

Une approche combinatoire pour le recalage rigide d'images en deux dimensions

Yukiko Kenmochi

En collaboration avec :

Phuc Ngo Nicolas Passat Akihiro Sugimoto Hugues Talbot



15 novembre 2013

Pourquoi une approche combinatoire ?

État de l'art

- Le problème du recalage rigide d'images est formulé en général dans le cadre **continu**, et dans un contexte d'**optimisation**.

Pourquoi une approche combinatoire ?

État de l'art

- Le problème du recalage rigide d'images est formulé en général dans le cadre **continu**, et dans un contexte d'**optimisation**.
- Cela exige souvent de **re-discrétiser** des résultats après leur transformation, et provoque parfois des artefacts indésirables.

Pourquoi une approche combinatoire ?

État de l'art

- Le problème du recalage rigide d'images est formulé en général dans le cadre **continu**, et dans un contexte d'**optimisation**.
- Cela exige souvent de **re-discrétiser** des résultats après leur transformation, et provoque parfois des artefacts indésirables.

Motivation et indices

- Est-il possible d'éviter cette re-discrétisation ?

Pourquoi une approche combinatoire ?

État de l'art

- Le problème du recalage rigide d'images est formulé en général dans le cadre **continu**, et dans un contexte d'**optimisation**.
- Cela exige souvent de **re-discrétiser** des résultats après leur transformation, et provoque parfois des artefacts indésirables.

Motivation et indices

- Est-il possible d'éviter cette re-discrétisation ?
Oui, on peut utiliser une **approche purement discrète** qui permet de transformer une image pixel par pixel.

Pourquoi une approche combinatoire ?

État de l'art

- Le problème du recalage rigide d'images est formulé en général dans le cadre **continu**, et dans un contexte d'**optimisation**.
- Cela exige souvent de **re-discrétiser** des résultats après leur transformation, et provoque parfois des artefacts indésirables.

Motivation et indices

- Est-il possible d'éviter cette re-discrétisation ?
Oui, on peut utiliser une **approche purement discrète** qui permet de transformer une image pixel par pixel.
- Comment **explorer** explicitement un tel espace discret de solutions ?

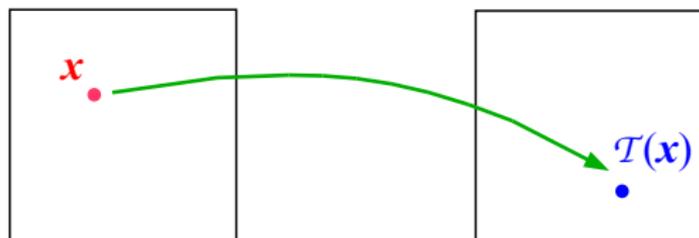
Transformation rigide sur \mathbb{R}^2

Définition (Transformation rigide $\mathcal{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$)

Une transformation rigide est une bijection, définie pour tout $\mathbf{x} = (x, y) \in \mathbb{R}^2$, telle que :

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

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



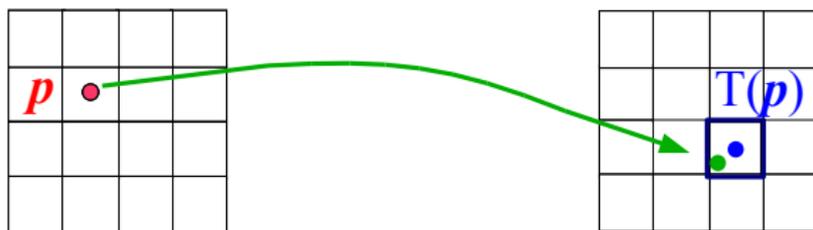
Transformation rigide sur \mathbb{Z}^2

Définition (Transformation rigide digitale $T : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$)

Une transformation rigide digitale est une fonction définie pour tout $\mathbf{p} = (p, q) \in \mathbb{Z}^2$ telle que :

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

où $D : \mathbb{R}^2 \rightarrow \mathbb{Z}^2$ est une discrétisation (un arrondi).

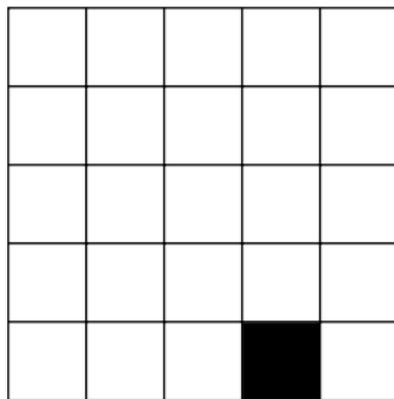
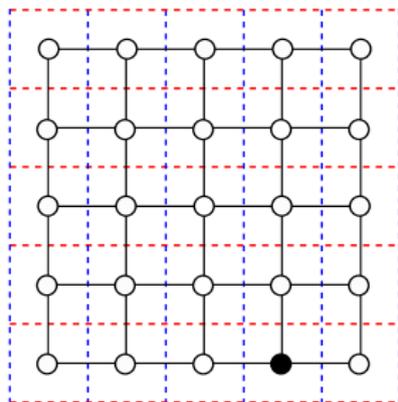


Discontinuité des transformations rigides sur \mathbb{Z}^2

Les transformations rigides sur \mathbb{Z}^2 :

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

sont **discontinues**.

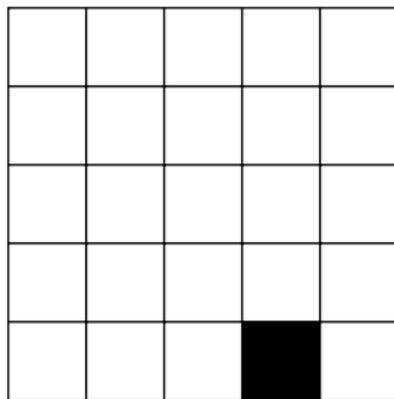
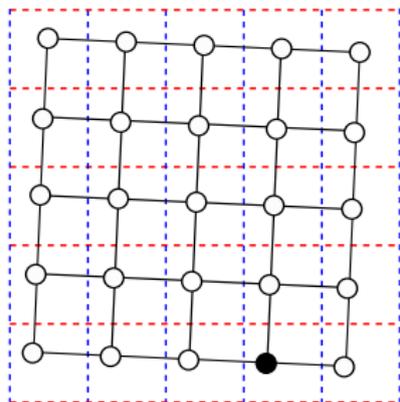


Discontinuité des transformations rigides sur \mathbb{Z}^2

Les transformations rigides sur \mathbb{Z}^2 :

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

sont **discontinues**.

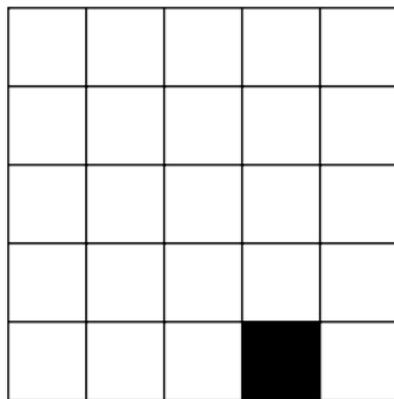
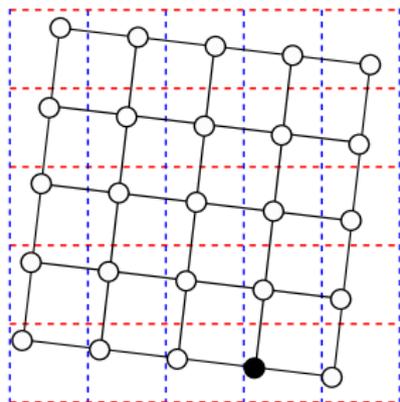


Discontinuité des transformations rigides sur \mathbb{Z}^2

Les transformations rigides sur \mathbb{Z}^2 :

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

sont **discontinues**.

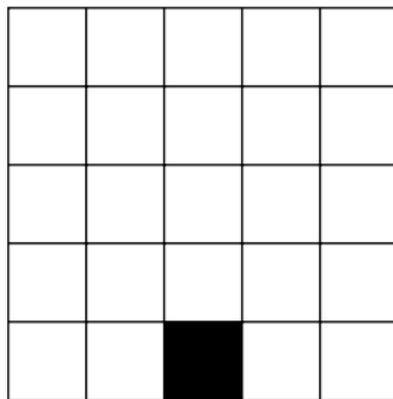
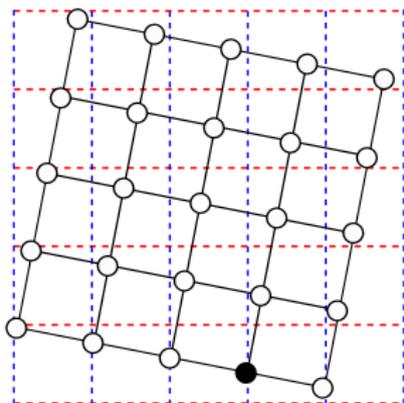


Discontinuité des transformations rigides sur \mathbb{Z}^2

Les transformations rigides sur \mathbb{Z}^2 :

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

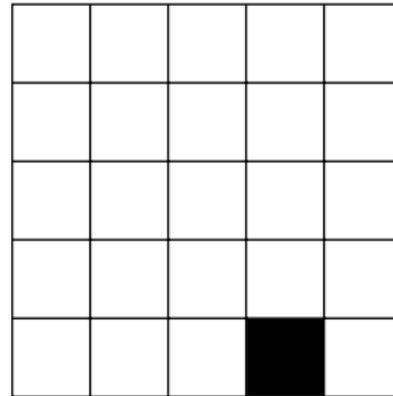
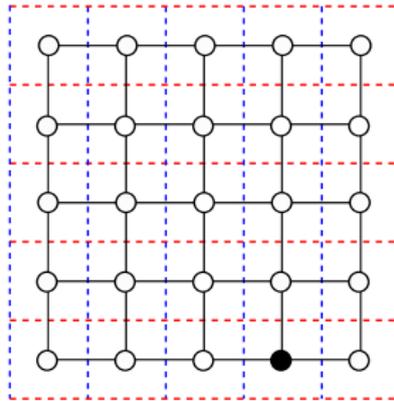
sont **discontinues**.



Transformation rigide discrète

Définition

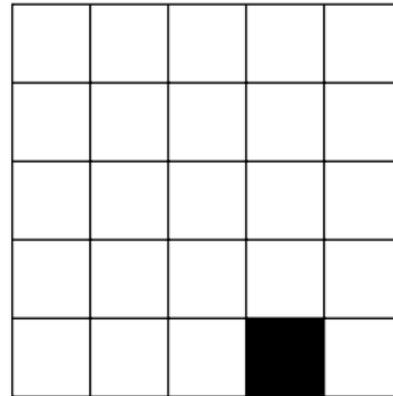
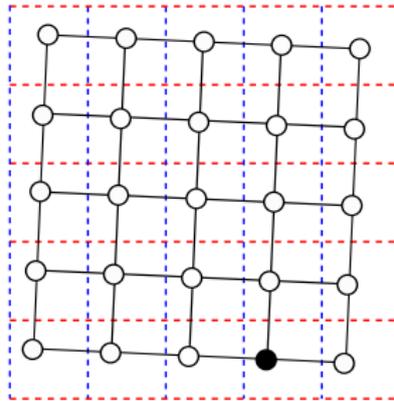
Une **transformation rigide discrète (TRD)** est un ensemble de toutes les transformations rigides qui génèrent la même image.



Transformation rigide discrète

Définition

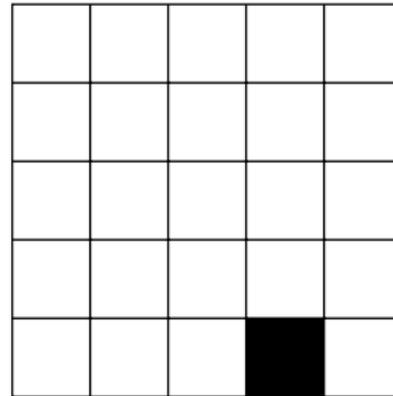
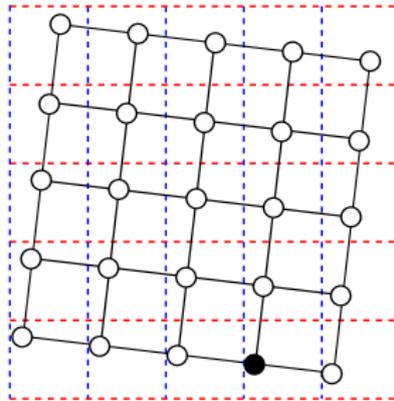
Une **transformation rigide discrète (TRD)** est un ensemble de toutes les transformations rigides qui génèrent la même image.



Transformation rigide discrète

Définition

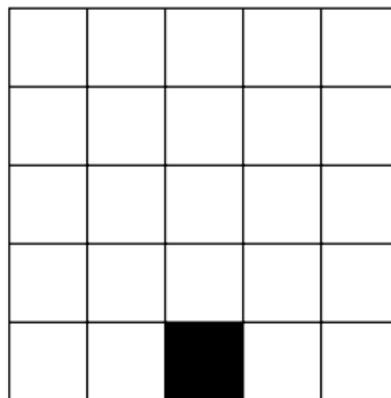
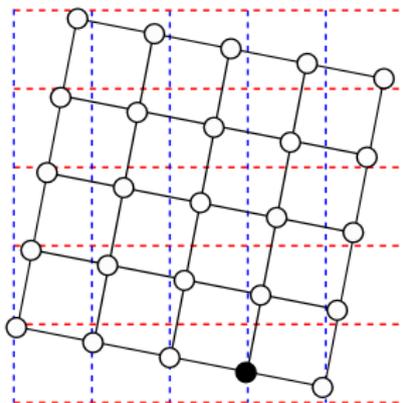
Une **transformation rigide discrète (TRD)** est un ensemble de toutes les transformations rigides qui génèrent la même image.



Transformation rigide discrète

Définition

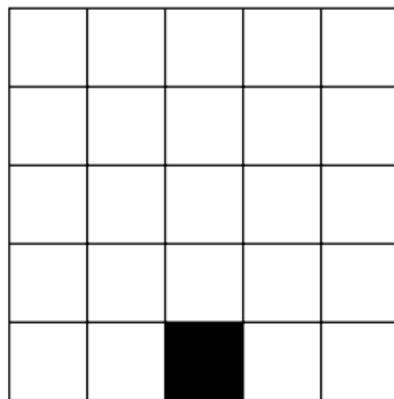
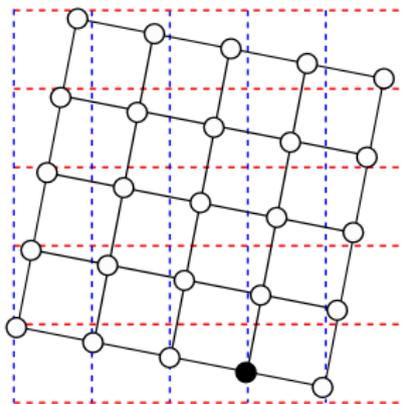
Une **transformation rigide discrète (TRD)** est un ensemble de toutes les transformations rigides qui génèrent la même image.



Transformation rigide discrète

Définition

Une **transformation rigide discrète (TRD)** est un ensemble de toutes les transformations rigides qui génèrent la même image.

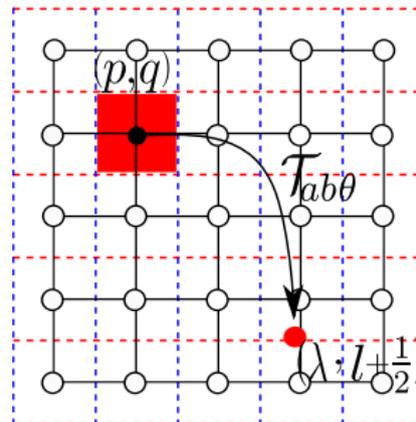
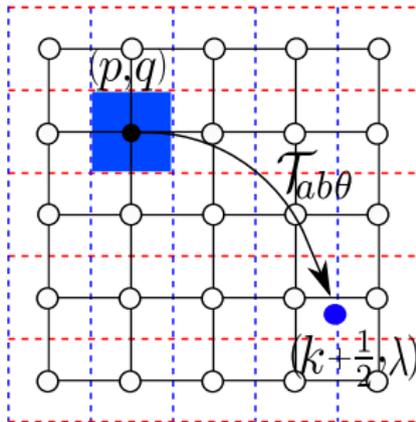


L'espace des paramètres (a, b, θ) est partitionné en des ensembles disjoints de transformations continues.

Transformation rigide critique

Définition

Une transformation rigide est **critique** si elle projette au moins un point de \mathbb{Z}^2 sur la semi-grille *verticale* ou *horizontale*.



Surfaces de basculement

Définition

Les **surfaces de basculement** sont des surfaces associées à des transformations critiques dans l'espace des paramètres (a, b, θ) :

$$a = k + \frac{1}{2} + q \sin \theta - p \cos \theta, \quad (\text{verticales})$$

$$b = l + \frac{1}{2} - p \sin \theta - q \cos \theta, \quad (\text{horizontales})$$

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, θ) :

$$a = k + \frac{1}{2} + q \sin \theta - p \cos \theta, \quad (\text{verticales})$$

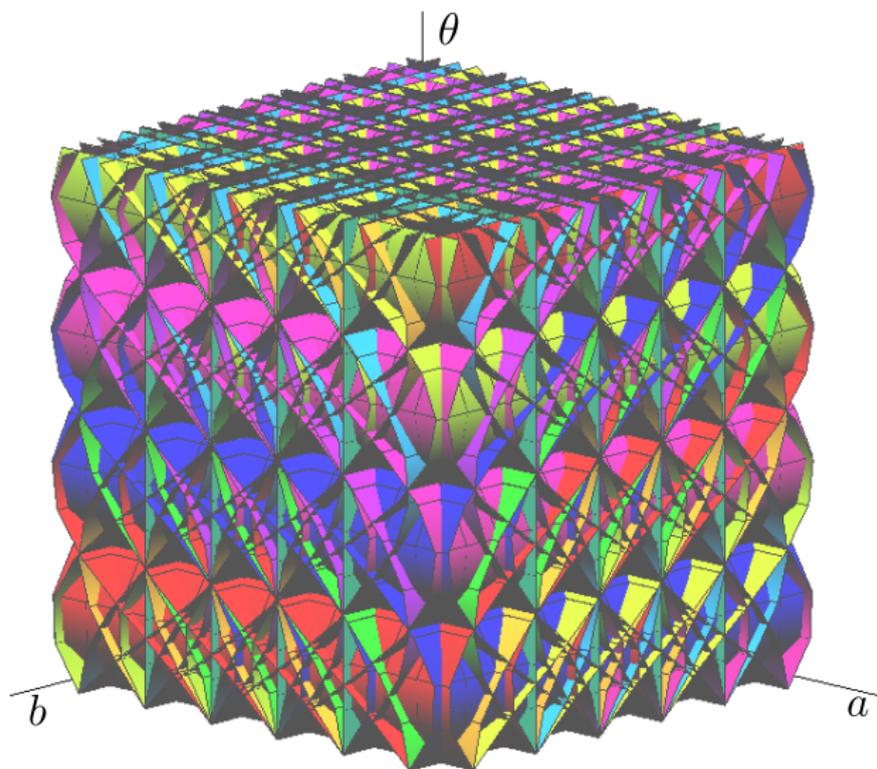
$$b = l + \frac{1}{2} - p \sin \theta - q \cos \theta, \quad (\text{horizontales})$$

où $p, q, k, l \in \mathbb{Z}$.

Chaque **surface de basculement** :

- est indexée par un **triplet d'entiers** (p, q, k) (resp. (p, q, l));
- indique que le pixel (p, q) dans une image transformée **change sa valeur** de celle du pixel $(k, *)$ (resp. $(*, l)$) dans l'image initiale à celle du pixel $(k + 1, *)$ (resp. $(*, l + 1)$).

Exemple de surfaces de basculement



Surfaces verticales Φ_{pqk} et horizontales Ψ_{pql} pour $p, q \in [0, 2]$ et $k, l \in [0, 3]$



Courbes de basculement

Définition

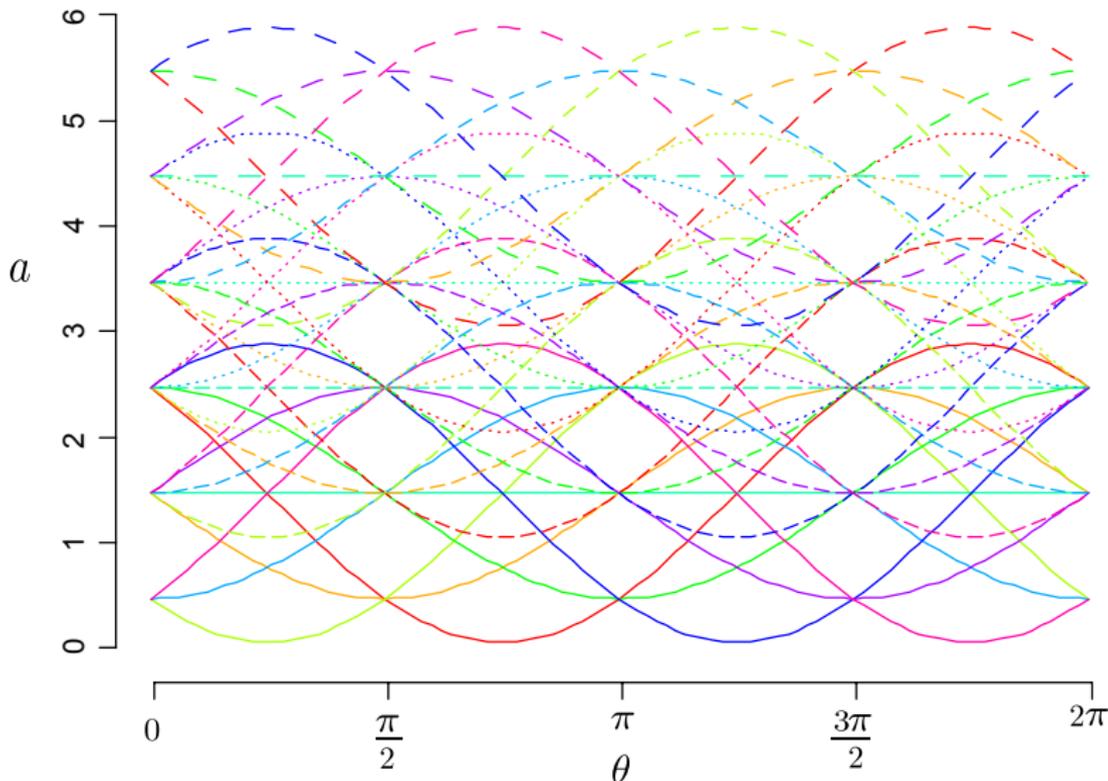
Les ***courbes de basculement*** sont les sections orthogonales des surfaces ***verticales*** (resp. ***horizontales***) de basculement vis-à-vis de l'axe ***b*** (resp. l'axe ***a***) sur le plan :

$$a = k + \frac{1}{2} + q \sin \theta - p \cos \theta, \quad (\text{verticales})$$

$$b = l + \frac{1}{2} - p \sin \theta - q \cos \theta, \quad (\text{horizontales})$$

où $p, q, k, l \in \mathbb{Z}$.

Exemple de courbes de basculement



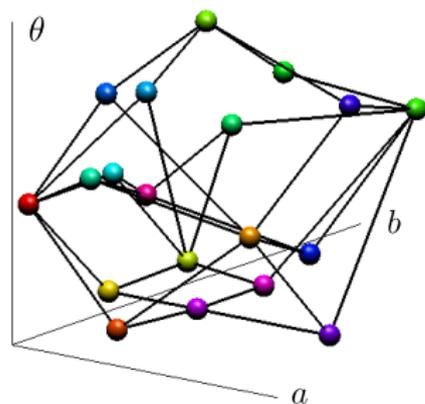
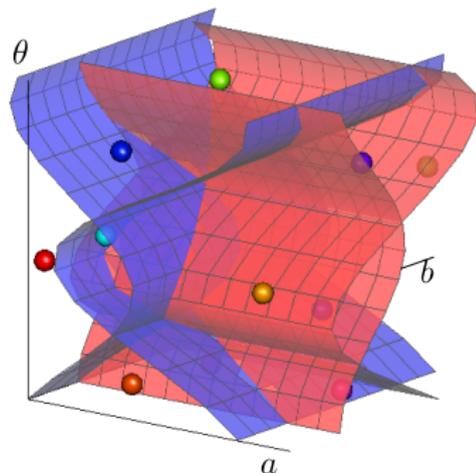
Courbes verticales de basculement ϕ_{pqk} pour $p, q \in [0, 2]$ et $k \in [0, 3]$.

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 $v \in V$ correspond à une TRD,
- chaque arête $e \in E$ relie deux sommets partageant une surface de basculement.



Propriétés du graphe des transformations rigides discrètes

Avantages

- Les transformations rigides discrètes sont **calculées dans un processus purement discret**.

Propriétés du graphe des transformations rigides discrètes

Avantages

- Les transformations rigides discrètes sont **calculées dans un processus purement discret**.
- Leur **structure combinatoire** est représentée par un **graphe des TRD G** dont la complexité est $O(N^9)$ pour les images de taille $N \times N$.

Propriétés du graphe des transformations rigides discrètes

Avantages

- Les transformations rigides discrètes sont **calculées dans un processus purement discret**.
- Leur **structure combinatoire** est représentée par un **graphe des TRD G** dont la complexité est $O(N^9)$ pour les images de taille $N \times N$.
- G modélise toutes les transformations rigides discrètes avec leur **information topologique** de telle sorte que :
 - un sommet correspond à chaque image transformée ;
 - une arête correspond au changement d'un pixel, *i.e. une surface de basculement* (chaque arête possède l'information de cette transition de pixel).

Propriétés du graphe des transformations rigides discrètes

Avantages

- Les transformations rigides discrètes sont **calculées dans un processus purement discret**.
- Leur **structure combinatoire** est représentée par un **graphe des TRD G** dont la complexité est $O(N^9)$ pour les images de taille $N \times N$.
- G modélise toutes les transformations rigides discrètes avec leur **information topologique** de telle sorte que :
 - un sommet correspond à chaque image transformée ;
 - une arête correspond au changement d'un pixel, *i.e. une surface de basculement* (chaque arête possède l'information de cette transition de pixel).
- Il permet de **générer de manière exhaustive et progressive toutes les images transformées** en temps linéaire.

Recalage comme un problème d'optimisation combinatoire

Formulation du problème

Étant données deux images A et B dont la taille est $N \times N$, le recalage d'images consiste à trouver une TRD telle que :

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

où $\mathbb{T} = \{T_v \mid v \in V\}$ est l'ensemble de tous les TRD, et d est une distance déterminée entre deux images.

Recalage comme un problème d'optimisation combinatoire

Formulation du problème

Étant données deux images A et B dont la taille est $N \times N$, le recalage d'images consiste à trouver une TRD telle que :

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

où $\mathbb{T} = \{T_v \mid v \in V\}$ est l'ensemble de tous les TRD, et d est une distance déterminée entre deux images.

On a un choix pour d ; ici, on utilise la **distance signée**. (Boykov *et al.*, 2006)

Recalage comme un problème d'optimisation combinatoire

Formulation du problème

Étant données deux images A et B dont la taille est $N \times N$, le recalage d'images consiste à trouver une TRD telle que :

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

où $\mathbb{T} = \{T_v \mid v \in V\}$ est l'ensemble de tous les TRD, et d est une distance déterminée entre deux images.

On a un choix pour d ; ici, on utilise la **distance signée**. (Boykov *et al.*, 2006)

Point faible

La recherche exhaustive sur un graphe des TRD coûte $O(N^9)$.

Recalage comme un problème d'optimisation combinatoire

Formulation du problème

Étant données deux images A et B dont la taille est $N \times N$, le recalage d'images consiste à trouver une TRD telle que :

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

où $\mathbb{T} = \{T_v \mid v \in V\}$ est l'ensemble de tous les TRD, et d est une distance déterminée entre deux images.

On a un choix pour d ; ici, on utilise la **distance signée**. (Boykov *et al.*, 2006)

Point faible

La recherche exhaustive sur un graphe des TRD coûte $O(N^9)$.

Point fort

Une recherche locale sur un graphe des TRD peut déterminer un **optimum local**.

Recherche locale sur un graphe des TRD

Algorithme (Recherche locale)

- **Entrée** : Une TRD initiale $v_0 \in V$.
- **Sortie** : Un optimum local $\hat{v} \in V$.
- **Approche** : Descente de gradient sur un graphe des TRD.

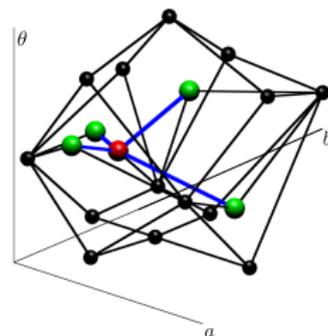
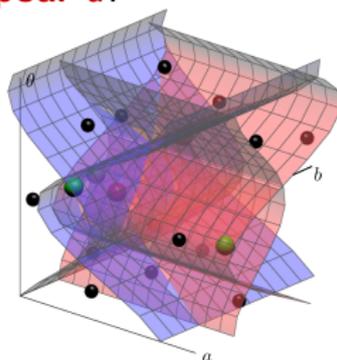
Recherche locale sur un graphe des TRD

Algorithme (Recherche locale)

- **Entrée** : Une TRD initiale $v_0 \in V$.
- **Sortie** : Un optimum local $\hat{v} \in V$.
- **Approche** : Descente de gradient sur un graphe des TRD.

Le graphe des TRD offre :

- la relation de voisinage $N(v)$,
- un calcul efficace pour d .



Construction d'un voisinage sur un graphe des TRD

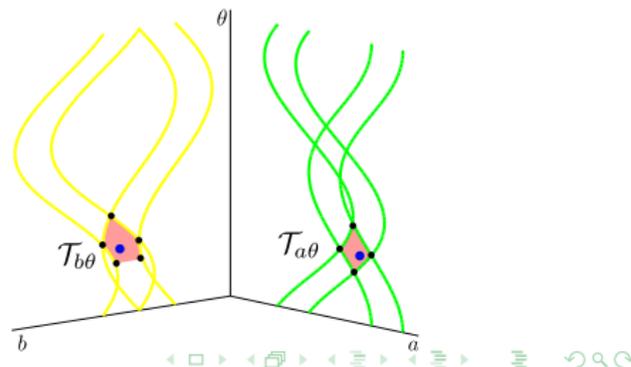
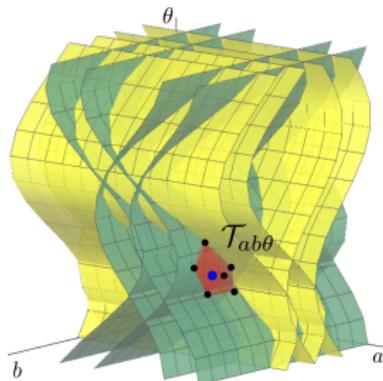
Algorithme (Obtenir les voisins à partir d'un ensemble de paramètres)

- **Entrée** : Une TRD v et le triplet de ses paramètres représentatifs $P(v) = (a_v, b_v, \theta_v)$.
- **Sortie** : Les voisins $N(v)$.

Construction d'un voisinage sur un graphe des TRD

Algorithme (Obtenir les voisins à partir d'un ensemble de paramètres)

- **Entrée** : Une TRD v et le triplet de ses paramètres représentatifs $P(v) = (a_v, b_v, \theta_v)$.
- **Sortie** : Les voisins $N(v)$.
- **Approche** : Dans l'espace des paramètres, trouver les surfaces de basculement les plus proches (i.e. les arêtes dans le graphe) autour de $P(v)$ en utilisant les propriétés des TRDs duales, i.e. cellules 3D (les a - et b -convexités (Ngo et al., 2013)).



Construction d'un voisinage sur un graphe des TRD

Algorithme (Obtenir les voisins à partir d'un ensemble de paramètres)

- **Entrée** : Une TRD v et le triplet de ses paramètres représentatifs $P(v) = (a_v, b_v, \theta_v)$.
- **Sortie** : Les voisins $N(v)$.
- **Approche** : Dans l'espace des paramètres, trouver les surfaces de basculement les plus proches (i.e. les arêtes dans le graphe) autour de $P(v)$ en utilisant les propriétés des TRDs duales, i.e. cellules 3D (les a - et b -convexités (Ngo et al., 2013)).

Complexité : $O(mN^2)$ où m est la taille de $N(v)$.

Construction d'un voisinage sur un graphe des TRD (suite)

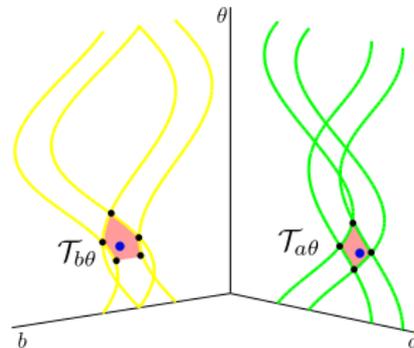
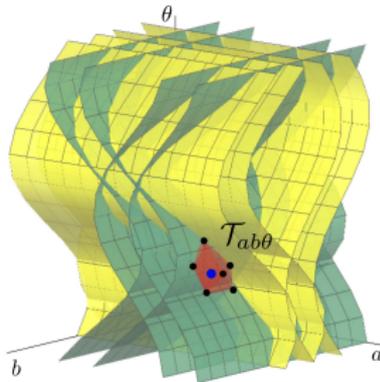
Algorithme (Trouver un triplet de paramètres associé à une TRD)

- **Entrée** : Une TRD v .
- **Sortie** : Le triplet représentatif de ses valeurs de paramètres $P(v) = (a_v, b_v, \theta_v)$.

Construction d'un voisinage sur un graphe des TRD (suite)

Algorithme (Trouver un triplet de paramètres associé à une TRD)

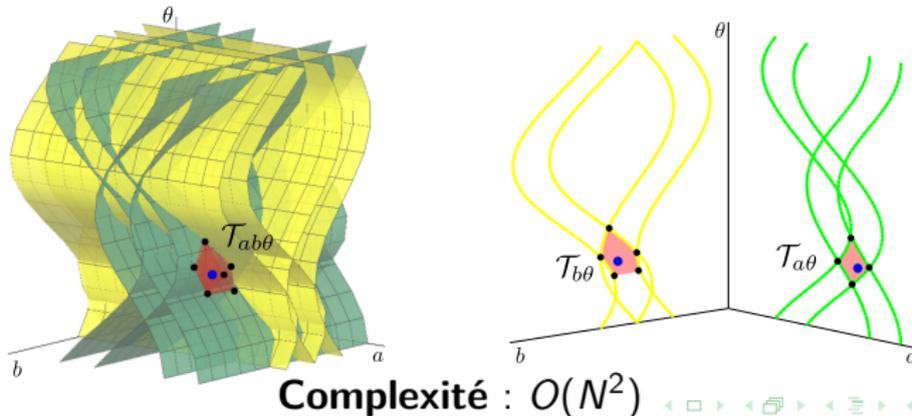
- **Entrée** : Une TRD v .
- **Sortie** : Le triplet représentatif de ses valeurs de paramètres $P(v) = (a_v, b_v, \theta_v)$.
- **Approche** : Dans l'espace des paramètres, trouver $P(v)$ à partir de $4N^2$ surfaces de basculement qui sont associées à v en utilisant les propriétés des TRDs duales, i.e. cellules 3D (les a - et b -convexités (Ngo et al., 2013)).



Construction d'un voisinage sur un graphe des TRD (suite)

Algorithme (Trouver un triplet de paramètres associé à une TRD)

- **Entrée** : Une TRD v .
- **Sortie** : Le triplet représentatif de ses valeurs de paramètres $P(v) = (a_v, b_v, \theta_v)$.
- **Approche** : Dans l'espace des paramètres, trouver $P(v)$ à partir de $4N^2$ surfaces de basculement qui sont associées à v en utilisant les propriétés des TRDs duales, i.e. cellules 3D (les a - et b -convexités (Ngo et al., 2013)).



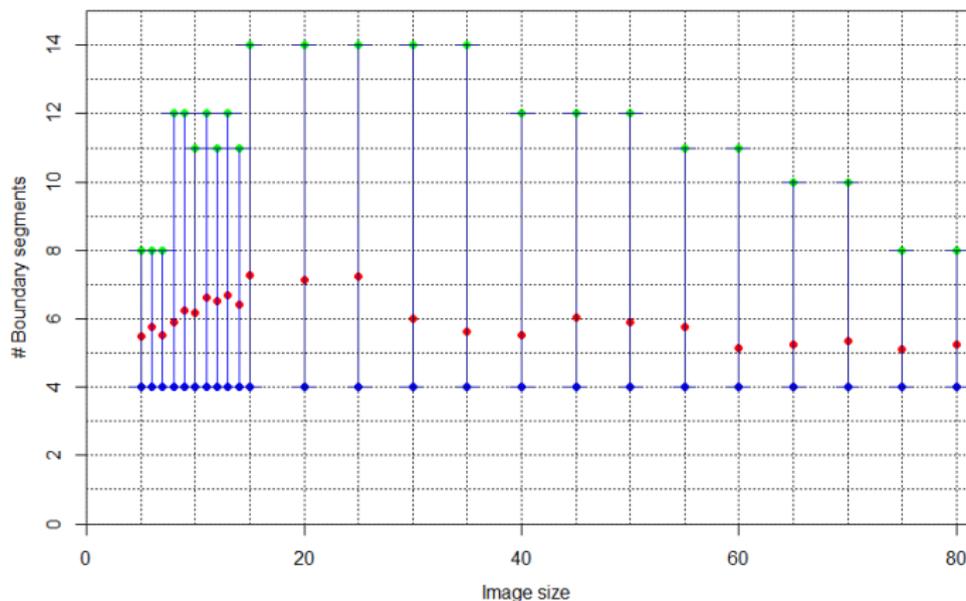
Complexité : $O(N^2)$

Taille du voisinage du graphe des TRD

- En théorie, la taille du voisinage du graphe des TRD $m = \mathbf{O(N^2)}$.

Taille du voisinage du graphe des TRD

- En théorie, la taille du voisinage du graphe des TRD $m = O(N^2)$.
- En pratique, m est **bornée par une petite constante**.
(Observations sur 960 expériences pour des images de tailles variant de 5×5 à 80×80 .)

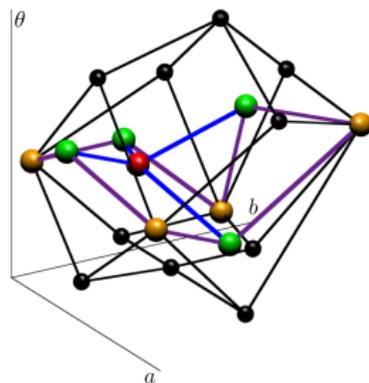
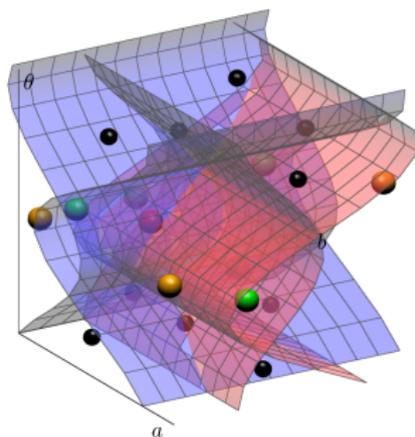


k -voisinage et sa construction

Définition

$$N^k(v) = N^{k-1}(v) \cup \bigcup_{u \in N^{k-1}(v)} N(u)$$

où $N^1(v) = N(v)$.

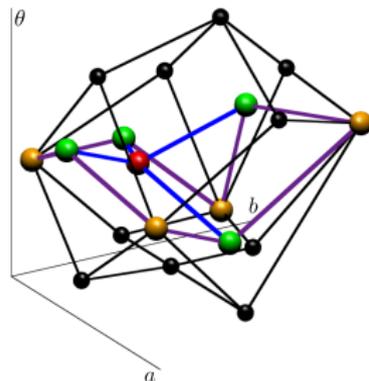
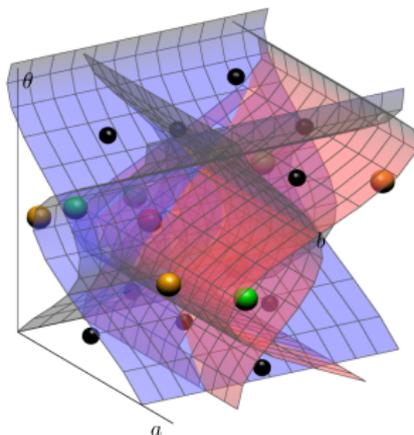


k -voisinage et sa construction

Définition

$$N^k(v) = N^{k-1}(v) \cup \bigcup_{u \in N^{k-1}(v)} N(u)$$

où $N^1(v) = N(v)$.

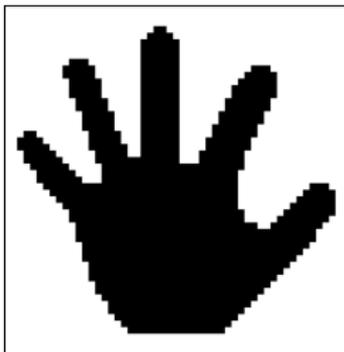


Complexité : $O(m^k N^2)$ où m est la taille du 1-voisinage.

Expérience 1



(a) Image de référence



(b) Image cible



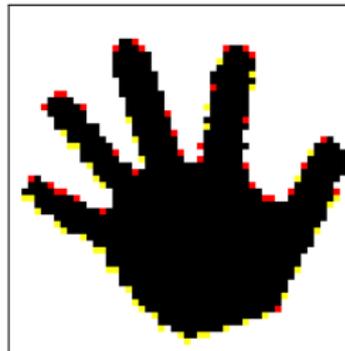
(c) Solution initiale



(d) 1-voisinage

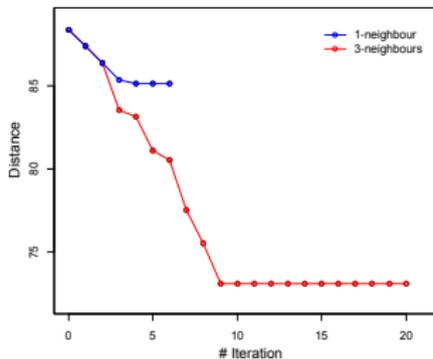
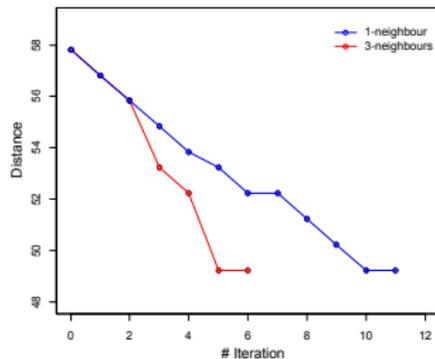
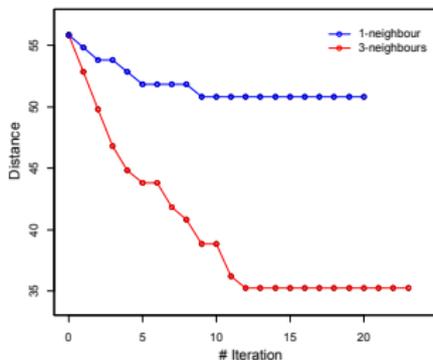
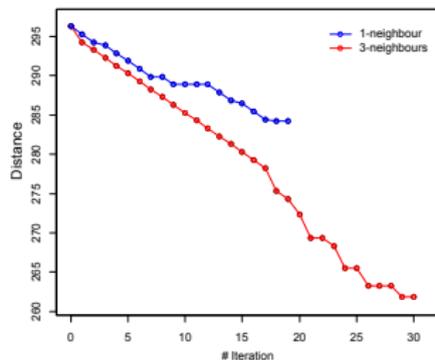


(e) 3-voisinage



(f) (d) \ (e)

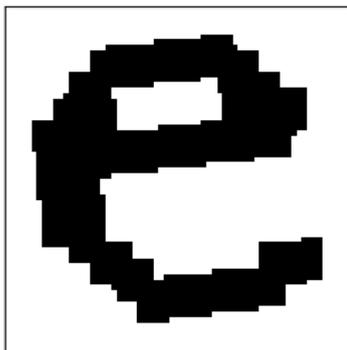
Expérience 1 (suite)

(a) Solution initiale $(0, 0, 0.1)$ (b) $(0.49, 0.35, 0.15)$ (c) $(0.12, 0.05, 0.1314)$ (d) $(0.52, 0.79, 0.3107)$

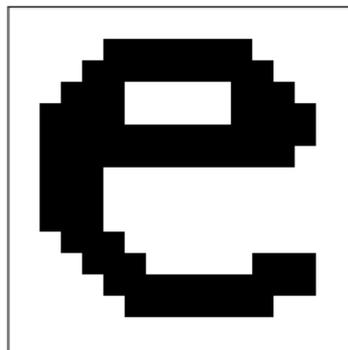
Expérience 1 (suite)

Une sequence d'images transformées en utilisant le 3-voisinage.

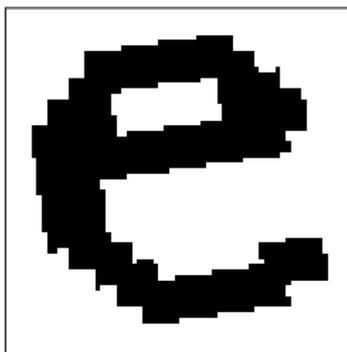
Expérience 2



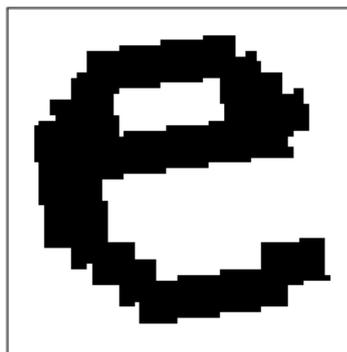
(a) Image de référence



(b) Image cible

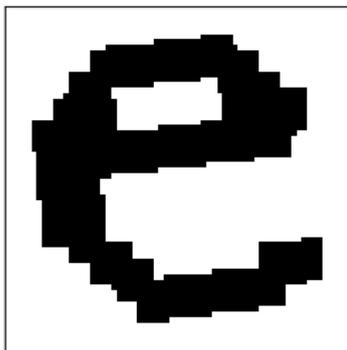


(c) 1-voisinage

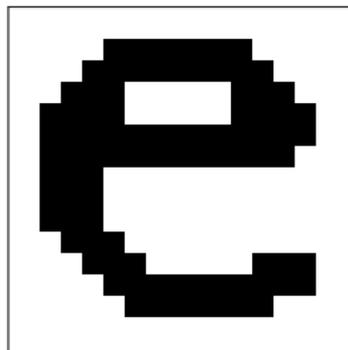


(d) 3-voisinage

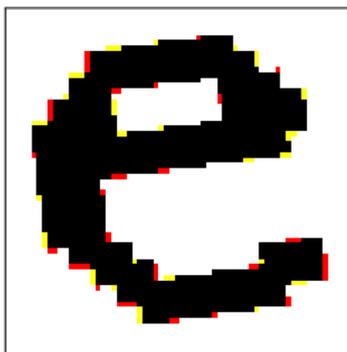
Expérience 2



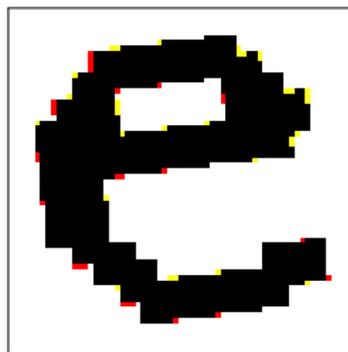
(a) Image de référence



(b) Image cible



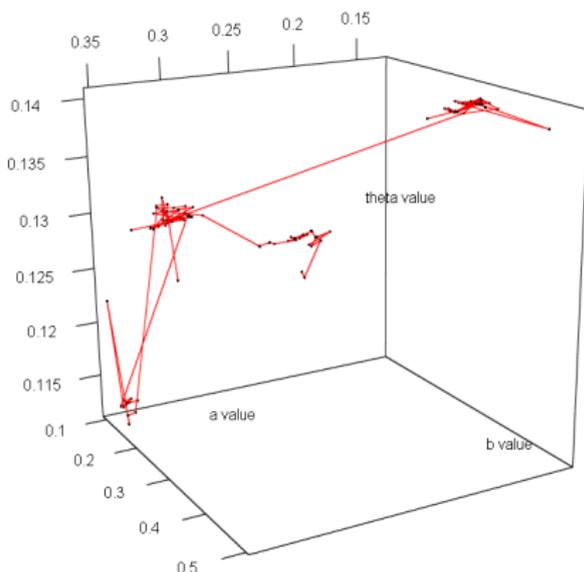
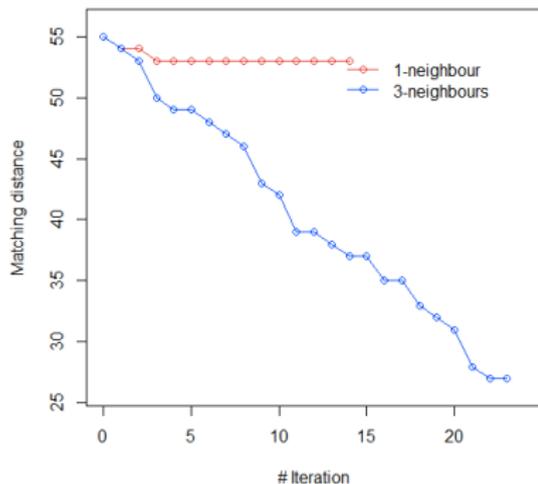
(c) 1-voisinage



(d) 3-voisinage

Expérience 2 (suite)

Les séquences de distances et de paramètres de transformations rigides (uniquement pour le 3-voisinage) pendant l'itération.



Conclusion et perspectives

Conclusion

- Un cadre purement discret pour le recalage rigide d'images 2D, qui est basé sur le graphe des TRD, a été proposé.

Conclusion et perspectives

Conclusion

- Un cadre purement discret pour le recalage rigide d'images 2D, qui est basé sur le graphe des TRD, a été proposé.
- Une procédure de descente de gradient construit le k -voisinage exact à chaque étape dont la complexité en temps est $O(m^k N^2)$.

Conclusion et perspectives

Conclusion

- Un cadre purement discret pour le recalage rigide d'images 2D, qui est basé sur le graphe des TRD, a été proposé.
- Une procédure de descente de gradient construit le k -voisinage exact à chaque étape dont la complexité en temps est $O(m^k N^2)$.
- En pratique, m se comporte comme une constante.

Conclusion et perspectives

Conclusion

- Un cadre purement discret pour le recalage rigide d'images 2D, qui est basé sur le graphe des TRD, a été proposé.
- Une procédure de descente de gradient construit le k -voisinage exact à chaque étape dont la complexité en temps est $O(m^k N^2)$.
- En pratique, m se comporte comme une constante.
- Un algorithme efficace qui calcule les voisins pour la recherche locale a été proposé avec une complexité de temps presque linéaire.

Conclusion et perspectives

Conclusion

- Un cadre purement discret pour le recalage rigide d'images 2D, qui est basé sur le graphe des TRD, a été proposé.
- Une procédure de descente de gradient construit le k -voisinage exact à chaque étape dont la complexité en temps est $O(m^k N^2)$.
- En pratique, m se comporte comme une constante.
- Un algorithme efficace qui calcule les voisins pour la recherche locale a été proposé avec une complexité de temps presque linéaire.

Perspectives

- Améliorer théoriquement la borne supérieure pour m , qui est actuellement en $O(N^2)$.

Conclusion et perspectives

Conclusion

- Un cadre purement discret pour le recalage rigide d'images 2D, qui est basé sur le graphe des TRD, a été proposé.
- Une procédure de descente de gradient construit le k -voisinage exact à chaque étape dont la complexité en temps est $O(m^k N^2)$.
- En pratique, m se comporte comme une constante.
- Un algorithme efficace qui calcule les voisins pour la recherche locale a été proposé avec une complexité de temps presque linéaire.

Perspectives

- Améliorer théoriquement la borne supérieure pour m , qui est actuellement en $O(N^2)$.
- Combiner notre méthode avec d'autres approches combinatoires.

Conclusion et perspectives

Conclusion

- Un cadre purement discret pour le recalage rigide d'images 2D, qui est basé sur le graphe des TRD, a été proposé.
- Une procédure de descente de gradient construit le k -voisinage exact à chaque étape dont la complexité en temps est $O(m^k N^2)$.
- En pratique, m se comporte comme une constante.
- Un algorithme efficace qui calcule les voisins pour la recherche locale a été proposé avec une complexité de temps presque linéaire.

Perspectives

- Améliorer théoriquement la borne supérieure pour m , qui est actuellement en $O(N^2)$.
- Combiner notre méthode avec d'autres approches combinatoires.
- Étendre notre méthode aux dimensions supérieures, et aux images en niveaux de gris ou labélisées.