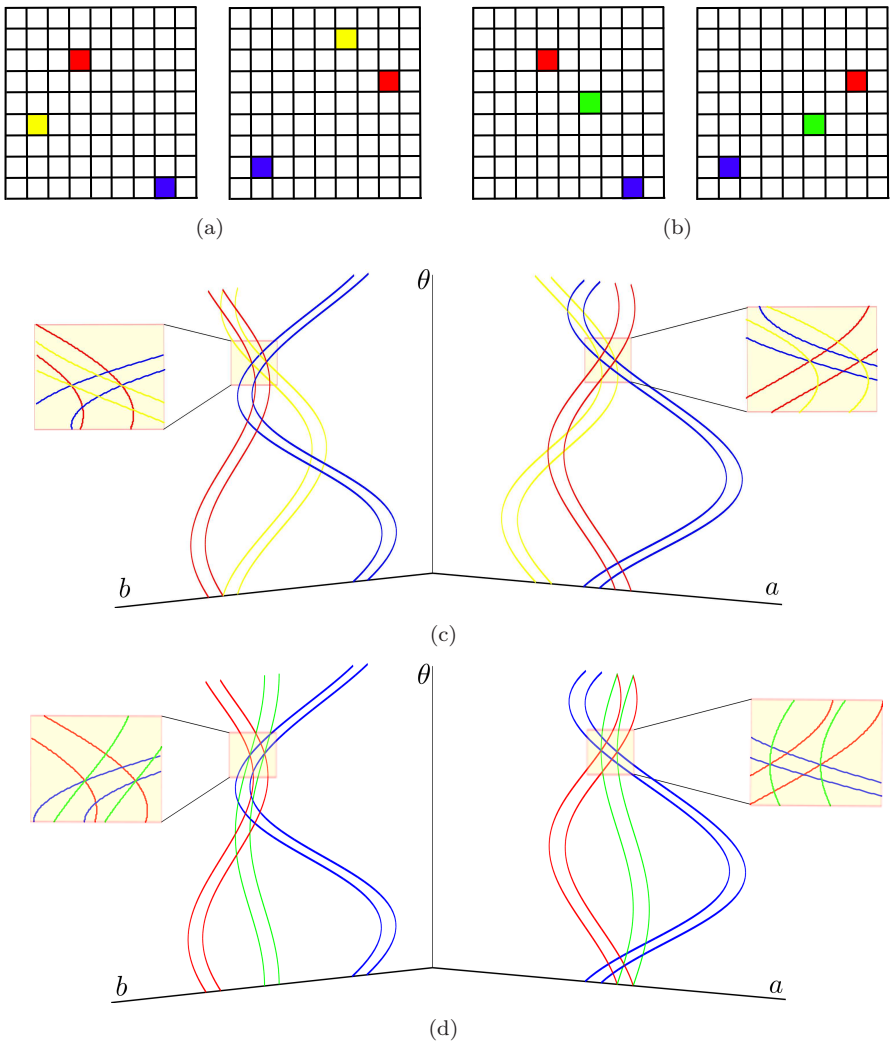


Fig. 17 For two given pixel-invariance constraints (in red and blue), there exists a supplementary pixel-invariance constraint (in yellow) that contributes to reducing the associated FRTS (a). There also exists a supplementary pixel-invariance constraint (in green) that does not (b). (c) and (d) illustrate the cross-sections of the FRTS, on the planes (a, θ) and (b, θ), induced by the constraints given in (a) and (b) respectively.

the supplementary constraint determined by green pixels in Fig. 17(b). We can observe that such a constraint does not reduce the FRTS, and the FDRT graph G then remains unchanged (see Fig. 17(d)).

However, in practice, the higher the number of constraints, the lower the complexity of the FDRT. This is illustrated in Figure 18(b–d) that corresponds to experiments for 3, 5 and 10 random pixel-invariance constraints, respectively.

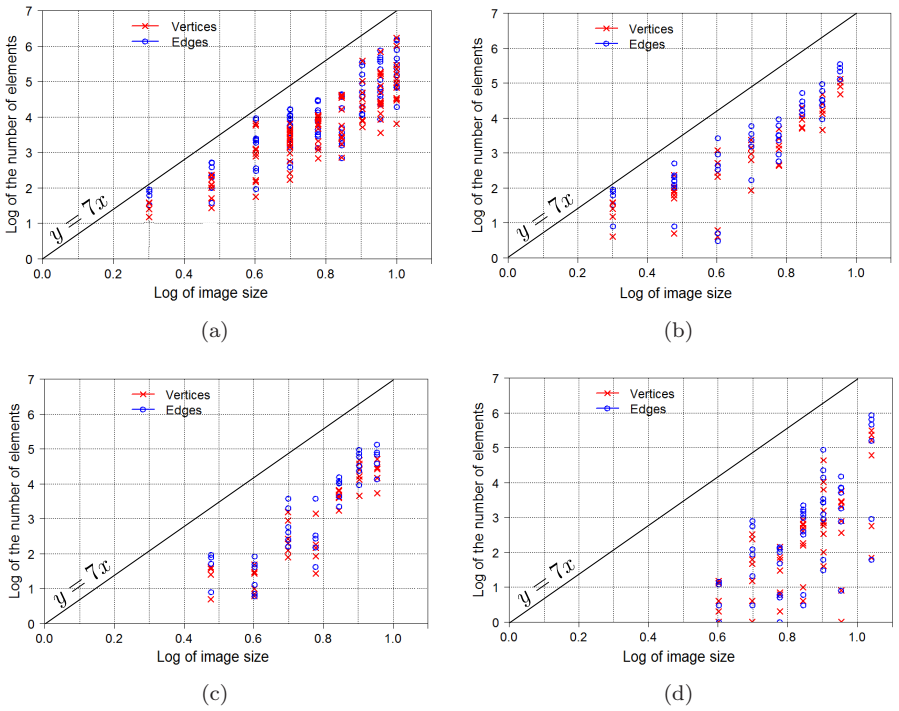


Fig. 18 Experimental results of space complexities of FDRT graphs for two (a), three (b), five (c) and ten (d) pixel-invariance constraints.

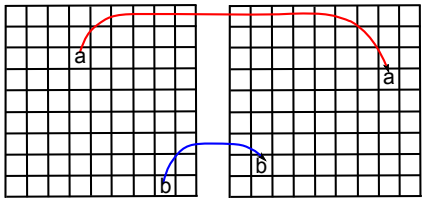
5.2 Time complexity of feasible discrete rigid transformation graph construction

From Section 4, we know that the construction of an FDRT graph is obtained from its projections on the planes (a, θ) and (b, θ), and can be performed in three successive steps:

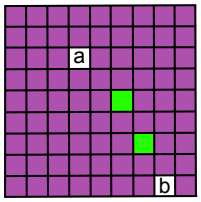
1. finding the boundary of the FRTS $\mathcal{R}$;
2. finding the event points in $\mathcal{R}$, more precisely, in the projections $\mathcal{R}_H$ and $\mathcal{R}_V$ of $\mathcal{R}$ on the planes (a, θ) and (b, θ) respectively; and
3. building the FDRT graph G associated to $\mathcal{R}$.

In each of the (a, θ) and (b, θ) planes, the event points are generated by $2m$ tipping curves, induced by the m given constraints (with $0 \leq m \leq N^2$). Moreover, it is proved in [18] that the number of event points for $2m$ tipping curves is $\mathcal{O}(m^2)$. In Step 1, the event points need to be sorted, then at each event a verification of the cut separation is done. This sorting and verification lead to a complexity of $\mathcal{O}(m^2 \log m)$ and $\mathcal{O}(m^3)$ respectively. Therefore, Step 1 requires a time complexity of $\mathcal{O}(m^3)$, which is equivalent to $\mathcal{O}(N^6)$.

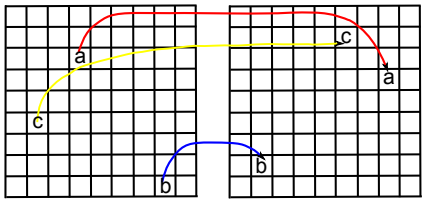
In Step 2, the process of detecting whether a tipping surface passes through $\mathcal{R}$ can be done in linear time with respect to the number of boundary segments



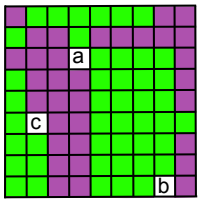
(a) Two pixel-invariance constraints.



(b) Static *vs.* non-static pixels in the left image of (a).



(c) Three pixel-invariance constraints.



(d) Static *vs.* non-static pixels in the left image of (c).

Fig. 19 (a,c) Pixel-invariance constraints (denoted by the arrows). (b,d) Classification of the remainder pixels in the image under constraints given in (a,c) respectively. The green pixels are static and the purple ones are not (see text).

of $\mathcal{R}$, *i.e.*, $\mathcal{O}(m)$. Since there are $\mathcal{O}(N^3)$ tipping surfaces, the time complexity of this sub-step is $\mathcal{O}(mN^3)$. Then, we search the event points in $\mathcal{R}$. Due to the periodicity of tipping curves [18, Property 4], we know that the total number of event points in $\mathcal{R}$ is $\mathcal{O}(N^5)$, and thus the mandatory sorting of these event points needs a time complexity of $\mathcal{O}(N^5 \log N)$.

The sweep of a cut, in Step 3, requires $\mathcal{O}(N^5)$ iterations (one for each event point), and at each iteration, $\mathcal{O}(N^2)$ vertices are generated. Therefore, Step 3 requires a time complexity of $\mathcal{O}(N^7)$, which is the most costly step in the algorithm.

Finally, the FDRT graph G for a given image of size $N \times N$ under m constraints is then constructed with a time cost of $\mathcal{O}(N^7)$.

6 Conclusion

This article continued the study initiated in [17,18] by investigating the effects of geometric constraints on rigid transformations, applied to digital images. By enforcing the correspondence between one or several pairs of pixels, we restricted allowable transformations to a parameter subspace, called a feasible rigid transformation set (FRTS), in which all such constraints are satisfied. A proposed algorithm allowed us to build a combinatorial structure (namely a graph) for modeling the subdivision of the FRTS on a subset of $\mathbb{Z}^2$ of size

$N \times N$. We theoretically analysed the complexity of this graph with one given pixel-invariance constraint to be $\mathcal{O}(N^7)$. For two constraints, the complexity could not be theoretically given. However, we experimentally evaluated its complexity as $\Theta(N^{5.5}d^{-1.6})$, where d denotes the Euclidean distance between two pixel-invariance constraints.

From the discussion of Section 5.1.2 regarding the complexity of the FDRT graph under more than two constraints, it appears that pixels can be classified into two categories: those which can be involved in supplementary constraints that reduce the FRTS, and those that cannot (see Fig. 19). The later ones are called *static* pixels. Based on this classification, it may be possible to choose only those which actually reduce the FRTS when iteratively defining constraints. This strategy may be investigated in further works.

Acknowledgements The research leading to these results has received partial funding from the French *Agence Nationale de la Recherche* (Grant Agreement ANR-10-BLAN-0205).

References

1. Amintoosi, M., Fathy, M., Mozayani, N.: A fast image registration approach based on SIFT key-points applied to super-resolution. Imaging Science Journal **60**(4), 185–201 (2011)
2. Amir, A., Kapah, O., Tsur, D.: Faster two-dimensional pattern matching with rotations. Theoretical Computer Science **368**(3), 196–204 (2006)
3. Andres, E.: The quasi-shear rotation. In: DGCI, Proceedings, *Lecture Notes in Computer Science*, vol. 1176, pp. 307–314. Springer (1996)
4. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag New York, Inc. (2006)
5. Chan, T.M.: On levels in arrangements of surfaces in three dimensions. In: SODA, Proceedings, pp. 232–240. ACM-SIAM (2005)
6. Coeurjolly, D., Blot, V., Jacob-Da Col, M.A.: Quasi-affine transformation in 3-D: Theory and algorithms. In: IWCIA, Proceedings, *Lecture Notes in Computer Science*, vol. 5852, pp. 68–81. Springer (2009)
7. Edelsbrunner, H., Guibas, L.J.: Topologically sweeping an arrangement. In: STOC, Proceedings, pp. 389–403. ACM (1986)
8. Gose, E., Johnsonbaugh, R., Jost, S.: Pattern Recognition and Image Analysis. Prentice-Hall, Inc. (1996)
9. Gribaa, N., Noblet, V., Khalifa, N., Faisan, S., Hamrouni, K.: Binary image registration based on geometric moments: Application to the registration of 3D segmented CT head images. International Journal of Image and Graphics **12**(2) (2012)
10. Harris, C.: Tracking with rigid models. In: A. Blake, A. Yuille (eds.) Active Vision, pp. 59–73. MIT Press (1993)
11. Hundt, C., Liśkiewicz, M.: On the complexity of affine image matching. In: STACS, Proceedings, *Lecture Notes in Computer Science*, vol. 4393, pp. 284–295. Springer (2007)
12. Hundt, C., Liśkiewicz, M.: Combinatorial bounds and algorithmic aspects of image matching under projective transformations. In: MFCS, Proceedings, *Lecture Notes in Computer Science*, vol. 5162, pp. 395–406. Springer (2008)
13. Hundt, C., Liśkiewicz, M., Ragnar, N.: A combinatorial geometrical approach to two-dimensional robust pattern matching with scaling and rotation. Theoretical Computer Science **410**(51), 5317–5333 (2009)
14. Jacob, M.A., Andres, E.: On discrete rotations. In: DGCI (International Conference on Discrete Geometry for Computer Imagery), Proceedings, pp. 161–174 (1995)
15. Maintz, J., Viergever, M.: A survey of medical image registration. Medical Image Analysis **2**(1), 1–36 (1998)

16. Matousek, J.: On directional convexity. Discrete & Computational Geometry **25**(3), 389–403 (2001)
17. Ngo, P., Kenmochi, Y., Passat, N., Talbot, H.: Combinatorial properties of 2D discrete rigid transformations under pixel-invariance constraints. In: IWCIA, Proceedings, *Lecture Notes in Computer Science*, vol. 7655, pp. 234–248. Springer (2012)
18. Ngo, P., Kenmochi, Y., Passat, N., Talbot, H.: Combinatorial structure of rigid transformations in 2D digital images. Computer Vision and Image Understanding **117**(4), 393–408 (2013)
19. Ngo, P., Kenmochi, Y., Passat, N., Talbot, H.: Sufficient conditions for topological invariance of 2D digital images under rigid transformations. In: DGCI, Proceedings, *Lecture Notes in Computer Science*, vol. 7749, pp. 155–168. Springer (2013)
20. Nouvel, B., Rémila, E.: Configurations induced by discrete rotations: Periodicity and quasi-periodicity properties. Discrete Applied Mathematics **147**(2–3), 325–343 (2005)
21. Nouvel, B., Rémila, E.: Incremental and transitive discrete rotations. In: IWCIA, Proceedings, *Lecture Notes in Computer Science*, vol. 4040, pp. 199–213. Springer (2006)
22. Pennec, X., Ayache, N., Thirion, J.P.: Landmark-based registration using features identified through differential geometry. In: I.N. Bankman (ed.) Handbook of Medical Imaging, chap. 31, pp. 499–513. Academic Press (2000)
23. Reveillès, J.P.: Géométrie discrète, calcul en nombres entiers et algorithmique. Thèse d'état, Université Strasbourg 1 (1991)
24. Richman, M.S.: Understanding discrete rotations. In: ICASSP, Proceedings, vol. 3, pp. 2057–2060. IEEE (1997)
25. Rosen, K.H.: Elementary Number Theory and its Applications, 3rd edn. Addison-Wesley (1992)
26. Sharir, M.: Recent developments in the theory of arrangements of surfaces. In: FSTTCS, Proceedings, *Lecture Notes in Computer Science*, vol. 1738, pp. 1–21. Springer (1999)
27. Thibault, Y.: Rotations in 2D and 3D discrete spaces. Ph.D. thesis, Université Paris-Est (2010)
28. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys **38**(4), 1–45 (2006)
29. Zitová, B., Flusser, J.: Image registration methods: A survey. Image and Vision Computing **21**(11), 977–1000 (2003)

A – Procedures for modifying a graph associated to a cut at boundary event points

Let $G_{\gamma_a} = (V_{\gamma_a}, E_{\gamma_a})$ (resp. $G_{\gamma_b} = (V_{\gamma_b}, E_{\gamma_b})$) be the graph associated to the cut γ_a (resp. γ_b) at Step i. At each event point of Step $i+1$, the algorithm updates the status of the cuts γ_a and γ_b, by which their associated graphs G_{γ_a} and G_{γ_b} are respectively modified (see Fig. 14). The partial graph δG is then generated from the modified graphs G_{γ_a} and G_{γ_b}, and integrated in the final DRT graph G. This is called an *elementary step* of the algorithm (see Sections 4.1 and 4.5). Due to the similarity between γ_a and γ_b, in the sequel we deal with γ_a and a similar result is obtained for γ_b.

The procedure for modifying the graph G_{γ_a} with respect to the change of γ_a at boundary event points is as follows. Let $\mathbf{q} = \{\phi_u, \phi_v\}$ be a boundary event point which changes the upper (resp. lower) boundary from ϕ_u to ϕ_v, *i.e.*, type (a) (resp. (b)). Let $\gamma_a = (\phi_u, \phi_2, \ldots, \phi_n)$ (resp. $\gamma_a = (\phi_1, \ldots, \phi_{n-1}, \phi_u)$) be the cut on the left of $\mathbf{q}$, then after $\mathbf{q}$ we have $\gamma_a = (\phi_v, \phi_2, \ldots, \phi_n)$ (resp. $\gamma_a = (\phi_1, \ldots, \phi_{n-1}, \phi_v)$). We can generate the modified graph of G_{γ_a} at $\mathbf{q}$ according to the following steps:

- finding the current edge e of the boundary tipping surface ϕ_u;
- deleting e and replacing by a new edge e' having the same vertices as e and ϕ_v as its label;

The implementation is given in Procedure 1, which requires the following functions:

- $\vartheta(e)$ returns the two adjacent vertices of the edge e in V_{γ_a}.

Procedure 1: Modification of the graph associated to a cut with respect to a boundary event point of either type (a) or (b).

Input: A graph $G_{\gamma_a} = (V_{\gamma_a}, E_{\gamma_a})$ associated to a cut $\gamma_a = (\phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_n)$ and a boundary event point $\mathbf{q} = \{\phi_u, \phi_v\}$.
Output: The modified graph of G_{γ_a} at $\mathbf{q}$.

if *($\phi_u = \phi_1$ and $\phi_v \neq \phi_2$) or ($\phi_v = \phi_1$ and $\phi_u \neq \phi_2$)* **then**
  $e_1 \leftarrow \varepsilon(\phi_1)$; $e_2 \leftarrow \varepsilon(\phi_2)$
  $\{w\} \leftarrow \vartheta(e_1) \cap \vartheta(e_2)$
  $\{\phi'\} \leftarrow \{\phi_u, \phi_v\} \setminus \{\phi_1\}$ // $\phi' = \{$either ϕ_u or $\phi_v\}$ is a new upper boundary
  $E^-_{\gamma_a} \leftarrow \{(w_0, w, \phi_1)\}$ // $\{w_0\} = \vartheta(e_1) \setminus \{w\}$ is a top vertex
  $E^+_{\gamma_a} \leftarrow \{(w_0, w, \phi')\}$
if *($\phi_u = \phi_n$ and $\phi_v \neq \phi_{n-1}$) or ($\phi_v = \phi_n$ and $\phi_u \neq \phi_{n-1}$)* **then**
  $e_1 \leftarrow \varepsilon(\phi_n)$; $e_2 \leftarrow \varepsilon(\phi_{n-1})$
  $\{w\} \leftarrow \vartheta(e_1) \cap \vartheta(e_2)$
  $\{\phi'\} \leftarrow \{\phi_u, \phi_v\} \setminus \{\phi_n\}$ // $\phi' = \{$either ϕ_u or $\phi_v\}$ is a new lower boundary
  $E^-_a \leftarrow \{(w, w_n, \phi_n)\}$ // $\{w_n\} = \vartheta(e_1) \setminus \{w\}$ is a bottom vertex
  $E^+_{\gamma_a} \leftarrow \{(w, w_n, \phi')\}$
$E_{\gamma_a} \leftarrow E_{\gamma_a} \setminus E^-_{\gamma_a} \cup E^+_{\gamma_a}$ // No new vertex is generated at $\mathbf{q}$

Procedure 2: Modification of the graph associated to a cut with respect to a boundary event point of either type (c) or (d).

Input: A graph $G_{\gamma_a} = (V_{\gamma_a}, E_{\gamma_a})$ associated to a cut $\gamma_a = (\phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_n)$ and a boundary event point $\mathbf{q} = \{\phi_u, \phi_v\}$.
Output: The modified graph of G_{γ_a} at $\mathbf{q}$.

if $\phi_u = \phi_1$ **then**
  if $\phi_v \neq \phi_2$ **then**
    $e_u \leftarrow \varepsilon(\phi_u)$
    $\{w\} \leftarrow \vartheta(e_u) \setminus \{w_0\}$ // w_0 is a top vertex
    $E^-_{\gamma_a} \leftarrow \{(w_0, w, \phi_u)\}$
    $V^+_{\gamma_a} \leftarrow \{w'\}$ // w' is a new vertex
    $E^+_{\gamma_a} \leftarrow \{(w_0, w', \phi_u), (w', w, \phi_v)\}$
  if $\phi_v = \phi_2$ **then**
    $e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$
    $\{w\} \leftarrow \vartheta(e_u) \cap \vartheta(e_v)$
    $\{w'\} \leftarrow \vartheta(e_v) \setminus \{w\}$ // w' is an adjacent vertex of w
    $V^-_{\gamma_a} \leftarrow \{w\}$ // w is a removed vertex
    $E^-_{\gamma_a} \leftarrow \{(w_0, w, \phi_u), (w, w', \phi_v)\}$
    $E^+_{\gamma_a} \leftarrow \{(w_0, w', \phi_u)\}$
$V_{\gamma_a} \leftarrow V_{\gamma_a} \setminus V^-_{\gamma_a} \cup V^+_{\gamma_a}$
$E_{\gamma_a} \leftarrow E_{\gamma_a} \setminus E^-_{\gamma_a} \cup E^+_{\gamma_a}$

– $\varepsilon(\phi)$ returns the edge corresponding to the tipping curve ϕ in δE_{γ_a}.

Similarly, we have in Procedure 2 the algorithm for modifying the graph G_{γ_a} at a boundary event point $\mathbf{q} = \{\phi_u, \phi_v\}$ which has ϕ_v goes in and out by crossing the upper boundary ϕ_u, *i.e.*, type (c) and (d).