Master 2 "SIS"

Digital Geometry

Topic 6:
Discrete geometric transformations

Yukiko Kenmochi

October 31, 2012

Geometric transformations of digital images

Given a **source image** A, we generate a **target image** B depending on the chosen transformation, for example:

Geometric transformations of digital images

Given a **source image** A, we generate a **target image** B depending on the chosen transformation, for example:

- translation,

Geometric transformations of digital images

Given a **source image** A, we generate a **target image** B depending on the chosen transformation, for example:

- translation,
- rotation,

Geometric transformations of digital images

Given a **source image** A, we generate a **target image** B depending on the chosen transformation, for example:

- translation,
- rotation,
- rigid transformation,

Geometric transformations of digital images

Given a **source image** A, we generate a **target image** B depending on the chosen transformation, for example:

- translation,
- rotation,
- rigid transformation,
- scaling,

Geometric transformations of digital images

Given a **source image** A, we generate a **target image** B depending on the chosen transformation, for example:

- translation,
- rotation,
- rigid transformation,
- scaling,
- affine transformation,
- ...

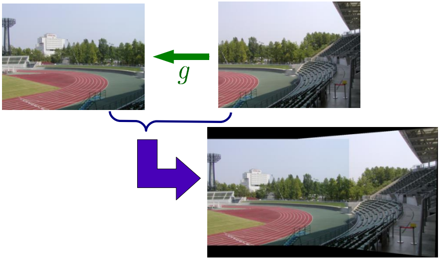
Geometric transformations of digital images

Given a **source image** A, we generate a **target image** B depending on the chosen transformation, for example:

- translation,
- rotation,
- rigid transformation,
- scaling,
- affine transformation,
- ...

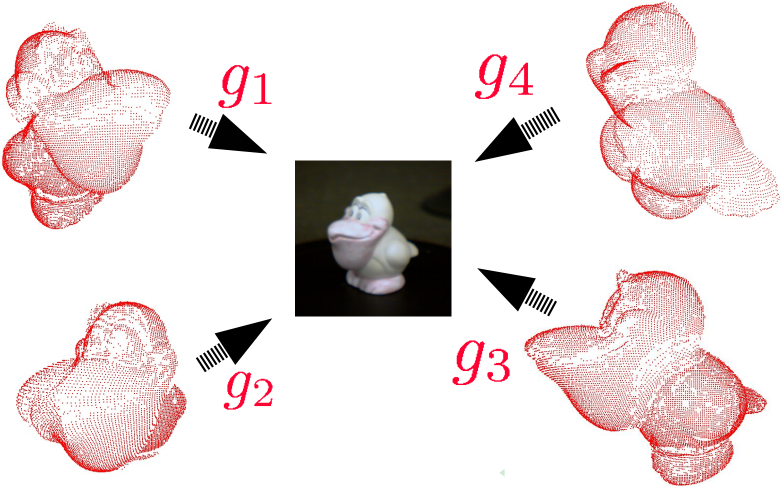
Application in 2D

Example: make a panoramic image.



Application in 3D

Example: reconstruct a 3D shape from a point cloud acquired by a laser rangefinder.



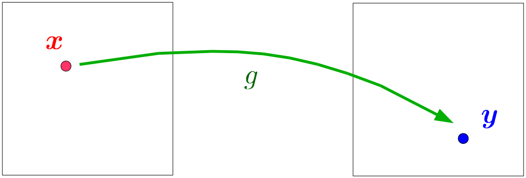
Geometric transformation

Definition

For a point $\mathbf{x} \in \mathbb{R}^d$, *we obtain the point* $\mathbf{y} \in \mathbb{R}^d$ *such that*

$$\mathbf{y} = g(\mathbf{x})$$

with a geometric transformation g.



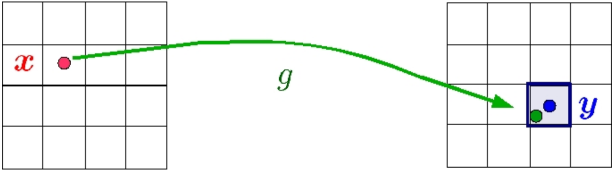
Discrete geometric transformation

Definition

For a point $\mathbf{x} \in \mathbb{Z}^d$, *we obtain the point* $\mathbf{y} \in \mathbb{Z}^d$ *such that*

$$g(\mathbf{x}) \in P(\mathbf{y})$$

where $P(\mathbf{y})$ *is the pixel whose center is* $\mathbf{y}$.



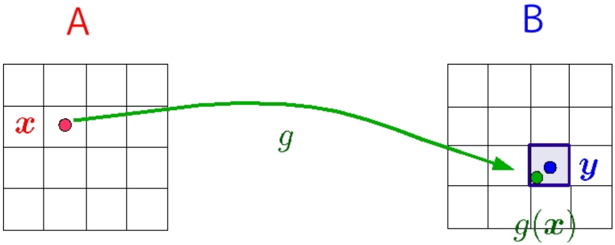
Remark: $\mathbf{y} \neq g(\mathbf{x})$ in general.

Lagrangian model of discrete transformations

Definition

For a discrete point $\mathbf{x}$ *of the source image* A*, we observe the pixel* $P(\mathbf{y})$ *of the target image* B *that includes the* **arrival point** $g(\mathbf{x})$*, i.e.,*

$$g(\mathbf{x}) \in P(\mathbf{y}).$$

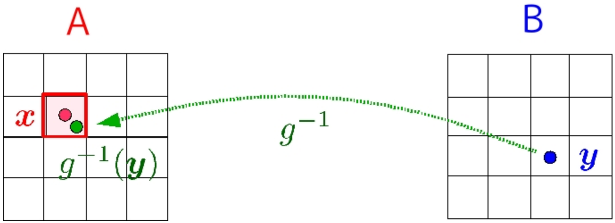


Eulerian model of discrete transformations

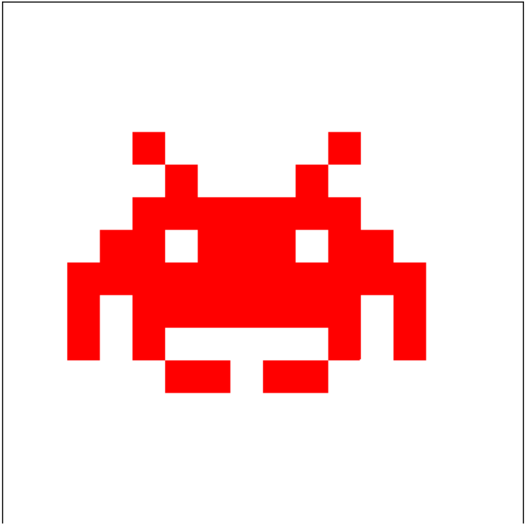
Definition

For a discrete point $\mathbf{y}$ *of the target image B, we observe the pixel* $P(\mathbf{x})$ *of the source image A that includes the starting point* $g^{-1}(\mathbf{y})$*, i.e.,*

$$g^{-1}(\mathbf{y}) \in P(\mathbf{x}).$$



Discrete rotation - Lagrangian model



Discrete rotation - Eulerian model

Criteria for discrete geometric transformation

Criteria expected to be preserved

Criteria for discrete geometric transformation

Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),

Criteria for discrete geometric transformation

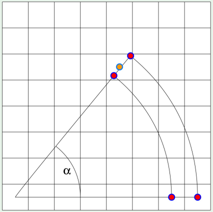
Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),

Criteria for discrete geometric transformation

Criteria expected to be preserved

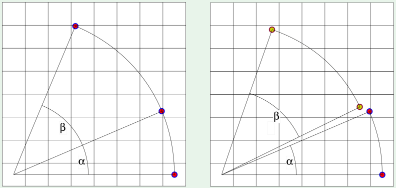
- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,



Criteria for discrete geometric transformation

Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,
- transitivity,



Criteria for discrete geometric transformation

Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,
- transitivity,
- incrementalism,

Criteria for discrete geometric transformation

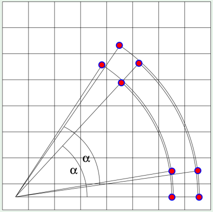
Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,
- transitivity,
- incrementalism,
- geometry,

Criteria for discrete geometric transformation

Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,
- transitivity,
- incrementalism,
- geometry,
- topology,



Criteria for discrete geometric transformation

Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,
- transitivity,
- incrementalism,
- geometry,
- topology,
- ...

Criteria for discrete geometric transformation

Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,
- transitivity,
- incrementalism,
- geometry,
- topology,
- ...

Criteria for discrete geometric transformation

Criteria expected to be preserved

- quality (the results equal to the discretized geometric transformation),
- exact computation (using only integers),
- bijection,
- transitivity,
- incrementalism,
- geometry,
- topology,
- ...

Discretized translation

Discretized translation

Definition (2D discretized translation)

A translation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \lfloor a + \frac{1}{2} \rfloor \\ \lfloor b + \frac{1}{2} \rfloor \end{pmatrix}$$

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

Discretized translation

Definition (2D discretized translation)

A translation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \lfloor a + \frac{1}{2} \rfloor \\ \lfloor b + \frac{1}{2} \rfloor \end{pmatrix}$$

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

It is

- bijective,

Discretized translation

Definition (2D discretized translation)

A translation taking a point $(x, y) \in \mathbb{Z}^2$ to a point $(x', y') \in \mathbb{Z}^2$ is defined by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \lfloor a + \frac{1}{2} \rfloor \\ \lfloor b + \frac{1}{2} \rfloor \end{pmatrix}$$

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

It is

- bijective,
- transitive (if $a, b \in \mathbb{Z}$),

Discretized translation

Definition (2D discretized translation)

A translation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \lfloor a + \frac{1}{2} \rfloor \\ \lfloor b + \frac{1}{2} \rfloor \end{pmatrix}$$

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

It is

- bijective,
- transitive (if $a, b \in \mathbb{Z}$),
- calculated exactly (if $a, b \in \mathbb{Z}$),

Discretized translation

Definition (2D discretized translation)

A translation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}$$

where $a, b \in \mathbb{Z}$ *.*

It is

- bijective,
- transitive (if $a, b \in \mathbb{Z}$),
- calculated exactly (if $a, b \in \mathbb{Z}$),
- geometry,

Discretized translation

Definition (2D discretized translation)

A translation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}$$

where $a, b \in \mathbb{Z}$.

It is

- bijective,
- transitive (if $a, b \in \mathbb{Z}$),
- calculated exactly (if $a, b \in \mathbb{Z}$),
- geometry,
- topology.

Discretized rotation

Definition (2D Euclidean rotation)

A rotation taking a point $(x, y) \in \mathbb{R}^2$ *to a point* $(x', y') \in \mathbb{R}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Discretized rotation

Definition (2D discretized rotation)

A rotation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor x\cos\theta - y\sin\theta + \frac{1}{2} \rfloor \\ \lfloor x\sin\theta + y\cos\theta + \frac{1}{2} \rfloor \end{pmatrix}.$$

Discretized rotation

Definition (2D discretized rotation)

A rotation taking a point $(x, y) \in \mathbb{Z}^2$ to a point $(x', y') \in \mathbb{Z}^2$ is defined by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor x\cos\theta - y\sin\theta + \frac{1}{2} \rfloor \\ \lfloor x\sin\theta + y\cos\theta + \frac{1}{2} \rfloor \end{pmatrix}.$$

It is **not**

- bijective,

Discretized rotation

Definition (2D discretized rotation)

A rotation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor x\cos\theta - y\sin\theta + \frac{1}{2} \rfloor \\ \lfloor x\sin\theta + y\cos\theta + \frac{1}{2} \rfloor \end{pmatrix}.$$

It is **not**

- bijective,
- transitive,

Discretized rotation

Definition (2D discretized rotation)

A rotation taking a point $(x, y) \in \mathbb{Z}^2$ to a point $(x', y') \in \mathbb{Z}^2$ is defined by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor x\cos\theta - y\sin\theta + \frac{1}{2} \rfloor \\ \lfloor x\sin\theta + y\cos\theta + \frac{1}{2} \rfloor \end{pmatrix}.$$

It is **not**

- bijective,
- transitive,
- calculated exactly,

Discretized rotation

Definition (2D discretized rotation)

A rotation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor x\cos\theta - y\sin\theta + \frac{1}{2} \rfloor \\ \lfloor x\sin\theta + y\cos\theta + \frac{1}{2} \rfloor \end{pmatrix}.$$

It is **not**

- bijective,
- transitive,
- calculated exactly,
- geometry (distances),

Discretized rotation

Definition (2D discretized rotation)

A rotation taking a point $(x, y) \in \mathbb{Z}^2$ to a point $(x', y') \in \mathbb{Z}^2$ is defined by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor x\cos\theta - y\sin\theta + \frac{1}{2} \rfloor \\ \lfloor x\sin\theta + y\cos\theta + \frac{1}{2} \rfloor \end{pmatrix}.$$

It is **not**

- bijective,
- transitive,
- calculated exactly,
- geometry (distances),
- topology.

Discrete rotations

1. *Quasi-shear rotation* is:

2. *Discrete rotation by hinge angles* is:

Discrete rotations

1. *Quasi-shear rotation* is:
   - calculated exactly,

2. *Discrete rotation by hinge angles* is:

Discrete rotations

1. *Quasi-shear rotation* is:
   - calculated exactly,
   - bijective,

2. *Discrete rotation by hinge angles* is:

Discrete rotations

1. *Quasi-shear rotation* is:
   - calculated exactly,
   - bijective,
   - an approximation to the discretized rotation.

2. *Discrete rotation by hinge angles* is:

Discrete rotations

1. *Quasi-shear rotation* is:
   - calculated exactly,
   - bijective,
   - an approximation to the discretized rotation.

2. *Discrete rotation by hinge angles* is:

Discrete rotations

1. *Quasi-shear rotation* is:
   - calculated exactly,
   - bijective,
   - an approximation to the discretized rotation.

2. *Discrete rotation by hinge angles* is:
   - calculated exactly,

Discrete rotations

1. *Quasi-shear rotation* is:
   - calculated exactly,
   - bijective,
   - an approximation to the discretized rotation.

2. *Discrete rotation by hinge angles* is:
   - calculated exactly,
   - equal to the discretized rotation,

Discrete rotations

1. *Quasi-shear rotation* is:
   - calculated exactly,
   - bijective,
   - an approximation to the discretized rotation.

2. *Discrete rotation by hinge angles* is:
   - calculated exactly,
   - equal to the discretized rotation,
   - incremental.

Shear rotation

Decomposition of a rotation into three shears

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin\theta & 1 \end{pmatrix} \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & -\frac{\alpha'}{\beta'} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\alpha}{\omega} & 1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{\alpha'}{\beta'} \\ 0 & 1 \end{pmatrix}$$

where $\omega > 0$ is a real value, $\alpha = \omega \sin\theta$, $\alpha' = \omega \sin\frac{\theta}{2}$ and $\beta' = \omega \cos\frac{\theta}{2}$.

Shear rotation

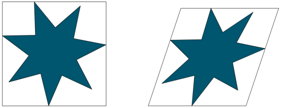
Decomposition of a rotation into three shears

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin\theta & 1 \end{pmatrix} \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & -\frac{\alpha'}{\beta'} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\alpha}{\omega} & 1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{\alpha'}{\beta'} \\ 0 & 1 \end{pmatrix}$$

where $\omega > 0$ is a real value, $\alpha = \omega \sin\theta$, $\alpha' = \omega \sin\frac{\theta}{2}$ and $\beta' = \omega \cos\frac{\theta}{2}$.

- Horizontal shear:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Shear rotation

Decomposition of a rotation into three shears

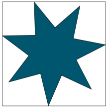
$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin\theta & 1 \end{pmatrix} \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}$$

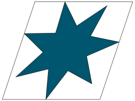
$$= \begin{pmatrix} 1 & -\frac{\alpha'}{\beta'} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\alpha}{\omega} & 1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{\alpha'}{\beta'} \\ 0 & 1 \end{pmatrix}$$

where $\omega > 0$ is a real value, $\alpha = \omega \sin\theta$, $\alpha' = \omega \sin\frac{\theta}{2}$ and $\beta' = \omega \cos\frac{\theta}{2}$.

- Horizontal shear:

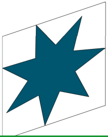
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$





- Vertical shear:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ m & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Quasi-shear

Definition (Andres, 1996)

For $(x, y), (x', y') \in \mathbb{Z}^2$, the horizontal quasi-shear $HQS(a, b)$ is defined by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + \lfloor \frac{a}{b} y + \frac{1}{2} \rfloor \\ y \end{pmatrix}$$

and the vertical quasi-shear $VQS(a, b)$ is defined by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y + \lfloor \frac{a}{b} x + \frac{1}{2} \rfloor \end{pmatrix}$$

where $a, b \in \mathbb{Z},\ b > 0$.

Quasi-shear rotation

Definition (Andres, 1996)

The quasi-shear rotation of angle θ is defined by

$$HQS(-a', b') \circ VQS(a, w) \circ HQS(-a', b')$$

where w is a chosen integer value and

$$\begin{aligned} a &= \lfloor w \sin \theta \rfloor, \\ a' &= \lfloor w \sin \frac{\theta}{2} \rfloor, \\ b' &= \lfloor w \cos \frac{\theta}{2} \rfloor. \end{aligned}$$

Remark: for example $w = 2^{15}$ is used for an image of size 2048×2048.

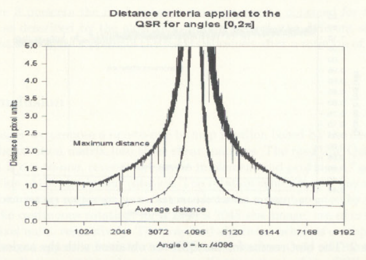
Advantages and disadvantages of quasi-shear rotation

Advantages

- memory saving (not necessary to store the original image),
- parallel computing (only shift).

Disadvantages

- large approximation errors around $\theta = \pi$.

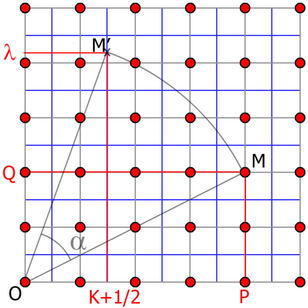


(Andres, 1996)

Hinge angles

Definition (Nouvel, 2006)

An angle α is a hinge angle for a discrete point $(p, q) \in \mathbb{Z}^2$ if the result of its rotation by α is a point on the half-grid.



Properties of hinge angels

Property (Nouvel, 2006; Thibault, 2009)

Properties of hinge angels

Property (Nouvel, 2006; Thibault, 2009)

- *The hinge angles are* dense *in* $\mathbb{R}$.

Properties of hinge angels

Property (Nouvel, 2006; Thibault, 2009)

- *The hinge angles are dense in $\mathbb{R}$.*
- *Each hinge angle α is represented by a triplet of integer numbers (p, q, k) with the uniqueness such that*

$$\cos\alpha = \frac{p\lambda + q(k+\frac{1}{2})}{p^2+q^2},$$
$$\sin\alpha = \frac{p(k+\frac{1}{2}) - q\lambda}{p^2+q^2},$$

where $\lambda = \sqrt{p^2+q^2-(k+\frac{1}{2})^2}$ and $k < \sqrt{p^2+q^2}$.

Properties of hinge angels

Property (Nouvel, 2006; Thibault, 2009)

- *The hinge angles are dense in $\mathbb{R}$.*
- *Each hinge angle α is represented by a triplet of integer numbers (p, q, k) with the uniqueness such that*

$$\cos\alpha = \frac{p\lambda + q(k+\frac{1}{2})}{p^2+q^2},$$
$$\sin\alpha = \frac{p(k+\frac{1}{2}) - q\lambda}{p^2+q^2},$$

where $\lambda = \sqrt{p^2+q^2-(k+\frac{1}{2})^2}$ and $k < \sqrt{p^2+q^2}$.

- *The comparison between two hinge angles can be made in constant time by using only integers.*

Properties of hinge angels

Property (Nouvel, 2006; Thibault, 2009)

- *The hinge angles are dense in $\mathbb{R}$.*
- *Each hinge angle α is represented by a triplet of integer numbers (p, q, k) with the uniqueness such that*

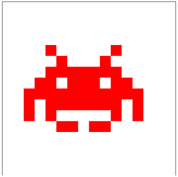
$$\begin{aligned} \cos\alpha &= \frac{p\lambda + q(k+\frac{1}{2})}{p^2+q^2}, \\ \sin\alpha &= \frac{p(k+\frac{1}{2}) - q\lambda}{p^2+q^2}, \end{aligned}$$

  where $\lambda = \sqrt{p^2+q^2-(k+\frac{1}{2})^2}$ and $k < \sqrt{p^2+q^2}$.
- *The comparison between two hinge angles can be made in constant time by using only integers.*
- *For an image of size $n \times n$, we have $8n^3$ hinge angles.*

Incremental discrete rotation

Algorithm (Thibault, 2009)

Input: an image A
Output: all the possible rotations of A

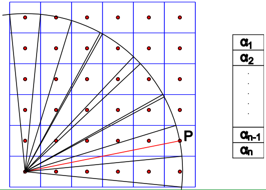


Incremental discrete rotation

Algorithm (Thibault, 2009)

Input: an image A
Output: all the possible rotations of A

- **for** each point (p, q) of A, calculate its hinge angles $\alpha(p, q, k)$ for all k, and store them in a sorted list $T_{(p,q)}$;

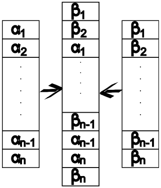


Incremental discrete rotation

Algorithm (Thibault, 2009)

Input: an image A
Output: all the possible rotations of A

- **for** each point (p, q) of A, calculate its hinge angles $\alpha(p, q, k)$ for all k, and store them in a sorted list $T_{(p,q)}$;
- fusion all the lists $T_{(p,q)}$ into a sorted list T;

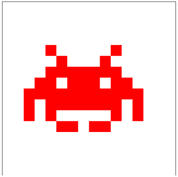


Incremental discrete rotation

Algorithm (Thibault, 2009)

Input: an image A
Output: all the possible rotations of A

- **for** each point (p, q) of A, calculate its hinge angles $\alpha(p, q, k)$ for all k, and store them in a sorted list $T_{(p,q)}$;
- fusion all the lists $T_{(p,q)}$ into a sorted list T;
- **for** each angle $\alpha(p, q, k)$ in T, move the point whose original coordinate is (p, q) from the current pixel $(k, \lfloor \lambda + \frac{1}{2} \rfloor)$ to the adjacent pixel $(k + 1, \lfloor \lambda + \frac{1}{2} \rfloor)$.



Incremental discrete rotation

Algorithm (Thibault, 2009)

Input: an image A
Output: all the possible rotations of A

- **for** each point (p, q) of A, calculate its hinge angles $\alpha(p, q, k)$ for all k, and store them in a sorted list $T_{(p,q)}$;
- fusion all the lists $T_{(p,q)}$ into a sorted list T;
- **for** each angle $\alpha(p, q, k)$ in T, move the point whose original coordinate is (p, q) from the current pixel $(k, \lfloor \lambda + \frac{1}{2} \rfloor)$ to the adjacent pixel $(k + 1, \lfloor \lambda + \frac{1}{2} \rfloor)$.

The complexity is $O(n^3)$ for an image of size $n \times n$.

Incremental discrete rotation

Algorithm (Thibault, 2009)

Input: an image A
Output: all the possible rotations of A

- **for** each point (p, q) of A, calculate its hinge angles $\alpha(p, q, k)$ for all k, and store them in a sorted list $T_{(p,q)}$;
- fusion all the lists $T_{(p,q)}$ into a sorted list T;
- **for** each angle $\alpha(p, q, k)$ in T, move the point whose original coordinate is (p, q) from the current pixel $(k, \lfloor \lambda + \frac{1}{2} \rfloor)$ to the adjacent pixel $(k + 1, \lfloor \lambda + \frac{1}{2} \rfloor)$.

The complexity is $O(n^3)$ for an image of size $n \times n$.
Practically, this complexity is not too large to generate all the rotated images.

Discretized affine transformation

Definition (2D Euclidean affine transformation)

An affine transformation taking a point $(x, y) \in \mathbb{R}^2$ *to a point* $(x', y') \in \mathbb{R}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{R}$.

Discretized affine transformation

Definition (2D discretized affine transformation)

An affine transformation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor ax + by + e \rfloor \\ \lfloor cx + dy + f \rfloor \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{R}$.

Discretized affine transformation

Definition (2D discretized affine transformation)

An affine transformation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor ax + by + e \rfloor \\ \lfloor cx + dy + f \rfloor \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{R}$*.*

In general, it is **not**

- bijective,

Discretized affine transformation

Definition (2D discretized affine transformation)

An affine transformation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor ax + by + e \rfloor \\ \lfloor cx + dy + f \rfloor \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{R}$.

In general, it is **not**

- bijective,
- transitive,

Discretized affine transformation

Definition (2D discretized affine transformation)

An affine transformation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor ax + by + e \rfloor \\ \lfloor cx + dy + f \rfloor \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{R}$.

In general, it is **not**

- bijective,
- transitive,
- calculated exactly,

Discretized affine transformation

Definition (2D discretized affine transformation)

An affine transformation taking a point $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor ax + by + e \rfloor \\ \lfloor cx + dy + f \rfloor \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{R}$*.*

In general, it is **not**

- bijective,
- transitive,
- calculated exactly,
- geometry,

Discretized affine transformation

Definition (2D discretized affine transformation)

An affine transformation taking a point $(x, y) \in \mathbb{Z}^2$ to a point $(x', y') \in \mathbb{Z}^2$ is defined by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor ax + by + e \rfloor \\ \lfloor cx + dy + f \rfloor \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{R}$.

In general, it is **not**

- bijective,
- transitive,
- calculated exactly,
- geometry,
- topology.

Discrete affine transformations

1. *Quasi-affine transformation* is:

2. *Combinatorial affine transformation* is:

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,

2. *Combinatorial affine transformation* is:

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,
   - calculated exactly,

2. *Combinatorial affine transformation* is:

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,
   - calculated exactly,
   - equal to the discretized affine transformation,

2. *Combinatorial affine transformation* is:

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,
   - calculated exactly,
   - equal to the discretized affine transformation,
   - efficiently computed.

2. *Combinatorial affine transformation* is:

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,
   - calculated exactly,
   - equal to the discretized affine transformation,
   - efficiently computed.

2. *Combinatorial affine transformation* is:
   - a generalization of the discrete rotation by hinge angles,

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,
   - calculated exactly,
   - equal to the discretized affine transformation,
   - efficiently computed.

2. *Combinatorial affine transformation* is:
   - a generalization of the discrete rotation by hinge angles,
   - calculated exactly,

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,
   - calculated exactly,
   - equal to the discretized affine transformation,
   - efficiently computed.

2. *Combinatorial affine transformation* is:
   - a generalization of the discrete rotation by hinge angles,
   - calculated exactly,
   - equal to the discretized affine transformation,

Discrete affine transformations

1. *Quasi-affine transformation* is:
   - a generalization of the quasi-shear rotation,
   - calculated exactly,
   - equal to the discretized affine transformation,
   - efficiently computed.

2. *Combinatorial affine transformation* is:
   - a generalization of the discrete rotation by hinge angles,
   - calculated exactly,
   - equal to the discretized affine transformation,
   - incremental.

Quasi-affine transformation

Definition (Jacob, 1993)

A quasi-affine transformation *taking a point* $(x, y) \in \mathbb{Z}^2$ *to a point* $(x', y') \in \mathbb{Z}^2$ *is defined by*

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lfloor \frac{ax+by+e}{\omega} \rfloor \\ \lfloor \frac{cx+dy+f}{\omega} \rfloor \end{pmatrix}$$

where $a, b, c, d, e, f \in \mathbb{Z}$, $\omega \in \mathbb{N}^+$.

This is equivalent to the intersection of two arithmetic lines for a given point $(x', y') \in \mathbb{Z}^2$:

$$\omega x' \leq ax + by + e < \omega(x' + 1),$$
$$\omega y' \leq cx + dy + y < \omega(y' + 1).$$

Tiles and their periodicity

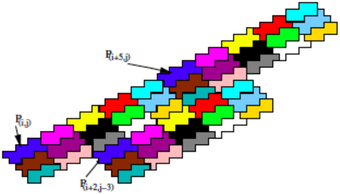
Definition (Tile)

Let f be a quasi-affine transformation. For a point $\mathbf{y} \in \mathbb{Z}^2$, the **tile** *of order 1 of $\mathbf{y}$ is defined by*

$$P_{\mathbf{y}} = \{\mathbf{x} \in \mathbb{Z}^2 : f(\mathbf{x}) = \mathbf{y}\}.$$

Theory

The set of quasi-affine transformation tiles is **periodic**.



(Coeurjolly et al., 2009)

Tile construction and quasi-affine transformation

By using Hermite normal form, efficient tile construction can be designed.

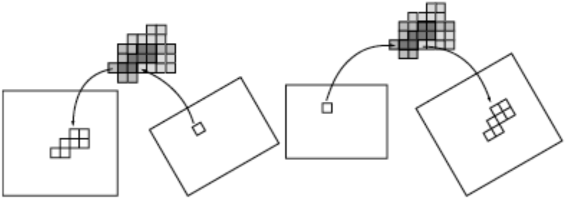


Fig. 3. Illustration in dimension 2 of the QAT algorithm when f is contracting (a) and dilating (b). In both cases, we use the canonical tiles contained in the super-tile to speed-up the transformation.

(Coeurjolly et al., 2009)

Quasi-affine transformation

- If f is contracting, we give to each pixel $\mathbf{y}$ of image B the average color of the tile $P_{\mathbf{y}}$ in image A.
- If f is dilating, we give the color of each pixel $\mathbf{y}$ of image A to each pixel of $P_{\mathbf{y}}$ in image B (replace f by f^{-1}).

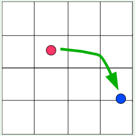
Combinatorial affine transformation

Given a discrete point $(x, y) \in \mathbb{Z}^2$, for an affine transformation:

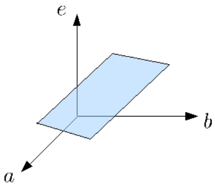
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

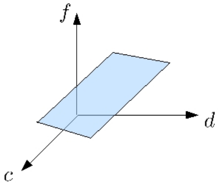
the critical cases are:

$$x' = k_x + \frac{1}{2} = ax + by + e \quad \text{where} \quad k_x \in \mathbb{Z},$$
$$y' = k_y + \frac{1}{2} = cx + dy + f \quad \text{where} \quad k_y \in \mathbb{Z}.$$



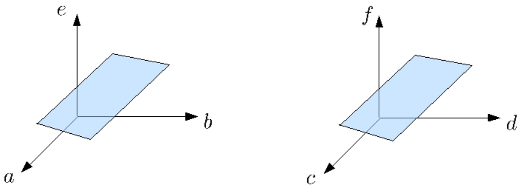
Dual space of affine transformation





$$k_x + \frac{1}{2} = ax + by + e \quad \text{for} \quad k_x, x, y \in \mathbb{Z},$$
$$k_y + \frac{1}{2} = cx + dy + f \quad \text{for} \quad k_y, x, y \in \mathbb{Z}.$$

Dual space of affine transformation



$$k_x + \frac{1}{2} = ax + by + e \quad \text{for} \quad k_x, x, y \in \mathbb{Z},$$
$$k_y + \frac{1}{2} = cx + dy + f \quad \text{for} \quad k_y, x, y \in \mathbb{Z}.$$

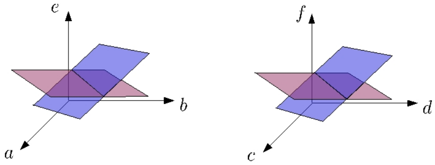
Remark

For an image of size $n \times n$, each dual space contains n^3 planes.

Combinatorial affine transformation

Each dual space is discretized by n^3 planes:

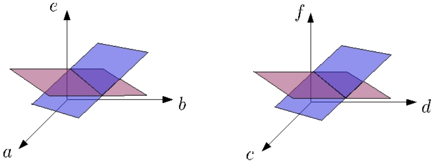
$$k_x + \frac{1}{2} = ax + by + e \quad \text{for} \quad k_x, x, y \in \mathbb{Z},$$
$$k_y + \frac{1}{2} = cx + dy + f \quad \text{for} \quad k_y, x, y \in \mathbb{Z}.$$



Combinatorial affine transformation

Each dual space is discretized by n^3 planes:

$$k_x + \frac{1}{2} = ax + by + e \quad \text{for} \quad k_x, x, y \in \mathbb{Z},$$
$$k_y + \frac{1}{2} = cx + dy + f \quad \text{for} \quad k_y, x, y \in \mathbb{Z}.$$



Property (Hundt et al., 2007)

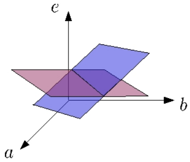
- *For an image of size $n \times n$, each dual space is divided in $O(n^9)$, i.e., the number of discrete transformations is $O(n^{18})$.*

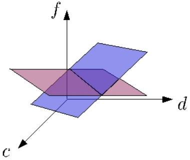
Combinatorial affine transformation

Each dual space is discretized by n^3 planes:

$$k_x + \frac{1}{2} = ax + by + e \quad \text{for} \quad k_x, x, y \in \mathbb{Z},$$

$$k_y + \frac{1}{2} = cx + dy + f \quad \text{for} \quad k_y, x, y \in \mathbb{Z}.$$





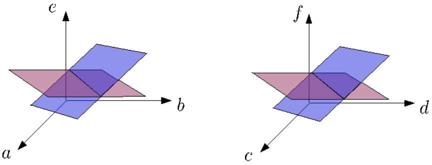
Property (Hundt et al., 2007)

- *For an image of size $n \times n$, each dual space is divided in $O(n^9)$, i.e., the number of discrete transformations is $O(n^{18})$.*
- *All the calculations are made by using integers.*

Combinatorial affine transformation

Each dual space is discretized by n^3 planes:

$$k_x + \frac{1}{2} = ax + by + e \quad \text{for} \quad k_x, x, y \in \mathbb{Z},$$
$$k_y + \frac{1}{2} = cx + dy + f \quad \text{for} \quad k_y, x, y \in \mathbb{Z}.$$



Property (Hundt et al., 2007)

- *For an image of size $n \times n$, each dual space is divided in $O(n^9)$, i.e., the number of discrete transformations is $O(n^{18})$.*
- *All the calculations are made by using integers.*
- *The discrete transformation corresponds to the discretized transformation.*

Combinatorial image matching

For a 2D digital image of size $n \times n$, the numbers of the generated images under different transformations are as follow.

Transformation	complexity
Rotation (Amir, et al., 2003)	$O(n^3)$
Scaling (Amir, et al., 2003)	$O(n^3)$
Rotation and scaling (Hundt, Liskiewicz, 2009)	$O(n^6)$
Rigid transformation (Ngo, et al., 2011)	$O(n^9)$
Linear transformation (Hundt, Liskiewicz, 2008)	$O(n^{12})$
Affine transformation (Hundt, 2007)	$O(n^{18})$
Projective transformation (Hundt, Liskiewicz, 2008)	$O(n^{24})$

References

- R. Klette and A. Rosenfeld.
  "Transformations," Chapter 14 in "Digital geometry: geometric methods for digital picture analysis," Morgan Kaufmann, 2004.
- E. Andres et M.-A. Jacob-Da Col.
  "Transformations affines discrètes," Chapitre 7 dans "Géométrie discrète et images numériques," Hermès, 2007.
- B. Nouvel.
  "Rotations discrètes et automates cellulaires," Thèse, ENS de Lyon, 2006.
- Y. Thibault.
  "Rotations in 2D and 3D discrete spaces," Thèse, Université Paris-Est, 2010.