



# Computing upper and lower bounds of rotation angles from digital images

Yohan Thibault<sup>a,b,\*</sup>, Yukiko Kenmochi<sup>b</sup>, Akihiro Sugimoto<sup>a,b</sup>

<sup>a</sup>National Institute of Informatics, Japan

<sup>b</sup>Université Paris-Est, Laboratoire d'Informatique de l'Institut Gapard-Monge, UMR CNRS 8049, ESIEE Paris, France

## ARTICLE INFO

### Article history:

Received 1 August 2008

Received in revised form 1 December 2008

Accepted 6 December 2008

### Keywords:

Discrete geometry

Rotation

Discrete rotation

Hinge angles

Pythagorean angles

Admissible rotation angles

## ABSTRACT

Rotations in the discrete plane are important for many applications such as image matching or construction of mosaic images. We suppose that a digital image  $A$  is transformed to another digital image  $B$  by a rotation. In the discrete plane, there are many angles giving the rotation from  $A$  to  $B$ , which we call admissible rotation angles from  $A$  to  $B$ . For such a set of admissible rotation angles, there exist two angles that achieve the lower and the upper bounds. To find those lower and upper bounds, we use hinge angles as used in Nouvel and Rémila [Incremental and transitive discrete rotations, in: R. Reulke, U. Eckardt, B. Flash, U. Knauer, K. Polthier (Eds.), *Combinatorial Image Analysis, Lecture Notes in Computer Science*, vol. 4040, Springer, Berlin, 2006, pp. 199–213]. A sequence of hinge angles is a set of particular angles determined by a digital image in the sense that any angle between two consecutive hinge angles gives the identical rotation of the digital image. We propose a method for obtaining the lower and the upper bounds of admissible rotation angles using hinge angles from a given Euclidean angle or from a pair of corresponding digital images.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

Rotations in the discrete plane are required in many applications for image computation such as image matching, construction of mosaic images [2]. For the moment, the most popular method to estimate the rotation angle is to approximate the rotation matrix by minimizing errors [2].

In the continuous plane, the Euclidean rotation is well defined and possesses the property of bijectivity. This implies that for two angles  $\gamma_1, \gamma_2$  and a set of points  $A$ , if the Euclidean rotation with angle  $\gamma_1$  applied to  $A$  gives the same result as the Euclidean rotation with angle  $\gamma_2$  applied to  $A$ , then we have  $\gamma_1 = \gamma_2$ .

In the discrete plane, however, two different points can arrive at the same grid point after a discretization of the Euclidean rotation (DER). Because of this reason, two different angles can give the same rotation result for a set  $A$  of grid points.<sup>1</sup> In other words, we can define a set of *admissible rotation angles* (ARA)  $S$  such that any angle in  $S$  gives the same rotation result for  $A$ . Note that  $S$  depends on  $A$ . The two most interesting angles in  $S$  are the lower and the

upper bounds because with only these two angles we can deduce the other angles in  $S$ . This paper aims to find these two angles from a given rotation angle or from two given corresponding sets of grid points. In order to identify the exact bounds, we should not involve any computation error. Thus, we work with discrete geometry tools which guarantee to avoid computation with real numbers. Moreover, because we assume that our data are discretized from continuous images of an object, we enforce the property that the discrete rotation<sup>2</sup> between two different sets of grid points gives the same result as DER.

Some work on discrete rotations already exists. The most widely used discrete rotation in the beginning is the CORDIC algorithm [3]. The CORDIC algorithm uses a sequence of fractions for an approximation of  $\pi$ . It multiplies/adds fractions in this sequence to approximate values of sine and cosine. Thus multiple approximations lead to little differences between results of DER and those of CORDIC. Andres [4,5] described some discrete rotations such as the rotation by discrete circles, the rotation by Pythagorean lines or the quasi-shear rotation. Computation executed during these rotations are exact. However, because they preserve the bijectivity, they cannot give the same results as DER.

On the other hand, Nouvel and Rémila [1] proposed another discrete rotation based on hinge angles, which gives the same results as

\* Corresponding author at: Université Paris-Est, Laboratoire d'Informatique de l'Institut Gapard-Monge, UMR CNRS 8049, ESIEE Paris, France.

E-mail addresses: [thibault@esiee.fr](mailto:thibault@esiee.fr) (Y. Thibault), [y.kenmochi@esiee.fr](mailto:y.kenmochi@esiee.fr) (Y. Kenmochi), [sugimoto@nii.ac.jp](mailto:sugimoto@nii.ac.jp) (A. Sugimoto).

<sup>1</sup>Accordingly, DER is not bijective.

<sup>2</sup>A discrete rotation is a rotation designed for the discrete space. It transforms a set of grid points into another set of grid points.

DER. It is known that a sequence of hinge angles is a set of particular angles determined by a digital image in the sense that any angle between two consecutive hinge angles gives the identical rotated digital image. This means that hinge angles correspond to the discontinuity of DER. Nouvel and Rémila showed that each hinge angle is represented by an integer triplet, so that any discrete rotation of a digital image is realized only with integer calculation. Because their algorithm gives the same rotation results as DER, we see that hinge angles represented by integer triplets give sufficient information for executing any digital image rotation.

In this paper, we propose a discrete method for finding the lower and the upper bounds of ARA. Our method uses hinge angles because we can obtain the same result as DER and because they allow exact computations. The input data of our method is two sets of grid points where point correspondences across the two sets are known. The output is two hinge angles that give the lower and the upper bounds of the ARA for the two sets. Note that a part of this work was presented in [6].

### 2. Discrete rotation

It can appear strange that the Euclidean rotation used for the common task in geometric computation is problematic for many applications. Data are usually represented in the computer by integers or rational numbers. But the Euclidean rotation which requires sine and cosine functions is designed for real numbers. Therefore, the computation results given by a Euclidean rotation are, in most cases, represented by floating numbers which are approximations of real numbers. When an algorithm uses the Euclidean rotation for integer or rational data and then converts the floating values obtained by the algorithm into integer or rational numbers, precisions of these results may decrease. Another problem also arises for rotations in discrete space. It is well known that the Euclidean rotation is bijective and transitive. But when we convert the result obtained by the Euclidean rotation into a discrete space, it is easy to see that these two properties are lost [7,8]. The loss of these two properties leads to research on the discrete rotation.

There are two ways to compute a discrete rotation: using floating numbers and using only integers. The first way, in most cases, is easiest and allows us to use floating computation followed by the rounding function to obtain the set of grid points as the output. The main problem with the rounding function is the approximation due to the loss of the value after the decimal point. This approximation leads to lack of precision during computation. The second way does not have this problem, but avoiding floating numbers implies that sine and cosine functions should not be used. Computing rotations without trigonometrical functions requires development of a new method, which is a tough problem.

### 3. Hinge angles

We consider grid points in  $\mathbb{Z}^2$  as the centers of pixels and rotate them in such a way that the rotation center has integer coordinates. Hinge angles are particular angles that make some points in  $\mathbb{Z}^2$  rotated to points on the frontier between adjacent pixels. In this section, we give the definition of hinge angles and their properties related to Pythagorean angles.

#### 3.1. Definition of hinge angles

Let  $\mathbf{x} = (x, y)$  be a point in  $\mathbb{R}^2$ . We say that  $\mathbf{x}$  has a semi-integer coordinate if  $x + \frac{1}{2} \in \mathbb{Z}$  or  $y + \frac{1}{2} \in \mathbb{Z}$ . The set of points each of which has a semi-integer coordinate is called the half-grid and is denoted

by  $\mathcal{H}$ . Thus,  $\mathcal{H}$  represents the set of points on the frontiers of all pixels whose centroids are points in  $\mathbb{Z}^2$ .

**Definition 1.** An angle  $\alpha$  is called a hinge angle if at least one point in  $\mathbb{Z}^2$  exists such that its image by the Euclidean rotation with  $\alpha$  belongs to  $\mathcal{H}$ .

Because  $\mathcal{H}$  can be seen as the discontinuity of the rounding function, hinge angles can be regarded as the discontinuity of the DER. More simply, hinge angles determine a transit of a grid point from a pixel to its adjacent pixel during the rotation.

The following theorem is important because it shows that we can represent every hinge angle with three integers.

**Theorem 2 (Nouvel and Rémila [1]).** An angle  $\alpha$  is a hinge angle for a grid point  $(P, Q) \in \mathbb{Z}^2$  if and only if there exists  $K \in \mathbb{Z}$  such that

$$2Q \cos \alpha + 2P \sin \alpha = 2K + 1. \tag{1}$$

Geometrically, a hinge angle  $\alpha$  is formed by two rays going through  $(P, Q)$  and a half-grid point  $(K + \frac{1}{2}, \lambda)$  where the two rays share the origin as their endpoints as shown in Fig. 1 (left). This theorem indicates that all calculations related to hinge angles can be done only with integers. Hereafter,  $\alpha$  indicates a hinge angle.

We denote by  $\alpha(P, Q, K)$  the hinge angle generated by an integer triplet  $(P, Q, K)$ . Setting  $\lambda = \sqrt{P^2 + Q^2 - (K + \frac{1}{2})^2}$ , we easily derive the following equations from (1) and Fig. 1 (left),

$$\cos \alpha = \frac{P\lambda + Q(K + \frac{1}{2})}{P^2 + Q^2}, \quad \sin \alpha = \frac{P(K + \frac{1}{2}) - Q\lambda}{P^2 + Q^2}. \tag{2}$$

Note that we have a case where a half-grid point is  $(\lambda, K + \frac{1}{2})$  instead of  $(K + \frac{1}{2}, \lambda)$ . In such a case, the above equations become

$$\cos \alpha = \frac{Q\lambda + P(K + \frac{1}{2})}{P^2 + Q^2}, \quad \sin \alpha = \frac{P\lambda - Q(K + \frac{1}{2})}{P^2 + Q^2}. \tag{3}$$

The symmetries on hinge angles are important because it allows us to restrict rotations in the first quadrant of the circle such that  $\alpha \in [0, \pi/2]$ .

**Corollary 3.** Each triplet  $(P, Q, K)$  corresponds to four symmetrical hinge angles  $\alpha + \pi k/2$  where  $k = 0, 1, 2, 3$ .

Fig. 1 (right) gives an example of Corollary 3. In order to distinguish the case of  $(K + \frac{1}{2}, \lambda)$  from that of  $(\lambda, K + \frac{1}{2})$ , we change the sign of  $K$ ; we use  $\alpha(P, Q, K)$  for the case of  $(K + \frac{1}{2}, \lambda)$ , and  $\alpha(P, Q, -K)$  for the case of  $(\lambda, K + \frac{1}{2})$ . Because the symmetries allow us to restrict  $\alpha$  to the range  $[0, \pi/2]$ , as mentioned above, we may assume that  $K$  is positive.

#### 3.2. Properties related to Pythagorean angle

Because hinge angles are strongly related to Pythagorean angles, certain properties of Pythagorean angles are required to prove some properties of hinge angles. Thus, we first give the definition of Pythagorean angles and their properties.

**Definition 4.** An angle  $\theta$  is called Pythagorean if and only if both its cosine and sine belong to the set of rational numbers  $\mathbb{Q}$ .

We can deduce from Definition 4 that a Pythagorean angle  $\theta$  is represented by an integer triplet  $(a, b, c)$  such that

$$\cos \theta = \frac{a}{c}, \quad \sin \theta = \frac{b}{c}. \tag{4}$$

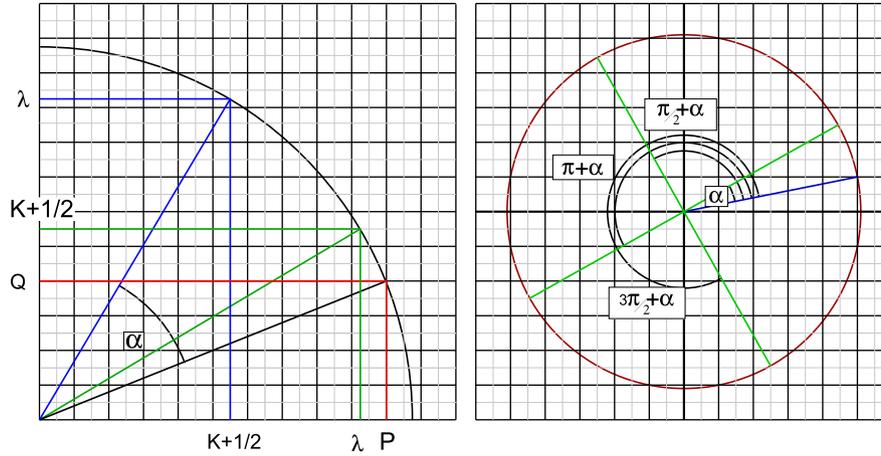


Fig. 1. A hinge angle  $\alpha(P, Q, K)$  (left) and four symmetrical hinge angles (right).

In the following,  $\theta$  indicates a Pythagorean angle. The lemma below for Pythagorean angles is well known.

**Lemma 5.** Let  $(a, b, c)$  be an integer triplet generating a Pythagorean angle where  $|c| = \max\{|a|, |b|, |c|\}$ . If  $\gcd(a, b, c) = 1$ , then  $c$  is odd.

If  $\gcd(a, b, c) = i$ , then  $\gcd(a/i, b/i, c/i) = 1$  and the triplet of integers  $(a/i, b/i, c/i)$  generates the same Pythagorean angle as  $(a, b, c)$ .

**Theorem 6.** Let  $E_h$  be the set of hinge angles and  $E_p$  be the set of Pythagorean angles. Then we have  $E_h \cap E_p = \emptyset$ .

**Proof.** Assume that there exists an angle  $\alpha$  such that  $\alpha \in E_h$  and  $\alpha \in E_p$ . Since  $\alpha \in E_p$ , we can find an integer triplet  $(a, b, c)$  generating  $\alpha$  where  $\gcd(a, b, c) = 1$ . By substitution of (4) in (1), we obtain

$$2 \frac{Qa + Pb}{c} = 2K + 1, \tag{5}$$

from which we derive  $2(Qa + Pb)/c \in \mathbb{Z}$ . Because we know that  $c$  is odd according to Lemma 5, we obtain  $(Qa + Pb)/c \in \mathbb{Z}$ . This means that  $2(Qa + Pb)/c$  is even, while  $2K + 1$  is always odd, which leads to a contradiction.  $\square$

This theorem shows that it is not possible to rotate a point  $(i, j) \in \mathbb{Z}^2$  to a point  $(x, y)$  such as  $x = i + \frac{1}{2}$ ,  $y = j + \frac{1}{2}$ , if the angle of the rotation is a hinge angle.

The next theorem shows an interesting relation between hinge angles and Pythagorean angles.

**Theorem 7 (Nouvel [9]).** Let  $\theta$  be a Pythagorean angle and  $\alpha$  be a hinge angle. The angle  $\alpha' = \alpha + \theta$  is a hinge angle.

#### 4. Computing the lower bound rotation angle from a Pythagorean angle

Because we work here in the discrete space, the rotation of a grid point by two different angles can give the same result. Namely, two different angles give the same result after the rotation of a grid point followed by discretization. Generally there exists a range of angles in which the same result is obtained. We thus define ARA, that is the abbreviation for admissible rotation angles, to represent this range of angles.

In this section, we propose a method for computing the lower bound  $\alpha_{\text{inf}}$  of ARA for a given digital image from a given angle. Note that with minor modifications, this method can also find the upper bound  $\alpha_{\text{sup}}$  of ARA. Thus applying any rotation to the given digital

image with an angle between  $\alpha_{\text{inf}}$  and  $\alpha_{\text{sup}}$  gives the same result. We note that both  $\alpha_{\text{inf}}$  and  $\alpha_{\text{sup}}$  are hinge angles.

Our input is a Euclidean angle. However, we can replace it by a Pythagorean angle because there exists a method with linear time complexity  $O(m)$  to approximate a given Euclidean angle with a Pythagorean angle with a precision of  $1/10^m$  [10], where  $m$  is a fixed integer that represents the quality of approximation. Below, we assume that a Pythagorean angle is given as in [1].

Nouvel and Rémila presented a method for computing all possible hinge angles for a grid point or a pixel in a digital image [1]. Their method can be used for finding the hinge angle that is the lower bound of the ARA. Its time complexity is  $O(n \log(n))$  where  $n$  is the number of all hinge angles for a given grid point. Note that  $n$  depends on the coordinates of the grid point.

##### 4.1. Computing the lower bound rotation angle for a grid point

For each grid point  $\mathbf{p} = (P, Q) \in \mathbb{Z}^2$ , there are less than  $n = \lfloor \sqrt{P^2 + Q^2} + \frac{1}{2} \rfloor$  different hinge angles in each quadrant [1]. We can compare in magnitude any pair of hinge angles. This means that we have a totally ordered set  $\{\alpha(P, Q, K_1), \alpha(P, Q, K_2), \dots, \alpha(P, Q, K_n)\}$  of hinge angles in the ascending order where  $K_i \in \mathbb{Z}$ . Given a Pythagorean angle  $\theta$ , in order to find the lower bound rotation angle  $\alpha(P, Q, K_i)$  such that  $\alpha(P, Q, K_i) < \theta < \alpha(P, Q, K_{i+1})$ , we use a binary search. The binary search allows us to find  $\alpha(P, Q, K_i)$  in  $O(\log(n))$ , providing that we can compare a hinge angle with a Pythagorean angle in a constant time. The algorithm is described in Function 1. Note that, thanks to Theorem 9, a Pythagorean angle for the input can be replaced by a hinge angle.

**Function 1.** Function for the lower bound rotation angle for a point.

**INPUT:** A point  $\mathbf{p}(P, Q)$ , a Pythagorean angle  $\theta$

**OUTPUT:** A hinge angle  $\alpha(P, Q, K)$

$$K_{\max} \leftarrow \lfloor \sqrt{P^2 + Q^2} - 1 \rfloor;$$

$$K_{\min} \leftarrow 0;$$

$$K \leftarrow \lfloor \frac{K_{\max} + K_{\min}}{2} \rfloor;$$

**While**  $K_{\max} - K_{\min} \neq 1$  **do**

**if**  $\alpha(P, Q, K) > \theta$  **then**

$$K_{\max} = K;$$

**else**

$$K_{\min} = K;$$

**end if**

$$K = \lfloor \frac{K_{\max} + K_{\min}}{2} \rfloor;$$

**end while**

**return**  $\alpha(P, Q, K)$ ;

The following theorem shows that the comparison between a hinge angle and a Pythagorean angle is executed in a constant time.

**Theorem 8.** Let  $\alpha$  be a hinge angle and  $\theta$  be a Pythagorean angle. We can check whether  $\alpha > \theta$  in a constant time with integer calculation.

**Proof.** Let  $\alpha(P, Q, K)$  be a hinge angle in  $[0, \pi/2]$  and  $\theta(a, b, c)$  be a Pythagorean angle in  $[0, \pi/2]$ . From Eqs. (2) and (4), we obtain

$$\cos \alpha - \cos \theta = \frac{Q(K + \frac{1}{2}) + P\lambda}{p^2 + Q^2} - \frac{a}{c}.$$

If  $\theta$  is greater than  $\alpha$ ,  $\cos \alpha - \cos \theta > 0$ . Thus

$$cQ(2K + 1) - 2a(p^2 + Q^2) > -2cP\lambda. \quad (6)$$

Since we know that  $c, P, \lambda$  are positive, the right-hand side of (6) is always negative. Thus, if the left-hand side of (6) is not negative, then  $\theta > \alpha$ . Otherwise, we take the square of (6), so that we only have to check whether the following inequality holds:

$$[cQ(2K + 1) - 2a(p^2 + Q^2)]^2 < 4c^2P^2\lambda^2. \quad (7)$$

Note that because  $\lambda = \sqrt{p^2 + Q^2 - (K + \frac{1}{2})^2}$ , we see that  $4\lambda^2$  in the right-hand side of (7) contains only integer values. Therefore, we can verify (7) with integer calculation. If it is true,  $\theta > \alpha$ ; otherwise,  $\alpha > \theta$ . Note that because of Theorem 6, it is impossible to obtain  $\theta = \alpha$ .

We claim that this comparison of a hinge angle and a Pythagorean angle is executed in a constant time because even in the worst case, we only have to check Eqs. (6) and (7).  $\square$

We mention the importance of the rotation with angle  $\pi/2$  and its multiplications. In fact, if the angle of a rotation is equal to  $\pi/2, \pi, 3\pi/2$ , we just have to flip  $x$  and/or  $y$ -coordinates by changing their signs. It gives the justification that we can restrict the input angle  $\theta$  to  $0 < \theta < \pi/2$ .

#### 4.2. Computing the lower bound rotation angle for a set of grid points

In this subsection, we present an algorithm for computing the lower bound rotation angle from a given Pythagorean angle  $\theta$  for a digital image consisting of  $m$  grid points  $A$ . The output is a triplet of integers that represents the lower bound rotation angle for  $A$ . We note that the lower bound rotation is a hinge angle.

The algorithm computes all hinge angles for all points in  $A$ , and sorts them to keep the largest one. More precisely, we first compute the lower bound rotation angle for the first point of  $A$ , and store it as a reference. Then, we compute the lower bound rotation angle for the second point in  $A$  and compare it with the reference to keep the larger one. After repeating this procedure for all points in  $A$ , our algorithm returns the lower bound rotation angle  $\alpha$  such that  $\alpha < \theta$ . The time complexity of this algorithm is  $O(m \log(n))$  because we call  $m$  times the binary search (Function 1) whose time complexity is  $O(\log(n))$ . Function 2 illustrates our algorithm. As shown in Theorem 9, the comparison between two hinge angles is realized in a constant time, so that our algorithm does not change the global complexity.

**Theorem 9.** Let  $\alpha_1, \alpha_2$  be two hinge angles. We can check whether  $\alpha_1 > \alpha_2$  in a constant time and with integer calculation.

**Proof.** Let  $\alpha_1(p, q, k)$  and  $\alpha_2(r, s, l)$  be two hinge angles in  $[0, \pi/2]$ . From (3) we obtain

$$\cos \alpha_1 - \cos \alpha_2 = \frac{p(k + \frac{1}{2}) + q\lambda_1}{p^2 + q^2} - \frac{r(l + \frac{1}{2}) + s\lambda_2}{r^2 + s^2}. \quad (8)$$

If  $\alpha_2$  is greater than  $\alpha_1$ ,  $\cos \alpha_1 - \cos \alpha_2 > 0$ . Thus

$$(r^2 + s^2)p(2k + 1) - (p^2 + q^2)r(2l + 1) > 2(p^2 + q^2)s\lambda_2 - 2(r^2 + s^2)q\lambda_1. \quad (9)$$

If the left-hand side of (9) is negative and the right-hand side of (9) is positive, then  $\alpha_1 > \alpha_2$ . If the left-hand side of (9) is positive and the right-hand side of (9) is negative, then  $\alpha_2 > \alpha_1$ . We can easily check the signs of the left-hand side and the right-hand side of (9) with integer computation. Note that  $p, q, k, r, s, l$  are all positive, and that  $(2(p^2 + q^2)s\lambda_2)^2$  and  $(2(r^2 + s^2)q\lambda_1)^2$  contain only integer values.

If the signs of the left-hand side and the right-hand side of (9) are the same, we first compute the square of each side and then compare the values to identify which is the greater. For simplicity, we assume that the signs of the both sides of (9) are positive, and let  $A = (r^2 + s^2)p(2k + 1)$ ,  $B = (p^2 + q^2)r(2l + 1)$ ,  $C = (r^2 + s^2)q$  and  $D = (p^2 + q^2)s$ . Now (9) is rewritten as

$$A - B > 2D\lambda_2 - 2C\lambda_1. \quad (10)$$

Then we take the square of Eq. (10) to obtain

$$(A - B)^2 - 4(C^2\lambda_1^2 + D^2\lambda_2^2) > -8CD\lambda_1\lambda_2. \quad (11)$$

If the sign of the left-hand side of (11) is positive, we can deduce that  $\alpha_2 > \alpha_1$ . Otherwise, taking the square of each side gives us

$$[(A - B)^2 - 4(C^2\lambda_1^2 + D^2\lambda_2^2)]^2 < 64C^2D^2\lambda_1^2\lambda_2^2. \quad (12)$$

We note that we can easily verify whether (12) is satisfied with integer computation alone. If (12) is true,  $\alpha_1 < \alpha_2$ ; otherwise  $\alpha_2 < \alpha_1$ . The same logic can be applied to the case where the signs of the both sides of (9) are negative.

This comparison of a pair of hinge angles is executed in a constant time because in the worst case, we only have to check Eqs. (9), (11), and (12).  $\square$

**Function 2.** Function for finding the lower bound rotation angle for a digital image.

**INPUT:** A set of points  $A$ , a Pythagorean angle  $\theta$

**OUTPUT:** A hinge angle  $\alpha$

$\alpha =$  Function 1 (first point  $p_1$  of  $A$ ,  $\theta$ );

**for all**  $p \in A \setminus \{p_1\}$  **do**

$\alpha_{\text{temps}} =$  Function 1 ( $p$ ,  $\theta$ );

**if**  $\alpha < \alpha_{\text{temps}}$  **then**

$\alpha = \alpha_{\text{temps}}$ ;

**end if**

**end for**

**return**  $\alpha$ ;

#### 5. Digital image rotation by a lower bound rotation angle

This section uses the results obtained in Section 4 to present an algorithm for rotating a set of points with a given lower bound rotation angle.

It is already proved in [1] that we can obtain the same result as the DER with respect to the original rotation angle. Note that our input is a lower bound rotation angle and the input of the algorithm presented in Function 1 is a Pythagorean angle. In spite of this difference, we can apply the same algorithm thanks to Theorem 9, since we are looking for  $K$ , which gives the arriving pixel  $(K, \lfloor \lambda + \frac{1}{2} \rfloor)$ , for each pixel  $(P, Q)$ . The algorithm is presented in Function 3. It supposes that the center of rotation is the origin.

For each point, our algorithm calls the binary search (Function 1) to find the corresponding hinge angle, which designates its new position. If we consider  $n$  as the biggest coordinate of all points in  $A$ , we can assume that there are less than  $4n^2$  points in  $A$ . Thus we can

conclude that the complexity of our algorithm is  $O(n^2 \log(n))$ . The first advantage of our method is that it does not require any floating number calculation. The second advantage is that the exact rotation of the set of points is obtained with only an integer triplet. We need neither matrices nor angles for realizing the rotation.

**Function 3.** Discrete rotation algorithm by a lower bound rotation angle.

**INPUT:** A set of points  $A$ , a lower bound rotation angle  $\alpha$  (hinge angle)

**OUTPUT:** A set of points  $A'$

**for all**  $\mathbf{p} \in A$  **do**

$\alpha_1 = \text{Function 1}(\mathbf{p}, \alpha)$ ;

Move  $\mathbf{p}$  to  $(K, \lfloor \lambda + \frac{1}{2} \rfloor)$  or  $(\lfloor \lambda + \frac{1}{2} \rfloor, K)$ ,

depending on the sign of  $K$  and store the new point in  $A'$ ;

**end for**

**return**  $A'$ ;

### 6. Obtaining ARA from two digital images

Let us assume that a set of grid points in the first image and its corresponding set in the second image are given:  $A = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_l\}$  and  $B = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_l\}$  are given where  $\mathbf{p}_i$  corresponds to  $\mathbf{q}_i$ . Given  $A$  and  $B$ , we obtain a hinge angle pair  $\{\alpha_{\text{inf}}, \alpha_{\text{sup}}\}$ . This pair of hinge angles is the lower and the upper bounds of the ARA. Therefore, each  $\gamma$  such that  $\alpha_{\text{inf}} \leq \gamma < \alpha_{\text{sup}}$  is consistent with the point correspondences between  $A$  and  $B$ . Hereafter, we assume that  $A$  is the original point set and  $B$  is the rotated point set by angle  $\gamma$ . In this section, we show how to obtain the ARA from  $A$  and  $B$ .

#### 6.1. Setting rotation centers

For any rotation, we need to set a rotation center. Without loss of generality, we may choose any grid point in a digital image for the rotation center. Assuming that rotation centers for  $A$  and  $B$  are  $\mathbf{p}_1$  and  $\mathbf{q}_1$ , respectively, we define two translation functions  $\mathcal{T}_A$  and  $\mathcal{T}_B$  such that

$$\mathcal{T}_A(\mathbf{p}_i) = \mathbf{p}_i - \mathbf{p}_1,$$

$$\mathcal{T}_B(\mathbf{q}_i) = \mathbf{q}_i - \mathbf{q}_1$$

for all  $\mathbf{p}_i \in A, \mathbf{q}_i \in B$ . We can regard the origin as the rotation centers after these translations. Hereafter, we assume that these translations have been already applied in order to obtain  $A = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_l\}$  and  $B = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_l\}$ .

#### 6.2. Computing lower and upper bounds of rotation angles from two corresponding point pairs

In this subsection, we consider a special case of two corresponding point pairs.

We let  $A = \{\mathbf{p}_1, \mathbf{p}_2\}$  and  $B = \{\mathbf{q}_1, \mathbf{q}_2\}$  where  $\mathbf{p}_i = (P_i, Q_i)$  and  $\mathbf{q}_i = (R_i, S_i)$ . We then define a circle  $\mathcal{C}(\mathbf{p}_2)$  going through  $\mathbf{p}_2$  whose center is  $\mathbf{p}_1$ . Thus the radius of  $\mathcal{C}(\mathbf{p}_2)$  is  $r = d(\mathbf{p}_1, \mathbf{p}_2)$  where  $d(\mathbf{p}_1, \mathbf{p}_2)$  is the Euclidean distance between  $\mathbf{p}_1$  and  $\mathbf{p}_2$ .

Let us define the half-grid  $\mathcal{H}(\mathbf{q}_2)$  around  $\mathbf{q}_2$ :

$$\mathcal{H}(\mathbf{q}_2) = \{(x, y) \in \mathcal{H} : S_2 - \frac{1}{2} \leq y \leq S_2 + \frac{1}{2} \text{ if } x = R_2 \pm \frac{1}{2} \\ R_2 - \frac{1}{2} \leq x \leq R_2 + \frac{1}{2} \text{ if } y = S_2 \pm \frac{1}{2}\}.$$

We set  $\mathbf{p}_1$  and  $\mathbf{q}_1$  to be the rotation centers. Then, we need to detect intersections between  $\mathcal{C}(\mathbf{p}_2)$  and  $\mathcal{H}(\mathbf{q}_2)$  in order to find a

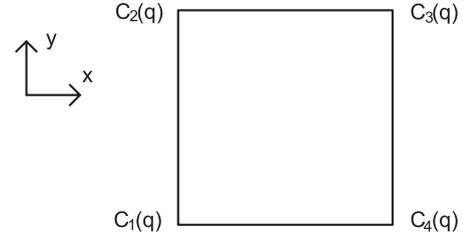


Fig. 2. The corners of  $\mathcal{H}(\mathbf{q})$ , namely, four corners of a pixel around  $\mathbf{q}$ .

hinge angle pair. We thus investigate which corners of  $\mathcal{H}(\mathbf{q}_2)$  are inside of  $\mathcal{C}(\mathbf{p}_2)$ .

Setting four corners of  $\mathcal{H}(\mathbf{q}_2)$  to be  $C_1(\mathbf{q}_2) = (R_2 - \frac{1}{2}, S_2 - \frac{1}{2})$ ,  $C_2(\mathbf{q}_2) = (R_2 - \frac{1}{2}, S_2 + \frac{1}{2})$ ,  $C_3(\mathbf{q}_2) = (R_2 + \frac{1}{2}, S_2 + \frac{1}{2})$ ,  $C_4(\mathbf{q}_2) = (R_2 + \frac{1}{2}, S_2 - \frac{1}{2})$  as shown in Fig. 2, we define a binary function  $\mathcal{F}$ :

$$\mathcal{F}(C_i(\mathbf{q}_2)) = \begin{cases} 1 & \text{if } C_i(\mathbf{q}_2) \text{ is inside of } \mathcal{C}(\mathbf{p}_2), \\ 0 & \text{otherwise.} \end{cases}$$

In order to obtain  $\mathcal{F}(C_i(\mathbf{q}_2))$  with integer calculation, we compare each of  $\|(2(R_2 \pm \frac{1}{2}), 2(S_2 \pm \frac{1}{2}))\|^2$  with  $(2r)^2$ . Note that we may assume that  $\mathcal{C}(\mathbf{p}_2)$  and  $\mathcal{H}(\mathbf{q}_2)$  always intersect with each other. This is because no intersection between  $\mathcal{C}(\mathbf{p}_2)$  and  $\mathcal{H}(\mathbf{q}_2)$  indicates that  $\mathbf{p}_2$  and  $\mathbf{q}_2$  are not corresponding.

The following lemmas are needed to prove Theorem 12.

**Lemma 10.** For a circle  $\mathcal{C}(\mathbf{p}_2)$  centered on  $\mathbf{p}_1$ , any  $C_i(\mathbf{q}_2) = (R_2 \pm \frac{1}{2}, S_2 \pm \frac{1}{2})$  cannot be on  $\mathcal{C}(\mathbf{p}_2)$  for  $i \in \{1, 2, 3, 4\}$  where  $R_2, S_2 \in \mathbb{Z}$ .

**Proof.** Let  $r$  be the radius of  $\mathcal{C}(\mathbf{p}_2)$ . Because  $\mathcal{C}(\mathbf{p}_2)$  goes through  $\mathbf{p}_2$ ,  $r^2 \in \mathbb{Z}$ . Let us assume that  $C_i(\mathbf{q}_2)$  is on  $\mathcal{C}(\mathbf{p}_2)$ . Thus the Euclidean distance between the origin and  $C_i(\mathbf{q}_2)$  is equal to  $r$ . This indicates that  $r^2 = (R_2 \pm \frac{1}{2})^2 + (S_2 \pm \frac{1}{2})^2$ . However, this contradicts the above fact that  $r^2 \in \mathbb{Z}$ .  $\square$

**Lemma 11.** Let  $\mathcal{D}$  be a line that belongs to  $\mathcal{H}$ . If  $\mathcal{C}(\mathbf{p}_2)$  is a circle centered on  $\mathbf{p}_1$ , then, the number of distinct intersections of  $\mathcal{C}(\mathbf{p}_2)$  and  $\mathcal{D}$  is two or zero.

**Proof.** Let  $\mathbf{p}_2 = (P_2, Q_2) \in \mathbb{Z}^2$  and the equation representing  $\mathcal{D}$  be  $x = i + \frac{1}{2}$  where  $i \in \mathbb{Z}$ . Letting  $(x_1, y_1)$  be the coordinates of the intersecting point of  $\mathcal{D}$  and  $\mathcal{C}(\mathbf{p}_2)$ . Then we have

$$\begin{cases} x_1^2 + y_1^2 = P_2^2 + Q_2^2, \\ x_1 = i + \frac{1}{2}. \end{cases}$$

From these equations, we obtain  $y_1^2 = P_2^2 + Q_2^2 - (i + \frac{1}{2})^2$ . Because  $P_2^2 + Q_2^2 - (i + \frac{1}{2})^2$  does not belong to  $\mathbb{Z}$ ,  $y_1$  cannot be equal to 0. Therefore, there are two distinct solutions for  $y_1$  if  $P_2^2 + Q_2^2 - (i + \frac{1}{2})^2 > 0$ ; no solution otherwise. Similar discussion can be done for the case of  $y_1 = j + \frac{1}{2}$  where  $j \in \mathbb{Z}$ .  $\square$

**Theorem 12.** If two points  $\mathbf{p}_2$  and  $\mathbf{q}_2$  are corresponding, the circle  $\mathcal{C}(\mathbf{p}_2)$  and the half-grid  $\mathcal{H}(\mathbf{q}_2)$  always have two or four distinct intersections.

**Proof.** In general, if a circle intersects with a square and the center of the circle is not inside of the square, we have 1, 2, or 4 intersections. Having just one intersection means that the circle goes through one corner of the square or the circle is tangential to a half-grid. Lemma 10 shows that the circle cannot go through one corner. Lemma 11 shows that the circle cannot be tangential to any half-grid. Therefore, we have only 2 or 4 intersections.  $\square$

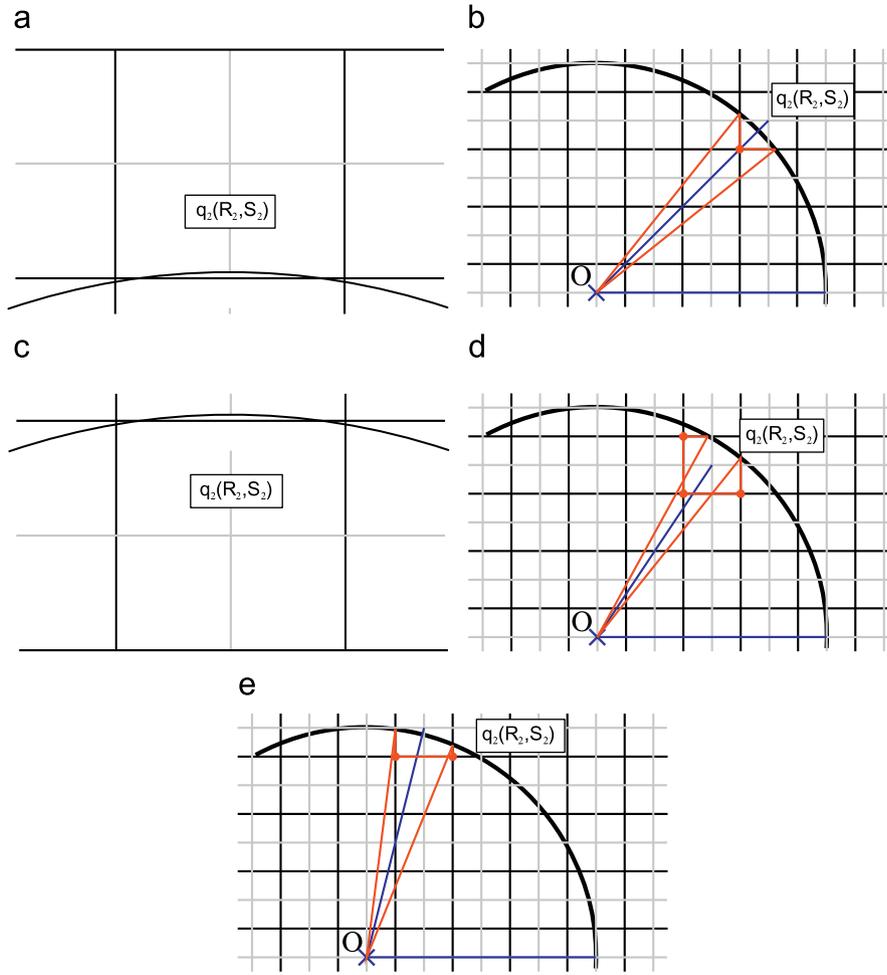


Fig. 3. Illustration of cases  $\sum \mathcal{F}(C_i(\mathbf{q}_2)) = 0$  (a), 1 (b), 2 (c and d), or 3 (e).

From Theorem 12, we always have two or four distinct intersections between  $\mathcal{C}(\mathbf{p}_2)$  and  $\mathcal{H}(\mathbf{q}_2)$ , and we see that there are five cases corresponding to different possibilities to have 0, 1, 2, or 3 corners inside of  $\mathcal{C}(\mathbf{p}_2)$ , as illustrated in Fig. 3. Note that in [6], Proposition 5 which corresponds to Theorem 12 in this paper shows that there are always two intersections. During the period of writing this extended paper, we discovered a hypothetical case where four intersections arise.

**Remark 13.** The cases represented in (a) and (c) in Fig. 3 never happen unless two conditions are satisfied.

- (1)  $\mathbf{q}_2$  is on the  $x$ -axis or on the  $y$ -axis.
- (2) The radius  $r$  of  $\mathcal{C}(\mathbf{p}_2)$  is sufficiently close to a half-integer.

The case of Fig. 3(c) is problematic because it generates two distinctive ranges of ARA. These two ranges are symmetric with respect to the  $x$ -axis or the  $y$ -axis. Note that this case was not addressed in [6]; it represents the case of four intersections discovered during the redaction of this paper. However, this case rarely occurs as explained below.

Supposing that  $\mathbf{q}_2$  is neither on the  $x$ -axis nor the  $y$ -axis, we see that neither  $R_2$  nor  $S_2$  is zero. We assume that they are positive. In the first quadrant, the  $y$ -coordinate (respectively, the  $x$ -coordinate) of points in  $\mathcal{C}(\mathbf{p}_2)$  is strictly decreasing with respect to  $x$  (respectively,  $y$ ). Thus it cannot intersect twice with a line parallel to the  $x$ -axis (respectively, the  $y$ -axis).

Supposing that  $\mathbf{q}_2$  is on the  $y$ -axis, we see that the distance between the origin and  $C_1(\mathbf{q}_2)$  (respectively,  $C_3(\mathbf{q}_2)$ ) is greater (respectively, smaller) than  $r$ . Thus we have  $r < \sqrt{(R_2 - \frac{1}{2})^2 + (\frac{1}{2})^2}$  (respectively,  $r > \sqrt{(R_2 + \frac{1}{2})^2 + (\frac{1}{2})^2}$ ). Letting  $\varepsilon = r - (R_2 - \frac{1}{2})$  (respectively,  $\varepsilon = r - (R_2 + \frac{1}{2})$ ) where  $\varepsilon < \frac{1}{2}$ , we obtain  $\varepsilon(8r - 4\varepsilon) < 1$  (in both cases). We experimentally observe that as far as  $r \leq 10^5$ ,  $\varepsilon(8r - 4\varepsilon) < 1$  never holds. On the other hand, if  $r$  is sufficiently large enough to have  $8r - 4\varepsilon \approx 8r$  then we have  $\varepsilon < 1/8r$ , which means that  $r$  is sufficiently close to a half-integer. We can thus conclude that (2) in Remark 13 is satisfied only if  $r$  is sufficiently large. Note that a similar discussion can be done when supposing that  $\mathbf{q}_2$  is on the  $x$ -axis.

As explained above, the case of Fig. 3(c) occurs extremely rarely. Thus, we take no account of this case from now on, which causes no problem from the practical point of view.

The main function of our algorithm for finding the lower and the upper bounds of ARA, consists of three steps. The first step sets the rotation center at  $\mathbf{p}_1$  and  $\mathbf{q}_1$ , as described in Section 6.1. The second step computes which corners are inside of  $\mathcal{C}(\mathbf{q}_2)$  and then compute the index  $I_{\mathbf{q}_2} = \sum_i 2^i \times \mathcal{F}(C_i(\mathbf{q}_2))$ . Therefore, we can easily identify which corners are inside of  $\mathcal{C}(\mathbf{p}_2)$  from  $I_{\mathbf{q}_2}$ . The third step calls a function that returns hinge angles corresponding to  $I_{\mathbf{q}_2}$ . There exist fourteen possible values for  $I_{\mathbf{q}_2}$  from 0 till 15 except for 5 and 10. Note that geometrically  $I_{\mathbf{q}_2}$  can be neither 5 nor 10. The value 15 of  $I_{\mathbf{q}_2}$  implies an error such that all corners are inside of  $\mathcal{C}(\mathbf{q}_2)$ . Since  $I_{\mathbf{q}_2}$  whose value is 0 corresponds to the case  $\sum \mathcal{F}(C_i(\mathbf{q}_2)) = 0$ ,

we should verify whether  $\mathcal{H}(\mathbf{q}_2)$  really intersects with  $\mathcal{C}(\mathbf{p}_2)$ . Note that for the other values for  $l_{q_2}$ , we can make a pair  $(d, e)$  such that  $d + e = 15$ . The two indices of each pair design the same pair of lower and upper bounds of ARA. Table 1 gives the corresponding lower and upper bound rotation angles for each value of  $l_{q_2}$ . Each step of this algorithm has the constant time complexity. Thus the global complexity of this algorithm is also  $O(1)$ .

### 6.3. Incremental computing lower and upper bounds of rotation angles

In general, the corresponding point sets contain more than two points. Therefore, in this section, we extend our algorithm in Section 6.2 to two sets of corresponding point pairs,  $A$  and  $B$ , each of which has  $l$  points where  $l > 2$ .

To simplify the notation, we denote by  $ARA(\mathbf{p}_i, \mathbf{q}_i) = (\alpha_{i\text{inf}}, \alpha_{i\text{sup}})$  the pair of angles that gives the lower and the upper bounds of ARA for the pair of points  $(\mathbf{p}_i, \mathbf{q}_i)$ . Note that  $\alpha_{i\text{inf}}, \alpha_{i\text{sup}}$  are hinge angles.  $ARA(A_n, B_n)$  denotes the two most restrictive angles for all points  $i$

**Table 1**  
Corners of  $\mathcal{H}(\mathbf{q}_2)$  inside of  $\mathcal{C}(\mathbf{p}_2)$  and ARA.

Value of $l_{q_2}$	$\alpha_{\text{inf}}$	$\alpha_{\text{sup}}$
$l_{q_2} = 1$ or $14$	$\alpha(P_1, Q_1, R_2 - 1)$	$\alpha(P_1, Q_1, 1 - S_2)$
$l_{q_2} = 2$ or $13$	$\alpha(P_1, Q_1, R_2 - 1)$	$\alpha(P_1, Q_1, -S_2)$
$l_{q_2} = 3$ or $12$	$\alpha(P_1, Q_1, 1 - S_2)$	$\alpha(P_1, Q_1, -S_2)$
$l_{q_2} = 4$ or $11$	$\alpha(P_1, Q_1, R_2)$	$\alpha(P_1, Q_1, -S_2)$
$l_{q_2} = 6$ or $9$	$\alpha(P_1, Q_1, R_2 - 1)$	$\alpha(P_1, Q_1, R_2)$
$l_{q_2} = 7$ or $8$	$\alpha(P_1, Q_1, R_2)$	$\alpha(P_1, Q_1, 1 - S_2)$
$l_{q_2} = 0$ and $R_2 = 0$	$\alpha(P_1, Q_1, 1 - S_2)$	$\alpha(P_1, Q_1, 1 - S_2)$
$l_{q_2} = 0$ and $S_2 = 0$	$\alpha(P_1, Q_1, R_2 - 1)$	$\alpha(P_1, Q_1, R_2 - 1)$

such as  $i \leq n$ . We recursively define it by

$$ARA(A_n, B_n) = ARA(A_{n-1}, B_{n-1}) \cap ARA(\mathbf{p}_n, \mathbf{q}_n).$$

A new algorithm handles all points incrementally. This algorithm is divided into two parts. The first part is to initialize the algorithm by computing  $ARA(\mathbf{p}_2, \mathbf{q}_2)$ . The second part computes  $ARA(A_i, B_i)$  for  $i = l$ . Note that  $ARA(\mathbf{p}_1, \mathbf{q}_1)$  cannot be computed because  $\mathbf{p}_1$  and  $\mathbf{q}_1$  are the centers of the rotation.

The time complexity of this algorithm is  $O(l)$ . As explained in Section 6.2, the function giving a pair of lower and upper bound rotation angles from a pair of points is realized in a constant time  $O(1)$ . Moreover, as explained in Section 3, we can compare two hinge angles in a constant time  $O(1)$ . Therefore, the computation of this algorithm for  $l$  points takes the time complexity of  $l \times (O(1) + O(1)) = O(l)$  as a whole.

Fig. 4 gives an example of the incremental algorithm for two sets of three points. Given input data of the algorithm as shown in Fig. 4(a), we first obtain the result of the translation described in Section 6.1 as illustrated in (b). We then compare, for each pair of points  $(\mathbf{p}_i, \mathbf{q}_i)$  with  $i \geq 2$ , the distance of  $\mathbf{p}_i$  from the origin with that of each corner from  $\mathcal{H}(\mathbf{q}_i)$  to deduce the corresponding hinge angle as explained in Section 6.2. Finally, we obtain (d) which shows the intersection of all  $ARA(\mathbf{p}_i, \mathbf{q}_i)$  obtained in (c).

## 7. Experimental evaluation using synthetic data

We evaluated our algorithm using synthetic data and verified the quality of obtained ARA. To test our algorithm, we need two sets of points. We randomly generated the set  $F$  of 100 floating points in a  $200 \times 200$  square. The first set  $I$  is obtained by discretizing  $F$ . The second set  $I'$  is obtained by applying the DER with angle  $\theta = 50^\circ$  to

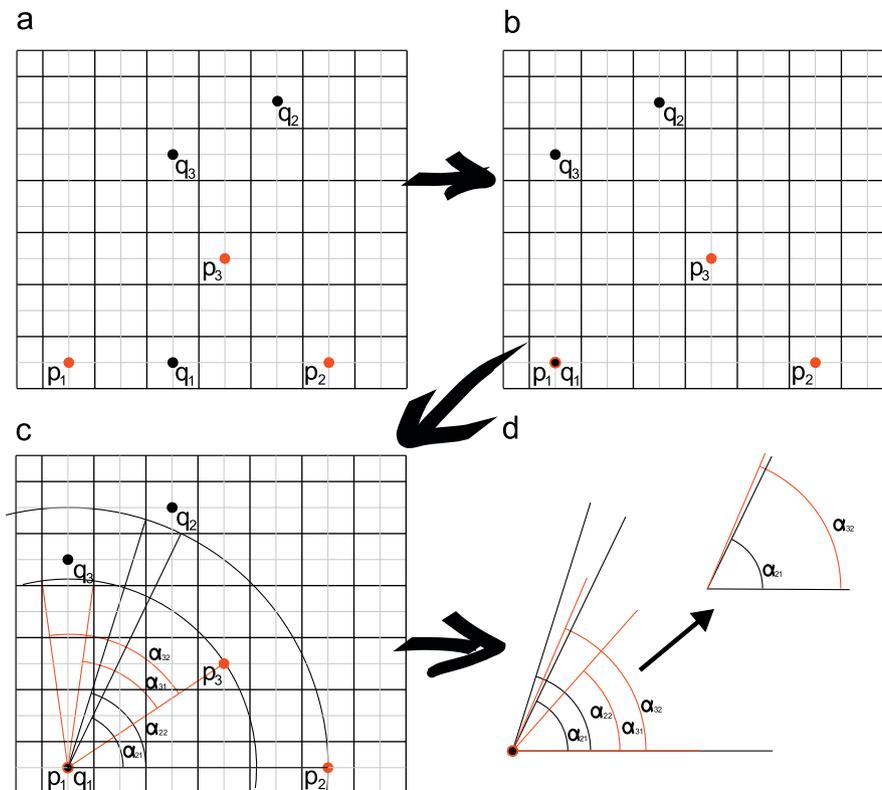


Fig. 4. Running of the incremental algorithm.

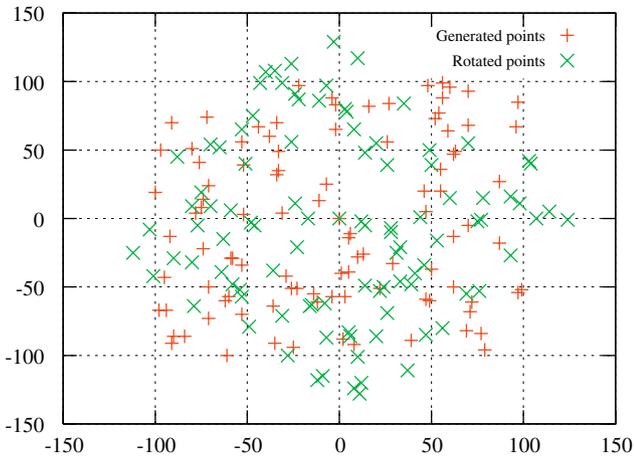


Fig. 5. Randomly generated points and their corresponding rotated points.

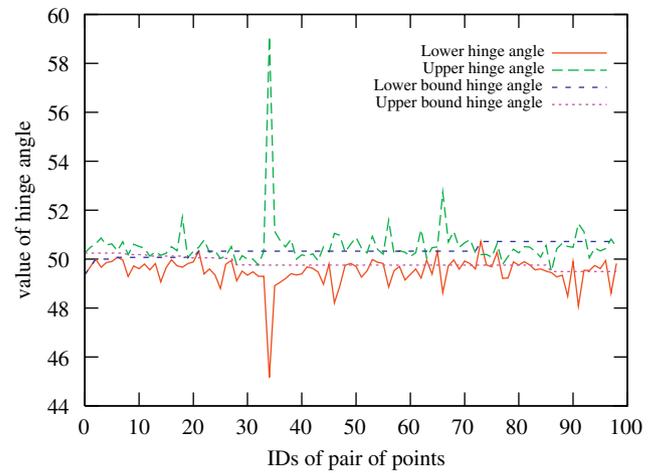


Fig. 7. Result of our algorithm applied on a sets of floating points.

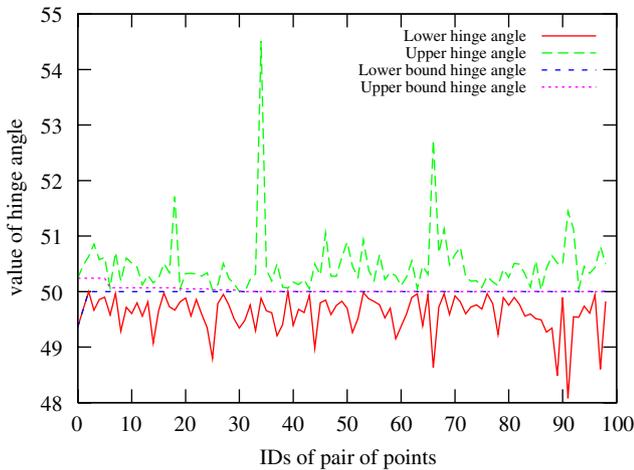


Fig. 6. Result of our algorithm applied on the sets of points in Fig. 5.

the first set  $I$  (Fig. 5). We assumed that the point correspondences across the two sets  $I$  and  $I'$  are known.

Fig. 6 shows the hinge angles obtained by our algorithm. The green and the red curves give the lower and the upper bound rotation angles for each pair of point in correspondence. We can see that lower bound rotation angles are always lower than  $\theta$  and that upper bound rotation angles are greater than  $\theta$ . The blue and purple lines give the lower and the upper bound rotation angles for points with the lower IDs than the point of interest. As we can see, after only ten pairs of points, the range of ARA becomes less than  $0.1^\circ$  while the range is reduced to  $0.02^\circ$  after twenty pairs of points. Then the precision increases slowly. This shows that the precisions of ARA acquired after twenty pairs of points are not significant in this particular case. Fig. 6 also indicates that most pairs of points give the ARA range smaller than  $1^\circ$  (for example, pair #39). Pair #34 and pair #66, however, give a range greater than  $4^\circ$ . In fact, the maximum range for a pair of points is directly related to the distance between the center of rotation and the pair of points. We denote by  $d$  the distance between the rotation center and the pair of points. Then, the maximum difference between the lower and the upper bound rotation angles is  $\sin^{-1}(\sqrt{2}/d)$ . For the pair #34,  $d = \sqrt{173}$ ; thus the maximum range is approximately  $6.1^\circ$ . For the pair #39,  $d = \sqrt{16505}$  and thus the maximum range is approximately  $0.63^\circ$ . This is consistent with our experimental result.

## 8. Discussion on practical application to digital images

In the previous section, we start with grid points in the discrete plane. In other words, all coordinates of the points are integers. But in reality, points in the Euclidean space are not represented by integers but real numbers. Thus we start here with the set of floating points to see how our algorithm works. Then we test our algorithm with real data acquired by a digital camera.

### 8.1. Synthetic data

To test our algorithm using synthetic data more similar to real data, we applied the Euclidean rotation with angle  $\theta = 50^\circ$  directly to the set  $F$  that is used in Section 7. Then we discretized the rotated image to obtain the second set  $I'$ . To the two sets  $I$  and  $I'$ , we applied our algorithm. We note that the set  $I$  comes from Section 7.

Fig. 7 shows the hinge angles obtained. Because we started with the set of floating points, we observe some errors in bounding ARA. In fact, we see that some pairs of points, the pairs #64 and #73 for instance, do not contain  $\theta$ .

Fig. 8 (left) explains how errors arise in rotating floating points. Let us assume that point  $q_1$  is obtained after the Euclidean rotation of the floating point  $p_1$  with angle  $\gamma$ . Let  $p_2$  and  $q_2$  denote the discretization of  $p_1$  and  $q_1$ . If we apply our algorithm to the pair of points  $\{p_2, q_2\}$ , we obtain  $ARA(p_2, q_2) = (\alpha_{inf}, \alpha_{sup})$ . In this case,  $\gamma$  does not belong to the interval  $[\alpha_{inf}, \alpha_{sup}]$  (Fig. 8 (right)). We see that this is caused by discretization of the floating points. We can give a bound for this error. In fact, this bound directly depends on the distance between the rotation center and the pair of points. If we denote by  $d$  this distance, the maximum error is equal to  $\sin^{-1}(\sqrt{2}/(2d))$ . Note that the pair of points  $\{p_3, q_3\}$  and the angle  $\rho$  bring the same problem.

To avoid this problem, we modify our algorithm for example as follows. We keep all the pairs of bound rotation angles determined by all pairs of points. After sorting all lower and upper bound rotation angles into two lists  $\mathcal{L}_{inf}, \mathcal{L}_{sup}$ , we remove all hinge angles  $\alpha_{inf}$  from  $\mathcal{L}_{inf}$  such that for  $\exists \alpha_{sup} \in \mathcal{L}_{sup}, \alpha_{inf} > \alpha_{sup}$ . We also remove all hinge angles  $\alpha_{sup}$  from  $\mathcal{L}_{sup}$  such that for  $\exists \alpha_{inf} \in \mathcal{L}_{inf}, \alpha_{sup} < \alpha_{inf}$ .

With this procedure, we can guarantee that the lower bound hinge angle is smaller than the rotation angle applied to the points and that the upper bound hinge angle is larger than the rotation angle. The complexity  $O(n \log n)$  is required in sorting the remaining hinge angle pairs, which does not increase the computational cost of the algorithm as a whole.

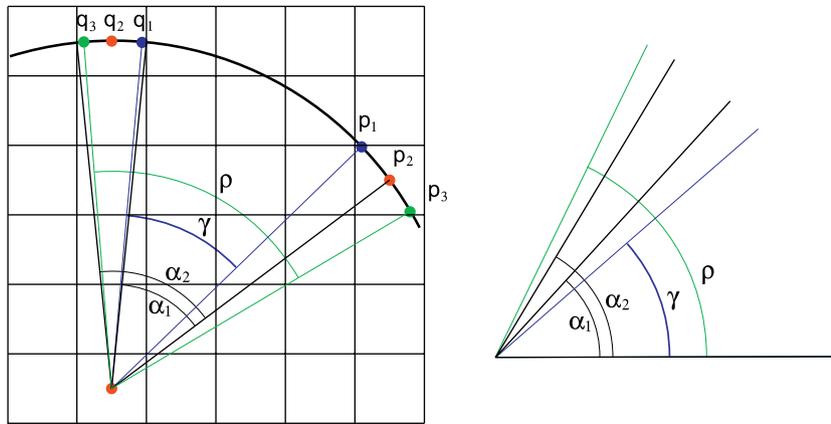


Fig. 8. Examples of errors for computing lower and upper bounds of rotation angles introduced by rotation of floating points.

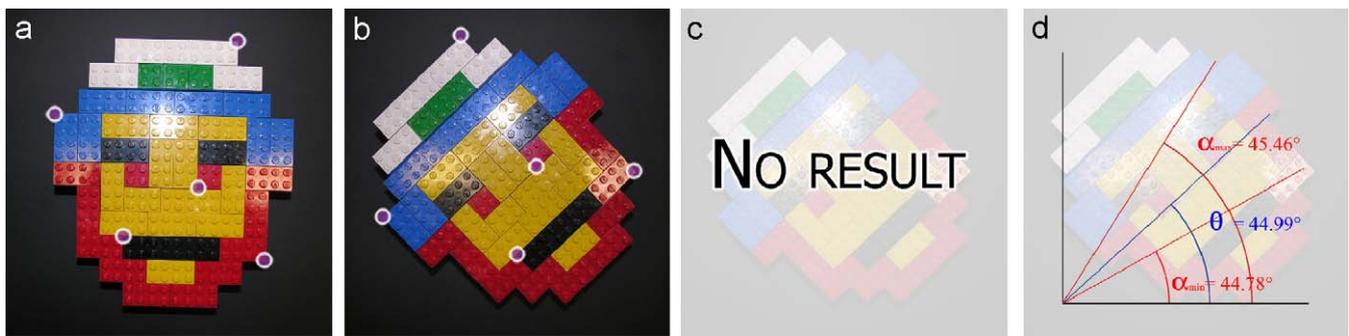


Fig. 9. Result of our algorithm applied on real data.

Another way to avoid the problem is to compute the smallest distance  $d$  of all the points from the rotation center and then compute a Pythagorean angle  $\theta$  satisfying  $\theta > \sin^{-1}(\sqrt{2}/(2d))$ . We then add (subtract)  $\theta$  to the upper (lower) bound rotation angle obtained by our algorithm in Section 6. As a result we obtain the upper and the lower bound rotation angles that define the interval accurately including the true rotation angle. Thanks to Theorem 7, the addition of a Pythagorean angle does not change the nature of the upper (lower) bound rotation angle.

Both methods return a valid ARA. It is preferable to use the second method when there are few pairs of points because it does not remove any hinge angles. The first method is preferable when data contains many pairs of points in correspondence because the obtained ARA is more restrictive than the ARA obtained by the second method.

8.2. Real data

We applied our algorithm to see its practical usefulness. We used a turntable that is rotated with respect to the vertical axis with respect to a digital image plane. The precision of rotations in control is  $10^{-3}^\circ$ .

We put a toy block on the turntable and then took its image using a standard digital camera where the camera was fixed so that its optical axis is parallel with the rotation axis (Fig. 9(a)). Next we rotated the turntable with the angle of  $44.99^\circ$  and then took another image of the toy block by the fixed camera (Fig. 9(b)).

We manually selected five points in the first image and their corresponding points in the second image. Then we applied our algorithm to the five pairs of corresponding points.

Our algorithm might return the empty result (Fig. 9(c)). This is because no intersection between  $\mathcal{C}(p_i)$  and  $\mathcal{H}(q_i)$  was found for all  $i = 1, 2, \dots, 5$ .

In the case of real data, we cannot always guarantee to detect correct corresponding pairs of points even manually. This indicates that a problem different from the discretization problem (see Section 8.1) arises.<sup>3</sup> Namely the maximum difference of distances from the rotation center between two points in correspondence can become greater than  $\sqrt{2}$ . This is illustrated in Fig. 10 where points  $p$  and  $q$  are in correspondence and  $\varepsilon_q = |d_p - d_q|$  is greater than  $\sqrt{2}$  where  $d_p, d_q$  are, respectively, the distances from  $p, q$  to the origin. We see that no intersection exists between the circle  $\mathcal{C}(p)$  going through  $p$  and  $\mathcal{H}(q)$ .

To avoid this problem on corresponding points, we can change the resolution of the images. Namely, we degrade the image resolution until we find an intersection between  $\mathcal{C}(p)$  and  $\mathcal{H}(q)$ .<sup>4</sup> Because of the change in image resolution, we cannot accept the obtained hinge angles as they are. Instead, we add (subtract) a Pythagorean angle  $\theta$  to (from) the obtained upper (lower) hinge angle. Let us assume that  $d_p < d_q$  and that  $\mathcal{H}(q)$  is defined for the image whose resolution is degraded with  $2^{-n}$  from the original one. Then we have to choose  $\theta$  satisfying  $\theta > (n\sqrt{2})/(2d_p)$ . As mentioned in Section 8.1, thanks to Theorem 7, the addition of a Pythagorean angle does not change the nature of the upper (lower) bound rotation angle.

Fig. 9(d) shows the results obtained by the modification above to the same data. We can see that the real rotation angle ( $44.99^\circ$ ) is included in the ARA obtained by our algorithm with this mod-

<sup>3</sup> We assume here that we are to find ARA from given correspondences.

<sup>4</sup> There are other criteria for finding a reasonable image resolution. For example, Brimkov presents criteria of faithfully digitization for continuous objects in [11].

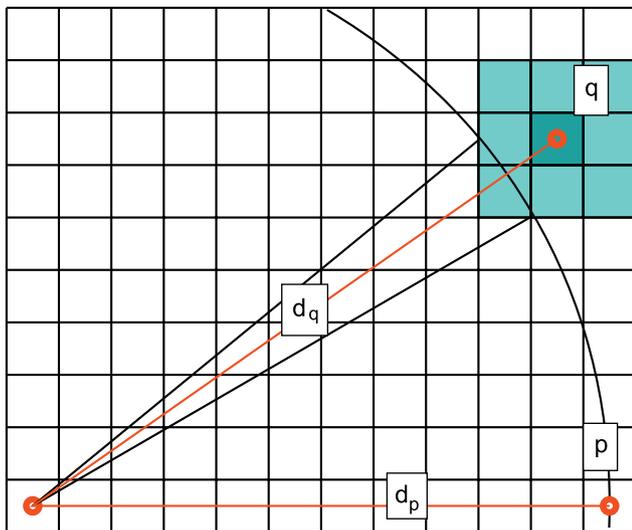


Fig. 10. Problem of corresponding points with real data.

ification. Accordingly, we can conclude that our modification is effective.

## 9. Conclusion

In this paper, we have introduced problems of rotations in the discrete space. Because rotations are mathematically defined for the continuous space, it is necessary to develop new operational tools for such discrete rotations.

Our problem was from two digital images to find all possible rotation angles to rotate the first image into the second, namely, to find the ARA between two digital images. Since we need to have only exact computation without any approximation, we decide to adopt hinge angles which allow us rotations using only integer computation. By using their properties, we have proposed a method for incrementally computing the lower/upper bound rotation angles from a Pythagorean angle for a set of discrete points. Based on this method, we have also proposed a linear algorithm for finding from two digital images the lower and the upper bound rotation angles representing the ARA between them.

In addition, we have shown with experiments that our algorithm is efficient with synthetic data and gives consistent results. Then we also have done some experiments to real data. Results given by our algorithm were not consistent because of the loss of correct point correspondences in our input. We then explained why these problems appear and gave necessary modifications to obtain consistent results. These modifications did not increase the time complexity while keeping the integer computation.

**About the Author**—YOHAN THIBAULT received his license degree in Computer Science from the University of Luminy, Marseille France in 2004. In 2006 he obtained his master degree in Computer Science from the University of Luminy. His master thesis on SAT problems and their complexities was accomplished under the supervision of Prof. Nadia Creignou. In November 2006, he joined A2SI Laboratory and Institute of Gaspard-Monge of Université Paris-Est as a Ph.D. candidate under the supervision of Prof. Michel Couprie and Dr. Yukiko Kenmochi. His Ph.D. research subject is on discrete geometry tools and their application in computer vision. Since November 2007, he has been visiting the National Institute of Informatics in Japan under the framework of the France–Japan doctoral college where he is supervised by Prof. Akihiro Sugimoto.

**About the Author**—YUKIKO KENMOCHI received her B. Eng., M. Eng., and D. Eng. degrees in Information and Computer Sciences from Chiba University, Japan, in 1993, 1995, and 1998, respectively. She joined Japan Advanced Institute of Science and Technology as a Research Associate in 1998, and Okayama University, Japan, as a Lecturer in 2003. Since 2004, she has been a CNRS Researcher at Gaspard-Monge Institute, Université Paris-Est, France, and a member of the A2SI laboratory, ESIEE. She received a Best Paper Award of Information and System Society from IEICE in 2005. Her research interest includes discrete geometry for computer imagery.

**About the Author**—AKIHIRO SUGIMOTO received his B.Sc., M.Sc., and D.Eng. degrees in mathematical engineering from the University of Tokyo in 1987, 1989, and 1996, respectively. After working at Hitachi Advanced Research Laboratory, ATR, and Kyoto University, he joined the National Institute of Informatics, Japan, where he is currently a Professor. From 2006 to 2007, he was a visiting Professor at ESIEE, France. He received a Paper Award from the Information Processing Society in 2001. He is a member of IEEE. He is interested in mathematical methods in engineering. In particular, his current main research interests include discrete mathematics, approximation algorithm, vision geometry, and modeling of human vision.

The range size of ARA can be considered as the unreliability of the point correspondence. In other words, larger the range of ARA between two points, less is the reliability of their point correspondence. Therefore, it may be interesting if we can propose a new method for evaluating “good” point correspondence from a given pair of digital images by using our results.

In this paper we always assume that centers of rotations are on integer points. But with real data it is not always the case. One of our future work will be to adapt the presented method to such rotations. In this purpose, studies on digital discs whose centers are not only on integer points [12,13] would help.

Another future work is to extend our proposed method to the 3D space. In 3D, hinge angles are not yet defined and hence the first direction we should take is to design 3D hinge angles and then develop a 3D discrete rotation algorithm based on 3D hinge angles. Then we will improve such a rotation algorithm in order to find ARA in 3D as we did for 2D in this paper.

## Acknowledgments

This work was supported by Grants from Région Ile-de-France and the French Foreign Ministry.

## References

- [1] B. Nouvel, E. Rémila, Incremental and transitive discrete rotations, in: R. Reulke, U. Eckardt, B. Flash, U. Knauer, K. Polthier (Eds.), *Combinatorial Image Analysis, Lecture Notes in Computer Science*, vol. 4040, Springer, Berlin, 2006, pp. 199–213.
- [2] M. Hansen, P. Anandan, K. Dana, G. van der Wal, P. Burt, Real-time scene stabilization and mosaic construction, in: *Applications of Computer Vision, IEEE Computer Society*, NJ, 1994, pp. 54–64.
- [3] J.E. Volder, The cordic trigonometric computing technique, *IRE Transactions on Electronic Computers EC-8* (1959) 330–334.
- [4] E. Andres, *Cercles discrets et rotations discrètes*, Ph.D. Thesis, Université Louis Pasteur, Strasbourg, France, 1992.
- [5] E. Andres, The quasi-shear rotation, in: S. Miguet, A. Montanvert, S. Ubéda (Eds.), *Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science*, vol. 1176, Springer, Berlin, 1996, pp. 307–314.
- [6] Y. Thibault, Y. Kenmochi, A. Sugimoto, Computing admissible rotation angles from rotated digital images, in: V. Brimkov, R. Barneva, H. Hauptman (Eds.), *Combinatorial Image Analysis, Lecture Notes in Computer Science*, vol. 4953, Springer, Berlin, 2008, pp. 99–111.
- [7] R.D. Hersch, Raster rotation of bilevel bitmap images, in: C. Vandoni (Ed.), *Eurographics*, vol. 85, Elsevier, Amsterdam, 1985, pp. 295–308.
- [8] A.W. Paeth, A fast algorithm for general raster rotation, in: A. Glassner (Ed.), *Graphics Gems*, Academic Press, New York, 1990, pp. 179–195.
- [9] B. Nouvel, *Rotations discrètes et automates cellulaires*, Ph.D. Thesis, Ecole Normale Supérieure de Lyon, France, 2006.
- [10] W.S. Anglin, Using pythagorean triangles to approximate angles, *American Mathematical Monthly* 95 (6) (1988) 540–541.
- [11] V.E. Brimkov, Scaling of plane figures that assures faithful digitization, in: V. Brimkov, R. Barneva, H. Hauptman (Eds.), *Combinatorial Image Analysis, Lecture Notes in Computer Science*, vol. 4953, Springer, Berlin, 2008, pp. 87–98.
- [12] M.N. Huxley, J.D. Zunic, The number of  $n$ -point digital discs, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (1) (2007) 159–161.
- [13] B. Nagy, An algorithm to find the number of the digitizations of discs with a fixed radius, *Electronic Notes in Discrete Mathematics* 20 (2005) 607–622.