Fitting objects with implicit polynomials by deep neural network*

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

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

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

3. School of Integrated Circuits, University of Chinese Academy of Sciences, Beijing 100049, China

4. Shenzhen DAPU Microelectronics Co., Ltd., Shenzhen 518116, China

(Received 18 April 2022; Revised 23 August 2022)

Implicit polynomials (IPs) are considered as a powerful tool for object curve fitting tasks due to their simplicity and fewer parameters. The traditional linear methods, such as 3L, MinVar, and MinMax, often achieve good performances in fitting simple objects, but usually work poorly or even fail to obtain closed curves of complex object contours. To handle the complex fitting issues, taking the advantages of deep neural networks, we designed a neural network model continuity-sparsity constrained network (CSC-Net) with encoder and decoder structure to learn the coefficients of IPs. Further, the continuity constraint is added to ensure the obtained curves are closed, and the sparseness constraint is added to reduce the spurious zero sets of the fitted curves. The experimental results show that better performances have been obtained on both simple and complex object fitting tasks.

Document code: A **Article ID:** 1673-1905(2023)01-0060-5

DOI https://doi.org/10.1007/s11801-023-2065-6

Implicit polynomials (IPs) are particularly suitable for vision tasks like fast shape registration[1-3], image compression[4], recognition[5,6], smoothing and denoising[7,8], etc. Compared with B-splines[9-11], the advantages of IPs include fewer parameters, algebraic/geometric invariants, and robustness against noise and occlusion[12]. And Ref.[7] further shows IPs have the capability of recovering object shape with missing data in three-dimensional (3D) surface modeling.

To obtain the proper coefficients of the polynomial, linear methods, like 3L[1], gradient-one[8], MinMax, MinVar[13], are proposed in the framework of least square: minimizing the mean square error (*MSE*) between the predicted value and 0. Through matrix operation, the above methods can fit simple objects well with a polynomial. However, when handling more complex objects which have more inflection points, the fitting results are not good enough. This shows that the traditional linear methods lack the ability to handle complex objects. Therefore, a new method is considered to be proposed to solve this problem.

With the successful applications in computer vision[14] and natural language process[15], the structure of the encoder-decoder[16] has received a lot of attention for its strong fitting ability. Taking the advantage of the fitting ability of the encoder-decoder, Encoder-X[17] was proposed to learn the coefficients with deep model and achieved better performance. Along its way, a neural encoder net continuity-sparsity constrained network (CSC-Net) is designed to receive the input of ordered data points (zero sets) and output the coefficients of the polynomial. Then, the decoder takes the coefficients as input to revert the implicit polynomial. The obtained zero sets are the fitting result of the object. Different from Enocder-X which uses data augmentation to constrain the search space, we employ continuous constrain to obtain better fitting results. Furthermore, to make the fitting results more smoothly, sparse techniques are used for implicit polynomial fitting. And our contributions can conclude as follows. An encoder-decoder network CSC-Net is designed to find the best coefficients of a polynomial to fit the objects. Prior knowledge like continuity is added as constraining into the network to make the model find closed curves. The sparse technique is used to make the fitting curve as smooth as possible. The capability of IPs of recovering objects from missing data in two-dimensional (2D) space is studied.

Formally, for a 2D object, the algebraic curve represented by a 2D IP with degree n is given by

* This work has been supported by the Key-Area Research and Development Program of Guangdong Province (No.2019B010107001), and the Fund of Guangdong Support Program (No.2019TY05X071).

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

$$f_n(x,y)=\sum_{\substack{0\le i,j\le n\\0\le i+j\le n}} a_{ij}x^i y^j =$$
$$a_{00}+a_{10}x+a_{01}y+...+a_{ij}x^i y^j+...+a_{0n}y^n. \quad (1)$$

To make the representation more concisely, we use matrix to represent the equation as

$$f_n(x,y)=\boldsymbol{M}^{\mathrm{T}}\boldsymbol{A}=0, \quad (2)$$

where

$$\boldsymbol{M}=[1,x,y,xy,...,y^n]^{\mathrm{T}}, \quad (3)$$

$$\boldsymbol{A}=[a_{00},a_{10},a_{01},a_{11},...,a_{0n}]^{\mathrm{T}}. \quad (4)$$

Matrix $\boldsymbol{M}$ is the monomial matrix, which has $p=(n+1)(n+2)/2$ terms. And $\boldsymbol{A}$ is the coefficient vector which also has p terms.

The classical and simplest way to fit an algebraic curve to data is to minimize the algebraic distance over the set of given data points as

$$e=\sum_{1<k<m} f_n(x_k,y_k)^2=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{A}, \quad (5)$$

where m is the data number, and $\boldsymbol{S}=\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}$ is the scatter matrix of the monomials. The solution of Eq.(2) is the unit eigenvector $\boldsymbol{A}$ with smallest eigenvalue of $\boldsymbol{S}\boldsymbol{A}=\lambda\boldsymbol{A}$. But the fitting results of this method always produce the polynomial with discontinuous curves. This is because the optimized distance is not the exact distance between the data points and the curves. Although there have many other distance functions, algebraic distance is used for computation efficiency.

To improve the fitting performance, different constraints are added into the optimized equation by processing scatter matrix $\boldsymbol{S}$. The 3L[1] method adds two more curves to constrain the 3D surface to go through the additional curves to make the surface as steep as possible around the zero set, thus improving the stability and fitting performance. Different from 3L, the gradient-one[8] method achieves comparable results through constraining the continuity of the fitting polynomial curves. Specifically, to make the directional derivatives parallel to the tangent and the gradients parallel to the normal, furthermore, to improve the coefficient stability, MinMax and MinVar[13] methods are proposed by regularizing the scatter matrix.

For more elegant capability, RR[8] method is proposed to handle the problem of spurious zero sets by forcing the variables that do not contribute significantly to the fit attain values as close to zero as possible. QR decomposition[18] is used to adaptively obtain the best degree of polynomial. And our work focuses on improving the fitting performance of more complex objects by the deep neural network. Inspired by Encoder-X[17], to make full use of deep neural network to obtain better fitting results, our work uses the continuity and sparse constraints to constrain the search space. Different from us, Encoder-X uses data augmentation to constrain the search space. However, this will increase the model size and when handling with complex objects, the results are not ideal.

The whole framework of our method is shown in Fig.1. The CSC-Net is composed of encoder and decoder. The objects that need to be fitted are processed with a contour tracking algorithm to obtain ordered data. This process is to obtain the tangent vector and norm vector of the zero set points because tangent and norm vectors will be used in the decoder. Then the encoder encodes input data to coefficients, the decoder receives coefficients, tangent vector, and norm vectors to recover polynomial and compute loss. Finally, after the model update is finished, the fitting curve is plotted with learned coefficients.

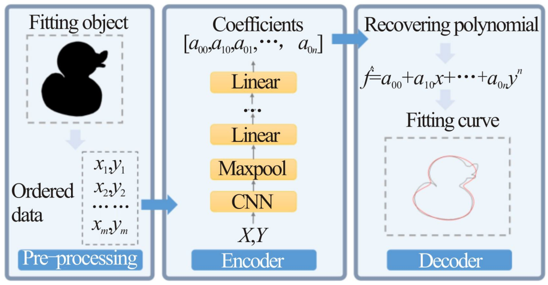

Fig.1 Framework of our method

For the input of the encoder has two dimensions, and also for the future extension in fitting 3D objects, we set the first layer of the encoder as a convolutional layer. Compared with the fully-connected layer, the convolutional layer can reduce parameter size and also has a faster training speed. The shape of the last layer is the number of coefficients with degree n .

Formally, the input of encoder is denoted as follows

$$\boldsymbol{input}=[X,Y],\boldsymbol{input}\in R^{m\times 2}. \quad (6)$$

And the encoder is denoted as Encoder, it is a multi-layer neural network, and the output of the Encoder is calculated by

$$\hat{\boldsymbol{A}}=\mathrm{Encoder}(\boldsymbol{input}),\hat{\boldsymbol{A}}\in R^p, \quad (7)$$

where $\hat{\boldsymbol{A}}$ is the predicted coefficients of the fitted polynomial.

The decoder receives the output of encoder to revert the polynomial as

$$\hat{\boldsymbol{A}}=[\hat{a}_{00},\hat{a}_{10},\hat{a}_{01},...,\hat{a}_{0n}], \quad (8)$$

$$\hat{f}_n(x,y)=\hat{a}_{00}+\hat{a}_{10}x+\hat{a}_{01}y+...+\hat{a}_{0n}y^n. \quad (9)$$

The reverted polynomial has the form of Eq.(9). $\hat{a}_{ij}$ is the predicted coefficient of the monomial $x^i y^j$.

Same as the traditional linear method, our method uses the algebraic distance as training goal. Formally, with the predicted value and the true value, the training object is to minimize the mean square loss as

$$MSE=\frac{1}{m}\sum_{i=1}^{m}\hat{f}_n(x_i,y_i)^2. \quad (10)$$

This loss has no constrain in search space, and only requires the value of the pixel points to be close to zero. This will cause the problem of discontinuity which makes the curve is not a closure. To make the curve as a closed curve, we add the constrain of continuity as[8]

$$C_1=\frac{1}{m}\sum_{i=1}^{m}\frac{f_x N_x+f_y N_y}{\sqrt{f_x^2+f_y^2}}, \tag{11}$$

where f_x is the partial derivative in the x-axis, and f_y is the partial derivative in the y-axis. And N_x and N_y are the norm vectors of the object. C_1 is the mean cosine distance of the norm vector and the gradient vector, which should be close to 1. The norm vectors are parallel to gradient vectors.

Furthermore, to reduce the number of spurious zero sets, we use sparse technique to make the coefficients which span small spaces to be close to zero:

$$C_2=\frac{1}{p}\sum_{i=1}^{p}|\hat{A}_i|. \tag{12}$$

Finally, our loss function is the weighted sum of three parts as

$$loss=MSE+\mu|C_1-1|+\lambda C_2, \tag{13}$$

where μ and λ are penalty factors.

IPs had been used to fit simple objects like butterfly, boot, etc[1,8,13]. We select butterfly, boot, airplane as basic simple object as a part of our dataset. They are shown in Fig.2. Besides, to access the stability of traditional method, we add another object bear into the dataset.

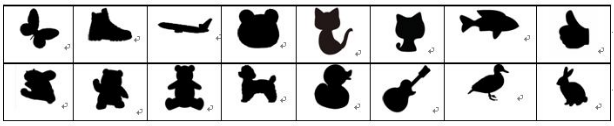

Fig.2 Dataset of fitting objects

Finally, to show the capability of fitting complex objects which have more bending points of our model, we add duck, rabbit, bear1, etc to the dataset. They are also shown in Fig.2.

We use the Adam as optimizer, and set the learning rate as 0.01, μ=0.1 and λ=0.5. To compare the performance of CSC-Net with others, we choose the well-performed methods 3L, MinMax, MinVar, and Encoder-X as baselines. For each object, the same maximum degree n is used in different methods.

To show the effectiveness of constrains added by CSC-Net, we conduct ablation study on all the objects. From Fig.3 we can see that making the predicted polynomial value close to zero alone is not enough to obtain ideal fitting curves.

Except the simple objects like butterfly, boot, airplane, and bear, the obtained polynomial cannot make the curve be a closure. The continuity constraint C_1 can make the curve be a closed one. Because the model requires the directional derivative of the polynomial to be perpendicular to the tangent and parallel to the norm vector of the true curves. The obtained curves need to be a closed curve to reach a low loss. Experimental results show the effectiveness of C_1 constraint.

Furthermore, C_2 reduces the spurious zero sets by making the coefficient as sparse as possible to decrease the small space spanned by coefficients. This is useful because when dealing with complex objects like rabbit, duck, etc, it has to use higher maximum degree n to fit the object, thus the number of coefficients expand rapidly and has more spanned small space which is not needed.

The proposed model CSC-Net is compared with the methods of 3L, MinMax, MinVar, and Encoder-X. The former three are traditional methods, while Encoder-X and CSC-Net are neural network-based methods. The difference between Encoder-X and CSC-Net is the process of constraint. In Encoder-X, constraints are added through data augmentation: scaling the curve in and out, and making the value as positive number and negative number inside and outside. This constraint is treated more like 3L. However, in CSC-Net, the constraints are added by directional derivatives and normals.

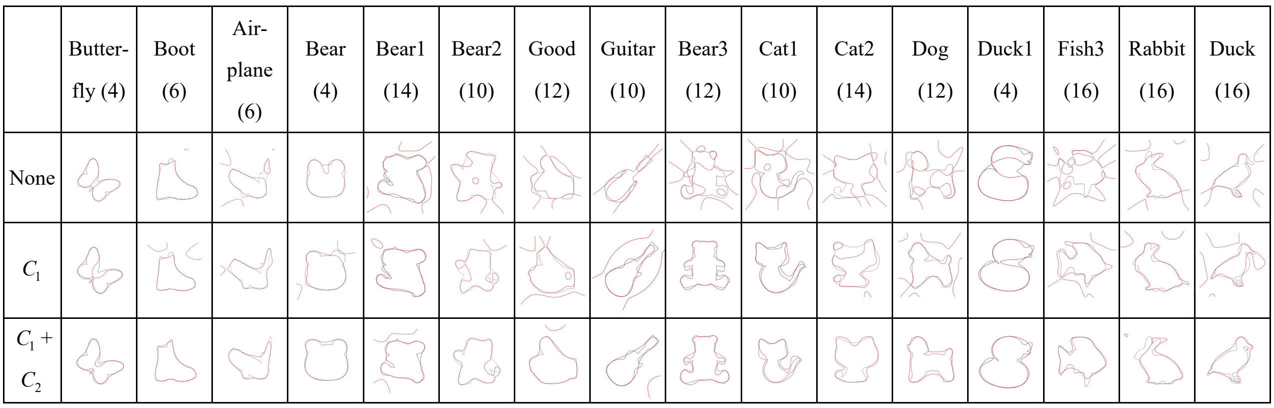

Fig.3 Ablation study

The fitting results are shown in Fig.4. From Fig.4 we can find that neural network-based methods have more stability than the traditional methods. While in simple objects like butterfly, boot, airplane, bear, and duck1, all the methods can obtain good fitting results. However, the fitting results get worse when dealing with complex objects. The traditional methods tend to fit the local optimal, while losing the relatively better global best. Such characteristic can be shown in some objects. The MinMax fits every detail of bear1, but in most times, it cannot obtain the global optimal thus prioritizes local optimal, and the fitted curve fits part of the curve very well. The 3L and MinVar are the same.

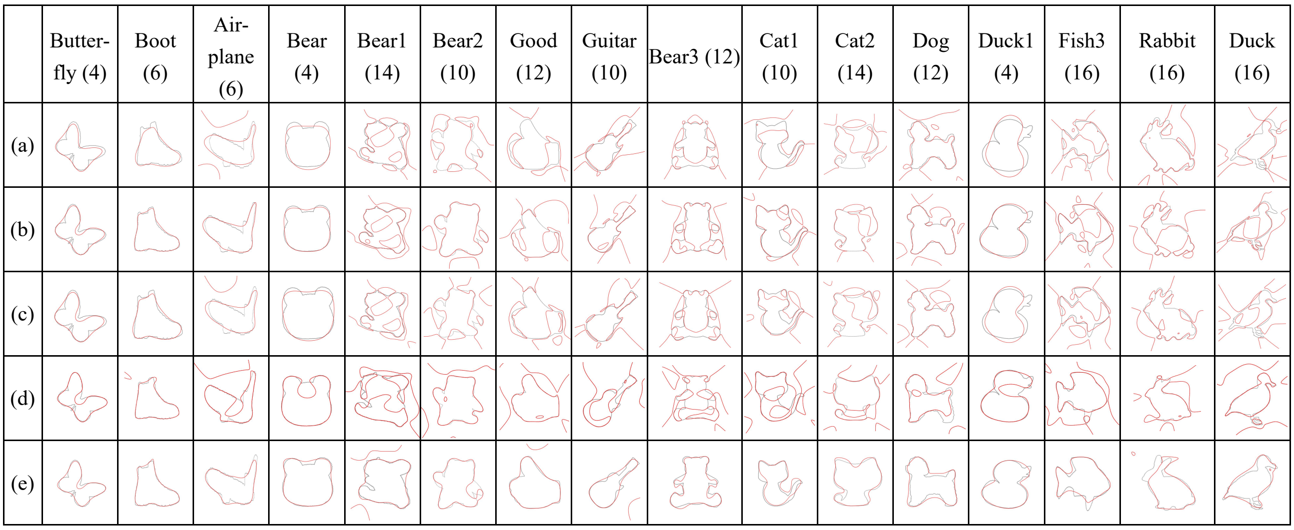

Fig.4 Comparison with other methods: (a) 3L; (b) MinMax; (c) MinVar; (d) Encoder-X; (e) CSC-Net

As deep model-based methods, Encoder-X and CSC-Net can both obtain better results than traditional methods. Especially in fitting objects like rabbit and duck, the fitting results surpass others significantly. However, in fitting objects like airplane, bear, bear1, etc, Encoder-X has limited performance. This illustrates that the continuity constraint is effective and has better practical performance.

Judging from the number of spurious zero sets, 3L, MinMax, MinVar and Encoder-X have much more unnecessary curves. This shows that sparse constraint can reduce the number of spurious zero sets by making the coefficients of polynomial as sparse as possible. Because unnecessary coefficients with small values have a big impact on the reverted curves.

There have been some works on showing the inference ability to recover the contour with missing data in 3D but the same analysis is lacked in 2D. Thus, we run the related experiments to see if IPs have the same ability in 2D.

We manually removed a certain degree of contour data, and specifically, 10% data points are removed. From Fig.5, we can see that the IPs can infer the contour of the missing part. In all 16 objects, the learned IPs are a good complement to the missing parts. This experiment results show IPs also have the ability to infer the missing part of the objects in 2D object.

However, from Fig.5 we can also see that IPs have limited inference ability and the missing part will influence the fitted precision. And it is easier for IPs if the missing part is smoothy.

When evaluating the space complexity of deep model, it usually uses parameter size and *FLOPs*. The smaller the parameter size and *FLOPs* are, the better the performance is.

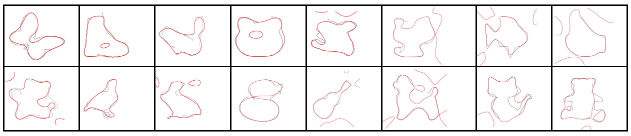

Fig.5 Fitting results of missing data

In Tab.1, (a) represents the model of Encoder-X, and (b) represents the CSC-Net. From Tab.1 we can find that the parameter size and *FLOPs* are significantly smaller than those of Encoder-X. In average, CSC-Net is 2.65% of Encoder-X.

Tab.1 Space complexity comparison

	Parameter size		*FLOPs*	
	(a)	(b)	(a)	(b)
Airplane (6)	67.80M	**1.64M**	67.77M	**1.65M**
Bear (4)	81.91M	**2.24M**	81.89M	**2.26M**
Bear1 (14)	90.60M	**2.63M**	90.58M	**2.65M**
Bear2 (10)	59.42M	**1.29M**	59.40M	**1.30M**
Bear3 (12)	87.31M	**2.48M**	87.29M	**2.50M**
Boot (6)	75.69M	**1.98M**	75.67M	**1.99M**
Butterfly (4)	74.97M	**1.95M**	74.95M	**1.96M**
Cat1 (10)	102.07M	**3.11M**	102.05M	**3.13M**
Cat2 (14)	69.65M	**1.73M**	69.63M	**1.74M**
Dog (12)	79.85M	**2.17M**	79.83M	**2.18M**
Duck (16)	72.22M	**1.85M**	72.20M	**1.86M**
Duck1 (4)	87.36M	**2.48M**	87.33M	**2.49M**
Fish3 (16)	80.50M	**2.20M**	80.48M	**2.22M**
Good (12)	66.05M	**1.58M**	66.03M	**1.59M**
Guitar (10)	75.24M	**1.97M**	75.22M	**1.98M**
Rabbit (16)	64.95M	**1.54M**	64.93M	**1.54M**
Average	77.22M	**2.05M**	77.20M	**2.07M**

Tab.2 lists the time cost of Encoder-X and CSC-Net. Compared with space complexity, the time cost between

CSC-Net and Encoder-X is not conspicuous. For computing the derivative of predicted curve, CSC-Net needs more time to update the network. Overall, CSC-Net can achieve better performance than Encoder-X in smaller time and space complexity.

Tab.2 Time complexity comparison

Object	(a)	(b)	Object	(a)	(b)
Airplane	106.1 s	**79.2 s**	Cat2	228.0 s	**188.9 s**
Bear	108.5 s	**76.1 s**	Dog	164.1 s	**163.3 s**
Bear1	**192.7 s**	224.7 s	Duck	306.7 s	**241.1 s**
Bear2	131.1 s	**107.7 s**	Duck1	152.5 s	**65.4 s**
Bear3	168.5 s	**166.1 s**	Fish3	325.5 s	**248.9 s**
Boot	130.1 s	**78.0 s**	Good	154.5 s	**152.4 s**
Butterfly	143.2 s	**61.3 s**	Guitar	138.9 s	**128.0 s**
Cat1	213.6 s	**166.2 s**	Rabbit	**193.4 s**	230.5 s

In this paper, to enhance the fitting performance of IPs in complex objects, we propose a neural network CSC-Net in an encoder-decoder structure to achieve that goal. Although our model has good performance, the maximum degree used by each polynomial is high, thus it needs sparse constraint to obtain better results. However, fractional polynomial can have much more expressive ability when the maximum degree is the same with polynomial. In future, we will consider to transfer this model into fractional polynomial to obtain better performance.

Statements and Declarations

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

References

[1] BLANE 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 & machine intelligence, 2000, 22：298-313.

[2] FORSYTH D, MUNDY J, ZISSERMAN A, et al. Invariant descriptors for 3D object recognition and pose[J]. IEEE transactions on pattern analysis & machine intelligence, 1991, 13(10)：971-991.

[3] TAREL J P, COOPWE D B. The complex representation of algebraic curves and its simple exploitation for pose estimation and invariant recognition[J]. IEEE transactions on pattern analysis & machine intelligence, 2000, 22(7)：663-674.

[4] HELZER A, BARZOHAR M, MALAH D. Using implicit polynomials for image compression[C]//Proceedings of 21st IEEE Convention of the Electrical & Electronic Engineers in Israel, April 11-12, 2000, Tel-Aviv, Israel. New York：IEEE, 2000：384-388.

[5] KEREN D. Topologically faithful fitting of simple closed curves[J]. IEEE transactions on pattern analysis and machine intelligence, 2004, 26(1)：118-123.

[6] MAROLA G. A technique for finding the symmetry axes of implicit polynomial curves under perspective projection[J]. IEEE transactions on pattern analysis and machine intelligence, 2005, 27(3)：465-470.

[7] SAHIN T, UNEL M. Fitting globally stabilized algebraic surfaces to range data[C]//10th IEEE International Conference on Computer Vision (ICCV'05), October 17-21, 2005, Beijing, China. New York：IEEE, 2005, 2：1083-1088.

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

[9] ROUHANI M, SAPPA A D, BOYER E. Implicit B-spline surface reconstruction[J]. IEEE transactions on image processing, 2015, 24(1)：22-32.

[10] MAJEED A, ABBAS M, QAYYUM F, et al. Geometric modeling using new cubic trigonometric B-spline functions with shape parameter[J]. Mathematics, 2020, 8：2102.

[11] PATRIZI F, DOKKEN T. Linear dependence of bivariate minimal support and locally refined B-splines over LR-meshes[J]. Computer aided geometric design, 2022, 77：101803.

[12] WU G, ZHANG Y C. A novel fractional implicit polynominal approach for stable representation of complex shapes[J]. Journal of mathematical imaging and vision, 2016, 55：89-104.

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

[14] VOULODIMOS A, DOULAMIS N, DOULAMIS A, et al. Deep learning for computer vision：a brief review[J]. Computational intelligence and neuroscience, 2018, 2018：1-13.

[15] DEVLIN J, CHANG M W, LEE K, et al. BERT：pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the As-sociation for Computational Linguistics, June, 2019, Minneapolis, Minnesota, USA. Stroudsburg：Association for Computational Linguistics, 2019：4171-4186.

[16] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//31st Conference on Neural Information Processing Systems (NIPS 2017), December 4-9, 2017, Long Beach, CA, USA. New York：Curran Associates Inc, 2017.

[17] WANG G J, LI W J, ZHANG L P, et al. Encoder-X：solving unknown coefficients automatically in polynomial fitting by using an autoencoder[J]. IEEE transactions on networks and learning systems, 2022, 33(8)：3264-3276.

[18] ZHENG B, TAKAMATSU J, IKEUCHI K. An adaptive and stable method for fitting implicit polynomial curves and surfaces[J]. IEEE transactions on pattern analysis & machine intelligence, 2010, 32(3)：561-568.