Transformations rigides sur les images numériques 2D : analyse combinatoire et topologique

Hoai Diem Phuc Ngo

UNIVERSITÉ PARIS-EST

UNIVERSITÉ PARIS-EST
ÉCOLE DOCTORALE MSTIC

T H È S E

pour obtenir le grade de

Docteur en Informatique – Signal, Image, Automatique

de l'Université Paris-Est, Marne-la-Vallée

Présentée par

Hoai Diem Phuc NGO

RIGID TRANSFORMATIONS ON 2D DIGITAL IMAGES

COMBINATORIAL AND TOPOLOGICAL ANALYSIS

Transformations rigides sur les images numériques 2D

Analyse combinatoire et topologique

Directeur de thèse : Michel COUPRIE

Encadrants : Yukiko KENMOCHI, Nicolas PASSAT et Hugues TALBOT

Soutenue le 18 octobre 2013

Membres du jury :

Rapporteurs :	Éric ANDRÈS	Professeur, Université de Poitiers
	David COEURJOLLY	Directeur de Recherche, CNRS, Lyon
Examinateurs :	Jean-Pierre REVEILLÈS	Docteur d'État
	Michel COUPRIE	Professeur, ESIEE - Paris
	Yukiko KENMOCHI	Chargée de Recherche, CNRS, Paris
	Nicolas PASSAT	Professeur, Université de Reims Champagne-Ardenne
	Hugues TALBOT	Professeur Associé, ESIEE - Paris
Invitée :	Fanny BUYENS	Ingénieur - Chercheur, CEA LIST - DIGITEO Labs

Abstract

In this thesis, we study rigid transformations in the context of computer imagery. In particular, we develop a fully discrete framework for handling such transformations.

Rigid transformations, initially defined in the continuous domain, are involved in a wide range of digital image processing applications. In this context, the induced *digital rigid transformations* present different geometrical and topological properties with respect to their continuous analogues. In order to overcome the issues raised by these differences, we propose to formulate rigid transformations on digital images in a fully discrete framework.

In this framework, Euclidean rigid transformations producing the same digital rigid transformation are put in the same equivalence class. Moreover, the relationship between these classes can be modeled as a graph structure. We prove that this graph has a polynomial space complexity with respect to the size of the considered image, and presents useful structural properties. In particular, it allows us to generate incrementally *all* digital rigid transformations without numerical approximation.

This structure constitutes a theoretical tool to investigate the relationships between geometry and topology in the context of digital images. It is also interesting from the methodological point of view, as we illustrate by its use for assessing the topological behavior of images under rigid transformations.

Keywords: Discrete rigid transformation, digital image, combinatorial structure, discrete algorithm, topological invariance.

Résumé

Dans cette thèse, nous étudions les transformations rigides dans le contexte de l'imagerie numérique. En particulier, nous développons un cadre purement discret pour traiter ces transformations.

Les transformations rigides, initialement définies dans le domaine continu, sont impliquées dans de nombreuses applications de traitement d'images numériques. Dans ce contexte, les *transformations rigides digitales* induites présentent des propriétés géométriques et topologiques différentes par rapport à leurs analogues continues. Afin de s'affranchir des problèmes inhérents à ces différences, nous proposons de formuler ces transformations rigides dans un cadre purement discret.

Dans ce cadre, les transformations rigides sont regroupées en classes correspondant chacune à une transformation digitale donnée. De plus, les relations entre ces classes de transformations peuvent être modélisées par une structure de graphe. Nous prouvons que ce graphe présente une complexité spatiale polynômiale par rapport à la taille de l'image. Il présente également des propriétés structurelles intéressantes. En particulier, il permet de générer de manière progressive *toute* transformation rigide digitale, et ce sans approximation numérique.

Cette structure constitue un outil théorique pour l'étude des relations entre la géométrie et la topologie dans le contexte de l'imagerie numérique. Elle présente aussi un intérêt méthodologique, comme l'illustre son utilisation pour l'évaluation du comportement topologique des images sous des transformations rigides.

Mots clés: Transformation rigide discrète, image numérique, structure combinatoire, algorithmique discrète, invariance topologique.

Résumé long

Le terme *transformation rigide* est employé pour décrire les transformations euclidiennes qui résultent de la combinaison d'une translation et d'une rotation. Une transformation rigide est, en particulier, définie par une fonction bijective de $\mathbb{R}^n$ dans $\mathbb{R}^n$ qui conserve notamment la colinéarité, les distances et les angles. Cette thèse propose une étude dédiée aux *transformations rigides digitales*, c'est-à-dire celles qui peuvent être concrètement utilisées pour traiter informatiquement les images définies sur $\mathbb{Z}^n$. Même si ces transformations digitales sont définies de manière très similaire à leurs homologues continues, elles diffèrent néanmoins par plusieurs aspects. En géométrie euclidienne, souvent présentée comme une géométrie "de la règle et du compas", les objets considérés sont idéalisés. Par exemple, une droite est infiniment fine et se compose d'un ensemble infini de points. En informatique, de tels objets idéaux n'existent pas. En effet, les objets sont discrétisés/numérisés et notamment représentés par un ensemble fini d'entités élémentaires, les *pixels*. Sur un écran d'ordinateur, les objets géométriques observés sont donc issus de notre interprétation psychovisuelle de ces ensembles de pixels. Une image numérique correspond alors à la représentation d'objets réels au travers d'un processus de digitalisation. D'un point de vue théorique, les images numériques sont ainsi modélisées comme des fonctions qui associent à chaque point de $\mathbb{Z}^2$ une valeur dans un espace "de couleurs" donné.

Les transformations rigides, lorsqu'elles sont considérées sur l'espace discret $\mathbb{Z}^2$ des images numériques, sont généralement appliquées dans l'espace continu associé ($\mathbb{R}^2$) et nécessitent ensuite le recours à un procédé de digitalisation afin d'obtenir un résultat sur $\mathbb{Z}^2$. Ce sont, en particulier, ces transformations rigides suivies d'un processus de digitalisation qui sont appelées *transformations rigides digitales*. Dans ce contexte, ces transformations digitales perdent, dans la plupart des cas, leurs "bonnes" propriétés par rapport à leurs homologues continues, du fait de la digitalisation. Ainsi, elles sont discontinues, non bijectives, et ne conservent plus certaines propriétés géométriques fondamentales telles que les angles et les distances entre les points, ainsi que les propriétés topologiques des objets.

Ces transformations rigides digitales sont néanmoins importantes. Dans le domaine de l'analyse d'images numériques, elles sont impliquées dans de nombreuses applications, par exemple, en recalage d'images, en suivie de mouvement, *etc.* Dans la plupart de ces applications sur les image numériques, les transformations rigides utilisées s'appuient sur une phase de traitement continu, suivie d'une numérisation finale, comme indiqué ci-avant. Néanmoins, pour d'autres applications comme l'appariement d'images ou la reconnaissance de motifs, il peut s'avérer nécessaire de générer toutes les images transformées. Si ces images ont des tailles finies, le nombre d'images numériques obtenues par transformation

rigide est intuitivement fini. Néanmoins, lors du passage au modèle continu, il est impossible de générer toutes les images transformées en raison de l'infinité de transformations rigides possibles. De plus, ces transformations sont basées sur des réels, modélisés informatiquement sous forme de nombres à virgule flottante, ce qui induit des erreurs de calcul. Au regard de ces considérations, nous cherchons à développer une nouvelle approche pour la modélisation et la manipulation des transformations rigides, adaptée aux espaces discrets des images numériques, et n'utilisant notamment que des calculs en nombres entiers.

Au cours de ce travail, qui participe à améliorer notre compréhension des transformations rigides digitales et de leur espace de paramètres, plusieurs questions sont abordées :

(i) Ces transformations peuvent-elles être gérées dans un cadre totalement discret ?

(ii) Combien de transformations rigides digitales existe-t-il dans un sous-ensemble fini de $\mathbb{Z}^2$?

(iii) Comment générer toutes ces transformations ?

(iv) Quelle est la structure de leur espace de paramètres ?

(v) Quelles sont les relations entre les transformations rigides et la topologie sur $\mathbb{Z}^2$?

Récemment, plusieurs travaux ont été menés afin de gérer les spécificités des transformations sur $\mathbb{Z}^2$. Dans ce cadre, des réponses combinatoires et/ou algorithmiques ont été fournies pour des classes de transformations telles que les rotations, les changements d'échelle, les transformations linéaires, affines et projectives, *etc.* À notre connaissance, une approche totalement discrète dédiée aux transformations rigides n'a cependant pas encore été proposée. Nous visons ainsi à combler cette lacune en répondant aux questions ci-dessus.

Afin de mieux comprendre les transformations rigides sur $\mathbb{Z}^2$, nous proposons de les formuler dans un cadre totalement discret. Pour cela, il convient tout d'abord d'étudier les propriétés de leur espace de paramètres. Sur la base du concept de rotations discrètes par des angles charnières [Nouvel 2005, Nouvel 2006b, Thibault 2010] et en nous inspirant de la technique de discrétisation pour le problème d'appariement d'images proposé par Hundt *et al.* [Hundt 2008a, Hundt 2009], nous montrons, dans le chapitre 2, que les transformations rigides digitales peuvent être modélisées et manipulées directement dans un espace discret. En effet, on observe que dans l'espace de paramètres associé, ces transformations se regroupent en classes d'équivalence dont chacune correspond à une unique transformation digitale et correspond alors à une seule image transformée. Une telle classe d'équivalence est appelée une *transformation rigide discrète*. En conséquence, l'espace de paramètres est subdivisable en un nombre fini de cellules ayant chacune une interprétation discrète homogène.

Dans le chapitre 3, nous montrons que les relations entre ces classes peuvent être modélisées par une structure combinatoire, et plus précisément par un graphe. Ce graphe, appélé un *graphe des transformations rigides discrètes* et noté G, se compose d'un ensemble fini de sommets et d'arêtes, de telle sorte que chaque sommet correspond à une classe d'équivalence, et chaque arête à un unique pixel dont la valeur diffère entre les images transformées associées aux deux sommets incidents. D'une part, nous proposons un algorithme permettant de construire G par un calcul exact et en temps linéaire par rapport à sa taille. D'autre part, nous présentons une analyse combinatoire, et prouvons notamment que G présente une complexité spatiale en $\mathcal{O}(N^9)$, où $N \times N$ correspond à la taille de l'image considérée. Par ailleurs, nous montrons que cette complexité peut être réduite si l'on induit des contraintes spatiales sur cette structure.

L'utilisation de ce graphe G permet de décrire toutes les transformations rigides d'un sous-ensemble fini de $\mathbb{Z}^2$, et ce sans avoir recours à la moindre erreur de calcul. En particulier, G modélise les "relations de voisinage" entre les transformations rigides sur $\mathbb{Z}^2$. Ainsi, l'étude de ce graphe permet (i) une meilleure compréhension du comportement topologique des transformations rigides sur $\mathbb{Z}^2$, et (ii) la génération de toutes les images obtenues par les transformations rigides pour une image donnée en modifiant progressivement (au plus) une valeur de pixel entre deux images transformées successives. Grâce à cet outil, nous pouvons ainsi décrire de manière purement discrète des transformations rigides sur $\mathbb{Z}^2$, permettant d'étudier, mettre en œuvre et utiliser ces transformations au sein d'un processus unique. Au-delà de ces aspects combinatoires et algorithmiques, ce nouvel outil peut également être impliqué dans des tâches de traitement d'image, et en particulier celles basées sur l'analyse de sous-échantillons [Pennec 2000, Buades 2005, Ngo 2013b].

Dans le chapitre 4, nous étudions plus précisément les relations existant entre la géométrie et la topologie dans le cadre de l'imagerie numérique, où ces deux notions sont plus étroitement liées que dans le cadre continu, et présentent au demeurant une réelle importance pour le traitement et l'analyse des images numériques. Dans $\mathbb{R}^2$, les transformations rigides sont des opérations préservant la topologie. Cependant, cette propriété est généralement perdue quand nous considérons les transformations rigides dans l'espace discret de $\mathbb{Z}^2$. En effet, comme il a déjà été énoncé ci-avant, du fait de la digitalisation de $\mathbb{R}^2$ à $\mathbb{Z}^2$, les transformations rigides discrètes ne préservent pas les distances ni les angles entre les points. Ces changements géométriques conduisent à des modifications de topologie dans l'espace transformé.

Dans cette étude, nous nous sommes concentrés sur des conditions pour lesquelles les images numériques préservent leurs propriétés topologiques par toutes les transformations rigides. In particulier, nous considérons le type d'homotopie en tant qu'invariant topologique pour ces images. D'une part, nous nous appuyons sur le graphe G d'une image I modélisant toutes les transformations rigides possibles de I et permettant de déformer cette image pro-

gressivement, c'est-à-dire pixel par pixel. D'autre part, nous tirons profit de la notion de point simple, définie sur les images digitales et permettant de garantir la préservation du type d'homotopie entre deux images. En combinant ces deux concepts, il est possible de garantir qu'une image I est topologiquement invariante si toutes ses images transformées sont obtenues en modifiant successivement les valeurs de points simples, lors du parcours exhaustif du graphe G. En particulier, la caractérisation locale des points simples convient parfaitement à une exploration progressive de G pour évaluer cette invariance topologique. Cette approche globale présente une complexité directement liée à la taille du graphe G, c'est-à-dire de l'ordre de N^9. En tirant profit de la nature locale des tests de caractérisation des points simples, nous proposons finalement d'élaborer une stratégie de vérification locale basée sur la décomposition spatiale de l'image en motifs autorisant – ou non – l'invariance topologique. Cette nouvelle approche mène à des conditions suffisantes pour évaluer l'invariance topologique des images en temps linéaire. Elle conduit par ailleurs à la proposition de stratégies de prétraitement efficaces des images visant à garantir leur invariance topologique. Les résultats et méthodes obtenus sont valables pour différents types d'images (binaires, à niveaux de gris, labellisées), ouvrant ainsi la voie à leur utilisation dans diverses applications de traitement et d'analyse d'images.

Remerciements

Je souhaiterais remercier celles et ceux qui m'ont permis, directement ou indirectement, la réalisation de ce travail et de devenir ce que je suis aujourd'hui.

J'aimerais avant tout dire un très grand merci à Yukiko Kenmochi qui m'a accompagnée dès mes premiers pas dans le monde de la recherche. Pendant ces trois années de thèse, elle m'a encouragée, aidée avec la plus grande bienveillance, et suivie pas à pas. Ce fut un véritable compagnonnage, sans cesse stimulant grâce à nos échanges. Je la remercie pour ses judicieux conseils, son écoute ainsi que ses relectures attentives, sa disponibilité et sa grande patience face à mon caractère parfois têtu et toutes les autres choses qui seraient trop longues à énumérer ici. Mes remerciements vont également à Nicolas Passat et Hugues Talbot pour leur encadrement ainsi que pour toute l'aide qu'ils m'ont apportée pour mes recherches pendant ces trois années. À Nicolas pour son incroyable organisation et sa considérable rapidité ainsi que l'efficacité qui me permet de finir ma thèse à temps. Plus particulièrement, sur sa patience et gentillesse pour relire et corriger les détails de mes articles. À Hugues qui m'a fourni des conseils scientifiques pour des nombreux choix importants dans mes travaux, et qui a consacré son précieux temps à la relecture de mes articles. Merci à vous trois pour m'avoir accordé votre confiance et m'ont permis de développer mes idées et de mener mes travaux jusqu'au bout. J'ai eu beaucoup de chance de travailler avec vous qui avez partagé avec moi vos connaissances et vos enthousiasme.

Je voudrais exprimer ma reconnaissance à Éric Andrès et David Coeurjolly d'avoir bien voulu être les rapporteurs de ce travail, ainsi qu'à Jean-Pierre Reveillès et Fanny Buyens de m'avoir fait l'honneur de prendre part à ce jury.

Merci au Professeur Akihiro Sugimoto et son équipe de m'avoir accueillie dans le laboratoire NII à Tokyo durant deux mois et ainsi permis de découvrir le Japon, et la culture (notamment la gastronomie) japonaise que j'ai bien aimée et qui m'a beaucoup impressionnée.

Je tiens à remercier Mai Quyen Pham, plus qu'une amie mais une sœur adorable, sympathique et sincère avec qui j'ai passé de bons moments. Un grand merci à Mai pour l'organisation et la préparation du pot de thèse. (J'ai beaucoup apprécié les plats vietnamiens qu'elle a préparé et qui me manquent beaucoup pendant ce temps en France.)

Je remercie également tous les membres de l'équipe de recherche A3SI pour l'excellente ambiance et les très bonnes conditions de travail qui ont régné tout au long de ma thèse. Merci pour toutes les discussions constructives et intéressantes (à la cantine, autour du café,

ou dans les couloirs) sur les sujets de recherche scientifique ainsi que de culture générale. Merci donc à Michel COUPRIE, Gilles BERTRAND, Laurent NAJMAN, Venceslas BIRI, Vincent NOZICK, Jean SERRA, Jean COUSTY, Benjamin PERRET, Denis BUREAU, Jean-claude GEOGES et Thierry GERAUD. Merci à tous mes compagnons de bureau, Nobert et Hiroko, pour votre bonne humeur et votre gentillesse qui ont rendu ma thèse très agréable. Un merci particulier aux personnes avec qui j'ai partagé de bons moments informels, et scientifiques : à Ravi pour les discussions sur les sujets très philosophiques hors du travail, les partages de musiques et de films et aussi les repas expérimentaux, à Christophe, pour les repas "chez bibi" suivis par les jeux de société, à Olena pour ses cartes restaurants riches en saveurs (ce qui m'a permis de découvrir la cuisine française, indonésienne, coréenne, ...), et à Camille pour son aide, ses multiples encouragements et ses précieux conseils et surtout pour sa grande gentillesse, à John, Benjamin, Yongchao, Imen et Olivia pour leur sympathie et leur amitié, et au professeur Jean Paul BOYER pour sa grande gentillesse à mon égard.

Je tiens enfin à remercier du fond du cœur mes parents, mes sœurs, mon copain et mes amis de leurs accompagnement et de leur soutien pendant toutes ces années.

Et à beaucoup d'autres personnes pour qu'elles soient cités sur cette simple page ...

List of enclosed articles

The work exposed in this thesis was mainly presented in the following research articles and research reports which are enclosed in the manuscript with permission from the publisher:

- [Ngo 2013a] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. *Combinatorial structure of rigid transformations in 2D digital images.* Computer Vision and Image Understanding, vol. 117, no. 4, pages 393–408, 2013.
- [Ngo 2013b] P. Ngo, Y. Kenmochi , N. Passat and H. Talbot. *Topology-preserving conditions for 2D digital images under rigid transformations.* Journal of Mathematical Imaging and Vision, 2013. DOI: 10.1007/s10851-013-0474-z.
- [Ngo 2013c] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. *On 2D constrained discrete rigid transformations.* HAL 00838184, 2013. (Submitted in an international journal.)

The complete list of publications and communications related to this PhD thesis, is provided in page 45.

List of Figures

Contents

Part I

Discrete rigid transformation

Chapter 1

Introduction

"*If we move an object around normally, it will in some sense remain rigid, and will not distort.*"

–Bill Casselman–

Rigid transformation is a term generally used to describe Euclidean transformations[1] that preserve the same shape and size of any object. The most basic form of rigid transformation that we can observe in nature is a combination of rotations and translations. Formally, a rigid transformation is defined as a bijective function from $\mathbb{R}^n$ to $\mathbb{R}^n$ that preserves collinearity, distances and angles. In this thesis, we have chosen to study *digital* rigid transformations, namely those that can be used for computer images. Despite intrinsic similarities, these digital transformations are rather different from their Euclidean analogues. So, what is a digital rigid transformation? And why do we study these objects?

The origin of digital transformations lies in computers and images. In Euclidean geometry, objects are idealized. For instance, a straight line is *infinitely* thin and contains an *infinite* number of points. In digital computers, such ideal objects do not exist. Indeed, they are digitized, and what we see on a computer screen is a *finite* collection of small picture elements, namely *pixels*, that our eyes and brain put together to recognize objects. These results in *digital images* that are a representation of reality, approximated by a digitization process. Rigid transformations acting on such images – and then also associated to a digitization process – are called *digital rigid transformations*.

From a theoretical point of view, if a computer screen is defined as a regular orthogonal grid, then a set of pixels can be considered as a subset of $\mathbb{Z}^2$. In such a way, a digital image is simply defined as a function that associates a given value to each point of $\mathbb{Z}^2$, *i.e.*, to each pixel. In this discrete context, (digital) rigid transformations need to be also defined as functions from $\mathbb{Z}^2$ to $\mathbb{Z}^2$. However, studying and performing digital rigid transformations are not obvious. Indeed, these transformations lose most of their "good" properties compared to their continuous counterparts. In particular, they are discontinuous, not bijective, do not preserve distances and angles between points, and neither do they preserve topology as well. This thesis deals with these specific issues. To this end, the study aims to, first, improve our understanding of digital rigid transformations and their associated parameter space; and, second, consider several questions related to this task for such transformations in the discrete space of digital images, including the following:

[1] Euclidean transformations simply because they form the basis of Euclidean geometry.

Classes of transformations	Complexity
Rotations [Amir 2006, Nouvel 2006a]	$\mathcal{O}(N^3)$
Scalings [Amir 1992, Amir 2009]	$\mathcal{O}(N^3)$
Rotations and scalings [Hundt 2009]	$\mathcal{O}(N^6)$
Linear transformations [Hundt 2008a]	$\mathcal{O}(N^{12})$
Affine transformations [Hundt 2007, Hundt 2010]	$\mathcal{O}(N^{18})$
Projective transformations [Hundt 2008a]	$\mathcal{O}(N^{24})$

Table 1.1: Number of the generated images under different classes of transformations on a subspace of $\mathbb{Z}^2$ of size $N \times N$.

(i) Is it possible to perform digital rigid transformations directly in a discrete space?

(ii) How many digital rigid transformations can we define on a finite set of $\mathbb{Z}^2$?

(iii) How to generate all of these transformations?

(iv) What is the structure of their space?

(v) What are the relationships between rigid transformations and topology in $\mathbb{Z}^2$?

In particular, the "continuous-to-discrete" transition induces an "infinite-to-finite" transition, from which the difficulty of these questions arise. Recently, some works have been devoted to understand the particularity of transformations on $\mathbb{Z}^2$. In this context, some combinatorial and algorithmic answers were provided for several classes of transformations, as shown in Table 1.1. Despite these efforts, to the best of our knowledge a fully discrete approach dedicated to rigid transformation has not been proposed. Throughout this thesis, we seek to contribute to this research area and provide answers for the above questions.

Following the idea of describing discrete rotations by hinge angles [Nouvel 2005, Nouvel 2006b, Thibault 2010] and inspired by the discretization technique of the problem of 2D pattern matching developed by Hundt *et al.* [Hundt 2008a, Hundt 2009], our goal is to develop a discrete framework for the class of rigid transformations on digital images. The rest of the manuscript is divided into two parts: the first presents a frame/summary of our contributions and results around the work presented in the three included articles, appearing in the second part of the manuscript. In this first part, the study is organized as follows.

Chapter 2 presents basic notions related to rigid transformations in the context of computer imagery, and investigates the effects of digitization on these transformations. We observe that digital rigid transformations are discontinuous because of such digitization, and they subdivide their parameter space into clusters of transformations – called *discrete rigid transformations* – having a same digitized result. In particular, these clusters are computed and handled by only using exact computation on integers.

In Chapter 3, we show that this subdivision of the parameter space of digital rigid transformations can be represented as a graph, in which each vertex corresponds to a discrete rigid transformation. This structure models *all* the rigid transformations defined on a finite subset of $\mathbb{Z}^2$. In particular, the concept of neighboring relationship between discrete transformations is derived from this graph. Indeed, each edge between two vertices models a "single pixel" difference between the rigid transformations associated to these vertices. We prove that this graph has a polynomial complexity in the order of N^9, where $N \times N$ is the size of the considered subset of $\mathbb{Z}^2$. Such complexity presents the number of transformed images under rigid transformations. Moreover, an algorithm is provided to compute this graph, using *only* integers and in linear time with respect to its size.

From a more practical point of view, we show, in Chapter 4, how the investigated graph can be associated to topological notions in order to develop algorithmic processes that enable to study the topological behavior of digital images under rigid transformations. More precisely, it is shown that the elementary modifications modeled by this graph can be associated to the topological notion of simple point to determine classes of topologically equivalent transformed images. This final result highlights the strong links that exist between geometry and topology in the framework of digital images.

Finally, Chapter 5 concludes this manuscript by a summary of our contributions and some perspectives.

Chapter 2

Rigid transformations on digital images

Contents

In the continuous space $\mathbb{R}^2$, rigid transformations are well-defined and have bijective property. Such transformations, while applied on the discrete space $\mathbb{Z}^2$ of digital images, require in addition a subsequent digitization in order to transfer their results into this discrete space. The rigid transformations followed by a process of digitization are called *digital rigid transformations*. They present very different properties compared to their continuous counterparts, due to this digitization process. Indeed, digital rigid transformations lose in general their bijectivity. It follows that they do not preserve some geometric properties such as angles and distances between points.

In order to improve the understanding of rigid transformations in discrete spaces, we propose to formulate digital rigid transformations in a purely discrete way. To this end, it is necessary to investigate what happens in their associated parameter space. Recently, some combinatorial and algorithmic studies of digital transformations have been carried out for 2D pattern matching under rotations [Amir 2006, Thibault 2010], scalings [Amir 1992, Amir 2009], combined scalings and rotations [Hundt 2008b, Hundt 2009], affine transformations [Hundt 2007, Hundt 2010], projective and linear transformations [Hundt 2008a]. However, there has been very little progress concerning the class of rigid transformations.

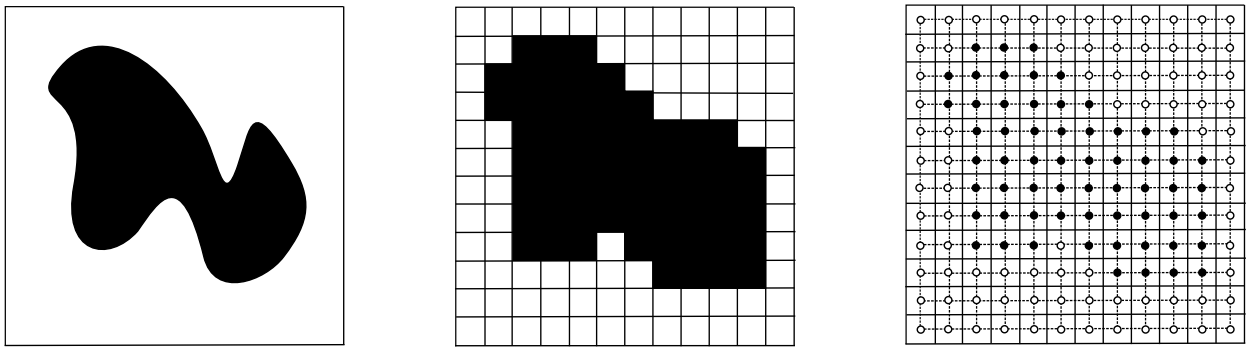


Figure 2.1: Illustration of an image and its digitization. A 2D image (left), the 2D digital image (center) obtained by digitizing the left image, and the grid associated to this digital image (right).

In this chapter, we first investigate the properties of digital rigid transformations resulting from the digitization process. Then, we propose a purely discrete formalization of rigid transformations on $\mathbb{Z}^2$. Our main result is to demonstrate that digital rigid transformations can be modeled and handled directly in their discrete space, by gathering them into equivalence classes leading to the subdivision of their parameter space.

The work exposed in this chapter was published in [Ngo 2013a].

2.1 Background notions

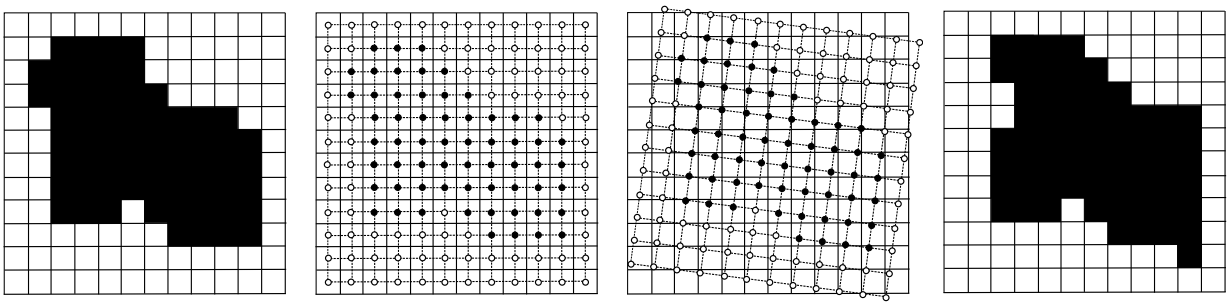
2.1.1 2D (digital) image

In the 2D continuous space, an image can be defined as a set of points in $\mathbb{R}^2$ where each point has a value in a set $\mathbb{V}$, and then formalized as a function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{V}$. Depending on the definition of $\mathbb{V}$, $\mathcal{I}$ can be considered as a binary, grey-level, label, color image, *etc.* So far, images are defined as infinite on $\mathbb{R}^2$. Practically, we consider images only inside some finite area. Under such assumption, images have a support defined on a finite subset of $\mathbb{R}^2$.

In computer vision, images are represented in a pictorial way as discrete sets, and called *digital images*. Such discrete sets are obtained by a sampling process, called *digitization* and denoted by D. In general, this process relies on a partitioning of $\mathbb{R}^2$ into unit squares, namely the *pixels*. Quite often, pixels have integer coordinates and their boundaries have semi-integer coordinates. By that way, a digital image is written as $I = D \circ \mathcal{I} : \mathbb{S} \to \mathbb{V}$, where $\mathbb{S} \subset \mathbb{Z}^2$ is a finite set of integer points. The digitization function D is a standard rounding operator[1].

Figure 2.1 (left) illustrates an image which is the representation of a real object on a finite support. After digitizing the given image, we obtain a digital image as a set of pixels appearing on

[1]The rounding operator considered for D can be the floor, ceiling, *etc.* All these operators are equal for all the points which do not have semi-integer coordinates, and particularly differ on these last points.



(a) (b) (c) (d)

Figure 2.2: Digital rigid transformation. (a) A digital image, (b) the associated grid of the image (a), (c) a rigid transformation applied on the discrete grid of the image (a), and (d) the digital transformed image.

a screen. Digital images are stored in computers as 2D array data structures, in which each pixel is represented as a point with integer coordinates.

In this work, we will focus on the effects of rigid transformations on the support of digital images. In particular, we will not consider their effects on the value set $\mathbb{V}$ of such images.

2.1.2 2D (digital) rigid transformation

A rigid transformation is defined as a combination of rotations and translations[2],

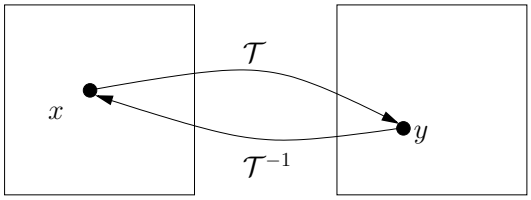
and expressed as a bijection $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$. Generally, $\mathcal{T}$ is modeled by a triplet of parameters (a, b, θ), where θ denotes an angle of rotation, and (a, b) denotes a vector of translation. Such transformation $\mathcal{T}$ will sometimes be denoted by $\mathcal{T}_{ab\theta}$.

In the discrete framework of digital images (where images are defined on $\mathbb{Z}^2$), it is however not possible to apply directly $\mathcal{T}$, since there is no guarantee that $\mathcal{T}(\boldsymbol{p}) \in \mathbb{Z}^2$ for any $\boldsymbol{p} \in \mathbb{Z}^2$. Based on the digitization D proposed in Section 2.1.1, a *digital rigid transformation* $T : \mathbb{Z}^2 \to \mathbb{Z}^2$ associated to $\mathcal{T}$ can be conveniently defined by setting $T = D \circ \mathcal{T}$, as illustrated in Figure 2.2.

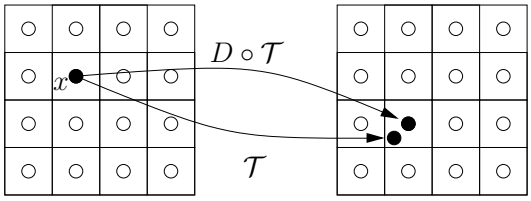
2.1.3 Transformation models

Two models are usually considered for digital image transformations. The first – namely, Lagrangian model – consists of observing the movement of discrete points of the initial space in the transformed

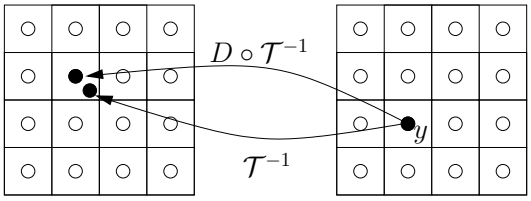
[2] Rigid transformations are composed of reflections, rotations and translations. While reflections are excluded, they are called *proper rigid transformations*. In this thesis work, we deal with the latter. The reason for this is twofold. First, any reflection can be trivially decomposed into a reflection across a given axis, combined with two rotations with the same center and opposite angles. Second, reflections are not as generally useful in matching and tracking applications, and may needlessly complicate the search space. By an abuse of language, the term rigid transformation used here refers to proper rigid transformation.



(a) Transformation models.



(b) Lagrangian model.



(c) Eulerian model.

Figure 2.3: Image transformation models: (a) transformation models in $\mathbb{R}^2$, (b) Lagrangian and (c) Eulerian transformation models in $\mathbb{Z}^2$. (a), (b) and (c) left: image support before transformation; right: image support after transformation. (b) and (c): the digital support $\mathbb{S} \subset \mathbb{Z}^2$ is depicted by white or black dots, and pixels are depicted by squares.

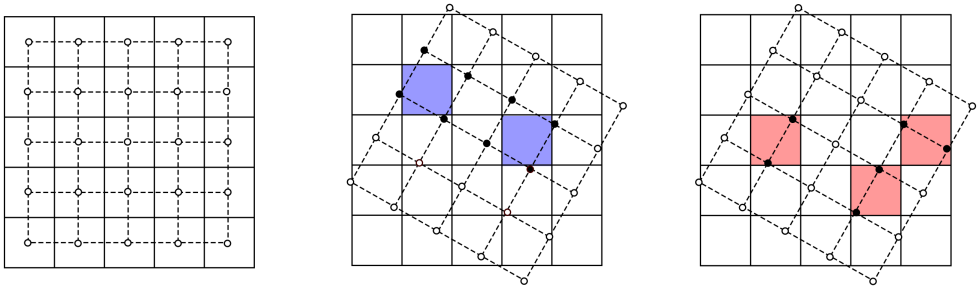
one (*i.e.*, finding $\boldsymbol{y} = \mathcal{T}(\boldsymbol{x})$ such that $\boldsymbol{x} \in \mathbb{Z}^2$). The second – namely, Eulerian model – consists of doing the inverse (*i.e.*, finding $\boldsymbol{x} = \mathcal{T}^{-1}(\boldsymbol{y})$ such that $\boldsymbol{y} \in \mathbb{Z}^2$)[3].

Figure 2.3 illustrates them in both continuous and discrete spaces. These models are equal in $\mathbb{R}^2$, since $\mathcal{T}$ is bijective. On the contrary, in $\mathbb{Z}^2$, they are actually distinct. Indeed, $D \circ \mathcal{T}^{-1}$ and $D \circ \mathcal{T}$ are generally not bijective. Without loss of generality, we are conveniently able to handle these two models by setting $T = D \circ \mathcal{T}_{|\mathbb{Z}^2}$ for the Lagrangian model, and $T^{-1} = D \circ \mathcal{T}^{-1}_{|\mathbb{Z}^2}$ for the Eulerian ones. All along the chapters of the thesis, the Eulerian model is used to generate the transformed images under rigid transformation.

2.2 Non-bijectivity of digital rigid transformations

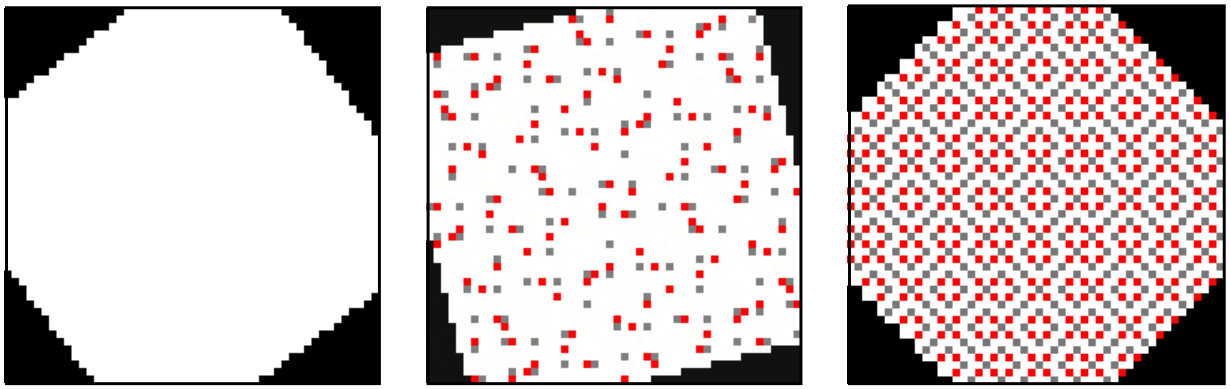
A continuous rigid transformation $\mathcal{T}$ establishes a bijection from $\mathbb{R}^2$ to itself. By opposition, due to the discrete nature of the digitization process D, a digital rigid transformation $T = D \circ \mathcal{T}$ is, most of the time, not bijective from $\mathbb{Z}^2$ to itself. Indeed, it is possible that T maps no discrete point of the initial space into a pixel of the transformed space, as illustrated in Figure 2.4(b). Such a pixel is called a *null* pixel, and the existence of null pixels implies that T is not surjective. The digital rigid transformation T may also map two discrete points of the initial space into a same pixel of the transformed space, as illustrated in Figure 2.4(c). Such a pixel is called a *double* pixel, and

[3]These two models are used in physics to observe the elements in time. In particular, the Lagrangian model is expressed in (element - time) coordinates and the Eulerian model is expressed in a (space - time) coordinates. The Lagrangian model is also known as forwards model while the Eulerian model is called backwards model.



(a) Original image. (b) Null pixels. (c) Double pixels.

Figure 2.4: Digital rigid transformation is not necessarily bijective. (a) Original image with the associated grid. (b,c) Rigid transformation applied on the grid defined in (a). (b) The blue pixels, containing no images of any discrete points after a digital rigid transformation T (namely, null pixels) show the non surjectivity of T. (c) The red pixels, containing two images of two discrete points after T (namely, double pixels) show the non injectivity of T.

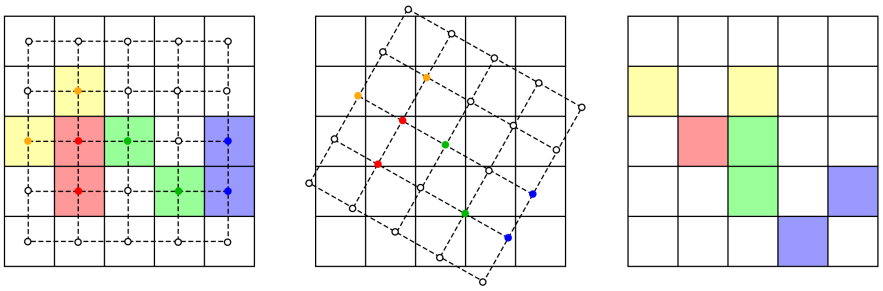


(a) $\theta = \arcsin(\frac{3}{5})$. (b) $\theta = \pi/12$. (c) $\theta = \pi/4$.

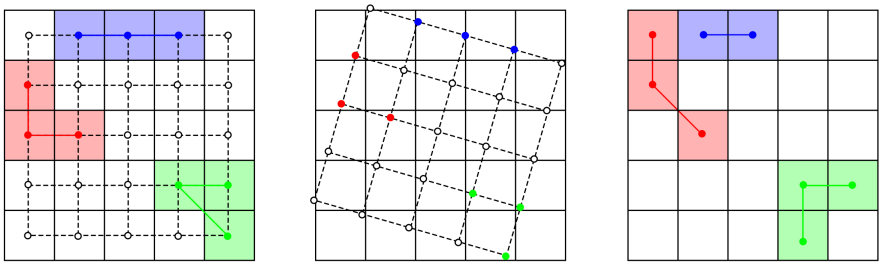
Figure 2.5: Some digital rotations by angles θ of a white square of size 100×100. Double pixels are depicted in red, and null pixels in grey.

the existence of double pixels implies that T is not injective. (We also use the term *single* pixel for the pixels having only one discrete point mapped into it under T.) This issue had already been identified in [Jacob 1995, Nouvel 2005, Nouvel 2006a, Thibault 2009, Thibault 2010] for rotations in discrete spaces. Some examples of null and double pixels appearing after some digital rotations are provided in Figure 2.5.

Any digital transformation T can not map a triplet of points into a same pixel. Indeed, let us consider three distinct points of $\mathbb{Z}^2$ forming a unit right isosceles triangle. In this configuration, the maximal Euclidean distance between these three points is $\sqrt{2}$, and the diagonal of a pixel is $\sqrt{2}$ as well. Therefore, the digitizing result of these three points under any digital rigid transformation can not be the same. In other words, a pixel can have only three *statuses* (either null, single or double) with respect to a digital rigid transformation.



(a) Distance alterations.



(b) Angle alterations.

Figure 2.6: Digital rigid transformations do not preserve distances (a) and angles (b). Left column ($\mathbb{S}$): original image with the associated grid. Middle ($\mathcal{T}(\mathbb{S})$): rigid transformation of the grid defined in the left column, where $\mathbb{S}$ is the support of the image. Right ($T(\mathbb{S}) = D \circ \mathcal{T}(\mathbb{S})$): result of the induced digital rigid transformation. (a) Two points in red (resp. blue, green and yellow) are considered for distance alterations changing from 1 (resp. 1, $\sqrt{2}$ and $\sqrt{2}$) to 0 (resp. $\sqrt{2}$, 1 and 2). (b) Three points in red (resp. blue and green) are considered for angle alterations changing from $\frac{\pi}{2}$ (resp. π and $\frac{\pi}{4}$) to $\frac{3\pi}{4}$ (resp. 0 and $\frac{\pi}{2}$).

2.3 Non-preservation of geometric properties

Beyond the issue of bijectivity, the digitization process from $\mathbb{R}^2$ to $\mathbb{Z}^2$ also causes some geometric alterations in the digital transformed space with respect to the initial one. Indeed, rigid transformations in $\mathbb{R}^2$ preserve (1) the distance (in particular, the Euclidean one) between any two points, and (2) the angle induced by any triplet of (distinct) points, but these properties are generally lost for digital rigid transformations.

Figure 2.6 illustrates both distance and angle alterations. For example, let us consider in the initial space two integer points (in red) of distance 1. After a digital rigid transformation, their images are now in the same pixel. Consequently, the distance of the two points in the transformed space is 0. Similarly, the angle between three points (in red) changes from $\frac{\pi}{2}$ to $\frac{3\pi}{4}$.

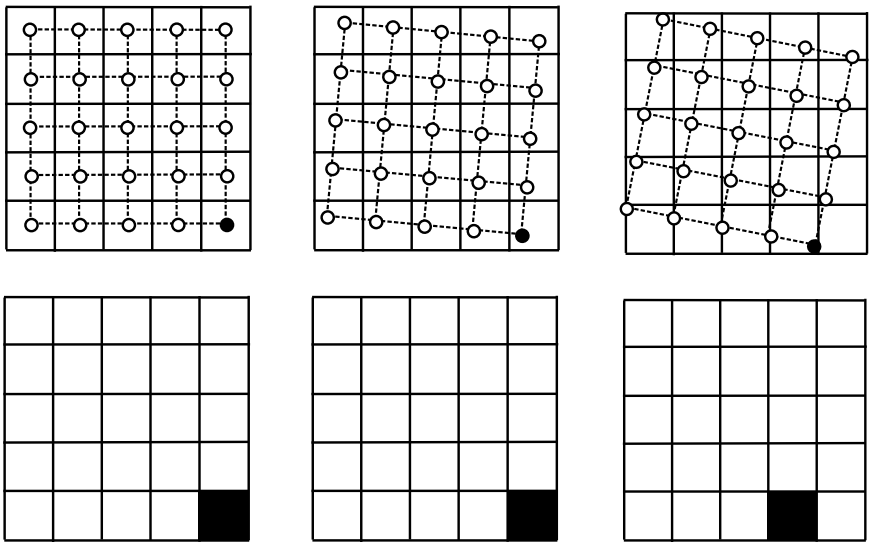


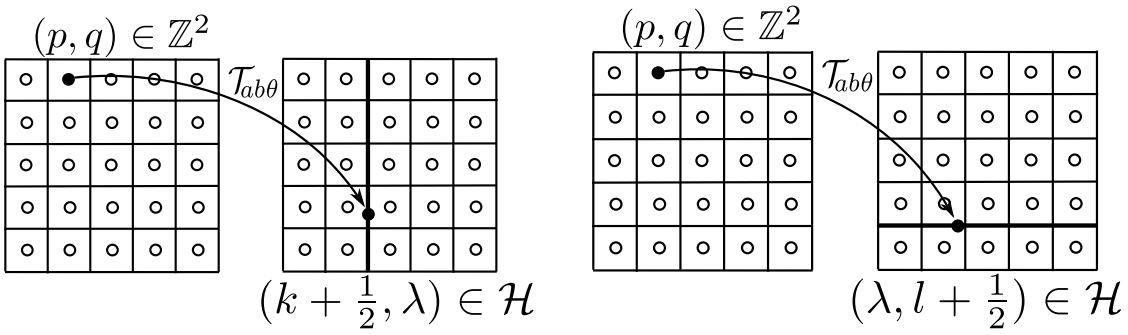
Figure 2.7: Illustration for the discontinuity of digital rigid transformations. Left: Original image with its associated grid (first row) and the associated digital image (second row). Middle: a digital rigid transformation is applied on the grid of the image (first row), the transformed image (second row) is identical to the original one. Right: another digital rigid transformation is applied on the grid of the image (first row) such that it moves the pixel center over the boundary of the pixel; the transformed image (second row) differs from the original one. The discrete points (center of pixel) are depicted by dots, their associated Voronoi cells boundary are depicted by lines.

2.4 Partition of the parameter space

2.4.1 Discontinuity of digital rigid transformations

A rigid transformation $\mathcal{T}$ in $\mathbb{R}^2$ is a continuous function. On the contrary, the digital analogue of $\mathcal{T}$, *i.e.*, the digital rigid transformation $T = D \circ \mathcal{T}$, is discontinuous. Indeed, the digitization function D decomposes the space of images into pixels, whose boundaries are points having semi-integer coordinates. The later points, namely the "inter-pixel" elements[4], are of great importance in digital topology [Kovalevsky 1989, Kong 1989]. They are also of first importance for understanding the parameter space of digital rigid transformations. Indeed, the discontinuities of transformations on $\mathbb{Z}^2$ induced by D occur *only* on these semi-integer points. Examples are shown in Figure 2.7. We observe that the transition of a pixel $\boldsymbol{p}$ into one of its four adjacent pixels occurs when $\boldsymbol{p}$ crosses a pixel boundary. In other words, a digital rigid transformation generates a new transformed image if it moves at least one pixel center over the discrete half-grid $\mathcal{H} = \big(\mathbb{R} \times (\mathbb{Z}+1/2)\big) \cup \big((\mathbb{Z}+1/2) \times \mathbb{R}\big)$; otherwise there is no new digital transformed image.

[4] A 2D digital image can be represented in a cellular model, such that we have: 0-cells, 1-cells and 2-cells for elements of dimension 0 (points), 1 (lines) and 2 (pixels) respectively. Then inter-pixel elements can be 0-cells or 1-cells.



(a) Vertical critical transformation. (b) Horizontal critical transformation.

Figure 2.8: Examples of critical transformations $\mathcal{T}_{ab\theta}$, each of which moves at least one pixel center onto its boundary which can be either "vertical" (a) or "horizontal" (b). The pixel centers are depicted by dots, while their boundary are depicted by lines.

We focus in particular on the triplets (a, b, θ) and their associated transformations $\mathcal{T}_{ab\theta}$, each of which maps at least one discrete point onto the discrete half-grid $\mathcal{H}$. Such transformations, called *critical transformations* and exemplified in Figure 2.8, characterize the discontinuities in the space of digital rigid transformations.

2.4.2 Tipping surfaces and tipping curves

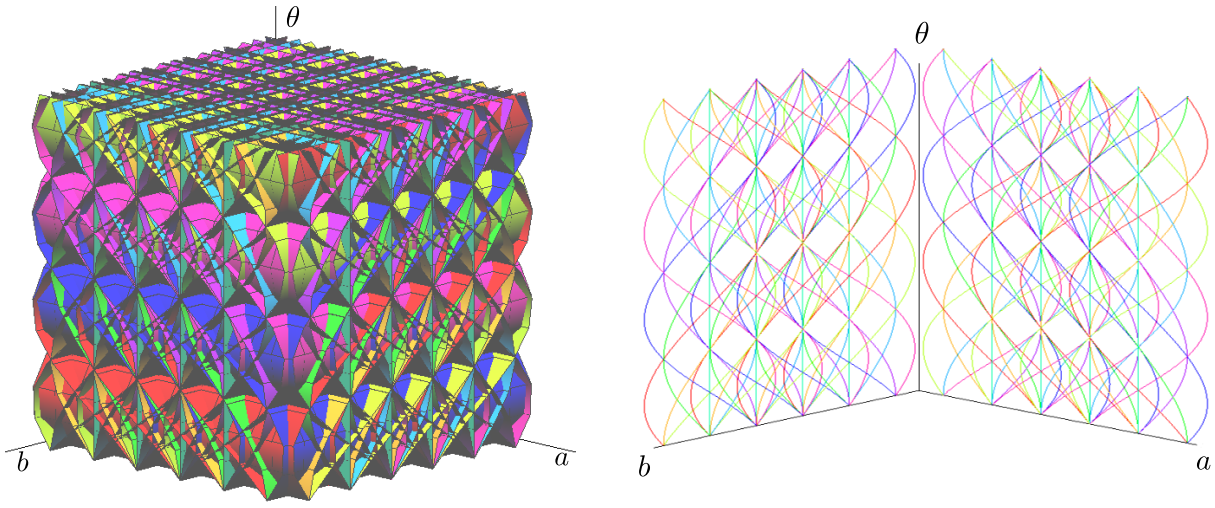
Analytically, each critical transformation can be expressed as the mapping of $\boldsymbol{p} = (p, q) \in \mathbb{Z}^2$ onto a half-grid point which can be either a horizontal $(k + \frac{1}{2}, \lambda) \in \mathcal{H}$ or a vertical one $(\lambda, l + \frac{1}{2}) \in \mathcal{H}$ (with $k, l \in \mathbb{Z}$ and $\lambda \in \mathbb{R}$) (see Figure 2.8). The sets of parameters (a, b, θ) modeling critical transformations $T_{ab\theta}$ form, in the parameter space, 2D surfaces, called *tipping surfaces*[5]. Concretely, a vertical (resp. horizontal) tipping surface that maps $\boldsymbol{p} = (p, q)$ onto $(k + \frac{1}{2}, \lambda)$ (resp. $(\lambda, l + \frac{1}{2})$) is modeled by the integer triplet (p, q, k) (resp. (p, q, l)), and then denoted by Φ_{pqk} (resp. Ψ_{pql}).

These two families of tipping surfaces can be projected on the planes (a, θ) and (b, θ). We then obtain two families of trigonometric curves, denoted by ϕ_{pqk} and ψ_{pql}, and called *tipping curves*. Examples of tipping surfaces and tipping curves are given in Figure 2.9. One may notice that the tipping surfaces Φ_{pqk} and Ψ_{pql} can be straightforwardly recovered by extrusion of the tipping curves ϕ_{pqk} and ψ_{pql}.

2.4.3 Digitization of the parameter space of rigid transformations

It has been observed that some distinct rigid transformations may have an identical digitization. This leads us to consider *equivalence classes* of digital transformations. More precisely, we can

[5]The term *tipping* refers to the fact that such surfaces characterize a transition of a grid point from a pixel to one of its adjacent pixels during a digital rigid transformation.



(a) Tipping surfaces.

(b) Tipping curves.

Figure 2.9: (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).

define the equivalence relation $\sim$ between transformations as $\left(\mathcal{T}_{ab\theta} \sim \mathcal{T}_{a'b'\theta'}\right) \iff \left(T_{ab\theta} = T_{a'b'\theta'}\right)$. It should be noticed that this relation is only defined for *non-critical* rigid transformations. From this equivalence relation, we can define the notion of *discrete rigid transformation*, that will be studied in the sequel.

Definition 1 (Discrete rigid transformation) *For a given digital image $I : \mathbb{S} \to \mathbb{V}$, each equivalence class of rigid transformations under $\sim$ (induced by $\mathbb{S}$) is called a* discrete rigid transformation *(DRT, for short).*

One may observe that the term *digital* refers to the digitization process of images and transformations for such images, while the term *discrete* refers to the non-continuous structure of these transformations.

The parameter space (a, b, θ) of rigid transformations is then partitioned into disjoint sets of DRTs, in which the function $(a, b, \theta) \mapsto T_{ab\theta} = D \circ \mathcal{T}_{ab\theta}$ is piecewise constant. Roughly speaking, any transformation in a same DRT generates an identical digital transformed image. In particular, the surfaces bounding these DRTs are tipping surfaces which correspond to discontinuities of the digital rigid transformations. The subdivision of the parameter space $\mathbb{R}^3$ of (a, b, θ) into DRTs is exemplified in Figure 2.10.

In this isomorphic framework, the number of DRTs is thus the number of *all* possible digital transformed images of a given image. Since these digital images are defined on a finite support, it is straightforwardly followed that the set of all DRTs is also finite.

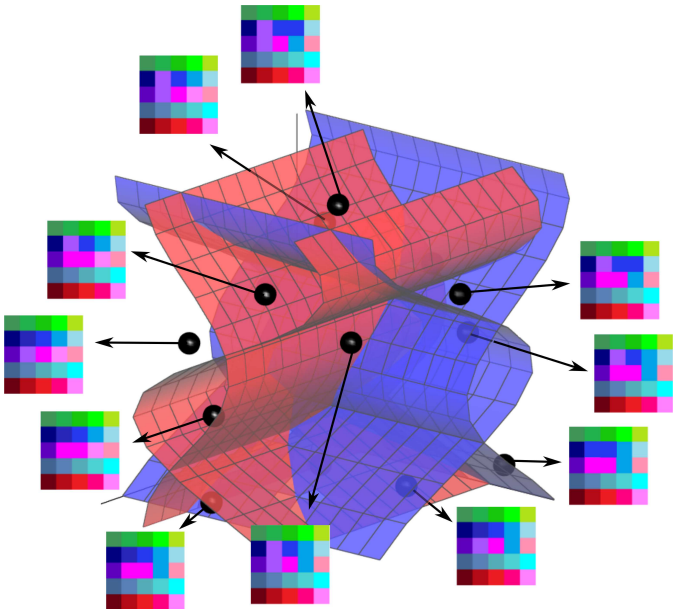


Figure 2.10: The parameter space of digital rigid transformations is divided into 3D open cells, and the 2D (tipping) surfaces bounding these open cells correspond to discontinuities of the digital rigid transformations. Each 3D open cell corresponds to a DRT and generates *one* transformed image. Each tipping surface, either Ψ_{pqk} or Φ_{pql}, indicates that the pixel $\boldsymbol{p} = (p, q) \in \mathbb{Z}^2$ will cross the half-grid line, either $x = k + \frac{1}{2}$ or $y = k + \frac{1}{2}$. Thus, two adjacent 3D open cells have their associated transformed images differing only at $\boldsymbol{p}$.

2.5 Summary

In this chapter, we have studied rigid transformations on digital images and shown that digital rigid transformations have very different properties with respect to their continuous analogues, namely, discontinuity, non-bijectivity and non-preservation of geometric properties. By investigating the parameter space of rigid transformations on $\mathbb{Z}^2$, it has been observed that this space could be subdivided into regions, each of which corresponds to a unique digital rigid transformation. This framework has led to the proposal of a purely discrete formalization of digital rigid transformations.

In the next chapter, we define a combinatorial structure used to represent the subdivision of the parameter space of digital rigid transformations. It will be shown in particular that this structure has a polynomial complexity, and can be explicitely constructed in optimal time.

Chapter 3

Combinatorial analysis of discrete rigid transformations

Contents

In this chapter, we introduce a dual structure (namely, a graph) in order to represent the subdivision of the parameter space of rigid transformations. This graph consists of finite sets of vertices and edges, such that each vertex represents a discrete rigid transformation, and each edge corresponds to a "one-pixel" difference between the two associated vertices. This structure describes all the possible rigid transformations of any subset of $\mathbb{Z}^2$ of size $N \times N$.

On one hand, we propose an algorithm with exact computation for building this graph in optimal time with respect to its size. On the other hand, we carry out a combinatorial analysis proving that this graph has a polynomial space complexity with respect to N. In particular, it is shown that this complexity can be decreased when considering spatial constraints.

With this novel tool, we are able to characterize neighboring relations between rigid transformations on $\mathbb{Z}^2$, and to explicitly determine and generate all these transformations. This provides a contribution to the field of combinatorial analysis of image transformations [Amir 2006, Hundt 2008a, Hundt 2009, Thibault 2010].

The work exposed in this chapter was published in [Ngo 2012, Ngo 2013a, Ngo 2013c].

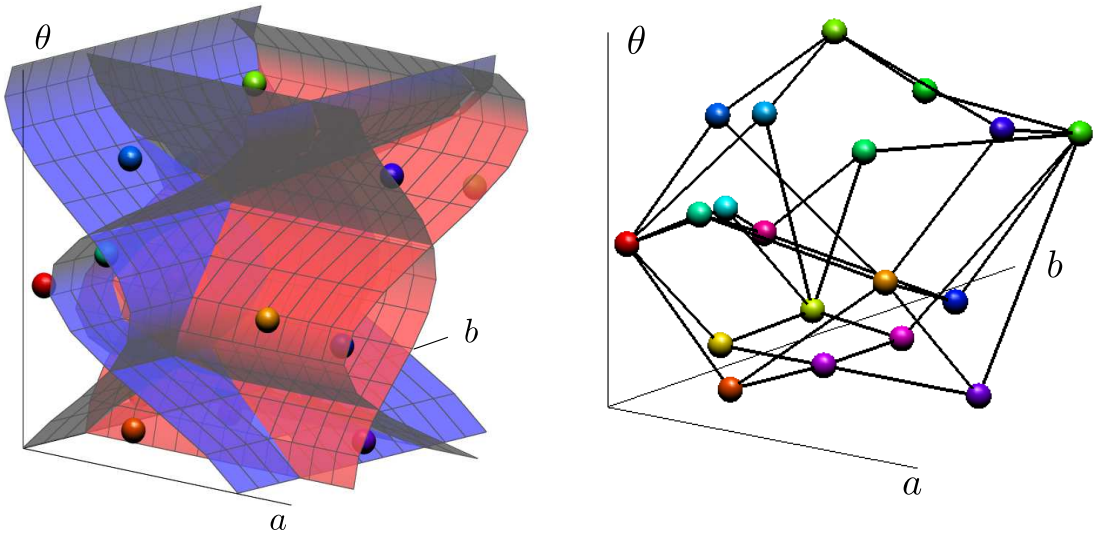


Figure 3.1: Parameter space of rigid transformations subdivided by four tipping surfaces (left), and the associated DRT graph (right).

3.1 Graph representation of discrete rigid transformations

As observed in the previous chapter, the parameter space of rigid transformations is subdivided into equivalence classes of transformations (namely, discrete rigid transformations or DRTs), which are modeled by 3D open cells. These cells are separated by tipping surfaces, modeled by 2D closed cells, which correspond to critical rigid transformations. This subdivision of the parameter space can be modeled, in a dual way, as a combinatorial structure. More precisely, we map any 3D cell onto a 0D point, and any 2D tipping-surface segment (linked to a critical transformation), shared by two adjacent 3D open cells, onto a 1D edge. The resulting structure is illustrated in Figure 3.1 and defined as follows.

Definition 2 (Discrete rigid transformation graph, DRT graph) *Given a set of tipping surfaces, Φ_{pqk} and Ψ_{pql}, the graph $G = (V, E)$ associated to DRTs induced by these tipping surfaces is defined such that:*

- *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 Ψ_{pqk} or Φ_{pql}) represents the tipping surface f between two adjacent vertices $v, w \in V$.*

This graph G is called a discrete rigid transformation graph *(*DRT graph*, for short).*

As mentioned in Section 2.4.3, any rigid transformation associated to a same DRT corresponds to a same digital transformation. Therefore, each DRT can be considered as a digital transformed

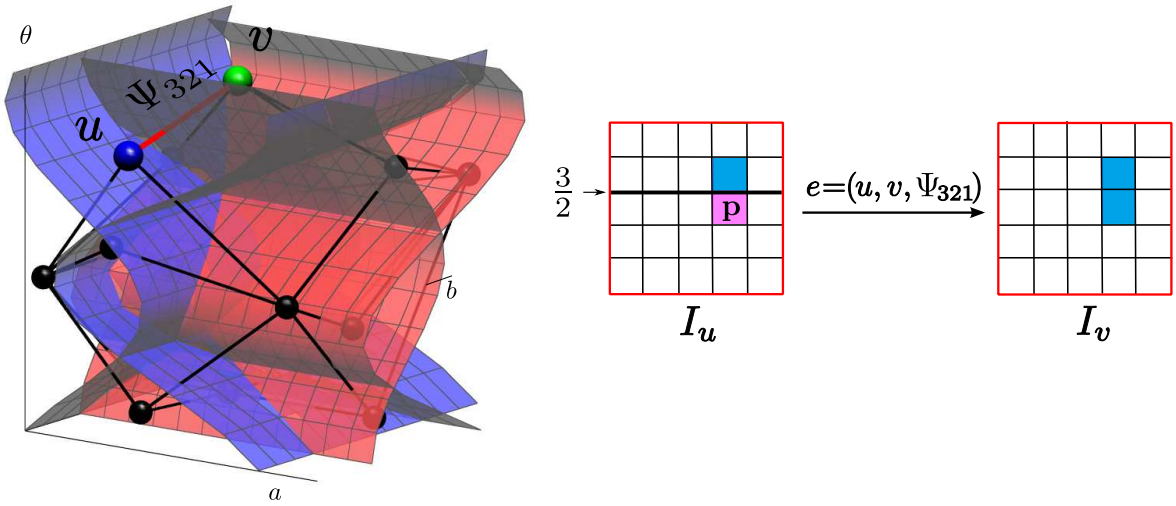


Figure 3.2: Left: subdivision of the parameter space of rigid transformations by tipping surfaces, and the associated DRT graph. Let us focus on the edge $e = (v, w, f)$ (in red) which connects two vertices u and v (in blue and green, respectively), and has the tipping surface Ψ_{321} as label. Right: two digital images I_u and I_v associated to the vertices u and v, respectively. The tipping surface Ψ_{321} encoded on e indicates that the pixel (in rose) $\boldsymbol{p} = (3, 2)$ will get its value from the one (in blue) after the half-grid line $y = 1 + \frac{1}{2}$, according to the Eulerian model. Thus, two vertices u and v separated by Ψ_{321} have their associated transformed images I_u and I_v differing *only* at $\boldsymbol{p}$.

image generated by any rigid transformation of the associated DRT. In the sequel, we denote by I_v the image associated to the DRT v. It should be noticed that a DRT graph does not contain any geometric parameter (a, b, θ) of the rigid transformations, but only structural information, encoded on the edges, which model the relationships between the adjacent transformed images. Concretely, let us consider an edge $e = (v, w, f) \in E$ between two distinct vertices $v, w \in V$, where the function f is either equal to Ψ_{pqk} or Φ_{pql}. Let $\boldsymbol{p} = (p, q) \in \mathbb{Z}^2$ and $\boldsymbol{q}$ be the point having as coordinates either (k, λ) or (λ, l) with respect to f. The function f indicates that the point $T^{-1}(\boldsymbol{p}) = \boldsymbol{q}$ will cross the half-grid line at coordinates $(x, y) \in \mathbb{R}^2$, with either $x = k + \frac{1}{2}$ or $y = l + \frac{1}{2}$ for $k, l \in [\![0, N]\!]$. 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 modification* at the edge e, we obtain a new transformed image I_w by simply changing the pixel value at $\boldsymbol{p}$ as $I_w(\boldsymbol{p}) = I(\boldsymbol{q} + \boldsymbol{\delta})$ where $\boldsymbol{\delta} = (\pm 1, 0)$ or $(0, \pm 1)$ with respect to f. In other words, there is exactly *one* pixel $\boldsymbol{q}$ differing between I_v and I_w. This property is exemplified in Figure 3.2. Typically, such elementary modifications allow the incremental manipulation of images, via pixel-by-pixel modifications.

Remark 3 *Let $I : \mathbb{S} \to \mathbb{V}$ be an image, and $G = (V, E)$ be its DRT graph. For each elementary modification at the edge $e \in V$ connecting two vertices v and w, two cases can occur:*

(i) $I_v(\boldsymbol{p}) = I_w(\boldsymbol{p})$, i.e., *the transformed images of I by the DRTs v and w are equal ($I_v = I_w$);*
(ii) $I_v(\boldsymbol{p}) \neq I_w(\boldsymbol{p})$, i.e., $I_v \neq I_w$.

The value of $\boldsymbol{p}$ may then be flipped from one value to another, and this may constitute the only

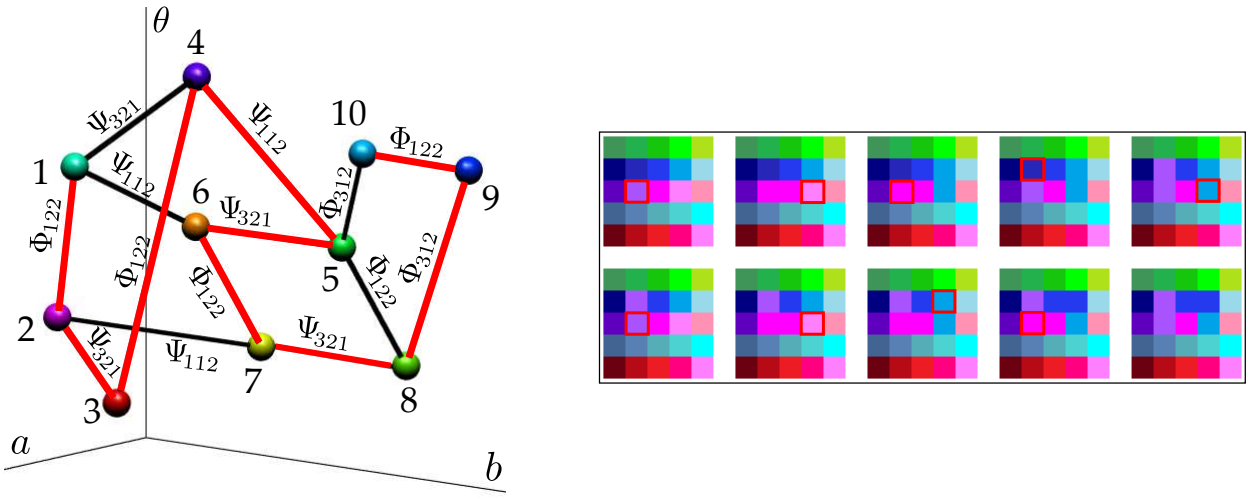


Figure 3.3: Left: a DRT graph in which each vertex/DRT represents a digital transformed image, and each edge indicates *only* one pixel whose value is modified between the two associated images. More precisely, an edge $e = (v, w, f)$ between two vertices v and w indicates that one pixel value is different between the associated transformed images I_v and I_w (see text). Right: the transformed images associated to the vertices of the DRT graph, and generated by using the Eulerian model. The images from upper-left to bottom-right correspond to the path (in red) of the vertices ordered from 1 to 10 in the graph. Along this path, two consecutive transformed images have one pixel of difference, which is depicted by the red frame.

modification between the transformed images of I by the DRTs v and w.

In other words, the case (i) appears when $I_w(\boldsymbol{p})$ has its value being identical to $I_v(\boldsymbol{p})$; then the images I_v and I_w are equal. Otherwise, we have the case (ii) where I_v and I_w are different at $\boldsymbol{p}$.

A DRT graph models a kind of "neighboring" relationship between DRTs (and between transformed images). Typically, two neighboring digital transformed images differ in (at most) one pixel among the N^2 ones. This particularity, exemplified in Figure 3.3, will open the way to the use of this combinatorial structure in image processing applications (see Chapter 4).

3.2 Algorithm for discrete rigid transformation graph construction

In this section, we present a method for computing the DRT graph for a given digital image I of size $N \times N$. This problem can be formulated as the problem of *arrangement of surfaces* [Sharir 1999, Chan 2005]. The algorithm for computing the arrangement of surfaces presents a complexity of $\Omega(n^4)$ for general surfaces, where n is the number of surfaces. In our – more specific – case, we propose an algorithm with a better complexity of $\mathcal{O}(n^3)$ based on the idea of the sweeping method described in [Edelsbrunner 1991a] using incident graph.

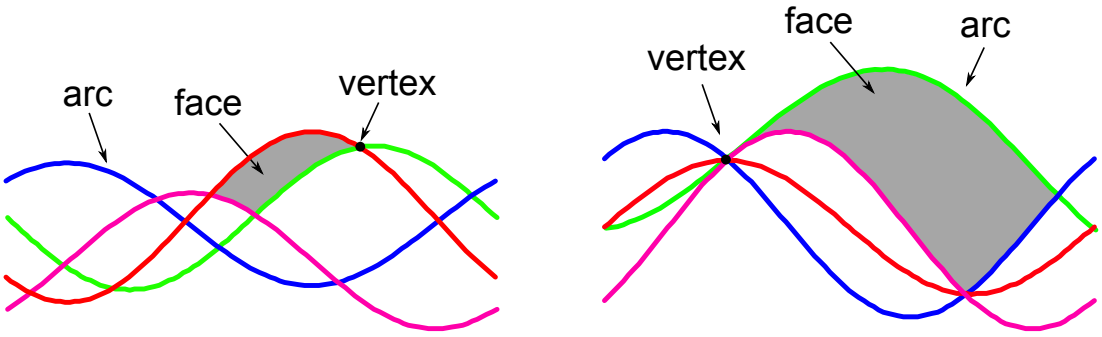


Figure 3.4: Examples of 2D arrangements of curves: simple (left) and non simple (right).

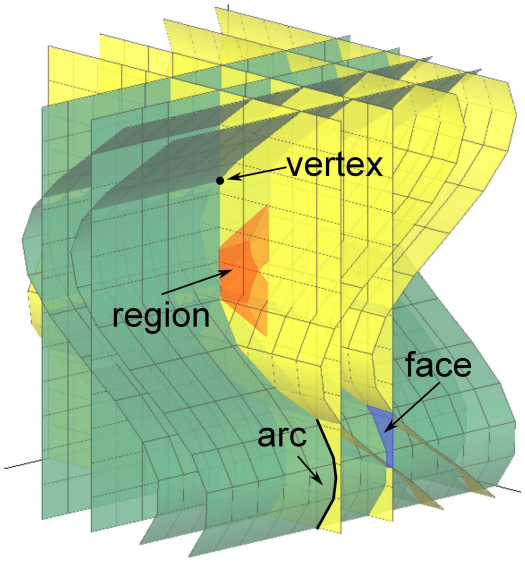


Figure 3.5: Example of 3D simple arrangement of surfaces.

3.2.1 Arrangement problem formalization

In the 2D space, an *arrangement* is defined as the decomposition of a finite collection C of curves in the plane [Edelsbrunner 1987, Snoeyink 1989, Halperin 2004]; these curves have no self-intersections, and are endless (either closed or bi-finite). Any two curves intersect in a finite number of points. The curves partition the plane into three types of maximally connected regions that are called cells of dimensions 0, 1 and 2: a 0-dimension cell (*a vertex*) is the intersection point of two curves of C, a 1-dimension cell (*an arc*) is a maximally connected portion of a curve that is not intersected by any other curve in C; and a 2-dimension cell (*a face*) is a maximally connected region of $\mathbb{R}^2$ that is not intersected by any other curve in C. An arrangement of C is a set of those cells and their relations, denoted by $A(C)$. An arrangement is *simple* if there are no three curves that intersect at a same point. An example of curve arrangements is shown in Figure 3.4. Various methods can be used to obtain the arrangement of curves: linear sweep construction [Edelsbrunner 1992], zone theorem [Edelsbrunner 1991b], topological sweep [Edelsbrunner 1991a], *etc.* A comprehensive discussion on the arrangement can be found in [Halperin 2004, chapter 24].

The problem of arrangement has been extended to the 3D space for the case of surfaces [Sharir 1999, Chan 2005]. Given a finite set S of surfaces, this set S partitions the 3D space into four types of cells: a *3-dimension cell* is a maximally connected region that is not divided by any other surfaces in S, while the other 2-, 1-, and 0-dimension cells are defined as above. An example is shown in Figure 3.5. It is shown in [Sharir 1999] that the surface-arrangement algorithm has a complexity of $\Omega(n^4)$ for general surfaces, where n is the number of surfaces. To compute a DRT graph, we only deal with the tipping surfaces. From Section 2.4.2, we know that tipping surfaces in the 3D parameter space (a, b, θ) can be fully described from their two cross-sections on the planes (a, θ) and (b, θ), respectively. These cross-sections are expressed by two sets of tipping curves. In addition, we are only interested in the information about regions and faces in the arrangement. Therefore, instead of using the basic algorithm of surface arrangement, we propose a variation of the sweeping method [Edelsbrunner 1991a] for constructing the DRT graph with a better complexity of $\mathcal{O}(n^3)$, where n is the number of tipping surfaces.

3.2.2 Sweeping algorithm for discrete rigid transformation graph construction

Let us formalize the specific case – when the surfaces are tipping surfaces – as follows. Given a set of tipping surfaces S, we would like to construct a graph modeling the subdivision of the parameter space (a, b, θ) induced by S. Such a graph is called a *DRT graph* (see Definition 2) and denoted by G. We recall that in G, each vertex is associated to a 3D open cell of the subdivision, and each tipping-surface segment shared by two adjacent 3D open cells, is associated to an edge. When projecting two families of tipping surfaces on the planes (a, θ) and (b, θ), we obtain the corresponding families of tipping curves (see Figure 2.9(b) in Section 2.4.2). Based on these relations that link tipping surfaces and tipping curves, instead of constructing directly the partition graph in the 3D parameter space (a, b, θ), we first consider the structures of graphs in the 2D planes (namely, the (a, θ) and (b, θ) planes), and then combine them to build a complete DRT graph in the 3D parameter space (a, b, θ).

The main idea of the sweeping method in 2D is that a *cut* is swept through all tipping curves on the plane, and allows us to construct a graph afterwards. We define such a cut for a plane – either (a, θ) or (b, θ) – denoted by γ, as a monotonic line intersecting exactly once for each tipping curve in the plane. A cut γ is then represented by its sequence of intersecting tipping curves and modeled by a directed graph G_γ according to its sequence, as illustrated in Figure 3.6.

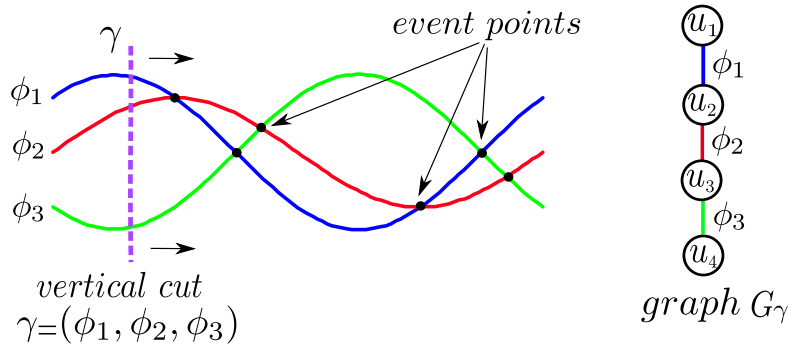


Figure 3.6: Example of a cut γ and its associated graph G_γ.

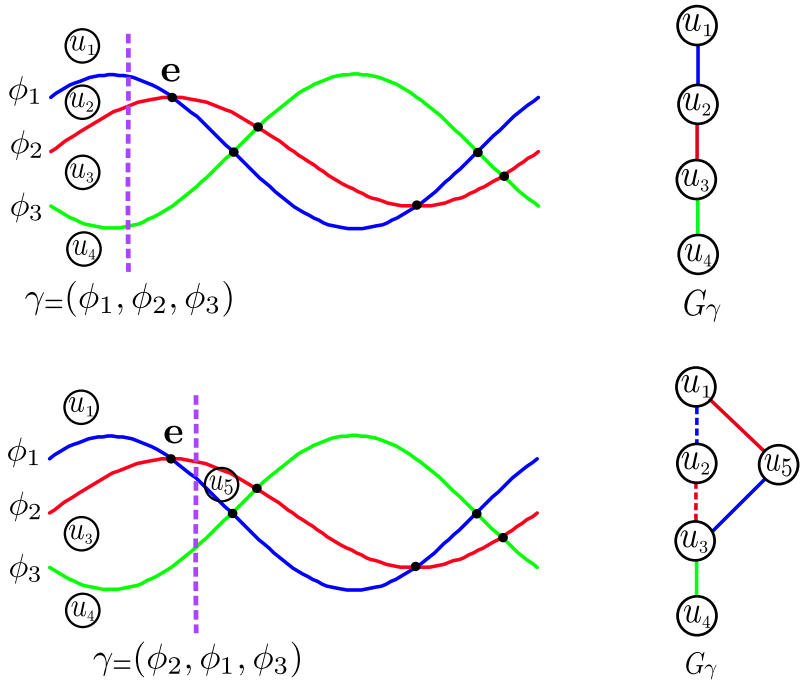


Figure 3.7: Progress of the cut at an event point **e** by which the cut γ is updated and the corresponding graph G_γ is modified.

We assume that the cut γ sweeps from $\theta = 0$ to $\theta = 2\pi$. While sweeping γ, its sequence changes in a discrete fashion. Indeed, as we observe in Figure 3.6, the sequence γ changes only at intersections of tipping curves, called *event points*. Therefore, instead of moving the cut continuously, we only need to maintain a set of sorted event points with respect to θ, and to progress in their increasing order to compute the graph incrementally. In particular, when γ reaches an event point, the algorithm performs an update of its sequence, and generates new vertices and edges in the graph. We call this an *elementary step* of the algorithm. The set of event points derives from a series of elementary steps.

For the simple cases (*i.e.,* when an event point is generated from the intersection of *exactly* two tipping curves), each elementary step products one new vertex and the associated cut γ changes after passing this event point by interchanging the order of the intersecting curves in γ, as illustrated in

Figure 3.7. There exist however some degenerate cases (or, non-simple cases), such as tangent and multiple intersections. They can be handled by modifying the simple case; instead of dealing with a pair of tipping curves, we now deal with a family of tipping curves that go through the intersection of each event point. The details of degenerate cases handling can be found in [Ngo 2013a].

In the sweeping method, all event points need to be detected and sorted. The coordinates of event points are irrational numbers whose value can not be computed exactly. Nevertheless, each value of θ can be represented by a pair of $\cos\theta$ and $\sin\theta$. These values $\cos\theta$ and $\sin\theta$ are further used in order to compute the a and b values. In [Ngo 2013a], we observed that a, b, $\cos\theta$ and $\sin\theta$ are all quadratic irrationals[1]. It is known in [Rosen 1992] that two quadratic irrationals can be compared by an exact method[2] using periodic continued fraction representation. Moreover, it is proved in [Flajolet 1998] that the average cost of the comparison of periodic continued fractions is in constant time. The fact of using integer arithmetic avoids technical problems due to the use of floating point representations such as the potential detection of "false" event points that may occur in degenerate cases.

We now explain the principle of the algorithm for building a DRT graph G in the parameter space of (a, b, θ). Two *cuts* are used such that each cut sweeps on either the plane (a, θ) or (b, θ). We denote these 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. By combining the two graphs G_{γ_a} and G_{γ_b}, we then generate a part of the DRT graph G, called a *partial graph* and denoted by δG. Typically, δG can be seen as a Cartesian product of G_{γ_a} and G_{γ_b}, as illustrated in Figure 3.8.

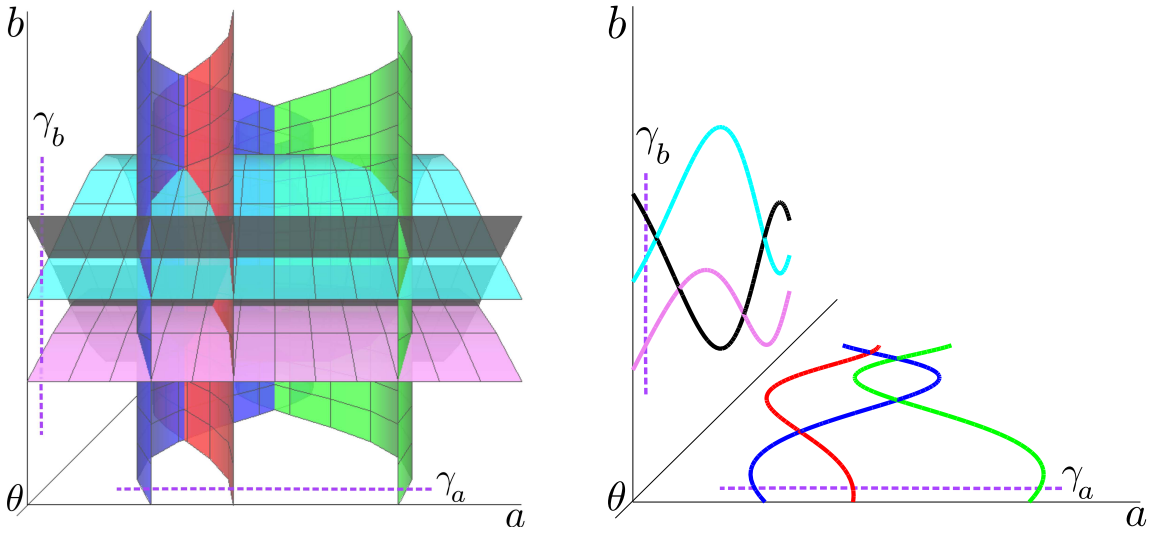
Then, at each event point, an elementary step is applied to G_{γ_a} or G_{γ_b}; the sweep progresses as the partial graph δG is generated and integrated in G for constructing the final graph. It is shown in [Ngo 2013a] that the proposed algorithm presents a linear time complexity with respect to the size of the DRT graph, which is $\mathcal{O}(N^9)$ where $N \times N$ is the image size. More details about the sweeping algorithm for tipping surfaces can be found in [Ngo 2013a].

3.3 Discrete rigid transformation graph under constraints

So far, we know that a DRT graph models the subdivision of the *whole* parameter space of rigid transformations. In other words, it models *all* the possible rigid transformations of a given subset of $\mathbb{Z}^2$ of size $N \times N$. It has been proved in [Ngo 2013a] that the DRT graph has a polynomial complexity $\mathcal{O}(N^9)$ (see Section 3.4). Due to this high complexity, it is difficult to involve directly a DRT graph in image analysis applications. To tackle this issue, two approaches can be considered. The first consists of using the DRT graph in a local fashion, thus reducing the space complexity of the research area where it is involved, as proposed in [Ngo 2013e, Ngo 2013b] (see Chapter 4). The sec-

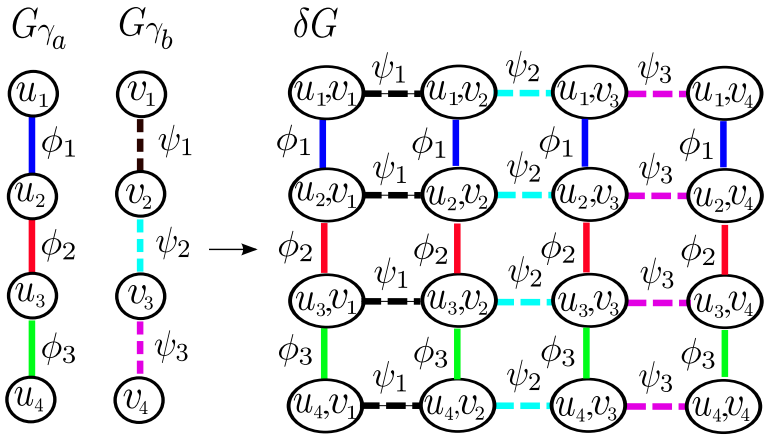
[1] A quadratic irrational is an irrational number that is a solution of some quadratic equations.

[2] An exact method employs only integers during the computation.



(a) Tipping surfaces in the parameter space (a, b, θ) with the cuts γ_a and γ_b on the planes (a, θ) and (b, θ) respectively.

(b) Tipping curves on the planes (a, θ) and (b, θ) with the associated cuts γ_a and γ_b.



(c) Generation of δG from G_{γ_a} and G_{γ_b}.

Figure 3.8: 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.

ond consists of providing spatial constraints in order to guide the transformations. Such constrained search paradigms are often used in image analysis and computer vision, *e.g.*, for matching, registration and warping purposes [Pennec 2000, Zitová 2003, Amintoosi 2011, Xiao 2011, Jyoti 2012]. In general, these constraints introduce prior knowledge on transformations that contribute to reduce the searching space. We now investigate this second approach from a combinatorial point of view. More precisely, we address the effects of geometric constraints, called *pixel–invariance constraints*.

Pixel–invariance constraints consist of forcing correspondences between points in an initial (sub)space of $\mathbb{Z}^2$ and transformed points – or more generally regions – in the transformed one, as illustrated in Figure 3.9.

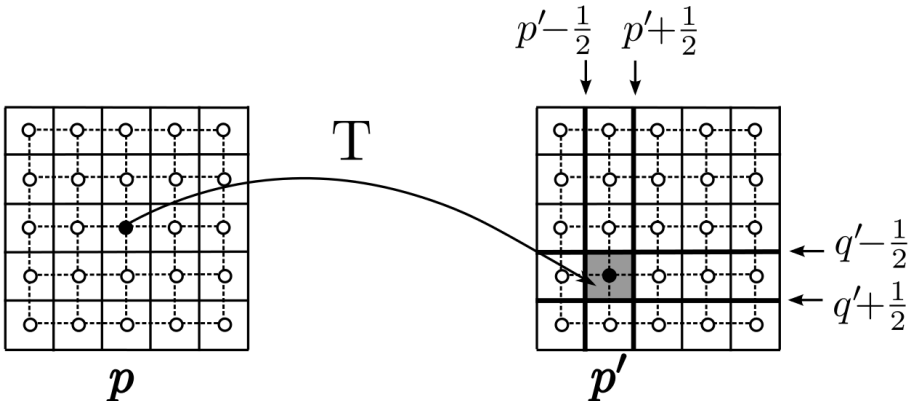


Figure 3.9: A pixel-invariance constraint establishes a correspondence between points $\boldsymbol{p} = (p, q)$ in the initial space and $\boldsymbol{p'} = (p', q')$ in the transformed. This leads to a restriction of the authorised rigid transformations in the parameter space.

Definition 4 (Pixel-invariance constraint) *Let $\boldsymbol{p} = (p, q) \in A \subset \mathbb{Z}^2$ and $\boldsymbol{p'} = (p', q') \in B \subset \mathbb{Z}^2$, such that A and B are of size $N \times N$. There exists a* pixel-invariance constraint *between $\boldsymbol{p}$ and $\boldsymbol{p'}$ if the authorised digital rigid transformation T between A and B satisfies $T(\boldsymbol{p}) \in \boldsymbol{p'} +]-\frac{1}{2}, \frac{1}{2}[$.*

More generally, there exist pixel-invariance constraints between two sets $\{\boldsymbol{p}_i\}_{i=1}^m \subseteq A$ and $\{\boldsymbol{p'_i}\}_{i=1}^m \subseteq B$ ($m \geq 1$) if $T(\boldsymbol{p}_i) \in \boldsymbol{p'_i} +]-\frac{1}{2}, \frac{1}{2}[$ for every $i \in [\![1, m]\!]$.

In the context of rigid transformations in $\mathbb{R}^2$, forcing correspondences between points in the initial space and in the transformed one leads to a restriction for transformations $\mathcal{T}$ (see Figure 3.10(a)). Moreover, forcing correspondences for two or more distinct pairs of points restricts the number of feasible transformations $\mathcal{T}$ to at most one (see Figure 3.10(b)). In $\mathbb{Z}^2$, restricting DRTs under similar constraints is more permissive. Indeed, when forcing correspondences between one or several pairs of pixels, a larger space of transformations may remain valid, as exemplified in Figure 3.10(c–f).

In absence of constraints, the subdivision of the parameter space (a, b, θ) is represented by a DRT graph studied in Section 3.1. Therefore, the *whole* volume of the parameter space models adequate rigid transformations. By contrast, under pixel-invariance constraints, some DRTs may become irrelevant. Equivalently, only a part of the parameter space – namely, a subspace of (a, b, θ) that satisfies these constraints – remains valid.

Geometrically, the subspace induced by a constraint of one pixel pair, is defined by the intersection of four half-spaces associated to four tipping surfaces related to this constraint. Practically, it corresponds to a tube-like volume as illustrated in Figure 3.10(c). For two pixel-invariance constraints, the parameter subspace of feasible transformations is the intersection of such tubes generated by these constraints, and it is a bounded and connected set, or possibly becomes empty. Figure 3.10(d) shows the parameter subspace for two given constraints.

In particular, the subspace obtained from m pixel correspondences, is divided into DRTs induced

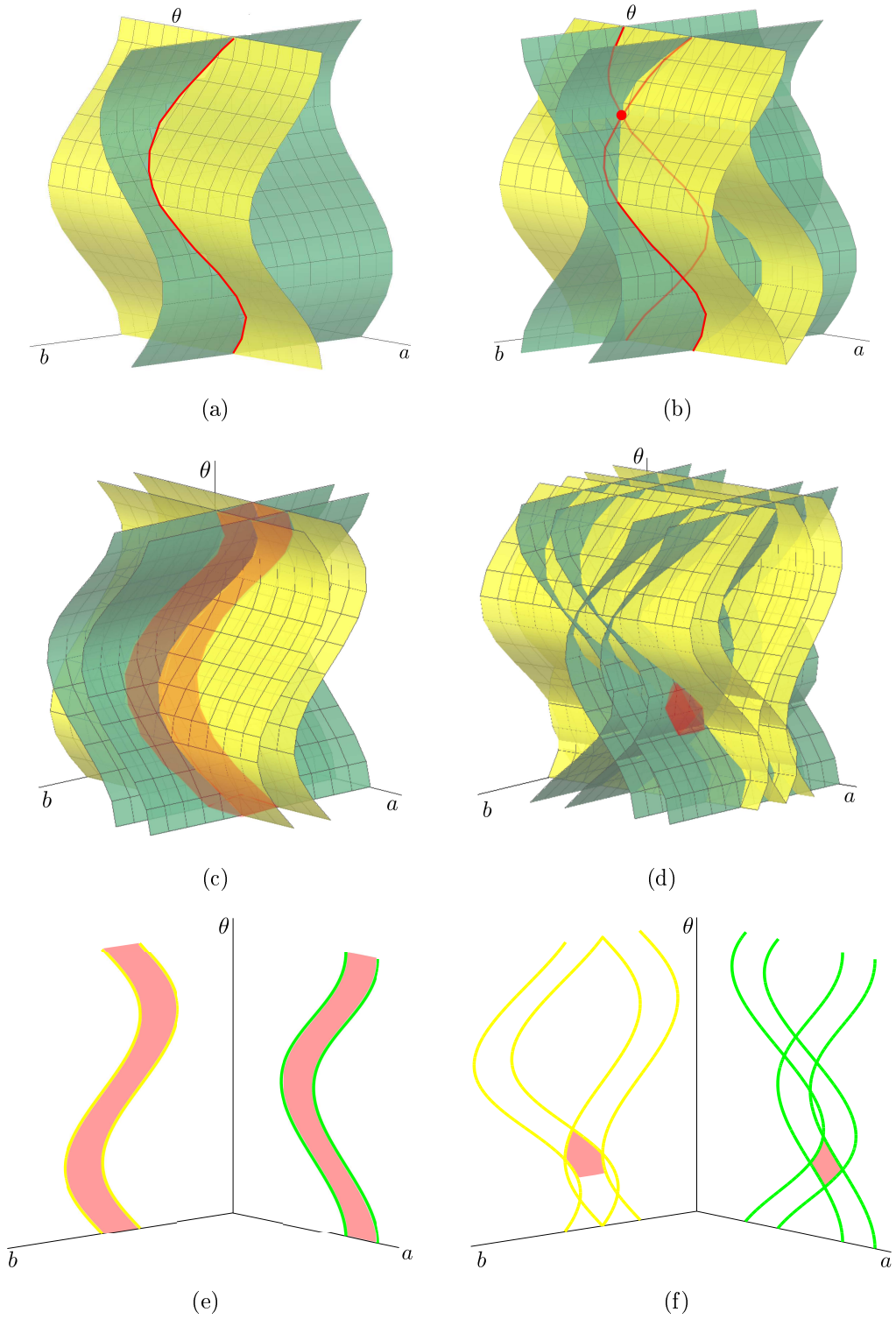


Figure 3.10: Feasible rigid transformations induced by spacial constraints in continuous (a,b) and discrete (c–f) frameworks. (a) Transformations with one-point correspondence (red curve). (b) Transformation with two-point correspondences (red dot at the intersection of the two red curves). (c) Transformations with one-pixel correspondence (red tube-like volume). (d) Transformations with two-pixel correspondences (red volume). (e,f) Projection/intersection of the red volume parts of (c,d) respectively on the planes (a, θ) and (b, θ) with the associated tipping curves.

by the (N^2-m) remaining pixels of the given image of size $N\times N$. Thus, the combinatorial structure of DRTs in this subspace, modeling its subdivision, is a subgraph of the DRT graph of the image of size $N \times N$. The method presented in Section 3.2 can be modified to build this subgraph in linear time with respect to the size of the graph. More details on this algorithm can be found in [Ngo 2013c].

3.4 Space complexity of discrete rigid transformation graphs

The space complexity of a DRT graph is proportional to its number of vertices and edges. These values directly depend on the number of event points involved in its construction, and the number of vertices and edges generated at each event point. As mentioned in Section 3.2, the construction of a DRT graph G in the parameter space (a, b, θ) is obtained from its projections on the planes (a, θ) and (b, θ). Therefore, the number of event points is the number of intersecting points on the plane either (a, θ) or (b, θ). It is proved in [Ngo 2013a] that for a given set of n tipping curves, this number of event points is at most $n(n-1)$.

For a subset of $\mathbb{Z}^2$ of size $N \times N$, we have tipping surfaces (resp. curves) Ψ_{pqk} and Φ_{pql} (resp. ψ_{pqk} and ϕ_{pql}) with $p, q \in [\![0, N-1]\!]$ and $k, l \in [\![0, N]\!]$. Thus, we have $n = \mathcal{O}(N^3)$, which is the size of the cut as well. The number of event points is then $\mathcal{O}(N^6)$. At each elementary step, there are one vertex and two edges generated on a plane, either (a, θ) or (b, θ). As mentioned in Section 3.2, the construction of a DRT graph G is obtained from its projections on the planes (a, θ) and (b, θ) using two cuts. Since the size of each cut is $\mathcal{O}(N^3)$, the numbers of vertices and edges generated at each elementary step is thus $\mathcal{O}(N^3)$. Finally, (i) the number of event points of the whole space is $\mathcal{O}(N^6)$, and (ii) at each elementary step (*i.e.*, for each event point), $\mathcal{O}(N^3)$ vertices are generated in the partial graph δG_i of G. This justifies the following result.

Theorem 5 ([Ngo 2013a]) *The DRT graph G associated to an image of size $N \times N$ has a space complexity of $\mathcal{O}(N^9)$.*

Given one pixel-invariance constraint, some DRTs become irrelevant. Following a similar proof scheme as below, we can show that the number of event points (i) decreases from $\mathcal{O}(N^6)$ to $\mathcal{O}(N^5)$ (see Property 4 in [Ngo 2013a]), and (ii) at each elementary step, $\mathcal{O}(N^2)$ vertices are generated instead of $\mathcal{O}(N^3)$ since δG is generated from two graphs G_{γ_a} and G_{γ_b} associated to the cuts γ_a and γ_b. Each of the cut intersects at most $\mathcal{O}(N^2)$ tipping curves on the plane. Then at each intersection, there are $\mathcal{O}(N^2)$ vertices generated in δG. This leads to the following result.

Theorem 6 ([Ngo 2013c]) *The DRT graph G associated to a digital image of size $N \times N$ under one pixel-invariance constraint has a space complexity of $\mathcal{O}(N^7)$.*

For more than one pixel-invariance constraint, we can not use a similar approach to obtain the theoretical upper bound complexity of G induced by these constraints. This is due to the fact that the space complexity of the DRT graph under constraints does not only depend on the number of constraints, but also on the geometric configuration of the points involved in these constraints. In the worst case, it is shown in [Ngo 2013c] that the space complexity does not exceed $\mathcal{O}(N^7)$. It is however experimentally observed that this complexity tends to decrease when more than one constraints are considered, and in particular, when the distance between the constraints increases. Following this observation, in [Ngo 2013c] a conjecture has been proposed in which the complexity of G is linked to the distance between points of given constraints.

Conjecture 7 ([Ngo 2013c]) *The DRT graph G associated to a digital image of size $N \times N$ under two pixel-invariance constraints has a space complexity of $\mathcal{O}(N^{5.5}d^{-1.6})$, where d denotes the Euclidean distance between two pixel-invariance constraints.*

3.5 Summary

We have introduced in this chapter a combinatorial structure (namely, a discrete rigid transformation graph or DRT graph) for representing the subdivision of the parameter space of rigid transformations. Such a structure models all DRTs and their relations for any considered subspace of $\mathbb{Z}^2$ of size $N \times N$, and presents a complexity of $\mathcal{O}(N^9)$. We have also shown that under pixel-invariance constraints, this complexity decreases; typically under one constraint, the complexity of the DRT graph becomes $\mathcal{O}(N^7)$, and tends to decrease for several constraints. An algorithm has been proposed to compute such DRT graph in linear time with respect to its size.

In particular, this graph explicitly models the "neighboring" links between DRTs. This contributes to (*i*) a better understanding of the topological behavior of rigid transformations in $\mathbb{Z}^2$, and (*ii*) the generation of all transformed images by incrementally modifying (at most) one pixel value between two successive transformed images (see Remark 3 and Figure 3.11). The DRT graphs provide a purely discrete description of rigid transformations on $\mathbb{Z}^2$ and allow to study, implement and utilise such transformations as a single process. Beyond these combinatorial and algorithmic aspects, this novel tool can also be involved in image processing tasks, *e.g.*, image matching [Hundt 2009], registration or warping [Pennec 2000, Ngo 2013g]. In particular, the DRT graph can be used in imaging applications based on sub-sample analysis, *e.g.*, in pattern-based strategies for analysing the topological invariance of images [Ngo 2013b], or patch-based methods for image denoising application [Buades 2005].

In the next chapter, we show how it contributes to the understanding of the relationships existing between geometry and topology in the framework of computer imagery, where both notions are more strongly linked than in continuous spaces.

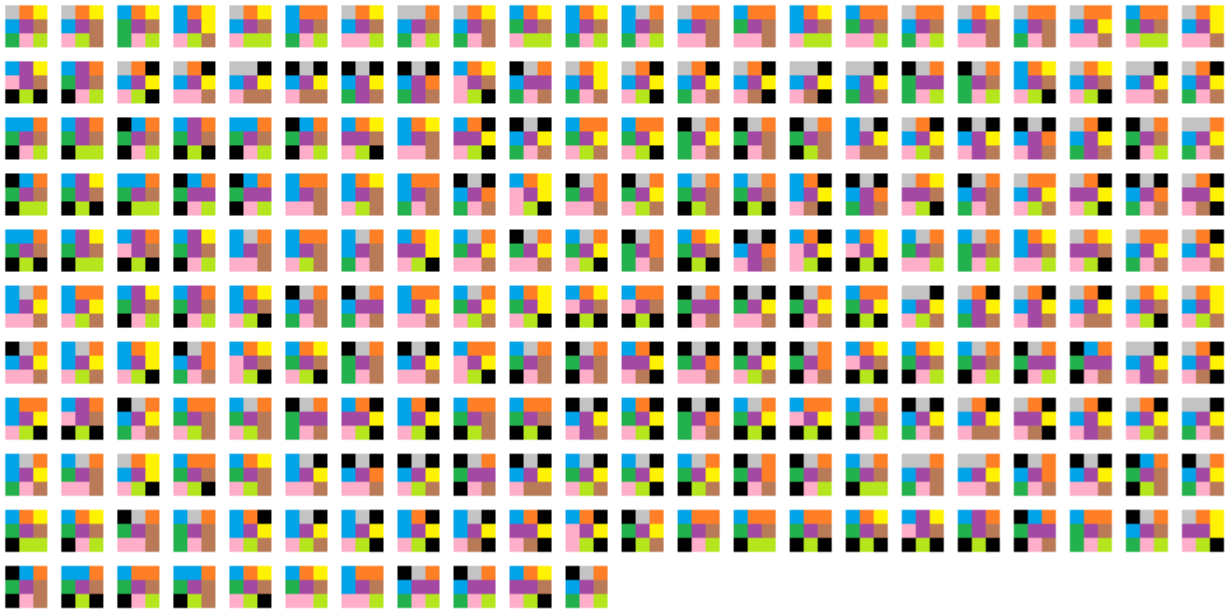


Figure 3.11: The 231 images generated by using the Eulerian model from the part of a DRT graph for an image of size 3×3 where $a, b \in \ ]\frac{1}{2}, \frac{3}{2}[$ and $\theta \in \ ]0, \frac{\pi}{4}[$. The first image is obtained by the identity transformation. We do not show pixels that are outside of the image support, and background pixels are formally colored in black.

Chapter 4

Topological image analysis under rigid transformations

Contents

In $\mathbb{R}^2$, rigid transformations are topology-preserving operations. This property is generally lost when such transformations are applied on digital images. Indeed, due to the mandatory digitization from $\mathbb{R}$ to $\mathbb{Z}$, digital rigid transformations, as shown in Chapter 2, do not preserve distances, as well as angles between points. These geometrical changes lead to the alterations of topology in the transformed space, as exemplified in Figure 4.1. In this chapter, we investigate this specific issue that is of high importance in digital image processing and analysis. Indeed, the topology-preserving geometric deformations of digital images are involved in various applications such as atlas-based segmentation [Bazin 2007a, Faisan 2008], clustering analysis [Pham 2001, Bazin 2007b] and medical imaging [Lam 1992, Tustison 2011]).

In particular, we address the following questions:

(i) Do there exist digital images that preserve their topology under all possible rigid transformations?
(ii) Is it possible to verify topological preservation for a given image?
(iii) What are the conditions for digital images to preserve their topology?

To this end, we introduce a notion of *topological invariance*, which characterizes the preservation of topological properties of digital images under rigid transformations. Based on the notions of DRT graphs and of simple points [Rosenfeld 1970, Couprie 2009], we provide an approach to evaluate

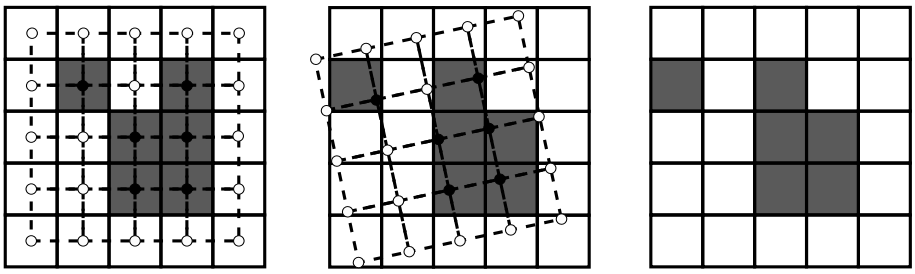


Figure 4.1: Left: binary digital image and the grid modeling its discrete structure. Middle: a rigid transformation applied on this grid. Right: the resulting transformed image, which is not topologically equivalent to the initial image (for instance, the black component is split if it is considered to be 8-connected).

the topological invariance of an image by scanning its whole DRT graph. Then, we show that this global approach, which presents a high complexity, can be simplified into a local approach, based on a spatial decomposition of the image into covering patterns. The study of these patterns provides sufficient conditions to assess the topological invariance of images in quasi linear time.

For the sake of readability, our study is carried out in the context of binary images. However, the obtained results remain relevant for other kinds (*e.g.*, grey-level, labeled) of digital images, as discussed in Section 4.4.

The work exposed in this chapter was presented in [Ngo 2013e, Ngo 2013b].

4.1 Digital topology: background notions

In this section, we recall some basic concepts of digital topology, which studies the topological properties of discrete objects on $\mathbb{Z}^n$. Several frameworks have been proposed to deal with digital topology, *e.g.*, Khalimsky grids [Khalimsky 1987], abstract cell complexes [Kovalevsky 1989], or topology defined on the Jordan surfaces [Herman 1998]. Most of these frameworks regard the topological spaces as finite combinatorial ones. It is shown in [Mazo 2012a] that in $\mathbb{Z}^2$, these frameworks can be conveniently unified within the frequently used graph-based approach proposed by A. Rosenfeld and J. Pfaltz in [Rosenfeld 1966].

In the context of binary images, the discrete space of images is divided into two graphs, on which we define the relations between points of foreground (*e.g.*, black points) and those of background (*e.g.*, white points). These relations are first defined by adjacencies (that correspond to the edges of the graphs modeling the image), and then lead to notions of connectedness and connected components, associated to these adjacencies. In 2D, we generally use the classical 4- and 8-adjacencies, which are illustrated in Figures 4.2 and 4.3.

It is well known that, to deal with topological paradoxes [Rosenfeld 1966, Kong 1989] related to the discrete version of the Jordan theorem[1], we generally use a pair of dual adjacency relations

[1]The Jordan curve theorem, also called separation theorem, asserts that a simple closed curve separates

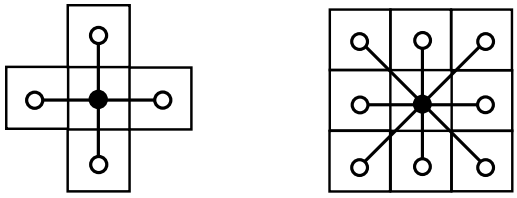
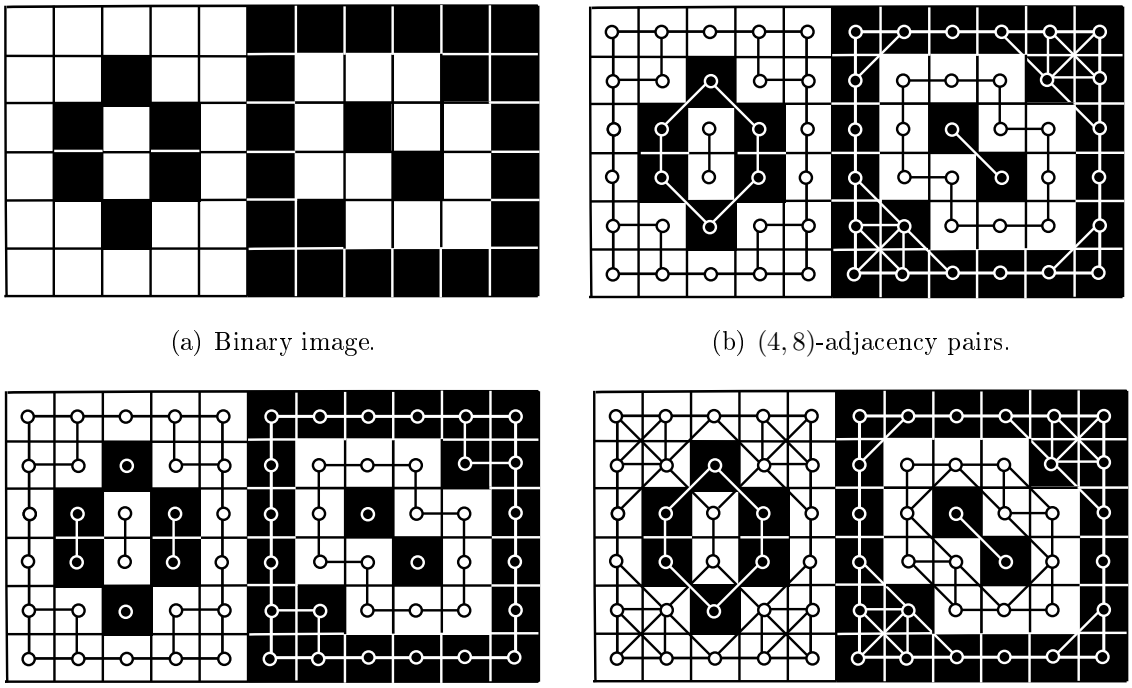


Figure 4.2: Left: the four 4-adjacent points (or 4-neighbors) of the center point are its vertical and horizontal neighbors. Right: the eight 8-adjacent points (or 8-neighbors) of the center point are its vertical and horizontal neighbors together with the four diagonal ones.



(a) Binary image. (b) $(4,8)$-adjacency pairs.

(c) $(4,4)$-adjacency pairs. (d) $(8,8)$-adjacency pairs.

Figure 4.3: (a) A binary image. (b–d) Adjacency graphs of the black and white points in (a), in which we use 4- and 8-adjacencies (b), 4- and 4-adjacencies (c) and 8- and 8-adjacencies (d) for black and white points respectively.

[Duda 1967] for the foreground and the background. More precisely, we consider 4-adjacency for black points and 8-adjacency for white points, or vice-versa, and we call these configurations $(4,8)$- and $(8,4)$-adjacency relations, respectively.

Our goal is to investigate the preservation of topological properties – more precisely, the preservation of the homotopy-type – of digital images under rigid transformations. In particular, we aim to provide conditions on digital images for the preservation of such topological properties. In digital topology, homotopy-type preservation can be dealt with via the notion of *simple point* [Rosenfeld 1970, Couprie 2009]. Indeed, a homotopic transformation can be regarded as a sequence of deletion and/or addition of (successively) simple points. Practically, a point of a binary image

the plane into exactly two components, which can be interpreted as the interior and exterior of the curve.

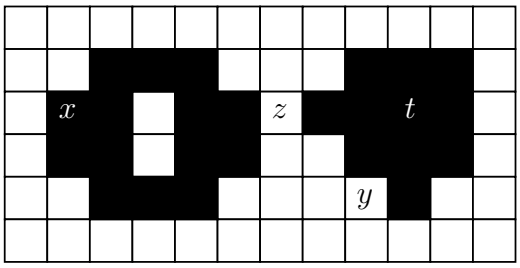
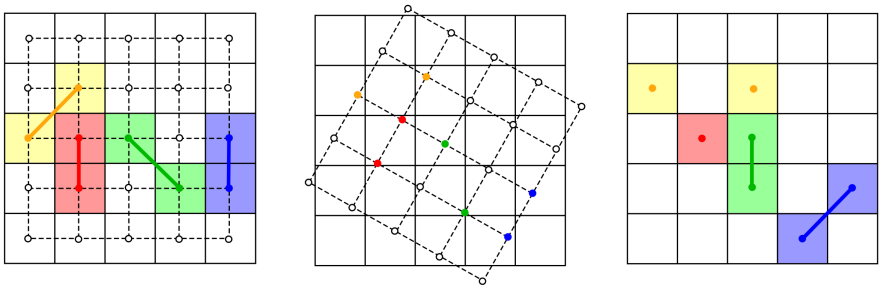


Figure 4.4: Some simple $(\boldsymbol{x}, \boldsymbol{y})$ and non-simple $(\boldsymbol{z}, \boldsymbol{t})$ points. Modifying the value of $\boldsymbol{z}$ would merge two black connected components, while modifying the value of $\boldsymbol{t}$ would create a white connected component. In both cases, the homotopy-type of the image would be modified.



(a) (b) (c)

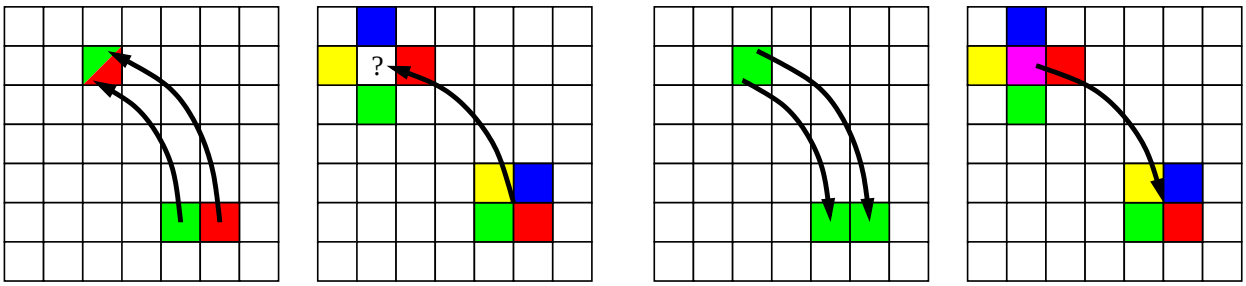
Figure 4.5: Examples of alteration of adjacency pairs between points induced by digital rigid transformations: (a) an original image with the associated grid; (b) a rigid transformation applied on the grid of (a); (c) the digitized result of (b). We observe that the two points in blue (resp. green) which were 4-adjacent (resp. 8-adjacent) in (a) become 8-adjacent (resp. 4-adjacent) in (c), and the two points in red (resp. yellow) which were 4-adjacent (resp. 8-adjacent) in (a) are merged (resp. are neither 4- nor 8-adjacent) in (c).

I is *simple* if its value can be switched without modifying the topological properties of I (see Figure 4.4). For various kinds of topological structure [Khalimsky 1987, Kong 1989, Kovalevsky 1989] mapped on I, and for other kinds of value spaces (grey-level images, label images), the simplicity of a point can be tested, in constant time, by only studying its 3×3 neighborhood.

4.2 Topological issues on the discrete structure of images

As observed in Section 2.3, the non-preservation of distances and angles between points, when applying a digital rigid transformation, results in modifications of the adjacency relations between such points, as illustrated in Figure 4.5. In particular, some connected components may be either split or merged, as illustrated in the first and second configurations depicted in Figure 4.7(d).

Beyond alterations of adjacency relations, we can also observe "cardinality-based" issues, due to the existence of null and double points (see Section 2.2). On one hand, such points can cause topological alterations, *e.g.*, some connected components may be lost when applying a digital rigid transformation, as exemplified in the third configuration depicted in Figure 4.7(d). On the other hand, they can raise topological issues in both Lagrangian and Eulerian transformation models



(a) Lagrangian model. (b) Eulerian model.

Figure 4.6: The effects of double (left) and null (right) points are exemplified in the context of digital rigid transformations in the Lagrangian (a) and Eulerian (b) models.

(see Section 2.1.3). Indeed, as illustrated in Figure 4.6, null and double points lead to different interpretations in the two models. In the Lagrangian model (see Figure 4.6(a)), a double pixel in the transformed space may receive two different pixel values, and a null pixel receives no value. This leads to a result being both incomplete and ambiguous in the case of binary images and in particular for multivalued images. In the Eulerian model (see Figure 4.6(b)), a double pixel of the initial space may transfer its value to two pixels of the transformed space, while the value of a null pixel will be lost. This may be conveniently handled in the case where images are considered in a set-based paradigm; moreover, there are no incomplete and ambiguous issues as in the Lagrangian model.

In this chapter, we consider the Eulerian model, which allows us to focus on the topological issues raised by the adjacency alterations. The Lagrangian model implies additional difficulties that have not been dealt with in this work, and that will be discussed in Chapter 5.

4.3 Topological invariance of digital images under rigid transformations

Let us consider a 2×2 pixel sample of the transformed space (see Figure 4.7(a)), and all the possible local pixel configurations of the initial space from which the sample is generated (see Figure 4.7(b)). Despite local alterations among pixels, the topologies of some generated samples may be preserved (see Figure 4.7(c)). However, such local alterations may also lead to topological alterations in some samples (see Figure 4.7(d)), and further in the whole image. Consequently, the study of topological alterations induced by digital rigid transformations must be considered not only as a transformation-dependent problem, but also as an image-dependent one.

We would like to know if a given image I preserves its topological properties under all digital rigid transformations, and furthermore what conditions on I to preserve such properties. As stated above, we focus on the homotopy-type as topological invariant. Let us first formalise the preservation of homotopy-type via the notion of *topological invariance*.

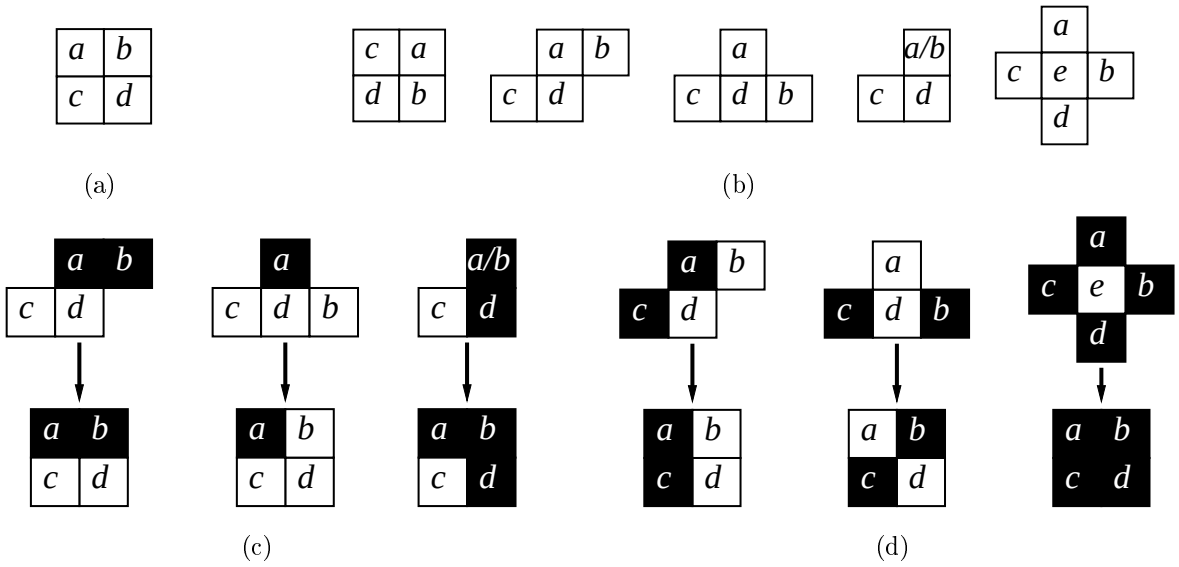
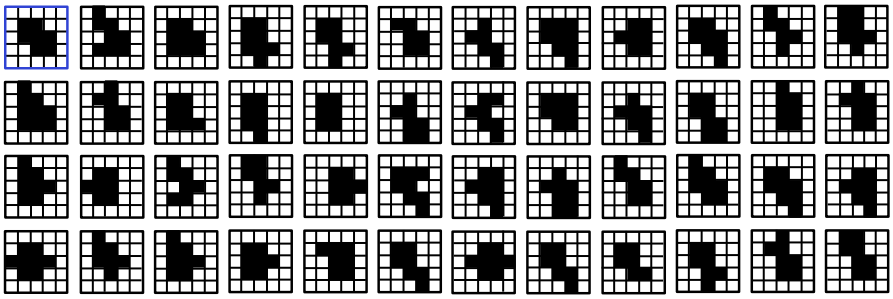
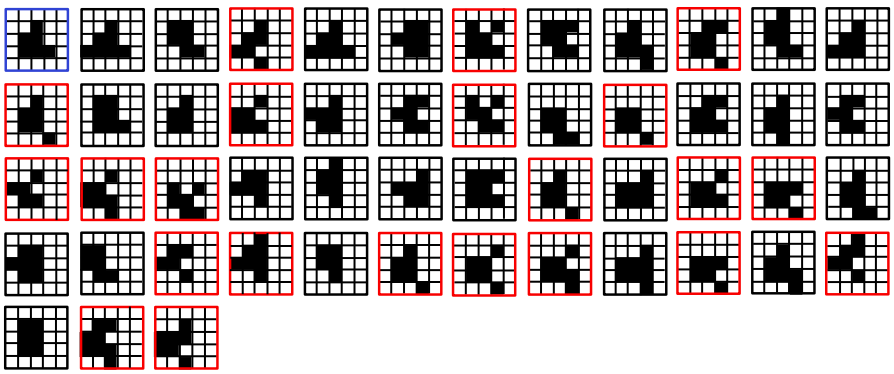


Figure 4.7: (a) A 2×2 pixel sample with values a, b, c, d. (b) All the local pixel configurations (up to rotations and symmetries) leading to the sample in (a) when applying digital rigid transformations. (c) Examples of transformations in which the samples preserve the topology of local pixel configurations. (d) Examples of transformations in which the samples are topologically altered.



(a) Topologically invariant image.



(b) Topologically variant image.

Figure 4.8: (a) A topologically invariant 5×5 image (in blue frame), and all its transformed images (up to symmetries, in black frames) generated by using its DRT graph. (b) A topologically variant 5×5 image (in blue frame), and all its transformed images (up to symmetries) generated by using its DRT graph. Those whose homotopy-type is equivalent are depicted in black. Those whose homotopy-type differs are depicted in red.

Definition 8 (Topological invariance) *We say that a digital image I is* topologically invariant *if all its rigidly transformed images have the same homotopy-type as I.*

Figure 4.8 provides some examples of binary images being topologically invariant and topologically variant. In the next section, we will develop methods to verify such topological invariance of a given image I defined on $\mathbb{S}$.

4.4 Topological invariance verification

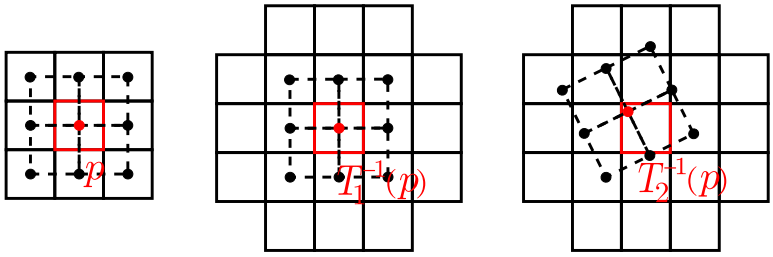
4.4.1 Discrete rigid transformation graph as a topological analysis tool

An image I is topologically invariant if all its transformed images share the same homotopy-type. On one hand, from Chapter 3, we know that the DRT graph G of I models *all* the possible rigid transformations. It is important to recall that G deforms the image progressively, *i.e.*, pixel by pixel (see Remark 3 and Figure 3.3). On the other hand, from Section 4.1, we know that the notion of simple points can be used to characterise the homotopy-type preservation between two images. It follows that an image I is topologically invariant if all its transformed images are obtained by successively modifying the values of simple points. The local characterization of simple points is then compliant with an incremental exploration of the DRT graph of the image I, in order to evaluate its topological invariance.

More precisely, in the DRT graph G, any elementary modification[2] between transformed images is encoded in an edge. Consequently, by analyzing the edges of the whole DRT graph G, the topological invariance of I can be verified. Practically, the edges of the DRT graph $G = (V, E)$ with respect to I can be classified in two categories: those that do not modify the topology of I (*i.e.,* the edges that correspond to the case (i) and parts of the case (ii) for which the changing pixel $\boldsymbol{q}$ is a simple point, in Remark 3) and those that do modify the topology (the edges that correspond to parts of the case (ii) for which $\boldsymbol{q}$ is not simple, in Remark 3). Based on this binary classification, we can straightforwardly create a partial graph $G' = (V, E')$ of G by preserving in $E' \subseteq E$ only the edges of the first category. In particular, if G' is connected, then I is topologically invariant. Otherwise, I is not topologically invariant, and every connected component in G' corresponds to a set of transformed images of same homotopy-type. Such an approach presents an algorithmic complexity that is linear with respect to the (polynomial) space complexity of the DRT graph. The details of this algorithm can be found in [Ngo 2013b].

Despite the theoretical interest of the proposed approach, its high algorithmic complexity practically forbids to use it for large images, and therefore to actually evaluate their topological invariance. In the next section, we show that this problem can however be decomposed spatially, thus leading to a much lower complexity algorithm.

[2]The modification of exactly one point between two transformed images.



(a) $N_8(\boldsymbol{p})$ (b) $N_{20}(T_1^{-1}(\boldsymbol{p}))$ (c) $N_{20}(T_2^{-1}(\boldsymbol{p}))$

Figure 4.9: Any digital transformation maps a 8-neighborhood into at most a 20-neighbohood. (a) In the transformed space, 8-neighbors of the center point $\boldsymbol{p}$. (b,c) In the initial space, 20-neighbors of the transformed point of $\boldsymbol{p}$, for two digital transformations T_1, T_2 respectively.

4.4.2 A local analysis for evaluating topological invariance

In the previous section, we have proposed to explore the whole DRT graph of an image I in order to check its potential topological invariance. More precisely, for each edge of the DRT graph, we verified if the modified pixel is actually a simple point or not. This test was performed *locally*, more precisely in the 3×3 neighborhood centered on the considered pixel in the transformed space [Kong 1989]. We now take advantage of the local nature of this test to develop a spatial decomposition strategy that leads to a *local* version of the previously proposed *global* method. To this end, we first need to introduce basic properties related to the influence of DRTs on pixel neighborhoods.

Property 9 *Let $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^2$, $\boldsymbol{q} \in N_8(\boldsymbol{p})$ and a digital rigid transformation T. We have $T^{-1}(\boldsymbol{q}) \in N_{20}(T^{-1}(\boldsymbol{p}))$, where $N_8(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{Z}^2 \mid ||\boldsymbol{x} - \boldsymbol{y}||_2 < 2\}$ and $N_{20}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{Z}^2 \mid ||\boldsymbol{x} - \boldsymbol{y}||_2 < 2\sqrt{2}\}$.*

This property, illustrated in Figure 4.9, derives from the fact that a digital rigid transformation T implies a possible increase of $\sqrt{2}$ for distances between transformed points, with respect to its associated rigid transformation $\mathcal{T}$.

Let us now come back to the DRT graph G of I involved in the global process defined in the previous section. Let $e = (v, w, f)$ be an edge of G, I_v and I_w be the associated transformed images of the vertices v and w, respectively. We assume that I_v and I_w differ at the pixel $\boldsymbol{p'}$, and that $\boldsymbol{p}$ is the pixel corresponding to $\boldsymbol{p'}$ in the initial image I. On one hand, from Property 9, any edge, that does not involve any change of pixels in $N_{20}(\boldsymbol{p})$, has no influence on topological modifications in $N_8(\boldsymbol{p'})$. Therefore, topological modifications at $\boldsymbol{p'}$ in $N_8(\boldsymbol{p'})$ (and thus of $\boldsymbol{p'}$ in the whole image) only depend on the part of the DRT graph that corresponds to the restriction of I to $N_{20}(\boldsymbol{p})$, denoted by $I_{|N_{20}(\boldsymbol{p})}$. On the other hand, thanks to the integer translation invariance of I, it is plain that the set of all DRTs defined on a subset of $\mathbb{Z}^2$ does not depend on the way to locate this subset into $\mathbb{Z}^2$. Therefore, the DRT graph G associated to $I : \mathbb{S} \to \mathbb{V}$ is isomorphic to the DRT graph of any translated image of I by an integer vector. Based on these considerations and the local characterization of simple points, we then derive that the topological invariance for every

pixel $\boldsymbol{p}$ of I can be verified locally by using the DRT graph associated to $I_{|N_{20}(\boldsymbol{p})}$. This leads to the following sufficient condition that guarantees homotopy-type preservation of an image under rigid transformations.

Proposition 10 *Given a binary image $I : \mathbb{S} \to \mathbb{V}$, if $I_{|N_{20}(\boldsymbol{p})}$, for every $\boldsymbol{p} \in \mathbb{S}$, is a local binary sample which is topologically invariant under all digital rigid transformations, then the image I is topologically invariant.*

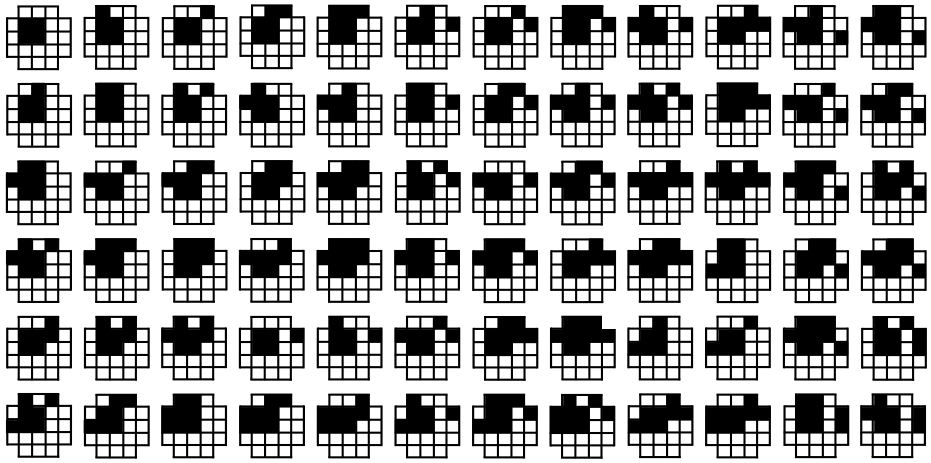
As a finite number of binary images can be defined on a 20-neighborhood sample, this topological analysis can be performed beforehand (and only once) for all the images defined on such a sample. Then, these images, stored in a look-up table (LUT), can be used to characterize locally the topological invariance of I.

Practically, for a given value space $\mathbb{V} = \{0, 1\}$ and a given adjacency (namely, $(4, 8)$ and $(8, 4)$) we can generate 2^{20} images defined on a 20-neighborhood sample, and any image $I : N_{20}(\mathbf{0}) \to \mathbb{V}$ has to be processed via its DRT graph to generate the LUT. In [Ngo 2013b], we obtain LUTs of 10 643 and 19 446 topologically invariant patterns in $(4, 8)$- and $(8, 4)$- adjacency pairs respectively. Figure 4.10 shows some examples of such LUTs. A full description of the proposed algorithm as well as experiments can be found in [Ngo 2013b].

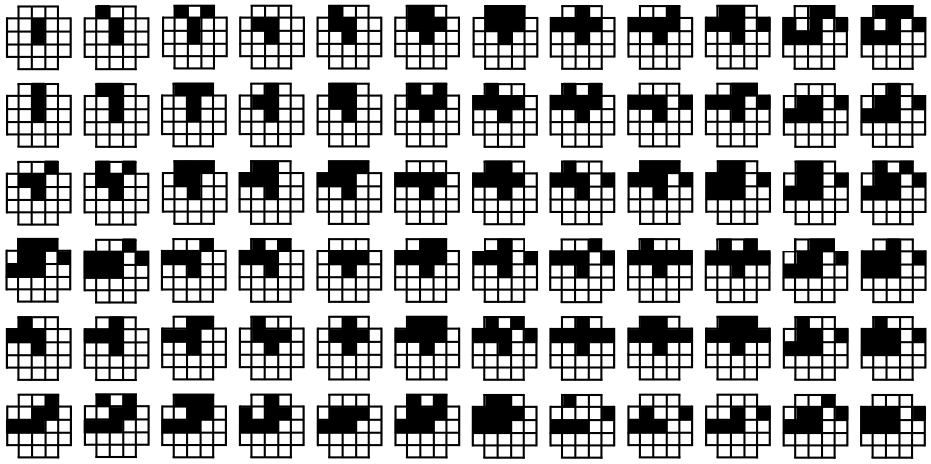
So far, the process has been presented for binary images. Nevertheless, from a methodological viewpoint the concepts developed can be conveniently extended to other categories of digital images (*e.g.*, well-composed sets [Latecki 1995], grey-level images [Couprie 2001] and n-ary images [Mazo 2012b]), since it only requires to have a notion of simple points (or more generally, a local characterisation of topology preservation). Indeed, on one hand, the proposed approaches rely on the notion of DRT graphs, which defines explicitly the structure of the transformed images, but does not require any knowledge on the image values. On the other hand, the topological space, that is mapped on $\mathbb{Z}^2$ with respect to the value space, is only considered via the notion of simple point. The only constraint related to the choice of topology is the necessity to characterize locally the preservation of homotopy-type with respect to images of $\mathbb{S} \to \mathbb{V}$. Consequently, the proposed approach can be parametrized by a couple composed of (i) a value space $\mathbb{V}$, and (ii) a notion of simple points for the space of images. This matter is discussed in [Ngo 2013b].

4.5 Summary

In this chapter, we have proposed an algorithmic process for determining the topological invariance of digital images under all discrete rigid transformations. This work is based on the notions of simple points and DRT graphs. The latters present a polynomial complexity that generally forbids its practical application on large images. Nevertheless, DRT graphs have been successfully involved in a local analysis that finally leads to a low complexity methodology, relying on image spatial decomposition.



(a) Topologically invariant patterns in $(4,8)$-adjacency pairs.



(b) Topologically invariant patterns in $(8,4)$-adjacency pairs.

Figure 4.10: Some binary patterns defined on a 20-neighborhood sample, which are topologically invariant under rigid transformations in the case of $(4,8)$-adjacency pairs (a), and $(8,4)$-adjacency pairs (b).

This work contributes to the better understanding of the relationships that exist between geometry and topology in the framework of computer imagery. In [Serra 1983, Heijmans 1992, Latecki 1998], the notion of regularity has been proposed to assess the preservation of topological properties during the digitization of an image from $\mathbb{R}^2$ to $\mathbb{Z}^2$. In the considered framework, the proposed algorithm may provide an efficient tool for studying/verifying this notion of regularity in the discrete space of digital images to guarantee the topological invariance properties of images under all digital rigid transformations.

This study was carried out in the context of binary images. However, it remains relevant, e.g., for grey-level images, and then constitutes a tool that is actually useful in applications in computer vision [Pham 2001, Bazin 2007b], or medical imaging [Lam 1992, Tustison 2011].

Chapter 5

Conclusion

Contents

This thesis focuses on studying rigid transformations on 2D digital images from combinatorial and algorithmic points of view. We have observed that discrete rigid transformations present specific properties compared with their continuous analogues. In this context, we have developed a novel framework for studying, implementing and utilizing rigid transformations on $\mathbb{Z}^2$. We recall in Section 5.1 our main contributions, and in Section 5.2, we provide some perspectives related to this work.

5.1 Contributions

We have proposed a purely discrete formulation for rigid transformations on digital images, and in particular a graph structure that models the parameter space of these transformations. This formulation allows us to describe *all* the possible DRTs of any subset of $\mathbb{Z}^2$ of size $N \times N$ with a complexity of $\mathcal{O}(N^9)$. In this context, a new concept related to the *neighboring* relationship between DRTs has been introduced to model the transitions by *only* one pixel between the transformations. Thanks to this tool, we can generate exhaustively and progressively *all* transformed images via elementary image modifications. Beyond such combinatorial considerations, this tool also permits the study of the relationships between rigid transformations and topology in the context of computer imagery. In particular, we have developed an algorithm allowing to assess the topological behaviors of an image under rigid transformations in quasi-linear time with respect to the image size, and further provided sufficient conditions for topological invariance of any image under these transformations.

From a theoretical point of view, this work contributes to improve our understanding of rigid transformations and their relationships with topology in the discrete space of digital images. It also provides some combinatorial as well as algorithmic answers in the research field of combinatorial image analysis. From a practical point of view, the proposed data structure may be involved in applications of digital image processing, and in particular those based on sub-sample analysis [Pennec 2000, Buades 2005, Ngo 2013e, Ngo 2013b]. However, for many applications such

as registration or warping [Pennec 2000, Zitová 2003, Amintoosi 2011], its high-order polynomial complexity raises methodological challenges.

5.2 Perspectives

The research reported in this thesis has mainly led to theoretical results. A first perspective work could consist of using these results for developing digital image processing tools. As stated above, this work is quite close from the research field of registration, that consists of finding admissible transformations between two given digital images. To our knowledge, this problem has not yet been studied from a purely discrete point of view. Indeed, in computer vision, such registration problem is generally formulated as an optimization problem using an objective function defined in a continuous space. Despite – and because of – the infinite nature of rigid transformations, such registration methods generally discretize the parameter space by using regular sampling intervals. In the proposed framework, we already have a finite structure modeling the *whole* space of rigid transformations. It may then be relevant to take advantage of the exhaustion of this discrete framework, and of the efficiency of combinatorial optimisation, in order to progress toward new paradigms of image registration.

In order to apply the proposed structure for digital image applications, it seems however mandatory to reduce its complexity and computational cost. A solution has been investigated in Section 3.3 which consists of providing spatial constraints. Indeed, by forcing some correspondences between pixels, the authorised transformations are restricted to a parameter subspace, and then lead to a sub-structure of the graph with a lower complexity. However, due to the difficulty to satisfy the isometric properties between different constraints, the problem may be ill posed in most cases. To deal with this issue, it may be possible to use the graph with a multi-scale strategy. More precisely, we could consider larger regions of pixels of size $[-\frac{3^n}{2}, \frac{3^n}{2}]^2$ for $n \geq 0$ and a resolution of $\frac{1}{3^n}$. In the parameter space of rigid transformations, this multi-scale strategy introduces a notion of "offset" surfaces, as illustrated in Figure 5.1. The appearance of offset surfaces, when the resolution increases, can lead to develop coarse-to-fine methods that do not require to build the entire structure, but to only focus on its relevant parts.

From a methodological point of view, the concepts developed in this thesis for modeling digital rigid transformations may also be considered for other kinds of transformations, *e.g.*, scalings, affine transformations, *etc.* Indeed, the parameter space subdivision paradigm, such as its modeling by a graph structure, remains valid for any such transformations. Beyond the handling of the spatial complexity induced by other transformations, the main challenge may then consist of finding solutions for computing these new graph structures in an exact fashion. Indeed, the hypotheses of rigid transformations, that allow us to use quadratic irrationals, may be no longer valid for other transformations. A similar issue also appears when considering a further extension of the proposed approach from 2D to 3D rigid transformations.

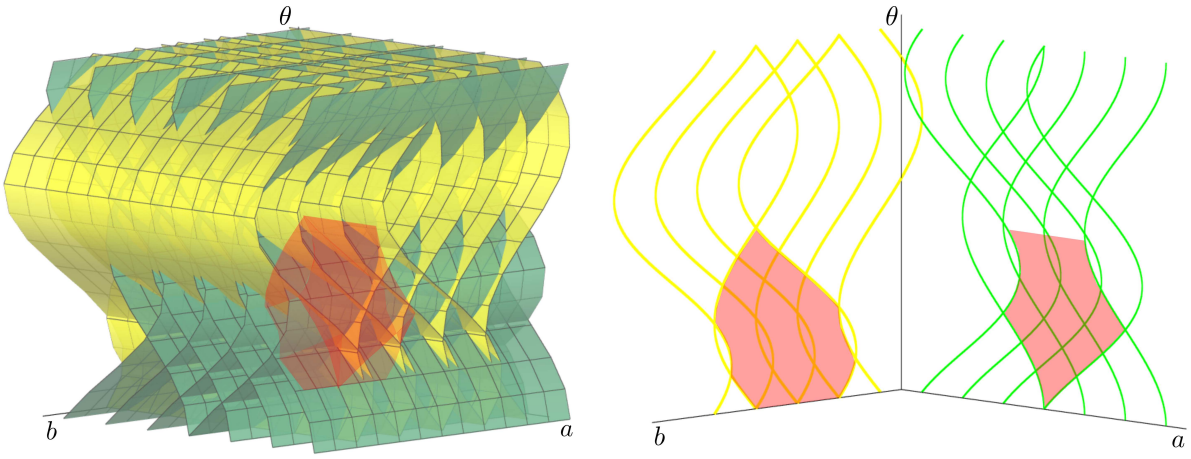


Figure 5.1: Left: the authorised transformations (in red) induced by partial constraints using larger regions of size $[-\frac{3}{2}, \frac{3}{2}]^2$, and expressed *via* "offset" surfaces. Right: their projections to the planes (a, θ) and (b, θ) with the associated "offset" curves.

Going back to the case of 2D rigid transformations, the results regarding the topological invariance of digital images under rigid transformations have been studied only in the case of the Eulerian transformation model. It is then important to seek similar conditions with respect to the Lagrangian transformation model. As discussed in Section 4.2, for the Lagrangian model, beyond the topological alterations caused by digital rigid transformations, additional difficulties related to pixel values are arisen. Typically, double pixels in the transformed space may receive two different values, while null pixels may receive no value (see Figure 4.6). In order to deal with these supplementary issues, it may be necessary to study more deeply the relations that exist between the digital images and the continuous ones, as they are linked via the digitization processes. More precisely, it may be necessary to consider "contour-based" strategies, rather than "pixel-based" ones, in order to handle the ambiguous cases that can not be solved by relevant topological interpolation techniques.

Finally, by emphasising the links between discrete geometry and discrete topology in the context of digital images, this thesis work has recently led us to investigate a new discrete notion of regularity, introduced in [Ngo 2013d]. This work is currently in progress, and it has then been chosen not to include it in this manuscript. This notion of regularity presents similarities with the continuous notion of regularity considered in [Serra 1983, Heijmans 1992, Latecki 1998] for assessing the preservation of topological properties during the digitization of an image from $\mathbb{R}^2$ to $\mathbb{Z}^2$. With this new discrete notion, it is in particular possible to easily and efficiently evaluate the topological invariance of any digital image in linear time with respect to the image size. More details about this notion can be found in [Ngo 2013d, Ngo 2013f].

Publications and communications

Articles in international journals

[Ngo 2013a] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. *Combinatorial structure of rigid transformations in 2D digital images.* Computer Vision and Image Understanding, vol. 117, no. 4, pages 393–408, 2013.

[Ngo 2013b] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. *Topology-preserving conditions for 2D digital images under rigid transformations.* Journal of Mathematical Imaging and Vision, 2013. DOI: 10.1007/s10851-013-0474-z.

Articles submitted in international journals

[Ngo 2013c] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. *On 2D constrained discrete rigid transformations.* HAL 00838184, 2013.

[Ngo 2013d] P. Ngo, N. Passat, Y. Kenmochi and H. Talbot. *Topology-preserving rigid transformation of 2D digital images.* HAL 00795054, 2013.

Articles in international conference proceedings

[Ngo 2012] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. *Combinatorial properties of 2D discrete rigid transformations under pixel-invariance constraints.* In IWCIA, Proceedings, vol. 7655 of *Lecture Notes in Computer Science*, pages 234–248. Springer, 2012.

[Ngo 2013e] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. *Sufficient conditions for topological invariance of 2D digital images under rigid transformations.* In DGCI, Proceedings, volume 7749 of *Lecture Notes in Computer Science*, pages 155–168. Springer, 2013.

[Ngo 2013f] P. Ngo, N. Passat, Y. Kenmochi and H. Talbot. *Well-composed images and rigid transformations.* In ICIP, Proceedings, pages 3035–3039. IEEE, 2013.

[Ngo 2013g] P. Ngo, A. Sugimoto, Y. Kenmochi, N. Passat and H. Talbot. *Discrete rigid transformation graph search for 2D image registration.* In PSIVT 2013 Workshops, Proceedings, volume 8334 of *Lecture Notes in Computer Science*, pages 228–239. Springer, 2014.

Communications

- P. Ngo, Y. Kenmochi and H. Talbot. *Efficient exact computation for incremental discrete rotation using continued fraction.* Demonstration. DGCI. Nancy, March 2011.
- P. Ngo. *Combinatorial structure for rigid transformations in 2D digital images.* Oral presentation. Groupe de travail en géométrie discrète. Clermont-Ferrand, October 2011.

- P. Ngo. *Combinatorial properties of rigid transformations for 2D digital images.* Poster. SubTile. Marseille, January 2013.
- P. Ngo. *Structure combinatoire des transformations rigides sur* $\mathbb{Z}^2$. Seminar. LAMA, Université de Savoie. Chambéry, June 2013.
- P. Ngo. *Préservation topologique des images numériques 2D par transformations rigides.* Oral presentation. Groupe de travail en géométrie discrète. Paris, June 2013.

References

[Amintoosi 2011] M. Amintoosi, M. Fathy and N. Mozayani. *A fast image registration approach based on SIFT key-points applied to super-resolution.* Imaging Science Journal, vol. 60, no. 4, pages 185–201, 2011. (Cited on pages 25 and 42.)

[Amir 1992] A. Amir, G. M. Landau and U. Vishkin. *Efficient pattern matching with scaling.* Journal of Algorithms, vol. 13, no. 1, pages 2–32, 1992. (Cited on pages 4 and 7.)

[Amir 2006] A. Amir, O. Kapah and D. Tsur. *Faster two-dimensional pattern matching with rotations.* Theoretical Computer Science, vol. 368, no. 3, pages 196–204, 2006. (Cited on pages 4, 7 and 17.)

[Amir 2009] A. Amir, A. Butman, M. Lewenstein and E. Porat. *Real two dimensional scaled matching.* Algorithmica, vol. 53, no. 3, pages 314–336, 2009. (Cited on pages 4 and 7.)

[Bazin 2007a] P.-L. Bazin and D. L. Pham. *Statistical and topological atlas based brain image segmentation.* In MICCAI, Proceedings, volume 4791 of *Lecture Notes in Computer Science*, pages 94–101. Springer, 2007. (Cited on page 31.)

[Bazin 2007b] P.-L. Bazin and D. L. Pham. *Topology-preserving tissue classification of magnetic resonance brain images.* IEEE Transactions on Medical Imaging, vol. 26, no. 4, pages 487–496, 2007. (Cited on pages 31 and 40.)

[Buades 2005] A. Buades, B. Coll and J. M. Morel. *A review of image denoising algorithms with a new one.* Multiscale Modeling & Simulation, vol. 4, no. 2, pages 490–530, 2005. (Cited on pages vii, 29 and 41.)

[Chan 2005] T. M. Chan. *On levels in arrangements of surfaces in three dimensions.* In SODA, Proceedings, pages 232–240. ACM-SIAM, 2005. (Cited on pages 20 and 22.)

[Couprie 2001] M. Couprie, F. N. Bezerra and G. Bertrand. *Topological operators for grayscale image processing.* Journal of Electronic Imaging, vol. 10, no. 4, pages 1003–1015, 2001. (Cited on page 39.)

[Couprie 2009] M. Couprie and G. Bertrand. *New characterizations of simple points in 2D, 3D, and 4D discrete spaces.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pages 637–648, 2009. (Cited on pages 31 and 33.)

[Duda 1967] R. O. Duda, P. E. Hart and J. H. Munson. *Graphical data processing research study and experimental investigation.* Research Report AD650926, Stanford Research Institute, 1967. (Cited on page 33.)

[Edelsbrunner 1987] H. Edelsbrunner. Algorithms in Combinatorial Geometry. Springer-Verlag, 1987. (Cited on page 21.)

[Edelsbrunner 1991a] H. Edelsbrunner and L. J. Guibas. *Topologically sweeping an arrangement.* Journal of Computer and System Sciences, vol. 38, no. 1, pages 165–194, 1991. (Cited on pages 20, 21 and 22.)

[Edelsbrunner 1991b] H. Edelsbrunner, R. Seidel and M. Sharir. *On the zone theorem for hyperplane arrangements.* In NRNTCS, Proceedings, volume 555 of *Lecture Notes in Computer Science*, pages 108–123. Springer, 1991. (Cited on page 21.)

[Edelsbrunner 1992] H. Edelsbrunner, L. J. Guibas, J. Pach, R. Pollack, R. Seidel and M. Sharir. *Arrangements of curves in the plane-topology, combinatorics, and algorithms.* Theorical Computer Science, vol. 92, no. 2, pages 319–336, 1992. (Cited on page 21.)

[Faisan 2008] S. Faisan, N. Passat, V. Noblet, R. Chabrier and C. Meyer. *Topology preserving warping of binary images: Application to atlas-based skull segmentation.* In MICCAI, Proceedings, volume 5241, pages 211–218, 2008. (Cited on page 31.)

[Flajolet 1998] P. Flajolet and B. Vallée. *Continued fraction algorithms, functional operators, and structure constants.* Theoretical Computer Science, vol. 194, no. 1–2, pages 1–24, 1998. (Cited on page 24.)

[Halperin 2004] D. Halperin. *Arrangements.* In J. E. Goodman and J. O'Rourke, editeurs, Handbook of Discrete and Computational Geometry, chapitre 24, pages 529–562. CRC Press LLC, 2004. (Cited on page 21.)

[Heijmans 1992] H. J. A. M. Heijmans. *Discretization of morphological operators.* Journal of Visual Communication and Image Representation, vol. 3, no. 2, pages 182–193, 1992. (Cited on pages 40 and 43.)

[Herman 1998] G. T. Herman. Geometry of Digital Spaces. Birkhauser Boston, 1998. (Cited on page 32.)

[Hundt 2007] C. Hundt and M. Liśkiewicz. *On the complexity of affine image matching.* In STACS, Proceedings, volume 4393 of *Lecture Notes in Computer Science*, pages 284–295. Springer, 2007. (Cited on pages 4 and 7.)

[Hundt 2008a] C. Hundt and M. Liśkiewicz. *Combinatorial bounds and algorithmic aspects of image matching under projective transformations.* In MFCS, Proceedings, volume 5162 of *Lecture Notes in Computer Science*, pages 395–406. Springer, 2008. (Cited on pages vi, 4, 7 and 17.)

[Hundt 2008b] C. Hundt and M. Liśkiewicz. *Two-dimensional pattern matching with combined scaling and rotation.* In CPM, Proceedings, volume 5029 of *Lecture Notes in Computer Science*, pages 5–17. Springer, 2008. (Cited on page 7.)

[Hundt 2009] C. Hundt, M. Liśkiewicz and R. Nevries. *A combinatorial geometrical approach to two-dimensional robust pattern matching with scaling and rotation.* Theoretical Computer Science, vol. 410, no. 51, pages 5317–5333, 2009. (Cited on pages vi, 4, 7, 17 and 29.)

[Hundt 2010] C. Hundt. *Affine image matching is uniform TC^0-complete.* In CPM, Proceedings, volume 6129 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 2010. (Cited on pages 4 and 7.)

[Jacob 1995] M.-A. Jacob and E. Andres. *On discrete rotations.* In DGCI, Proceedings, pages 161–174, 1995. (Cited on page 11.)

[Jyoti 2012] J. Jyoti and S. G. Shirish. *Area based image matching methods – A survey.* Emerging Technology and Advanced Engineering, vol. 2, no. 1, pages 130–136, 2012. (Cited on page 25.)

[Khalimsky 1987] E. Khalimsky. *Topological structures in computer science.* Journal of Applied Mathematics and Simulation, vol. 1, no. 1, pages 25–40, 1987. (Cited on pages 32 and 34.)

[Kong 1989] T. Y. Kong and A. Rosenfeld. *Digital topology: Introduction and survey.* Computer Vision Graphics & Image Processing, vol. 48, no. 3, pages 357–393, 1989. (Cited on pages 13, 32, 34 and 38.)

[Kovalevsky 1989] V. A. Kovalevsky. *Finite topology as applied to image analysis.* Computer Vision, Graphics & Image Processing, vol. 46, no. 2, pages 141–161, 1989. (Cited on pages 13, 32 and 34.)

[Lam 1992] L. Lam, S.-W. Lee and C. Y. Suen. *Thinning methodologies – A comprehensive survey.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 9, pages 869–885, 1992. (Cited on pages 31 and 40.)

[Latecki 1995] L. J. Latecki, U. Eckhardt and A. Rosenfeld. *Well-composed sets.* Computer Vision and Image Understanding, vol. 61, no. 1, pages 70–83, 1995. (Cited on page 39.)

[Latecki 1998] L. J. Latecki, C. Conrad and A. Gross. *Preserving topology by a digitization process.* Journal of Mathematical Imaging and Vision, vol. 8, no. 2, pages 131–159, 1998. (Cited on pages 40 and 43.)

[Mazo 2012a] L. Mazo, N. Passat, M. Couprie and C. Ronse. *Digital imaging: A unified topological framework.* Journal of Mathematical Imaging and Vision, vol. 44, no. 1, pages 19–37, 2012. (Cited on page 32.)

[Mazo 2012b] L. Mazo, N. Passat, M. Couprie and C. Ronse. *Topology on digital label images.* Journal of Mathematical Imaging and Vision, vol. 44, no. 3, pages 254–281, 2012. (Cited on page 39.)

[Nouvel 2005] B. Nouvel and E. Rémila. *Configurations induced by discrete rotations: Periodicity and quasi-periodicity properties.* Discrete Applied Mathematics, vol. 147, no. 2–3, pages 325–343, 2005. (Cited on pages vi, 4 and 11.)

[Nouvel 2006a] B. Nouvel. *Rotations discrètes et automates cellulaires.* PhD thesis, École Normale Supérieure de Lyon, 2006. (Cited on pages 4 and 11.)

[Nouvel 2006b] B. Nouvel and E. Rémila. *Incremental and transitive discrete rotations.* In IWCIA, Proceedings, volume 4040 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2006. (Cited on pages vi and 4.)

[Pennec 2000] X. Pennec, N. Ayache and J.-P. Thirion. *Landmark-based registration using features identified through differential geometry.* In I. N. Bankman, editeur, Handbook of Medical Imaging, chapitre 31, pages 499–513. Academic Press, 2000. (Cited on pages vii, 25, 29, 41 and 42.)

[Pham 2001] D. L. Pham. *Spatial models for fuzzy clustering.* Computer Vision and Image Understanding, vol. 84, no. 2, pages 285–297, 2001. (Cited on pages 31 and 40.)

[Rosen 1992] K. H. Rosen. Elementary Number Theory and its Applications. Addison-Wesley, 3rd édition, 1992. (Cited on page 24.)

[Rosenfeld 1966] A. Rosenfeld and J. L. Pfaltz. *Sequential operations in digial picture processing.* Journal of the Association for Computing Machinery, vol. 13, no. 4, pages 471–494, 1966. (Cited on page 32.)

[Rosenfeld 1970] A. Rosenfeld. *Connectivity in digital pictures.* Journal of the Association for Computing Machinery, vol. 17, no. 1, pages 146–160, 1970. (Cited on pages 31 and 33.)

[Serra 1983] J. Serra. Image Analysis and Mathematical Morphology. Academic Press, Inc., 1983. (Cited on pages 40 and 43.)

[Sharir 1999] M. Sharir. *Recent developments in the theory of arrangements of surfaces.* In FSTTCS, Proceedings, volume 1738 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 1999. (Cited on pages 20 and 22.)

[Snoeyink 1989] J. Snoeyink and J. Hershberger. *Sweeping arrangements of curves.* In SOCG, Proceedings, pages 354–363. ACM, 1989. (Cited on page 21.)

[Thibault 2009] Y. Thibault, Y. Kenmochi and A. Sugimoto. *Computing upper and lower bounds of rotation angles from digital images.* Pattern Recognition, vol. 42, no. 8, pages 1708–1717, 2009. (Cited on page 11.)

[Thibault 2010] Y. Thibault. *Rotations in 2D and 3D discrete spaces.* PhD thesis, Université Paris-Est, 2010. (Cited on pages vi, 4, 7, 11 and 17.)

[Tustison 2011] N. J. Tustison, B. B. Avants, M. Siqueira and J. C. Gee. *Topological well-composedness and glamorous glue: A digital gluing algorithm for topologically constrained front propagation.* IEEE Transactions on Image Processing, vol. 20, no. 6, pages 1756–1761, 2011. (Cited on pages 31 and 40.)

[Xiao 2011] C. Xiao, M. Liu, N. Yongwei and Z. Dong. *Fast exact nearest patch matching for patch-based image editing and processing.* IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 8, pages 1122–1134, 2011. (Cited on page 25.)

[Zitová 2003] B. Zitová and J. Flusser. *Image registration methods: A survey.* Image and Vision Computing, vol. 21, no. 11, pages 977–1000, 2003. (Cited on pages 25 and 42.)

Part II

Enclosed articles

Computer Vision and Image Understanding 117 (2013) 393–408

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

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

ARTICLE INFO

Article history:
Received 3 November 2011
Accepted 28 August 2012
Available online 8 December 2012

Keywords:
Digital rigid transformations
Combinatorial structure
Discrete algorithm

ABSTRACT

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

1. Introduction

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

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

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

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

* Corresponding author.
E-mail address: ngo.diemphuc@gmail.com (P. Ngo).

1077-3142/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2012.08.014

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

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

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

2. Geometric transformations and digitization

2.1. 2D images and digitization

Let V be a set called *value space* containing at least two elements, including one, noted $\perp$, corresponding to the "background". A *2D image* is a function $\mathcal{I} : \mathbb{R}^2 \to V$ such that $\mathcal{I}^{-1}(V \setminus \{\perp\})$ is finite. (As examples, if $|V| = 2$, we say that $\mathcal{I}$ is a binary image; if V is equipped with a total order, we say that $\mathcal{I}$ is a grey-level image). Without loss of generality, we can suppose that $\mathcal{I}^{-1}(V \setminus \{\perp\}) \subseteq E = \left]-\frac{1}{2}, N-\frac{1}{2}\right[^2$ for a given $N \in \mathbb{N}$. The set E is called the *support* of $\mathcal{I}$ and N is called the *size* of $\mathcal{I}$.

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

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

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

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

where $[\,\cdot\,]$ is the rounding function. Note in particular that we have $I = \mathcal{I}_{|S}$, that is $I(\boldsymbol{x}) = \mathcal{I}(\boldsymbol{x})$ for any $\boldsymbol{x} \in S$. In other words, we have $D_{|S} = \mathrm{Id}_{|S}$ where Id is the identity function.

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

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

Broadly speaking, the digitization function D decomposes E into unit (open) squares, namely the *pixels*, whose centres are points of $\mathbb{Z}^2$. In other words, the half-grid $\mathcal{H}$ represents the union of the boundaries of all pixels.

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

2.2. Geometric transformations for digital images

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

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

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

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

Remark 1. In the "nearest neighbour" digitization paradigm considered in Section 2.1 (see Eq. (1)), we can consider two models for image transformations. Given a transformation $\mathcal{T}$, the first (the Lagrangian model) consists of determining $\mathcal{T}(\boldsymbol{x})$ for any point x in the initial space, while the second (the Eulerian model) consists of determining $\mathcal{T}^{-1}(\boldsymbol{y})$ for any point 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 models are actually distinct, since $D \circ \mathcal{T}^{-1}$ and $D \circ \mathcal{T}$ are not bijective, in general. In the current work, and without loss of generality, we focus on the Lagrangian model (see Fig. 2). The results shown in this article remain however valid for both models. Still without loss of generality (due to symmetry considerations), we will consider in the sequel T and $D \circ \mathcal{T}$ instead of T^{-1} and $D \circ \mathcal{T}^{-1}$ (see Eq. (3)). Moreover, we focus our study on the effects of rigid transformations on the support of digital images. We do not consider their effects on the actual image values in this article.

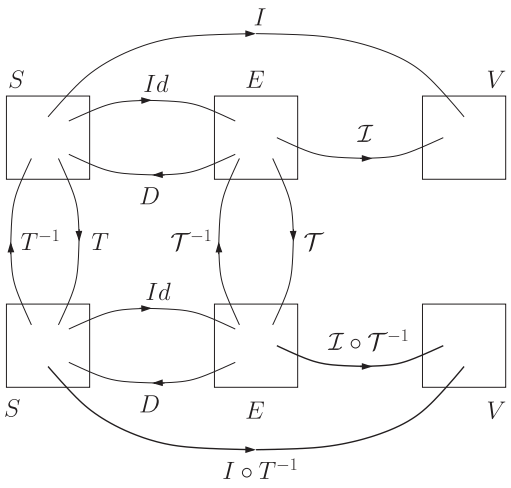
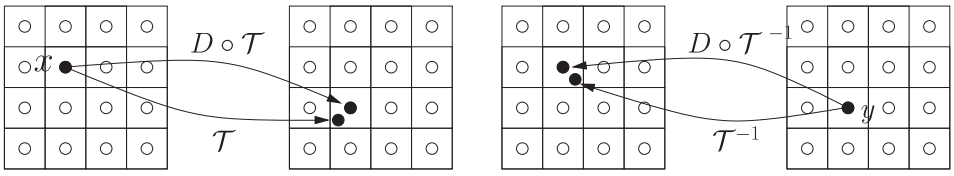


Fig. 1. Geometric transformation and the associated digital image transformation (see text). First row: image support before transformation. Second row: image support after transformation.



(a) Lagrangian model (b) Eulerian model

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

3. Rigid transformations for digital images

3.1. Digital rigid transformations

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

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

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

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

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

This transformation is called a *2D digital rigid transformation*. Note that any rigid transformation (resp. digital rigid transformation) can be represented by a triplet of parameters (a, b, θ), noted $\mathcal{T}_{ab\theta}$ (resp. $T_{ab\theta}$). We note $\mathbb{T}$ (resp. $\mathbb{T}_d$) the set of all rigid transformations (resp. digital rigid transformations):

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

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

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

3.2. Discontinuities of digital rigid transformations

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

$$\left| \begin{array}{rccl} \mathfrak{T} : & \mathbb{R}^3 & \to & \mathbb{T} \\ & (a, b, \theta) & \mapsto & \mathfrak{T}(a, b, \theta) = \mathcal{T}_{ab\theta}, \end{array} \right. \tag{8}$$

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

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

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

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

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

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

Such points $\boldsymbol{c} \in \mathbb{R}^3$ are critical points in the parameter space of the digital rigid transforms. They characterise, in particular, some transformations which map at least one point onto the half grid (see Fig. 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

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

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

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

3.3. Discrete rigid transformations

As a result of the discontinuity of digital rigid transformations, it is possible for the digitization D of two rigid transformations of a digital image to be identical. This leads to the following consideration about the equivalence between transformations.

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

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

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

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

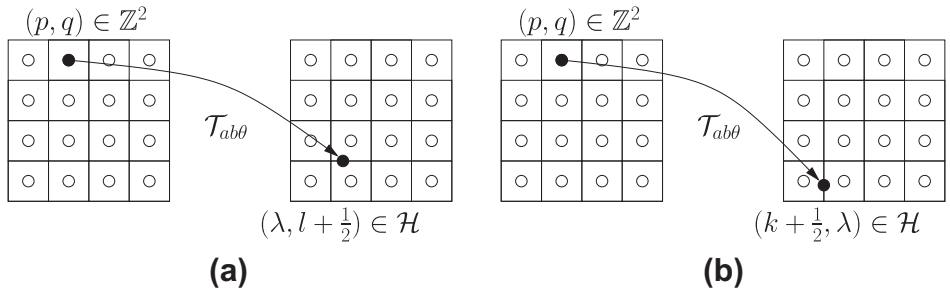


Fig. 3. Examples of critical transformations $\mathcal{T}_{ab\theta}$, which map at least one integer value onto a "horizontal" (a) or "vertical" (b) half-grid point. The digital support $S \subset \mathbb{Z}^2$ is depicted by white dots. The Voronoi cells of E associated to the points of S are depicted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

4. Tipping surfaces and tipping curves

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

4.1. Tipping surfaces

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

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

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

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

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

4.2. Tipping curves

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

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

$$\left| \begin{array}{rccl} \phi_{pqk}: & \mathbb{R} & \to & \mathbb{R}, \\ & \theta & \mapsto & a = k+\frac{1}{2}+q\sin\theta - p\cos\theta, \end{array} \right. \tag{15}$$

$$\left| \begin{array}{rccl} \psi_{pql}: & \mathbb{R} & \to & \mathbb{R}, \\ & \theta & \mapsto & b = l+\frac{1}{2}-p\sin\theta - q\cos\theta. \end{array} \right. \tag{16}$$

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

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

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

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

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

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

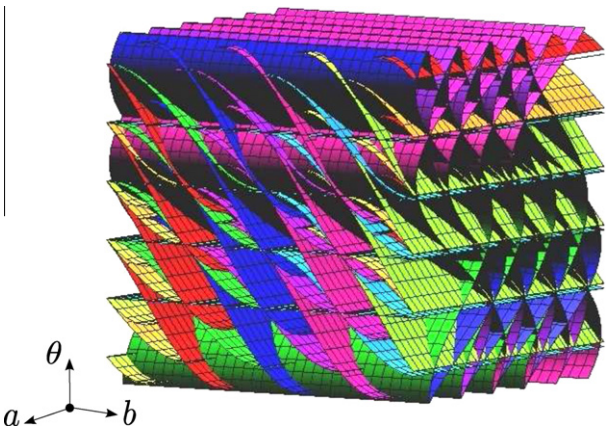


Fig. 4. Discrete rigid transformations (DRTs) are observed as regions separated by tipping surfaces.

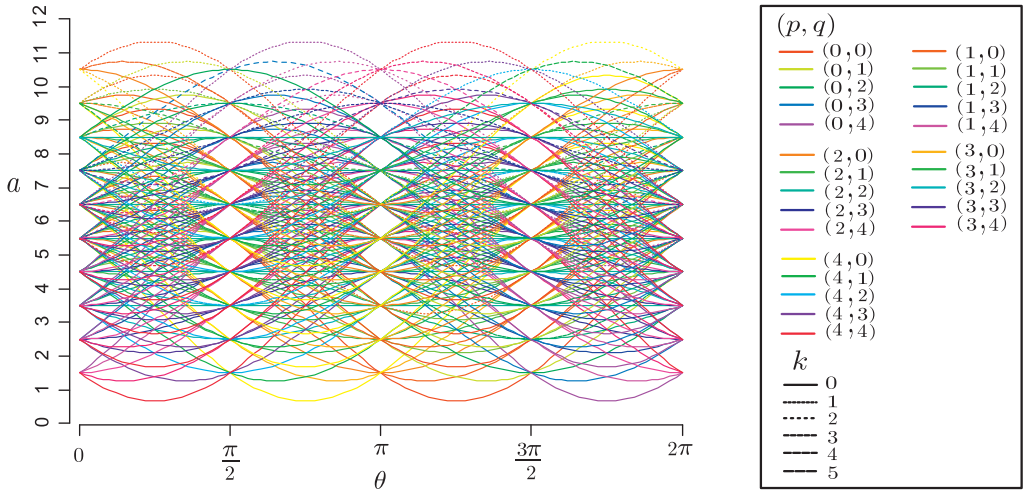


Fig. 5. A part of the family of tipping curves F_ϕ.

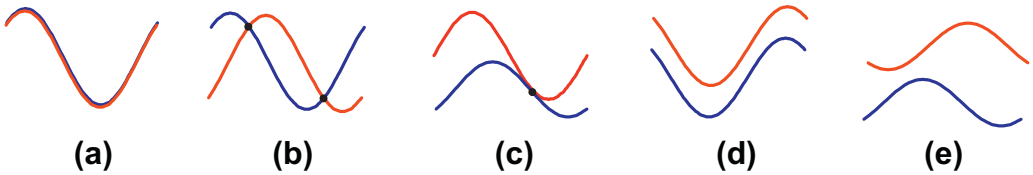


Fig. 6. Relationships between two tipping curves, which are identical (a), intersecting (b), tangent (c) and not intersecting (d and e).

4.3. Properties of tipping curves

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

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

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

The next property shows the relationships between two tipping curves. This will be used in the algorithm of construction of the combinatorial structure of discrete rigid transformations. Geometrically, two tipping curves can relate to each other in four different ways (see Fig. 6): they can be identical, intersecting,[1] tangent[2] or not intersecting. Analytical interpretation of these relationships is expressed as follows.

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

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

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

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

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

$$K - P\cos\theta = -Q\sin\theta. \tag{19}$$

Squaring both sides of Eq. (19), we obtain

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

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

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

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

which are quadratic equations if $P^2 + Q^2 \neq 0$. As we assume that ϕ_{pqk} and $\phi_{p'q'k'}$ are different, either P, Q or K are not equal to 0. If $P = Q = 0$ and $K \neq 0$, then Eq. (19) has no solution (ii). Otherwise, $P^2 + Q^2 \neq 0$ so that the discriminants Δ_1 and Δ_2 of Eqs. (20) and

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

[2] 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.

(21) determine the number and nature of the roots. There are three cases; the curves are:

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

Given two curves ϕ_{pqk} and $\phi_{p'q'k'}$, we can find out their relationship by evaluating the integer triplets (p,q,k) and (p',q',k') exactly, i.e., by using only integers during computation. Indeed, the values P, Q, K, Δ_1, Δ_2 and Δ are all integers, while the inequalities $|KP \pm \sqrt{\Delta_1}| \leqslant P^2 + Q^2$ and $|KQ \pm \sqrt{\Delta_2}| \leqslant P^2 + Q^2$ can be verified by using squaring procedures. The proof of Property 2 implies the following corollary.

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

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

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

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

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

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

$$F_\phi(\theta) = -F_\phi(\theta),$$

$$F_\phi(\theta) = F_\phi\left(-\theta + m\frac{\pi}{4}\right), \quad \forall m \in \mathbb{Z}.$$

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

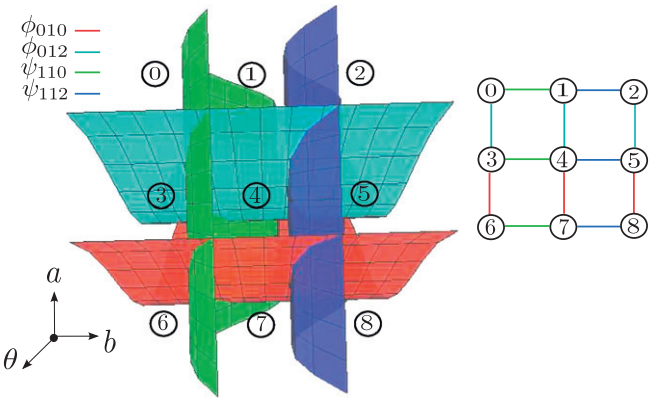


Fig. 7. Four tipping surfaces in the parameter space (a,b,θ) (left) and its DRT graph (right).

$$F_\phi(\theta) = F_\phi\left(\theta + m\frac{\pi}{2}\right), \quad \forall m \in \mathbb{Z},$$

$$F_\phi(\theta) = F_\phi(\theta) + m, \quad \forall m \in \mathbb{Z}.$$

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

5. Graph representation of discrete rigid transformations

As discussed in the previous section, the parameter space of rigid transformations is divided into DRTs, whose boundaries are the tipping surfaces. This subdivision can be represented by a graph which is defined as follows.

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

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

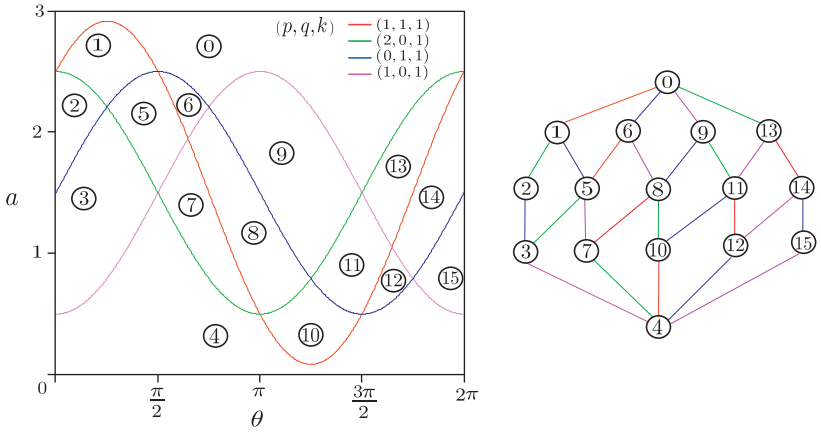


Fig. 8. Four tipping curves on the plane (a,θ) (left) and the associated 2D DRT graph (right).

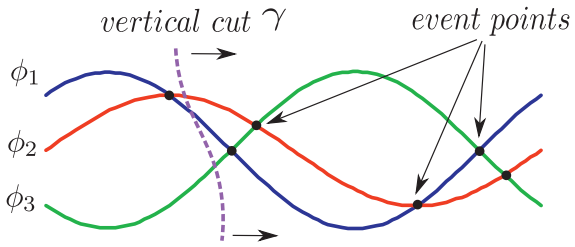


Fig. 9. Event points and a vertical cut $\gamma = (\phi_2, \phi_1, \phi_3)$.

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

In order to distinguish between the integer triplets of vertical and horizontal tipping surfaces, we use an integer quadruple (p, q, k, i) for modelling each tipping surface, where $i = 0$ or 1 indicates respectively the vertical or horizontal set.

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

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

6. Construction of 2D discrete rigid transformation graph

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

6.1. Problem formalisation

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

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

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

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

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

6.2. Principles of incremental 2D DRT graph reconstruction

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

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

Definition 9. Let $\gamma = (\phi_1, \phi_2, \ldots, \phi_n)$ 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

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

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

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

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

$$G^C = \bigcup_{i=0}^{m} \delta G_i^C, \tag{22}$$

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

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

6.3. Initial graph construction

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

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

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

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

6.4. Incremental 2D DRT graph construction for simple cases

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

6.4.1. Detecting and ordering event points

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

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

Definition 11. For any pair of event points $\boldsymbol{u} = (u_x, u_y)$ and $\boldsymbol{v} = (u_x, u_y)$ 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 performed with exact computation. We know that the coordinates (θ, a) of event points are typically irrational numbers and that we generally cannot compute an exact value for θ. Nevertheless, in order to sort event points with respect to $\theta \in [0, 2\pi[$, we can use the values cos θ and sin θ which are calculated from Corollary 1. We can also easily obtain the value a from Formula (15) with cos θ and sin θ. Note that a, cos θ and sin θ are all quadratic irrationals.[3] Therefore, sorting event points then relies on the capacity to compare quadratic irrationals. It is known in [19] that two quadratic irrationals can be compared by an exact method. In fact, a quadratic irrational can be represented exactly using a periodic continued fraction modelled by a sequence of integers, and this representation is unique. Moreover the comparison of periodic continued fractions can be performed in constant time [20] (see Appendix A for more details), so that sorting all event points requires $\mathcal{O}(|\mathcal{Q}| \log |\mathcal{Q}|)$ times of such a comparison, where $|\mathcal{Q}|$ is the event queue size.

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

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

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 Fig. 11. Following a change in γ, the partial graph δG^C is modified.

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

1. finding the current vertex w bounded by ϕ_u and ϕ_v;
2. deleting two edges that are adjacent to w;
3. replacing w by a new vertex w';
4. creating two new edges that are linked to w'.

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

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

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 the subtracting part from δG_i^C and $\Delta G_+^C = (\Delta V_+^C, \Delta E_+^C)$ is the adding part to δG_i^C */

6.4.3. Algorithm

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

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

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

Initialize G^C and δG^C; (see Section 6.3)
Generate an event queue $\mathcal{Q}$; (see Section 6.4.1)
while $\mathcal{Q} \neq \emptyset$ **do**
  $\mathbf{q} \leftarrow dequeue(\mathcal{Q})$;
  $\delta G^C \leftarrow$ Procedure 1 $(\delta G^C, \mathbf{q})$;
  $G^C \leftarrow G^C \cup \delta G^C$;

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

6.5. Incremental 2D DRT graph construction for degenerate cases

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

6.5.1. Detecting and sorting event points

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

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

In fact, those degeneracies can be detected by using Property 2 (iii), (iv), and Corollary 1 as mentioned in Section 6.4.1. More precisely, comparing the intersections allows us to detect degenerate event points since they have the same coordinates[4] at the point. Such a degenerate event point is now represented by a family of tipping curves that go through the intersection point (see Fig. 13).

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

At each event point, the elementary step consists of modifying the partial graph δG^C according to the change of γ in a similar way to Procedure 1. Note that in degenerate cases, each event point contains a family of tipping curves instead of a pair as in simple cases. Let $\mathbf{q}$ be an event point, then $\mathbf{q} = \{\tau_0, \tau_1, \ldots, \tau_m\}$ is a family of tipping curves where each $\tau_j = \{\phi_{j_1}, \phi_{j_2}, \ldots\}$ is a set of tipping curves with the same tangent at $\mathbf{q}$. Let δG_i^C and δG_{i+1}^C be respectively the partial graphs with respect to γ_i and γ_{i+1}, which go through on the left and right of the event point $\mathbf{q}$. The construction of δG_{i+1}^C from δG_i^C proceeds following these steps:

1. Generating two lists of tipping curves which give the orders of tipping curves before and after $\mathbf{q}$ and storing them into two Last-In First-Out (LIFO) stacks $\mathbf{S_1}$ and $\mathbf{S_2}$ respectively. Details are given below.
2. Finding the initial and terminal vertices u and v for an event point $\mathbf{q}$, between which δG_i^C changes.
3. Finding the current vertices and edges between u and v.
4. Replacing the current vertices and edges by the new vertices and edges between u and v.

While creating new edges in the last step, each edge is given a label of a tipping curve taken from $\mathbf{S_2}$.

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

We now explain how to fill two LIFO stacks $\mathbf{S_1}$ and $\mathbf{S_2}$ according to the orders of tipping curves before and after the event point $\mathbf{q}$. For this, we first need to sort the sets τ_j and the tipping curves ϕ in each set τ_j of $\mathbf{q}$ with the order obtained from δE_i^C. After sorting, we assume that $\mathbf{q} = (\tau_0, \tau_1, \ldots, \tau_m)$ and each $\tau_j = (\phi_{j_1}, \phi_{j_2}, \ldots)$ with respect to δE_i^C. The stacks $\mathbf{S_1}$ and $\mathbf{S_2}$ are generated by using the reverse sequences of τ_j, which is $\bar{\tau}_j = (\ldots, \phi_{j_2}, \phi_{j_1})$, as follows.

```
for j = 0 → m do
    for each φ ∈ τ̄_{m−j} do
      S1 ← push(φ);
    end for
    for each φ ∈ τ̄_j do
      S2 ← push(φ);
    end for
end for
```

Note that in $\mathbf{S_1}$, tipping curves have the same order as in δE_i^C, and in $\mathbf{S_2}$ the order of sets of tipping curves are reversed while the order of curves in each set is preserved. Procedure 2 generates δG_{i+1}^C from δG_i^C at the event point $\mathbf{q}$.

Procedure 2. Elementary step for degenerate cases.

Input: A partial graph δG_i^C and an event point $\mathbf{q}$.
Output: A partial graph δG_{i+1}^C.

1 Fill $\mathbf{S_1}$ and $\mathbf{S_2}$ as explained above.
2 $\phi_u \leftarrow pop(\mathbf{S_1})$; $\phi_v \leftarrow pop(\mathbf{S_1})$;
3 $e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$;
4 $\{u\} \leftarrow \vartheta(e_u) \setminus (\vartheta(e_u) \cap \vartheta(e_v))$; // u is the initial vertex
5 $\mathbf{S_1} \leftarrow push(\phi_v)$; $\mathbf{S_1} \leftarrow push(\phi_u)$;
6 $\phi_u \leftarrow pop(\mathbf{S_2})$; $\phi_v \leftarrow pop(\mathbf{S_2})$;
7 $e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$;
8 $\{v\} \leftarrow \vartheta(e_u) \setminus (\vartheta(e_v) \cap \vartheta(e_v))$; // v is the terminal vertex
9 $\mathbf{S_2} \leftarrow push(\phi_v)$; $\mathbf{S_2} \leftarrow push(\phi_u)$;
10 **while** $\mathbf{S_1} \neq \emptyset$ **do**
11 $\phi_u \leftarrow pop(\mathbf{S_1})$; $\phi_v \leftarrow pop(\mathbf{S_1})$;
12 $e_u \leftarrow \varepsilon(\phi_u)$; $e_v \leftarrow \varepsilon(\phi_v)$;
13 $\{w\} \leftarrow \vartheta(e_u) \cap \vartheta(e_v)$;
14 $\{w_u\} \leftarrow \vartheta(e_u) \setminus \{w\}$;
15 $\Delta V_-^C \leftarrow \Delta V_-^C \cup \{w\}$; // w is a removed vertex
16 $\Delta E_-^C \leftarrow \Delta E_-^C \cup \{(w_u, w, \phi_u)\}$;
17 **if** $\mathbf{S_1} \neq \emptyset$ **then** $\mathbf{S_1} \leftarrow push(\phi_v)$;
18 $j \leftarrow 0$;
19 $\phi \leftarrow pop(\mathbf{S_2})$;
20 $\Delta V_+^C \leftarrow \{w_j\}$; // w_j is a new vertex
21 $\Delta E_+^C \leftarrow \{(u, w_j, \phi)\}$;
22 **while** $\mathbf{S_2} \neq \emptyset$ **do**
23 $j \leftarrow j + 1$;
24 $\phi \leftarrow pop(\mathbf{S_2})$;
25 $\Delta V_+^C \leftarrow \Delta V_+^C \cup \{w_j\}$; // w_j is a new vertex
26 **if** $\mathbf{S_2} \neq \emptyset$ **then** $\Delta E_+^C \leftarrow \Delta E_+^C \cup \{(w_{j-1}, w_j, \phi)\}$;
27 **else** $\Delta E_+^C \leftarrow \Delta E_+^C \cup \{(w_j, v, \phi)\}$;
28 $\delta G_{i+1}^C \leftarrow \delta G_i^C \setminus \Delta G_-^C \cup \Delta G_+^C$;

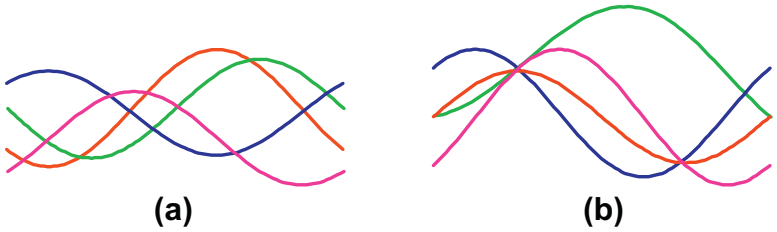


Fig. 10. Illustration for simple (a) and non-simple (b) cases.

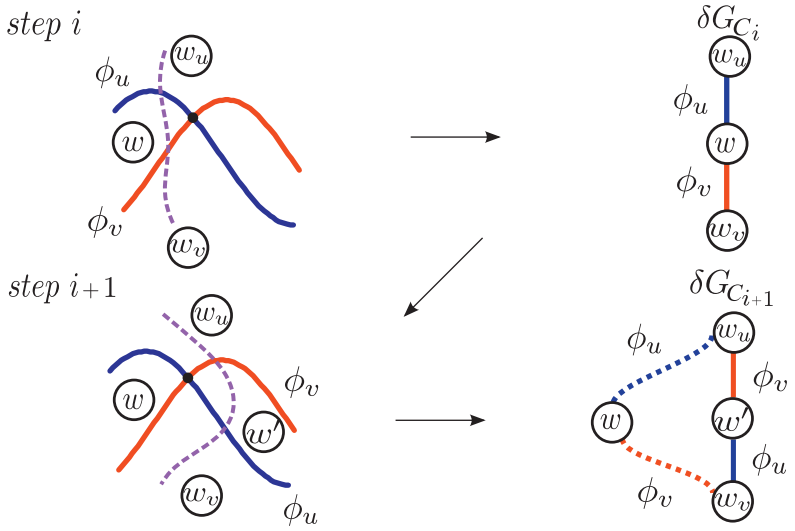


Fig. 11. Illustration of a progress of a cut by which the partial graph δG^C_{i+1} is modified from δG^C_i in simple case.

7. Construction of a discrete rigid transformation graph

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

7.1. Principles of incremental DRT graph construction

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

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

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

$$b\left(\theta+\frac{\pi}{2}\right) = l+\frac{1}{2}-p\sin\left(\theta+\frac{\pi}{2}\right)-q\cos\left(\theta+\frac{\pi}{2}\right) = l+\frac{1}{2}-p\cos\theta+q\sin\theta = a(\theta).$$

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

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

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

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

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

7.2. Initial graph construction

The initial DRT graph $G_0 = (V_0, E_0)$ is generated from $\delta G^a_0 = (\delta V^a_0, \delta E^a_0)$ and $\delta G^b_0 = \left(\delta V^b_0, \delta E^b_0\right)$ as follows:

- $V_0 = \left\{(v_a, v_b) \mid v_a \in \delta V^a_0, v_b \in \delta V^b_0\right\}$;
- $E_0 = \left\{((u_1, v), (u_2, v), \phi_u) | u_1, u_2 \in \delta V^a_0, v \in \delta V^b_0, (u_1, u_2, \phi_u) \in \delta E^a_0\right\}$
  $\cup \left\{((u, v_1), (u, v_2), \phi_v) | v_1, v_2 \in \delta V^b_0, u \in \delta V^a_0, (v_1, v_2, \phi_v) \in \delta E^b_0\right\}$.

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

7.3. Elementary step

The additional part ΔG_i at each elementary step is constructed by Procedure 3, from δG_i^a and δG_i^b in a similar way to the initialization. The difference being that a new vertex is generated from a pair of a new vertex in δV_i^a and one of the vertices in δV_i^b or vice versa. Thus a new edge connects two vertices sharing an edge in δE_i^a and δE_i^b. Procedure 3 requires the additional parts ΔV_+^a and ΔV_+^b of δG_i^a and δG_i^b respectively. We modify Procedure 2 and obtain Procedure 2', such that it returns not only a partial graph δG_{i+1}^C but also an adding vertex set ΔV_+^C, which is already calculated in Procedure 2 as an intermediate parameter.

Procedure 3. Elementary step for DRT graph construction.

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

Initialize $\Delta G_i = \emptyset$, with $\Delta V_i = \emptyset$ and $\Delta E_i = \emptyset$;
foreach $u \in \Delta V_+^a$ *and* $v \in \delta V_i^b$ **do**
  $\Delta V_i \leftarrow \Delta V_i \cup \{(u, v)\}$;
foreach $u \in \delta V_i^a$ *and* $v \in \Delta V_+^b$ **do**
  $\Delta V_i \leftarrow \Delta V_i \cup \{(u, v)\}$;
foreach $e_u = (u_1, u_2, \phi_u) \in \delta E_i^a$ **do**
  foreach $v \in \Delta V_+^b$ **do**
    $\Delta E_i \leftarrow \Delta E_i \cup \{((u_1, v), (u_2, v), \phi_u)\}$;
foreach $e_v = (v_1, v_2, \phi_v) \in \delta E_i^b$ **do**
  foreach $u \in \Delta V_+^a$ **do**
    $\Delta E_i \leftarrow \Delta E_i \cup \{((u, v_1), (u, v_2), \phi_v)\}$;

7.4. Algorithm

The final algorithm (Algorithm 2) builds a DRT graph G by taking two event points $\mathbf{q}_a \in \mathcal{Q}_a$, $\mathbf{q}_b \in \mathcal{Q}_b$, whose coordinates are identical. At each iteration, we generate ΔG simply by using Procedure 3 and then 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; (see Section 6.3)
Initialize $G = G_0$ from δG_0^a and δG_0^b; (see Section 7.2)
Generate $\mathcal{Q}_a$ and $\mathcal{Q}_b$; (see Section 6.5.1)
while $\mathcal{Q}_a \neq \emptyset$ *and* $\mathcal{Q}_b \neq \emptyset$ **do**
  $\mathbf{q}_a \leftarrow dequeue(\mathcal{Q}_a)$;
  $\{\delta G^a, \Delta G_+^a\} \leftarrow$ Procedure 2' $(\delta G^a, \mathbf{q}_a)$;
  $\mathbf{q}_b \leftarrow dequeue(\mathcal{Q}_b)$;
  $\{\delta G^b, \Delta G_+^b\} \leftarrow$ Procedure 2' $(\delta G^b, \mathbf{q}_b)$;
  $\Delta G \leftarrow$ Procedure 3 $(\delta G^a, \delta G^b, \Delta G_+^a, \Delta G_+^b)$;
  $G \leftarrow G \cup \Delta G$;

8. Complexity analysis and experiments

8.1. Space complexity of DRT graph

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

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

(i) *the number of event points is at most n(n − 1); and the generated 2D DRT graph G^C has*
(ii) *at most $n^2 + 1$ vertices;*
(iii) *at most $2n^2 - n$ edges.*

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

The following property allows us to study the complexity of F_ϕ for a digital image I with finite size. This property induces the fact that the number of possible DRTs is finite and bounded by the size of support S of I. We say two tipping curves are *vertically offset* if one can be obtained by translating the other vertically. In fact, the case of two vertically offset curves is a special case of two tipping curves with no intersection, which corresponds to (ii) in Property 2.

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

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

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

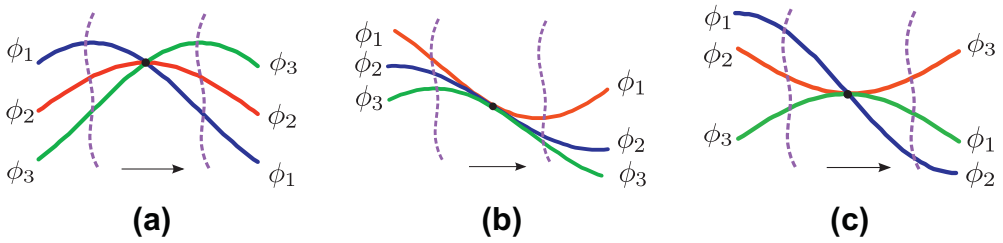


Fig. 12. Degenerate cases: multiple intersection (a), multiple tangent point (b) and multiple mixed point (c).

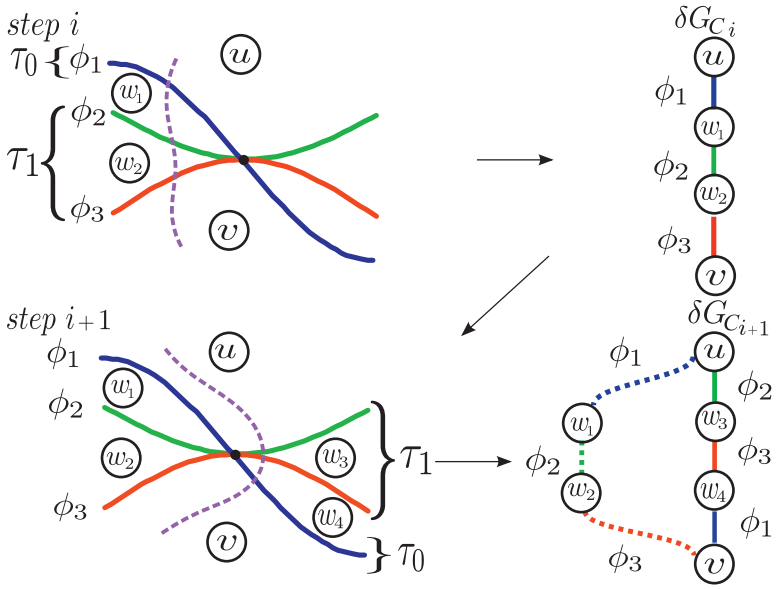


Fig. 13. Illustrating the progress of a cut by which the partial graph δG^c_{i+1} is modified from δG^c_i in a degenerate case.

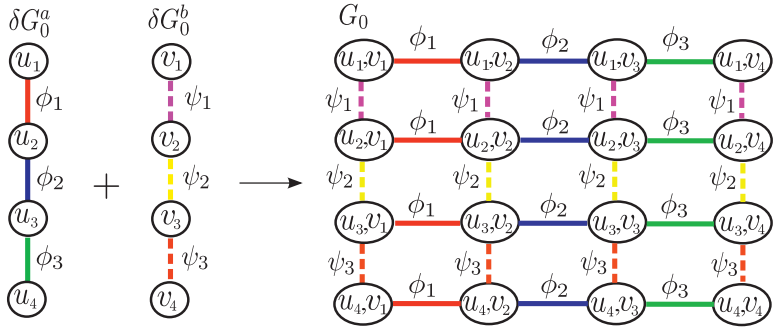


Fig. 14. Illustration of construction of an initial DRT graph G_0 from two 2D DRT partial graphs δG^a_0 and δG^b_0 on the planes (a,θ) and (b,θ).

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

As mentioned in Section 7, the construction of a DRT graph G is obtained from its projections on the planes (a,θ) and (b,θ) using two cuts. Then we notice that the initial graph has a complexity $\mathcal{O}(N^3) \times \mathcal{O}(N^3)$. We also know that at each elementary step, there are $\mathcal{O}(N^3)$ vertices generated. As the number of event points is $\mathcal{O}(N^6)$, in total there are $\mathcal{O}(N^6) \times \mathcal{O}(N^3)$ vertices added in G. This justifies the following theorem.

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

N	2D DRT graph		DRT graph	
	Vertices	Edges	Vertices	Edges
1	1	0	1	0
2	49	144	1033	5040
3	431	1472	29,631	160,512
4	2277	8144	357,421	1,993,696
5	8371	30,304	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

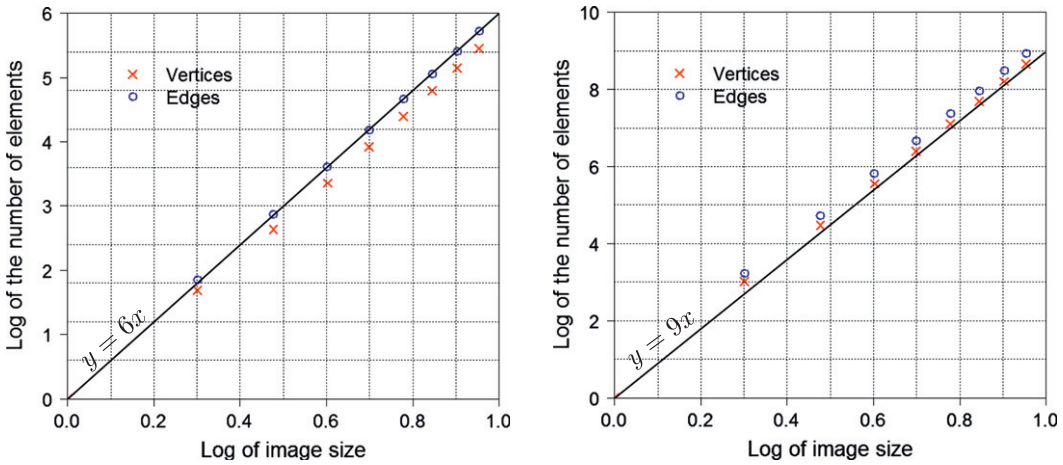


Fig. 15. The relation between image size and number of elements, (i.e., vertices or edges) in Table 2 for 2D DRT graphs (left) and DRT graphs (right).

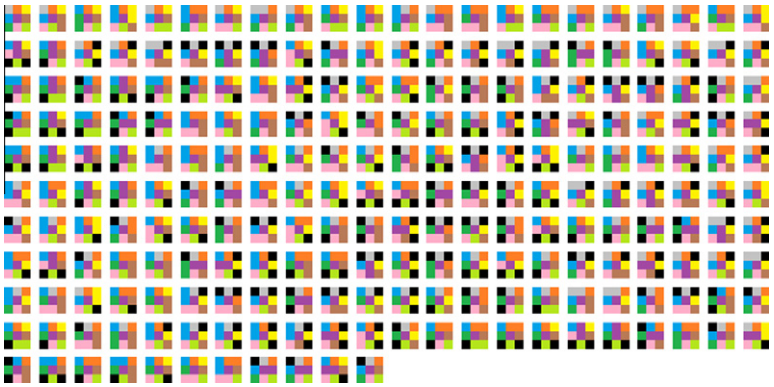


Fig. 16. The 231 images are generated by using the Eulerian model from the part of a DRT graph for a given image of size 3×3 where $a, b \in]\frac{1}{2}, \frac{3}{2}[$ and $\theta \in]0, \frac{\pi}{4}[$. The first image is the identity transformation. In the transformed images, we do not show pixels that are pushed out of the image support. Also, formerly background pixels are "filled" with a black colour.

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

8.2. Run-time complexity of construction algorithm

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

8.3. Experiments

We have implemented our algorithm in C++. From the experiments, the numbers of vertices (and edges) of the DRT graphs have been computed for images of sizes varying from 1×1 to 9×9. The experiments were carried out on a personal computer equipped with a processor 3.0 GHz Intel® Core™ 2 Duo and 4 GB of memory. The results, shown in Table 2 and Fig. 15, validate the theoretical $\mathcal{O}(N^9)$ space complexity stated in the previous theorem.

9. Applications

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

9.1. Generation of all digital transformed images

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

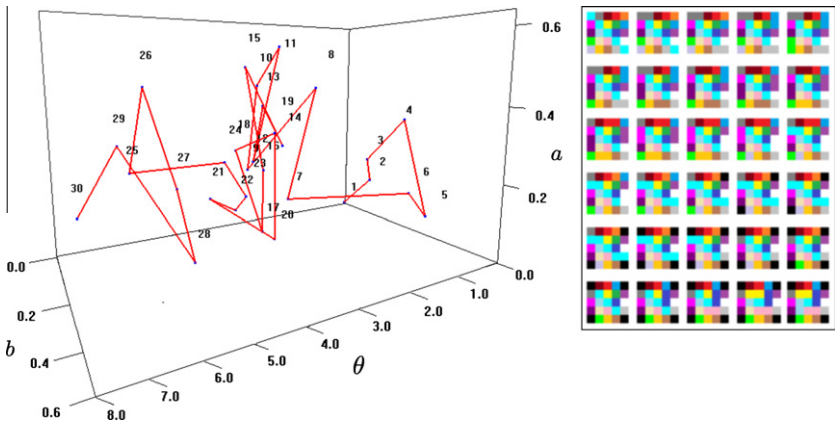


Fig. 17. A discrete transition path of the DRT graph in the parameter space (a,b,θ) (left) and a sequence of transformed images (right). Each vertex of the path represents a DRT which gives a digital transformed image. The first image corresponds to the identity transformation and the images from left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

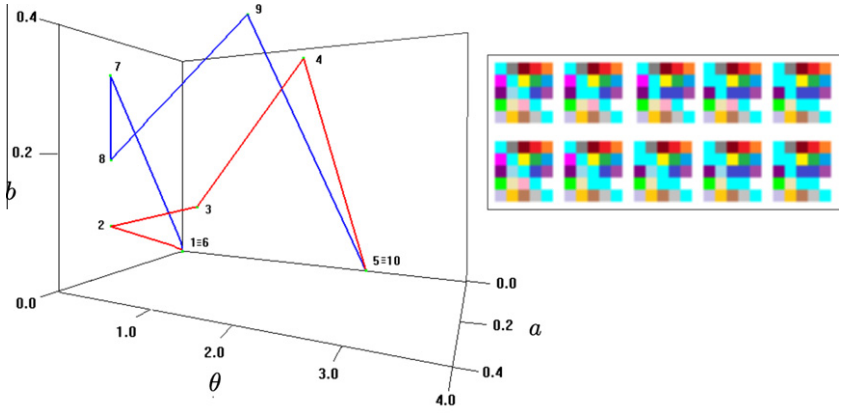


Fig. 18. Illustration of two different discrete transition paths with the same starting and ending vertices in a DRT graph: the two paths (left) and their sequences of images (right). The first row images correspond to the red path and the bottom ones correspond to the blue path. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

all possible images under any rigid transformations of I. Fig. 16 shows all generated digital transformed images by using the Eulerian model[5] from the part of the DRT graph for an image of size 3×3 where $a, b \in \left]\frac{1}{2}, \frac{3}{2}\right[$ and $\theta \in \left]0, \frac{\pi}{4}\right[$.

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

Given an image I and its transformed image I', we define a *transition path* between I and I' as the (possibly non-unique) locus of rigid transformations transitioning I to I' in the parameter space (a,b,θ). By using the DRT framework, such a transition path can be represented as a connected path in the DRT graph G, called a *discrete transition path*. Indeed, we can find a discrete transition path in G between I and I' using the topological information of G providing the neighbouring relations of DRTs. Fig. 17 shows a discrete transition path between two vertices of G and the sequence of transformed images corresponding to the vertices through the path. This representation of such a sequence illustrates the fact that only one pixel is changed between two incident transformed images. This allows us to define a metric on the DRT graph for measuring similarity between different transformed images, simply by counting the number of pixel changes, i.e., the length of the path.

[5] The Eulerian model of a given rigid transformation $\mathcal{T}$ consists of determining $\mathcal{T}^{-1}(\boldsymbol{y})$ for any point $\boldsymbol{y} \in \mathbb{Z}^2$ in the deformed space (see Remark 1 in Section 2.2).

9.3. Evaluation of discrete transition paths

Given two DRT graph vertices, it is observed in Fig. 18 that their discrete transition paths are not unique. Assuming the existence of several different transition paths, we would like to find the best path which preserves some additional criteria such as topology and geometry of our object of interest in images. We know that for any two consecutive images in a discrete transition path only one pixel changes and thus it may be easy to verify our criteria incrementally along the path by checking only this pixel. By using the DRT framework, it may be possible to propose a new evaluation strategy for discrete transition paths.

10. Conclusion

In this article, we have introduced a combinatorial structure represented as a graph for modelling the parameter space of digital rigid transformations. This graph consists of finite sets of vertices and edges. In this graph, each vertex represents a digital transformed image, and each edge linking two vertices represents a transition changing only one pixel between two transformed images. This structure presents a space complexity of $\mathcal{O}(N^9)$, where $N \times N$ is the size of any considered subspace of $\mathbb{Z}^2$. An algorithm has also been proposed in order to define this structure in linear time with respect to this space complexity.

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

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

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

Acknowledgments

The research leading to these results has received funding from the French *Agence Nationale de la Recherche* (Grant Agreement ANR-2010-BLAN-0205 03). The authors thank the anonymous reviewers for their helpful comments to improve the manuscript.

Appendix A. Exact comparison of quadratic irrationals

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

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

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

Given $Q_1 = \frac{p_1+\sqrt{q_1}}{r_1} = [a_0; a_1, a_2, \ldots]$ and $Q_2 = \frac{p_2+\sqrt{q_2}}{r_2} = [b_0; b_1, b_2, \ldots]$, let $k \in \mathbb{N}$ be the smallest index for which $a_k \neq b_k$. If $Q_1 \neq Q_2$ (the equality can be easily checked by comparing the values p_i, q_i and r_i), the order between Q_1 and Q_2 is characterised by the sign of the value $E = (-1)^k(a_k - b_k)$. In particular, we have $Q_1 < Q_2$ (resp. $Q_1 > Q_2$) if $E < 0$ (resp. $E > 0$) (see Algorithm 3). Note that for a quadratic irrational $Q = \frac{p+\sqrt{q}}{r}$, it is proved in [25] that the length of repeating block for Q is $\mathcal{O}(\sqrt{q} \ln q)$. In the worst case, all terms of the repeating block are compared, i.e., we have $\mathcal{O}(\sqrt{q} \ln q)$ comparisons. However, it is proved in [20] that the comparison of continued fractions has an average-case complexity of $\mathcal{O}(1)$.

Algorithm 3. Comparison of two quadratic irrationals

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

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

if $(p_1, q_1, r_1) = (p_2, q_2, r_2)$ **then**
  return $=$;
else
  $E \leftarrow 0$;
  $(p_1^0, q_1^0, r_1^0, a_1^0) \leftarrow (p_1, q_1, r_1, \lfloor \frac{p_1+\sqrt{q_1}}{r_1} \rfloor)$;
  $(p_2^0, q_2^0, r_2^0, a_2^0) \leftarrow (p_2, q_2, r_2, \lfloor \frac{p_2+\sqrt{q_2}}{r_2} \rfloor)$;
  $i \leftarrow 1$;
  while $E = 0$ **do**
    // calculate the term a_1^i of Q_1
    $(p_1^i, r_1^i, a_1^i) \leftarrow (a_1^{i-1} r_1^{i-1} - p_1^{i-1}, \frac{q_1 - (p_1^i)^2}{r_1^{i-1}}, \lfloor \frac{p_1^i + \sqrt{q_1}}{r_1^i} \rfloor)$;
    // calculate the term a_2^i of Q_2
    $(p_2^i, r_2^i, a_2^i) \leftarrow (a_2^{i-1} r_2^{i-1} - p_2^{i-1}, \frac{q_2 - (p_2^i)^2}{r_2^{i-1}}, \lfloor \frac{p_2^i + \sqrt{q_2}}{r_2^i} \rfloor)$;
    $E \leftarrow (-1)^{i-1}(a_1^i - a_2^i)$;
    $i \leftarrow i + 1$;
  if $E > 0$ **then** return $>$;
  else return $<$;

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38 (4) (2006) 1–45.
[2] B. Zitová, J. Flusser, Image registration methods: a survey, Image and Vision Computing 21 (11) (2003) 977–1000.
[3] E. Andres, The quasi-shear rotation, in: Discrete Geometry for Computer Imagery, DGCI, Proceedings, Lecture Notes in Computer Science, vol. 1176, Springer, 1996, pp. 307–314.
[4] D. Coeurjolly, V. Blot, M.-A. Jacob-Da Col, Quasi-affine transformation in 3-D: theory and algorithms, in: International Workshop on Combinatorial Image Analysis, IWCIA, Proceedings, Lecture Notes in Computer Science, Springer, 2009, pp. 68–81.
[5] C. Hundt, M. Liśkiewicz, N. Ragnar, A combinatorial geometrical approach to two-dimensional robust pattern matching with scaling and rotation, Theoretical Computer Science 410 (51) (2009) 5317–5333.
[6] C. Hundt, M. Liśkiewicz, On the complexity of affine image matching, in: Proceedings of Symposium on Theoretical Aspects of Computer Science, STACS, Lecture Notes in Computer Science, vol. 4393, Springer, 2007, pp. 284–295.
[7] C. Hundt, Affine image matching is uniform TC^0-complete, in: Proceedings of Combinatorial Pattern Matching, CPM, Lecture Notes in Computer Science, vol. 6129, Springer, 2010, pp. 13–25.
[8] C. Hundt, M. Liśkiewicz, Combinatorial bounds and algorithmic aspects of image matching under projective transformations, in: Proceedings of Mathematical Foundations of Computer Science, MFCS, Lecture Notes in Computer Science, vol. 5162, Springer, 2008, pp. 395–406.
[9] A. Amir, O. Kapah, D. Tsur, Faster two-dimensional pattern matching with rotations, Theoretical Computer Science 368 (3) (2006) 196–204.
[10] Y. Thibault, Rotations in 2d and 3d discrete spaces, Ph.D. thesis, University Paris-Est, 2010.
[11] A. Amir, G.M. Landau, U. Vishkin, Efficient pattern matching with scaling, Journal of Algorithms 13 (1) (1992) 2–32.
[12] A. Amir, A. Butman, M. Lewenstein, E. Porat, Real two dimensional scaled matching, Algorithmica 53 (3) (2009) 314–336.
[13] V.A. Kovalevsky, Finite topology as applied to image analysis, Computer Vision, Graphics & Image Processing 46 (2) (1989) 141–161.
[14] T.Y. Kong, A. Rosenfeld, Digital topology: introduction and survey, Computer Vision, Graphics & Image Processing 48 (3) (1989) 357–393.
[15] M. Sharir, Recent developments in the theory of arrangements of surfaces, in: Proceedings of Foundations of Software Technology and Theoretical Computer Science, FSTTCS, Lecture Notes in Computer Science, vol. 1738, Springer, 1999, pp. 1–21.

[16] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel, M. Sharir, Arrangements of curves in the plane-topology, combinatorics, and algorithms, Theorical Computer Science 92 (2) (1992) 319–336.
[17] J. Snoeyink, J. Hershberger, Sweeping arrangements of curves, in: Proceedings of Symposium on Computational Geometry, SOCG, ACM, 1989, pp. 354–363.
[18] D. Halperin, Arrangements, in: J.E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press LLC, 2004, pp. 529–562 (Chapter 24).
[19] K.H. Rosen, Elementary Number Theory and its Applications, third ed., Addison-Wesley, 1992.
[20] P. Flajolet, B. Vallée, Continued fractions, comparison algorithms, and fine structure constants, in: M. Théra (Ed.), Volume in the Honour of Jonathan Borwein, vol. 27, Amer. Math. Soc., Providence, 2000, pp. 53–82.
[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, pp. 499–513 (Chapter 31).
[23] K. Fredriksson, E. Ukkonen, Combinatorial methods for approximate pattern matching under rotations and translations in 3d arrays, in: Proceedings of String Processing and Information Retrieval, SPIRE, IEEE, 2000, pp. 96–104.
[24] A.Y. Khinchin, Continued Fractions, Dover Publications, 1964.
[25] E.V. Podsypanin, Length of the period of a quadratic irrational, Journal of Mathematical Sciences 18 (6) (1982) 919–923.

Journal of Mathematical Imaging and Vision manuscript No.
(will be inserted by the editor)

Topology-preserving conditions for 2D digital images under rigid transformations

Phuc Ngo · Yukiko Kenmochi · Nicolas Passat · Hugues Talbot

Received: date / Accepted: date

Abstract In the continuous domain $\mathbb{R}^n$, rigid transformations are topology-preserving operations. Due to digitization, this is not the case when considering digital images, *i.e.*, images defined on $\mathbb{Z}^n$. In this article, we begin to investigate this problem by studying conditions for digital images to preserve their topological properties under all rigid transformations on $\mathbb{Z}^2$. Based on (*i*) the recently introduced notion of DRT graph, and (*ii*) the notion of simple point, we propose an algorithm for evaluating digital images topological invariance.

Keywords Rigid transformation · 2D digital image · discrete topology · simple point · DRT graph

1 Introduction

Rigid transformations (*i.e.*, rotations composed with translations) are involved in numerous 2D and 3D image processing and analysis tasks, for instance in registration [1] and tracking [2]. In such applications, the images are necessarily digital, and can then be considered as functions $I : \mathbb{S} \to \mathbb{V}$ from a finite subset $\mathbb{S} \subset \mathbb{Z}^n$ to a value space $\mathbb{V}$.

Rigid transformations are topology-preserving in $\mathbb{R}^n$. By "topology-preserving", we mean that they do not alter intrinsic structural properties, generally modeled by topological invariants (*e.g.*, Euler characteristic, homotopy-type, fundamental groups, etc.). Unfortunately, this property, which is highly desirable in image analysis and processing, is generally lost in $\mathbb{Z}^n$. Typically, digital rigid transformations (*i.e.*, rigid transformations followed by a digitization process) alter, in most cases, the topological properties of digital images, such as the homotopy-type, as exemplified in Fig. 1. This is due to the sampling effects induced by the mandatory digitization process from $\mathbb{R}^n$ to $\mathbb{Z}^n$.

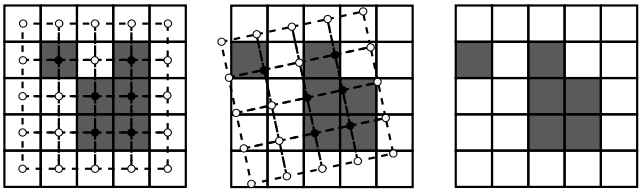


Fig. 1 Left: binary digital image and the grid modeling its discrete structure. Middle: a rigid transformation applied on this grid. Right: the resulting transformed image, which is not topologically identical to the initial image (for instance, if considered 8-connected, the black component was split).

Several works have been devoted to topology-preserving transformations, in particular in the context of image warping [3,4], and atlas-based segmentation [5], where topology preservation is a crucial issue. However topology topology-preservation in the case of rigid transformations in images has not yet been considered. In this article – that is an extended and improved version of the conference paper [6] – we study this specific issue. More precisely, we focus on the 2D case, and on defining some conditions for digital images such that their homotopy-type is preserved under all rigid transformations.

To this end, we consider the notion of DRT graph, that we recently introduced and studied from a theoretical point of view in [7]. It defines a combinatorial model of all the rigid transformations of a 2D digital image. We also consider the classical notion of simple points, which can be used to guarantee the preservation of homotopy-type, and

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

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

has been extended to several categories (binary, grey-level, labeled) of digital images. By combining these two notions, we provide a combinatorial analysis of 2D digital image topology under rigid transformations. The basic idea of the proposed method is to generate all the transformed images using the DRT graph, and simultaneously evaluate their homotopy-type preservation using the notion of simple points. This analysis finally leads us to an efficient algorithm for evaluating the homotopy-type preservation of digital images under all rigid transformations.

The article is organized as follows. The state of the art in both digital rigid transformation and topology preservation is exposed in Sec. 2. Basic definitions and notations are provided in Sec. 3. In Sec. 4, we introduce the main issues related to topology alterations induced by the embedding of rigid transformations into digital spaces. Our contribution is exposed in Secs. 5–7. Specifically, in Sec. 5, we introduce the DRT graph [7] as a tool for studying the behaviour of rigid transformations on digital images from a topological point of view. In particular, we propose a first algorithm for assessing the topological invariance of a digital image under all possible rigid transformations, with a superlinear time and space complexity, corresponding to that of the associated DRT graph. In Sec. 6, we refine this first approach, by spatially decomposing the image analysis process, leading to an equivalent algorithm that presents a low complexity with respect to the image size. In Sec. 7, this method is experimentally assessed in terms of complexity and correctness. Sec. 8 finally summarizes the contributions and proposes some future work.

2 State of the art

2.1 Rigid transformations on digital images

To reach our stated goal of finding conditions for topology preservation of digital images under all rigid transformations, it is necessary to compare the topological properties of the initial image and all of its transformed images. Studying the problem in the continuous domain is however unfeasible due to the uncountable number of transformations in $\mathbb{R}^2$, and so a discrete method must be devised, preferably involving only integer operations for exactness.

Over the last two decades, several methods were proposed to study transformations on digital images as fully discrete processes. In particular, different studies in combinatorial analysis for the problem of 2D pattern matching under different classes of geometric transformations have been considered for: rotations [8,9]; scalings [10,11]; combined scalings and rotations [12]; affine transformations [13,14]; projective and linear transformations [15]. To the best of our knowledge, fully discrete approaches devoted to rigid transformations are in quite limited number.

Digital rigid transformations include discrete rotations. For this class, one can cite the quasi-shear rotations [16,17] which were introduced to preserve bijectivity. The approach consists of decomposing a rotation into three horizontal or vertical quasi-shears in order to obtain a discrete transformation. Based on this strategy, this quasi-shear composition however trades bijectivity for transitivity. As a consequence, the result obtained by a quasi-shear rotation is not always identical to the result obtained by applying a Euclidean rotation followed by a discretization.

In this article, we propose an alternative approach that provides the same discretized result as the Euclidean discretized method, but we do not guarantee bijectivity. For this purpose, a discrete formulation of rotations based on hinge angles has been proposed in [18–21]. Informally, for a discrete finite set, rotations around a fixed center and with "small" angle variations will not result in any change. Conversely, some larger angle rotations will indeed result in pixel modifications. The notion of hinge angles formalizes this property. In particular, hinge angles (represented by integer triplets) hold sufficient information for incrementally generating and performing all rotations.

Following the idea of rotations by hinge angles and inspired by the discretization technique of the problem of 2D pattern matching, we have recently studied in [7] the combinatorial aspects and properties of the class of rigid transformations, by simultaneously considering the parameter space for both translations and rotations. Our approach discretizes the induced parameter space of rigid transformations on 2D digital images, and models this space by a combinatorial structure, namely a graph. This structure presents a space complexity of $O(N^9)$ for any subset of $\mathbb{Z}^2$ of size $N \times N$. Moreover, an algorithm to build this graph with *exact computation* (*i.e.*, using only integers), in linear time with respect to its space complexity is proposed in [7].

2.2 Topology-preserving digital image transformations

The study of discrete deformations involving topological alteration, relies mostly – but not exclusively [22] – on the notion of simple point, which provides conditions for the preservation of strong topological invariants, and in particular the homotopy-type. Intuitively, a point is called simple if its value can be modified without changing the digital topology of the associated image.

Simple points were initially defined for binary images on $\mathbb{Z}^2$ [23]. This notion was later formulated in the framework of digital topology [24], and was recently shown [25,26] to extend to richer discrete frameworks that explicitly describe cubic grids as topological spaces [27,28]. Several extensions have then been proposed during the following forty years, in terms of dimensions (3D [29] and 4D [30] simple points); of

cardinality (deletable sets [31], P-simple points [32], minimal simple sets [33]); and in terms of image value spaces (grey-level images [34], label images [35–37]).

On the applicative front, simple points have been intensively used in the development of various pattern recognition methods, for instance in the field of medical image analysis [38]. In particular, many such methods have been defined in monotonic transformation paradigms [39] (*i.e.*, reduction, skeletonization, region-growing) or in – continuous [40] or discrete [36] – deformation model paradigms.

Only a few works have involved topology preservation notions combined with geometric transformations, for instance in the field of digital image warping [41,4] based on continuous deformation fields obtained from registration procedures. In particular, the question of digital topology preservation in the case of rigid transformations has – to the best of our knowledge – not been considered until now.

3 Background notions

3.1 Digital images

In the continuous domain, a (2D) image can be formalised as a function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{V}$, where $\mathbb{V}$ is a value space. We assume that $\mathbb{V}$ contains at least two elements, including one, noted $\perp$, corresponding to the "background" value. In particular:

- if $|\mathbb{V}| = 2$, then I is a binary image;
- if $|\mathbb{V}| \geq 3$ and is equipped with a total order, then I is a grey-level image;
- if $|\mathbb{V}| \geq 3$ and is not equipped with a total order, then I is a label image.

A (2D) *digital* image associated to $\mathcal{I}$ can be defined as $I : \mathbb{Z}^2 \to \mathbb{V}$, by sampling $\mathcal{I}$ on the discrete space $\mathbb{Z}^2$. In other words, we have $I = \mathcal{I}_{|\mathbb{Z}^2}$, and for each $\mathbf{p} \in \mathbb{Z}^2$, the value $I(\mathbf{p})$ of the digital image models the value of $\mathcal{I}$ on the associated pixel $\mathbf{p} + [-\frac{1}{2}, \frac{1}{2}]^2$, namely the Voronoi cell of $\mathbb{R}^2$ induced by $\mathbb{Z}^2$ around $\mathbf{p}$. This paradigm relies on the digitization function D defined as

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

where $[\,\cdot\,]$ is a standard rounding function (for instance, the floor function $\mathbf{x} \to \lfloor \mathbf{x} + (\frac{1}{2}, \frac{1}{2}) \rfloor$). We assume that a digital image is actually defined on a subset of $\mathbb{Z}^2$, namely $I^{-1}(\mathbb{V} \setminus \{\perp\})$, which is finite. Then, it is plain that $I^{-1}(\mathbb{V} \setminus \{\perp\}) \subseteq \mathbb{S} = [0, N-1]^2 \cap \mathbb{Z}^2$, for a given $N \in \mathbb{N}$. The set $\mathbb{S}$ is called the support of I and $N \times N$ is the size of I. By abuse of notation – and without loss of generality – we will sometimes note a digital image as $I : \mathbb{S} \to \mathbb{V}$ instead of $I : \mathbb{Z}^2 \to \mathbb{V}$.

3.2 Digital rigid transformations

In the continuous framework, a rigid transformation (composed of translations and rotations) is expressed as a bijection $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$ defined, for any $\mathbf{x} = (x, y) \in \mathbb{R}^2$ by

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

where $a, b \in \mathbb{R}$ and $\theta \in [0, 2\pi[$. Such a transformation is unambiguously modeled by the triplet of parameters (a, b, θ), and will sometimes be noted $\mathcal{T}_{ab\theta}$.

It is not possible to apply directly $\mathcal{T}$ on a digital image $I : \mathbb{S} \to \mathbb{V}$, since there is no guarantee that $\mathcal{T}(\mathbf{x}) \in \mathbb{Z}^2$, for any $\mathbf{x} \in \mathbb{Z}^2$. The correct handling of *digital* rigid transformations then requires to define a digital analogue $T : \mathbb{Z}^2 \to \mathbb{Z}^2$ of $\mathcal{T}$. By considering the digitization paradigm proposed in Equation (1), this can be conveniently performed by setting

$$T = D \circ \mathcal{T}_{|\mathbb{S}} \tag{3}$$

In other words, the transformation T is obtained by applying $\mathcal{T}$ and then digitising the result by the function D, as illustrated in the diagram below.

$$\begin{array}{ccc} \mathbb{S} \subseteq \mathbb{Z}^2 & \xrightarrow{T = D \circ \mathcal{T}_{|\mathbb{S}}} & T(\mathbb{S}) \subseteq \mathbb{Z}^2 \\ \Big\downarrow Id & & \Big\uparrow D \\ \mathbb{S} \subseteq \mathbb{R}^2 & \xrightarrow{\mathcal{T}} & \mathcal{T}(\mathbb{S}) \subseteq \mathbb{R}^2 \end{array} \tag{4}$$

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

$$T(\mathbf{p}) = D \circ \mathcal{T}(\mathbf{p}) = \begin{pmatrix} [p\cos\theta - q\sin\theta + a] \\ [p\sin\theta + q\cos\theta + b] \end{pmatrix} \tag{5}$$

In $\mathbb{R}^2$, the transformation $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$ is bijective. Consequently, determining $\mathbf{y} \in \mathbb{R}^2$ such that $\mathcal{T}(\mathbf{x}) = \mathbf{y}$, and determining $\mathbf{x} \in \mathbb{R}^2$ such that $\mathcal{T}^{-1}(\mathbf{y}) = \mathbf{x}$, are equivalent questions. The first issue corresponds to the forwards model for image transformation, while the second issue corresponds to the backwards model (Fig. 2(a)).

In general, the bijectivity hypothesis is no longer verified in the digital case, for $T = D \circ \mathcal{T}_{|\mathbb{S}} : \mathbb{Z}^2 \to \mathbb{Z}^2$. In such a context, the forwards model (namely the Lagrangian model, illustrated in Fig. 2(b)) can be correctly handled, but not the backwards model (namely the Eulerian model). However, by setting $T^{-1} = D \circ \mathcal{T}^{-1}_{|\mathbb{Z}^2} : \mathbb{Z}^2 \to \mathbb{Z}^2$, we can define a transformed digital image $I \circ T^{-1} : \mathbb{Z}^2 \to \mathbb{V}$ that conveniently enables to handle the Eulerian model (Fig. 2(c)). (Note that T^{-1} is not the inverse function of T in general.)

In the sequel, we only focus on the Eulerian model (the justification of this choice will be discussed in Sec. 8). 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 $\mathcal{T}^{-1}$.

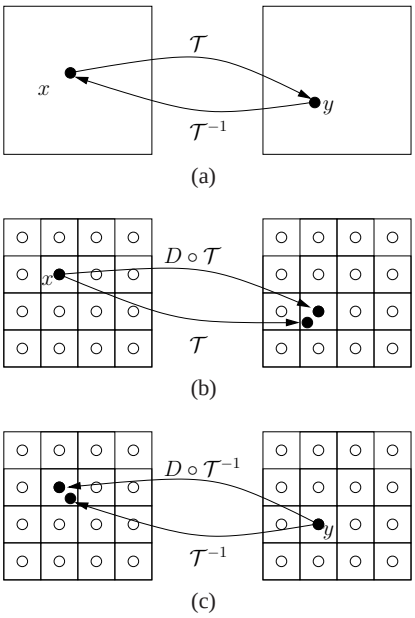


Fig. 2 (a) Forwards and backwards transformation models in $\mathbb{R}^2$. (b) Lagrangian and (c) Eulerian transformation models in $\mathbb{Z}^2$.

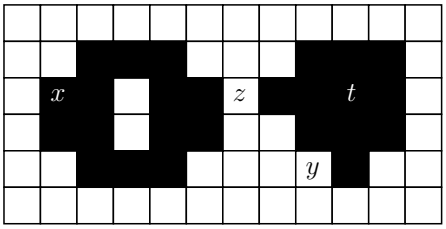


Fig. 3 Examples of simple ($\mathbf{x}, \mathbf{y}$) and non-simple ($\mathbf{z}, \mathbf{t}$) points in a binary image. Modifying the value of $\mathbf{z}$ would merge two black connected components, while modifying the value of $\mathbf{t}$ would create a white connected component. In both cases, the homotopy-type of the image would be modified.

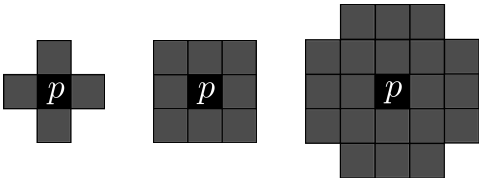


Fig. 4 The neighbourhoods N_4, N_8 and N_{20} of a point $\mathbf{p}$.

3.3 The topology of digital images

In the context of digital image transformations, the preservation of the image topology is often required, that is the preservation of given topological invariants. Among these topological invariants, the homotopy-type [42] is generally considered. As stated in Sec. 2.2, such homotopy-type preservation can be conveniently handled thanks to the notion of simple point, as described in the following property.

Property 1 ([23]) *Let $I : \mathbb{Z}^2 \to \mathbb{V}$ be a digital image. Let $\mathbf{p} \in I$ be a simple point of I. Then, the modified image I', obtained from I by modifying the value of I at $\mathbf{p}$ into a licit value (depending on $\mathbb{V}$ and I) has the same homotopy-type as I.*

Some examples and counter-examples of simple points are provided in Fig. 3.

Remark 2 *The notion of simplicity can be extended to sets of (successively) simple points between the initial image I and a final image I', that still preserves the homotopy-type between I and I'. This leads to a notion of* simple-equivalence *[43] between images.*

Independently from the kind of topological structure [24, 27, 28] mapped on $\mathbb{Z}^2$, and from the value space $\mathbb{V}$, simple points have the virtue of being characterizable by considering their immediate neighbourhood.

Property 3 *Let $I : \mathbb{Z}^2 \to \mathbb{V}$ be a digital image. Let $\mathbf{p} \in \mathbb{Z}^2$ be a point of I. The characterization of $\mathbf{p}$ as a simple point can be computed locally and in constant time by only considering the points $\mathbf{q} \in \mathbb{Z}^2$ such that $\|\mathbf{p} - \mathbf{q}\|_\infty \leq 1$.*

Based on these considerations, the concepts developed in the sequel of this article require only the following two hypotheses related to the considered images $I : \mathbb{Z}^2 \to \mathbb{V}$:

(H1) $\mathbb{Z}^2$ is equipped with a standard topological structure [24, 27, 28]; and
(H2) for this topological structure and the value space $\mathbb{V}$, a notion of simple point is available (this is, for instance, the case for binary, grey-level or label images).

For the sake of readability – but without loss of generality – we will hereafter focus on binary images endowed with the digital topology [24]. In this framework, the topological notions derive from a graph structure induced by two dual adjacency (*i.e.*, irreflexive and symmetric) relations, namely the 4- and 8-adjacencies, which are defined as follows. However, it is important to note that the results stated hereafter remain valid whenever hypotheses (H1) and (H2) are satisfied, as discussed in Sec. 6.3.

Definition 4 ([24]) *Given a point $\mathbf{p} = (p_1, p_2) \in \mathbb{Z}^2$, we consider the two neighbourhoods N_4 and N_8, which are defined for $\mathbf{p}$ as sets of points $\mathbf{q} = (q_1, q_2) \in \mathbb{Z}^2$ such that*

$$N_4(\mathbf{p}) = \{\mathbf{q} \in \mathbb{Z}^2 \mid \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{2} |p_i - q_i| \leq 1\} \tag{6}$$

$$N_8(\mathbf{p}) = \{\mathbf{q} \in \mathbb{Z}^2 \mid \|\mathbf{p} - \mathbf{q}\|_\infty = \max_{i=1}^{2} |p_i - q_i| \leq 1\} \tag{7}$$

We say that the point $\mathbf{q}$ is 4- (resp. 8-) adjacent *to $\mathbf{p}$ if $\mathbf{q} \in N_4(\mathbf{p}) \setminus \{\mathbf{p}\}$ (resp. $\mathbf{q} \in N_8(\mathbf{p}) \setminus \{\mathbf{p}\}$).*

Remark 5 *For reasons that will be justified in Sec. 6.1, we introduce a third neighbourhood for point $\mathbf{p}$, namely N_{20}, as*

well as the induced adjacency relation: the 20*-adjacency. It is defined by*

$$N_{20}(\mathbf{p}) = \left\{\mathbf{q} \in \mathbb{Z}^2 \mid \|\mathbf{p}-\mathbf{q}\|_2 = \Big(\sum_{i=1}^{2}(p_i-q_i)^2\Big)^{1/2} < 2\sqrt{2}\right\} \quad (8)$$

From the adjacency relations induced by these neighbourhoods (illustrated in Fig. 4), we define the notion of paths and derive important topological concepts from connectedness to fundamental groups. In this framework, the characterization of simple points $\mathbf{p}$ (which are either 4- or 8-simple, according to the chosen adjacency for the point value) can be made by only considering $N_8(\mathbf{p})$ (see Property 3).

4 Digital rigid transformations: Topological issues

As stated in Sec. 3.2, going from rigid transformations in $\mathbb{R}^2$ (Equation (2)) to digital rigid transformations in $\mathbb{Z}^2$ (Equation (5)) requires considering a digitization function (Equation (1)) that discretizes both the space and the transformation. In this section, we investigate such digitization effects on the topological properties of digital images during rigid transformations.

4.1 Non-preservation of geometric properties

In $\mathbb{R}^2$, rigid transformations are isometries and so preserve distances (in particular the Euclidean distance) between any pair of points, as well as the angles induced by any triplet of (distinct) points. However, when rigid transformations are digitised from $\mathbb{R}^2$ to $\mathbb{Z}^2$, these properties are often lost. Indeed, let us consider a point $\mathbf{p} \in \mathbb{Z}^2$ and a point $\mathbf{q} \in N_8(\mathbf{p})\setminus\{\mathbf{p}\}$. Let $\mathbf{p'}$ and $\mathbf{q'}$, obtained from a digital rigid transformation of $\mathbf{p}$ and $\mathbf{q}$, respectively. We have

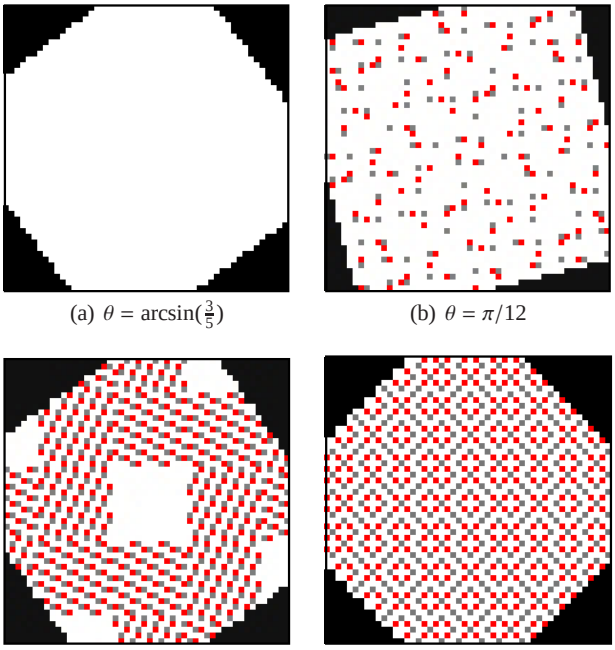
$$d_2(\mathbf{p},\mathbf{q}) = 1 \Longrightarrow d_2(\mathbf{p'},\mathbf{q'}) \in \{0, 1, \sqrt{2}\} \quad (9)$$

$$d_2(\mathbf{p},\mathbf{q}) = \sqrt{2} \Longrightarrow d_2(\mathbf{p'},\mathbf{q'}) \in \{1, \sqrt{2}, 2\} \quad (10)$$

where d_2 denotes the Euclidean distance. Similarly, alterations related to the angles between points can be derived as well.

Remark 6 *The fact that we may have* $d_2(\mathbf{p'},\mathbf{q'}) = 0$ *when* $d_2(\mathbf{p},\mathbf{q}) = 1$ *also implies that digital rigid transformations are non-injective in general. Due to the discrete nature of* $\mathbb{Z}^2$*, this also implies that such transformations are non-surjective.*

We show, in the remainder of this section, how such alterations can raise topological issues in the transformed spaces. To this end, we will first study some properties of pixels related to the influence of digital rigid transformations on their neighbourhoods.



(a) $\theta = \arcsin(\frac{3}{5})$ (b) $\theta = \pi/12$ (c) $\theta = \pi/5$ (d) $\theta = \pi/4$

Fig. 5 Some digital rotations by angles θ of a white square of size 100×100. Double points are depicted in red, and null points in grey.

4.2 Topological alterations due to digitization

From Remark 6, a digital rigid transformation T is generally not bijective. It is plain that for any three distinct points $\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3} \in \mathbb{Z}^2$, we have $\max_{i,j\in\{1,2,3\}}\{d_2(\mathbf{p_i},\mathbf{p_j})\} \geq \sqrt{2}$. From Equation (10), we derive that the three points $\mathbf{p'_1}, \mathbf{p'_2}, \mathbf{p'_3}$ obtained by a digital rigid transformation T of $\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}$ can not be mapped into the same pixel by the associated rigid transformation $\mathcal{T}$. It follows that we can characterise the *status* of a point $\mathbf{p} \in \mathbb{Z}^2$ with respect to T by using the set $P_T(\mathbf{p}) = \{\mathbf{q} \in \mathbb{Z}^2 \mid T(\mathbf{q}) = \mathbf{p}\}$ containing all points $\mathbf{q} \in \mathbb{Z}^2$ whose images by T is $\mathbf{p}$. In particular, there exist only three possibilities.

Definition 7 *Let us consider a point* $\mathbf{p} \in \mathbb{Z}^2$*, and a digital rigid transformation* T*.*

- *If* $|P_T(\mathbf{p})| = 0$*, then* $\mathbf{p}$ *is called a* null *point.*
- *If* $|P_T(\mathbf{p})| = 1$*, then* $\mathbf{p}$ *is called a* single *point.*
- *If* $|P_T(\mathbf{p})| = 2$*, then* $\mathbf{p}$ *is called a* double *point.*

From Definition 7, a digital rigid transformation T behaves like a bijection for single points, while the possible existence of null and double ones generally forbids T to be a surjection and an injection, as already evoked in Remark 6. This is a well-known issue, which has already been identified in the literature dealing with rotations in discrete spaces, for instance [8,44,19,21,45]. Some examples are provided in Fig. 5.

The existence of null and double points is a major source of topological alterations. Indeed, some connected compo-

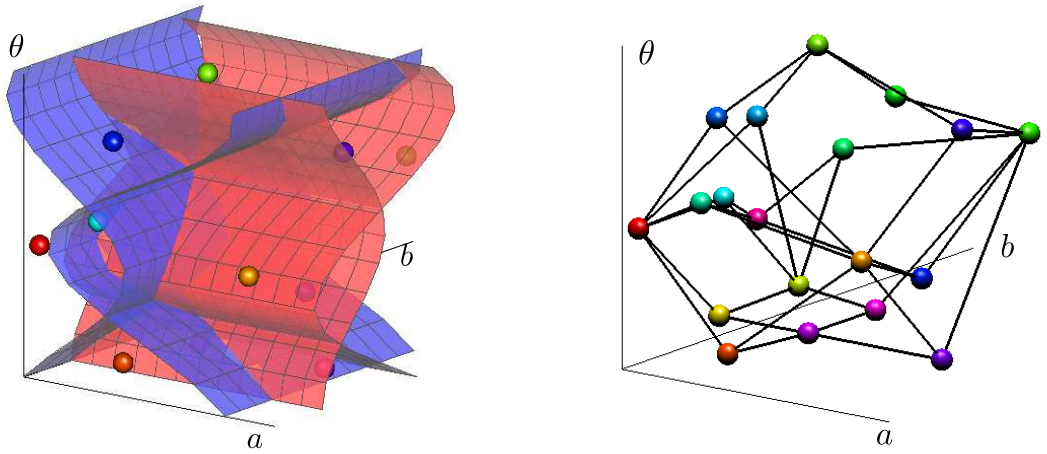


Fig. 6 A part of the parameter space subdivided by four 2D surfaces bounding the DRTs (left), and the associated (part of the) DRT graph (right).

nents may be lost when applying a digital rigid transformation, in particular the one-pixel components.

In addition to such cardinality-based problems, some adjacency-based issues are derived from the geometric alterations evoked in Sec. 4.1. Indeed, the non-preservation of distances between points, when applying a digital rigid transformation, has a direct interpretation in terms of modification of the adjacency relations between such points. The adjacency relations between points may change from 4- to 8-adjacency or *vice versa*, or could even lead to a loss of adjacency between points initially 8-adjacent. In such situations, some connected components may be either split or merged.

Remark 8 *Some topological alterations of the discrete structure of a subset $\mathbb{S}$ of $\mathbb{Z}^2$ do not necessarily lead to topological modifications of an image I defined on $\mathbb{S}$. Consequently, the study of the potential topological alterations induced by digital rigid transformations must be considered not only as a transformation-dependent problem, but also as an image-dependent one.*

5 DRT graphs and image topology

In this section, we briefly recall the notion of DRT graph proposed in [7], which is used to model the subdivision of the parameter space (a, b, θ) of rigid transformations. Then, we discuss how to use this structure as a topological analysis tool for rigidly transformed images.

5.1 A brief presentation of the DRT graph

Contrarily to rigid transformations in $\mathbb{R}^2$ (see Equation (2)), *digital rigid transformations* are not continuously defined with respect to the parameters a, b (controlling the "translation" part) and θ (controlling the "rotation" part). More precisely, the parameter space $\mathbb{R}^3$ of (a, b, θ) is divided into 3D open cells, in which the transformations $T_{ab\theta}$ are equal, while the 2D surfaces bounding these open cells correspond to discontinuities of the digital rigid transformations, induced by the digitization process (see Equation (5)).

From a theoretical point of view, each 3D open cell of the parameter space (a, b, θ) can be seen as an equivalence class of rigid transformations $\mathcal{T}$ of $\mathbb{R}^2$ that lead to a same transformation $T = D \circ \mathcal{T}$ in $\mathbb{Z}^2$. Such an equivalence class is called a *discrete rigid transformation*[1] (DRT) [7]. Each 3D open cell can also be considered as the resulting digital transformed space generated by any digital rigid transformation of the associated DRT. Moreover, the existence of a 2D surface between two cells indicates that the two associated transformed images differ by exactly one pixel. By mapping any 3D cell onto a 0D point, and any 2D surface onto a 1D edge, the combinatorial structure of the parameter space can be modeled, in a dual way, as a (connected) graph, namely a *DRT graph* (see Fig. 6).

Definition 9 ([7]) *Let $G = (V, E)$ be the graph defined such that:*

- *any vertex $v \in V$ models a 3D open cell associated to a DRT;*
- *any edge $e = (v, w) \in E$ models a 2D surface between two distinct vertices $v, w \in V$.*

The graph $G = (V, E)$ is called a DRT graph.

A DRT graph models the subdivision of the whole rigid transformation parameter space; therefore, it models *all* the

[1] The term *digital* refers to the digitization process of numeric images and transformations for such images, while the term *discrete* refers to the non-continuous structure of these transformations.

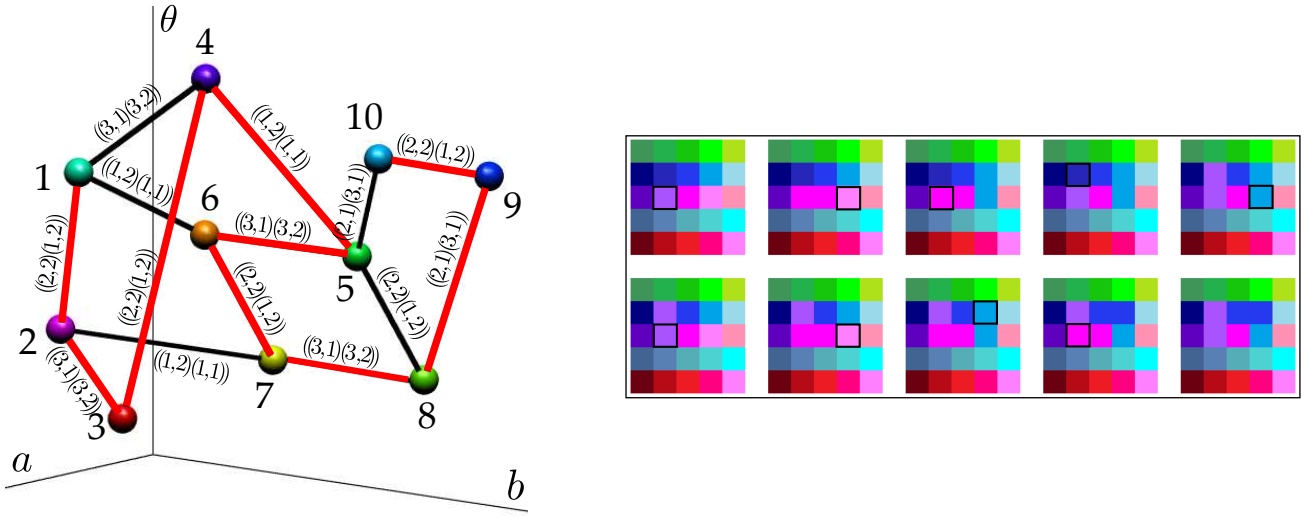


Fig. 7 Left: part of a DRT graph G in which each vertex is a DRT representing a digital transformed image, and each edge indicates the *only* value modification of a pixel between two connected vertices. More precisely, if an edge $e = (v, w, (\mathbf{p}, \mathbf{p}'))$ connects two vertices v and w, then the associated images I_v and I_w of v and w differ at the *single* pixel $\mathbf{p}'$, and $\mathbf{p}$ is the pixel corresponding to $\mathbf{p}'$ in the original image (see text). Right: the transformed images associated to the vertices of the DRT graph G and their relations according to the edges in G. The images from left to right and from top to bottom correspond to the path from the vertices 1 to 10 (in red), in which the differing pixel $\mathbf{p}'$ between two transformed images of the consecutive vertices I_{v_i} and $I_{v_{i+1}}$ is depicted by the black frame in I_{v_i}, and the first image corresponds to the original image.

possible rigid transformations of a given set $\mathbb{S}$. Despite the fact that the space of these transformations is infinite, the DRT graph is actually defined as a *finite* structure. In [7], the space complexity of the DRT graph for any set $\mathbb{S}$ of size $N \times N$ has been proved to be polynomial. An exact computation algorithm also exists to build this graph in linear time with respect to its size.

Property 10 ([7]) *The DRT graph associated to a set $\mathbb{S}$ of size $N \times N$ has a space complexity of $O(N^9)$.*

DRT graphs do not depend on the values that are assigned to the pixels of $\mathbb{S}$. In other words, their structure is invariant for any image defined on a same support $\mathbb{S}$. In the sequel, we will however consider – without loss of generality – a DRT graph with respect to a given image I defined on $\mathbb{S}$. In this context, any edge $e = (v, w) \in E$ of the DRT graph can be "enriched" as $e = (v, w, (\mathbf{p}, \mathbf{p}'))$, where $\mathbf{p}'$ is the only pixel where the transformed images differ with respect to the DRTs v and w, respectively, while $\mathbf{p}$ is the pixel corresponding to $\mathbf{p}'$ in the initial image I.

The DRT graph relies on geometric information provided by (a, b, θ). However, it does not explicitly model such geometric information. Indeed, it only provides structural information, that models the relationship between any "neighbouring" transformed images. In particular, the label $(\mathbf{p}, \mathbf{p}')$ of each edge e is implicitly associated to a function indicating the value modification of the pixel $\mathbf{p}'$ that differs between the transformed images corresponding to the DRTs v and w. More precisely, the rigid transformation associated to the 2D surface of the edge e modifies only the pixel value of $\mathbf{p}'$ (which is initially equal to the value of $\mathbf{p}$), such that $\mathbf{p}'$ will get its value from one of the 4-neighbouring pixels of $\mathbf{p}$. This property is exemplified in Fig. 7. Practically, let I_v and I_w be the transformed images corresponding to the vertices v and w respectively. The value of $\mathbf{p}'$ at the vertex v is defined by $I_v(\mathbf{p}') = I(\mathbf{p})$ where $I : \mathbb{S} \to \mathbb{V}$ is the original image function. After the elementary modification at the edge e, we obtain a new transformed image I_w by simply changing the pixel value at $\mathbf{p}'$ as $I_w(\mathbf{p}') = I(\mathbf{p} + \boldsymbol{\delta})$ where $\boldsymbol{\delta} = (\pm 1, 0)$ or $(0, \pm 1)$. In this way, one can generate all the transformed images of I by incrementally and exhaustively scanning the associated DRT graph.

Remark 11 *Let $G = (V, E)$ be a DRT graph associated to a given image $I : \mathbb{S} \to \mathbb{V}$. For each edge $e = (v, w, (\mathbf{p}, \mathbf{p}')) \in E$, two cases can occur:*

(*i*) $I_v(\mathbf{p}') = I_w(\mathbf{p}')$, i.e., *the transformed images of I by the DRTs v and w are equivalent ($I_v = I_w$);*
(*ii*) $I_v(\mathbf{p}') \neq I_w(\mathbf{p}')$, i.e., $I_v \neq I_w$.

In the case of binary images, the value of $\mathbf{p}'$ may then be flipped from white to black or *vice versa*, and this may constitute the only modification between the images of I by two consecutive DRTs. Such a value change may consequently alter the topological property of the binary images. In the sequel, we will show how to verify whether this actually occurs, for any arbitrary transformations, using a DRT graph.

5.2 DRT graph as a topological analysis tool

On the one hand, we would like to know if a given image I defined on $\mathbb{S}$ preserves its topological properties under all

digital/discrete rigid transformations. Let us first formalise this preservation, via the notion of *topological invariance*.

Definition 12 *A digital image I is* topologically invariant *if all its transformed images have the same homotopy-type as I.*

On the other hand, as mentioned in Sec. 5.1 the DRT graph allows us to generate exhaustively *all* the transformed images of I. From the definition of the DRT graph and from Remark 11, this can be achieved by incrementally modifying (at most) one pixel value between two successive transformed images. Moreover, from Property 1, the notion of simple point can be used to handle the topological-invariance concept (in particular, the homotopy-type) between two images that differ in exactly one point. We also know from Remark 2 that this preservation of the homotopy-type is also guaranteed via the notion of simple-equivalence, that consists of considering successively simple points. The local characterization of simple points and the incremental notion of simple-equivalence are therefore compatible with an incremental exploration of the DRT graph of image I, in order to evaluate its topological invariance.

Practically, the edges of the DRT graph $G = (V, E)$ of I can be classified in two categories: those that do not modify the topology of the transformed images and those that do. The first category contains the edges that correspond to the case (*i*) in Remark 11 as well as those that correspond to the case (*ii*) for which $\mathbf{p'}$ is a simple point; and the second one contains the edges that correspond to the case (*ii*) in Remark 11 for which $\mathbf{p'}$ is not simple.

Based on this binary classification, we can straightforwardly create a partial graph $G' = (V, E')$ of G by preserving in $E' \subseteq E$ only the edges of the first category. In particular, if G' is connected, it is plain that I is topologically invariant. Otherwise, I is not topologically invariant, and every connected component in G' corresponds to a set of simple-equivalent transformed images. It should be noticed that if there exist at least 3 such connected components, there is no guarantee of maximality of this simple-equivalence property. In other words, two separate (and non-adjacent) connected components in G' may be composed of images that all present the same topology. This property derives from the fact that the DRT graph does not model all the possible paths associated to transformations between two images, but is restricted to those that have a rigid transformation semantics.

Such an approach presents an algorithmic complexity that is linear with respect to the (polynomial) space complexity of the DRT graph. It is however possible to reach a better (mean) complexity by using a standard spanning-tree algorithm (see Algorithm 1), that provides two outputs: a Boolean evaluating the topological invariance of I, and a (non-necessarily maximal) set of simple-equivalent transformed images with respect to the image associated to the initial vertex u in the DRT graph (*e.g.*, I or any other transformed image of I). In Algorithm 1, the graph G'' providing the set of simple-equivalent images is in fact a partial sub-graph of G' (and of G as well).

Algorithm 1: Generation of simple-equivalent images and topological invariance verification.

Input: A DRT graph $G = (V, E)$ associated to an image I.
Input: An initial vertex $u \in V$ corresponding to the image I.
Output: A partial sub-graph $G'' = (V'', E'')$ of G such that, for any $v \in V''$, the images I_v are simple-equivalent to I.
Output: A Boolean B that indicates if I is a topologically invariant image.

```
1  (V'', E'') ← ({u}, ∅)
2  S ← {u}
3  while S ≠ ∅ do
4      Let v ∈ S
5      S ← S \ {v}
6      foreach e = (v, w, (p, p')) ∈ E, such that w ∉ V'' do
7          if (I_v(p') = I_w(p')) or ((I_v(p') ≠ I_w(p')) and (p' is a
           simple point in I_v)) then
8              (V'', E'') ← (V'' ∪ {w}, E'' ∪ {e})
9              S ← S ∪ {w}
10 B ← (V = V'')
```

Nevertheless, the high algorithmic complexity of this approach practically forbids the generation the whole graph for large images, and therefore to consequently verify topological invariance. In the next section, we show that this problem can however be decomposed spatially, thus leading to a much lower complexity algorithm.

6 A local approach for analyzing topological invariance under DRTs

In the previous section, we have proposed to explore the whole DRT graph of a given image I in order to evaluate its topological invariance for all DRTs. More precisely, for each edge $e = (v, w, (\mathbf{p}, \mathbf{p'}))$ of the DRT graph, this exploration consists of verifying that $\mathbf{p'}$ is a simple point between the transformed images I_v and I_w with respect to the DRTs v and w, if $I_v \neq I_w$. From Property 3, we know that this verification can be carried out locally, more precisely in the neighbourhood $N_8(\mathbf{p'})$ of the transformed image space(s). We now propose to take advantage of the local nature of these tests to develop a space decomposition strategy that leads to a *local* version of the previously proposed *global* method.

6.1 From global to local DRTs

On the one hand, it is plain that the set of all DRTs defined on a subset of size $N \times N$ of $\mathbb{Z}^2$, does not depend on the way to locate this subset into $\mathbb{Z}^2$. In other words – provided that

we choose a set $\mathbb{S} \subset \mathbb{Z}^2$ "sufficiently large" to include the informative part of I – the DRT graph $G = (V, E)$ associated to an image $I : \mathbb{S} \to \mathbb{V}$ is isomorphic to the DRT graph of any translated image of I (this isomorphism actually concerns the vertices/edges that involve at least one point with a value distinct from $\perp$). In particular, for a given $\mathbf{p} \in \mathbb{Z}^2$ let us consider the image $I_{\mathbf{p}}$ such that for any $\mathbf{q} \in \mathbb{Z}^2$, $I_{\mathbf{p}}(\mathbf{q}) = I(\mathbf{q} - \mathbf{p})$. Then, any edge $e = (v, w, (\mathbf{p}, \mathbf{p'}))$ of the DRT graph G of I is equivalent to the edge $e' = (v', w', (\mathbf{0}, \mathbf{p'}))$ (also denoted by $e' = (v', w', (\mathbf{o_1}, \mathbf{p'}))$ in the following) of the DRT graph $G_{\mathbf{p}}$ of $I_{\mathbf{p}}$ and v', w' are the DRTs corresponding to v, w, respectively, up to the translation of vector $-\mathbf{p}$.

On the other hand, let us consider an edge $e = (v, w, (\mathbf{p}, \mathbf{p'}))$ of the DRT graph G of I. For a given $\mathbf{p'} \in \mathbb{Z}^2$, we can have two images $I_{v'}$ and $I_{w'}$ with respect to I_v and I_w such that, for any $\mathbf{q} \in \mathbb{Z}^2$, $I_{v'}(\mathbf{q}) = I_v(\mathbf{q} - \mathbf{p'})$ and $I_{w'}(\mathbf{q}) = I_w(\mathbf{q} - \mathbf{p'})$. Therefore, any edge $e = (v, w, (\mathbf{p}, \mathbf{p'}))$ is considered to be equivalent to $e' = (v', w', (\mathbf{p}, \mathbf{0}))$ (also denoted by $e' = (v', w', (\mathbf{p}, \mathbf{o_2}))$ in the following), where v', w' correspond to v, w, respectively, up to the translation of vector $-\mathbf{p'}$.

From the two above paragraphs, we derive the following statement.

Remark 13 *The study of any edge of label* $(\mathbf{p}, \mathbf{p'})$ *in the DRT graph* $G = (V, E)$ *associated to an image* $I : \mathbb{S} \to \mathbb{V}$ *can be carried out by considering the edge of label* $(\mathbf{o_1}, \mathbf{o_2})$ *in the equivalent DRT graph* $G_{\mathbf{p}} = (V_{\mathbf{p}}, E_{\mathbf{p}})$ *associated to a translated image* $I_{\mathbf{p}}$ *of* I.

In order to establish our local strategy, we now state some lemmas related to the behaviour of DRTs with respect to the 8-neighbourhood. Our first lemma, derived from Definition 4 and Equation (5), deals with the extension of a 8-neighbourhood induced by digital rigid transformations.

Lemma 14 *Let* $\mathbf{p} \in \mathbb{Z}^2$ *and* $\mathbf{q} \in N_8(\mathbf{p})$. *For any digital rigid transformation* $T : \mathbb{Z}^2 \to \mathbb{Z}^2$, *we have* $T(\mathbf{q}) \in N_{20}(T(\mathbf{p}))$.

From the result of Remark13 and the local characterization of simple points (Property 3), we then derive the following lemma where we consider T_v as the digital rigid transformation associated to a DRT v, thanks to Lemma 14. Our next lemma states that it is sufficient to consider a local neighbourhood to evaluate simple points under rigid transformations.

Lemma 15 *Let* $I : \mathbb{S} \to \mathbb{V}$ *be a digital image. Let* $I' : N_{20}(\mathbf{p}) \to \mathbb{V}$ *be the restriction of* I *to* $N_{20}(\mathbf{p})$ *for any* $\mathbf{p} \in \mathbb{S}$. *Let* v, w *(resp.* v', w'*) be two adjacent vertices of the DRT graph* G *(resp.* $G'_{\mathbf{p}}$*) associated to* I *(resp.* I'*) such that the DRTs* T_v, T_w *(resp.* $T_{v'}, T_{w'}$*) differ only in* $\mathbf{p'}$ *and* $T_v(\mathbf{p'}) = \mathbf{p}$ *(resp.* $T_{v'}(\mathbf{p'}) = \mathbf{p}$*). Let* $I_v, I_w : \mathbb{S} \to \mathbb{V}$ *(resp.* $I'_{v'}, I'_{w'} : N_{20}(\mathbf{p}) \to \mathbb{V}$*) be the transformed images of* I *(resp.* I'*) with respect to* v, w *(resp.* v', w'*), according to the Eulerian model. Then* $\mathbf{p'}$ *is a simple point in* I_v *(and* I_w*) iff* $\mathbf{p'}$ *is a simple point in* $I'_{v'}$ *(and* $I'_{w'}$*).*

In the DRT graph G of an image I, we can define an *equivalence relation* between the edges of G as follows.

Definition 16 *Let* $G = (V, E)$ *be the DRT graph associated to an image* I, *and* $E_{(\mathbf{p},\mathbf{p'})} \subset E$ *be the set of edges with* $(\mathbf{p}, \mathbf{p'})$ *as their label. Two edges* $e_1 = (v_1, w_1, (\mathbf{p}, \mathbf{p'}))$ *and* $e_2 = (v_2, w_2, (\mathbf{p}, \mathbf{p'}))$ *in* $E_{(\mathbf{p},\mathbf{p'})}$ *are* equivalent, *and denoted by* $e_1 \sim e_2$, *iff* $T_{v_1 \mid N_8(\mathbf{p'})} = T_{v_2 \mid N_8(\mathbf{p'})}$ *(and* $T_{w_1 \mid N_8(\mathbf{p'})} = T_{w_2 \mid N_8(\mathbf{p'})}$*).*

In other words, an equivalence class of any $e = (v, w, (\mathbf{p}, \mathbf{p'})) \in E_{(\mathbf{p},\mathbf{p'})}$ under $\sim$, denoted by $[(v, w, (\mathbf{p}, \mathbf{p'}))]_\sim$, contains the set of T_v that provide the same transformed image in the restriction of I to $N_{20}(\mathbf{p})$. Let us consider the DRT graph $G'_{\mathbf{p}}$ associated to the 20-neighbourhood of $\mathbf{o_1}$ in the translated image $I_{\mathbf{p}}$. According to the Eulerian model and Lemma 14, this DRT graph $G'_{\mathbf{p}}$ contains edges $(v', w', (\mathbf{o_1}, \mathbf{o_2}))$ that "summarize" the edges $(v, w, (\mathbf{p}, \mathbf{p'}))$ of the DRT graph G associated to I.

Proposition 17 *Let* $E_{(\mathbf{p},\mathbf{p'})}$ *(resp.* $E'_{\mathbf{p}\ (\mathbf{o_1},\mathbf{o_2})}$*) be the set of edges with* $(\mathbf{p}, \mathbf{p'})$ *(resp.* $(\mathbf{o_1}, \mathbf{o_2})$*) as their label of the graph* $G = (V, E)$ *(resp.* $G'_{\mathbf{p}} = (V'_{\mathbf{p}}, E'_{\mathbf{p}})$*) associated to* I *(resp.* $I'_{\mathbf{p}}$, *the restriction of* $I_{\mathbf{p}}$ *to* $N_{20}(\mathbf{o_1})$*). We have* $E_{(\mathbf{p},\mathbf{p'})}/\sim$ *equivalent to* $E'_{\mathbf{p}\ (\mathbf{o_1},\mathbf{o_2})}$, *by associating each equivalence class* $[(v, w, (\mathbf{p}, \mathbf{p'}))]_\sim \in E$ *to the edge* $(v', w', (\mathbf{o_1}, \mathbf{o_2})) \in E'_{\mathbf{p}\ (\mathbf{o_1},\mathbf{o_2})}$ *such that* $T_{v \mid N_8(\mathbf{p'})} = T_{v'}$ *and* $T_{w \mid N_8(\mathbf{p'})} = T_{w'}$.

Since $\mathbf{o_1}$ and $\mathbf{o_2}$ are the origins $\mathbf{0}$ of the images I and $I'_{\mathbf{p}}$ respectively, the relations between I, G, $E_{(\mathbf{0},\mathbf{0})}$ and $I'_{\mathbf{p}}$, $G'_{\mathbf{p}}$, $E'_{\mathbf{p}\ (\mathbf{0},\mathbf{0})}$ are illustrated in Figs. 8 and 9.

Based on Lemma 15 and Proposition 17, it follows that the "topological" behaviour of any edge of $[(v, w, (\mathbf{p}, \mathbf{p'}))]_\sim$ in the DRT graph G associated to image I can be determined from the edges $(v', w', (\mathbf{o_1}, \mathbf{o_2}))$ in the DRT graph $G'_{\mathbf{p}}$. In other words, the study of the *local* DRT graph $G'_{\mathbf{p}}$ associated to the partial images of I defined on $N_{20}(\mathbf{p})$ directly provides access to a subset of the required *global* knowledge related to the topological invariance of I under any DRTs.

In particular, from Remark 13, Lemmas 14, 15, and Proposition 17, it becomes possible to develop a local approach for the topological invariance verification of digital images under all rigid transformations.

6.2 LUT-based algorithm

Practically, an image I is topologically invariant with respect to all DRTs if all its transformed images share the same homotopy-type, and in particular if they are simple-equivalent (Remark 2). This simple-equivalence can be locally determined using the notion of simple point (Properties 1 and 3). In particular, any elementary modification between transformed images is encoded in an edge of the DRT graph G of I, and such an edge models the modification of exactly one point between two transformed images.

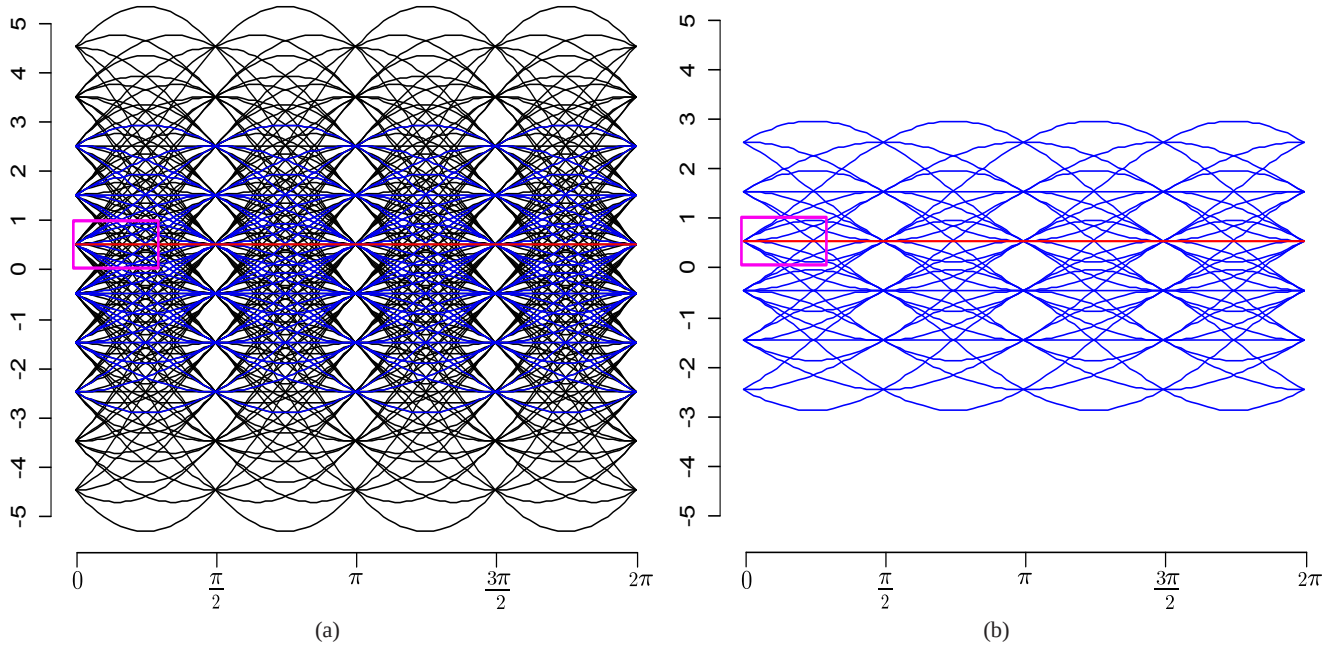


Fig. 8 (a) A cross-section in the plane (a, θ) of the 2D surfaces bounding the DRTs (see Sec. 5.1) associated to the image I, and inducing the DRT graph $G = (V, E)$. (b) A cross-section in the plane (a, θ) of the 2D surfaces bounding DRTs associated to $I' = I_{|N_{20}(\mathbf{0})}$, and inducing the DRT graph $G' = (V', E')$ (see text). In both figures, the red segments correspond to the edges of label $(\mathbf{0}, \mathbf{0})$ (*i.e.*, $E_{(\mathbf{0},\mathbf{0})}$ and $E'_{(\mathbf{0},\mathbf{0})}$), while the blue ones are the edges in E' and the black ones are the edges in $E \setminus E'$.

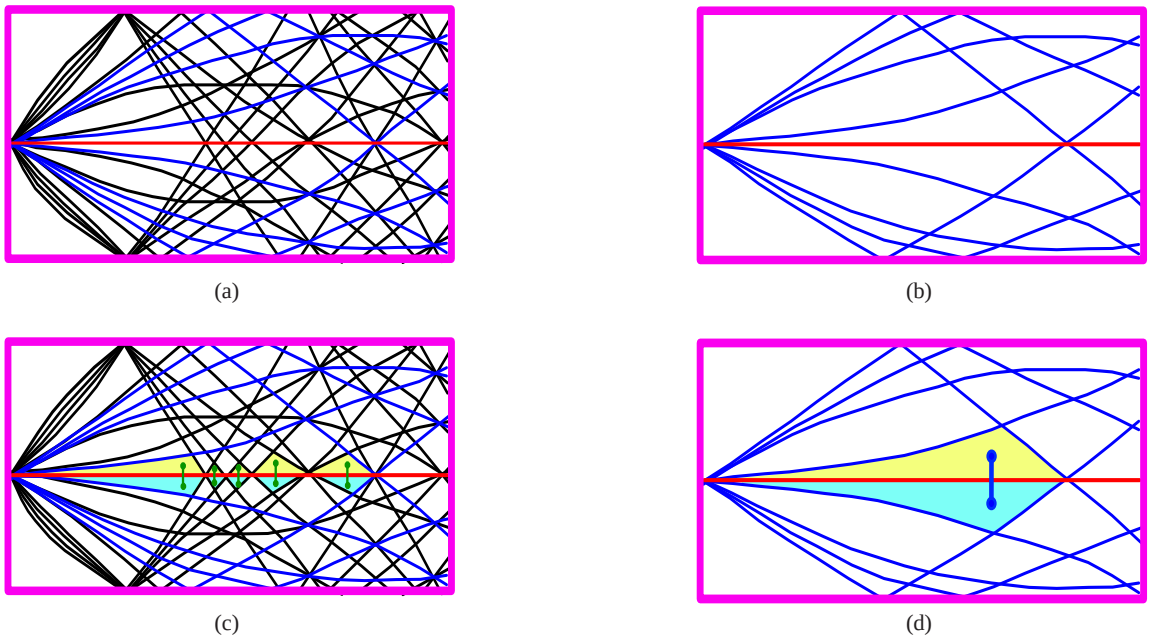


Fig. 9 (a,b) Zoom in the curves of Fig. 8. (c,d) Illustration of the dual structures of (a,b) for the part of the DRT graph corresponding to the edges with label $(\mathbf{0}, \mathbf{0})$. By Definition 16, the green edges in (c) form an equivalence class $[(v, w, (\mathbf{0}, \mathbf{0}))]_\sim \in E$. From Proposition 17, the equivalence class $[(v, w, (\mathbf{0}, \mathbf{0}))]_\sim$ can be associated to the blue edge $(v', w', (\mathbf{0}, \mathbf{0}))$ in (d).

This point can in particular be characterised as simple or not. Consequently, by analysing the edges of the whole DRT graph G, the topological invariance of I can be determined. This is the strategy developed in Algorithm 1, that processes these edges in a exhaustive fashion, leading to a computational cost directly linked to the size of the DRT graph.

In the previous section, it was observed that any edge of the DRT graph G of $I : \mathbb{S} \to \mathbb{V}$ is equivalent to an edge in a smaller DRT graph $G'_{\mathbf{p}}$, associated to the restriction of I in

Algorithm 2: LUT generation for topological invariance verification.

Input: The DRT graph $G'_{\mathbf{0}} = (V'_{\mathbf{0}}, E'_{\mathbf{0}})$ associated to $N_{20}(\mathbf{0})$.
Input: The set C of all different images $I : N_{20}(\mathbf{0}) \to \mathbb{V}$ (computed in a greedy fashion).
Output: The set $P \subseteq C$ of topologically preserving samples for the center point $\mathbf{0}$.

$P \leftarrow \emptyset$
foreach $I \in C$ **do**
  $B \leftarrow true$
  $S \leftarrow E_{\mathbf{0}}$
  while $(S \neq \emptyset)$ *and* $(B = true)$ **do**
    Let $e = (v, w, (\mathbf{p}, \mathbf{p}')) \in S$
    $S \leftarrow S \setminus \{e\}$
    if $\mathbf{p} = \mathbf{0}$ **then**
      if $((I_v(\mathbf{p}') \neq I_w(\mathbf{p}'))$ *and* ($\mathbf{p}'$ *is not a simple point in* I_v)) **then**
        $B \leftarrow false$
  if $B = true$ **then**
    $P \leftarrow P \cup \{I\}$

Algorithm 3: Local verification of the topological invariance of a digital image.

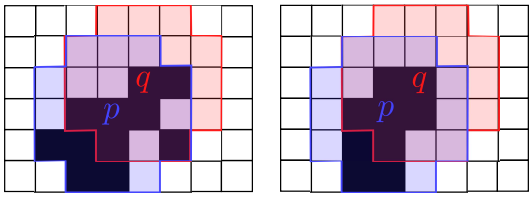
Input: A digital image $I : \mathbb{S} \to \mathbb{V}$.
Input: The set P (computed from Algorithm 2).
Output: A Boolean value B evaluating the topological invariance of I.

$B \leftarrow true$
$S \leftarrow \mathbb{S}$
while $(S \neq \emptyset)$ *and* $(B = true)$ **do**
  Let $p \in S$
  $S \leftarrow S \setminus \{p\}$
  $B \leftarrow (I_{|N_{20}(\mathbf{p})} \in P)$ (up to a translation of $-\mathbf{p}$)

the 20-neighbourhood of a given point $\mathbf{p} \in \mathbb{S}$ (Remark 13 and Proposition 17). In particular, the characterization of this edge as topologically preserving is algorithmically the same in G and in $G'_{\mathbf{p}}$ (Lemma 15 and Proposition 17).

From these facts, we deduce that the topological invariance of I can be equivalently analyzed from G or from the set $\{G'_{\mathbf{p}}\}_{\mathbf{p}\in\mathbb{S}}$ of all the local DRT graphs in the 20-neighbourhoods of the points $\mathbf{p} \in \mathbb{S}$. In particular, in any of these local DRT graphs $G'_{\mathbf{p}}$, it is sufficient to focus on a (strict) subset of edges, namely those that involve $\mathbf{p}$.

Moreover, since only a finite number of images can be defined on a 20-neighbourhood, this topological analysis can be performed exhaustively just once for all the images defined on a 20-neighbourhood; these images can then be used to characterize the topological invariance of I. This pre-computation, formalised in Algorithm 2, leads to the definition of a look-up table (LUT) P that contains all the 20-neighbourhood images that authorize topological invariance in a larger image, for a given value space $\mathbb{V}$, and a given topology.



(a) (b)

Fig. 10 (a) A 20-neighbourhood image, centered on $\mathbf{p}$ (in blue), that belongs to the LUT P, and an (overlapped) image, centered on $\mathbf{q}$ (in red), that does not belong to P. (b) Two 20-neighbourhood images, centered on $\mathbf{p}$ and $\mathbf{q}$, respectively, that both belong to the LUT P. (See Remark 18.)

Remark 18 *The LUT P obtained from Algorithm 2 potentially constitutes a* strict superset *of the actual set of the 20-neighbourhood images that authorize in the LUT some patterns that necessarily imply the existence of neighbouring patterns that are not themselves in the LUT (see Fig. 10). Algorithm 2 can then be optimised by a post-processing that removes from P some non-relevant configurations. Such a post-processing, that leads to a smaller LUT, presents a time complexity $O(|P|^3)$.*

We discuss in more details experimental results obtained with this LUT in the case of binary images in Sec. 7.2. Once the LUT P has been computed, any image $I : \mathbb{S} \to \mathbb{V}$ can be characterized by a simple pixelwise process, that checks, for every $\mathbf{p} \in \mathbb{S}$, that the restriction of I to $N_{20}(\mathbf{p})$ belongs to P. This LUT-based approach is formalised in Algorithm 3.

6.3 Parametrisation of the approach

The proposed approach for evaluating the topological invariance of digital images I defined on $\mathbb{S}$, under all DRTs, has been presented – for the sake of readability – in the classical framework of digital topology [24], *i.e.*, by considering binary images ($|\mathbb{V}| = 2$), equipped with a standard pair of dual $(8, 4)$- or $(4, 8)$-adjacencies. Nevertheless, the nature (and thus the cardinality) of $\mathbb{V}$, such as the topological space used to equip $\mathbb{S}$ with respect to $\mathbb{V}$, can be conveniently modified without loss of generality, making the proposed approach parametric from both the structural and the spectral points of view.

Indeed, on the one hand, the proposed algorithms (and in particular Algorithms 2 and 3) rely on the notion of DRT graph, that defines explicitly the structure of the transformed spaces, but neither the transformed images nor their associated value space $\mathbb{V}$, which are implicitly handled.

On the other hand, the topological space that is mapped on $\mathbb{S}$ (and more generally on $\mathbb{Z}^2$) with respect to $\mathbb{V}$, is only considered via the notion of simple point. More precisely, the only constraint related to the choice of the topology is the

necessity to characterise locally the preservation of homotopy-type, with respect to the images of $\mathbb{S} \to \mathbb{V}$.

Consequently, the proposed approach can be parametrized by a couple composed of (*i*) a value space $\mathbb{V}$, and (*ii*) a notion of simple point for the space of the images of $\mathbb{S} \to \mathbb{V}$.

Based on this assertion, several topological frameworks that provide different notions of simple point can be considered, including the following:

- binary images, equipped with the digital topology [24], deriving from the dual $(4,8)$- or $(8,4)$-adjacencies (proposed in this article);
- binary images, defined as well-composed sets [46], deriving from the $(4,4)$-adjacencies;
- label images, defined as well-composed sets [35,36], deriving from the $(4,4)$-adjacencies;
- label images, equipped with the notion of digitally simple Xels [26] defined from the topology of cubical complexes;
- label images, equipped with the notion of simple point in covering images [47];
- grey-level images, equipped with the notion of λ-destructible point [34].

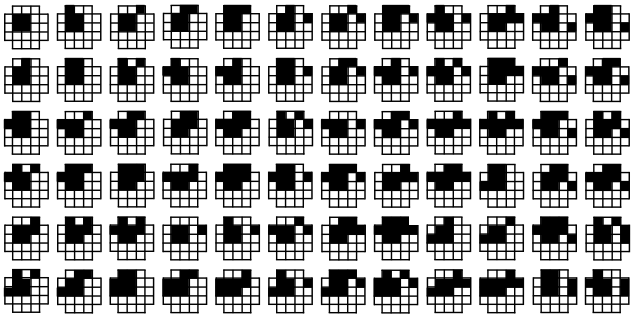
7 Analysis and experiments

In this section, we first analyse the complexity of the proposed algorithms. We then provide experiments devoted to validate the behaviour of the developed approach.

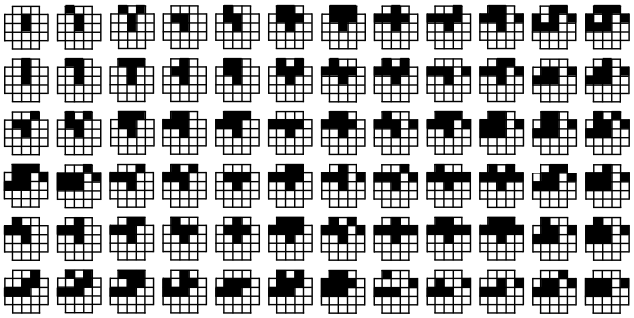
7.1 Theoretical complexity analysis

Given a digital image I of size $N \times N$, the first algorithm (Algorithm 1, in Sec. 5.2) relies on the DRT graph of I, and scans it entirely in the worst case. Consequently, both space and time complexities of this algorithm are $O(N^9)$, due to the space complexity of the DRT graph (Property 10).

The second algorithm (Algorithm 3, in Sec. 6.2) relies on (*i*) a LUT P of topology-preserving 20-neighbourhood images; and (*ii*) the verification of the compliance of I with P for any point of I. The generation of P (Algorithm 2, in Sec. 6.2), for a given value space $\mathbb{V}$ and a given adjacency, has a time complexity $O(5^9\ |\mathbb{V}|^{20}) = O(|\mathbb{V}|^{20})$, since the complexity for generating the DRT graph for an image defined on a 20-neighbourhood is $O(5^9)$ [7], and any image $I : N_{20}(\mathbf{0}) \to \mathbb{V}$ has to be processed via its DRT graph. Note however that this process has to be carried out only once, if P – that has a space complexity of $O(|\mathbb{V}|^{20})$ – is stored. The topological invariance verification (Algorithm 3) then presents a quasi-linear time complexity with respect to the size $N \times N$ of image $O(N^2\ \log_2(|P|)) = O(N^2\ |\mathbb{V}|)$, since the LUT P can be ordered and processed as a tree structure. One may notice that this algorithm can trivially be parallelized, leading in particular to a constant time complexity $O(|\mathbb{V}|)$, when processed as N^2 subtasks.



(a) P in $(4,8)$-adjacency (samples).



(b) P in $(8,4)$-adjacency (samples).

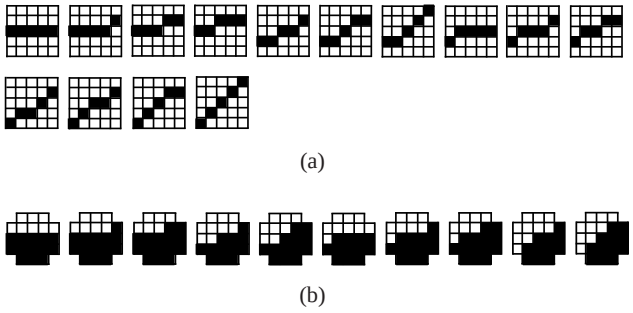
Fig. 11 Some samples defined on 20-neighbourhoods which are topology preserving, computed from Algorithm 2, in the case of $(4,8)$-adjacency (a), and $(8,4)$-adjacency (b). The foreground pixels are depicted in black, while the background pixels ($\perp$) are depicted in white.

7.2 Computational and space cost: The binary case

In this section, we experimentally assess the actual cost of the algorithm, previously discussed from a theoretical point of view. To this end, we consider the case of binary images, *i.e.*, images defined on a set of values $\mathbb{V}$ such that $|\mathbb{V}| = 2$.

Let C be the set of all the binary images defined on $N_{20}(\mathbf{0})$, that is used to build P via Algorithm 2. We have, in particular, $|C| = 2^{20}$. However, from Remark 11, we only have to consider the images such that at least one point in the 4-neighbourhood of $\mathbf{0}$ has a distinct (binary) value from the one of $\mathbf{0}$. By using this fact, plus considerations related to invariance up to rotations and symmetries, the set C can be reduced, without loss of completeness to a subset $C' \subset C$ such that $|C'| = 124\ 260 \ll |C|$.

Using Algorithm 2 on this set C', we obtain some sets P of 10 643 and 19 446 elements, in the $(4,8)$- and $(8,4)$-adjacency, respectively. Fig. 11 provides some samples of P in both cases.



(a)

(b)

Fig. 12 (a) The 14 samples of digital lines of length 5, and (b) the half-planes generated by these lines in a 20-neighbourhood.

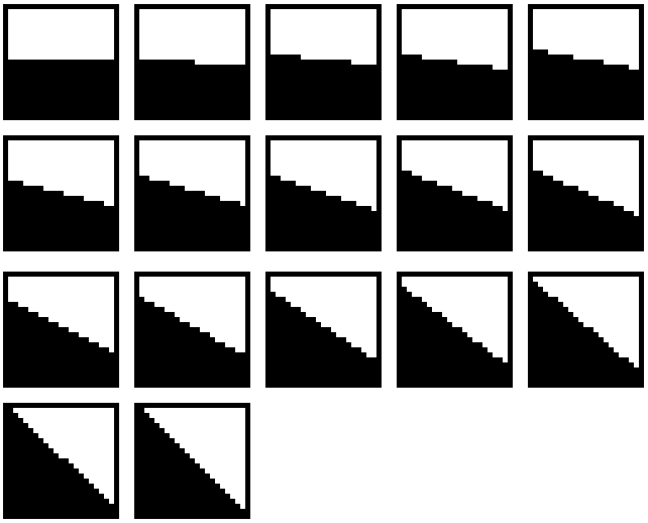
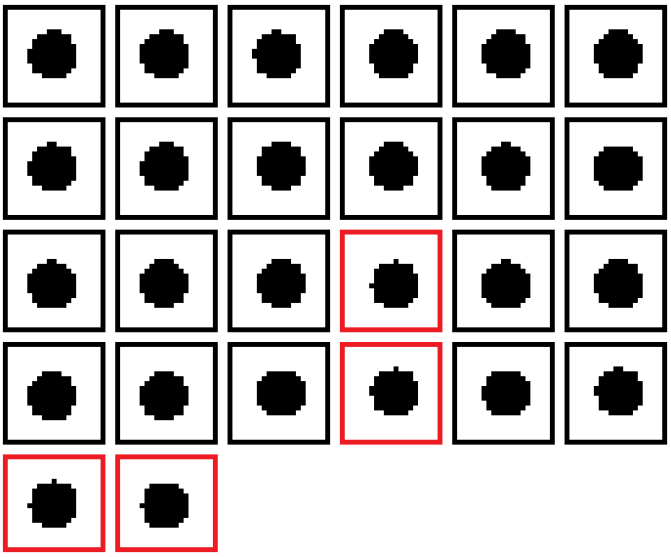
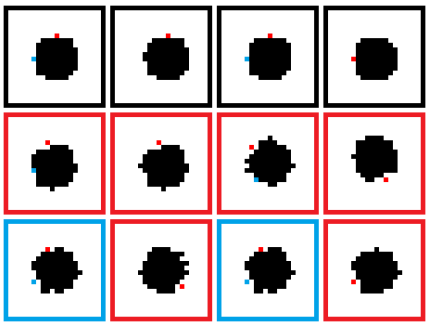


Fig. 13 Some examples of half-planes rigidly transformed from an image of size 20×20. All transformed half-planes images are simple-equivalent.



(a)



(b)

Fig. 14 (a) Some disks of radius 5, generated in a image of size 20×20. Some of them are topologically invariant (in black frames), while the others (in red frame) have been characterised as not topologically invariant by Algorithm 3. (b) Concerning the four non topologically invariant disks in (a), the pixels detected by our algorithm are those that alter the topology of the four disks. The frame of the picture that surrounds them is coloured in red or blue according to the colour of the pixel for which the topology changes.

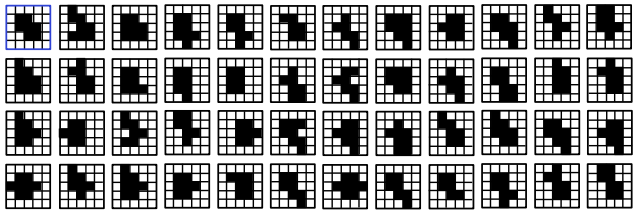
7.3 Experiments: The binary case

We now propose some experiments to illustrate the behaviour of the algorithms on images representing different kinds of objects using the $(4, 8)$-adjacent relations. We first consider basic geometric primitives, namely half-planes and disks, the evolution of which is (in theory) predictable with respect to rigid transformations. Then, we consider more generally, arbitrary shapes, whose topological invariance is not easily predictable.

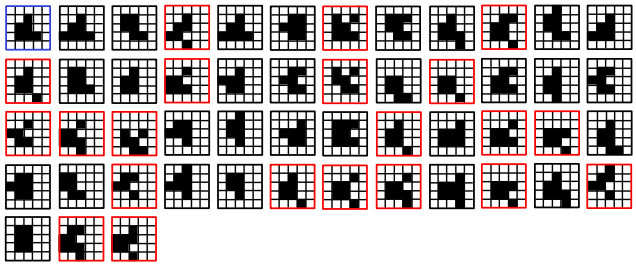
7.3.1 Topological (in)variance of geometric primitives

We define a discrete half-plane as the set of all discrete points on one side of a digital straight line. The number of digital line segments was studied in [48]. In particular, it is known that there exist 14 digital segments of length 5 inside a pattern of size 5×5 (see Fig. 12(a)). From this knowledge, we can generate all the possible half-planes in a 20-neighbourhood, as illustrated in Fig. 12(b). By using Algorithm 2 to study the properties of these patterns, we find that all of them are topologically invariant. Therefore, we can conclude that any discrete half-plane preserves homotopy-type during digital/discrete rigid transformations. Some examples of rigidly transformed half-planes are illustrated in Fig. 13.

The digital disks, defined on $\mathbb{Z}^2$ and studied, *e.g.*, in [49], can be defined as the sets of all discrete points lying inside a real disc (defined on $\mathbb{R}^2$). It is plain that the digitization of a disc depends on its size (*i.e.*, its radius) but also on its position (*i.e.*, the position of its centre) with respect to the discrete grid. Some examples of digital disks with the same radius are shown in Fig. 14. In the continuous domain, the real disks are – of course – topologically invariant under rigid transformations. In contrast to the half-planes, this



(a)



(b)

Fig. 15 (a) A topologically invariant 5×5 image (in blue frame), and (some of) its transformations (in black frames). (b) A topologically-variant 5×5 image (in blue frame), and (some of) its simple-equivalent (in black frames) and non-simple-equivalent transformations (in red frames).

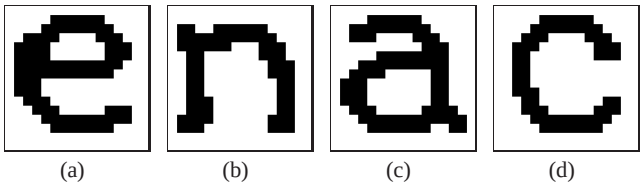
property is lost in the digital case. Indeed, Algorithm 3, performed on the images of Fig. 14, detects that some of them are not topologically invariant. This emphasizes the influence of the position of the disk center for this property. It also sheds light on the influence of the differential properties of the object boundaries – and in particular their curvature – on the potential preservation of image topology. From a methodological point of view, it can motivate the use of image simplification procedures that decompose boundaries into discrete line segments, since such approach may present more desirable topological properties.

7.3.2 Topological (in)variance of arbitrary shapes

To complete these experiments, we finally exhibit some examples of arbitrary binary images that have been characterised by Algorithm 3 as being topologically invariant (Figs. 15(a) and 16), or not (Figs. 15(b) and 17).

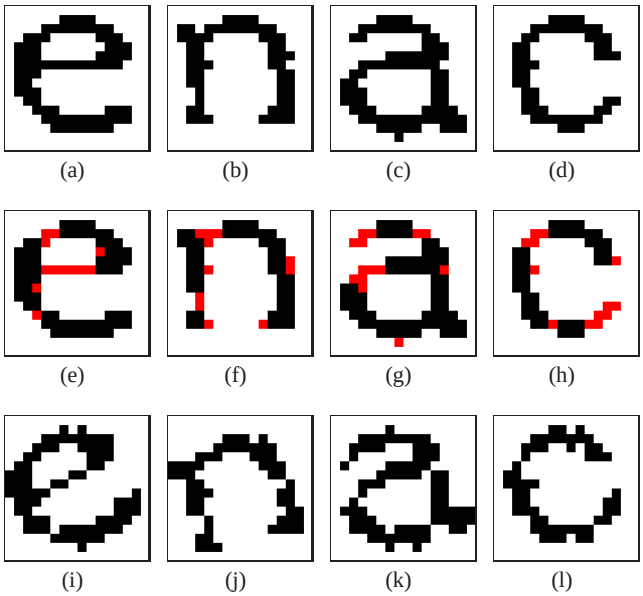
8 Conclusion

In this article, we have considered geometrical and topological concepts, to propose an approach for studying the topological behaviour of rigid transformations in $\mathbb{Z}^2$. More precisely, we combined the notion of simple point with the notion of DRT graph, leading to algorithmic processes that can characterise the topological invariance of digital images under *all* rigid transformations. In particular, by taking advantage of the respective strengths of both notions, it has been possible to develop an efficient algorithm, able to evaluate this topological invariance in a quasi-linear time with respect to the image size.



(a) (b) (c) (d)

Fig. 16 (a–d) Examples of topologically invariant character images.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 17 (a–d) Examples of non topologically invariant character images. (e–h) Detection of pixels (in red) which potentially change homotopy-type of (a–d) during DRTs. (i–l) Non-simple-equivalent transformed images of (a–d).

Beyond its algorithmic aspects, this work may contribute to the better understanding of the relationships between geometry and topology in the framework of digital imaging, where both notions are less strongly linked than in the continuous space. In particular, the proposed algorithm may provide an efficient tool for further studying the notion of regularity [50–52], that is currently used to assess the preservation of topological properties during the digitization of an image from $\mathbb{R}^2$ to $\mathbb{Z}^2$. In particular, a *discrete* notion of regularity may be derived from the continuous one, in order to assess the topological behaviour of image transformations in a fully discrete framework.

In this article, we have considered the specific case of the Eulerian transformation model (see Sec. 3.2). Further work may consider the case of the Lagrangian transformation model. In the context of topological alterations induced by rigid transformations of digital images, this latter model comes with some additional difficulties. Indeed, while in the Eulerian model, a double (resp. null) point transfers its

value to two (resp. no) point(s) in the transformed image (Sec. 4.2), in the Lagrangian case, a double (resp. null) point in the transformed image will receive two (resp. no) values; this leads to a result that is both incomplete and ambiguous. In order to deal with these supplementary issues, it may be necessary to study more deeply the relations that exist between the digital images, defined on $\mathbb{Z}^2$, and the continuous ones, defined on $\mathbb{R}^2$, as they are linked via the digitization processes.

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 03).

References

1. B. Zitová, J. Flusser, Image registration methods: A survey, Image and Vision Computing 21 (11) (2003) 977–1000.
2. A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Computing Surveys 38 (4) (2006) 1–45.
3. V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstaedter, K. Briggman, W. Denk, J. Bowden, J. Mendenhall, W. Abraham, K. Harris, N. Kasthuri, K. Hayworth, R. Schalek, J. Tapia, J. Lichtman, S. Seung, Boundary learning by optimization with topological constraints, in: CVPR, Proceedings, IEEE, 2010, pp. 2488–2495.
4. S. Faisan, N. Passat, N. Noblet, R. Chabrier, C. Meyer, Topology preserving warping of 3-D binary images according to continuous one-to-one mappings, IEEE Transactions on Image Processing 20 (8) (2011) 2135–2145.
5. B. Dawant, S. Hartmann, J. Thirion, F. Maes, D. Vandermeulen, P. Demaerel, Automatic 3-D segmentation of internal structures of the head in MR images using a combination of similarity and free-form deformations: Part I, methodology and validation on normal subjects, IEEE Transactions on Medical Imaging 18 (10) (1999) 902–916.
6. P. Ngo, Y. Kenmochi, N. Passat, H. Talbot, Sufficient conditions for topological invariance of 2D digital images under rigid transformations, in: DGCI, Proceedings, Vol. 7749 of Lecture Notes in Computer Science, Springer, 2013, pp. 155–168.
7. P. Ngo, Y. Kenmochi, N. Passat, H. Talbot, Combinatorial structure of rigid transformations in 2D digital images, Computer Vision and Image Understanding 117 (4) (2013) 393–408.
8. M.-A. Jacob, E. Andres, On discrete rotations, in: DGCI, Proceedings, 1995, pp. 161–174.
9. A. Amir, O. Kapah, D. Tsur, Faster two-dimensional pattern matching with rotations, Theoretical Computer Science 368 (3) (2006) 196–204.
10. A. Amir, G. M. Landau, U. Vishkin, Efficient pattern matching with scaling, Journal of Algorithms 13 (1) (1992) 2–32.
11. A. Amir, A. Butman, M. Lewenstein, E. Porat, Real two dimensional scaled matching, Algorithmica 53 (3) (2009) 314–336.
12. 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.
13. C. Hundt, M. Liśkiewicz, On the complexity of affine image matching, in: STACS, Proceedings, Vol. 4393 of Lecture Notes in Computer Science, Springer, 2007, pp. 284–295.
14. C. Hundt, Affine image matching is uniform TC^0-complete, in: CPM, Proceedings, Vol. 6129 of Lecture Notes in Computer Science, Springer, 2010, pp. 13–25.
15. C. Hundt, M. Liśkiewicz, Combinatorial bounds and algorithmic aspects of image matching under projective transformations, in: MFCS, Proceedings, Vol. 5162 of Lecture Notes in Computer Science, Springer, 2008, pp. 395–406.
16. J.-P. Reveillès, Géométrie discrète, calcul en nombres entiers et algorithmique, Thèse d'ï¿L'tat, Université Strasbourg 1 (1991).
17. E. Andres, The quasi-shear rotation, in: DGCI, Proceedings, Vol. 1176 of Lecture Notes in Computer Science, Springer, 1996, pp. 307–314.
18. M. S. Richman, Understanding discrete rotations, in: ICASSP, Proceedings, Vol. 3, IEEE, 1997, pp. 2057–2060.
19. B. Nouvel, Rotations discrètes et automates cellulaires, Ph.D. thesis, École Normale Supérieure de Lyon (2006).
20. B. Nouvel, E. Rémila, Incremental and transitive discrete rotations, in: IWCIA, Proceedings, Vol. 4040 of Lecture Notes in Computer Science, Springer, 2006, pp. 199–213.
21. Y. Thibault, Y. Kenmochi, A. Sugimoto, Computing upper and lower bounds of rotation angles from digital images, Pattern Recognition 42 (8) (2009) 1708–1717.
22. G. Bertrand, On critical kernels, Comptes Rendus de l'Académie des Sciences—Série Mathématiques I (345) (2007) 363–367.
23. A. Rosenfeld, Connectivity in digital pictures, Journal of the ACM 17 (1) (1970) 146–160.
24. T. Y. Kong, A. Rosenfeld, Digital topology: Introduction and survey, Computer Vision Graphics & Image Processing 48 (3) (1989) 357–393.
25. L. Mazo, N. Passat, M. Couprie, C. Ronse, Paths, homotopy and reduction in digital images, Acta Applicandae Mathematicae 113 (2) (2011) 167–193.
26. L. Mazo, N. Passat, M. Couprie, C. Ronse, Digital imaging: A unified topological framework, Journal of Mathematical Imaging and Vision 44 (1) (2012) 19–37.
27. E. Khalimsky, Topological structures in computer science, Journal of Applied Mathematics and Simulation 1 (1) (1987) 25–40.
28. V. A. Kovalevsky, Finite topology as applied to image analysis, Computer Vision, Graphics & Image Processing 46 (2) (1989) 141–161.
29. G. Bertrand, G. Malandain, A new characterization of three-dimensional simple points, Pattern Recognition Letters 15 (2) (1994) 169–175.
30. M. Couprie, G. Bertrand, New characterizations of simple points in 2D, 3D, and 4D discrete spaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (4) (2009) 637–648.
31. C. Ronse, A topological characterization of thinning, Theoretical Computer Science 43 (1) (2007) 31–41.
32. G. Bertrand, On P-simple points, Comptes Rendus de l'Académie des Sciences—Série Mathématiques I (321) (1995) 1077–1084.
33. N. Passat, L. Mazo, An introduction to simple sets, Pattern Recognition Letters 30 (15) (2009) 1366–1377.
34. M. Couprie, F. N. Bezerra, G. Bertrand, Topological operators for grayscale image processing, Journal of Electronic Imaging 10 (4) (2001) 1003–1015.
35. L. J. Latecki, Multicolor well-composed pictures, Pattern Recognition Letters 16 (4) (1997) 425–431.
36. G. Damiand, A. Dupas, J.-O. Lachaud, Fully deformable 3D digital partition model with topological control, Pattern Recognition Letters 32 (9) (2011) 1374–1383.
37. L. Mazo, N. Passat, M. Couprie, C. Ronse, Topology on digital label images, Journal of Mathematical Imaging and Vision 44 (3) (2012) 254–281.
38. D. Pham, P.-L. Bazin, J. Prince, Digital topology in brain imaging, IEEE Signal Processing Magazine 27 (4) (2010) 51–59.

39. J.-F. Mangin, V. Frouin, I. Bloch, J. Régis, J. López-Krahe, From 3D magnetic resonance images to structural representations of the cortex topography using topology preserving deformations, Journal of Mathematical Imaging and Vision 5 (4) (1995) 297–318.
40. X. Han, C. Xu, J. L. Prince, A topology preserving level set method for geometric deformable models, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (6) (2003) 755–768.
41. P.-L. Bazin, L. M. Ellingsen, D. L. Pham, Digital homeomorphisms in deformable registration, in: IPMI, Proceedings, Vol. 4584 of Lecture Notes in Computer Science, Springer, 2007, pp. 211–222.
42. R. Ayala, E. Domínguez, A. R. Francés, A. Quintero, Homotopy in digital spaces, Discrete Applied Mathematics 125 (1) (2003) 3–24.
43. G. Bertrand, M. Couprie, N. Passat, A note on 3-D simple points and simple-equivalence, Information Processing Letters 109 (13) (2009) 700–704.
44. B. Nouvel, E. Rémila, Configurations induced by discrete rotations: Periodicity and quasi-periodicity properties, Discrete Applied Mathematics 147 (2–3) (2005) 325–343.
45. Y. Thibault, Rotations in 2D and 3D discrete spaces, Ph.D. thesis, Université Paris-Est (2010).
46. L. J. Latecki, U. Eckhardt, A. Rosenfeld, Well-composed sets, Computer Vision and Image Understanding 61 (1) (1995) 70–83.
47. L. Mazo, A framework for label images, in: CTIC, Proceedings, Vol. 7309 of Lecture Notes in Computer Science, Springer, 2012, pp. 1–10.
48. C. Berenstein, D. Lavine, On the number of digital straight line segments, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (6) (1988) 880–887.
49. B. Nagy, An algorithm to find the number of the digitizations of discs with a fixed radius, Electronic Notes in Discrete Mathematics 20 (2005) 607–622.
50. J. Serra, Image Analysis and Mathematical Morphology, Academic Press, Inc., 1983.
51. H. J. A. M. Heijmans, Discretization of morphological operators, Journal of Visual Communication and Image Representation 3 (2) (1992) 182–193.
52. L. J. Latecki, C. Conrad, A. Gross, Preserving topology by a digitization process, Journal of Mathematical Imaging and Vision 8 (2) (1998) 131–159.

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

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

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*

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

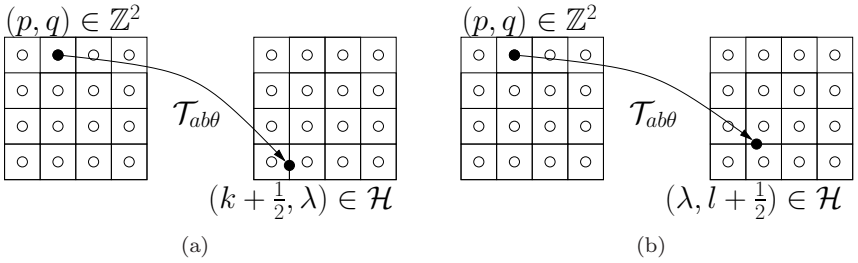


Fig. 1 Examples of critical transformations $\mathcal{T}_{ab\theta}$, each of which maps at least one integer-coordinate point onto a "vertical" (a) or "horizontal" (b) half-grid point. The integer-coordinate points in $\mathbb{Z}^2$ are depicted by dots, while the half-grid points are depicted by lines.

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:

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

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

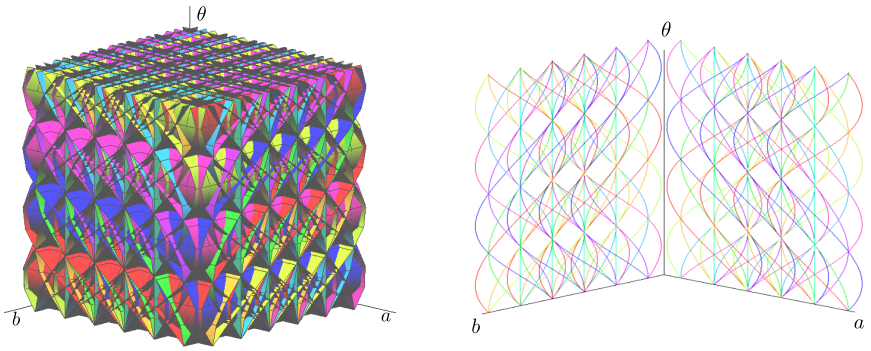
The 2D surfaces Φ_{pqk} (resp. Ψ_{pql}) defined in the parameter space (a,b,θ) are called *tipping surfaces* [18]. Their respective cross-sections ϕ_{pqk} (resp. ψ_{pql}) on the 2D plane (a,θ) (resp. (b,θ)) are called *tipping curves*. These tipping surfaces/curves, which correspond to the discontinuities 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:*

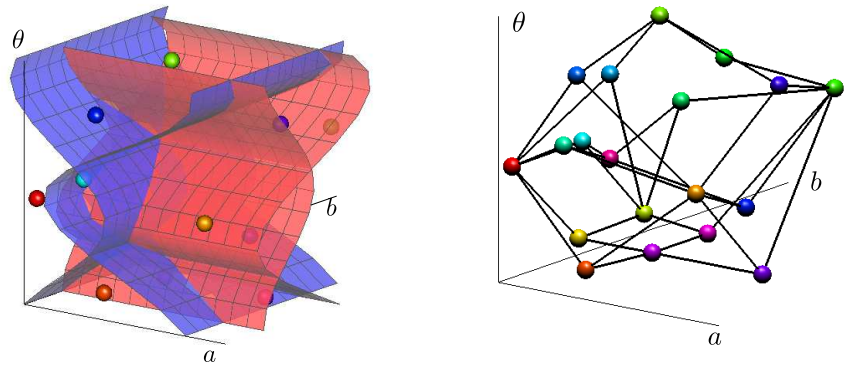
- *each vertex $v \in V$ models a 3D open cell associated to a DRT;*
- *each labelled edge $e = (u, w, f) \in E$ (where f is either Φ_{pqk} or Ψ_{pql}) models the tipping surface f between two adjacent vertices $v, w \in V$.*

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

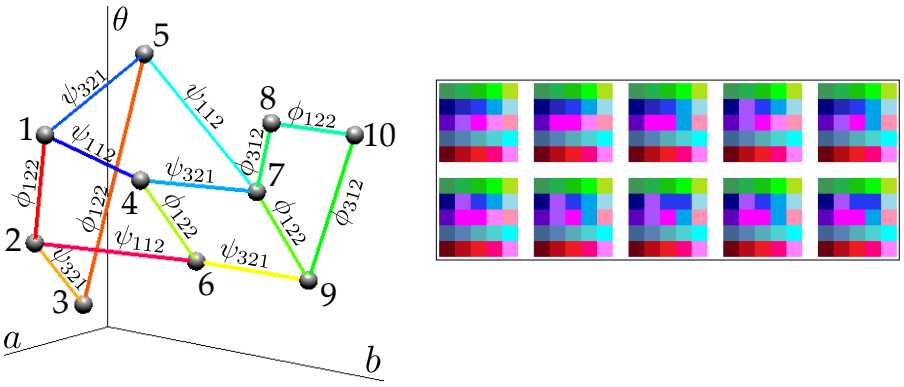


Fig. 4 Left: a part of a DRT graph in which a vertex/DRT represents a digital transformed image and an edge $e = (v, w, f)$ between two vertices v and w indicates that one pixel value is different between the associated transformed images I_v and I_w (see text). Right: the transformed images associated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

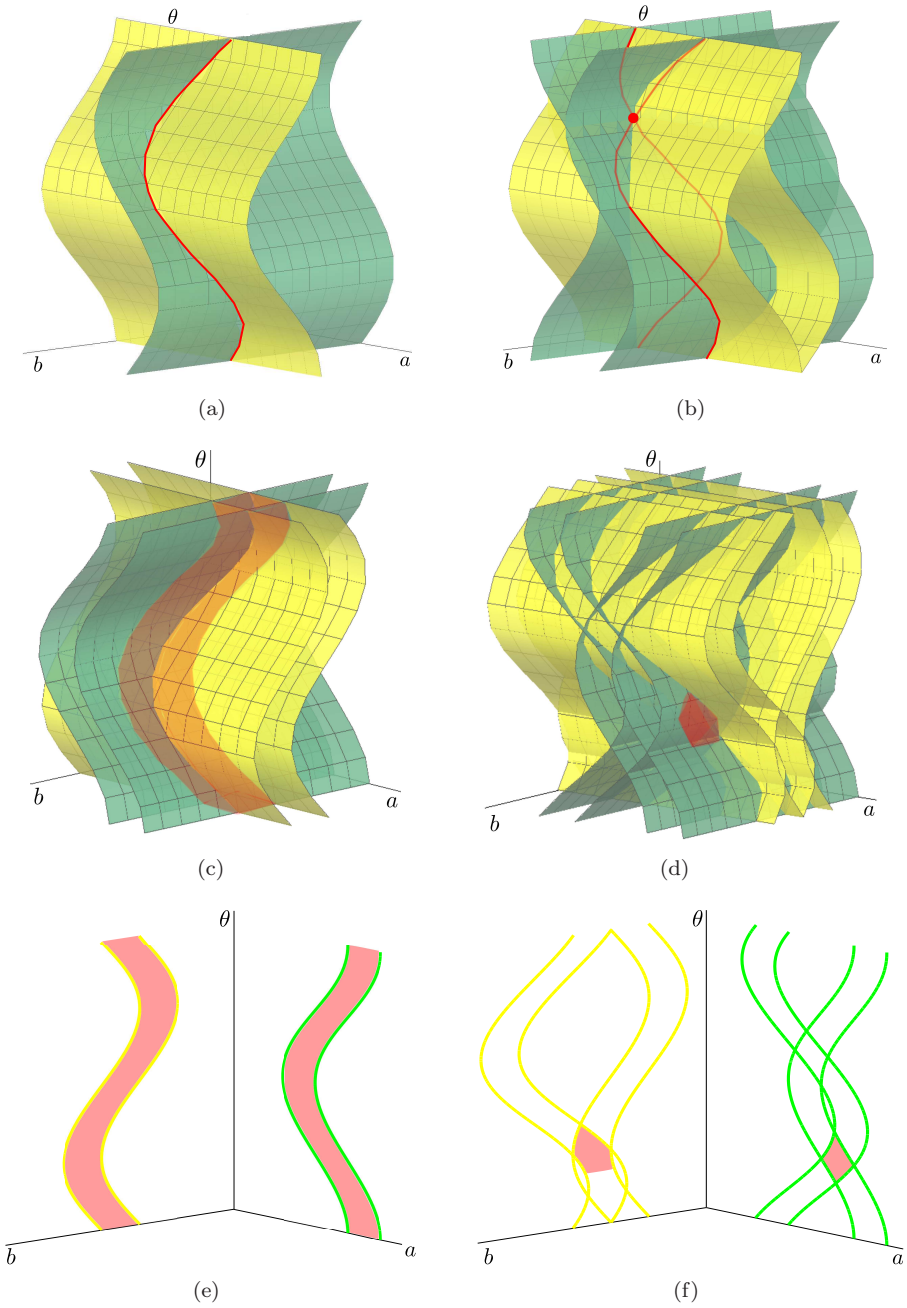


Fig. 5 Feasible rigid transformations induced by geometric constraints in the continuous (a,b) and discrete (c–f) frameworks. (a) Transformations with one point correspondence (red curve). (b) Transformation with two point correspondences (red dot at the intersection of the two red curves). (c) Transformations 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

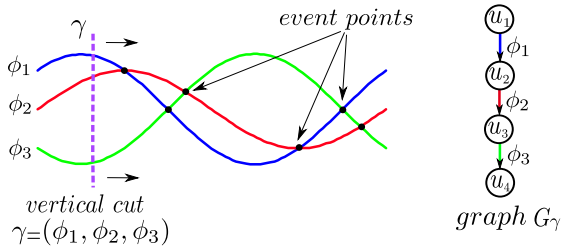


Fig. 6 Example of a cut γ and its associated graph G_γ.

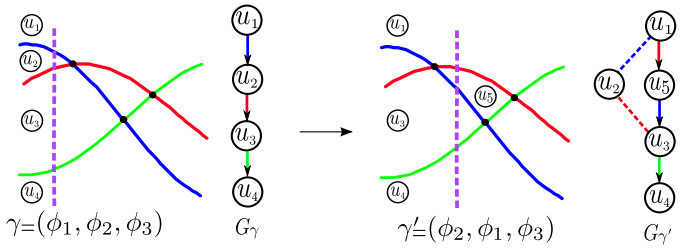


Fig. 7 Progress of the cut at an event point by which the cut is updated and the corresponding graph is modified.

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.

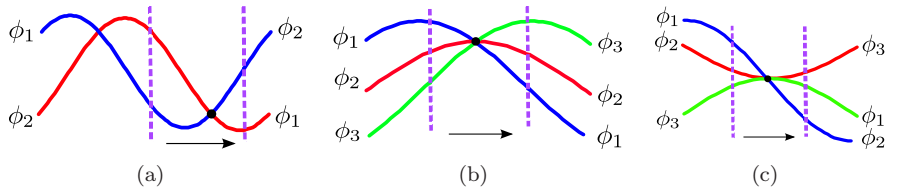


Fig. 8 Examples of a *simple* event point generated by only two tipping curves (a), and *degeneracies* (b,c) generated by more than two tipping curves.

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.

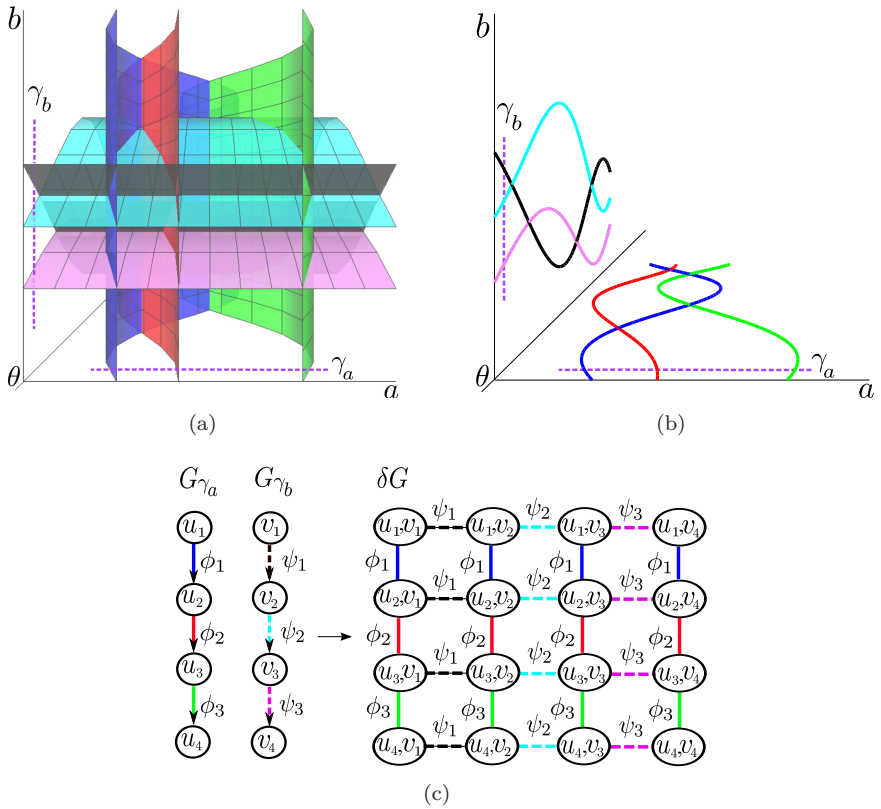


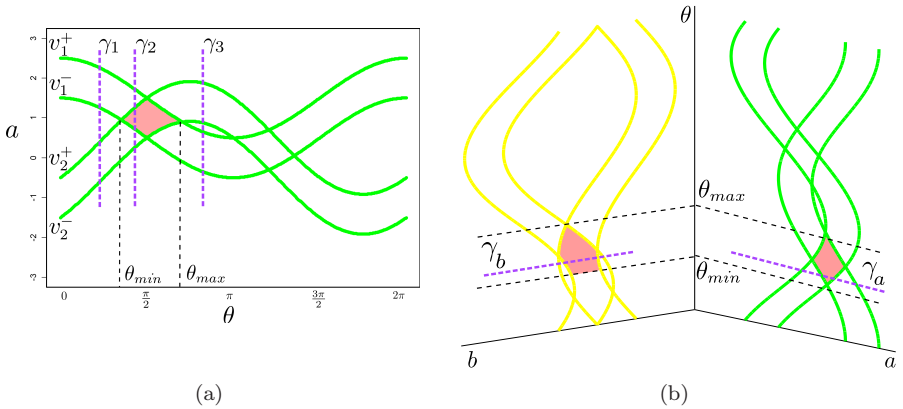
Fig. 9 Generation of a partial graph δG from two graphs G_{γ_a} and G_{γ_b} associated to cuts γ_a and γ_b respectively.

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

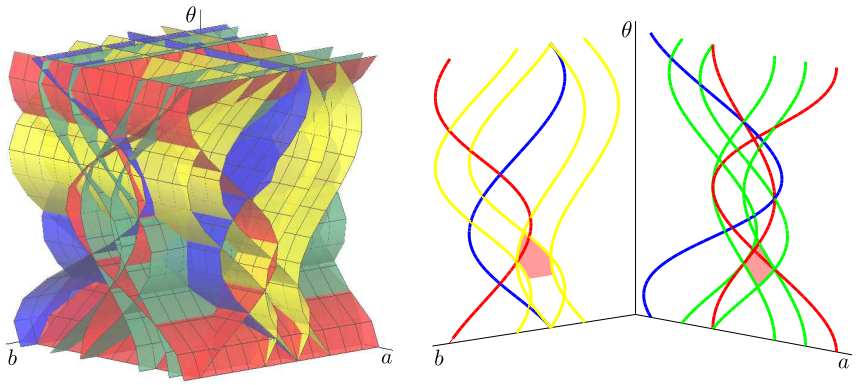
In order to compute an FDRT graph modeling the subdivision of an FRTS $\mathcal{R}$ into DRTs, we require not only the tipping surfaces constituting the boundary of $\mathcal{R}$ but also those passing through $\mathcal{R}$. In this section, we explain how to find these tipping surfaces.

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.

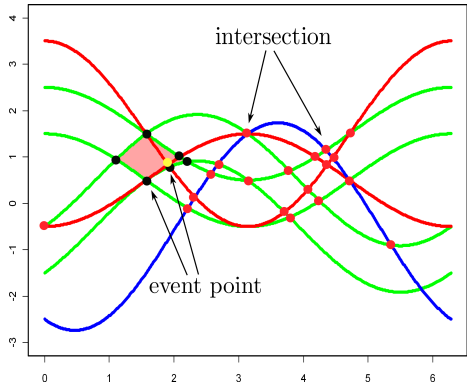


Fig. 12 Examples of boundary event points (in black), interior event points (in yellow) and intersections of tipping curves but not event points (in red) for computing a FRTS graph.

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.

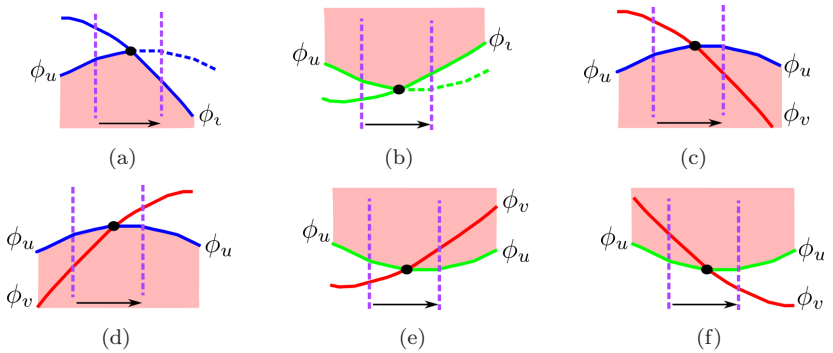


Fig. 13 Classification of simple boundary event points. An event point generated by two tipping curves ϕ_u and ϕ_v: (a,b) changes the boundary from ϕ_u to ϕ_v, (c,d) ϕ_v goes in and out by crossing the upper boundary ϕ_u, and (e,f) ϕ_v goes in and out by crossing the lower boundary ϕ_u.

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 deal 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.

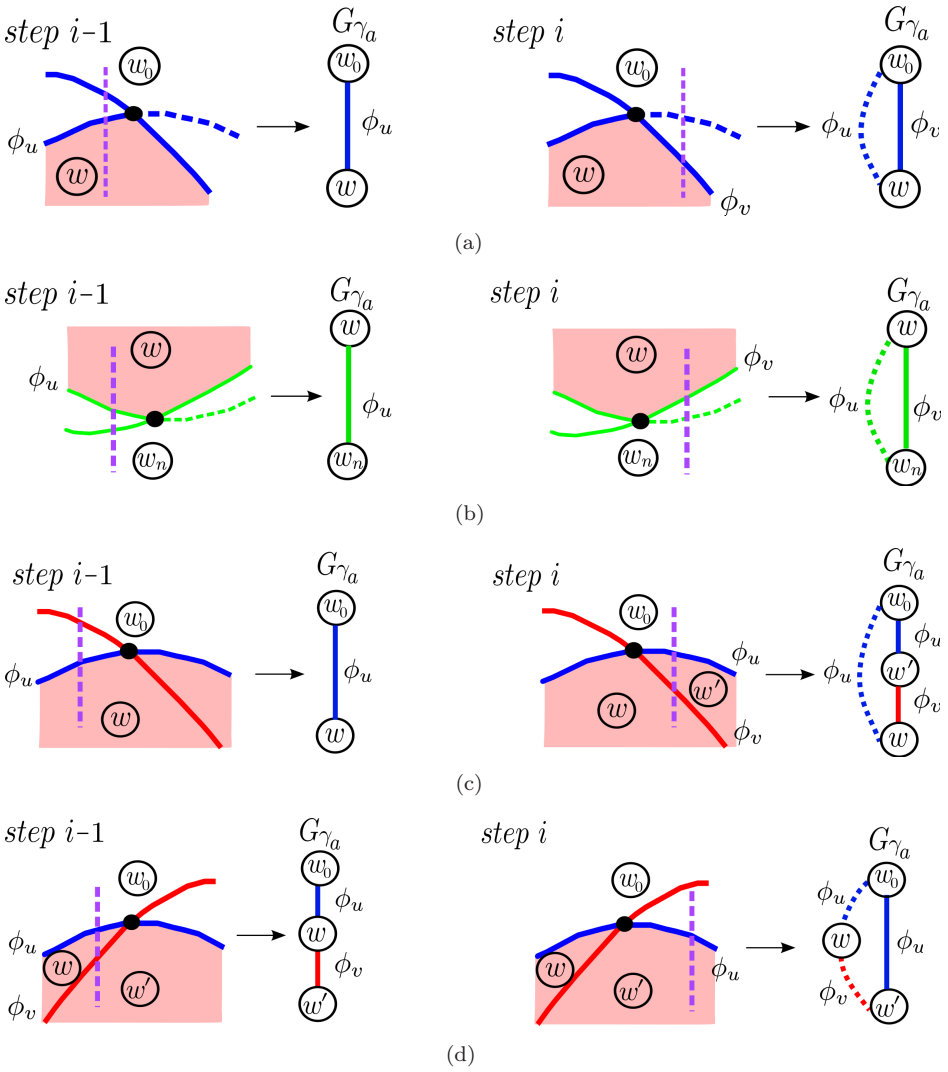


Fig. 14 Illustrations of elementary steps – update γ and generate its graph G_γ – for a tipping curve changing (a) an upper or (b) a lower boundary, and going (c) in or (d) out of an upper boundary.

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 constraint, 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

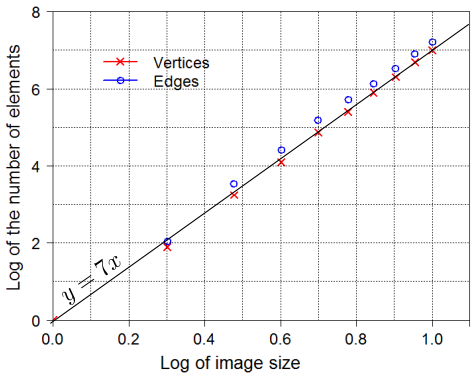


Fig. 15 Space complexities of FDRT graphs expressed as the numbers of vertices and edges in FDRT graphs under one pixel-invariance constraint, for images of size $N \times N$.

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

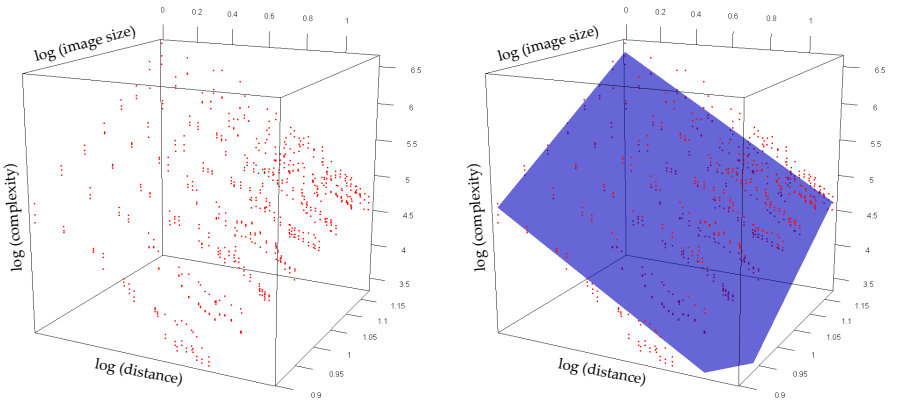


Fig. 16 Experimental space complexities of FDRT graphs in the two pixel-invariance constraints case. The complexities are measured over 676 experiments as a function of image size, varying from 8×8 to 15×15, and varying distances between two given constraints limited by the image sizes. Left: 3D log 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

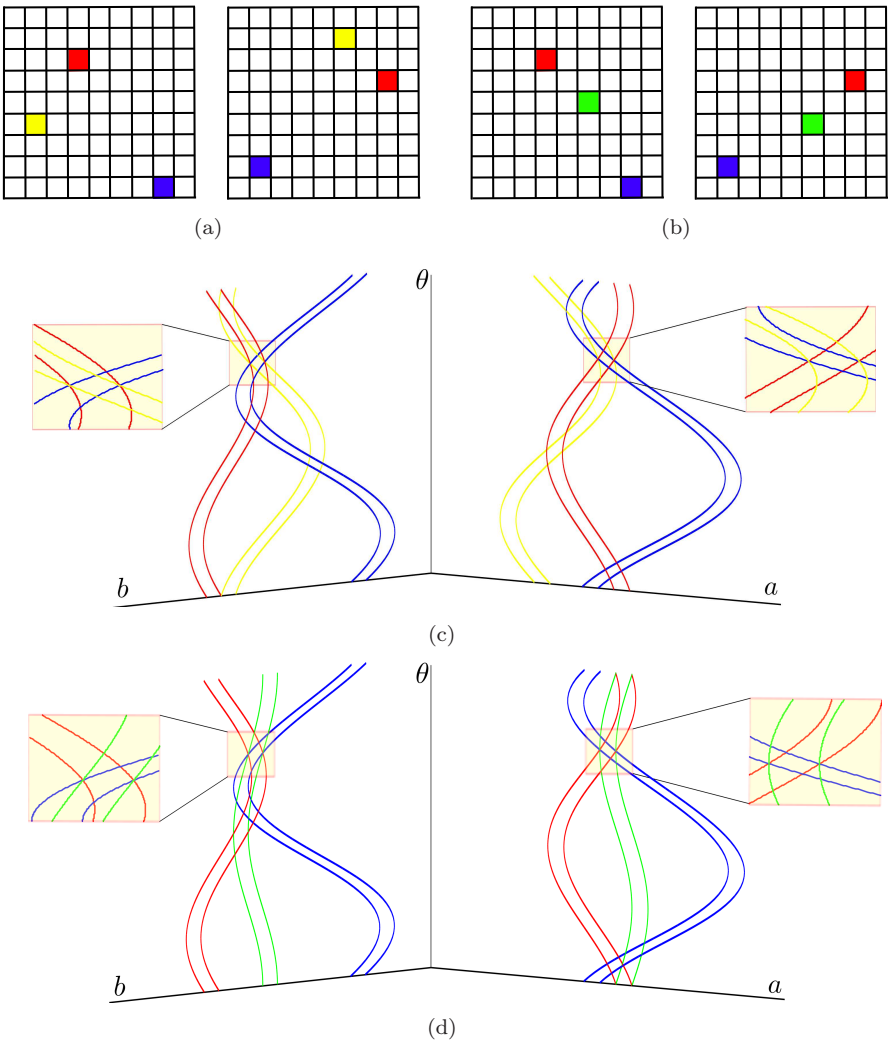


Fig. 17 For two given pixel-invariance constraints (in red and blue), there exists a supplementary pixel-invariance constraint (in yellow) that contributes to reducing the associated FRTS (a). There also exists a supplementary pixel-invariance constraint (in green) that does not (b). (c) and (d) illustrate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

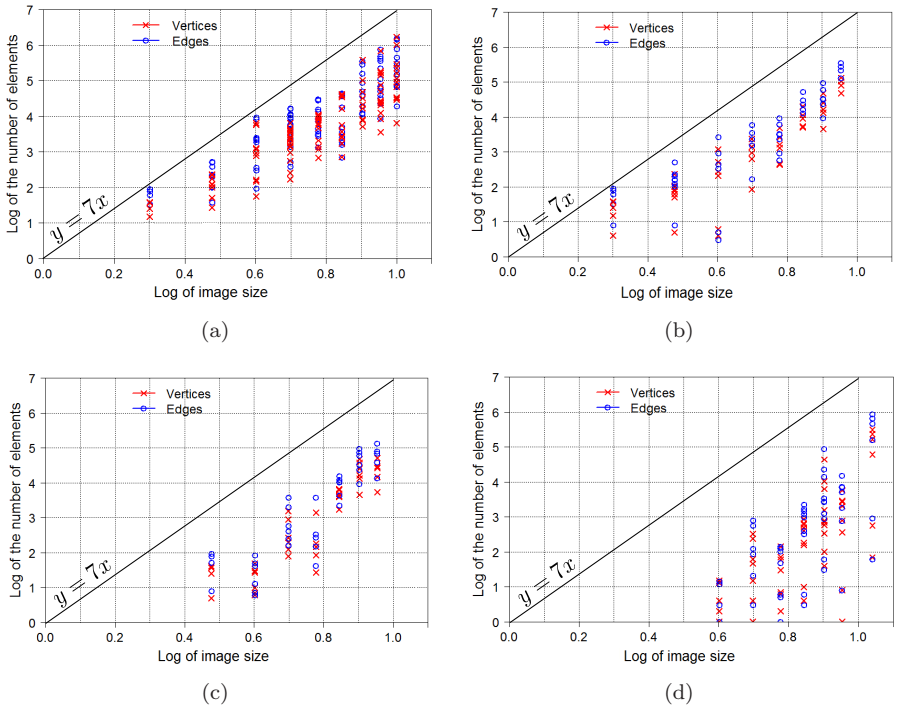


Fig. 18 Experimental results of space complexities of FDRT graphs for two (a), three (b), five (c) and ten (d) pixel-invariance constraints.

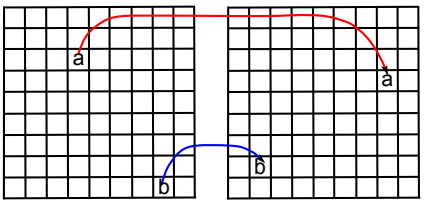
5.2 Time complexity of feasible discrete rigid transformation graph construction

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:

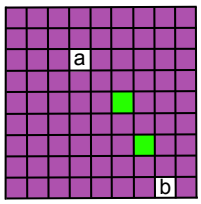
1. finding the boundary of the FRTS $\mathcal{R}$;
2. finding the event points in $\mathcal{R}$, more precisely, in the projections $\mathcal{R}_H$ and $\mathcal{R}_V$ of $\mathcal{R}$ on the planes (a, θ) and (b, θ) respectively; and
3. building the FDRT graph G associated to $\mathcal{R}$.

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 $\mathcal{O}(N^6)$.

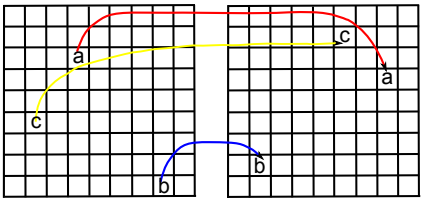
In Step 2, the process of detecting whether a tipping surface passes through $\mathcal{R}$ can be done in linear time with respect to the number of boundary segments



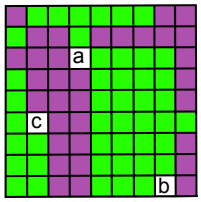
(a) Two pixel-invariance constraints.



(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).