Computing Admissible Rotation Angles from Rotated Digital Images

Yohan Thibault[1,2], Yukiko Kenmochi[1], Akihiro Sugimoto[1,2]

[1] Université Paris Est, A2SI-ESIEE, France
[2] National Institute of Informatics, Japan
thibauly@esiee.fr, y.kenmochi@esiee.fr, sugimoto@nii.ac.jp

Abstract. Rotations in the discrete plane are important for many applications such as image matching or construction of mosaic images. In this paper, we propose a method for estimating a rotation angle such that the rotation transforms a digital image A into another digital image B. In the discrete plane, there are many angles that can give the rotation from A to B, called admissible angles for the rotation from A to B. For such a set of admissible angles, there exist two angles α_1, α_2 that are its upper and lower bounds. To find those upper and lower bounds, we use hinge angles as used in Nouvel and Rémila [5]. Hinge angles are particular angles determined by a digital image, such that any angle between two consecutive hinge angles gives the identical digital image after the rotation with the angle. Our proposed method obtains the upper and lower bounds of hinge angles from a given Euclidean angle and from a pair of digital images.

1 Introduction

Rotations in the discrete plane are required in many applications for image computation such as image matching or construction of mosaic images [4]. For the moment, the method to estimate the rotation angle is to approximate the rotation matrix by minimizing errors [4]. In the continuous plane, the Euclidean rotation is well defined and possesses the property of bijectivity. This implies that for two angles γ_1, γ_2 and a set of points A, if the Euclidean rotation of angle γ_1 applied to A gives the same result as the Euclidean rotation of angle γ_2 applied to A, then we have $\gamma_1 = \gamma_2$.

In the discrete plane, however, the property of bijectvity does not hold. To understand this reason, we have to first define the discretized Euclidean rotation, abbreviated to DER hereafter. DER is the discretization of the Euclidean rotation, namely, the application of the rounding function after applying the rotation matrix to a set of points. Thus two points in the Euclidean plane may give the same point in the discrete plane after the discretization. Because of this reason, two angles γ_1, γ_2 give the same result for a set of points A even if $\gamma_1 \neq \gamma_2$. In other words, we can define the admissible rotation angles S such that any angle in S gives the same rotation result for a set of points A. Note

V.E. Brimkov, R.P. Barneva, H.A. Hauptman (Eds.): IWCIA 2008, LNCS 4958, pp. 99–111, 2008.

that S depends on A. Another way to define the admissible rotation angles is for two corresponding sets of points A and B, where B is the rotation of A by an unknown angle, to find the set S of angles which transforms A into B. The two most interesting angles in S are the upper and the lower bounds because with only these two angles we can deduce the other angles in S. Therefore, the aim of this paper is to find these two angles from a given rotation angle or from two given corresponding sets of grid points. Because we identify the exact bounds, we have to avoid computation with real numbers. Thus, in this paper, we only work with discrete geometry tools which guarantee to avoid computation errors. Moreover, because we assume that our data are discretized from continuous images of an object, the discrete rotation between two different sets of points has to give the same result as DER.

Some work on discrete rotations already exists. The first discrete rotation is the CORDIC algorithm [6]. Estimation of the rotation angle is done by addition or subtraction using pre-computed values to achieve the needed precision. It gives almost the same result as DER but an approximation of the angle. Andres described in [1],[2] some discrete rotations such as the rotation by discrete circles, the rotation by Pythagorean lines or the quasi-shear rotation. Computation done during these rotations are exact, but they are bijective. Thus they cannot give the same results as DER.

On the other hand, Nouvel and Rémila proposed in [5] another discrete rotation based on hinge angles which gives the same results as DER. It is known that hinge angles are particular angles determined by a digital image, such 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 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 upper bounds of admissible rotation angles. Our method uses hinge angles, because we can obtain the same result as DER and they allow exact computations. The input data of our method is two sets of points A and B, where point correspondences across the two sets are known. The output of the algorithm is two hinge angles that give the lower and upper bounds of the admissible rotation angles for A and B.

In the following of this paper, we first introduce the notion of hinge angles and their properties. Then, we show how to obtain such a hinge angle from a given angle so that we can efficiently obtain a rotated digital image from the integer triplet. We then present a method for obtaining from a pair of digital images, two hinge angles which constitute the upper and lower bounds of the admissible rotation angles from the pair of digital images.

2 Hinge Angles

Let us consider points of $\mathbb{Z}^2$ as centers of pixels and rotate them such that the rotation center has integer coordinates. Hinge angles are particular angles which make some points of $\mathbb{Z}^2$ 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.

2.1 Definition of Hinge Angles

Let $\boldsymbol{x}$ be a point in $\mathbb{R}^2$ such that $\boldsymbol{x} = (x, y)$. We say that $\boldsymbol{x}$ 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 denoted by $\mathscr{H}$, and is called the half grid. Thus, $\mathscr{H}$ represents the set of points on the frontiers of all pixels whose centroids are points in $\mathbb{Z}^2$.

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

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

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

Proposition 1. *An angle α is a hinge angle if there is an integer triple (P, Q, K) such that*

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

The proof is given in [5].

Geometrically, a hinge angle α is formed by two rays that go through (P, Q) and a half-grid point such as $(K + \frac{1}{2}, \lambda)$ respectively sharing the origin as their endpoints as shown in Figure 1 (left). From this proposition, all calculations related to hinge angles can be done only with integers. Hereafter, α indicates a hinge angle.

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

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

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

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

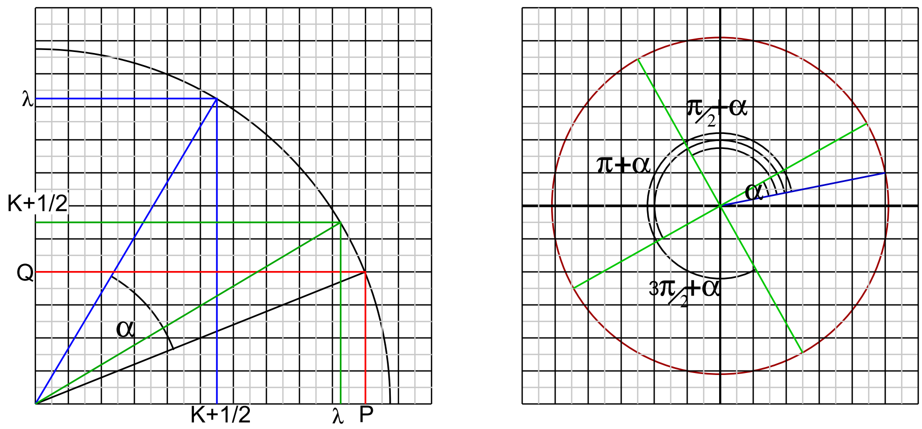

Fig. 1. A hinge angle $\alpha(P, Q, K)$ (left) and four symmetrical hinge angles (right)

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

Corollary 1. *Each triple (P, Q, K) corresponds to four symmetrical hinge angles such as $\alpha + \frac{\pi k}{2}$ where $k = 0, 1, 2, 3$.*

Figure 1(right) gives an example of Corollary 1. In order to distinguish the case $(K + \frac{1}{2}, \lambda)$ from the case $(\lambda, K + \frac{1}{2})$, 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 $(\lambda, K + \frac{1}{2})$. Note that the symmetries allow us to restrict α to the range $[0, \frac{\pi}{2}]$. Thus we know that K is always positive.

2.2 Properties Related to Pythagorean Angle

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

Definition 2. *An angle θ is called Pythagorean if both its cosine and sine belong to the set of rational numbers $\mathbb{Q}$.*

We can deduce from Definition 2 that each Pythagorean angle θ is defined by an integer triplet (a, b, c) such that

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

In the following, θ indicates a Pythagorean angle. The following lemma is needed for the proof of the next proposition.

Lemma 1. *Let (a, b, c) be an integer triplet generating a Pythagorean angle with $\mid a \mid < \mid b \mid < \mid c \mid$. If $\gcd(a, b, c) = 1$, then c is odd.*

Proof. We assume that c is even such that $c = 2d$ where d in $\mathbb{Z}$. Then we obtain a and b are both odd because of $a^2 + b^2 = c^2 = (2d)^2$. Otherwise, we would have $\gcd(a, b, c) = 2n$ for $n \in \mathbb{Z}$. Setting $a = 2e + 1$ and $b = 2f + 1$ where $e, f \in \mathbb{Z}$, we obtain $(2e + 1)^2 + (2f + 1)^2 = 4d^2$, which can be rewritten by $e^2 + e + f^2 + f + \frac{1}{2} = d^2$. This indicates that d does not belong to $\mathbb{Z}$. We therefore conclude that c is odd.

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

Proposition 2. *Let E_h be the set of hinge angles and E_p be the set of Pythagorean angles. Then we have $E_h \bigcap E_p = \emptyset$.*

Proof. Assume that there exists an angle α such that $\alpha \in E_h$ and $\alpha \in E_p$. Since α in E_p, we can find an integer triplet (a, b, c), generating α such that $\gcd(a, b, c) = 1$. By substitution of (4) in (2), we obtain

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

from which we derive $2\frac{Qa+Pb}{c} \in \mathbb{Z}$. Because we know that c is odd according to Lemma 1, we obtain $\frac{Qa+Pb}{c} \in \mathbb{Z}$. However, this contradicts the fact that for any pair n, m in $\mathbb{Z}$, we never have $2n = 2m + 1$. Therefore α cannot belong to both E_h and E_p simultaneously.

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

3 Computing the Lower Bound Hinge Angle from a Pythagorean Angle

In this section, we propose a method for computing a lower bound hinge angle α_1 from a given angle for rotating a given digital image. Note that with minor modifications, this method can also find the upper bound hinge angle α_2, and thus, by applying twice this method, we can obtain two hinge angles that enclose the given angle. The set S of angles γ such that $\alpha_1 \leq \gamma \leq \alpha_2$ is called admissible rotation angles, denoted by ARA. All rotations of the given digital image done by an angle in S give the same result. Nouvel and Rémila presented a method to compute all possible hinge angles for a grid point or a pixel in a digital image [5]. Their method can be used for finding our interesting hinge angle which is the lower bound of the admissible rotation angles. 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. In subsection 3.1, we improve their method by using a tree structure for hinge angles, so that our method brings the complexity $O(\log(n))$. In subsection 3.2, we present a method for finding the lower bound hinge angle for a given digital image, namely, for all pixels in the image.

3.1 Computing the Lower Bound Hinge Angle for a Grid Point

For each grid point $\boldsymbol{p} = (P, Q)$ in $\mathbb{Z}^2$, there are less than $n = \lfloor \sqrt{P^2 + Q^2} + \frac{1}{2} \rfloor$ different hinge angles [5]. This means that we have a sequence of $K_i, i = 0, 1, .., n-1$ in $\mathbb{Z}$, where $0 \leq K_i < n$ for each $\boldsymbol{p}$. Because we can compare any pair of associated hinge angles $\alpha_i(P, Q, K_i)$, we obtain a totally ordered set $\{\alpha_1(P, Q, K_1), \alpha_2(P, Q, K_2), ..., \alpha_{max}(P, Q, K_{max})\}$ in the ascending order such that $\alpha_1 < \alpha_2 < ... < \alpha_{max}$. Given a Pythagorean angle θ, in order to find the lower bound hinge angle α_i such that $\alpha_i < \theta < \alpha_{i+1}$, we use a tree structure. Binary search allows us to find α_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 Figure 2.

```
Function: Find a hinge angle
    Input (Point p(P,Q), Pythagorean angle θ)
    Output (α(P,Q,K))
    var K_max = ⌊√(P²+Q²) − 1⌋;
    var K_min = 0;
    var K = ⌊(K_max+K_min)/2⌋;
    While (K_max − K_min ≠ 1)
        if (α(P,Q,K) > θ)
            K_max = K;
        else
            K_min = K;
        K = ⌊(K_max+K_min)/2⌋;
    end while
    return α(P,Q,K);
```

Fig. 2. Function for finding a hinge angle

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

Proposition 3. *Let α be a hinge angle and θ 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 $[0, \frac{\pi}{2}]$ and $\theta(a, b, c)$ be a Pythagorean angle in $[0, \frac{\pi}{2}]$. From (2) and (4), we obtain

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

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

$$cP(2K + 1) - 2a(P^2 + Q^2) > -2cQ\lambda. \tag{6}$$

Since we know that c, Q, λ 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 squares of (6), so that we only have to check whether the following inequality holds:

$$\left[cP(2K+1) - 2a(P^2+Q^2)\right]^2 < 4c^2Q^2\lambda^2. \tag{7}$$

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 also verify (7) with integer calculation. If it is true, $\theta > \alpha$; otherwise $\alpha > \theta$. Note that because of Proposition 2, it is impossible to obtain $\theta = \alpha$.

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

3.2 Computing the Lower Bound Hinge Angle for a Set of Points

In this subsection, we present an algorithm, based on the previous one, for computing the lower bound hinge angle from a given Pythagorean angle θ for a digital image consisting of m grid points such that $A = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_m\}$. The output is a triplet of integers that represents 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 hinge angle for the first point of A, and store it as the reference. Then, we compute the hinge 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 hinge angle α such that $\alpha < \theta$. The time complexity of this algorithm is $\mathrm{O}(m \log(n))$ because we call m times the function of binary search (Figure 2) whose time complexity is $\mathrm{O}(\log(n))$. Figure 3 illustrates our algorithm. As shown in the following proposition, the comparison between two hinge angles is realized in a constant time, so that it does not change the global complexity.

Proposition 4. *Let α_1, α_2 be two hinge angles. We can check if $\alpha_1 > \alpha_2$ in a constant time and with full integer calculation.*

The proof is similar to that of Proposition 3.

Note that our input is a Pythagorean angle, as the one in [5], in this paper. However, we can replace it by an Euclidean angle because there exists a method in linear time complexity $\mathrm{O}(m)$ to approximate a given Euclidean angle with a Pythagorean angle with a precision of $\frac{1}{10^m}$ [3].

4 Digital Image Rotation by a Hinge Angle

In this section, we present an algorithm for rotating a digital image with a given lower bound hinge angle, which is obtained by the algorithm described in Subsection 3.2. It is already proved in [5] that we can obtain the same result as the DER with respect to the original angle. Note that our input is a hinge angle

```
Function: Find hinge angle for a digital image
     Input (Digital image A, Pythagorean angle θ)
     Output (hinge angle)
     var HA, HA_temps \* hinge angle *\;
     HA = Find hinge angle(first point of A, angle
of rotation);
     for each p ∈ A\{p_1}
         HA_temps = Find hinge angle(p, θ);
         if (HA < HA_temps) HA = HA_temps;
     end for
     Return (HA);
```

Fig. 3. Function for finding the lower bound hinge angle for a digital image

```
Function: Discrete rotation
     Input (a digital image A, a hinge angle α)
     Output (Rotated image A')
     var HA : hinge angle;
     for each p ∈ A
         HA = Find hinge angle(p, α);
         move p to (K, ⌊λ + 1/2,⌋) or (⌊λ + 1/2⌋, K),
depending on the sign of K and put it to A' ;
     end for
     Return (A');
```

Fig. 4. Discrete rotation algorithm by a hinge angle

and the input of the algorithm presented in Figure 2 is a Pythagorean angle. In spite of this difference, we can apply the same algorithm thanks to Proposition 4. The algorithm is presented in Figure 4. It supposes that the center of rotation is the origin. For each point, it calls the function of binary search (Figure 2) 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 $\mathrm{O}(n^2 \log(n))$. The first advantage of our method is that it does not require any float number calculation. The second advantage is that the exact rotation of the digital image is obtained with only an integer triplet. We need neither matrices nor angles for realizing the rotation.

5 Obtaining Admissible Rotation Angles 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: $A = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_l\}$ and $B = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_l\}$ are given where $\boldsymbol{p}_i$ corresponds to $\boldsymbol{q}_i$. Given A and B, we obtain a hinge angle pair α_1, α_2, such that $\alpha_1 \leq \gamma < \alpha_2$ where γ is an admissible rotation angle 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 γ. In this section, we show how to obtain the ARA from these two digital images.

To simplify the notation, we denote by $ARA(p_i, q_i) = (\alpha_{i1}, \alpha_{i2})$ the pair of angles, which gives the lower and the upper bounds of possible angles of the rotation for the pair of points (p_i, q_i). Note that the angles α_{i1}, α_{i2} are hinge angles. $ARA(A_{n+1}, B_{n+1})$ denotes the two most restrictive angles for all point i such as $i \leq n+1$. We formally define it by $ARA(A_{n+1}, B_{n+1}) = ARA(A_n, B_n) \bigcap ARA(p_{n+1}, q_{n+1})$.

5.1 Setting Rotation Centers

For any rotation, we need to set a rotation center. In this paper, we choose one of the grid points in a digital image for the rotation center. Assuming centers are $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$ for A and B respectively, we define two functions $\mathscr{T}_A$ and $\mathscr{T}_B$ such that

$$\mathscr{T}_A(\boldsymbol{p_i}) = \boldsymbol{p_i} - \boldsymbol{p_1},$$
$$\mathscr{T}_B(\boldsymbol{q_i}) = \boldsymbol{q_i} - \boldsymbol{q_1},$$

for all $\boldsymbol{p}_i \in A, \boldsymbol{q}_i \in B$, so that we can consider the rotation centers to be the origin after the translations. Hereafter, we will use new sets of points $A' = \{\mathscr{T}_A(\boldsymbol{p}_1), \mathscr{T}_A(\boldsymbol{p}_2), ..., \mathscr{T}_A(\boldsymbol{p}_l)\}$ and $B' = \{\mathscr{T}_B(\boldsymbol{q}_1), \mathscr{T}_B(\boldsymbol{q}_2), ..., \mathscr{T}_B(\boldsymbol{q}_l)\}$ instead of A and B. However, for simplicity, we will denote them by $A = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_l\}$ and $B = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_l\}$.

5.2 Computing Hinge Angles from Two Corresponding Point Pairs

In this subsection, we consider the case with $A = \{\boldsymbol{p_1}, \boldsymbol{p_2}\}$ and $B = \{\boldsymbol{q_1}, \boldsymbol{q_2}\}$ where $\boldsymbol{p}_i = (P_i, Q_i)$ and $\boldsymbol{q}_i = (R_i, S_i)$. Let us first define a circle $\mathscr{C}(\boldsymbol{p}_2)$ with center $\boldsymbol{p}_1$ that goes through $\boldsymbol{p}_2$. Thus the radius of $\mathscr{C}(\boldsymbol{p}_2)$ is $r = d(\boldsymbol{p}_1, \boldsymbol{p}_2)$ where $d(\boldsymbol{p}_1, \boldsymbol{p}_2)$ is the Euclidean distance between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. Let us consider the half grid around $\boldsymbol{q}_2$ such that

$$\begin{aligned}\mathscr{H}(\boldsymbol{q}_2) = \{(x, y) \in \mathscr{H} : S_2 - \frac{1}{2} \leq y \leq S_2 + \frac{1}{2} \text{ when } x = R_2 + \tfrac{1}{2},\\ R_2 - \frac{1}{2} \leq x \leq R_2 + \frac{1}{2} \text{ when } y = S_2 + \tfrac{1}{2}\}.\end{aligned}$$

Setting $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$ to be the rotation centers, for finding a hinge angle pair, we need to detect intersections between $\mathscr{C}(\boldsymbol{p}_2)$ and $\mathscr{H}(\boldsymbol{q}_2)$. In other words, we study corners of $\mathscr{H}(\boldsymbol{q}_2)$ in the interior of $\mathscr{C}(\boldsymbol{p}_2)$. Setting four corners of $\mathscr{H}(\boldsymbol{q}_2)$ such that $C_1(\boldsymbol{q}_2) = (R_2 - \frac{1}{2}, S_2 - \frac{1}{2}), C_2(\boldsymbol{q}_2) = (R_2 - \frac{1}{2}, S_2 + \frac{1}{2}), C_3(\boldsymbol{q}_2) = (R_2 + \frac{1}{2}, S_2 + \frac{1}{2}), C_4(\boldsymbol{q}_2) = (R_2 + \frac{1}{2}, S_2 - \frac{1}{2})$ as shown in Figure 5, we define a binary function $\mathscr{F}$ such as

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

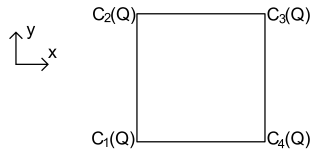

Fig. 5. The half grid $\mathscr{H}(\boldsymbol{q})$, namely a pixel around $\boldsymbol{q}$ and its four corners

In order to obtain $\mathscr{F}(C_i(\boldsymbol{q_2}))$ with integer calculation, we compare each of $4((R_2 \pm \frac{1}{2})^2 + (S_2 \pm \frac{1}{2})^2)$ with $4r^2$. Note that we may assume that the intersection between $\mathscr{C}(\boldsymbol{p}_2)$ and $\mathscr{H}(\boldsymbol{q}_2)$ is not null. If no intersection between $\mathscr{C}(\boldsymbol{p}_2)$ and $\mathscr{H}(\boldsymbol{q}_2)$ exists, then $\boldsymbol{p}_2$ and $\boldsymbol{q}_2$ are not corresponding.

Proposition 5. *If two points $\boldsymbol{p}_2$ and $\boldsymbol{q}_2$ are corresponding, a circle $\mathscr{C}(\boldsymbol{p}_2)$ and a pixel boundary $\mathscr{H}(\boldsymbol{q}_2)$ always have two distinct intersections.*

The mathematically rigorous proof is omitted in this paper because of the page limitation. The proof is accomplished by distinguishing the following two cases; in the other cases, we have always two intersections.

The first case is that $\mathscr{C}(\boldsymbol{p_2})$ goes through one of the four corners of $\mathscr{H}(\boldsymbol{q}_2)$. Because any angle between $\boldsymbol{p}_2$ and $C_i(\boldsymbol{q}_2)$ at the origin is a Pythagorean angle it cannot be a hinge angle from Proposition 2. Thus, this case never happens.

The second case is that $\mathscr{C}(\boldsymbol{p}_2)$ and $\mathscr{H}(\boldsymbol{q}_2)$ have the unique intersection on one of edges of $\mathscr{H}(\boldsymbol{q}_2)$. This case may happen only when one coordinate of $\boldsymbol{q}_2$ is zero. A circle centered at the origin can cross twice a half grid parallel to one of the axes if and only if the circle arc between those intersections cuts another axis. Therefore, if the intersection is single, it should be on an axis, so that λ should be null. However, it is impossible by the definition of hinge angles.

From Proposition 5, we always have two intersections between $\mathscr{C}(\boldsymbol{p}_2)$ and $\mathscr{H}(\boldsymbol{q}_2)$, and see that there are four cases corresponding to different possibilities to have 0,1,2 or 3 corners in the interior of $\mathscr{C}(\boldsymbol{p}_2)$, as illustrated in Figure 6.

– Case A: $\mathscr{C}(\boldsymbol{p_2})$ includes no corner. Thus we have $\mathscr{F}(C_i(\boldsymbol{q_2})) = 0$ for all $i = 1, 2, 3, 4$, similarly to the above second impossible case. This case can only happen when $R_2 = 0$ or $S_2 = 0$. Supposing that R_2 and S_2 are not null, we assume that they are positive. In the first quadrant, the y-coordinate (respectively x-coordinate) of points in $\mathscr{C}(\boldsymbol{p_2})$ is strictly decreasing with respect to x (respectively y). Thus it cannot intersect twice a line parallel to the x-axis (respectively y-axis). Therefore, if $S_2 = 0$, $ARA(p_2, q_2) = (\alpha_{21}(P_2, Q_2, R_2 - 1), \alpha_{22}(P_2, Q_2, R_2 - 1))$. In this case the two hinge angles are symmetrical with respect to the y-axis.
– Case B: $\mathscr{C}(\boldsymbol{p_2})$ includes only one corner. For example, if $C_1(\boldsymbol{q_2})$ is in the circle, we obtain $ARA(p_2, q_2) = (\alpha_{21}(P_2, Q_2, R_2 - 1), \alpha_{22}(P_2, Q_2, -S_2 + 1))$.
– Case C: $\mathscr{C}(\boldsymbol{p_2})$ includes two corners that should have one common coordinate. For example, if $C_1(\boldsymbol{q_2})$ and $C_2(\boldsymbol{q_2})$ are in the circle, we obtain $ARA(p_2, q_2) = (\alpha_{21}(P_2, Q_2, -S_2 + 1), \alpha_{22}(P_2, Q_2, -S_2))$.

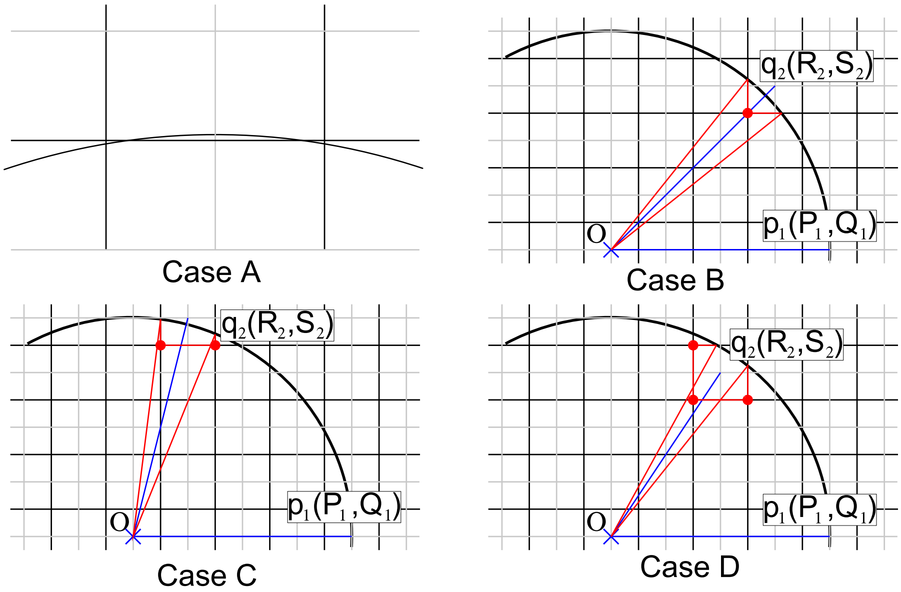

Fig. 6. Illustration of cases A,B,C and D

– Case D: $\mathscr{C}(\boldsymbol{p_2})$ includes three corners. For example, if $C_1(\boldsymbol{q_2}), C_2(\boldsymbol{q_2})$ and $C_4(\boldsymbol{q_2})$ are in $\mathscr{C}(\boldsymbol{q}_2)$, then we obtain $ARA(p_2, q_2) = (\alpha_{21}(P_2, Q_2, R_2), \alpha_{22}$ $(P_2, Q_2, -S_2))$.

The main function of our algorithm for finding the two hinge angles consist of three steps. The first step sets the rotation center at $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$, as described in the previous subsection. The second step computes which corners are in the interior of $\mathscr{C}(\boldsymbol{q}_2)$ and then stocks the result as an index. The index is calculated by $index = \sum_i 2^i \times \mathscr{F}(C_i(\boldsymbol{q_2}))$. Therefore we can easily identify which corners are in the interior of $\mathscr{C}(\boldsymbol{p}_2)$ from the index. The third step calls a function that returns hinge angles corresponding to the index. There exist fourteen possible values for the index. Note that geometrically the index can be neither 5 nor 10. The index value 15 implies an error such that all corners are in the interior of $\mathscr{C}(\boldsymbol{q}_2)$. Since the index value 0 corresponds to the case A, we should verify whether $\mathscr{H}(\boldsymbol{q}_2)$ really intersects with $\mathscr{C}(\boldsymbol{q}_2)$. Note that for all other index values, we can make a pair (d, e) such that $d + e = 15$. The two indices of such a pair design the same pair of hinge angles. Each step of this algorithm has the constant time complexity. Thus the global complexity of this algorithm is also O(1).

5.3 Incremental Hinge Angle Computing

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

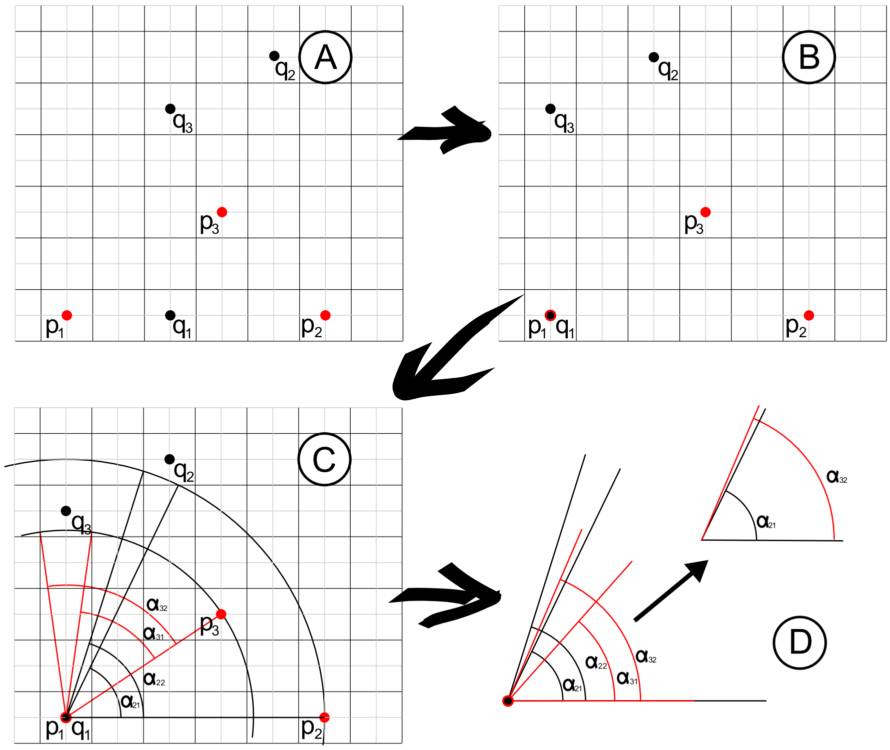

Fig. 7. Running of the incremental algorithm

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 (p_2, q_2). Note that $ARA(p_1, q_1)$ cannot be computed because $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$ are the centers of the rotation. The second part computes $ARA(A_{i+1}, B_{i+1})$ for $i = 2, \ldots, n-1$. The time complexity of this algorithm is O(l). As explained in the previous subsection, the function 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 full computation of this algorithm for l points takes the time complexity of $l \times (O(1) + O(1)) = O(l)$.

5.4 Example of the Running of the Algorithm

Figure 7 gives an example of the incremental algorithm for two sets of three points. Given input data of the algorithm as shown in Figure 7 (A), we first obtain the result of the translation described in subsection 5.1 as illustrated in (B). We then compare, for each pair of points $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ with $i \geq 2$, the distance of $\boldsymbol{p}_i$ from the origin with the distance of each corner from $\mathscr{H}(\boldsymbol{q}_i)$ to deduce the corresponding hinge angle as explained in subsection 5.2. Finally, we obtain (D) which shows the intersection of all $ARA(p_i, q_i)$ obtained in (C).

6 Conclusion

In this paper, we have shown how to obtain a hinge angle which is the lower bound approximation to a given Euclidean angle. We have also shown that we can efficiently obtain a rotated digital image from the integer triplet identically to that from the Pythagorean angle. We then have presented a method for obtaining the upper and lower bounds of the ARA from a pair of digital images.

The future work will extend our proposed method into two directions. The first direction is to extend this algorithm to the 3D case. The second direction is to create a 2D matching algorithm based on hinge angles. Current methods for matching can be improved by restricting the searching area. The admissible rotation angles obtained by our method will be useful for the restriction of the searching area.

Acknowledgment

This work was supported by grants from Région Ile-de-France.

References

1. Andres, E.: Cercles Discrets et Rotations Discrétes. Thesis, Université Louis Pasteur Strasbourg (1992)
2. Andres, E.: The Quasi-Shear Rotation. In: Miguet, S., Ubéda, S., Montanvert, A. (eds.) DGCI 1996. LNCS, vol. 1176, pp. 307–314. Springer, Heidelberg (1996)
3. Anglin, W.S.: Using Pythagorean triangles to approximate angles. American Mathematical Monthly 95, 540–541 (1988)
4. Hansen, M., Anandan, P., Dana, K., van der Wal, G., Burt, P.: Real-time scene stabilization and mosaic construction. In: 2nd IEEE workshop on Applications of Computer Vision, pp. 54–64 (1994)
5. Nouvel, B., Rémila, E.: Incremental and transitive discretized rotations. In: Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K. (eds.) IWCIA 2006. LNCS, vol. 4040, pp. 199–213. Springer, Heidelberg (2006)
6. Volder, J.E.: The CORDIC Trigonometric Computing Technique. IRE Transactions on Electronic Computers EC-8, 330–334 (1959)