

# Transformations rigides discrètes

Théorie et applications en traitement d'images numériques

Phuc NGO

CEA – Saclay, DIGITEO Labs



Collaboration avec Yukiko KENMOCHI, Nicolas PASSAT et Hugues TALBOT

19 Mars 2014

# Plan de la présentation

1. Introduction
2. Transformations rigides sur  $\mathbb{Z}^2$
3. Analyse combinatoire des transformations rigides discrètes
4. Applications : Recalage d'images et analyse topologique
5. Conclusion

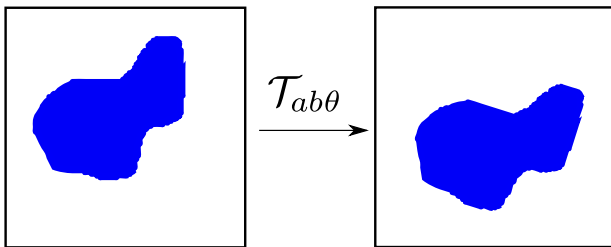
# Transformation rigide

## Définition

**Une transformation rigide** est une fonction  $\mathcal{T}_{ab\theta} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$

$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} p \cos \theta - q \sin \theta + a \\ p \sin \theta + q \cos \theta + b \end{pmatrix}$$

où  $a, b \in \mathbb{R}$ ,  $\theta \in [0, 2\pi[$  et  $(p, q), (p', q') \in \mathbb{R}^2$ .



# Transformation rigide

## Définition

**Une transformation rigide** est une fonction  $\mathcal{T}_{ab\theta} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$

$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} p \cos \theta - q \sin \theta + a \\ p \sin \theta + q \cos \theta + b \end{pmatrix}$$

où  $a, b \in \mathbb{R}$ ,  $\theta \in [0, 2\pi[$  et  $(p, q), (p', q') \in \mathbb{R}^2$ .

## Contexte

- Ces transformations sont définies dans l'espace continu, nécessitant un procédé de digitalisation lors de l'application sur les images numériques,
- cela provoque parfois des artefacts indésirables dans l'image transformée.



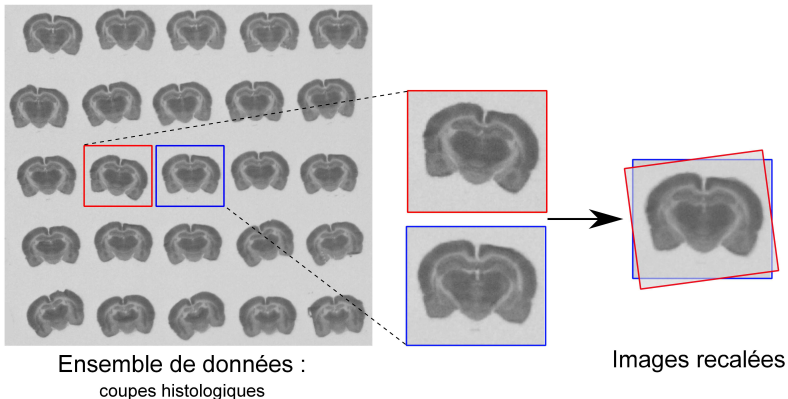
# Exemples de transformations rigides sur $\mathbb{Z}^2$ : recalage d'images



Ensemble de données :  
coupes histologiques

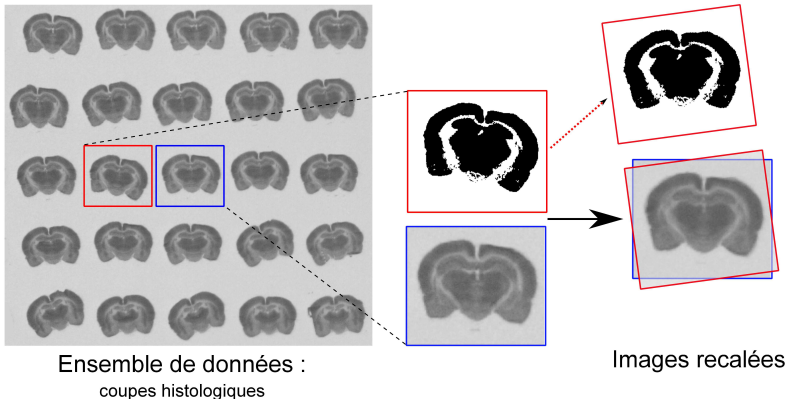
(Laboratoire ICube - Strasbourg)

# Exemples de transformations rigides sur $\mathbb{Z}^2$ : recalage d'images



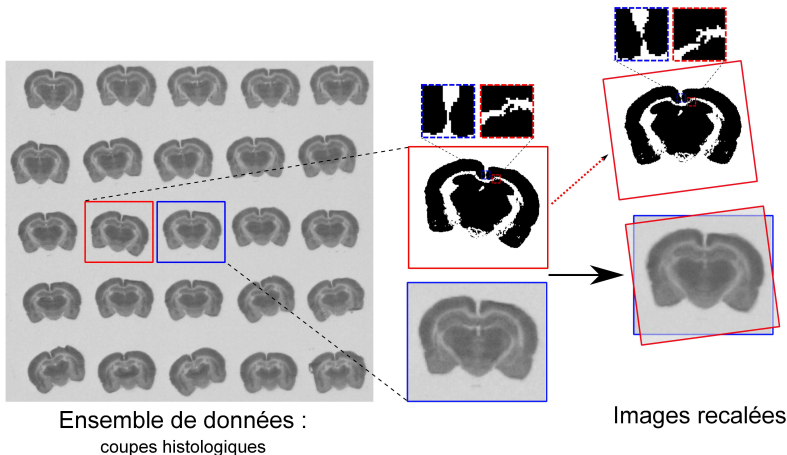
(Laboratoire ICube - Strasbourg)

# Exemples de transformations rigides sur $\mathbb{Z}^2$ : recalage d'images



(Laboratoire ICube - Strasbourg)

# Exemples de transformations rigides sur $\mathbb{Z}^2$ : recalage d'images



(Laboratoire ICube - Strasbourg)

# Motivation

- Ces transformations peuvent-elles être gérées dans un cadre purement discret ?

# Motivation

- Ces transformations peuvent-elles être gérées dans un cadre purement discret ?
- Combien d'images transformées existe-t-il dans un sous-ensemble fini de  $\mathbb{Z}^2$  ?

# Motivation

- Ces transformations peuvent-elles être gérées dans un cadre purement discret ?
- Combien d'images transformées existe-t-il dans un sous-ensemble fini de  $\mathbb{Z}^2$  ?
- Comment générer toutes ces images ?

# Motivation

- Ces transformations peuvent-elles être gérées dans un cadre purement discret ?
- Combien d'images transformées existe-t-il dans un sous-ensemble fini de  $\mathbb{Z}^2$  ?
- Comment générer toutes ces images ?

## Contributions

- Proposition d'une version discrète des transformations rigides pour les images numériques de taille  $N \times N$ .



## Motivation

- Ces transformations peuvent-elles être gérées dans un cadre purement discret ?
- Combien d'images transformées existe-t-il dans un sous-ensemble fini de  $\mathbb{Z}^2$  ?
- Comment générer toutes ces images ?

### Contributions

- Proposition d'une version discrète des transformations rigides pour les images numériques de taille  $N \times N$ .
- Définition d'un graphe représentant la structure combinatoire de ces transformations discrètes.

# Motivation

- Ces transformations peuvent-elles être gérées dans un cadre purement discret ?
- Combien d'images transformées existe-t-il dans un sous-ensemble fini de  $\mathbb{Z}^2$  ?
- Comment générer toutes ces images ?

## Contributions

- Proposition d'une version discrète des transformations rigides pour les images numériques de taille  $N \times N$ .
- Définition d'un graphe représentant la structure combinatoire de ces transformations discrètes.
- Complexité de ce graphe est  $O(N^9)$  avec l'algorithme de reconstruction en temps linéaire.

## Motivation

- Ces transformations peuvent-elles être gérées dans un cadre purement discret ?
- Combien d'images transformées existe-t-il dans un sous-ensemble fini de  $\mathbb{Z}^2$  ?
- Comment générer toutes ces images ?

## Contributions

- Proposition d'une version discrète des transformations rigides pour les images numériques de taille  $N \times N$ .
- Définition d'un graphe représentant la structure combinatoire de ces transformations discrètes.
- Complexité de ce graphe est  $O(N^9)$  avec l'algorithme de reconstruction en temps linéaire.
- Applications en recalage et en analyse topologique des images.

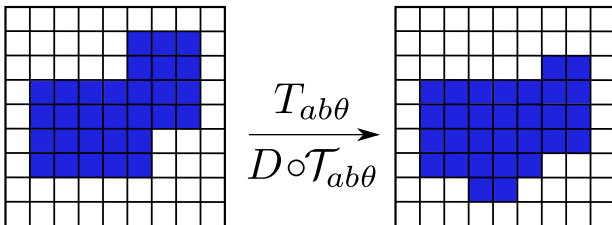
# Transformation rigide digitale

## Définition

Une **transformation rigide digitale** est une fonction  $T_{ab\theta} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$

$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p \cos \theta - q \sin \theta + a] \\ [p \sin \theta + q \cos \theta + b] \end{pmatrix}$$

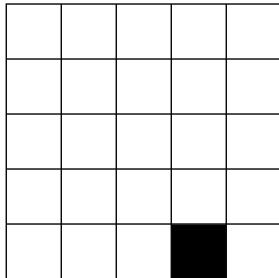
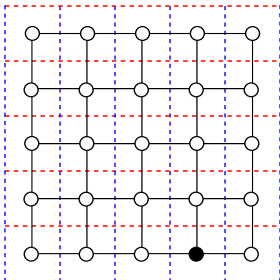
où  $a, b \in \mathbb{R}$ ,  $\theta \in [0, 2\pi[$  et  $(p, q), (p', q') \in \mathbb{Z}^2$ .



# Discontinuité des transformations rigides digitales

Transformation rigide digitale  $T_{ab\theta} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$

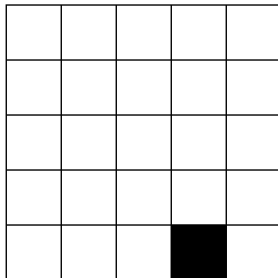
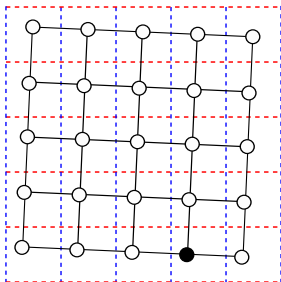
$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p \cos \theta - q \sin \theta + a] \\ [p \sin \theta + q \cos \theta + b] \end{pmatrix}$$



# Discontinuité des transformations rigides digitales

Transformation rigide digitale  $T_{ab\theta} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$

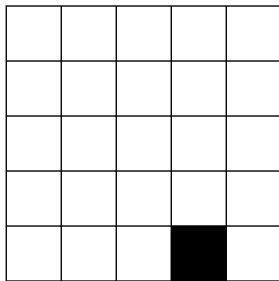
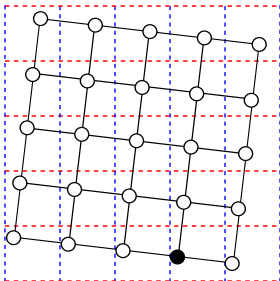
$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p \cos \theta - q \sin \theta + a] \\ [p \sin \theta + q \cos \theta + b] \end{pmatrix}$$



# Discontinuité des transformations rigides digitales

Transformation rigide digitale  $T_{ab\theta} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$

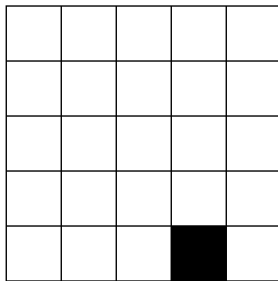
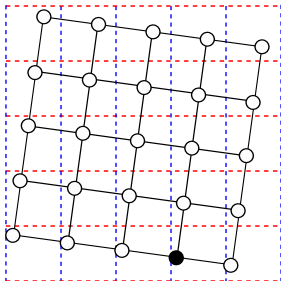
$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p \cos \theta - q \sin \theta + a] \\ [p \sin \theta + q \cos \theta + b] \end{pmatrix}$$



# Discontinuité des transformations rigides digitales

Transformation rigide digitale  $T_{ab\theta} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$

$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p \cos \theta - q \sin \theta + a] \\ [p \sin \theta + q \cos \theta + b] \end{pmatrix}$$

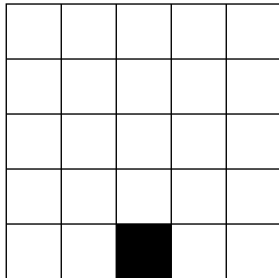
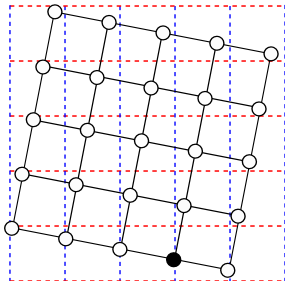




# Discontinuité des transformations rigides digitales

Transformation rigide digitale  $T_{ab\theta} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$

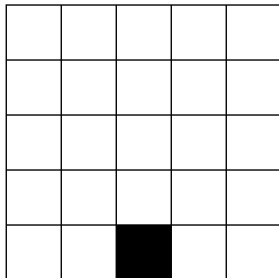
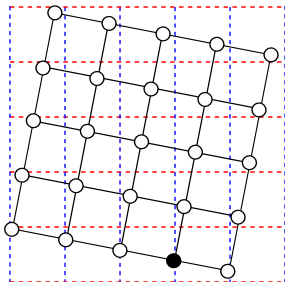
$$\begin{pmatrix} p' \\ q' \end{pmatrix} = \begin{pmatrix} [p \cos \theta - q \sin \theta + a] \\ [p \sin \theta + q \cos \theta + b] \end{pmatrix}$$



# Transformation rigide discrète

## Définition

Une **transformation rigide discrète** (TRD) est un ensemble de transformations continues équivalentes par application d'un opérateur de discrétisation.

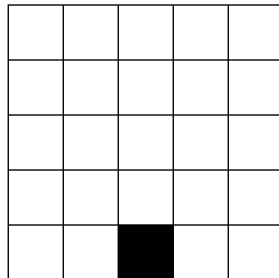
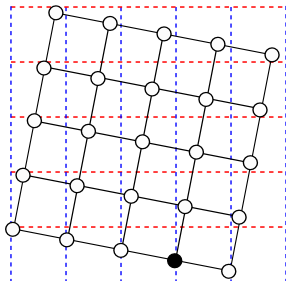


L'espace des paramètres est partitionné en des ensembles disjoints de TRD.

# Transformation critique

## Définition

Une transformation rigide est **critique** si elle projette au moins un point entier sur la semi-grille verticale ou horizontale.

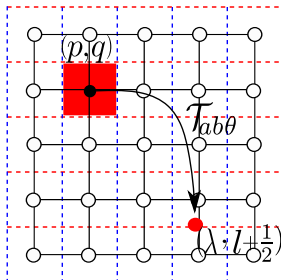
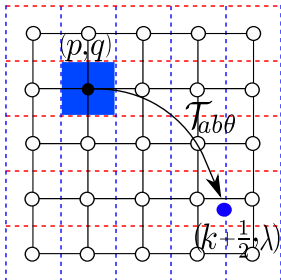


Les transformations critiques correspondent aux discontinuités de TRD.

# Transformation critique

## Définition

Une transformation rigide est **critique** si elle projette au moins un point entier sur la semi-grille **verticale** ou **horizontale**.



Les transformations critiques correspondent aux discontinuités de TRD.

# Surfaces de basculement

## Définition

**Les surfaces de basculement** sont des surfaces associées à des transformations critiques dans l'espace des paramètres  $(a, b, \theta)$ .

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

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

où  $p, q, k, l \in \mathbb{Z}$ .

## Surfaces de basculement

### Définition

**Les surfaces de basculement** sont des surfaces associées à des transformations critiques dans l'espace des paramètres  $(a, b, \theta)$ .

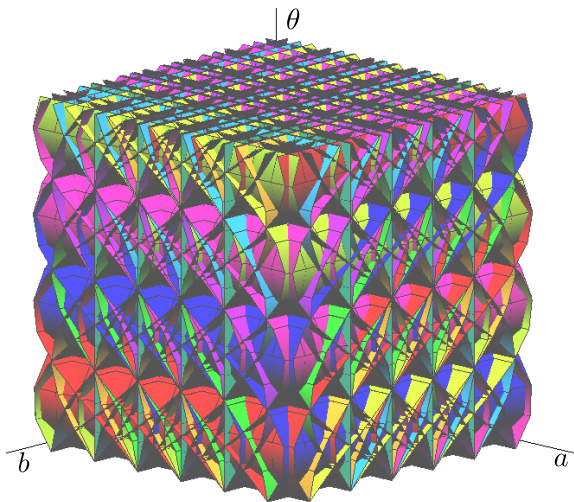
$$\left| \begin{array}{l} \Phi_{pqk} : \mathbb{R}^2 \longrightarrow \mathbb{R} \\ (b, \theta) \longmapsto a = k + \frac{1}{2} + q \sin \theta - p \cos \theta \quad (\textit{vertical}) \end{array} \right.$$

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

où  $p, q, k, l \in \mathbb{Z}$ .

$\implies$  Il y a  $2N^2(N+1)$  surfaces de basculement.

## Exemple des surfaces de basculement



Les surfaces de basculement  $\Phi_{pqk}$  et  $\Psi_{pql}$  pour  $p, q \in [0, 2]$  et  $k, l \in [0, 3]$ .

## Courbes de basculement

### Définition

**Les courbes de basculement** sont définies comme les projections orthogonales de  $\phi_{pqk}$  (resp.  $\psi_{pql}$ ) sur le plan  $(a, \theta)$  (resp.  $(b, \theta)$ ).

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

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

où  $p, q, k, l \in \mathbb{Z}$ .



## Courbes de basculement

### Définition

**Les courbes de basculement** sont définies comme les projections orthogonales de  $\phi_{pqk}$  (resp.  $\psi_{pql}$ ) sur le plan  $(a, \theta)$  (resp.  $(b, \theta)$ ).

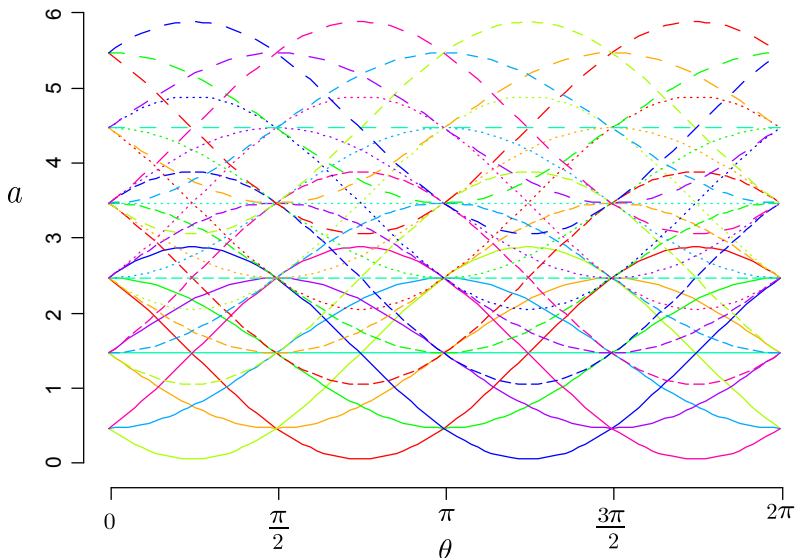
$$\left| \begin{array}{l} \phi_{pqk} : \mathbb{R} \longrightarrow \mathbb{R} \\ \theta \longmapsto a = k + \frac{1}{2} + q \sin \theta - p \cos \theta \end{array} \right.$$

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

où  $p, q, k, l \in \mathbb{Z}$ .

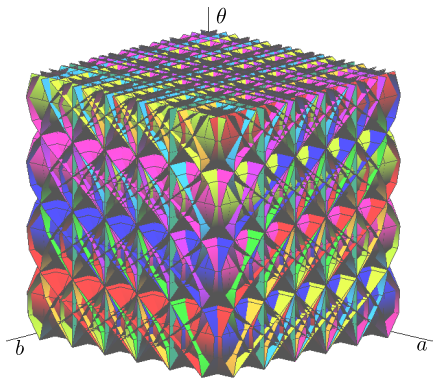
$\implies$  Il y a  $N^2(N+1)$  courbes de basculement.

## Exemple des courbes de basculement

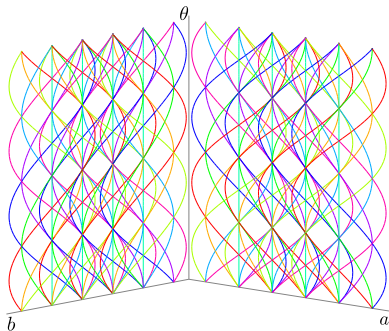


Les courbes de basculement  $\phi_{pqk}$  pour  $p, q \in [0, 2]$  et  $k \in [0, 3]$ .

# Exemple des surfaces/courbes de basculement



(a) Surfaces de basculement



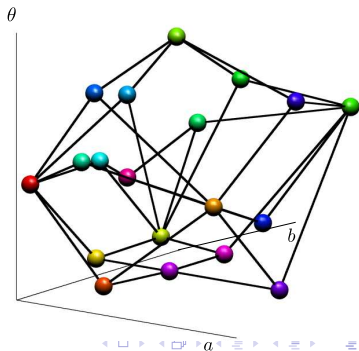
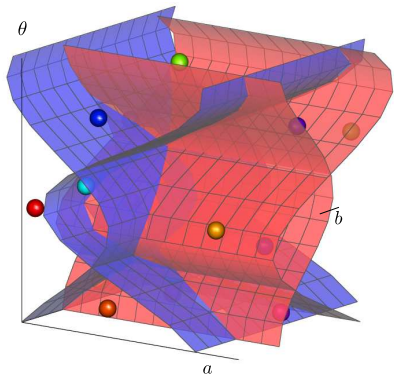
(b) Courbes de basculement

# Graphe des transformations rigides discrètes

## Définition

Un **graphe des transformations rigides discrètes** (graphe des TRD) est un graphe  $G = (V, E)$  tel que

- chaque sommet de  $V$  correspond à une TRD,
- chaque arête de  $E$  relie deux sommets partageant une surface de basculement.

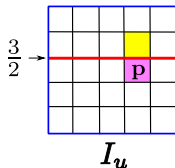
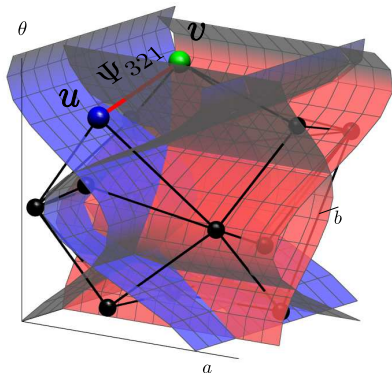


# Graphe des transformations rigides discrètes

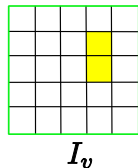
## Définition

Un **graphe des transformations rigides discrètes** (graphe des TRD) est un graphe  $G = (V, E)$  tel que

- chaque sommet de  $V$  correspond à une TRD,
- chaque arête de  $E$  relie deux sommets partageant une surface de basculement.



$$e = (u, v, \Psi_{321})$$

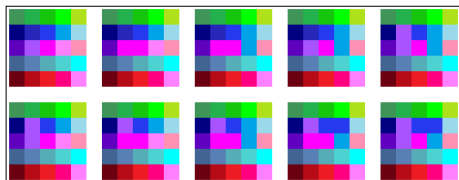
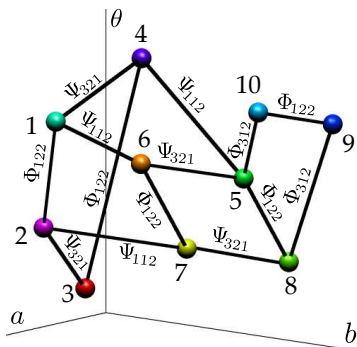


# Graphe des transformations rigides discrètes

## Définition

Un **graphe des transformations rigides discrètes** (graphe des TRD) est un graphe  $G = (V, E)$  tel que

- chaque sommet de  $V$  correspond à une TRD,
- chaque arête de  $E$  relie deux sommets partageant une surface de basculement.

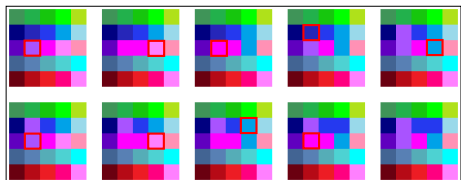
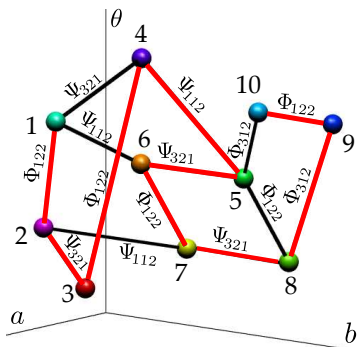


# Graphe des transformations rigides discrètes

## Définition

Un **graphe des transformations rigides discrètes** (graphe des TRD) est un graphe  $G = (V, E)$  tel que

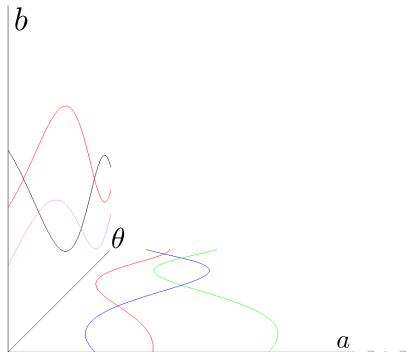
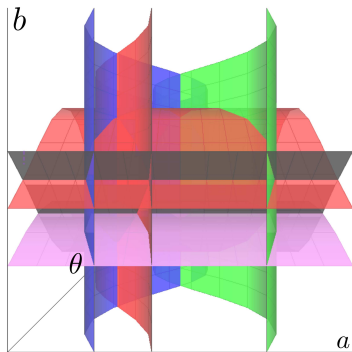
- chaque sommet de  $V$  correspond à une TRD,
- chaque arête de  $E$  relie deux sommets partageant une surface de basculement.



# Algorithme de construction du graphe des TRD

## Problème

- **Entrée** : Un ensemble de surfaces de basculement  $S$ .
- **Sortie** : Le graphe des TRD  $G$  de  $S$ .
- **Approche** : Méthode de balayage (sweeping method).

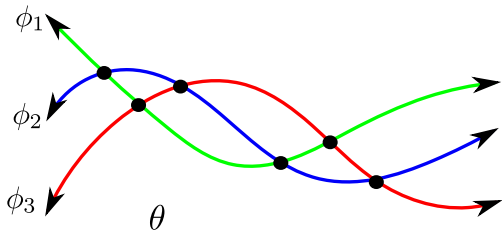




# Construction du graphe avec des courbes de basculement

## Problème

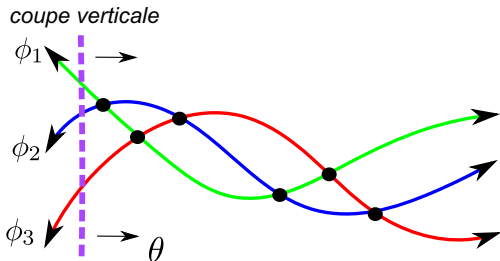
- **Entrée** : Un ensemble de courbes de basculement  $C$ .
- **Sortie** : Le graphe des TRD en 2D  $G_C$  de  $C$ .
- **Approche** : Méthode de balayage (sweeping method).



# Construction du graphe avec des courbes de basculement

## Méthode de balayage

La méthode de balayage utilise une coupe verticale pour parcourir l'espace des paramètres le long de  $C$  afin de construire  $G_C$ .

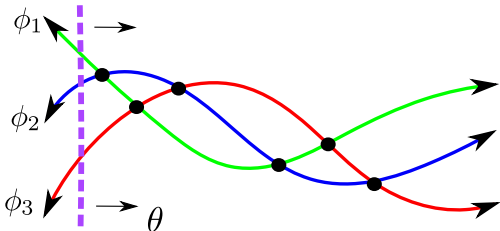


# Construction du graphe avec des courbes de basculement

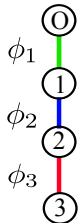
## Méthode de balayage

La méthode de balayage utilise une coupe verticale pour parcourir l'espace des paramètres le long de  $C$  afin de construire  $G_C$ .

*coupe verticale*



*graphe partiel*



# Construction incrémentale d'un graphe des TRD en 2D

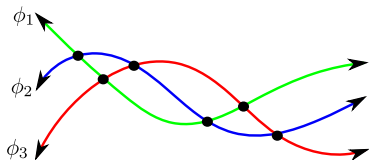
## Proposition

Soit  $C$  l'ensemble des courbes de basculement et  $G_C$  le graphe des TRD en 2D de  $C$ . Nous avons

$$G_C = \bigcup_{i=0}^m \delta G_{C_i}$$

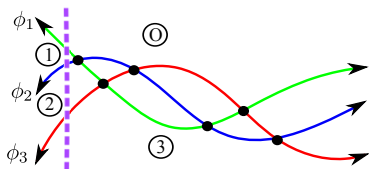
où  $\delta G_{C_i}$  est le graphe partiel à la  $i$ -ème étape, et  $m$  est le nombre total d'intersections.

# Illustration de l'algorithme de construction

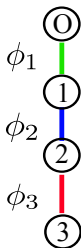


Les courbes de basculement

# Illustration de l'algorithme de construction

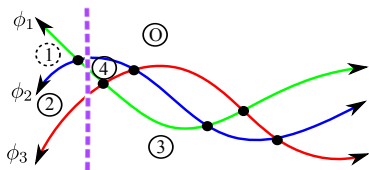


L'avancement de la coupe

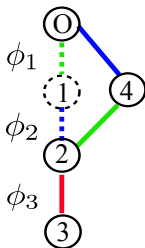


Le graphe partiel

# Illustration de l'algorithme de construction

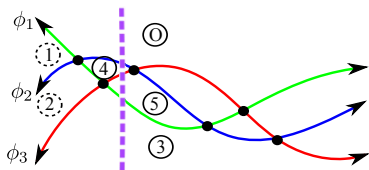


L'avancement de la coupe

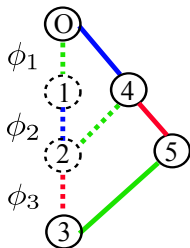


Le graphe partiel

# Illustration de l'algorithme de construction



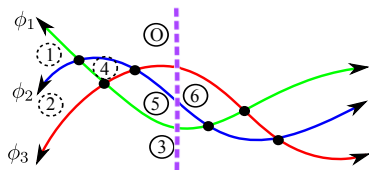
L'avancement de la coupe



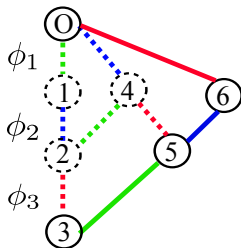
Le graphe partiel



# Illustration de l'algorithme de construction

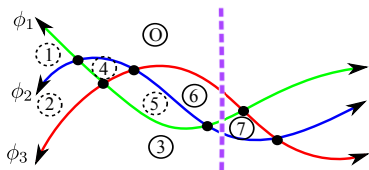


L'avancement de la coupe

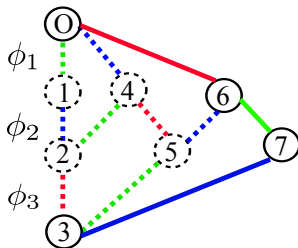


Le graphe partiel

# Illustration de l'algorithme de construction

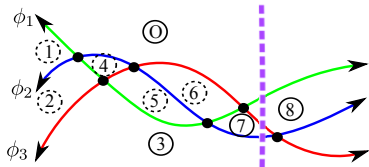


L'avancement de la coupe

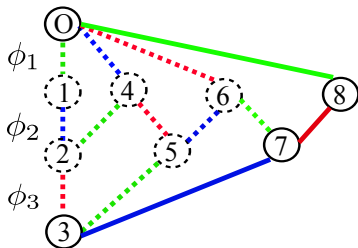


Le graphe partiel

# Illustration de l'algorithme de construction

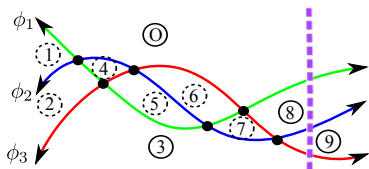


L'avancement de la coupe

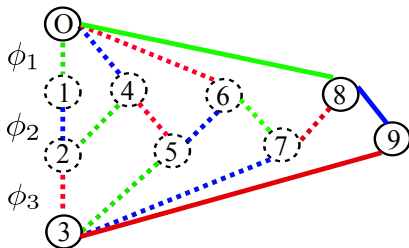


Le graphe partiel

# Illustration de l'algorithme de construction

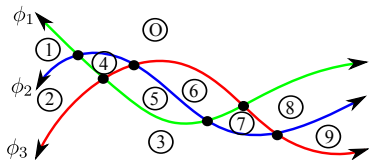


L'avancement de la coupe

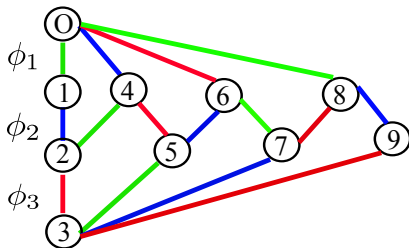


Le graphe partiel

# Illustration de l'algorithme de construction

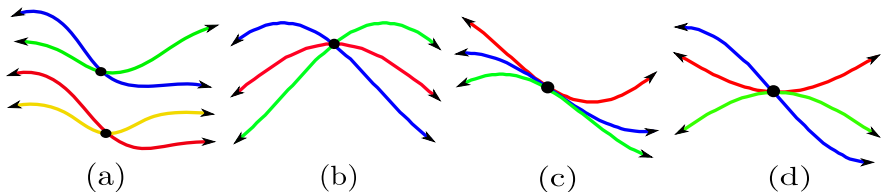


L'avancement de la coupe



Le graphe des TRD en 2D

## Cas dégénérés



Remarque : l'intersection de deux courbes de basculement peut être exprimée par un nombre quadratique irrationnel.

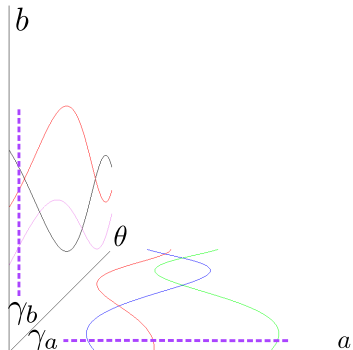
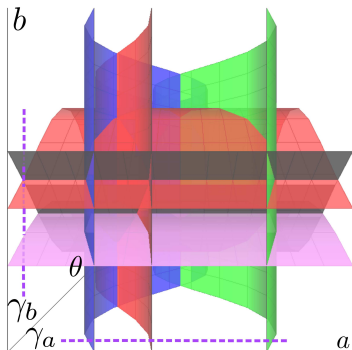
### Comparaison exacte

Nous pouvons comparer les intersections par un calcul exact en utilisant les fractions continues.

# Construction du graphe des TRD

## Problème

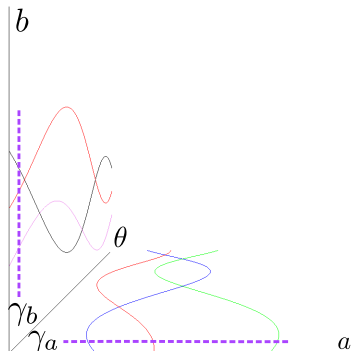
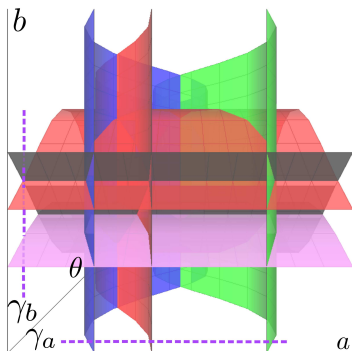
- **Entrée** : Un ensemble de surfaces de basculement  $S$ .
- **Sortie** : Le graphe des TRD  $G$  de  $S$ .
- **Approche** : Méthode de balayage (sweeping method).



# Construction du graphe des TRD

## Méthode de balayage

La méthode de balayage utilise deux coupes  $\gamma_a$  et  $\gamma_b$  pour parcourir l'espace des paramètres le long des courbes dans les plans  $(a, \theta)$  et  $(b, \theta)$  afin de construire  $G$ .





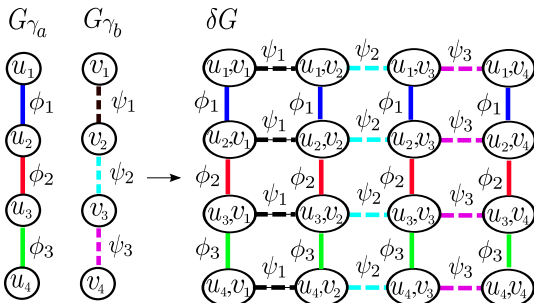
# Construction du graphe des TRD

## Proposition

Soit  $\delta G_a$  et  $\delta G_b$  les graphes partiels des coupes  $\gamma_a$  et  $\gamma_b$  respectivement. Nous avons

$$\delta G = \delta G_a \times \delta G_b$$

où  $\delta G$  est le graphe partiel du graphe des TRD  $G$ .



# Construction incrémentale d'un graphe des TRD

## Proposition

Soit  $S$  l'ensemble des surfaces de basculement et  $G$  le graphe des TRD de  $S$ . Nous avons

$$G = \bigcup_{i=0}^m \delta G_i$$

où  $\delta G_i$  est le graphe partiel à la  $i$ -ème étape et  $m$  est le nombre total d'intersections.

# Analyse de complexité du graphe des TRD

Étant donnée une image de taille  $N \times N$

## Propriétés des courbes de basculement

- Il y a  $N^2(N + 1)$  courbes de basculement.
- Deux courbes de basculement s'intersectent au plus en deux points.

## Complexité du graphe des TRD en 2D (sommets)

$$\begin{array}{ccccc} O(N^3) & + & O(N^6) & = & O(N^6) \\ \textit{Graphe initial} & & \textit{Nombre d'intersections} & & \end{array}$$

(P. Ngo, CVIU - 2013)

# Analyse de complexité du graphe des TRD

Étant donnée une image de taille  $N \times N$

## Propriétés des surfaces de basculement

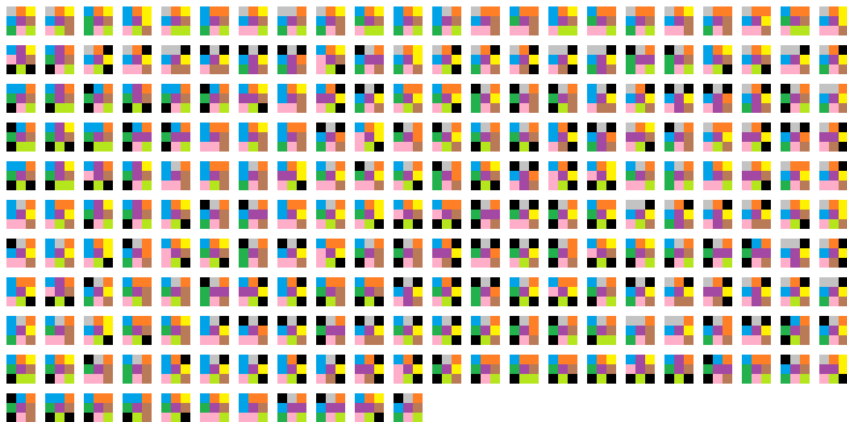
- Il y a  $2N^2(N + 1)$  surfaces de basculement.
- Deux courbes de basculement s'intersectent au plus deux fois.

## Complexité du graphe des TRD (sommets)

$$\begin{array}{l} O(N^3) \times O(N^3) \\ \textit{Graphe initial} \end{array} + \begin{array}{l} O(N^6) \times O(N^3) \\ \textit{Nombre d'intersections} \end{array} = O(N^9)$$

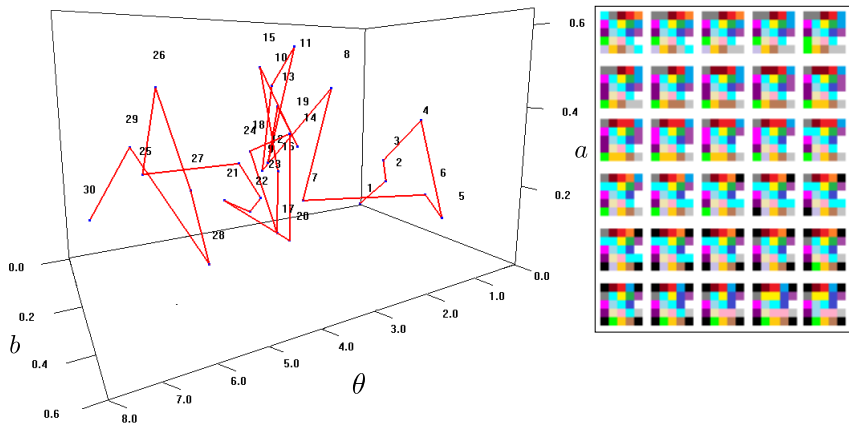
(P. Ngo, CVIU - 2013)

# Génération de toutes les images numériques transformées

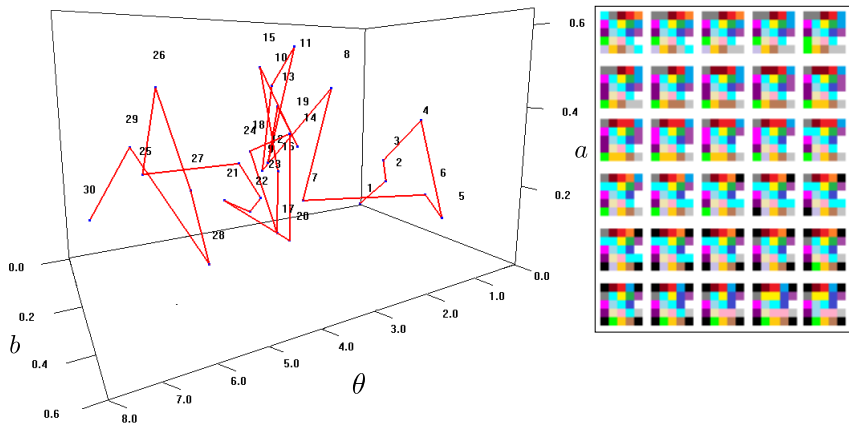


L'ensemble des 231 images générées à partir de la partie de graphe des TRD associée à l'image de taille  $3 \times 3$  avec  $a, b \in ]-\frac{1}{2}, \frac{1}{2}[$  et  $\theta \in ]0, \frac{\pi}{4}[$ .

# Chemin de transitions discrètes

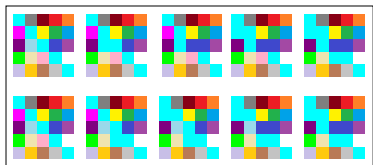
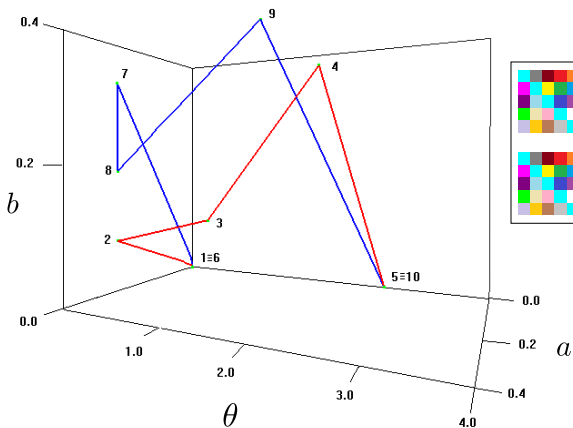


# Chemin de transitions discrètes



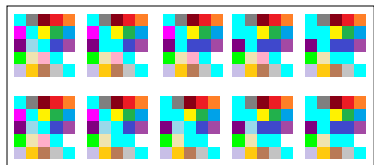
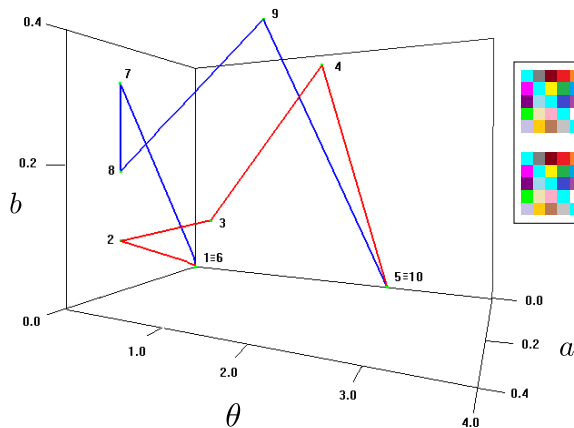
⇒ Utiliser le graphe des TRD en recalage d'images

# Évaluation des chemins de transitions discrètes





# Évaluation des chemins de transitions discrètes



⇒ Utiliser le graphe des TRD en analyse topologique d'images

# Graphe des TRD pour le recalage d'images

## Formalisation du problème

$$T_v^* = \arg \min_{T_v \in \mathbb{T}} d(A, T_v(B)),$$

où :

- $A$  et  $B$  sont deux images données de taille  $N \times N$ ,
- $T_v^*$  est une transformation rigide qui optimise le coût de l'équation,
- $\mathbb{T} = \{T_v \mid v \in V\}$  est l'ensemble des TRD, et
- $d(A, B)$  est une distance entre deux images  $A$  et  $B$ .

# Graphe des TRD pour le recalage d'images

## Formalisation du problème

$$T_v^* = \arg \min_{T_v \in \mathbb{T}} d(A, T_v(B)),$$

où :

- $A$  et  $B$  sont deux images données de taille  $N \times N$ ,
- $T_v^*$  est une transformation rigide qui optimise le coût de l'équation,
- $\mathbb{T} = \{T_v \mid v \in V\}$  est l'ensemble des TRD, et
- $d(A, B)$  est une distance entre deux images  $A$  et  $B$ .

Le graphe des TRD a une complexité de  $O(N^9)$ , où  $N \times N$  est la taille de l'image.

# Graphe des TRD pour le recalage d'images

## Formalisation du problème

$$T_v^* = \arg \min_{T_v \in \mathbb{T}} d(A, T_v(B)),$$

où :

- $A$  et  $B$  sont deux images données de taille  $N \times N$ ,
- $T_v^*$  est une transformation rigide qui optimise le coût de l'équation,
- $\mathbb{T} = \{T_v \mid v \in V\}$  est l'ensemble des TRD, et
- $d(A, B)$  est une distance entre deux images  $A$  et  $B$ .

## Algorithme de recherche locale

- **Entrée** : Une image de référence  $A$  et une image cible  $B$ , une TRD  $v_0 \in V$
- **Sortie** : Un optimum local  $\hat{v} \in V$  dans un voisinage de  $v_0$
- **Approche** : Algorithme de descente de gradient.

# Graphe des TRD pour le recalage d'images

## Formalisation du problème

$$T_v^* = \arg \min_{T_v \in \mathbb{T}} d(A, T_v(B)),$$

où :

- $A$  et  $B$  sont deux images données de taille  $N \times N$ ,
- $T_v^*$  est une transformation rigide qui optimise le coût de l'équation,
- $\mathbb{T} = \{T_v \mid v \in V\}$  est l'ensemble des TRD, et
- $d(A, B)$  est une distance entre deux images  $A$  et  $B$ .

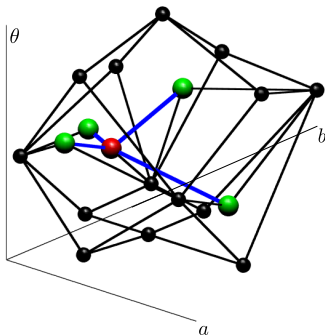
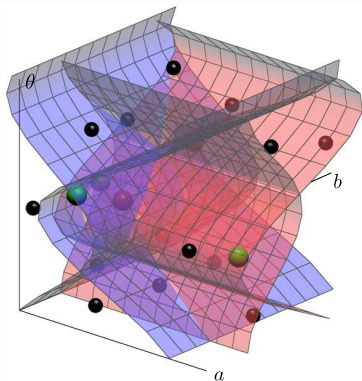
## Algorithme de recherche locale

- **Entrée** : Une image de référence  $A$  et une image cible  $B$ , une TRD  $v_0 \in V$
- **Sortie** : Un optimum local  $\hat{v} \in V$  dans un voisinage de  $v_0$
- **Approche** : Algorithme de descente de gradient.

L'algorithme a une complexité de  $O(N^2)$  pour chaque étape de la recherche locale.

# Voisinage des TRD

## Voisins d'une TRD

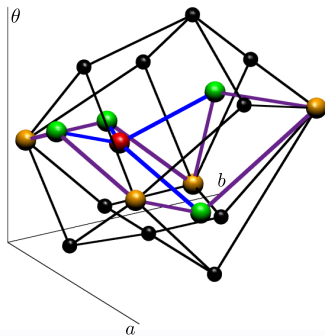
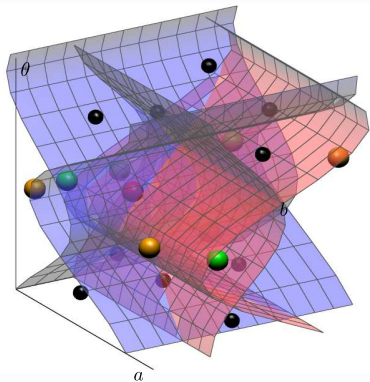


L'algorithme pour calculer les voisins d'une TRD coûte  $O(N^2)$  en moyenne.

(P. Ngo, GCCV - 2013)

# Voisinage des TRD

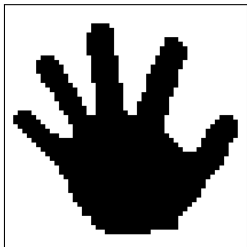
## K-voisins d'une TRD



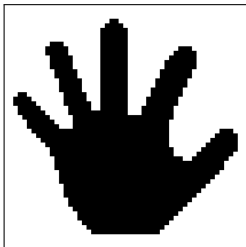
L'algorithme pour calculer les k-voisins d'une TRD coûte  $O(kN^2)$  en moyenne.

(P. Ngo, GCCV - 2013)

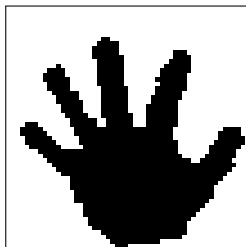
# Expériences avec des images binaires



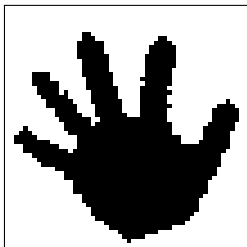
(a) Image de référence



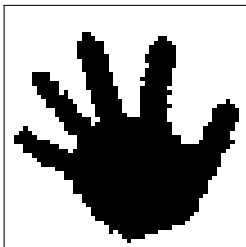
(b) Image cible



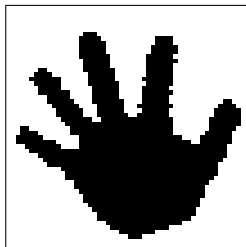
(c) 1-voisinage



(d) 3-voisinage



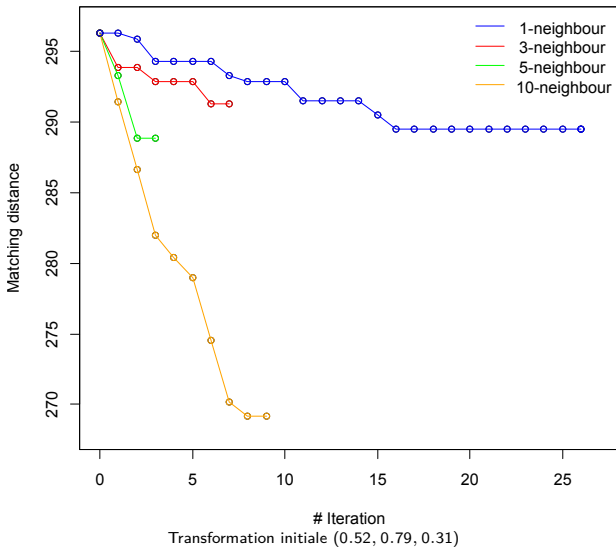
(e) 5-voisinage



(f) 10-voisinage



# Expériences avec des images binaires



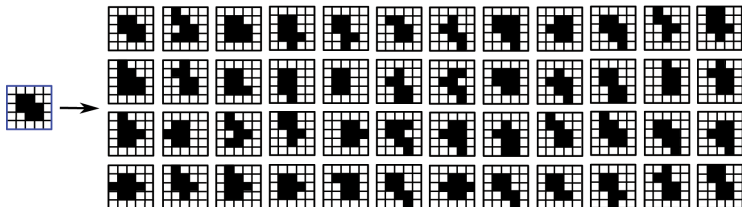
# Expériences avec des images binaires

Une séquence d'images transformées utilisant un 3-voisinage

# Graphe des TRD en analyse topologique d'images

## Définition

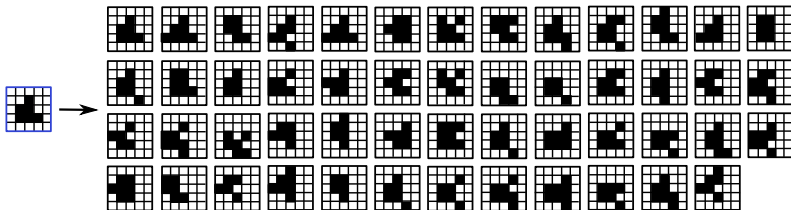
Une image  $I$  est **topologiquement invariante** si ses images transformées par toutes les transformations rigides ont le même type d'homotopie que  $I$ .



# Graphe des TRD en analyse topologique d'images

## Définition

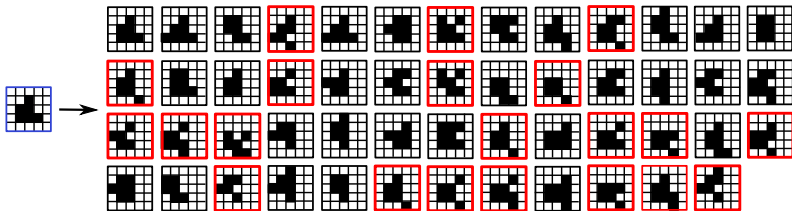
Une image  $I$  est **topologiquement invariante** si ses images transformées par toutes les transformations rigides ont le même type d'homotopie que  $I$ .



# Graphe des TRD en analyse topologique d'images

## Définition

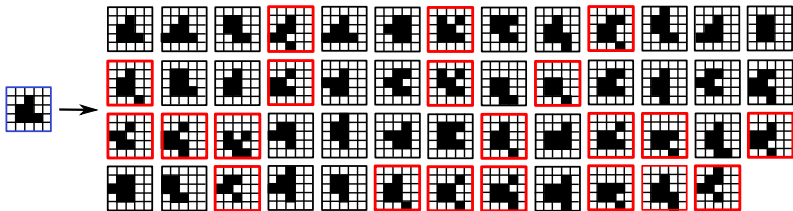
Une image  $I$  est **topologiquement invariante** si ses images transformées par toutes les transformations rigides ont le même type d'homotopie que  $I$ .



# Graphe des TRD en analyse topologique d'images

## Définition

Une image  $I$  est **topologiquement invariante** si ses images transformées par toutes les transformations rigides ont le même type d'homotopie que  $I$ .



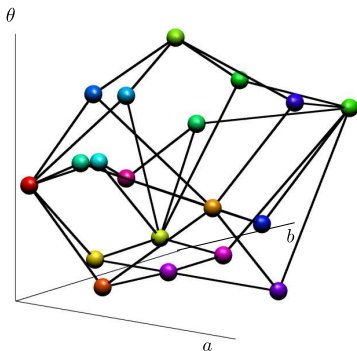
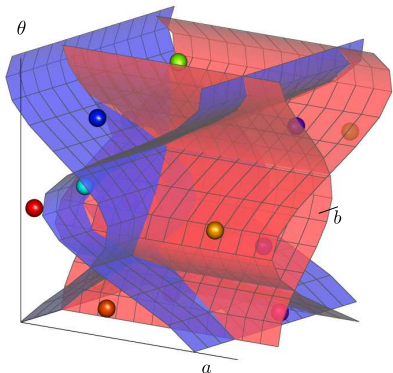
## Outils méthodologiques pour vérifier l'invariance topologique

- graphe des TRD, et
- points simples.

# Algorithme de vérification

## Méthode naïve

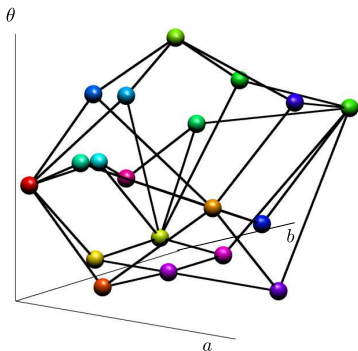
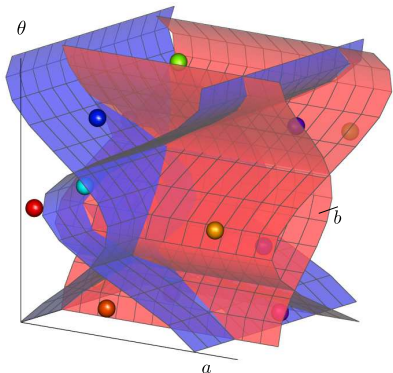
- **le graphe des TRD** pour générer incrémentalement toutes les images transformées,



# Algorithme de vérification

## Méthode naïve

- **le graphe des TRD** pour générer incrémentalement toutes les images transformées,
- **un test de simplicité** pour vérifier l'invariance topologique.

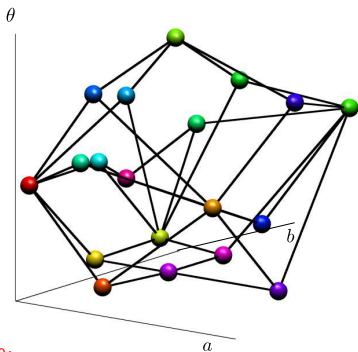
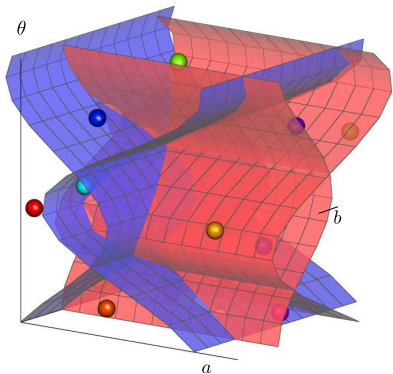




# Algorithme de vérification

## Méthode naïve

- **le graphe des TRD** pour générer incrémentalement toutes les images transformées,
- **un test de simplicité** pour vérifier l'invariance topologique.



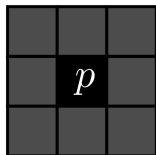
Le graphe des TRD a une complexité de  $O(N^9)$ , où  $N \times N$  est la taille de l'image.

## Algorithme de vérification

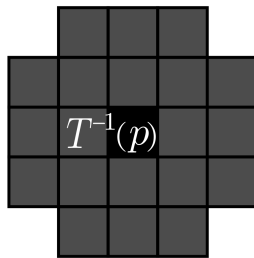
### Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$



$N_8(\mathbf{p})$



$N_{20}(T^{-1}(\mathbf{p}))$

## Algorithme de vérification

### Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$

### Pré-traitement : génération d'une table de correspondance

- Générer toutes les configurations binaires portées par un 20-voisinage.
- Utiliser le graphe des TRD pour étudier l'invariance topologique du point central de ces configurations.

## Algorithme de vérification

### Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$

### Pré-traitement : génération d'une table de correspondance

- Générer toutes les configurations binaires portées par un 20-voisinage.
- Utiliser le graphe des TRD pour étudier l'invariance topologique du point central de ces configurations.

L'algorithme avec la table de correspondance,

(P. Ngo, JMIV - 2013)

## Algorithme de vérification

### Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$

### Pré-traitement : génération d'une table de correspondance

- Générer toutes les configurations binaires portées par un 20-voisinage.
- Utiliser le graphe des TRD pour étudier l'invariance topologique du point central de ces configurations.

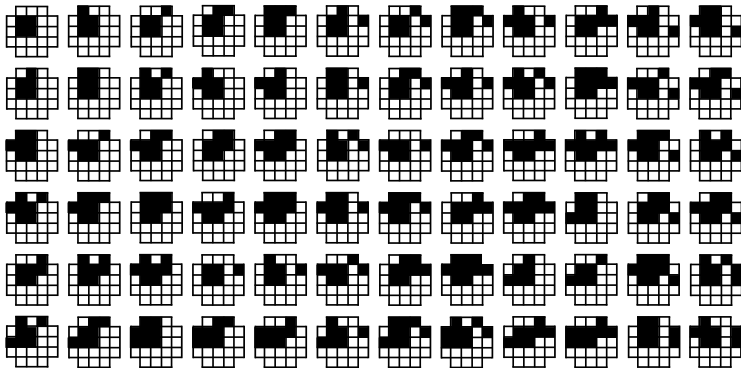
L'algorithme avec la table de correspondance, et la vérification est réalisée en temps linéaire par rapport à la taille de l'image.

(P. Ngo, JMIV - 2013)

## Les tables de correspondance

Parmi les 124 260 configurations binaires (hors symétries), il y a

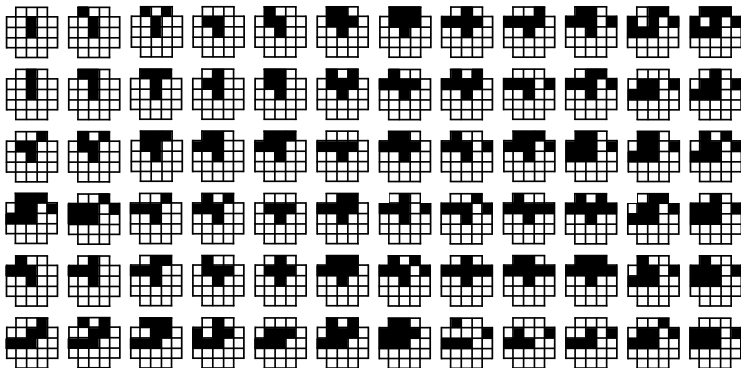
- 10 644 configurations dans  $P_4$  pour la (4, 8)-adjacence.



## Les tables de correspondance

Parmi les 124 260 configurations binaires (hors symétries), il y a

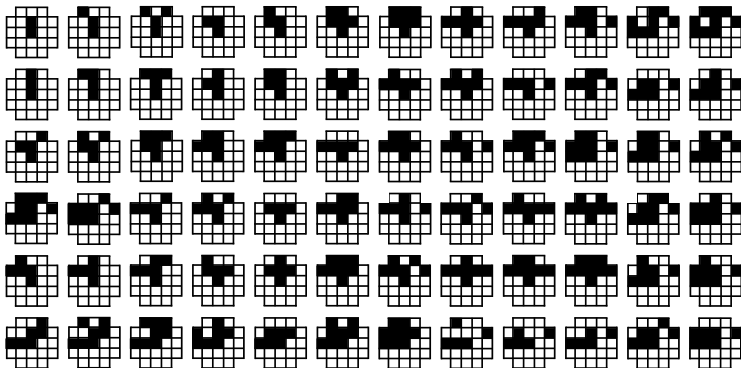
- 10 644 configurations dans  $P_4$  pour la (4, 8)-adjacence.
- 10 752 configurations dans  $P_8$  pour la (8, 4)-adjacence.



## Les tables de correspondance

Parmi les 124 260 configurations binaires (hors symétries), il y a

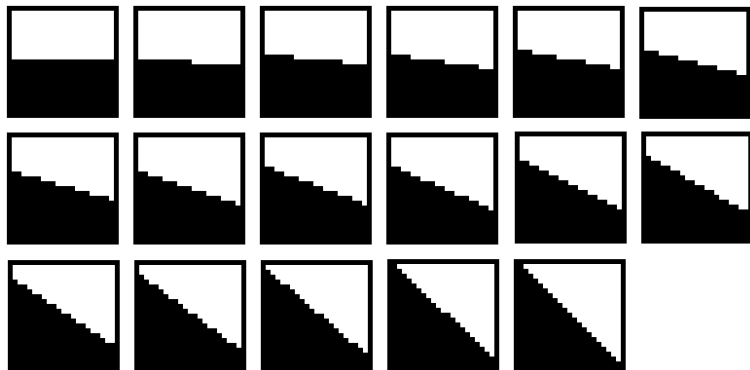
- 10 644 configurations dans  $P_4$  pour la (4, 8)-adjacence.
- 10 752 configurations dans  $P_8$  pour la (8, 4)-adjacence.



Conditions suffisantes pour l'invariance topologique.



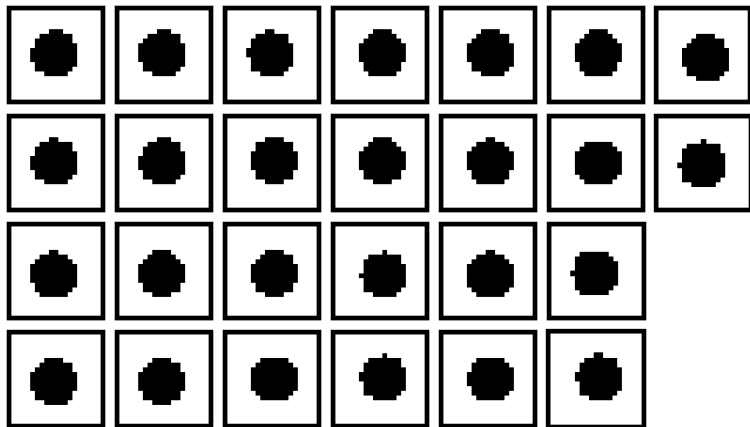
## Expériences avec des demi-plans



(C. Berenstein 98)

Les demi-plans centrés générés dans une image de taille  $21 \times 21$ .

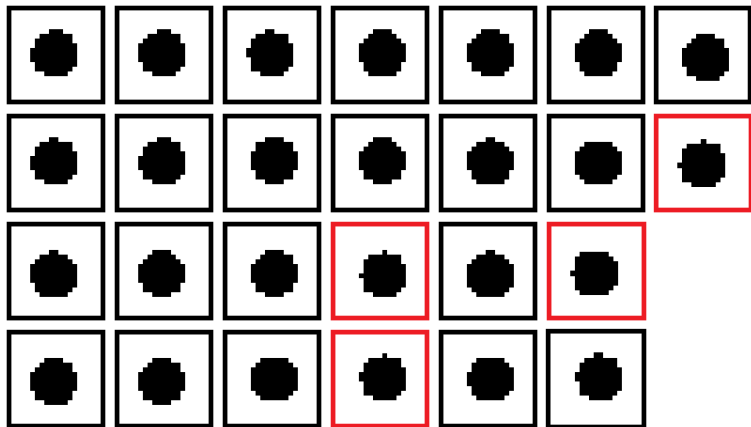
## Expériences avec des disques



(B. Nagy 98)

Disques numérisés de rayon 5 dont le centre est à l'intérieur du pixel central.

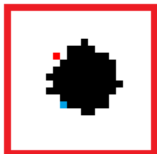
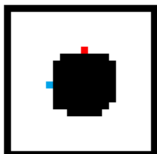
## Expériences avec des disques



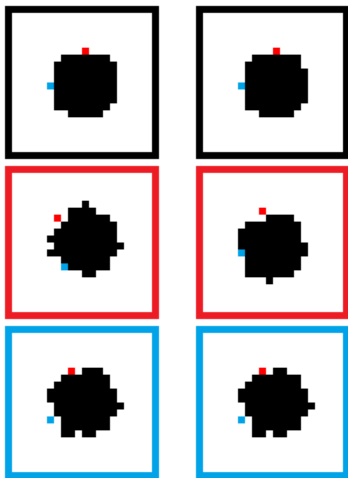
(B. Nagy 98)

Disques numérisés de rayon 5 dont le centre est à l'intérieur du pixel central.

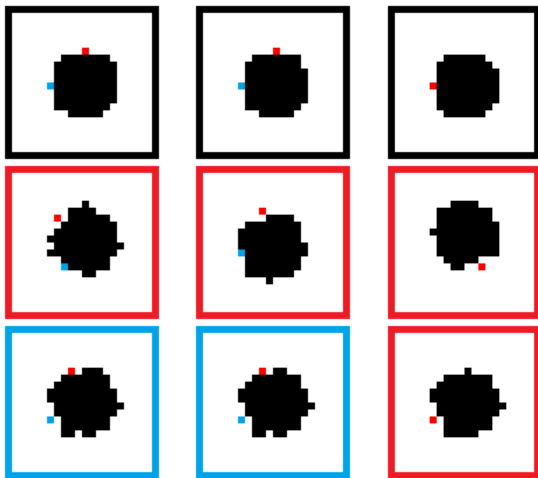
# Expériences avec des disques



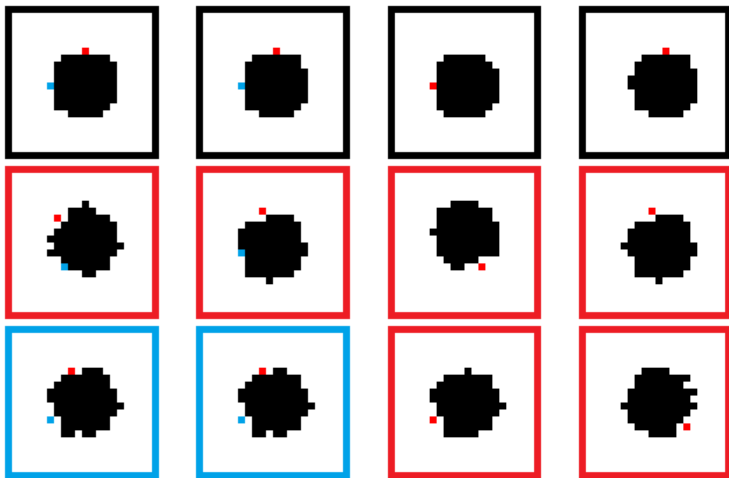
# Expériences avec des disques



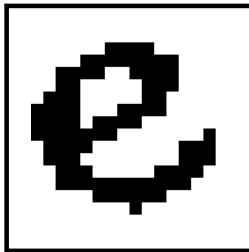
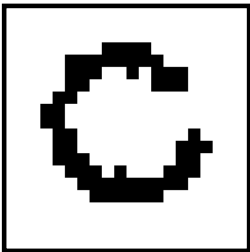
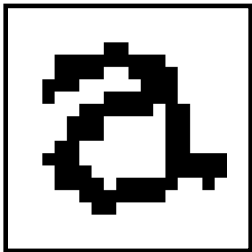
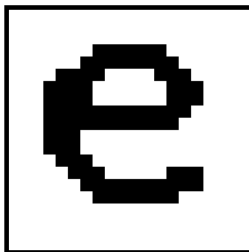
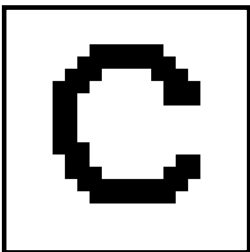
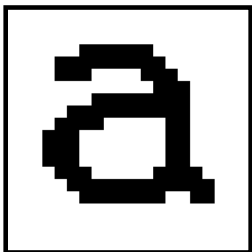
# Expériences avec des disques



# Expériences avec des disques

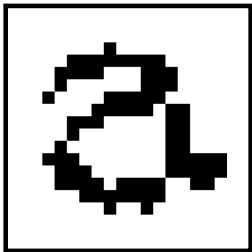
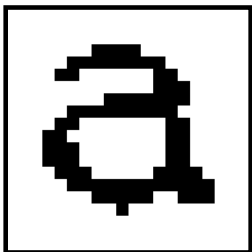


# Expériences avec des caractères : invariance topologique

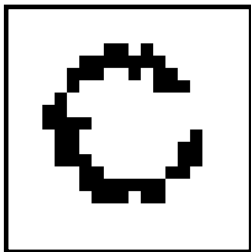
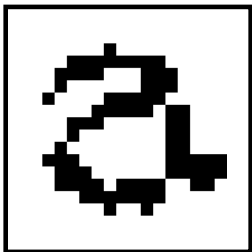
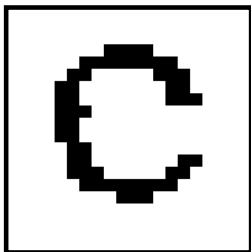
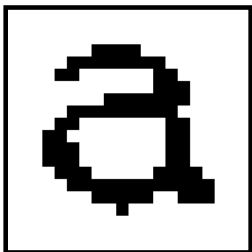




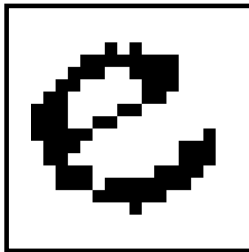
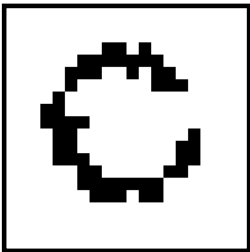
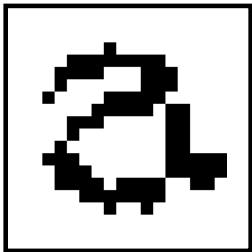
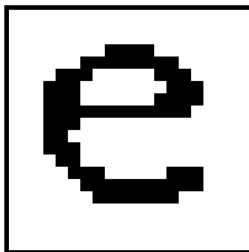
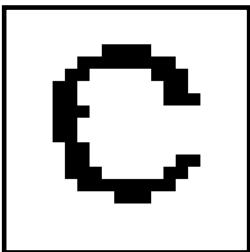
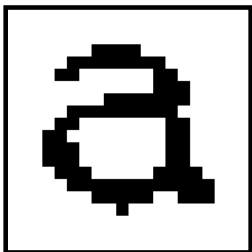
# Expériences avec des caractères : variance topologique



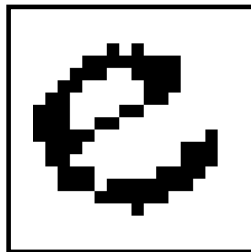
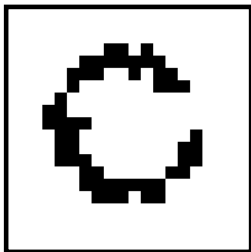
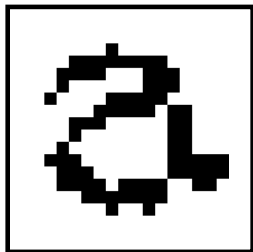
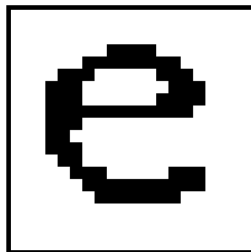
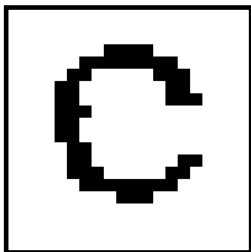
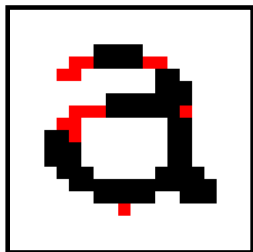
# Expériences avec des caractères : variance topologique



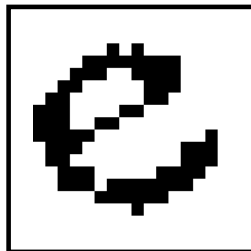
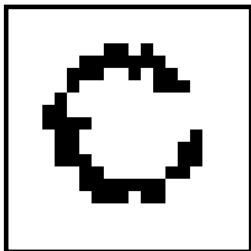
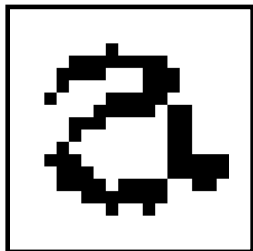
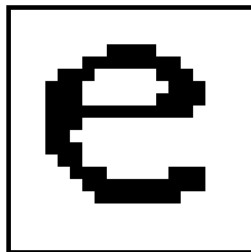
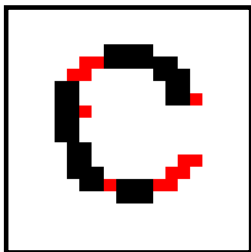
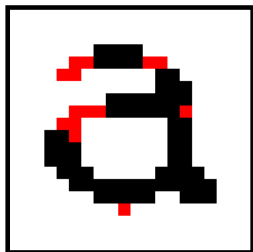
# Expériences avec des caractères : variance topologique



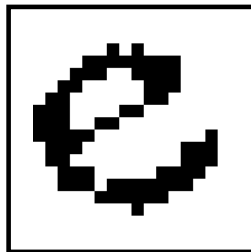
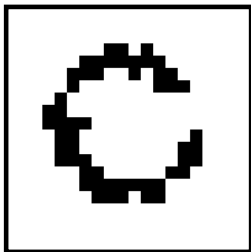
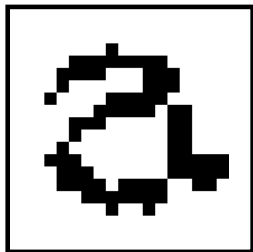
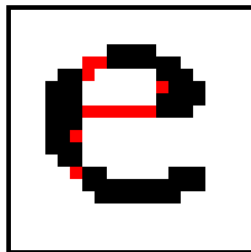
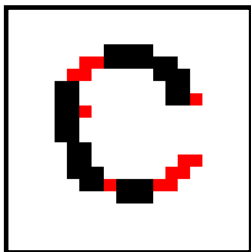
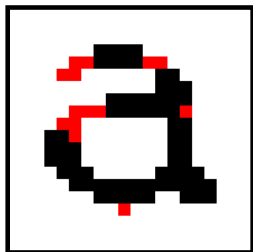
# Expériences avec des caractères : variance topologique



# Expériences avec des caractères : variance topologique



# Expériences avec des caractères : variance topologique



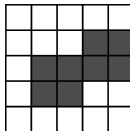
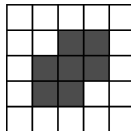
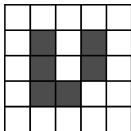
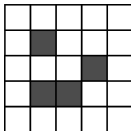
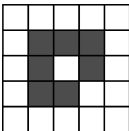
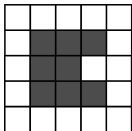
## Proposition

Une image est topologiquement invariante si elle est régulière.

## Définition

Une image binaire  $I$  est **régulière** si elle est bien composée, non-singulière et  $\forall \mathbf{p}, \mathbf{q} \in I$  tel que  $I(\mathbf{p}) = I(\mathbf{q}) = 1$  (resp. 0) :

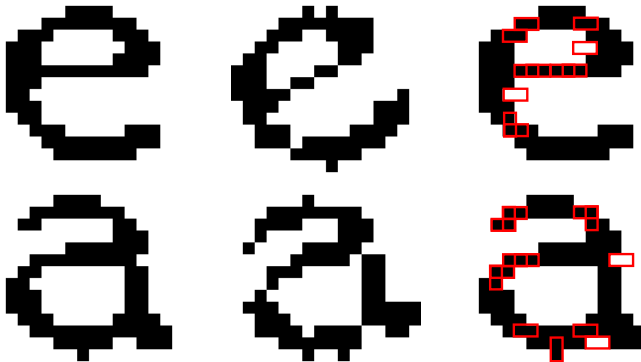
$$(\mathbf{p} \curvearrowright_4 \mathbf{q}) \implies (\exists \boxplus \subseteq I^{-1}(\{1\}(\text{resp. } \{0\})), \mathbf{p}, \mathbf{q} \in \boxplus).$$



(P. Ngo, IEEE-TIP - 2014)

# Proposition

Une image est topologiquement invariante si elle est régulière.

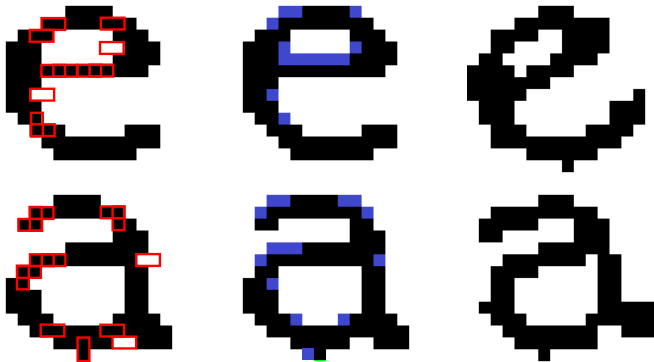


(P. Ngo, IEEE-TIP - 2014)



# Proposition

Une image est topologiquement invariante si elle est régulière.



(P. Ngo, IEEE-TIP - 2014)

# Proposition

Une image est topologiquement invariante si elle est régulière.

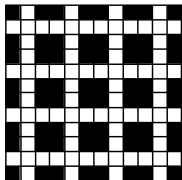


Image originale

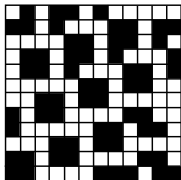


Image transformée

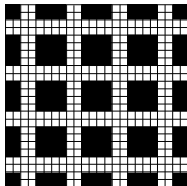


Image régulière

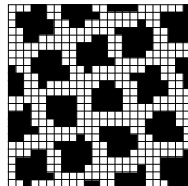


Image régulière transformée

(P. Ngo, IEEE-TIP - 2014)

## Extension : images à niveaux de gris

Une image à niveaux de gris  $I$  est modélisée par l'ensemble fini des images binaires  $\lambda_v(I)$  définies pour tout  $v \in \mathbb{G}$ , où  $\mathbb{G}$  est l'ensemble des niveaux de gris dans  $I$

$$\left| \begin{array}{l} \lambda_v(I) : \mathbb{Z}^2 \rightarrow \{0, 1\} \\ \mathbf{p} \mapsto \begin{cases} 1 & \text{si } v \leq I(\mathbf{p}) \\ 0 & \text{sinon} \end{cases} \end{array} \right.$$

### Définition

Une image à niveaux de gris  $I$  est **topologiquement invariante** si pour tout  $v \in \mathbb{G}$ ,  $\lambda_v(I)$  est topologiquement invariante.

## Expériences : images à niveaux de gris

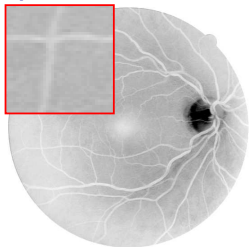


Image originale

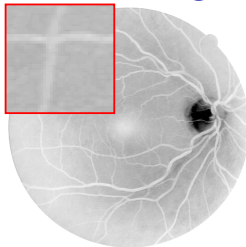


Image régulière

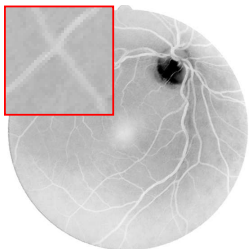


Image transformée

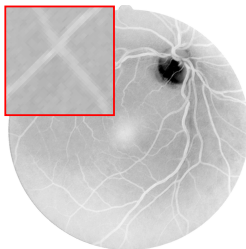


Image régulière transformée

# Expériences : images à niveaux de gris

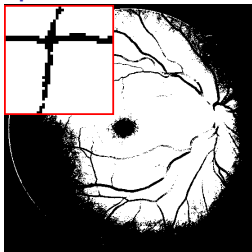


Image originale

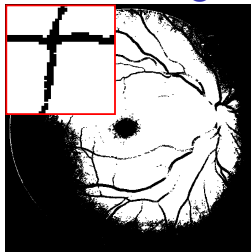


Image régulière

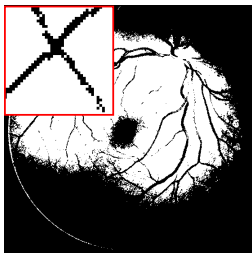


Image transformée

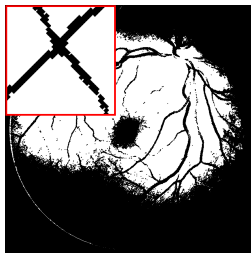


Image régulière transformée

# Expériences : images à niveaux de gris

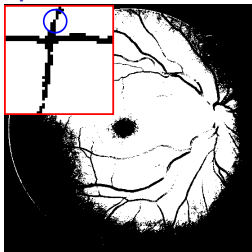


Image originale

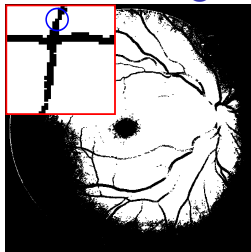


Image régulière

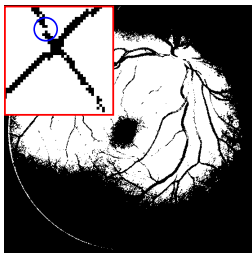


Image transformée

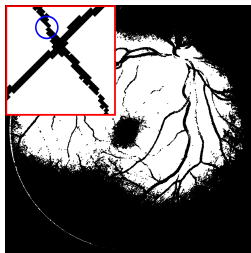


Image régulière transformée

## Extension : images de labels

Une image de labels  $I$  est modélisée par l'ensemble fini des images binaires  $\chi_\Lambda(I)$  définies pour tout  $\Lambda \in 2^{\mathbb{L}}$ , où  $\mathbb{L}$  est l'ensemble des labels de  $I$

$$\left| \begin{array}{l} \chi_\Lambda(I) : \mathbb{Z}^2 \rightarrow \{0, 1\} \\ \mathbf{p} \mapsto \begin{cases} 1 & \text{si } I(\mathbf{p}) \in \Lambda \\ 0 & \text{sinon} \end{cases} \end{array} \right.$$

### Définition

Une image de labels  $I$  est **topologiquement invariante** si pour tout  $\Lambda \in 2^{\mathbb{L}}$ ,  $\chi_\Lambda(I)$  est topologiquement invariante.

# Expériences : images de labels

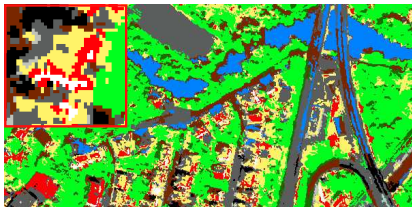


Image originale

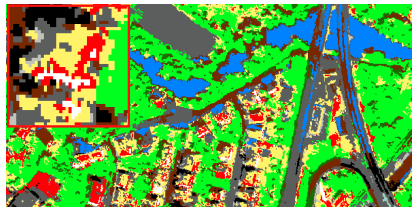


Image régulière

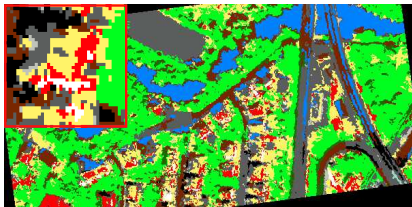


Image transformée

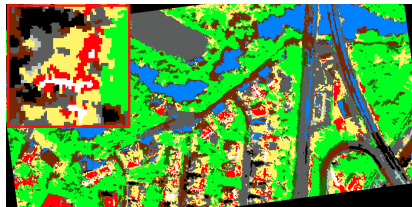


Image régulière transformée



# Expériences : images de labels

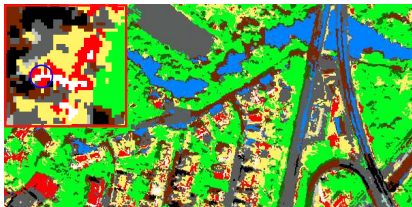


Image originale

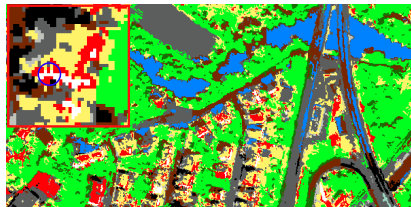


Image régulière

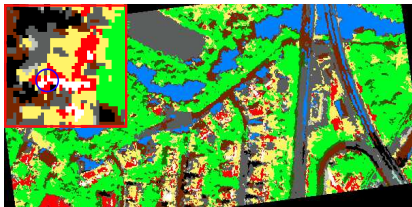


Image transformée

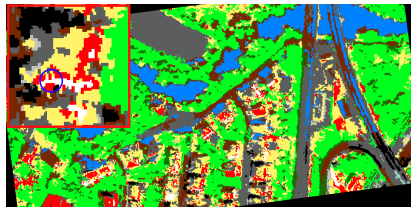


Image régulière transformée

# Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :

## Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :
  - Définition d'un **graphe** représentant la structure combinatoire de ces transformations rigides discrètes,

## Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :
  - Définition d'un **graphe** représentant la structure combinatoire de ces transformations rigides discrètes,
  - démonstration que la complexité de ce graphe est  $\mathbf{O(N^9)}$ ,

## Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :
  - Définition d'un **graphe** représentant la structure combinatoire de ces transformations rigides discrètes,
  - démonstration que la complexité de ce graphe est  **$O(N^9)$** ,
  - obtention d'un **algorithme en temps linéaire** pour le construire.

## Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :
  - Définition d'un **graphe** représentant la structure combinatoire de ces transformations rigides discrètes,
  - démonstration que la complexité de ce graphe est  $O(N^9)$ ,
  - obtention d'un **algorithme en temps linéaire** pour le construire.
- Utilisation du graphe des TRD dans les applications de traitement d'images :

## Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :
  - Définition d'un **graphe** représentant la structure combinatoire de ces transformations rigides discrètes,
  - démonstration que la complexité de ce graphe est  $O(N^9)$ ,
  - obtention d'un **algorithme en temps linéaire** pour le construire.
- Utilisation du graphe des TRD dans les applications de traitement d'images :
  - Recalage d'images,

## Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :
  - Définition d'un **graphe** représentant la structure combinatoire de ces transformations rigides discrètes,
  - démonstration que la complexité de ce graphe est  **$O(N^9)$** ,
  - obtention d'un **algorithme en temps linéaire** pour le construire.
- Utilisation du graphe des TRD dans les applications de traitement d'images :
  - Recalage d'images,
  - étude des relations entre **la géométrie et la topologie** des images numériques,



## Conclusion

- Proposition d'une **version discrète** des transformations rigides pour les images numériques 2D de taille  $N \times N$  :
  - Définition d'un **graphe** représentant la structure combinatoire de ces transformations rigides discrètes,
  - démonstration que la complexité de ce graphe est  $O(N^9)$ ,
  - obtention d'un **algorithme en temps linéaire** pour le construire.
- Utilisation du graphe des TRD dans les applications de traitement d'images :
  - Recalage d'images,
  - étude des relations entre **la géométrie et la topologie** des images numériques,
  - proposition de **conditions suffisantes** et d'un algorithme linéaire pour évaluer l'invariance topologique de ces images.

# Perspectives

- Utiliser une approche multi-échelle pour réduire la complexité et le coût de calcul du graphe des TRD.

# Perspectives

- Utiliser une approche multi-échelle pour réduire la complexité et le coût de calcul du graphe des TRD.
- Trouver les conditions nécessaires de l'invariance topologique.

# Perspectives

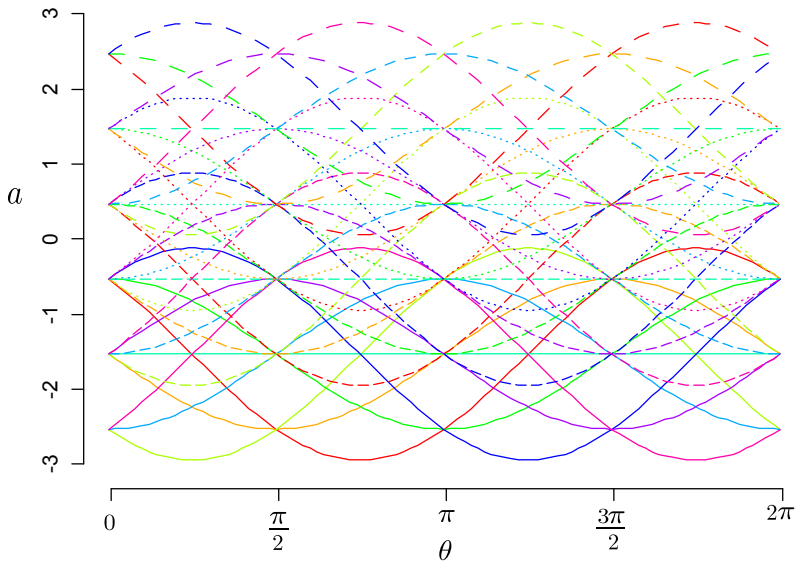
- Utiliser une approche multi-échelle pour réduire la complexité et le coût de calcul du graphe des TRD.
- Trouver les conditions nécessaires de l'invariance topologique.
- Réparer les altérations topologiques des images.

# Perspectives

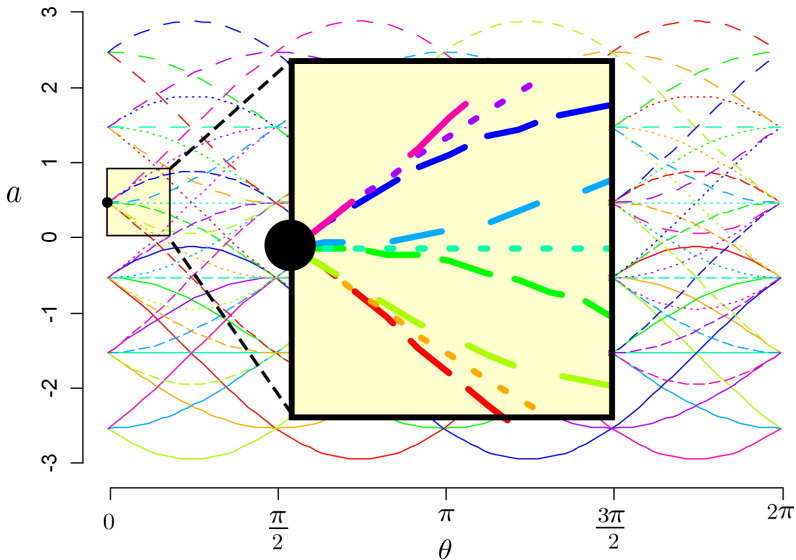
- Utiliser une approche multi-échelle pour réduire la complexité et le coût de calcul du graphe des TRD.
- Trouver les conditions nécessaires de l'invariance topologique.
- Réparer les altérations topologiques des images.
- Étendre la structure et la méthode aux images 3D et aux autres classes de transformations.

Merci pour votre attention

## Cas dégénérés

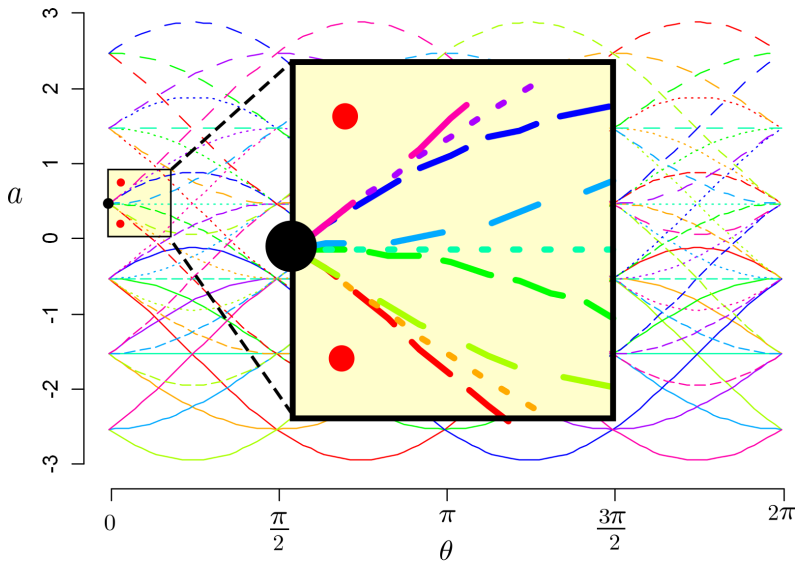


## Cas dégénérés

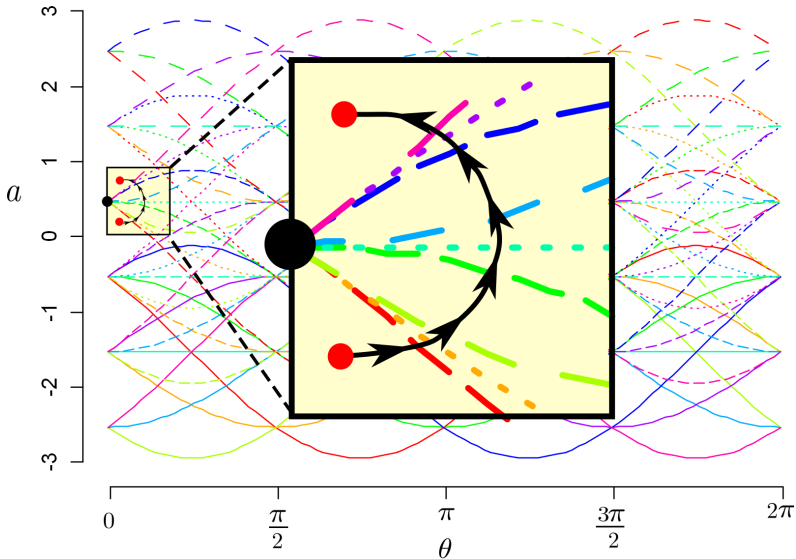




## Cas dégénérés



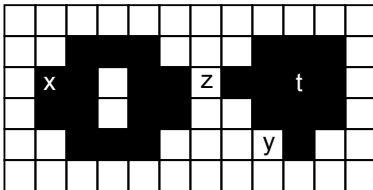
# Cas dégénérés



# Point simple

## Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.

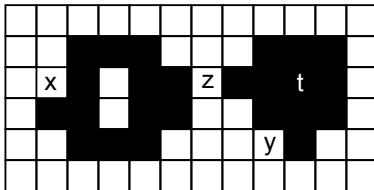


(A. Rosenfeld 70)

# Point simple

## Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.



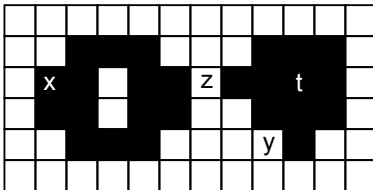
x est simple.

(A. Rosenfeld 70)

# Point simple

## Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.

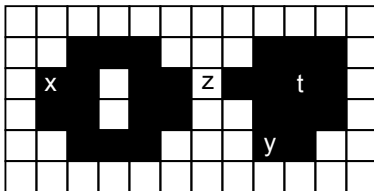


(A. Rosenfeld 70)

# Point simple

## Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.



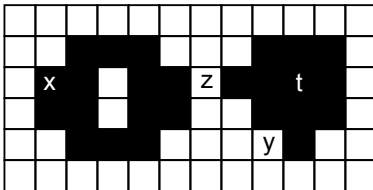
**y** est simple.

(A. Rosenfeld 70)

# Point simple

## Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.

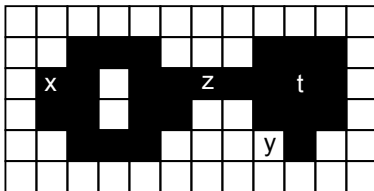


(A. Rosenfeld 70)

## Point simple

### Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.



z n'est pas simple.

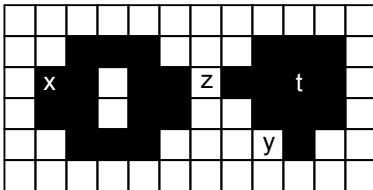
(A. Rosenfeld 70)



# Point simple

## Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.

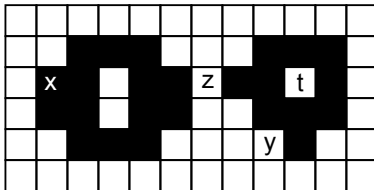


(A. Rosenfeld 70)

## Point simple

### Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.



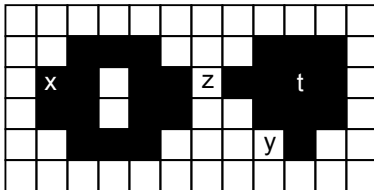
**t** n'est pas simple.

(A. Rosenfeld 70)

# Point simple

## Définition

Un **point simple** est un pixel dont la valeur peut être modifiée sans changer la topologie de l'image associée.



La simplicité d'un pixel est testée en temps constant, en utilisant le voisinage  $3 \times 3$ .

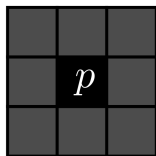
(A. Rosenfeld 70)

# Méthode utilisant des configurations locales

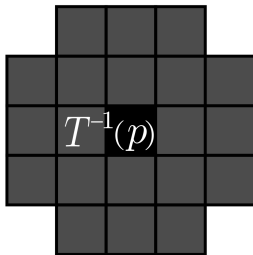
## Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$



$N_8(\mathbf{p})$



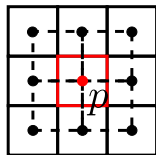
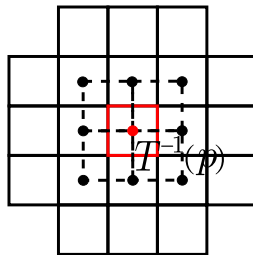
$N_{20}(T^{-1}(\mathbf{p}))$

# Méthode utilisant des configurations locales

## Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$

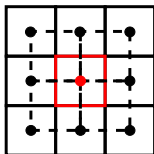

 $N_8(\mathbf{p})$ 

 $N_{20}(T^{-1}(\mathbf{p}))$

# Méthode utilisant des configurations locales

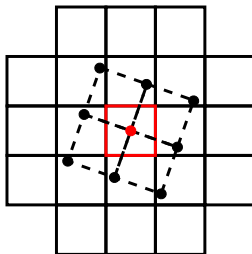
## Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$



$N_8(\mathbf{p})$



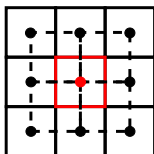
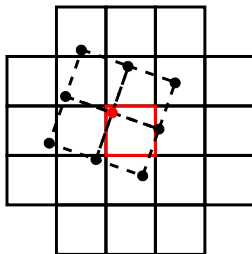
$N_{20}(T^{-1}(\mathbf{p}))$

# Méthode utilisant des configurations locales

## Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$

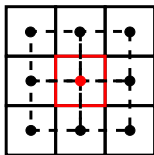

 $N_8(\mathbf{p})$ 

 $N_{20}(T^{-1}(\mathbf{p}))$

# Méthode utilisant des configurations locales

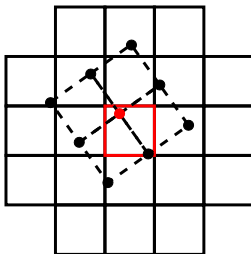
## Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$



$N_8(\mathbf{p})$



$N_{20}(T^{-1}(\mathbf{p}))$

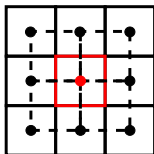


# Méthode utilisant des configurations locales

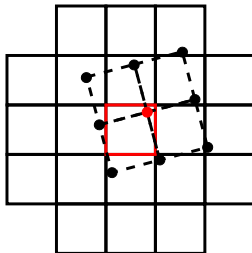
## Propriété locale

Soit  $\mathbf{p} \in \mathbb{Z}^2$  un pixel et  $\mathbf{q} \in N_8(\mathbf{p})$ . Nous avons

$$T^{-1}(\mathbf{q}) \in N_{20}(T^{-1}(\mathbf{p}))$$



$N_8(\mathbf{p})$



$N_{20}(T^{-1}(\mathbf{p}))$

# Expériences en imagerie rétinienne : variance topologique