Discrete rigid transformation graph search for 2D image registration

Phuc Ngo Akihiro Sugimoto Yukiko Kenmochi
Nicolas Passat Hugues Talbot

October 29, 2013

Why discrete rigid transformation graph search ?

Background

- Problem of rigid image registration is usually formulated in the **continuous** domain, in the context of optimization.

Why discrete rigid transformation graph search ?

Background

- Problem of rigid image registration is usually formulated in the **continuous** domain, in the context of optimization.
- This often requires **re-digitizing** results after transformation, and sometimes causes unwanted artifacts.

Why discrete rigid transformation graph search ?

Background

- Problem of rigid image registration is usually formulated in the **continuous** domain, in the context of optimization.
- This often requires **re-digitizing** results after transformation, and sometimes causes unwanted artifacts.

Motivation and clues

- Is it possible to avoid this re-digitization ?

Why discrete rigid transformation graph search ?

Background

- Problem of rigid image registration is usually formulated in the **continuous** domain, in the context of optimization.
- This often requires **re-digitizing** results after transformation, and sometimes causes unwanted artifacts.

Motivation and clues

- Is it possible to avoid this re-digitization ?

Yes, use a **fully discrete approach**
which allows to transform images pixel by pixel.

Why discrete rigid transformation graph search ?

Background

- Problem of rigid image registration is usually formulated in the **continuous** domain, in the context of optimization.
- This often requires **re-digitizing** results after transformation, and sometimes causes unwanted artifacts.

Motivation and clues

- Is it possible to avoid this re-digitization ?
  Yes, use a **fully discrete approach**
  which allows to transform images pixel by pixel.
- How to **explore** explicitely such a discrete parameter space ?

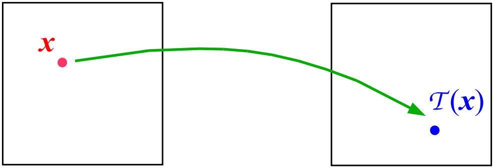
Rigid transformation on $\mathbb{R}^2$

Definition (Rigid transformation $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$)

A rigid transformation is a bijection defined for any $\boldsymbol{x} = (x, y) \in \mathbb{R}^2$, *as*

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

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



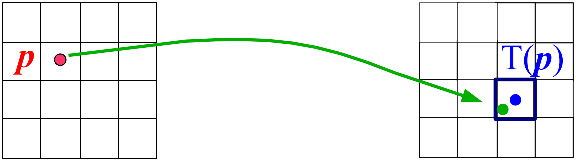
Rigid transformation on $\mathbb{Z}^2$

Definition (Digital rigid transformation $T : \mathbb{Z}^2 \to \mathbb{Z}^2$)

A digital rigid transformation on $\mathbb{Z}^2$ is defined for any $\boldsymbol{p} = (p, q) \in \mathbb{Z}^2$ as

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

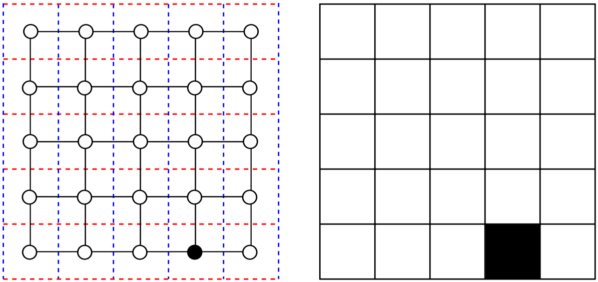
where $D : \mathbb{R}^2 \to \mathbb{Z}^2$ is digitization (a rounding function).



Discontinuities of rigid transformations in $\mathbb{Z}^2$

Discontinuities of rigid transformations on $\mathbb{Z}^2$

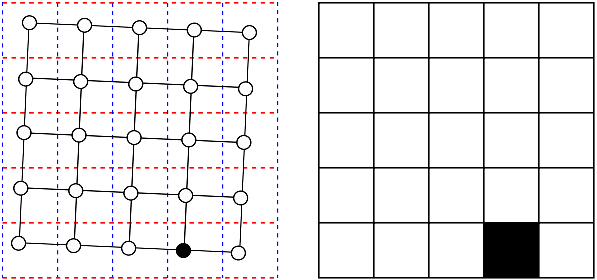
$$T(\mathbf{p}) = D \circ \mathcal{T}(\mathbf{p}) = \begin{pmatrix} [p\cos\theta - q\sin\theta + a] \\ [p\sin\theta + q\cos\theta + b] \end{pmatrix}$$



Discontinuities of rigid transformations in $\mathbb{Z}^2$

Discontinuities of rigid transformations on $\mathbb{Z}^2$

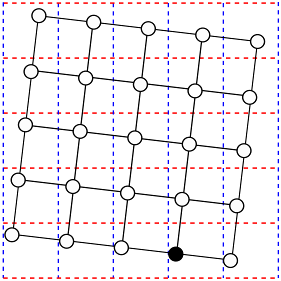
$$T(\boldsymbol{p}) = D \circ \mathcal{T}(\boldsymbol{p}) = \begin{pmatrix} [p\cos\theta - q\sin\theta + a] \\ [p\sin\theta + q\cos\theta + b] \end{pmatrix}$$

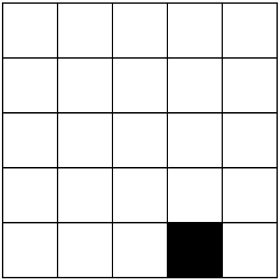


Discontinuities of rigid transformations in $\mathbb{Z}^2$

Discontinuities of rigid transformations on $\mathbb{Z}^2$

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

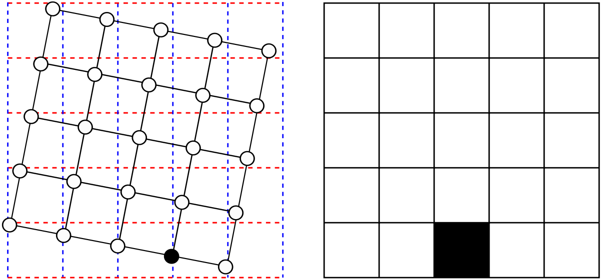




Discontinuities of rigid transformations in $\mathbb{Z}^2$

Discontinuities of rigid transformations on $\mathbb{Z}^2$

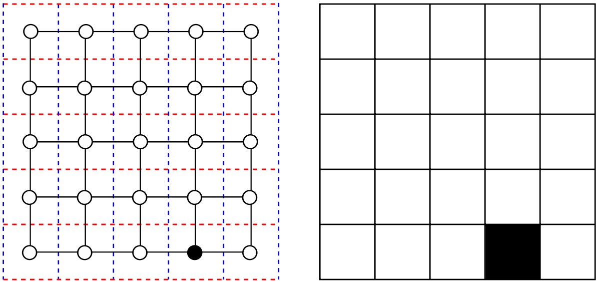
$$T(\boldsymbol{p}) = D \circ \mathcal{T}(\boldsymbol{p}) = \begin{pmatrix} [p\cos\theta - q\sin\theta + a] \\ [p\sin\theta + q\cos\theta + b] \end{pmatrix}$$



Discrete rigid transformation

Definition

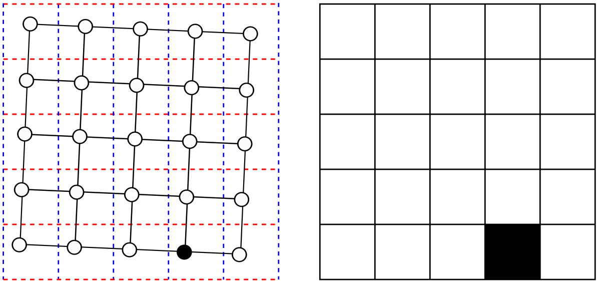
*A **discrete rigid transformation** (DRT) is a set of all the rigid transformations that generate the same image.*



Discrete rigid transformation

Definition

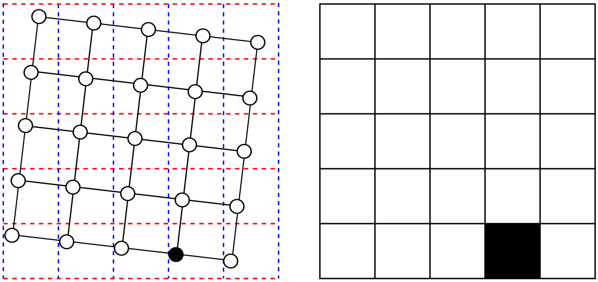
*A **discrete rigid transformation** (DRT) is a set of all the rigid transformations that generate the same image.*



Discrete rigid transformation

Definition

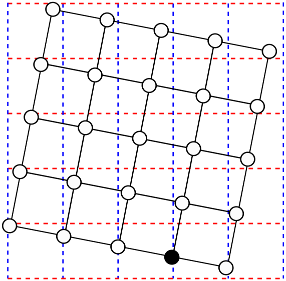
*A **discrete rigid transformation** (DRT) is a set of all the rigid transformations that generate the same image.*

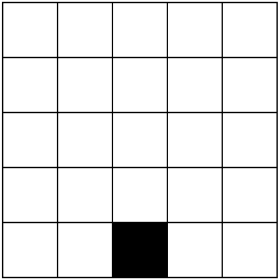


Discrete rigid transformation

Definition

*A **discrete rigid transformation** (DRT) is a set of all the rigid transformations that generate the same image.*

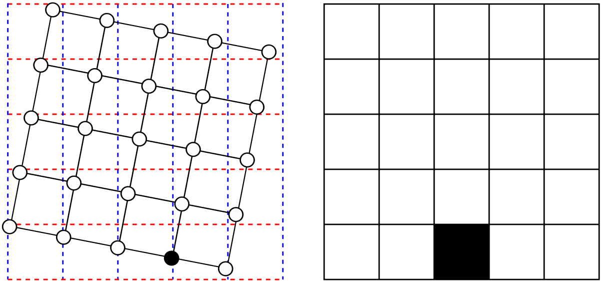




Discrete rigid transformation

Definition

A ***discrete rigid transformation*** *(DRT) is a set of all the rigid transformations that generate the same image.*

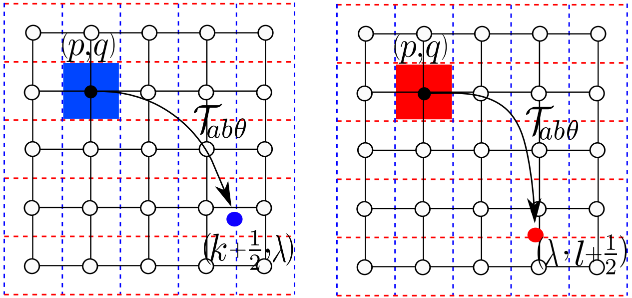


The parameter space (a, b, θ) is partitioned in the disjoint set of DRTs.

Critical rigid transformations

Definition

*A **critical rigid transformation** moves at least one point of $\mathbb{Z}^2$ to a point on the vertical or horizontal half-grid.*



Tipping surfaces

Definition

*The **tipping surfaces** are the surfaces associated to critical transformations in the parameter space* (a, b, θ) *:*

$$a = k + \frac{1}{2} + q\sin\theta - p\cos\theta, \qquad \textit{(vertical)}$$

$$b = l + \frac{1}{2} - p\sin\theta - q\cos\theta, \qquad \textit{(horizontal)}$$

for $p, q, k, l \in \mathbb{Z}$.

Tipping surfaces

Definition

*The **tipping surfaces** are the surfaces associated to critical transformations in the parameter space* (a, b, θ) *:*

$$a = k + \frac{1}{2} + q \sin\theta - p \cos\theta, \qquad \textit{(vertical)}$$

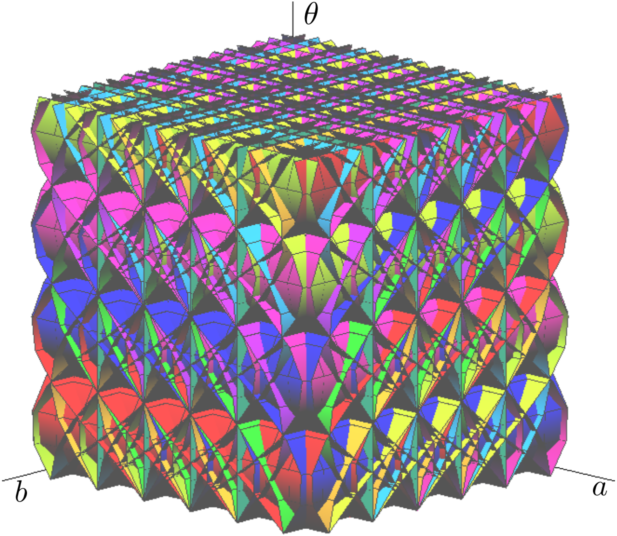
$$b = l + \frac{1}{2} - p \sin\theta - q \cos\theta, \qquad \textit{(horizontal)}$$

for $p, q, k, l \in \mathbb{Z}$.

Each **tipping surface**

- is indexed by a **triplet of integers** (p, q, k) (resp. (p, q, l)),
- indicates that the pixel (p, q) in a transformed image **changes its value** from the one at $(k, *)$ (resp. $(*, l)$) in an original image to the one at $(k + 1, *)$ (resp. $(*, l + 1)$).

Example of tipping surfaces



Vertical surfaces Φ_{pqk} and horizontal ones Ψ_{pql} for $p, q \in [0, 2]$ and $k, l \in [0, 3]$.

Tipping curves

Definition

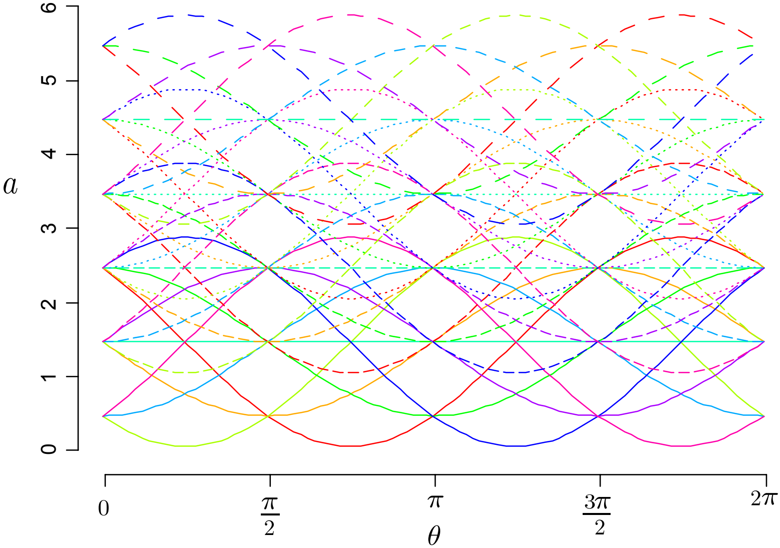
*The **tipping curves** are the orthogonal sections of vertical (resp. horizontal) tipping surfaces with respect to the axis a (resp. the axis b) on the plane :*

$$a = k + \frac{1}{2} + q \sin\theta - p \cos\theta, \qquad \textit{(vertical)}$$

$$b = l + \frac{1}{2} - p \sin\theta - q \cos\theta, \qquad \textit{(horizontal)}$$

for $p, q, k, l \in \mathbb{Z}$.

Example of tipping curves



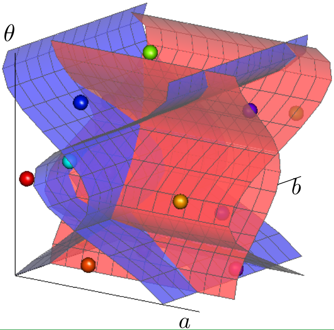
Vertical tipping curves ϕ_{pqk} for $p, q \in [0, 2]$ and $k \in [0, 3]$.

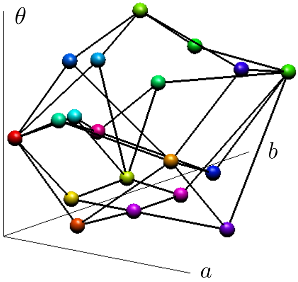
Graph of discrete rigid transformations

Definition

*A **graph of discrete rigid transformations** (DRT graph) is a graph $G = (V, E)$ such that :*

- *each vertex $v \in V$ corresponds to a DRT,*
- *each edge $e \in E$ connects two DRTs sharing a tipping surface.*





Properties of discrete rigid transformation graph

Advantages

- Discrete rigid transformations are **computed in a fully discrete process**,

Properties of discrete rigid transformation graph

Advantages

- Discrete rigid transformations are **computed in a fully discrete process**,
- Their **combinatorial structure** is represented by a **DRT graph** G whose complexity is $O(N^9)$ for images of size $N \times N$.

Properties of discrete rigid transformation graph

Advantages

- Discrete rigid transformations are **computed in a fully discrete process**,
- Their **combinatorial structure** is represented by a **DRT graph** G whose complexity is $O(N^9)$ for images of size $N \times N$.
- G models all the discrete rigid transformations with their **topological information** such that :
  - a vertex corresponds to each transformed image,
  - an edge corresponds to one pixel change, *i.e. a tipping surface*, (each edge posesses such pixel transition information).

Properties of discrete rigid transformation graph

Advantages

- Discrete rigid transformations are **computed in a fully discrete process**,
- Their **combinatorial structure** is represented by a **DRT graph** G whose complexity is $O(N^9)$ for images of size $N \times N$.
- G models all the discrete rigid transformations with their **topological information** such that :
  - a vertex corresponds to each transformed image,
  - an edge corresponds to one pixel change, *i.e. a tipping surface*, (each edge posesses such pixel transition information).
- It enables to **generate exhaustively and incrementally all the transformed images** in a linear complexity.

Registration as a combinatorial optimisation problem

Problem formulation

Given two images A and B of size $N \times N$, our image registration consists of finding a DRT such that

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

where $\mathbb{T} = \{ T_v \mid v \in V \}$ of all the DRTs, and d is a given distance between two images.

Registration as a combinatorial optimisation problem

Problem formulation

Given two images A and B of size $N \times N$, our image registration consists of finding a DRT such that

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

where $\mathbb{T} = \{ T_v \mid v \in V \}$ of all the DRTs, and d is a given distance between two images.

We have a choice for d ; here we use **signed distance**. (Boykov *et al.*, 2006)

Registration as a combinatorial optimisation problem

Problem formulation

Given two images A and B of size $N \times N$, our image registration consists of finding a DRT such that

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

where $\mathbb{T} = \{ T_v \mid v \in V \}$ of all the DRTs, and d is a given distance between two images.

We have a choice for d ; here we use **signed distance**. (Boykov *et al.*, 2006)

Disadvantage

Exhaustive search on DRT graph costs $O(N^9)$ in complexity.

Registration as a combinatorial optimisation problem

Problem formulation

Given two images A and B of size $N \times N$, our image registration consists of finding a DRT such that

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

where $\mathbb{T} = \{ T_v \mid v \in V \}$ of all the DRTs, and d is a given distance between two images.

We have a choice for d ; here we use **signed distance**. (Boykov *et al.*, 2006)

Disadvantage

Exhaustive search on DRT graph costs $O(N^9)$ in complexity.

Advantage

A local search on DRT graph can determine a **local optimum**.

Local search on discrete rigid transformation graph

Algorithm (Local search)

- ***Input*** : *An initial DRT* $v_0 \in V$.
- ***Output*** : *A local optimum* $\widehat{v} \in V$.
- ***Approach*** : *Gradient descent in a DRT graph* $G = (V, E)$.

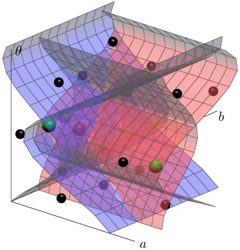
Local search on discrete rigid transformation graph

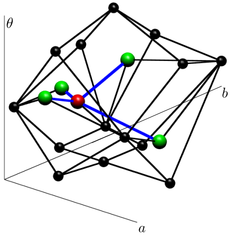
Algorithm (Local search)

- ***Input*** : *An initial DRT* $v_0 \in V$.
- ***Output*** : *A local optimum* $\widehat{v} \in V$.
- ***Approach*** : *Gradient descent in a DRT graph* $G = (V, E)$.

DRT graph provides

- **neighbourhood relation** $N(v)$,
- **efficient computation of** d.





Neighbourhood construction in a DRT graph

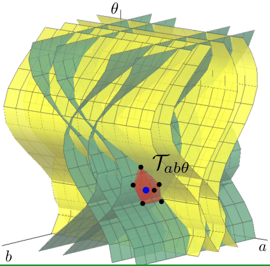
Algorithm (Obtain neighbourhood from a parameter set)

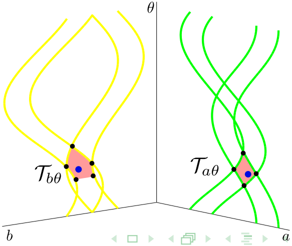
- ***Input*** : *A DRT* v *and its representative parameter triplet* $P(v) = (a_v, b_v, \theta_v)$
- ***Output*** : *Its neighbourhood* $N(v)$.

Neighbourhood construction in a DRT graph

Algorithm (Obtain neighbourhood from a parameter set)

- ***Input*** *: A DRT v and its representative parameter triplet* $P(v) = (a_v, b_v, \theta_v)$
- ***Output*** *: Its neighbourhood* $N(v)$.
- ***Approach*** *: In the (dual) parameter space, find the closest tipping surfaces (edges in the primal) around* $P(v)$ *by using the properties of dual DRTs, i.e. 3D cells (a- and b-convexity* (Ngo et al., 2013)*).*





Neighbourhood construction in a DRT graph

Algorithm (Obtain neighbourhood from a parameter set)

- ***Input*** : *A DRT v and its representative parameter triplet $P(v) = (a_v, b_v, \theta_v)$*
- ***Output*** : *Its neighbourhood $N(v)$.*
- ***Approach*** : *In the (dual) parameter space, find the closest tipping surfaces (edges in the primal) around $P(v)$ by using the properties of dual DRTs, i.e. 3D cells (a- and b-convexity (Ngo et al., 2013)).*

Complexity : $O(mN^2)$ where m is the size of $N(v)$.

Neighbourhood construction in a DRT graph (cont.)

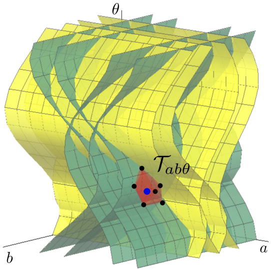
Algorithm (Find a representative parameter set associated to a DRT)

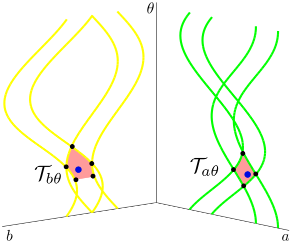
- ***Input*** : *A DRT* v.
- ***Output*** : *its representative parameter triplet* $P(v) = (a_v, b_v, \theta_v)$.

Neighbourhood construction in a DRT graph (cont.)

Algorithm (Find a representative parameter set associated to a DRT)

- ***Input*** *: A DRT v.*
- ***Output*** *: its representative parameter triplet $P(v) = (a_v, b_v, \theta_v)$.*
- ***Approach*** *: In the (dual) parameter space, find $P(v)$ from $4N^2$ tipping surfaces associated to v by using the properties of dual DRTs, i.e. 3D cells (a- and b-convexity (Ngo et al., 2013).)*

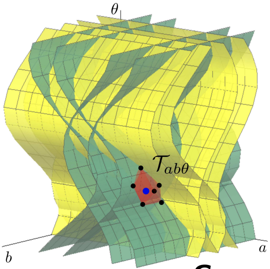


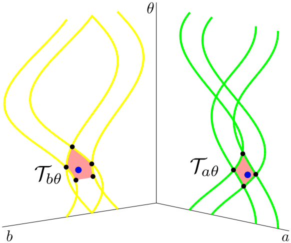


Neighbourhood construction in a DRT graph (cont.)

Algorithm (Find a representative parameter set associated to a DRT)

- ***Input*** : *A DRT v.*
- ***Output*** : *its representative parameter triplet $P(v) = (a_v, b_v, \theta_v)$.*
- ***Approach*** : *In the (dual) parameter space, find $P(v)$ from $4N^2$ tipping surfaces associated to v by using the properties of dual DRTs, i.e. 3D cells (a- and b-convexity* (Ngo et al., 2013)*.)*





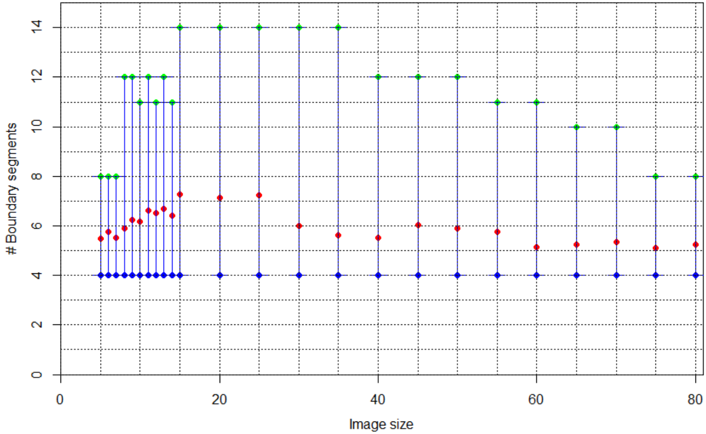
Complexity : $O(N^2)$

DRT graph neighbourhood size

- Theoretically, DRT graph neighbourhood size $m = \mathbf{O(N^2)}$.

DRT graph neighbourhood size

- Theoretically, DRT graph neighbourhood size $m = \mathbf{O(N^2)}$.
- In practice, m is **bounded by a small constant** (observed from 960 experiments for images of size from 5×5 to 80×80).

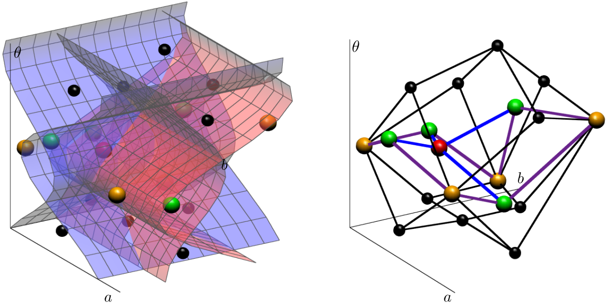


k-Neighbourhood and its construction

Definition

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

where $N^1(v) = N(v)$.

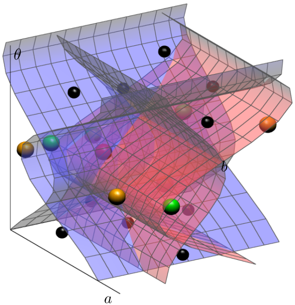


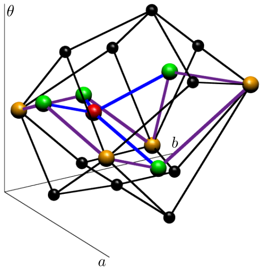
k-Neighbourhood and its construction

Definition

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

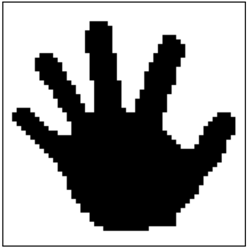
where $N^1(v) = N(v)$.



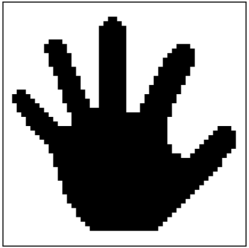


Complexity : $O(m^k N^2)$ where m is the neighbourhood size.

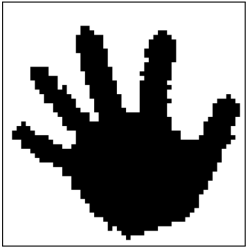
Experiment 1



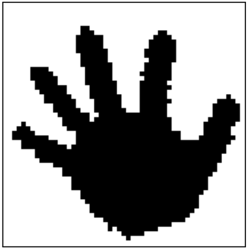
(a) Reference image



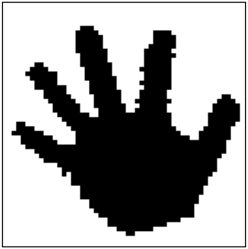
(b) Target image



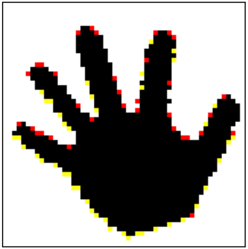
(c) Initial solution



(d) 1-neighbourhood

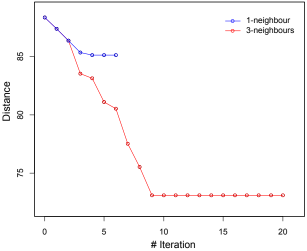


(e) 3-neighbourhood

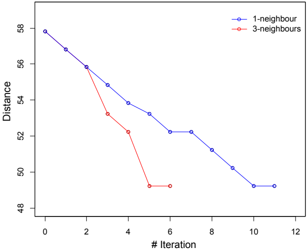


(f) (d) \ (e)

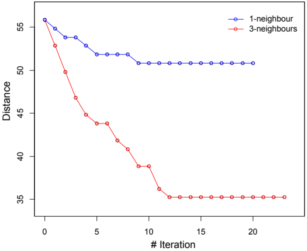
Experiment 1 (cont.)



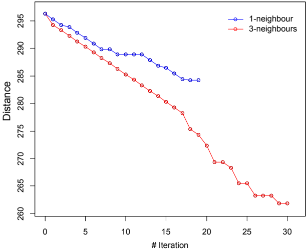
(a) Seed $(0, 0, 0.1)$



(b) Seed $(0.49, 0.35, 0.15)$



(c) Seed $(0.12, 0.05, 0.1314)$

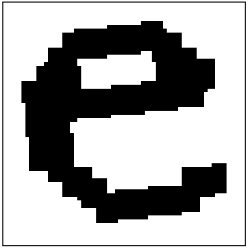


(d) Seed $(0.52, 0.79, 0.3107)$

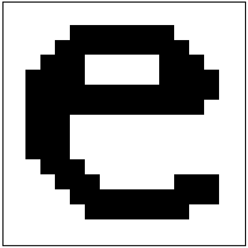
Experiment 1 (cont.)

Transformed image sequence by using 3-neighbours.

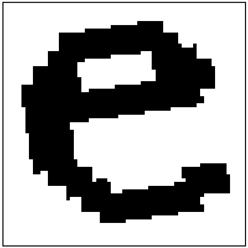
Experiment 2



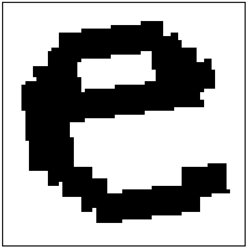
(a) Reference image



(b) Target image

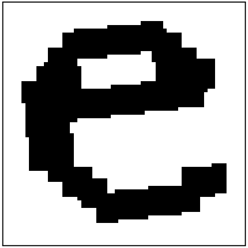


(c) 1-neighbourhood

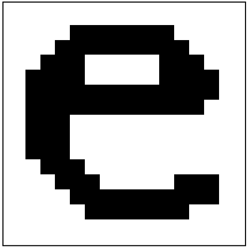


(d) 3-neighbourhood

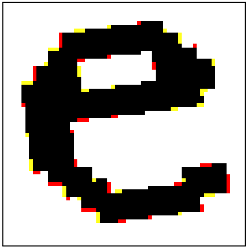
Experiment 2



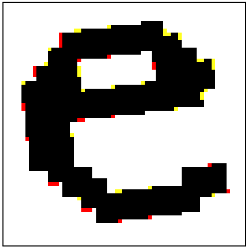
(a) Reference image



(b) Target image



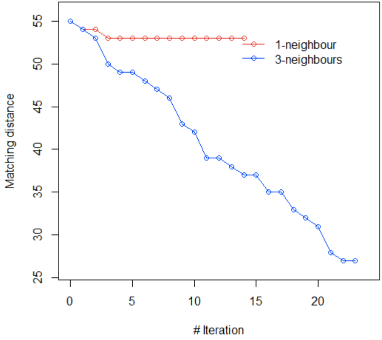
(c) 1-neighbourhood

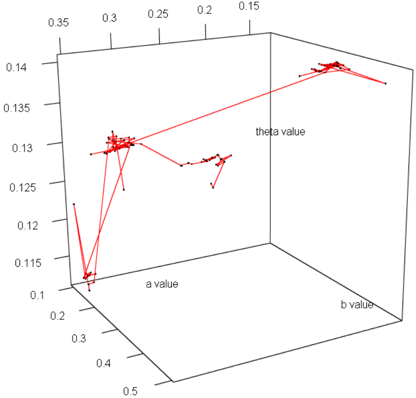


(d) 3-neighbourhood

Experiment 2 (cont.)

Sequences of distances and transformation parameters (only for 3-neighbofhoord) during iteration.





Conclusion and perspectives

Conclusion

- Purely discrete framework for 2D image registration under rigid transformation was proposed, based on DRT graph.

Conclusion and perspectives

Conclusion

- Purely discrete framework for 2D image registration under rigid transformation was proposed, based on DRT graph.
- A gradient descent procedure constructs the exact k-neighbourhood at each step with a $O(m^k N^2)$ time complexity.

Conclusion and perspectives

Conclusion

- Purely discrete framework for 2D image registration under rigid transformation was proposed, based on DRT graph.
- A gradient descent procedure constructs the exact k-neighbourhood at each step with a $O(m^k N^2)$ time complexity.
- In practice, m has a constant behavior with respect to $N \times N$.

Conclusion and perspectives

Conclusion

- Purely discrete framework for 2D image registration under rigid transformation was proposed, based on DRT graph.
- A gradient descent procedure constructs the exact k-neighbourhood at each step with a $O(m^k N^2)$ time complexity.
- In practice, m has a constant behavior with respect to $N \times N$.
- An efficient algorithm for computing the neighbours for the local search was proposed with almost linear time complexity.

Conclusion and perspectives

Conclusion

- Purely discrete framework for 2D image registration under rigid transformation was proposed, based on DRT graph.
- A gradient descent procedure constructs the exact k-neighbourhood at each step with a $O(m^k N^2)$ time complexity.
- In practice, m has a constant behavior with respect to $N \times N$.
- An efficient algorithm for computing the neighbours for the local search was proposed with almost linear time complexity.

Perspectives

- Improve the theoretical upper bound for m, which is so far $O(N^2)$.

Conclusion and perspectives

Conclusion

- Purely discrete framework for 2D image registration under rigid transformation was proposed, based on DRT graph.
- A gradient descent procedure constructs the exact k-neighbourhood at each step with a $O(m^k N^2)$ time complexity.
- In practice, m has a constant behavior with respect to $N \times N$.
- An efficient algorithm for computing the neighbours for the local search was proposed with almost linear time complexity.

Perspectives

- Improve the theoretical upper bound for m, which is so far $O(N^2)$.
- Combine our proposed method with other combinatorial approaches.

Conclusion and perspectives

Conclusion

- Purely discrete framework for 2D image registration under rigid transformation was proposed, based on DRT graph.
- A gradient descent procedure constructs the exact k-neighbourhood at each step with a $O(m^k N^2)$ time complexity.
- In practice, m has a constant behavior with respect to $N \times N$.
- An efficient algorithm for computing the neighbours for the local search was proposed with almost linear time complexity.

Perspectives

- Improve the theoretical upper bound for m, which is so far $O(N^2)$.
- Combine our proposed method with other combinatorial approaches.
- Extend to higher dimensions, and gray-level or labeled images.