Mathematical representation of 2D image boundary contour using fractional implicit polynomial*

TONG Yuerong[1,2], YU Lina[1,2,3], LI Weijun[1,2,3], LIU Jingyi[1,2,3], WU Min[1,2,3], and YANG Yafei[4]**

1. Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China

2. School of Materials Science and Optoelectronic Technology & School of Integrated Circuits, University of Chinese Academy of Sciences, Beijing 100049, China

3. Beijing Key Laboratory of Semiconductor Neural Network Intelligent Sensing and Computing Technology, Beijing 100083, China

4. DapuStor Corporation, Shenzhen 518100, China

(Received 22 November 2022; Revised 11 January 2023)

Implicit polynomial (IP) fitting is an effective method to quickly represent two-dimensional (2D) image boundary contour in the form of mathematical function. Under the same maximum degree, the fractional implicit polynomial (FIP) can express more curve details than IP and has obvious advantages for the representation of complex boundary contours. In existing studies, algebraic distance is mainly used as the fitting objective of the polynomial. Although the time cost is reduced, there are problems of low fitting accuracy and spurious zero set. In this paper, we propose a two-stage neural network with differentiable geometric distance, which uses FIP to achieve mathematical representation, called TSEncoder. In the first stage, the continuity constraint is used to obtain a rough outline of the fitting target. In the second stage, differentiable geometric distance is gradually added to fine-tune the polynomial coefficients to obtain a contour representation with higher accuracy. Experimental results show that TSEncoder can achieve mathematical representation of 2D image boundary contour with high accuracy.

Document code: A **Article ID:** 1673-1905(2023)04-0252-5

DOI https://doi.org/10.1007/s11801-023-2199-6

Mathematical representation of contours has many applications in geometric modeling[1], target recognition[2-4] and posture estimation[5,6]. Implicit polynomial (IP) can be used for image compression, transmission and reconstruction due to its simplicity, stability and robustness[7,8]. Specifically, IP is simple, can describe irregularly shaped objects with a few parameters, and has good analytical expression. IP is stable and its coefficients are insensitive to noise, so it can be used to identify objects quickly. IP is robust and can complete the analytic expression of the part of data missing due to occlusion, which is easy to operate and use.

In general, it is necessary to increase the maximum degree of IP to represent the two-dimensional (2D) object contours of complex shape. However, increasing degree will add computational cost and reduce stability. Therefore, it is difficult for the traditional integer IP to represent the complex 2D image boundary contour. Moreover, higher-order IP does not guarantee the accuracy of fitting complex objects. Even though the data set is always continuous and bounded, the result may appear spurious, discontinuous and unbounded zero sets phenomenon[9-12]. The higher degree of the polynomial, the more sensitive the roots of the polynomial are to the perturbations of the coefficients[13]. Minor changes of the coefficient of IP can produce significant changes to the contour representation results. Based on the IP, fractional implicit polynomial (FIP) transfers the degree of the polynomial from an integer to a fraction. FIP can improve the contour representation accuracy without increasing the maximum degree.

At present, many polynomial fitting methods have been proposed, which can be divided into traditional methods and deep learning methods. Traditional methods, such as 3L[14], Min-Max[15] and Min-Var[15], add different constraints to calculate polynomial coefficients by means of least square method. The most representative deep learning method is Encoder-X[16], which uses encoder to solve polynomial coefficients. The deep learning method is better than the traditional method in the accuracy of fitting and has the ability to resist noise interference. Therefore, we borrowed the idea of Encoder-X, in order to further exploit the advantages of deep learning. Since the training result of neural network is largely related to

* This work has been supported by the Key Research Program of the Chinese Academy of Sciences (No.XDPB22), and the Fund of Guangdong Support Program (No.2019TY05X071).

** E-mail: wjli@semi.ac.cn

the initial value of the network, the training process of neural network can be divided into two stages. In the first stage, continuity constraints are used to get the rough outline of the fitting object, which is used as a reliable initial value of the network, and then the details are optimized in the second stage.

In the IP fitting task, how to evaluate the fitting result is an important step. Current IP fitting studies generally evaluate the error from the original data point to the IP curve by calculating the algebraic distance or geometric distance[17]. The algebraic distance can be calculated simply by substituting the original data points into the IP expression, but the geometric information of the data is lost. In addition, algebraic distance depends on coordinates, is a biased parameter estimation, and does not conform to visual intuition[17]. Geometric distance is to calculate the orthogonal distance between the data point and the IP curve. Geometric distance solves the problem of algebraic distance as a measure of distance. Due to the difficulty of calculation and the consideration of calculation cost, the approximate value of geometric distance is generally used in practice to achieve the balance between calculation accuracy and calculation cost. In addition, in order to apply geometric distance to the optimization process of neural network, the calculation process of geometric distance needs to be differentiable.

Therefore, we propose a new framework for 2D image contour representation using FIP, called TSEncoder. The differentiable geometric distance is used as the optimization objective in the fitting process, and the two-stage optimization neural network is used to improve the contour representation accuracy. The experiment results show that TSEncoder can achieve higher accuracy.

IP fitting determines the most appropriate IP coefficient vector based on the specified data set. In recent years, many methods have been developed to achieve IP fitting, such as 3L algorithm[14], gradient-one[18], ridge-regression[18], Min-Max[15] and Min-Var[15]. The 3L algorithm[14] is applied to solved linear least squares problems by shrinking and extending the data set, incorporating additional linear constraints into the fitting process. Modified 3L[15] is modified on the basis of 3L, from scaling the data set to obtaining the 3L set through the gradient direction of the data set. Min-Max and Min-Var[15] use linear least squares with additional constraints to produce more stable and accurate results. In addition to traditional IP methods, due to the rapid development of deep learning in recent years, Encoder-X is proposed to solve polynomial coefficients by using neural networks. Encoder-X[16] regards polynomial coefficients as the eigenvalues of original data in polynomial space expression. It consists of an encoder defined by a neural network and a decoder defined by a polynomial mathematical expression.

In order to further improve the IP contour representation accuracy, HU et al[17] proposed the definition of FIP. FIP is an extension of IP. The FIP curve is the zero set of a smooth two-variable polynomial function $f_{nm}(x, y)$ which is

$$f_{nm}(x,y)=\sum_{\substack{m\le(i+j)\le(nm)\\ i,j=0 \text{ or } (i,j)\ge m}} a_{\frac{i}{m}\frac{j}{m}}\, x^{\frac{i}{m}} y^{\frac{j}{m}}=0, \tag{1}$$

where n represents the degree and m represents the base. n and m are both integers greater than 1, and m is required to be an odd-number. However, up to now, no further studies have been conducted to use FIP for contour representation due to the complexity of FIP solving.

Distance measures used in polynomial fitting can generally be classified into algebraic distance and geometric distance. All of the above polynomial fitting methods use algebraic distance as the optimization objective. Although the calculation of algebraic distance is simple, the geometric information of the data is lost. Therefore, in order to further improve the accuracy of mathematical representation of boundary contour, geometric distance can be used as the distance measure. Currently, some scholars have proposed some geometric distance calculation methods for IP fitting task[17,19-21], but the calculation process is quite complicated.

We apply deep learning to solve the coefficients of FIP. The framework of TSEncoder is shown in Fig.1. Once an image is received, data pre-processing is first performed, which includes contour detection and normalization.

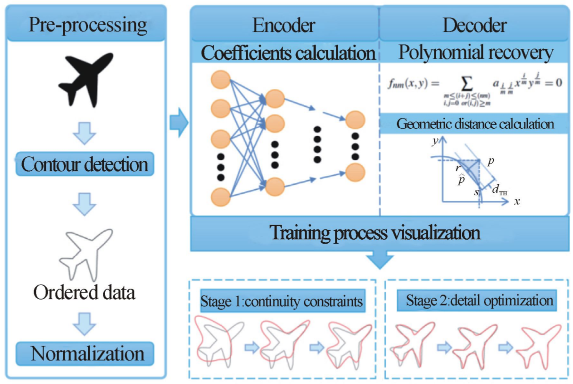

Fig.1 Framework of TSEncoder

The contour detection algorithm[22] can obtain ordered data points. And then TSEncoder performs the normalization processing on the ordered data points. The processed data is fed into the network, where the encoder calculates the polynomial coefficients and the decoder takes the coefficients as input to restore the FIP.

In order to evaluate the accuracy of the contour representation, the height of the triangle constructed between the data points and the FIP curve is calculated as an approximation of the geometric distance. The triangle used for estimating the geometric distance is shown in Fig.2.

The height of triangle d_{TH} can be easily obtained as follows

$$d_{\mathrm{TH}}=\frac{|pr|\cdot|ps|}{\sqrt{(|pr|)^2+(|ps|)^2}}. \tag{2}$$

The approximation along the x axis is obtained from the first order Taylor expansion

$$f(x,y_i)\simeq f(x_i,y_i)+f_x(x_i,y_i)\cdot(x-x_i). \tag{3}$$

The segment $|pr|$ can be easily estimated as

$$|pr|\simeq -f(p_i)/f_x(p_i). \tag{4}$$

Therefore, the approximate value d_{TH} of the differentiable geometric distance of point P can be estimated as

$$d_{\mathrm{TH}}\simeq\frac{|f/f_x|\cdot|f/f_y|}{f\sqrt{(1/f_x)^2+(1/f_y)^2}}=\frac{|f|}{\|\nabla f\|}. \tag{5}$$

Cumulative geometric distance estimates for all points can be obtained as follows

$$dist=\sum_{i=1}^{m}d_{\mathrm{TH}}^2(p_i)=\sum_{i=1}^{m}\frac{f^2}{f_x^2+f_y^2}. \tag{6}$$

The differentiable geometric distance metric has the advantage of preserving the gradient information, thus it can be applied to the optimization process of neural network. More details can be found in Ref.[23].

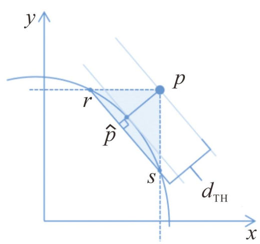

Fig.2 Geometric distance approximation

The initial value of the neural network has an important impact on the results, so the neural network is designed to use a two-stage optimization approach. The first stage uses a continuity constraint in order to obtain a rough outline of the target object as the initial value for the second stage of neural network training.

Continuity constraints refer to Ref.[18], where f_x is the partial derivative on the X-axis and f_y is the partial derivative on the Y-axis. N_x and N_y are the norm vectors of the original data points. The continuity constraint C is the average cosine distance between the norm vector and the gradient vector.

$$C=\frac{1}{m}\sum_{i=1}^{m}\frac{f_xN_x+f_yN_y}{\sqrt{f_x^2+f_y^2}}. \tag{7}$$

Since the model number vector in the table is parallel to the gradient vector when the continuity constraint C is close to 1, the loss in the optimization process of the first stage of the neural network is defined as

$$loss=|C-1|. \tag{8}$$

In the second stage, details are optimized. With the iteration of the neural network, the proportion of continuity constraints is continuously reduced and the proportion of geometric distance is increased. The second stage can make the IP curve closer to the contour of the object. The scale factor *ratio* is the ratio of the current epoch to the maximum number of epochs.

$$ratio=epoch\,/\,EPOCH. \tag{9}$$

Thus, the loss in the optimization process of the second stage of the neural network is defined as

$$loss=(1-ratio)\times|C-1|+ratio\times dist. \tag{10}$$

We use the Adam as optimizer. We set the initial learning rate as 0.003 and the learning rate to decay by 0.5 every 1 000 steps. We set the epoch to 5 000. When the network has been iterated 2 000 times, the TSEncoder starts the second stage optimization.

To perform the experiments for contour representation, 20 objects are collected as shown in Fig.3. The dataset is derived from Refs.[7,14,15,18] as well as the Internet.

Fig.3 Objects used in the experiments

For each fitting object, we select the most appropriate degree for FIP. In order to verify the effectiveness of the two-stage neural network, we conduct ablation experiments and results can be seen in Fig.4. The TSEncoder (one stage) only uses the geometric distance as the loss function of the neural network without the continuity constraint. By comparing the results of the ablation experiment, it can be observed that the fitting curves obtained by (b) TSEncoder (one stage) are neither closed nor ideal. In contrast, the experimental results of (a) TSEncoder show that the continuity constraint allows the network to obtain better initial values and facilitates a more accurate contour representation effect.

We conducted experiments to verify the effect of using different distance metrics on the contour representation. The experimental results in Fig.4(a) and (c) indicate that the algebraic distances do not preserve the geometric information and therefore do not reflect the contour representation realistically. Ultimately, it leads to a contour representation with low accuracy.

In order to verify the missing data completion ability of TSEncoder algorithm, experiments are conducted under different missing proportions. We set the proportion of missing data as 3%, 5% and 8%, respectively. The experimental results are shown in Fig.5. It can be seen that the TSEncoder can achieve contour representation with high accuracy at different missing proportions.

To verify the performance of TSEncoder, we conduct detailed experiments. We compare TSEncoder with the current popular polynomial fitting methods, namely Min-Max, Min-Var, Modified 3L, and Encoder-X, which all use IP. In order to ensure the fairness of our experiment, we modify these methods by using FIP. For Encoder-X, we refer to the parameter settings in Ref.[16].

Both TSEncoder and Encoder-X are polynomial fitting methods based on deep learning. The two methods are different in two aspects: distance measurement and neural network optimization. TSEncoder uses geometric distance as the measure of fitting results, while Encoder-X uses geometric distance. TSEncoder optimizes the polynomial coefficient solving process using a two-stage neural network. In the first stage, continuity constraints are used to provide reliable initial values. In the second stage, geometric distance is gradually added to optimize details. Encoder-X uses only one stage to optimize the neural network. By applying constraints similar to the 3L algorithm, data enhancement is carried out by zooming in and out of data points. The specific experimental results are shown in Fig.4. As can be seen from the experimental results, the results obtained by TSEncoder have the highest accuracy. TSEncoder can obtain closed, smooth FIP curves. But the other four methods can not fit the contour of the object well, and there is spurious zero set phenomenon. Tab.1 lists the geometric distance of different methods. It shows that TSEncoder has the lowest geometric distance of the 20 objects.

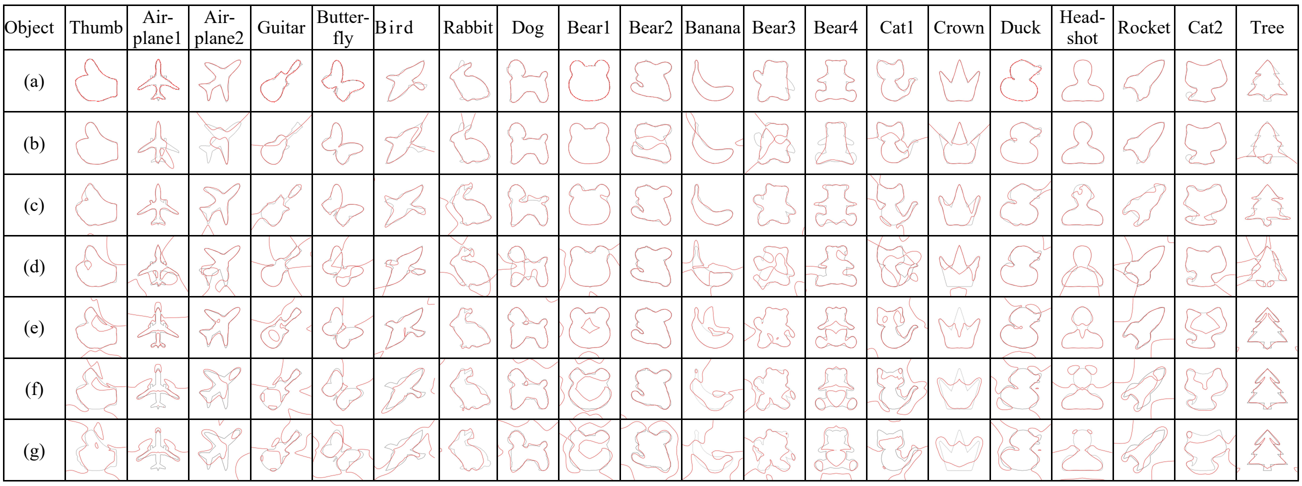

Fig.4 Experimental results: (a) TSEncoder; (b) TSEncoder (one stage); (c) TSEncoder (algebraic distance); (d) Encoder-X; (e) Min-Max; (f) Min-Var; (g) Modified 3L

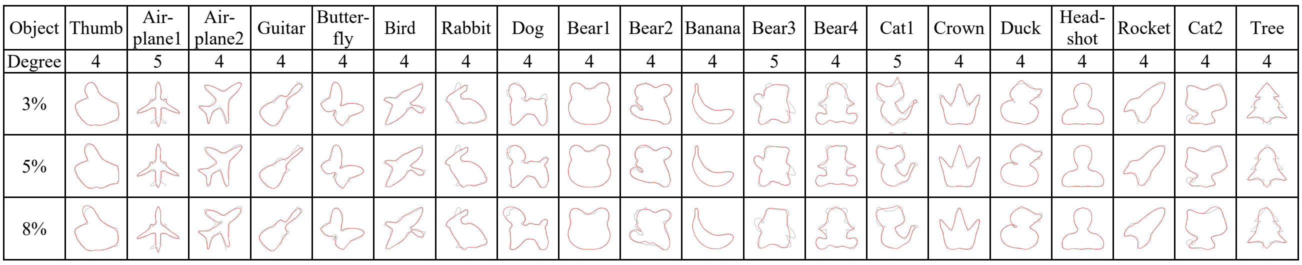

Fig.5 Missing data completion capacity

Tab.1 Geometric distance comparison

	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Thumb	**0.12**	0.19	2.79	1.04	0.81	4.92	15.15
Airplane1	**2.57**	9.66	5.57	104.73	11.77	37.30	52.25
Airplane2	**0.86**	41.75	5.83	28.80	3.75	27.79	66.07
Guitar	**1.09**	5.60	7.07	13.08	1.52	3.85	4.59
Butterfly	**1.13**	1.27	6.06	7.61	2.32	15.21	22.78
Bird	**2.45**	5.14	10.61	7.92	2.77	235.84	206.54
Rabbit	**1.98**	2.17	15.87	6.22	4.13	9.11	15.60
Dog	**0.69**	0.76	9.58	9.44	1.29	2.91	4.76
Bear1	**0.11**	0.27	1.19	1.35	0.35	4.35	11.20
Bear2	**0.81**	3.83	1.56	1.45	0.83	2.90	7.48
Banana	**0.07**	0.40	0.34	5.69	0.61	3.51	6.35
Bear3	**2.23**	3.34	3.45	10.17	2.72	13.06	17.68
Bear4	**1.00**	16.22	8.18	20.00	2.34	27.94	127.20
Cat1	**3.12**	4.51	284.01	35.03	3.71	15.94	65.49
Crown	**0.12**	2.23	3.04	16.73	1.85	15.53	67.89
Duck	**0.94**	1.06	21.02	7.31	7.21	8095.84	33.20
Headshot	**0.08**	0.40	1.94	2.47	0.75	3.53	6.05
Rocket	**2.09**	3.55	14.88	3.00	2.55	45.51	73.99
Cat2	**1.60**	1.66	10.58	1.71	1.72	15.94	35.96
Tree	**8.08**	12.05	123.97	23.48	8.95	9.11	9.18

Note: (a) TSEncoder; (b) TSEncoder (one stage); (c) TSEncoder (algebraic distance); (d) Encoder-X; (e) Min-Max; (f) Min-Var; (g) Modified 3L

In order to improve the fitting accuracy of IP, we propose a contour representation method based on deep learning using FIP, called TSEncoder. We adopt the approximate calculation method of differentiable geometric distance which can save the gradient information and be used in neural network optimization. Our ablation experiment proves the advantage of using two-stage neural network to optimize the coefficients. Continuity constraints can help the network get a better initial value, which is conducive to the optimization of the solution process of the neural network. The experiment of missing data completion proves that our method can still achieve high precision fitting when the object is blocked. In our comparison experiment, the contour representation accuracy of TSEncoder is much higher than that of the other four methods, and there is no spurious zero set phenomenon in the results. In the future, we will apply the proposed model to 3D surface modeling scenarios.

Statements and Declarations

The authors declare that there are no conflicts of interest related to this article.

References

[1] TAUBIN G, CUKIERMAN F, SULLIVAN S, et al. Parameterized families of polynomials for bounded algebraic curve and surface fitting[J]. IEEE transactions on pattern analysis and machine intelligence, 1994, 16(3): 287-303.

[2] HUAMIN T, JIANJUN Y, CHUNLEI Z. Polarization radar target recognition based on optimal curve fitting[C]//Proceedings of the IEEE 1998 National Aerospace and Electronics Conference, August 31- September 4, 1998, Dayton, USA. New York: IEEE, 1998: 434-437.

[3] JI C, YANG X, WANG W. A novel method for image recognition based on polynomial curve fitting[C]//2015 8th International Symposium on Computational Intelligence and Design, December 12-13, 2015, Hangzhou, China. New York: IEEE, 2015: 354-357.

[4] ODEN C, ERCIL A, BUKE B. Combining implicit polynomials and geometric features for hand recognition[J]. Pattern recognition letters, 2003, 24(13): 2145-2152.

[5] ODRY Á, FULLÉR R, RUDAS I J, et al. Kalman filter for mobile-robot attitude estimation: novel optimized and adaptive solutions[J]. Mechanical systems and signal processing, 2018, 110: 569-589.

[6] TAREL J P, COOPER D B. A new complex basis for implicit polynomial curves and its simple exploitation for pose estimation and invariant recognition[C]//1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 25, 1998, San Diego, CA, USA. New York: IEEE, 1998: 111-117.

[7] WU G, ZHANG Y. A novel fractional implicit polynomial approach for stable representation of complex shapes[J]. Journal of mathematical imaging and vision, 2016, 55(1): 89-104.

[8] WU G, YANG J. A representation of time series based on implicit polynomial curve[J]. Pattern recognition letters, 2013, 34(4): 361-371.

[9] TAUBIN G. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation[J]. IEEE transactions on pattern analysis & machine intelligence, 1991, 13(11): 1115-1138.

[10] TAUBIN G, CUKIERMAN F, SULLIVAN S, et al. Parameterized families of polynomials for bounded algebraic curve and surface fitting[J]. IEEE transactions on pattern analysis and machine intelligence, 1994, 16(3): 287-303.

[11] KEREN D, GOTSMAN C. Fitting curves and surfaces with constrained implicit polynomials[J]. IEEE transactions on pattern analysis and machine intelligence, 1999, 21(1): 31-41.

[12] KEREN D, COOPER D, SUBRAHMONIA J. Describing complicated objects by implicit polynomials[J]. IEEE transactions on pattern analysis and machine intelligence, 1994, 16(1): 38-53.

[13] GUILLAUME P, SCHOUKENS J, PINTELON R. Sensitivity of roots to errors in the coefficient of polynomials obtained by frequency-domain estimation methods[J]. IEEE transactions on instrumentation and measurement, 1989, 38(6): 1050-1056.

[14] BLANE M M, LEI Z, CIVI H, et al. The 3L algorithm for fitting implicit polynomial curves and surfaces to data[J]. IEEE transactions on pattern analysis and machine intelligence, 2000, 22(3): 298-313.

[15] HELZER A, BARZOHAR M, MALAH D. Stable fitting of 2D curves and 3D surfaces by implicit polynomials[J]. IEEE transactions on pattern analysis and machine intelligence, 2004, 26(10): 1283-1294.

[16] WANG G, LI W, ZHANG L, et al. Encoder-X: solving unknown coefficients automatically in polynomial fitting by using an autoencoder[J]. IEEE transactions on neural networks and learning systems, 2021.

[17] HU M, ZHOU Y, LI X. Robust and accurate computation of geometric distance for Lipschitz continuous implicit curves[J]. The visual computer, 2017, 33(6): 937-947.

[18] TASDIZEN T, TAREL J P, COOPER D B. Improving the stability of algebraic curves for applications[J]. IEEE transactions on image processing, 2000, 9(3): 405-416.

[19] WANG W, POTTMANN H, LIU Y. Fitting B-spline curves to point clouds by curvature-based squared distance minimization[J]. ACM transactions on graphics, 2006, 25(2): 214-238.

[20] SONG X, JÜTTLER B. Modeling and 3D object reconstruction by implicitly defined surfaces with sharp features[J]. Computers & graphics, 2009, 33(3): 321-330.

[21] UPRETI K, SONG T, TAMBAT A, et al. Algebraic distance estimations for enriched isogeometric analysis[J]. Computer methods in applied mechanics and engineering, 2014, 280: 28-56.

[22] PAPARI G, PETKOV N. Edge and line oriented contour detection: state of the art[J]. Image and vision computing, 2011, 29(2-3): 79-103.

[23] ROUHANI M, SAPPA A D. Implicit polynomial representation through a fast fitting error estimation[J]. IEEE transactions on Image Processing, 2011, 21(4): 2089-2098.