November 1994

Pattern Recognition Letters

ELSEVIER

Pattern Recognition Letters 15 (1994) 1081–1088

Detecting number of folds by a simple mathematical property ☆

Ja-Chen Lin *, Wen-Hsiang Tsai, Jun-Ann Chen

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, ROC

Received 20 April 1993; revised 2 May 1994

Abstract

Based on a simple mathematical property, a simple method is proposed for detecting the number of folds n contained in a given rotationally symmetric shape. Skills to overcome the difficulty caused by sampling and rounding errors are also discussed. Experimental results are included to show the feasibility of this new method. As for shape orientation, it is just a by-product of the proposed method because no extra computation load is needed to get the orientation.

Keywords: Symmetry; Number of folds; Factor; Shape orientation; Sampling error; Rounding error; Computation load

1. Introduction

Symmetry is an important feature for shape perception which is useful for a lot of computer vision applications. Some benefits are associated with symmetry. For example, the storage space representing a given symmetric shape can be reduced because it only needs to store one copy of the repeating portions (Parui and Majumder, 1983). Another example is that if certain parts of the symmetric shape are missing, then symmetry provides a clue to recover the shape. At least two kinds of symmetry can be identified, namely, mirror symmetry (Atallah, 1985), and rotational symmetry (Chou et al., 1991).

To make use of the shape symmetry property, it is usually necessary (Chou et al., 1991) to compute the number of symmetry axes (or the number of folds) of a given shape. Several methods (Atallah, 1985; Parui and Majumder, 1983) have been proposed to find the number and orientation of the symmetry axes contained in a given shape which is mirror symmetric, i.e., symmetric about certain lines called symmetry axes. On the other hand, the number of folds n contained in a given rotationally symmetric shape can be detected by a method described in (Leou and Tsai, 1987) where n is counted as the frequency that the boundary of the given rotationally symmetric shape runs across a special circle related to the given shape.

In this paper we use a simple mathematical property shared by all rotationally symmetric shapes to derive a new method for detecting the number of folds n contained in a given rotationally symmetric shape. The approach is simple, and is based on the observation that if the origin of the coordinate system is taken to be the centroid of the given rotationally symmetric shape composed of K sampled points $\{(x_k, y_k)\}_{k=1}^{K}$, then with the convention that $\mathrm{i}=\sqrt{-1}$ and $r_k e^{\mathrm{i}\theta_k}=x_k+\mathrm{i}y_k$, it can be derived that

$$\sum_{k=1}^{K} r_k^{1-l}(x_k+\mathrm{i}y_k)^l=0 \qquad (1)$$

☆ This study was supported, in part, by the National Science Council, Republic of China, under contract number: NSC80-0408-E-009-26.

* To whom all correspondence should be sent.

SSDI 0167-8655(94)00039-6

whenever l/n is not an integer. Once the value of n is found by our method, an important feature of the shape, namely, the orientation of the shape, also comes out immediately without further computation. The shape orientation can thus be viewed as a by-product of our method.

The remainder of this paper is organized as follows. Equality (1) will be proved in the Appendix by the use of the angularly periodic property of rotationally symmetric shapes. Based on this equality, a new method for detecting n using a rotation-matching technique is described in Section 2. In Section 3, we discuss the skill to overcome the difficulty caused by sampling and rounding errors. In Section 4, some experimental results are given to show that the proposed method really works well. The computation load is analyzed in Section 5. In Section 6 we show in detail why the orientations of rotationally symmetric shapes are just a by-product of our method. We also explained in that section why this by-product is important. A summary is given in Section 7.

2. Detecting numbers of folds

A shape S is called an n-fold rotationally symmetric shape (abbreviated as an n-RSS henceforth) if it becomes identical to itself after being rotated around its centroid through any multiple of the angle $2\pi/n$ for some $n>1$. For simplicity, we assume that no other larger integer n has this property. The purpose of this study is to find this value n when the given shape S has been known to be a rotationally symmetric shape. Throughout this paper, we take the origin of the coordinate system to be the centroid of S. Let shape S, when sampled in the X–Y coordinate system, be described as

$$S=\{(x_k, y_k)\}_{k=1}^{K} \tag{2}$$

where K is the number of sampled points in S, i.e., K is the number of the grid points contained in S when S is covered by a grid (also called mesh) used for digitalization. For any natural number l, define

$$x^{(l)}+\mathrm{i}y^{(l)}=\sum_{k=1}^{K} r_k e^{\mathrm{i}l\theta_k}=\sum_{k=1}^{K} r_k\left(\frac{x_k}{r_k}+\mathrm{i}\frac{y_k}{r_k}\right)^{l}$$

$$=\sum_{k=1}^{K} r_k^{1-l}(x_k+\mathrm{i}y_k)^{l} \tag{3}$$

where the polar length r_k and polar angle θ_k are defined by the convention

$$r_k e^{\mathrm{i}\theta_k}=x_k+\mathrm{i}y_k, \quad k=1, 2, ..., K. \tag{4}$$

Then we have the following theorem (see Appendix for the proof).

Theorem 1. *If l/n is not an integer, then*

$$x^{(l)}+\mathrm{i}y^{(l)}=0.$$

Notice that Theorem 1 above does not imply that $x^{(n)}+\mathrm{i}y^{(n)}\neq 0$ although $n/n=1$ is an integer. In fact, as will be seen in Section 4, $x^{(n)}+\mathrm{i}y^{(n)}\neq 0$ for almost every n-RSS and $x^{(n)}+\mathrm{i}y^{(n)}=0$ for some strange n-RSS. It is therefore not true to say that the smallest positive value l making $x^{(l)}+\mathrm{i}y^{(l)}$ nonzero is the value n we want. However, based on this theorem and the definition of n-RSS, the value n can still be found easily in the following way. Just evaluate $x^{(l)}+\mathrm{i}y^{(l)}$ for $l=2$, then for $l=3$, and so on, until an $l=l_1$ making $x^{(l)}+\mathrm{i}y^{(l)}\neq 0$ is found. Theorem 1 above implies that n must be a factor of this l_1. Otherwise, l_1/n is not an integer, and hence $x^{(l_1)}+\mathrm{i}y^{(l_1)}=0$, which contradicts the definition of l_1. Let $\{p_i\}_{i=1}^{I}$ be the set of all positive factors of l_1 with

$$l_1=p_1>p_2>...>p_I=1, \tag{5}$$

then $n\in\{p_i\}_{i=1}^{I-1}$. Consequently, the definition of n-RSS given at the beginning of this section implies that n is the greatest p_i in $\{p_i\}_{i=1}^{I-1}$ satisfying the requirement that the shape S is identical to itself if the shape is rotated through an angle of $2\pi/p_i$. In summary, the following algorithm can be used to find the value n.

ALGORITHM 1. Finding the value n for a given n-RSS S.

Step 1. Translate the coordinate system so that the origin O becomes the centroid of the given shape S.

Step 2. Evaluate the value of $x^{(l)}+\mathrm{i}y^{(l)}$ defined in (3) for $l=2$, then for $l=3$, and so on, until an $l=l_1$ making $x^{(l)}+\mathrm{i}y^{(l)}\neq 0$ is found.

Step 3. Arrange all of the positive factors $\{p_i\}_{i=1}^{I}$ of l_1 in a descending order, namely, $l_1=p_1>p_2>...>p_I=1$.

Step 4. Set $i=1$.

Step 5. If the shape S is identical to the shape S_{ROT} obtained by rotating S through an angle of $2\pi/p_i$, then go to Step 7. If these two shapes are not identical, then proceed to Step 6.

Step 6. Increase the value of i by 1, and then go to Step 5.

Step 7. Output the value p_i because it is the value n we want.

3. Skills to handle sampling and rounding errors

Due to sampling and rounding errors, special skills should be used to detect the value of l_1 because it is not easy to judge whether a complex number $x^{(l)}+\mathrm{i}y^{(l)}$ generated by a computer is zero or not (see Step 2 of Algorithm 1). Similarly, some special rules should also be used to judge whether a shape S and a rotated version S_{ROT} of it are "identical" or not (see Step 5 of Algorithm 1).

For Step 5 of the algorithm, i.e., for the rotation-matching step, we adopt the rule that shape S and its rotated version S_{ROT} are said to be "identical" if and only if more than ninety-nine percent of the sampled points of S are also some sampled points of S_{ROT}. Of course, if the grid used for sampling is quite coarse, then the sample error will become very severe, and the percentage mentioned here should be lowered accordingly.

As for the detection of l_1, since l_1 is a multiple of n, and $n>1$ (because by definition a rotationally symmetric shape can not just have one fold), we can see that $l_1\pm 1$ is not a multiple of n. Theorem 1 thus implies that the values of $x^{(l_1\pm 1)}+\mathrm{i}y^{(l_1\pm 1)}$ are theoretically zero. On the other hand, the definition of l_1 implies that $x^{(l_1)}+\mathrm{i}y^{(l_1)}$ is theoretically nonzero. Therefore, it seems reasonable to expect that the computed value of $|x^{(l_1)}+\mathrm{i}y^{(l_1)}|$ will be much larger than those of $|x^{(l_1\pm 1)}+\mathrm{i}y^{(l_1\pm 1)}|$. Hence, in practice when sampling and rounding errors both exist, instead of defining l_1 as the smallest positive integer making $x^{(l_1)}+\mathrm{i}y^{(l_1)}\neq 0$, we define l_1 as the smallest positive integer making

$$|x^{(l_1)}+\mathrm{i}y^{(l_1)}| \gg \max\{|x^{(l_1-1)}+\mathrm{i}y^{(l_1-1)}|, |x^{(l_1+1)}+\mathrm{i}y^{(l_1+1)}|\}, \tag{6}$$

or equivalently, making

$$|x^{(l_1)}+\mathrm{i}y^{(l_1)}|/K \gg \max\{|x^{(l_1-1)}+\mathrm{i}y^{(l_1-1)}|/K, |x^{(l_1+1)}+\mathrm{i}y^{(l_1+1)}|/K\} \tag{7}$$

where the K appearing above is the number of sampled points contained in the shape. Dividing both sides of (6) by K to get (7) means taking the average among sampled points. We prefer (7) to (6) because if the shape is sampled, say, in a 256×256 grid with the readings being 0, 1, ..., 255 on both of the horizontal and vertical axes, then the computed values of $|x^{(l_1\pm 1)}+\mathrm{i}y^{(l_1\pm 1)}|$ could be as large as several thousands although they are theoretically zero; and dividing them by K makes them look more natural, i.e., much closer to zero.

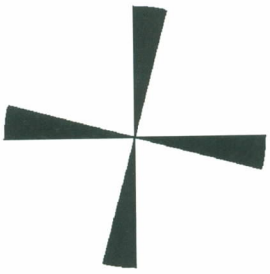
4. Experimental results of detecting numbers of folds

We show in this section some experimental results of detecting numbers of folds using the algorithm provided in Section 2. The effect caused by sampling and rounding errors has been taking into account, i.e., the value of l_1 is taken to be the smallest positive integer making (7) hold (instead of making $x^{(l_1)}+\mathrm{i}y^{(l_1)}\neq 0$); and the two sets S and S_{ROT} are said to be "identical" if they differ from each other only in a very small portion of the total area (see the second paragraph of Section 3 for explanation).

A 4-fold rotationally symmetric shape is sketched in Fig. 1(a). The number of sampled points contained in this shape is $K=12181$ when the shape is sampled in a 256×256 grid. The values of $|x^{(l)}+\mathrm{i}y^{(l)}|/K$ corresponding to $l=1, 2, 3, \ldots, 8$ are listed in Fig. 1(b). Note that we did not compute $|x^{(1)}+\mathrm{i}y^{(1)}|/K$ because

$$\begin{aligned}|x^{(1)}+\mathrm{i}y^{(1)}|/K &= |\textstyle\sum_{k=1}^{K}(x_k+\mathrm{i}y_k)|/K\\ &= |(\textstyle\sum_{k=1}^{K}x_k)+\mathrm{i}(\sum_{k=1}^{K}y_k)|/K\\ &= |0+\mathrm{i}0| = 0\end{aligned}$$

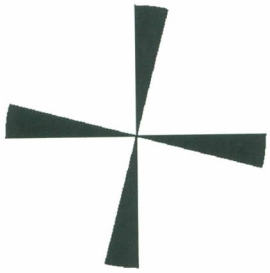
is an immediate consequence of the requirement that we always translate the coordinate system so that the origin coincides with the centroid of the given shape. $|x^{(1)}+\mathrm{i}y^{(1)}|/K$ is listed here just for reference use. From that list of $\{|x^{(l)}+\mathrm{i}y^{(l)}|/K\}_{l=1}^{8}$, we see that $|x^{(l)}+\mathrm{i}y^{(l)}|/K$ has a sharp increase as l goes from 3



(a)

12181 sampled points	
l	$\lvert x^{(l)}+iy^{(l)}\rvert/K$
1	zero
2	0.00000677
3	0.06872168
4	94.06779355
5	0.07474513
6	0.00010043
7	0.11169840
8	80.90273462

(b)

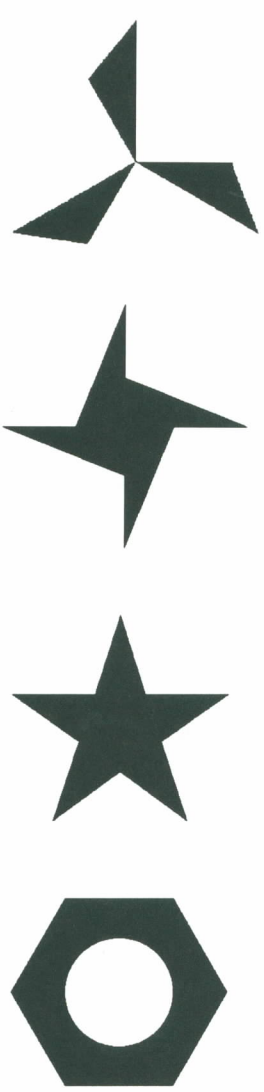


(c)

Fig. 1. A 4-fold rotationally symmetric shape and its corresponding values of $|x^{(l)}+iy^{(l)}|/K$. (a) The shape. (b) The values of $\{|x^{(l)}+iy^{(l)}|/K\}_{l=1}^{8}$ for this shape shows that $l_1=4$. (c) The shape remains unchanged if it is rotated through an angle of $2\pi/l_1=2\pi/4$. Hence, the shape is judged to be 4-fold.

to 4, then a sharp decrease as l goes from 4 to 5. Hence, $l_1=4$ meets the requirement stated in (7). The set of the positive factors of 4 are $\{4, 2, 1\}$, so we have $p_1=4>p_2=2>p_3=1$. According to the algorithm, we should use $2\pi/p_1=2\pi/4$ as the rotation angle in the first try of rotation-matching test. Immediately, we found that there is no need to try the remaining p_i because the shape is found to be identical to itself when it is rotated through an angle of $2\pi/4$. Hence, $n=p_1=4$ is the detected value used for output.

The shapes sketched in Fig. 2 have 3, 4, 5, and 6 folds, respectively. Analogously, the computation results listed there also show that $l_1=3$, 4, 5, and 6, respectively, because they are the smallest positive integers making (7) hold. After a rotation-matching test with the rotation angle being $2\pi/l_1$ for each of the cases, we found that the shape is unchanged after the rotation. Hence, the detected value of n is l_1, i.e., 3, 4, 5, and 6, respectively. Again, the detected value of n for each case is exact!



11743 sampled points	
l	$\lvert x^{(l)}+iy^{(l)}\rvert/K$
1	zero
2	0.18979682
3	73.41740861
4	0.23981518
5	0.26024903
6	52.24475213
7	0.39950036
8	0.27397838

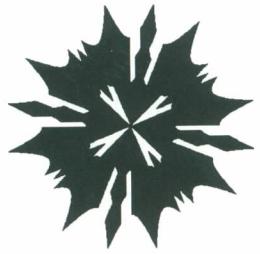
18481 sampled points	
l	$\lvert x^{(l)}+iy^{(l)}\rvert/K$
1	zero
2	1.91×10^{-14}
3	1.48×10^{-15}
4	31.38926749
5	6.72×10^{-16}
6	2.29×10^{-14}
7	3.87×10^{-15}
8	21.71325449

25410 sampled points	
l	$\lvert x^{(l)}+iy^{(l)}\rvert/K$
1	zero
2	0.19911809
3	0.14686640
4	0.27219356
5	36.69398282
6	0.33027821
7	0.09821857
8	0.12342570

40937 sampled points	
l	$\lvert x^{(l)}+iy^{(l)}\rvert/K$
1	zero
2	0.62949276
3	0.15918286
4	0.10564380
5	0.09614510
6	11.50829694
7	0.05603174
8	0.12041471

Fig. 2. Some rotationally symmetric shapes and their corresponding values of $\{|x^{(l)}+iy^{(l)}|/K\}_{l=1}^{8}$.

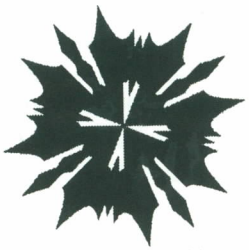
Our experience is that, although $n=l_1$ holds for almost every rotationally symmetric shape, some strange shapes (usually artificially-created) do have $l_1>n$. An example is given in Fig. 3. The 4-fold shape in Fig. 3(a), which is artificially-created, is introduced in (Lin et al., 1992). It was designed in such a way that it has the property $x^{(4)}+iy^{(4)}=0+i0=0$ (see (Lin et al., 1992) for details). Therefore, the value of $|x^{(4)}+iy^{(4)}|/K$ is also (theoretically) zero.



(a)

48315 sampled points	
l	$\lvert x^{(l)}+iy^{(l)}\rvert/K$
1	zero
2	0.00647269
3	0.00591212
4	0.00982650
5	0.00364557
6	0.00268533
7	0.00304035
8	13.19616575
9	0.00469254
10	0.00267312
11	0.00556702
12	7.11539952
13	0.00448080

(b)



(c)

(d)

Fig. 3. An n-RSS of which the value of l_1 is not n. (a) The shape. (b) The values of $\{|x^{(l)}+iy^{(l)}|/K\}_{l=1}^{8}$ for this shape show that l_1 is 8 instead of 4. (c) The shape is not identical to itself if it is rotated through an angle of $2\pi/l_1=2\pi/8$. Hence, the shape can not be 8-fold. (d) The shape is identical to itself if it is rotated through an angle of $2\pi/p_2=2\pi/4$ (where $p_2=4$ is the second largest factor of l_1). Hence, the shape is judged to be 4-fold by the proposed algorithm.

Hence, the computation result shown in Fig. 3(b) indicates that l_1 is 8 instead of 4. Since the set of all positive factors of 8 is $\{8,4,2,1\}$, we set $p_1=8>p_2=4>p_3=2>p_4=1$. Then we used $2\pi/p_1=2\pi/8$ as the rotation angle in the first try of the rotation-matching test and found that the shape S_{ROT} (sketched in Fig. 3(c)) is different from the original shape S (sketched in Fig. 3(a)). Therefore, we used $2\pi/p_2=2\pi/4$ as the rotation angle in the second try and found that the shape S_{ROT} (sketched in Fig. 3(d)) is identical to the original shape S. Hence, $n=p_2=4$ is the detected value for output, which is correct.

5. Computation load and an alternative version

The computation load of Algorithm 1 is of order Kn only (as usual, K is the number of sampled points and n is the number of folds). To see this, just note that most of the computations are done in Steps 2 and 5 of Algorithm 1. We first prove that Step 2 is of order Kn only. The evaluation of the values $\{x^{(l)}+iy^{(l)}\}_{l=2}^{1+l_1}$ has computation load of order Kl_1 because for each sampled point (x_k, y_k) the evaluation of the $\{|x_k+iy_k|((x_k+iy_k)/|x_k+iy_k|)^l\}_{l=2}^{1+l_1}$ has computation load of order l_1. When (x_k, y_k) ranges over all K sampled points to take the $\sum_{k=1}^{K}$ operator used in (3), we see that Step 2 of Algorithm 1 has computation load of order Kl_1, and hence, of order Kn because $l_1=n$ for almost every rotationally symmetric shape, and $l_1=2n$, $3n$, etc., for some very strange shapes (but the latter case seldom occurs). As for Step 5, there are two parts, namely, rotation and matching, respectively. The job of rotating a K-point shape (through an angle of β) is of order K because we only have to multiply each point x_k+iy_k by a complex number $\cos\beta+i\sin\beta$ to generate the corresponding rotated point in the complex plane. The job of comparing two binary-valued 256-by-256 images, or equivalently, the job of comparing two binary-valued 256-by-256 matrices representing S and S_{ROT}, each with K entries being ones and $256\times256-K$ entries being zeros, is of order 256×256. On the other hand, K is proportional to the number of the grid points, for example, if a 512×512 grid is used instead of 256×256, then the value of K will be approximately four times larger. Hence, we may also say that the job of comparing the two binary-valued matrices is of order K. Therefore, Steps 2 and 5 together has work load of order Kn.

A SUN workstation, whose computation speed is 3 megaflops, was used to compute the value of n, and the total time used for each shape is six seconds for the shape sketched in Fig. 3 and no more than four seconds for each of the shapes sketched in Figs. 1 and 2. The total time used for each shape includes the time for doing several jobs such as reading in a 256-by-256 matrix representing the given shape S; finding the value of l_1; generating a 256-by-256 matrix representing S_{ROT}; comparing the contents of the two matrices representing S and S_{ROT}; etc.

At the end of this section, we provide an alterna-

tive version of Algorithm 1 so that there is no rotation-matching step in the alternative version. More specifically, note first that Step 5 for Algorithm 1, i.e., the rotation-matching procedure to check whether n is l_1 or not, may be omitted in practical applications because it is very unusual to have a shape with $l_1 \neq n$ (although the shape sketched in Fig. 3 is an exception, notice that it is an artificially created shape which was designed to have $l_1 \neq n$ on purpose). The omission of the rotation-matching step of course reduces the computation time significantly. When we are not sure whether this omission is safe (for example, when artificially created strange shapes may possibly be among the shapes to be processed), if it is still desired not to use the rotation-matching operation, one can compute $\{x^{(l)}+iy^{(l)}\}$ for more l to generate not only l_1 but also l_2, l_3, etc. which satisfy $x^{(l_j)}+iy^{(l_j)} \neq 0$ for all l_j. Theorem 1 then implies that n is a common factor of these l_j. When sufficiently many l_j are generated, our experience is that it is usually safe to expect that the greatest common divisor of $\{l_j\}$ is the expected n. For example, for the second shape sketched in Fig. 2, because $l_1=4$, $l_2=8$, $l_3=12$, etc., we may then expect that the value of n is 4 because 4 is the greatest common divisor of $\{4,8,12,...\}$. As for the shape sketched in Fig. 3, because $l_1=8$, $l_2=12$, $l_3=16$, etc., we may then expect that the value of n is 4 because 4 is the greatest common divisor of $\{8,12,16,...\}$.

6. Shape orientation — a by-product of the method

A major advantage of using Algorithm 1 to detect the number of folds is that an important by-product, namely, the orientation of the given rationally symmetric shape, can be extracted easily from the computation results of Algorithm 1. In fact, the orientation of the shape comes out immediately without further computation because the direction can be defined as the half line $\vec{OA}$ starting from the centroid $O=(0, 0)$ and has directional angle Θ/n where Θ is the polar angle of the complex number $x^{(n)}+iy^{(n)}$ if this complex number is nonzero. For example, the orientations of the five shapes sketched in Figs. 1(a) and 2 are thus defined to be the five half-lines ejected from the corresponding centroids and with directional angles $(1/n)$ angle $(x^{(n)}+iy^{(n)})$ for $n=$ 4, 3, 4, 5, 6, respectively. Two of these five half-lines are sketched in Fig. 4(a). The reason why we may use the half-line $\vec{OA}$ to define the orientation of the given n-RSS is due to the fact that $\vec{OA}$ coincides with the half-line $\vec{OB}$ starting from the centroid O and passing through a shape-specific point B, called the fold-invariant centroid (F.I.C.), of the shape (see Fig. 4(b) for illustrations). In (Lin et al., 1992), with the assumption that the value of n is known, Lin et al. defined the F.I.C. first and then defined the orientation of the given n-RSS to be the half-line $\vec{OB}$. It is proved in (Lin et al., 1992) that defining the orientation of the given n-RSS to be $\vec{OB}$ is suitable because the orientation so defined is independent of the translation, rotation, and scaling of the coordinate system used. Therefore, it is also suitable to define the orientation using $\vec{OA}$ if we can prove that $\vec{OA}=\vec{OB}$, or equivalently, the F.I.C. has directional angle Θ/n. But this fact can be derived easily from the definition of F.I.C. (see the algorithm given in (Lin et al., 1992) for the reason).

We emphasize this by-product of Algorithm 1 because of the reasons listed below.

(i) Shapes orientations are very useful in com-

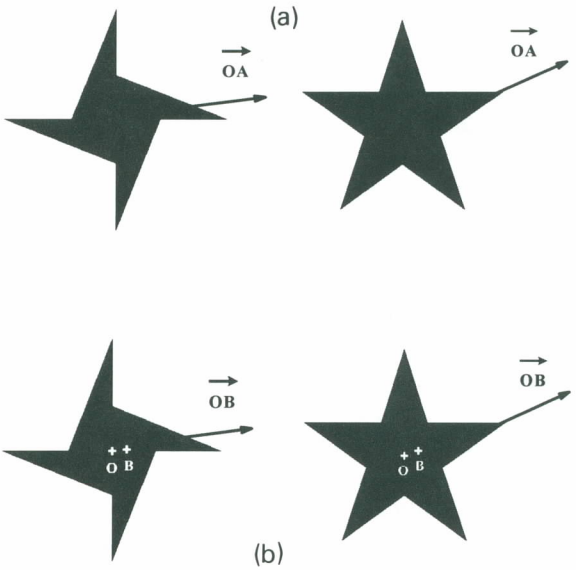


Fig. 4. $\vec{OA}$ and $\vec{OB}$ give the same shape orientation. (a) The shape orientation defined by $\vec{OA}$ starting from the shape centroid O and having directional angle $(1/n)$ angle $(x^{(n)}+iy^{(n)})$. (b) The shape orientation defined by $\vec{OB}$. Here, the point O is the shape centroid and the point B is the F.I.C. defined in (Lin et al., 1992).

puter vision applications related to robotics, geometric modeling, factory automations, etc.

(ii) Rotationally symmetric shapes belong to a special kind of shape in the sense that defining their orientations usually requires special tools because traditional methods such as using te principal axes (Rosenfeld and Kak, 1983) or the half-line starting from the shape centroid and passing through the radius weighted mean point, which is a special point introduced in (Mitiche and Aggarwal, 1983), both fail. The by-product of our method, i.e., the orientation detected by our method, provides a feasible way to overcome this difficulty.

(iii) Although the method used in (Leou and Tsai, 1987) can also be used to detect the number of folds, it requires an extra non-neglectable computation load to execute one of the orientation detection methods, such as those introduced in (Chou et al., 1991) or (Lin et al., 1992), if shapes orientations are also of interest. Our method is a "two-in-one" method in the sense that the computation time needed for detecting both the number of folds and the shape orientation is almost identical to that of detecting numbers of folds only.

(iv) Of course, the benefit stated in (iii) is no more a benefit if our algorithm is a time-consuming algorithm itself. However, as analyzed in the last section, this worry is not necessary because, as stated in Section 5, the computation load of Algorithm 1 is of order Kn only, and it takes only a few seconds for each shape sketched in this paper to get the value of n. As for the method used in (Leou and Tsai, 1987), it is slower than ours because besides the rotation-matching step, it also requires the use of some boundary-detection techniques. Also note that their method requires that the shape has no hole in it while our method does not (see Fig. 3 and the fourth shape of Fig. 2 for example).

7. Summary

We have successfully designed a simple method to compute the number of folds n, and hence the shape orientation, of an arbitrarily given rotationally symmetric shape. The idea is based on a simple but important mathematical property shared by all rotationally symmetric shapes. Skills to overcome the difficulty caused by sampling and rounding errors are also discussed. The experimental results showed that this new method works well, and hence it is not only of theoretical value but also useful in practice. The orientation of the given rotationally symmetric shape is just a by-product (without extra computational effort) when we use our method to detect the number of folds contained in the shape. The proposed method is thus "two-in-one" in nature, and this advantage makes our method superior to the method introduced in (Leou and Tsai, 1987) for detecting the numbers of folds contained in rotationally symmetric shapes. Another advantage is that the method in (Leou and Tsai, 1987) requires that the shape has no hole in it while our method does not.

As a concluding remark of this study, we emphasize below the simplicity of the proposed method. First, the proposed method does not require complicated image pre-processing procedures such as the boundary-detection technique that is used in (Leou and Tsai, 1987). Second, the operations used are just some simple operations like addition, multiplication, division, and comparison of two real numbers. All these operations are quite basic and standard for all computers. As a result, the proposed method can be implemented easily on any kind of computer. Third, the time complexity is not high. Fourth, shape orientation can also be detected without extra computational effort. And at last, all the benefits mentioned above are achieved using a very simple and short mathematical statement, namely, $x^{(l)}+\mathrm{i}y^{(l)}=0$ whenever l is not a multiple of the number of folds.

Appendix

Proof of Theorem 1

The polar coordinate representation of the given shape $S=\{(x_k, y_k)\}_{k=1}^{K}$ is

$$S=\{r_k \mathrm{e}^{\mathrm{i}\theta_k}\}_{k=1}^{K}. \tag{8}$$

Since S is an n-RSS, Eq. (8) can be rewritten as

$$S=\bigcup_{j=1}^{n}\{r_m \mathrm{e}^{\mathrm{i}\theta_{mj}} \mid m=1, 2, ..., M, \text{ and } \theta_{mj}=\theta_{m1}+j2\pi/n\} \tag{9}$$

where M is the number of points in a fold S and n is the number of mutually disjoint folds contained in S. Therefore $k=n\cdot m$. Here a fold of S is defined as any contiguous region of the n-RSS S bounded by an angular range of $2\pi/n$.

Substituting Eq. (9) into Eq. (3), we have

$$x^{(l)}+\mathrm{i}y^{(l)}=\sum_{m=1}^{M}\sum_{j=1}^{n}(r_m e^{\mathrm{i}l\theta_{mj}})=\sum_{m=1}^{M}r_m\left(\sum_{j=1}^{n}e^{\mathrm{i}l\theta_{mj}}\right)$$

$$=\sum_{m=1}^{M}r_m\left(\sum_{j=1}^{n}\cos(l\theta_{mj})+\mathrm{i}\sum_{j=1}^{n}\sin(l\theta_{mj})\right). \quad (10)$$

It suffices to show that both $\sum_{j=1}^{n}\cos(l\theta_{mj})$ and $\sum_{j=1}^{n}\sin(l\theta_{mj})$ are zero whenever l/n is not an integer. By Eq. (9) as well as the assumption that l/n is not an integer, we have

$$\sum_{j=1}^{n}\cos(l\theta_{mj})=\sum_{j=1}^{n}\cos[l(\theta_{m1}+j2\pi/n)]$$

$$=\sum_{j=1}^{n}[\cos(l\theta_{m1})\cos(j2\pi l/n)$$

$$-\sin(l\theta_{m1})\sin(j2\pi l/n)]$$

$$=\cos(l\theta_{m1})\sum_{j=1}^{n}\cos(j2\pi l/n)$$

$$-\sin(l\theta_{m1})\sum_{j=1}^{n}\sin(j2\pi l/n) \quad (11)$$

$$=\cos(l\theta_{m1})\left[\cos\left(\frac{n+1}{n}\pi l\right)\frac{\sin\pi l}{\sin\pi l/n}\right]$$

$$-\sin(l\theta_{m1})\left[\sin\left(\frac{n+1}{n}\pi l\right)\frac{\sin\pi l}{\sin\pi l/n}\right] \quad (12)$$

$$=\cos(l\theta_{m1})\cdot[0]-\sin(l\theta_{m1})\cdot[0]$$

$$=0 \quad (13)$$

where the reduction of expression (11) to (12)is based on the following pair of trigonometric formulas

$$\sum_{j=1}^{n}\cos(jt)=\cos[(n+1)t/2]\frac{\sin(nt/2)}{\sin(t/2)}$$

and

$$\sum_{j=1}^{n}\sin(jt)=\sin[(n+1)t/2]\frac{\sin(nt/2)}{\sin(t/2)}$$

which can be found in (Dwight, 1961). $\sum_{j=1}^{n}\sin l\theta_{mj}=0$ can be proved similarly. From Eq. (10) we have proved that $x^{(l)}+\mathrm{i}y^{(l)}=0$. □

References

Atallah, M.J. (1985). On symmetry detection. *IEEE Trans. Comput.* 34 (7), 663–666.

Chou, S.L., J.C. Lin and W.H. Tsai (1991). Fold-principal axis – a new tool for defining the orientations of rotationally symmetric shapes. *Pattern Recognition Lett.* 12 (2), 109–115.

Dwight, H.B. (1961). *Tables of Integrals and Other Mathematical Data*, 4th edition. MacMillan, New York.

Leou, J.J. and W.H. Tsai (1987). Automatic rotational symmetry determination for shape analysis. *Pattern Recognition* 20 (6), 571–582.

Lin, J.C., S.L. Chou and W.H. Tsai (1992). Detection of rotationally symmetric shape orientations by fold-invariant shape-specific points. *Pattern Recognition* 25 (5), 473–482.

Mitiche, A. and J.K. Aggarwal (1983). Contour registration by shape-specific points for shape matching. *Computer Vision, Graphics, and Image Processing* 22, 396–408.

Parui, S.K. and D.D. Majumder (1983). Symmetry analysis by computer. *Pattern Recognition* 16 (1), 63–67.

Rosenfeld, A. and A.C. Kak (1982). *Digital Picture Processing, Vol. II*. Academic Press, New York.