

A New Method for Fast Computation of Moments Based on 8-neighbor Chain Code Applied to 2-D Object Recognition

Guang-rong Ji, Guo-yu Wang, Zweitze Houkes*, Bing Zheng, Yan-ping Han

(Dept. of Electrical Engineering, Ocean University of Qingdao, Qingdao, 266003, P. R. China)

(* Dept. of Electrical Engineering, University of Twente, Enschede, The Netherlands)

Abstract – 2-D moment invariants have been successfully applied in pattern recognition tasks. The main difficulty of using moment invariants is the computational burden. To improve the algorithm of moments computation through an iterative way, an approach for fast computation of moments based on the 8-neighbor chain code is proposed in this paper. Then artificial neural networks are applied for 2D shape recognition with moment invariants. Compared with the method of polygonal approximation, this approach shows higher accuracy in shape representation and faster recognition speed in experiments.

I. INTRODUCTION

2-D moment invariants is an efficient descriptor for 2D shape representation and recognition because of its invariant quality under the operation of translation, rotation and scaling [3] [6]. But the main problem of using moment invariants in practice is the computational burden. Some methods for fast computation of moments have been proposed, such as polygonal-based method [7], δ -function [11], IIR-filtering [5] and the Pascal-Green-based method [2], etc.. The polygonal-based method saves the calculation time by reducing the number of feature points through polygonal approximation. But it should be noticed that the process of polygonal approximation is also time-consuming and such an approximation may induce large deviation in representing some complicated 2D objects. In case of good quality in image acquisition, the boundary description with 8-neighbor chain code can accurately represent a 2D shape. Jiang [7] proposed an iterative way to compute moments with 8-neighbor chain code. For the purpose of fast recognition of 2D objects, we proposed a method to improve the moment computation based on the approach of [7]. Then artificial neural networks are applied to finish the recognition task using invariant moments as the feature descriptor. Compared with the method of polygonal approximation, the method proposed in this paper showed higher accuracy in shape representation, and faster recognition speed with preliminary experimental results.

II. FAST COMPUTATION OF MOMENTS

A. Moments computation in an iterative way

A 2-D gray-level image can be represented with a continuous function $f(x,y)$. The moment of $f(x,y)$ in order of $(p+q)$ is defined as

$$m_{pq} = \iint x^p y^q f(x,y) dx dy, \quad (1)$$

where $p,q \in Z^+$, with $Z^+ = \{x \in Z | x \geq 0\}$ and Z belongs to set of all integers.

According to unique theorem [6], if $f(x,y)$ is piece-wise continuous and non-zero over a finite region of the x-y plane, moments of all orders exist and $\{m_{pq}\}$ are uniquely determined by $f(x,y)$. Conversely, the set of moments $\{m_{pq}\}$ uniquely determines $f(x,y)$. In many cases, shape of boundary is usually used for object recognition. After separation from the background, the binary image of the object can be described by

$$\begin{aligned} f(x,y) &= 1, \forall (x,y) \in D \\ f(x,y) &= 0, \forall (x,y) \notin D \end{aligned} \quad (2)$$

where D denotes the region of object in the image plane. Hence, (1) can be simplified as

$$m_{pq} = \iint_D x^p y^q dx dy \quad (3)$$

Let C denote the boundary curve embedding the simply-connected region D and assuming C to be piece-wise smooth. If the function $f(x,y)$ is integral over the region D and the curve-integration along C is counter-clockwise, then the right side of (3) can be decomposed by applying Stokes theorem in the form

$$\iint_C \left(\frac{\partial Q(x,y)}{\partial x} - \frac{\partial P(x,y)}{\partial y} \right) dx dy = \oint_C (P(x,y) dx + Q(x,y) dy) \quad (4)$$

$$\text{with } \frac{\partial Q(x,y)}{\partial x} - \frac{\partial P(x,y)}{\partial y} \Leftrightarrow x^p y^q \quad (5)$$

and the components $P(x,y)$ and $Q(x,y)$ are chosen as [7]

$$Q(x,y) = \frac{1}{p+1} x^{p+1} y^q, P(x,y) = 0, \quad (6)$$

Therefore, (3) can be formulated as

$$m_{pq} = \oint_C \frac{1}{p+1} x^{p+1} y^q dy \quad (7)$$

The integration is along the closed-curve of object boundary C . This curve is digitized of chain code in computation. The curve consists of n line segments $C_i, \forall i \in \{0, 1, \dots, n-1\}$, linked by chain coding. Supposing C_i starts at point $P_i(x_i, y_i)$ and ends at point $P_{i+1}(x_{i+1}, y_{i+1})$. Each link is determined by one of the eight directions.

Now m_{pq} in (7) can be computed with

$$m_{pq} = \frac{1}{p+1} \sum_{i=0}^{n-1} \int_{P_i}^{P_{i+1}} x^{p+1} y^q dy \quad (8)$$

Defining r_{pq} as

$$r_{pq}(i) = \int_{P_i}^{P_{i+1}} x^{p+1} y^q dy, \text{ then} \quad (9)$$

$$m_{pq} = \frac{1}{p+1} \sum_{i=0}^{n-1} r_{pq}(i)$$

According to the method proposed in [7], moments can be computed out in an iterative way with (9). It can be noticed that in the 8-neighbor chain code representation for the closed curve C , each segment C_i is inherently linked by the chain code. Concerning these characteristics, we proposed a way to improve the moment computation with the relationships among chain connections, so to reduce about half of the whole computational time. Let us think of possible cases along different chain directions for the computation of r_{pq}

Case 1: In the x-direction,

$$C_i: y = y_i, r_{pq}(i) = 0 \quad (10)$$

Case 2: In the y-direction,

$$C_i: x = x_i, \quad (11)$$

$$r_{pq}(i) = \int_{y_i}^{y_{i+1}} x_i^{p+1} y^q dy = \frac{1}{q+1} x_i^{p+1} (y_{i+1}^{q+1} - y_i^{q+1}) = \frac{1}{q+1} x_i^{p+1} r_{-1,q}(i)$$

Case 3: In other directions,

$$C_i: x = \frac{1}{k} y + x_i - \frac{1}{k} y_i, \quad (12)$$

where k is the slope of C_i ,

$$r_{pq} = \int_{y_i}^{y_{i+1}} \left(\frac{1}{k} y + \left(x_i - \frac{1}{k} y_i \right) \right)^{p+1} y^q dy = \frac{1}{k} \int_{y_i}^{y_{i+1}} \left(\frac{1}{k} y + \left(x_i - \frac{1}{k} y_i \right) \right)^p y^{q+1} dy + \left(x_i - \frac{1}{k} y_i \right) \int_{y_i}^{y_{i+1}} \left(\frac{1}{k} y + \left(x_i - \frac{1}{k} y_i \right) \right)^p y^q dy = \frac{1}{k} r_{p-1,q+1}(i) + \left(x_i - \frac{1}{k} y_i \right) r_{p-1,q}(i) \quad (13)$$

In case 3, the slope k can only be one of two possible values because of 8-neighbor chain coding. Its value can be $k=1$ or $k=-1$. Hence

$$r_{pq}(i) = \pm r_{p-1,q+1}(i) + \left(x_i \mp y_i \right) r_{p-1,q}(i) \quad (14)$$

Equation (14) can be used for computing $r_{pq}(i)$ iteratively. The finally required expression is directly obtained from the definition of $r_{pq}(i)$ by setting $p=-1$:

$$r_{-1,q}(i) = \int_{y_i}^{y_{i+1}} y^q dy = \frac{1}{q+1} (y_{i+1}^{q+1} - y_i^{q+1}) \quad (15)$$

From the discussion above, we can see that only the item $r_{-1,q}(i)$ has to be computed separately, while the other orders of $r_{pq}(i)$ can be computed iteratively using (13) or (14).

B. Fast computation of moments

From equation (15), the computation of $r_{pq}(i)$ is determined by the computation of polynomial $(y_{i+1}^{q+1} - y_i^{q+1})$. Considering the relationship of two adjacent points linked

with 8-neighbor chain code, there is $y_{i+1} = y_i \pm \delta$ with $\delta \in \{0,1\}$. In case of $\delta=1$, we have

$$\begin{aligned} y_{i+1} &= y_i \pm 1 \\ y_{i+1}^2 &= (y_i^2 \pm y_i) \pm y_{i+1}; \\ y_{i+1}^3 &= (y_i^3 \pm y_i^2) \pm (y_i^2 \pm y_i) \pm y_{i+1}^2; \\ &\dots \end{aligned} \quad (16)$$

Once the values of $y_i^1, y_i^2, \dots, y_i^{q+1}$, are known, computation for $y_{i+1}^1, y_{i+1}^2, \dots, y_{i+1}^{q+1}$ only need time for $1/2(q+1)(q+2)$ additions.

After the boundary extraction of an object and applying 8-neighbor chain coding to the closed curve C , the fast computation of moments is implemented as below:

- 1) For the starting point with $i=0$, compute the vector $(y_i^1, y_i^2, \dots, y_i^{q+1})$.
- 2) Using (16), compute the vector $(y_{i+1}^1, y_{i+1}^2, \dots, y_{i+1}^{q+1})$.
- 3) Determine the direction of the i th segment C_i . If C_i is in the x-direction, set $r_{pq}(i)=0$, $i=i+1$ and return to step 2, otherwise do step 4.
- 4) Compute $r_{-1,q}(i)$ and update $(y_i^1, y_i^2, \dots, y_i^{q+1})$ with $(y_{i+1}^1, y_{i+1}^2, \dots, y_{i+1}^{q+1})$, to prepare for calculation of next segment.
- 5) Determine the direction of C_i , and use (11) or (14) to compute $r_{pq}(i)$.
- 6) Let $i=i+1$, if $i=n$, the end point of the boundary has been reached, compute m_{pq} and stop. Otherwise return to step 2.

C. Simulation of computational time and comparison

As mentioned above, with the inherent iterative relationship of 8-neighbor chain code, the method for moment computation proposed in [7] is improved. Now the number of multiplication is reduced to a half time, for that the calculation of the polynomial for $r_{-1,q}(i)$ in (16) is finished with addition operations. According to (14) or (15), the computation of $r_{pq}(i)$ just needs one multiplication. Such improvement was confirmed by the simulated experiment to compare both methods. Fig.2 shows result of computational time (in seconds). The curve b indicates the result with the method of [7], while curve a indicates the result with our method. The order of moments is limited within $p+q \leq 3$.

III. APPLYING ARTIFICIAL NEURAL NETWORKS FOR MOMENT-BASED OBJECT RECOGNITION

A. Construction of the system

A system of moments-based object recognition with neural networks is depicted in Fig.1. Because artificial neural networks have advantages of parallel implementation, capability of learning and error tolerance, they have been widely used for recognition tasks. Since moment invariants can form the feature space for object description, it is suitable to construct the input vector of neural networks. The BP neural networks were used in our system.

B. Extraction of moment invariants

In order to eliminate the factors of rotation, translation and scaling in object measurement, the normalized central moments are used as the invariant feature for 2D recognition. Under the constraint of the order of moments within $p+q \leq 3$, there are seven functions for representation of the normalized central moments. These seven moment invariants were used to form the input vector of the neural networks.

C. Experimental results

Fig.4 are the five synthetic 2D objects that were used for recognition in our experiment. The BP networks consist of three layers. There are 7 input neurons corresponding to the 7 moment invariants and 5 output neurons corresponding to five patterns of the objects to be recognized.

Each object was rotated in step of one degree over 360° and the seven moment invariants were computed out respectively. The results were used as the training vectors for the BP networks. The BP networks were trained 15,000 times. The error curve is depicted in Fig.3.

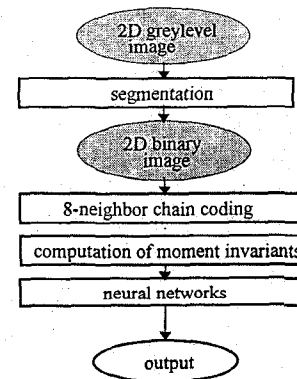


Fig.1 A block diagram of the recognition system

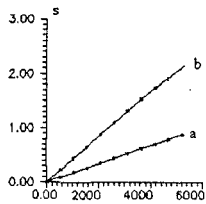


Fig. 2 Computational time as a function of the number of sides of polygon

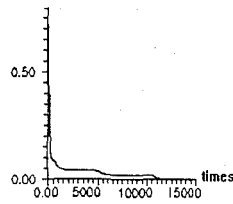


Fig.3 The training error

After training, the BP networks were used to recognize all the five objects in Fig.4. The result is shown in Tab.1, where the symbol (y_1, y_2, y_3, y_4, y_5) denote the values of the five outputs and (a,b,c,d,e) denote the type of object as input. The certain output with value near to 1 corresponds to a certain input object while other outputs have the values near to 0. All the five objects were correctly classified in our experiments. Furthermore, Other objects which were not included in the training set, were also distinguished to those known objects.

D. A comparison with the polygon-based method

The polygon-based method applies polygonal approximation algorithm for shape analysis and recognition [4] [10]. Usually, the feature points are extracted by detection of the local maximum of curvatures and the computational time is mainly consumed by the computation of the local maximum detection. According to the method proposed in [4], the shape of polygonal approximation is closely related to the a prior parameter D_{th} which is used as threshold for detection of local maximum of curvature (the readers are referred to [4] for more details). We noticed that in the computation of feature extraction, number of multiplying operations increases exponentially as D_{th} increases. Therefore, to simplify the shape of an object by polygonal approximation with a larger D_{th} , which causes time reduction in moments computation, the computational burden in feature detection has to be imposed in the stage of polygonal approximation.

Tab.2 is the experimental results for the comparison of the whole time used for recognition by our method and the polygon-based method. The time T (in ms) indicate the results with our method and t (in ms) indicate the polygon-based method with different D_{th} . The five objects shown in Fig.4 were used in experiments. The results were obtained by carrying out both methods on IBM-486 computer.

IV. CONCLUSIONS AND DISCUSSION

In this paper, we proposed an approach for fast computing moment invariants. 2-D objects are directly represented with 8-neighbor chain code. Based on the analysis of the inherent relationship among joints of boundary described by 8-

TABLE I
Experimental results of recognition with neural networks

	a	b	c	d	e
y_1	0.90	0.10	0.10	0.10	0.10
y_2	0.10	0.90	0.10	0.10	0.10
y_3	0.10	0.10	0.90	0.10	0.10
y_4	0.10	0.10	0.10	0.90	0.10
y_5	0.10	0.10	0.10	0.10	0.90

TABLE II
The whole time used for recognition in experiments

type of object	a	b	c	d	e
T	94	88	91	61	60
$D_{th}=0.10$ t	720	900	610	770	550
$D_{th}=0.20$ t	1320	1360	900	2200	2420
$D_{th}=0.25$ t	1430	1480	1540	2580	2530
$D_{th}=0.30$ t	1530	1590	1760	2800	2460
$D_{th}=0.40$ t	2420	2250	3070	3520	3080
$D_{th}=0.50$ t	3020	3130	4120	5660	3460

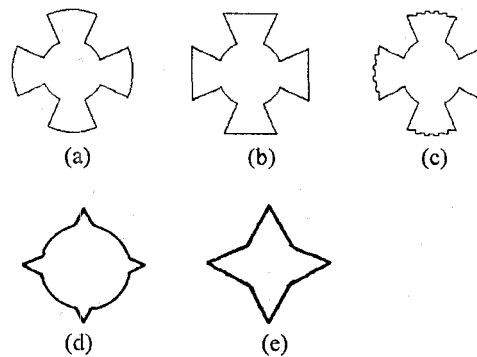


Fig. 4 The objects used in experiments

neighbor chain code, the method of moments computation in an iterative way has been improved. With the 7 functions describing the moment invariants, neural networks are directly applied to implement the recognition tasks. The improved method for moments computation in iterative way proposed in this paper can save about half of the computational time. The experimental results also showed that the whole recognition time taken in our approach is less than that of the polygon-based method. Obviously, in case of noise absence, 8-neighbor chain code used in this paper can yield to more details in object representation and so to high accuracy of moments computation and recognition. But if the acquisition of images are noise-corrupted, the approach used in this paper is expected to be improved to keep robustness of object representation and validity of moments computation the same time.

V. ACKNOWLEDGMENTS

This paper is supported by National "863" Foundation No: 03-08 and the National Science Foundation No: 69675005.

VI. REFERENCES

- [1] Bing-Cheng Li, "A new computation of geometric moments", *Pattern Recognition*, no. 26, 1993, pp.109-113.
- [2] Bing-Cheng Li and Jun Shen, "Fast computation of moment invariants", *Pattern Recognition*, no. 24, 1991, pp. 807-813.
- [3] Bing-Cheng Li, "The Moment Calculation of polyhedra", *Pattern Recognition*, no. 26, 1993, pp. 1229-1233.
- [4] C.C. Chang, S.M. Hwang and D.J.Buehrer, "A shape recognition scheme based on relative distances of feature points from the centroid" , *Pattern Recognition* , no. 24, 1991, pp. 1053-1063.
- [5] M. A Hatamian, "Real-time two dimensional moment generating algorithm and its single chip implementation", *IEEE Trans. ASSP*, vol. 34, no. 3, 1986, pp.546-553.
- [6] M.K. Hu, "Visual pattern recognition by moment invariance", *IRE Trans. Inf. Theory*, no. 8, 1962, pp. 179-187.
- [7] X.Y. Jiang and H. Bunke, "Simple and fast computation of moments", *Pattern Recognition* , no. 24, 1991, pp. 801-806.
- [8] D.A. Mitzias and B.C. Mertzios, "Shape recognition with a neural classifier based on a fast polygon approximation technique", *Pattern Recognition*, no. 27, 1994, pp. 627-636.
- [9] A.P. Reeves, "The general theory of moment operations", Purdue University Tech. Rep. TR-EE 81-37, October, 1981.
- [10] C.H. Teh. and R.T. Chin, "On the detection dominant points on digital curves", *IEEE Trans. PAMI* vol. 11, 1989, pp. 859-872.
- [11] M.F. Zakaria, L.J. Vroomen, P.J.A. Zsombor-Murray and M. Kessel, "Fast algorithm for computation of moment invariants", *Pattern Recognition*, no. 20, 1987, pp. 639-643.