

Curvilinear Mask Optimization for Inverse Lithography Based on B-splines and Constrained Delaunay Triangulation

Xiaoru Yi and Junqing Chen*

*Department of Mathematical Sciences, Tsinghua University,
Beijing 100084, China.*

Received 9 April 2025; Accepted 10 September 2025

Abstract. In this paper, we propose a gradient-based method to optimize curvilinear masks in optical lithography. The mask pattern is represented by periodic B-spline curves. We apply constrained Delaunay triangulation to discretize the domains circled by the spline curves. Subsequently, we establish an explicit relationship between the integral points and the control points of the boundary spline curve. Based on the relationship, we derive explicit formulas of the gradient of the optimization objective function with respect to the coordinates of the control points. Then we propose an inverse lithography algorithm to optimize the curvilinear mask pattern. Finally, the results of the numerical experiments demonstrate the feasibility and extensive adaptability of our method.

AMS subject classifications: 78M50, 65D17, 65D07

Key words: Curvilinear masks, periodic B-spline curve, inverse lithography techniques, gradient-based optimization, optical proximity correction.

1 Introduction

As a crucial part of semiconductor manufacturing, projection lithography technology is tasked with transferring the patterns on the photomask onto the wafer coated with photoresist. During this process, patterns with critical size several times the lithographic wavelength generally do not require any correction. However, as the critical size decreases to be comparable or even smaller than the wavelength of the lithography light

*Corresponding author. *Email addresses:* jqchen@tsinghua.edu.cn (J. Chen), yixr23@mails.tsinghua.edu.cn (X. Yi)

source, owing to the diffraction and interference effects of adjacent patterns, the patterns on the wafer will experience varying degrees of distortion, namely the so-called optical proximity effect (OPE).

In order to overcome the distortion from the perspective of the mask pattern, the optical proximity correction (OPC) technology has been proposed. By making certain corrections to the mask pattern, the pattern obtained on the wafer can be made to more closely resemble the designed pattern. In the early days, most OPC technologies are rule-based OPC. Methods of this kind set the pattern correction rules based on the operational experience of lithography engineers to correct simple mask patterns. But as the size of the mask pattern further decreases, the proximity effect becomes more severe and the correction rules for the mask pattern increase exponentially, even becoming unfeasible. To overcome this difficulty, Rieger and Stirniman [34], and Cobb [13] first proposed the model-based OPC method. The earliest OPC method was based on the Manhattan mask, which only makes fine-tuning of the segmented boundaries along specific angles, so the flexibility of this approach is relatively limited.

To extend the flexibility of OPC and obtain a better mask pattern, pixel-based inverse lithography technology (ILT) was proposed. This method is highly flexible and enables more precise adjustments to the photomask, thus resulting in a superior mask quality. Saleh and Sayegh [35] explored optimized photomasks by “pixel flipping” in 1981. Liu and Zakhor [25] used branch and bound and the simplex method and the “bacteria” algorithm [26] to solve the problem. Granik [18] considered an objective function with linear, quadratic, and nonlinear formulations. Ma and his collaborators extended the pixel-based optimization method from the scalar to the vector imaging model [29, 30] and introduced the source and mask optimization (SMO) in [28]. However, this method has some inherent drawbacks. For example, it introduces a large number of optimization variables, and the optimized mask pattern is relatively fragmented and difficult to manufacture in practice. Some regularization methods have been proposed to constrain the mask pattern. Choy *et al.* [12] developed a robust computational algorithm for the loss function with multiple regularization terms. Chen and Liu [8] applied the alternating direction method of multipliers (ADMM) method to the inverse lithography and performed convergence analysis. A fuller review of inverse lithography techniques could be found in [7, 31].

Variable shaped beam (VSB) mask writers are widely used in mask manufacturing. Since they cannot handle complex mask patterns, the vast majority of mask designs have to be simplified to the Manhattan pattern for production. This is also the reason why curvilinear masks were not highly regarded in the early days. With the introduction of multi-beam mask writers, it has become possible to produce curvilinear masks. Pang and Fujimura [32] reviewed the trend and benefits of OPC methods based on curvilinear mask. In recent years, many OPC methods based on curvilinear masks have emerged. For example, Chen *et al.* [9] extended traditional edge-based OPC to vertices and produced curvilinear OPC results by following the conventional OPC workflow. Huang *et al.* [21] proposed distance-versus-angle signature (DVAS), a one-dimensional function, to

represent a two-dimensional boundary of the mask and optimized the shape. Yang *et al.* [40, 41] proposed a curvilinear OPC method adopting the quasi-uniform B-spline curve and non-uniform B-spline curve to improve the optimization efficiency. Wei *et al.* [38] established an optimization method based on an implicit function to reduce the number of variables. Other attempts on the optimization of curvilinear mask can be found in [2–4, 19, 20]. Machine learning models are also applied for accurate curve correction [6, 23, 37]. For further information about the curvilinear mask, we recommend reading [11, 33].

In this paper, we propose an inverse lithography method based on periodic B-splines to construct the mask boundary. Compared with traditional pixel-based inverse lithography techniques, this approach can greatly reduce the number of optimization variables. Moreover, the spline curve itself is one of the standard formats in computer-aided design (CAD) [1], and the optimized mask shape has good continuity and high manufacturability. We use the spline boundary to generate a triangular mesh by constrained Delaunay triangulation (CDT) and establish an explicit relationship between the Gaussian integral points and the control points. Based on the photoresist imaging model, we establish an explicit gradient formula for optimization of control points, which can be generalized to a variety of gradient-based optimization algorithms. We also provide some numerical examples to verify the feasibility of our method.

The structure of this paper is as follows. In Section 2, the basic knowledge about projection lithography and periodic B-splines required in this paper is introduced. In Section 3, the numerical calculation formula for the forward lithography problem is presented, an explicit relationship between the integration points and the control points is established, and some important error bounds are estimated. Subsequently, based on the photoresist imaging model, an inverse lithography problem model is given and an explicit gradient formula is derived. In Section 4, a basic description of the optimization algorithm is provided. In Section 5, some numerical examples are given to verify the reliability of our method.

2 Preliminary

2.1 Basic theory of optical lithography

Projection microlithography is the key to the semiconductor production. Normally, the imaging system implements Köhler's illumination, details of which can be found in [5]. A basic representation of an optical lithography imaging system is shown in Fig. 1. When light is emitted from the source, it is condensed by the condenser lens L_c and becomes parallel light. The parallel light illuminates the mask uniformly and carries the information of mask pattern. Then the light is modulated by the projection lens, finally converges on the image plane, and prints the mask pattern onto the wafer. In the procedure, the light will be limited by the pupil size and lose some pattern information of high frequency. The light will be scattered by the mask pattern as well, then the image on the

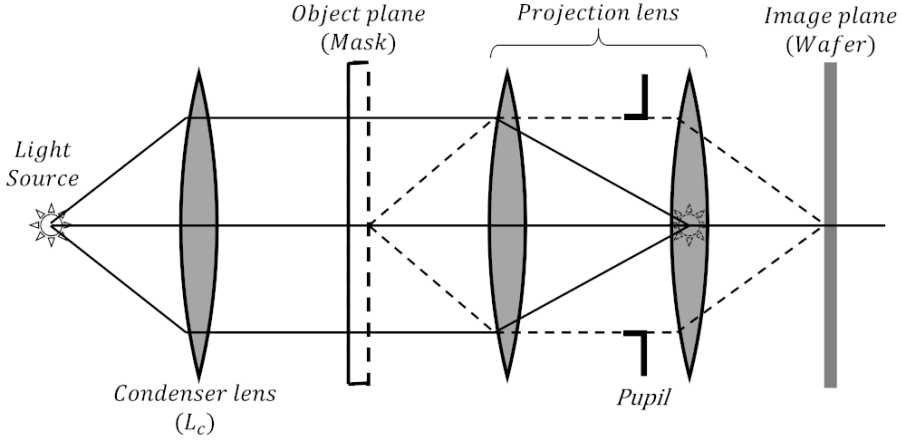


Figure 1: Basic representation of an optical lithography imaging system.

wafer is generally different from that on the mask, so we need to take some measures to correct the pattern on the mask.

According to the work of [39], the image intensity for an isoplanatic region under Köhler illumination is

$$\hat{I}(\hat{x}_i, \hat{y}_i) = \iiint_{-\infty}^{+\infty} TCC(\hat{f}', \hat{g}'; \hat{f}'', \hat{g}'') \tilde{O}(\hat{f}', \hat{g}') \tilde{O}^*(\hat{f}'', \hat{g}'') e^{-i2\pi[(\hat{f}' - \hat{f}'')\hat{x}_i + (\hat{g}' - \hat{g}'')\hat{y}_i]} d\hat{f}' d\hat{g}' d\hat{f}'' d\hat{g}'',$$

where $TCC(\hat{f}', \hat{g}'; \hat{f}'', \hat{g}'')$ is the transmission cross-coefficient

$$TCC(\hat{f}', \hat{g}'; \hat{f}'', \hat{g}'') = \iint_{-\infty}^{+\infty} \tilde{J}(\hat{f}, \hat{g}) \tilde{H}(\hat{f} + \hat{f}', \hat{g} + \hat{g}') \tilde{H}^*(\hat{f} + \hat{f}'', \hat{g} + \hat{g}'') d\hat{f} d\hat{g},$$

here \tilde{H}^* represents the conjugate of \tilde{H} and $\tilde{O}, \tilde{J}, \tilde{H}$ represent the Fourier integral of $\hat{O}, \hat{J}, \hat{H}$

$$\hat{O}(\hat{x}_o, \hat{y}_o) = \iint_{-\infty}^{+\infty} \tilde{O}(\hat{f}, \hat{g}) e^{-i2\pi(\hat{f}\hat{x}_o + \hat{g}\hat{y}_o)} d\hat{f} d\hat{g},$$

$$\hat{J}(\hat{x}'_o - \hat{x}''_o, \hat{y}'_o - \hat{y}''_o) = \iint_{-\infty}^{+\infty} \tilde{J}(\hat{f}, \hat{g}) e^{-i2\pi(\hat{f}(\hat{x}'_o - \hat{x}''_o) + \hat{g}(\hat{y}'_o - \hat{y}''_o))} d\hat{f} d\hat{g},$$

$$\hat{H}(\hat{x}_i - \hat{x}'_o, \hat{y}_i - \hat{y}'_o) = \iint_{-\infty}^{+\infty} \tilde{H}(\hat{f} + \hat{f}', \hat{g} + \hat{g}') e^{-i2\pi((\hat{f} + \hat{f}')(\hat{x}_i - \hat{x}'_o) + (\hat{g} + \hat{g}')(\hat{y}_i - \hat{y}'_o))} d\hat{x}'_o d\hat{y}'_o.$$

The function with a $\hat{\cdot}$ means that the variables in the function are normalized. The normalized variables that appear above are listed below

$$\hat{x}_o = -\frac{Mx_o}{\lambda_0/\text{NA}}, \quad \hat{y}_o = -\frac{My_o}{\lambda_0/\text{NA}}, \quad (2.1a)$$

$$\hat{x}_i = \frac{x_i}{\lambda_0/\text{NA}}, \quad \hat{y}_i = \frac{y_i}{\lambda_0/\text{NA}}, \quad (2.1b)$$

$$\hat{f} = \frac{f}{\text{NA}/\lambda_0}, \quad \hat{g} = \frac{g}{\text{NA}/\lambda_0}. \quad (2.1c)$$

The corresponding inverse normalization is

$$\begin{aligned} x_o &= -\frac{\lambda_0 \hat{x}_o}{M\text{NA}}, & y_o &= -\frac{\lambda_0 \hat{y}_o}{M\text{NA}}, \\ x_i &= \frac{\lambda_0 \hat{x}_i}{\text{NA}}, & y_i &= \frac{\lambda_0 \hat{y}_i}{\text{NA}}, \\ f &= \frac{\hat{f}\text{NA}}{\lambda_0}, & g &= \frac{\hat{g}\text{NA}}{\lambda_0}. \end{aligned} \quad (2.2)$$

Here (x_o, y_o) represent the original spatial coordinates on the object plane, (\hat{x}_o, \hat{y}_o) represent the normalized spatial coordinates on the object plane. (x_i, y_i) and (\hat{x}_i, \hat{y}_i) represent the original spatial coordinates and its normalization on the image plane. f, g represent spatial frequencies and \hat{f}, \hat{g} are normalized spatial frequencies. M is the lateral magnification, λ_0 is the wave length and NA is the numerical aperture of the imaging system. $\hat{I}(\hat{x}_i, \hat{y}_i)$ is the intensity at a point (\hat{x}_i, \hat{y}_i) on the image plane. $\hat{J}(\hat{x}'_o - \hat{x}''_o, \hat{y}'_o - \hat{y}''_o)$ is the mutual intensity between a pair of illuminated points (\hat{x}'_o, \hat{y}'_o) and $(\hat{x}''_o, \hat{y}''_o)$. $\hat{O}(\hat{x}'_o, \hat{y}'_o)$ is the function used to describe the mask pattern. Under the assumption that the imaging region is so small that it forms an isoplanatic region, $\hat{H}(\hat{x}_i - \hat{x}_o, \hat{y}_i - \hat{y}_o)$ is the normalized version of the optical system transfer function.

In this paper, we assume that the illumination is coherent and the effective source is a point at (\hat{f}_0, \hat{g}_0) , namely $\tilde{J}(\hat{f}, \hat{g}) = \delta(\hat{f} - \hat{f}_0, \hat{g} - \hat{g}_0)$, so that we can derive the coherent image intensity

$$\hat{I}(\hat{x}_i, \hat{y}_i) = |\hat{U}(\hat{x}_i, \hat{y}_i)|^2 \quad (2.3)$$

with

$$\hat{U}(\hat{x}_i, \hat{y}_i) = \iint_{-\infty}^{+\infty} \tilde{H}(\hat{f}, \hat{g}) \tilde{O}(\hat{f} - \hat{f}_0, \hat{g} - \hat{g}_0) e^{-i2\pi(\hat{f}\hat{x}_i + \hat{g}\hat{y}_i)} d\hat{f} d\hat{g},$$

where $\hat{U}(\hat{x}_i, \hat{y}_i)$ represents the amplitude of a point on the image plane under a normalized variable representation. The formula above implies

$$\tilde{U}(\hat{f}, \hat{g}) = \tilde{O}(\hat{f} - \hat{f}_0, \hat{g} - \hat{g}_0) \tilde{H}(\hat{f}, \hat{g}).$$

According to the convolution theorem, if we assume that $(\hat{f}_0, \hat{g}_0) = (0, 0)$, we can get the following formula:

$$\hat{U}(\hat{x}_i, \hat{y}_i) = \iint_{-\infty}^{+\infty} \hat{O}(\hat{x}_o, \hat{y}_o) \hat{H}(\hat{x}_i - \hat{x}_o, \hat{y}_i - \hat{y}_o) d\hat{x}_o d\hat{y}_o. \quad (2.4)$$

From [39, Eq. (4.10)], it is known that

$$\hat{H}(\hat{x}_i - \hat{x}_o, \hat{y}_i - \hat{y}_o) = \iint_{-\infty}^{+\infty} \hat{P}(\hat{f}, \hat{g}) e^{-i2\pi(\hat{f}(\hat{x}_i - \hat{x}_o) + \hat{g}(\hat{y}_i - \hat{y}_o))} d\hat{f} d\hat{g},$$

where $\hat{P}(\hat{f}, \hat{g})$ is the pupil function under the normalized coordinates, which is

$$\hat{P}(\hat{f}, \hat{g}) = \begin{cases} 1, & \text{if } \sqrt{\hat{f}^2 + \hat{g}^2} \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Here we define circ function in the following way:

$$\text{circ}\left(\sqrt{x^2 + y^2}\right) = \begin{cases} 1, & \text{if } \sqrt{x^2 + y^2} < 1, \\ 1/2, & \text{if } \sqrt{x^2 + y^2} = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Since the set in which the value of the circ function is 1/2 is a set of measure of zero, we can replace $\hat{P}(\hat{f}, \hat{g})$ with $\text{circ}(\sqrt{\hat{f}^2 + \hat{g}^2})$ without making any difference to the value of $\hat{H}(\hat{x}_i - \hat{x}_o, \hat{y}_i - \hat{y}_o)$, so we can get

$$\hat{H}(\hat{x}_i - \hat{x}_o, \hat{y}_i - \hat{y}_o) = \iint_{-\infty}^{+\infty} \text{circ}\left(\sqrt{\hat{f}^2 + \hat{g}^2}\right) e^{-i2\pi(\hat{f}(\hat{x}_i - \hat{x}_o) + \hat{g}(\hat{y}_i - \hat{y}_o))} d\hat{f} d\hat{g}.$$

This means that $\hat{H}(\hat{x}_i - \hat{x}_o, \hat{y}_i - \hat{y}_o)$ is in fact the standard Fourier transform of circ function. Here we give the result of transformation directly

$$\hat{H}(\hat{x}_i - \hat{x}_o, \hat{y}_i - \hat{y}_o) = \frac{J_1(2\pi\rho)}{\rho} = \frac{J_1\left(2\pi\sqrt{(\hat{x}_i - \hat{x}_o)^2 + (\hat{y}_i - \hat{y}_o)^2}\right)}{\sqrt{(\hat{x}_i - \hat{x}_o)^2 + (\hat{y}_i - \hat{y}_o)^2}}, \quad (2.5)$$

where J_1 is a first-order Bessel function of the first kind. For detailed derivation process, we recommend reading [17, Chapter 2.1].

2.2 Periodic B-spline curve boundary representation

In practical lithography for high-volume manufacturing, people always hope to find a relatively regular pattern which is easy to produce. Benefiting from the renewal of production equipment, it is possible to manufacture curvilinear masks whose pattern boundary consists of flexible curve instead of only axis-parallel rectangles or 45-degree triangles [11]. A natural assumption is that the boundary of the curvilinear mask pattern is continuous. We use Ω to represent the transparent part of the mask and the boundary of Ω is denoted by $\partial\Omega$. Moreover, since the transparent part of the mask is closed,

the boundary curve $\partial\Omega$ should be periodic. If we limit the parameter of the boundary curve into interval $[0,1]$, and consider that the transparent part of the mask is composed of several closed domains disconnected from each other, the boundary of curvilinear mask can be described as

$$\begin{aligned}\partial\Omega &= \{\mathbf{g}_1(t), \mathbf{g}_2(t), \dots, \mathbf{g}_n(t)\}, \\ \mathbf{g}_k(t) &= x_k(t)\mathbf{i} + y_k(t)\mathbf{j}, \quad k=1,2,\dots,n, \\ x_k(t), y_k(t) &\in C([0,1]), \quad x_k(0) = x_k(1), \quad y_k(0) = y_k(1).\end{aligned}$$

Here \mathbf{i}, \mathbf{j} are the unit vectors in x and y directions respectively.

In the followings, we will use the B-spline function to represent the curve boundary of Ω because of the powerful expressive ability of B-spline. Since the boundary is periodic, it is natural to use the periodic B-spline function. First, we will give some basic notions about the B-spline curve. For a better understanding of the theory of the spline function, we recommend reading [14, 15, 36].

B-spline curve is a kind of widely used polynomial spline curve, the basis of which is defined recursively:

Definition 2.1. Given a knot vector on the interval $[a, b]$ written $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, where $\xi_1 = a, \xi_{n+p+1} = b, \xi_k \leq \xi_{k+1}$, and $\xi_k \in \mathbb{R}$ is the k -th knot, $k=1, 2, \dots, n+p+1$, p is the polynomial degree, and n is the number of basis functions that comprise the B-spline. Then k -th B-spline basis function of degree p $N_{k,p}$ can be defined recursively according to the following de Boor-Cox formula for $p=0, 1, 2, \dots$:

$$\begin{cases} N_{k,0}(\xi) = \begin{cases} 1, & \xi \in [\xi_k, \xi_{k+1}), \\ 0, & \text{otherwise,} \end{cases} \\ N_{k,p}(\xi) = \frac{\xi - \xi_k}{\xi_{k+p} - \xi_k} N_{k,p-1}(\xi) + \frac{\xi_{k+p+1} - \xi}{\xi_{k+p+1} - \xi_{k+1}} N_{k+1,p-1}(\xi), \quad p \geq 1, \\ \text{define } \frac{0}{0} = 0. \end{cases}$$

With the definition of B-spline basis, we can further define the B-spline curves.

Definition 2.2. Given n space vectors $\{\mathbf{P}_k\}_{k=1}^n \in \mathbb{R}^d$, and assuming that $N_{k,p}(\xi)$ are the B-spline basis functions of degree p defined on the knot vector $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ ($\xi_k \leq \xi_{k+1}$, $k=1, 2, \dots, n+p$), we can define

$$\mathbf{P}(\xi) = \sum_{k=1}^n \mathbf{P}_k N_{k,p}(\xi), \quad \xi \in [\xi_p, \xi_{n+1}],$$

is the p -th order B-spline curve correspond to knot vector Ξ .

To generate a periodic B-spline curve, we first introduce the extended partition.

Definition 2.3. For a knot vector on the interval $[a, b]$ written $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, we can define its extended partition $\bar{\Xi}$ as follows:

$$\bar{\Xi} = \{\xi_{1-p}, \dots, \xi_1, \xi_2, \dots, \xi_{n+p+1}, \dots, \xi_{n+2p+1}\}.$$

Here we define

$$\xi_k = \begin{cases} \xi_{k+n+p} - L, & 1-p \leq k < 1, \\ \xi_{k-n-p} + L, & n+p+1 < k \leq n+2p+1, \end{cases}$$

where $L = b - a$.

Based on the extended partition in Definition 2.3, we can further define the periodic B-spline basis function by the following definition.

Definition 2.4. For a given extended partition $\bar{\Xi} = \{\xi_{1-p}, \dots, \xi_1, \xi_2, \dots, \xi_{n+p+1}, \dots, \xi_{n+2p+1}\}$ of the interval $[a, b]$, we can define the k -th periodic B-spline basis function of degree p by

$$\mathring{N}_{k,p}(\xi) = \begin{cases} N_{k,p}(\xi), & k = 1, 2, \dots, n, \\ N_{k,p}(\xi) + N_{k-n-p,p}(\xi), & k = n+1, n+2, \dots, n+p, \end{cases}$$

Although the definition we used here is a little different from the definition in [36], they are essentially the same. We can similarly define the periodic B-spline curve as follows.

Definition 2.5. Given $n+p$ space vectors $\{\mathbf{P}_k\}_{k=1}^{n+p} \in \mathbb{R}^d$, and assuming that $\mathring{N}_{k,p}(\xi)$ are the periodic B-spline basis functions of degree p defined on the extended partition

$$\bar{\Xi} = \{\xi_{1-p}, \dots, \xi_1, \xi_2, \dots, \xi_{n+p+1}, \dots, \xi_{n+2p+1}\}.$$

Then the periodic B-spline curve of degree p correspond to knot vector in $\bar{\Xi}$ is defined by

$$\mathring{\mathbf{P}}(\xi) = \sum_{k=1}^{n+p} \mathbf{P}_k \mathring{N}_{k,p}(\xi), \quad \xi \in [\xi_1, \xi_{n+p+1}].$$

And we call $\mathbf{P}_k, k = 1, 2, \dots, n+p$ the control points of periodic B-spline curve.

With the definition of periodic B-spline curve, we can use it to describe the boundary of mask pattern. [36, Theorem 8.12] guarantees that for any periodic continuous boundary curve, it can be approximated well by periodic B-spline curves. With the increasing of number of control points and the decreasing of $\bar{\Delta} = \max_{1 \leq k \leq n+p} \{\xi_{k+1} - \xi_k\}$, the distance between a continuous periodic function and its periodic B-spline approximation can be arbitrarily small, which means that it is reasonable to represent the boundary of mask pattern by periodic B-spline curve.

3 Forward and inverse lithography based on B-spline function

3.1 Region approximation and discretization

In this part, for the sake of simplicity, we omit the symbol $\hat{\cdot}$ which represents the normalization of function and variable, and rewrite the imaging formulas (2.3) and (2.4) as

$$U(x_i, y_i) = \iint_{-\infty}^{+\infty} O(x_o, y_o) H(x_i - x_o, y_i - y_o) dx_o dy_o, \quad I(x_i, y_i) = |U(x_i, y_i)|^2, \quad (3.1)$$

where $O(x_o, y_o)$ describes the pattern on the mask. In the following, we represent the pattern by binary function

$$O(x_o, y_o) = \begin{cases} 1, & \text{if location } (x_o, y_o) \text{ on the mask is transparent,} \\ 0, & \text{if location } (x_o, y_o) \text{ on the mask is not transparent.} \end{cases}$$

We denote the transparent part of the curvilinear mask by S_0 , whose boundary is represented by B-spline functions introduced in Section 2.2. Under the binary assumption about the mask pattern, the imaging formula (3.1) can be rewritten as

$$U(x_i, y_i) = \iint_{S_0} H(x_i - x_o, y_i - y_o) dx_o dy_o, \quad I(x_i, y_i) = |U(x_i, y_i)|^2. \quad (3.2)$$

To calculate the integral in domain S_0 , we have to approximate the domain by polygons and discretize the polygons by triangulation. The procedure is shown in Fig. 2, so that the formula (3.2) becomes

$$U(x_i, y_i) = \sum_{p=1}^{N_T} \iint_{S_p} H(x_i - x_o, y_i - y_o) dx_o dy_o, \quad I(x_i, y_i) = |U(x_i, y_i)|^2, \quad (3.3)$$

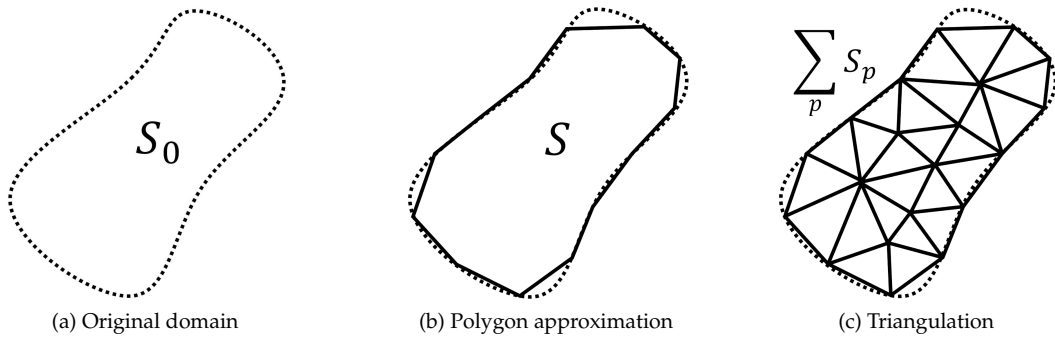


Figure 2: Procedure of polygonal approximation and triangulation.

where p is the index of small triangle, N_T is the total number of triangles, S_p is the p -th triangle whose measure is $|S_p|$. $\sum_{p=1}^{N_T} |S_p| = |S|$, which is the measure of the whole polygon. In the procedure of polygonal approximation and triangulation to the domain S_0 , the error is introduced by those triangles that have one edge adjacent to the curve boundary. We mark those triangles with S_1, S_2, \dots and the corresponding mismatching arch domain with $\Delta S_1, \Delta S_2, \dots$, as shown in Fig. 3.

To estimate the error of polygon approximation, we first introduce a lemma from [22].

Lemma 3.1. *Let $f(x)$ have an $(n+1)$ -th derivative, $f^{(n+1)}(x)$, in an interval $[a, b]$. Let $P_n(x)$ be the interpolation polynomial for $f(x)$ with respect to $n+1$ distinct points $x_k, k=0, 1, \dots, n$ in the interval $[a, b]$ (i.e. $P_n(x_k) = f(x_k)$ and $x_k \in [a, b]$). Then for each $x \in [a, b]$ there exists a point $\xi = \xi(x)$ in the open interval*

$$\min(x_0, x_1, \dots, x_n, x) < \xi < \max(x_0, x_1, \dots, x_n, x)$$

such that

$$f(x) - P_n(x) \equiv R_n(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(n+1)!} f^{(n+1)}(\xi).$$

As for the condition we need, we choose $n=1, x_0=a, x_n=b$, and immediately we can get

$$\begin{aligned} \min(a, b, x) < \xi < \max(a, b, x), \\ f(x) - P_1(x) = R_1(x) &= \frac{(x-a)(x-b)}{2} f''(\xi). \end{aligned}$$

Based on the error estimate of piecewise linear interpolation, in each small triangle on the boundary, we can express the curve boundary as $f_k(t) = x_k(t)\mathbf{i} + y_k(t)\mathbf{j}$, where each component function is $(p-1)$ -th order continuous since the boundary is represented by B-spline functions. The linear interpolation function $P_1^k(t) = P_{1x}^k(t)\mathbf{i} + P_{1y}^k(t)\mathbf{j}$ is used to describe the triangle edge adjacent to the curve boundary. Based on the Lemma 3.1, we give a theorem to estimate the error of discretization.

Theorem 3.1. *When we approximate the integral on the domain S_0*

$$U(x_i, y_i) = \iint_{S_0} H(x_i - x_o, y_i - y_o) dx_o dy_o, \tag{3.4}$$

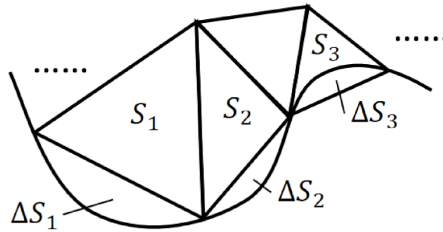


Figure 3: Error of polygon approximation.

by the sum of integral on each triangulation

$$\tilde{U}(x_i, y_i) = \sum_{p=1}^{N_T} \iint_{S_p} H(x_i - x_o, y_i - y_o) dx_o dy_o, \quad (3.5)$$

we have error estimate as follows:

$$|U(x_i, y_i) - \tilde{U}(x_i, y_i)| \leq C \max h_k^2 \rightarrow 0 \text{ as } \max h_k^2 \rightarrow 0,$$

where h_k represents the length of triangle edge adjacent to the curve boundary and C is a constant independent of h_k .

Proof. According to Lemma 3.1, we can get

$$\|f(x) - P_1(x)\|_\infty = \|R_1(x)\|_\infty \leq c_0 h^2,$$

where $c_0 = \max\{|f''(\xi)|\}/2$, h is the length of the interval for interpolation. For one arch domain on the boundary, we can use $f_k(t)$ to describe the curve boundary and $P_1^k(t)$ to describe the corresponding linear approximation. So that we have estimation

$$|\Delta S_k| \leq h_k d_k,$$

where ΔS_k represents the k -th arch domain, $|\Delta S_k|$ is the measure of ΔS_k , h_k is the length of the k -th linear interpolation edge on the boundary, and d_k is the height of the k -th arch. d_k can be bounded by the interpolation error estimate

$$d_k \leq \max_t \{|x_k(t) - P_{1x}^k(t)| + |y_k(t) - P_{1y}^k(t)|\} \leq c_1 h_k^2,$$

where c_1 is some constant independent of h_k . It depends on the maximum of $|x_k''(t)|$ and $|y_k''(t)|$. An illustration of these variables and the estimation is shown in Fig. 4.

So for the whole curve boundary, the upper bound for error of integral area can be written as

$$\sum_k |\Delta S_k| \leq \sum_k c_1 h_k^3. \quad (3.6)$$

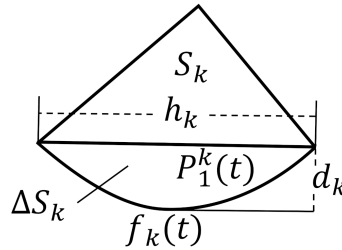


Figure 4: Error estimation in one arch.

Compare the two formulas (3.4)-(3.5), and consider that $\sum_k h_k$ can be controlled by the perimeter of the curve boundary C_{peri} , we have the following estimation:

$$\begin{aligned} |\Delta U(x_i, y_i)| &:= |U(x_i, y_i) - \tilde{U}(x_i, y_i)| \leq \sum_k \left| \iint_{\Delta S_k} H(x_i - x_o, y_i - y_o) dx_o dy_o \right| \\ &\leq \max |H(x_i - x_o, y_i - y_o)| \sum_k |\Delta S_k| \leq c_2 \sum_k h_k^3 \leq c_2 \max_k h_k^2 \sum_k h_k \\ &\leq c_2 C_{peri} \max h_k^2 := C \max h_k^2 \rightarrow 0 \quad \text{as} \quad \max h_k^2 \rightarrow 0. \end{aligned}$$

Here we use (3.6) in the second line of the above estimate and

$$c_2 = c_1 \max |H(x_i - x_o, y_i - y_o)|, \quad C = c_2 C_{peri}.$$

From the proof we can see that the constant C depends on the perimeter of polygon, maximum of the absolute value of function H and factor c_1 . \square

With more vertices added to the boundary of the spline boundary and refinement of triangulation, it is easy to ensure that $\max h_k^2 \rightarrow 0$, which means that the integral error caused by the polygonal approximation can be well controlled.

3.2 Numerical imaging formula

To numerically compute the integral on each triangle, we use the symmetric Gaussian quadrature rules provided in [16]. Then with the help of the quadrature rules, Eq. (3.3) can be further approximated by

$$U(x_i, y_i) \approx \sum_{p=1}^{N_T} \sum_{q=1}^{N_G} W_{G,q} H(x_i - x_o^{pq}, y_i - y_o^{pq}) |S_p|, \quad I(x_i, y_i) = |U(x_i, y_i)|^2, \quad (3.7)$$

where (x_o^{pq}, y_o^{pq}) is the coordinate of q -th Gaussian point in the p -th triangle in the triangulation of polygon S , $W_{G,q}$ is the weight of q -th Gaussian point. S_p represents the p -th triangle, the measure of which is $|S_p|$. And N_G is the total number of Gaussian points.

If the N_G -point Gaussian quadrature formula used in this article has precision of degree n , which means the scheme can achieve exact result for all polynomials with degree less than or equal to n , the error estimation of the formula in one triangle can be represented as follows:

$$\iint_{S_p} H(x_i - x_o, y_i - y_o) dx_o dy_o - \sum_{q=1}^{N_G} W_{G,q} H(x_i - x_o^{pq}, y_i - y_o^{pq}) |S_p| \leq C h_p^{n+1}, \quad (3.8)$$

where h_p is the length of the longest edge in the p -th triangle and C is a constant that depends on the $(n+1)$ -th derivative of function H while independent of h_p . We should

point out that for the polygon region S , if we assume that the family of triangulation is shape-regular quasi-uniform as defined in [10, Definitions 3.2 and 3.3], we can easily derive that the total number of triangles N_T generated by triangulation has a quadratic relationship with the reciprocal of the maximum of edge length h_p , i.e. $N_T \sim \mathcal{O}(1/\max h_p^2)$. Sum both sides of Eq. (3.8) over p and we can get

$$\begin{aligned} & \sum_{p=1}^{N_T} \iint_{S_p} H(x_i - x_o, y_i - y_o) dx_o dy_o - \sum_{p=1}^{N_T} \sum_{q=1}^{N_G} W_{G,q} H(x_i - x_o^{pq}, y_i - y_o^{pq}) |S_p| \\ & \leq \sum_{p=1}^{N_T} C h_p^{n+1} \leq N_T C \max h_p^{n+1} \sim \mathcal{O}(\max h_p^{n-1}). \end{aligned} \tag{3.9}$$

Remark 3.1. Theorem 3.1 implies that the accuracy of polygon approximation is of 2-nd order, and Eq. (3.9) indicates that the numerical integration formula has an accuracy of order $n - 1$. To prevent the loss of accuracy, here the Gaussian quadrature formula we apply should have at least a precision of order 3.

However, the coordinates of Gaussian points cannot exist independently. They are determined by the coordinates of the vertex of the triangle in which they are located. Without loss of generality, we represent the three vertices in the p -th triangle by $R_{p,a}, R_{p,b}, R_{p,c}$. According to the areal coordinate representation of Gaussian points, for the p -th triangle, we can obtain the relation between vertex coordinates and Gaussian point coordinates

$$\mathbf{G}_p = \begin{pmatrix} x_o^{p1} & x_o^{p2} & \cdots & x_o^{pN_G} \\ y_o^{p1} & y_o^{p2} & \cdots & y_o^{pN_G} \end{pmatrix} = \begin{pmatrix} R_{p,ax} & R_{p,bx} & R_{p,cx} \\ R_{p,ay} & R_{p,by} & R_{p,cy} \end{pmatrix} \mathbf{L}, \tag{3.10}$$

where $\mathbf{G}_p \in \mathbb{R}^{2 \times N_G}$ is the Gaussian point matrix in the p -th triangle. $\mathbf{L} \in \mathbb{R}^{3 \times N_G}$ is the weight matrix of the integral points with respect to the vertices, each column of which determined the location of a Gaussian point.

Then we consider how to build the relationship between the vertices of each triangle and the control points for periodic B-spline curve, so that we can explicitly compute the gradient later. For a given sequence of n control points P_1, P_2, \dots, P_n , we can generate a spline curve by these control points. Every point on the curve corresponds to a curve parameter. To discretize the curve and further approximate the region by polygon, we should arrange some points on the curve. Here we choose m curve parameters t_1, t_2, \dots, t_m and obtain the discrete points Q_1, Q_2, \dots, Q_m on the curve boundary in the following way:

$$\begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{pmatrix} = \begin{pmatrix} \mathring{N}_{1,p}(t_1) & \mathring{N}_{2,p}(t_1) & \cdots & \mathring{N}_{n,p}(t_1) \\ \mathring{N}_{1,p}(t_2) & \mathring{N}_{2,p}(t_2) & \cdots & \mathring{N}_{n,p}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathring{N}_{1,p}(t_m) & \mathring{N}_{2,p}(t_m) & \cdots & \mathring{N}_{n,p}(t_m) \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{pmatrix}, \tag{3.11}$$

where $t_k \in [0, 1]$, $k = 1, 2, \dots, m$ and $t_k < t_{k+1}$, $k = 1, 2, \dots, m-1$. So that the coordinates of discrete points Q_1, Q_2, \dots, Q_m can be represented by the linear combination of coordinates of control points. Now we can describe the process in matrix multiplication form

$$\mathbf{Q} = \mathbf{NP}, \quad \mathbf{Q} \in \mathbb{R}^{m \times 2}, \quad \mathbf{N} \in \mathbb{R}^{m \times n}, \quad \mathbf{P} \in \mathbb{R}^{n \times 2}. \quad (3.12)$$

The initial triangulation of domain S_0 is performed by CDT algorithm. To perform the triangulation, we should choose m points Q_1, Q_2, \dots, Q_m on the curve first to discretize the boundary curve. Then we perform CDT based on these points to obtain a rough triangular mesh, which does not have any inner vertices. After deleting triangles out of the boundary, the further triangulation of S_0 is performed by refinement of those initial rough meshes.

In the following, we consider the process of determining an inner triangle by the points Q_1, Q_2, \dots, Q_m on the spline curve. First, all the vertices of the triangles in the initial mesh are just Q_1, Q_2, \dots, Q_m , we rename them by $\tilde{R}_1, \tilde{R}_2, \dots, \tilde{R}_m$. In matrix form we have

$$\begin{pmatrix} \tilde{R}_1 \\ \tilde{R}_2 \\ \vdots \\ \tilde{R}_m \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{pmatrix} = \mathbf{I}_m \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{pmatrix}. \quad (3.13)$$

Then, to refine the mesh, we need to add new points. For a triangle larger than our tolerance, whose three vertices are $\tilde{R}_{i_0}, \tilde{R}_{j_0}, \tilde{R}_{k_0}$, we add its center of gravity as a new point. If we call the new point \tilde{R}_{m+1} , then we can modify the link matrix between $\{\tilde{R}_k\}_{k=1}^{m+1}$ and $\{Q_k\}_{k=1}^m$ in the following way:

$$\begin{pmatrix} \tilde{R}_1 \\ \tilde{R}_2 \\ \vdots \\ \tilde{R}_{m+1} \end{pmatrix} = \begin{pmatrix} & & & \mathbf{I}_m \\ \frac{1}{3}(\mathbf{e}_{i_0} + \mathbf{e}_{j_0} + \mathbf{e}_{k_0}) & & & \end{pmatrix} \begin{pmatrix} \tilde{R}_1 \\ \tilde{R}_2 \\ \vdots \\ \tilde{R}_m \end{pmatrix} = \mathbf{W}_{m+1,m} \begin{pmatrix} \tilde{R}_1 \\ \tilde{R}_2 \\ \vdots \\ \tilde{R}_m \end{pmatrix} = \mathbf{W}_{m+1,m} \mathbf{I}_m \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{pmatrix}, \quad (3.14)$$

where $\mathbf{e}_{i_0} \in \mathbb{R}^{1 \times m}$ is a row vector in which only the i_0 -th element is 1 and the other elements are 0. After adding the new point, we can refine the triangulation. We repeat the process until the longest edge of each triangle becomes shorter than 1.1 times of longest edge of initial polygon, thereby avoiding constant refinement of triangles adjacent to the boundary. In this process, every new point added can be represented by the average of three vertices of the triangle located. Assuming that there are K points in the final triangulation, we can get a series of link matrix $\mathbf{W}_{m+k+1,m+k}$ ($k = 0, 1, \dots, K-m-1$), the definition of which is

$$\mathbf{W}_{m+k+1,m+k} = \begin{pmatrix} & & & \mathbf{I}_{m+k} \\ \frac{1}{3}(\mathbf{e}_{i_0} + \mathbf{e}_{j_0} + \mathbf{e}_{k_0}) & & & \end{pmatrix}, \quad (3.15)$$

where i_0, j_0, k_0 are the indexes of the points forming the triangle which needs to be refined in the k -th adding points.

In this way, every vertex in the final triangulation can be represented by linear combinations of the points Q_1, Q_2, \dots, Q_m by the following formula:

$$\begin{pmatrix} \tilde{R}_1 \\ \tilde{R}_2 \\ \vdots \\ \tilde{R}_K \end{pmatrix} = \mathbf{W}_{K,K-1} \mathbf{W}_{K-1,K-2} \cdots \mathbf{W}_{m+1,m} \mathbf{I}_m \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{pmatrix} = \mathbf{W} \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{pmatrix} \quad (3.16)$$

with the definition

$$\mathbf{W} = \mathbf{W}_{K,K-1} \mathbf{W}_{K-1,K-2} \cdots \mathbf{W}_{m+1,m} \mathbf{I}_m. \quad (3.17)$$

As a result, every vertex of each triangle can be represented in the form of matrix multiplication

$$\tilde{\mathbf{R}} = \mathbf{W}\mathbf{Q}, \quad \tilde{\mathbf{R}} \in \mathbb{R}^{K \times 2}, \quad \mathbf{W} \in \mathbb{R}^{K \times m}, \quad \mathbf{Q} \in \mathbb{R}^{m \times 2}, \quad (3.18)$$

where \mathbf{W} is the coefficient of linear combination.

After refining the mesh, we will obtain all point coordinates $\tilde{\mathbf{R}}$ and connectivity list $\mathbf{C} \in \mathbb{R}^{N_T \times 3}$, each row of \mathbf{C} consists of indexes of the three vertexes of a single triangle. Then we will reshape the matrix $\tilde{\mathbf{R}}$ into \mathbf{R} according to the connectivity list. An intuitive structure is

$$\mathbf{R}(:, :, 1) = \begin{pmatrix} R_{1,ax} & R_{1,bx} & R_{1,cx} \\ R_{2,ax} & R_{2,bx} & R_{2,cx} \\ \vdots & \vdots & \vdots \\ R_{N_T,ax} & R_{N_T,bx} & R_{N_T,cx} \end{pmatrix}, \quad \mathbf{R}(:, :, 2) = \begin{pmatrix} R_{1,ay} & R_{1,by} & R_{1,cy} \\ R_{2,ay} & R_{2,by} & R_{2,cy} \\ \vdots & \vdots & \vdots \\ R_{N_T,ay} & R_{N_T,by} & R_{N_T,cy} \end{pmatrix}. \quad (3.19)$$

\mathbf{R} is a three-dimensional matrix, written $\mathbf{R} \in \mathbb{R}^{N_T \times 3 \times 2}$. $\mathbf{R}(:, :, 1)$ stores the X-coordinate information for each point, and $\mathbf{R}(:, :, 2)$ stores the Y-coordinate information for each point. Each row includes the information of three vertices of a single triangle. An easy way to carry out the reshaping process is to assign values in $\tilde{\mathbf{R}}$ by the indexes in \mathbf{C} . We note the procedure by

$$\mathbf{R} = \tilde{\mathbf{R}}(\mathbf{C}), \quad \mathbf{R} \in \mathbb{R}^{N_T \times 3 \times 2}, \quad \tilde{\mathbf{R}} \in \mathbb{R}^{K \times 2}, \quad \mathbf{C} \in \mathbb{R}^{N_T \times 3}. \quad (3.20)$$

As mentioned above, to obtain the coordinates of Gaussian points, we only need to multiply a weight matrix connecting the vertices and the Gaussian points \mathbf{L} to the right, which makes our calculations a lot easier. And finally, we have successfully established the process between Gaussian points and control points. An illustration of the process is shown in Fig. 5.

For the sampling points on the image plane, we usually choose them equidistantly to simplify the following data visualization. In this way, we finally finished the numerical imaging formula derivation.

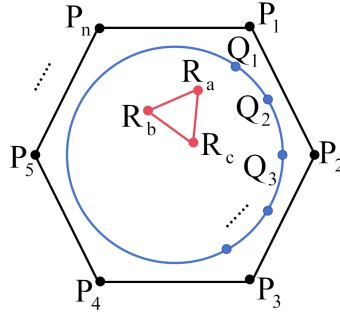


Figure 5: Explicit relation building.

3.3 Inverse lithography problem

After finishing the derivation of imaging formula and obtaining the explicit relationship between the control points and the Gaussian points, we are in a position to discuss the inverse lithography problem. We remark that normalization of coordinates has no essential effect on the inverse problem. So in this part, we will still omit the $\hat{\cdot}$ symbol.

When light passes through the mask, diffraction and interference effects will occur among adjacent mask patterns. Patterns on the wafer will experience varying degrees of distortion. The objective of the inverse lithography problem is to adjust the pattern of the mask to make the pattern we get on the wafer closer to our target pattern. Here we use coordinates of control points as the optimization variable and finally it becomes an unconstrained optimization problem.

First, we will introduce the optimization model of the inverse lithography problem. We note the intensity distribution function obtained by numerical computation as

$$I(x_i, y_i) = |U(x_i, y_i)|^2 = U(x_i, y_i) * \overline{U(x_i, y_i)},$$

where (x_i, y_i) represents the coordinate on the image plane. Taking into account the effects of photoresist on the wafer, in the optical lithography simulation, a widely used model is to truncate the intensity by a threshold function

$$T(x) = \begin{cases} 1, & x \geq tr, \\ 0, & x < tr, \end{cases}$$

where tr is the threshold. However, the hard truncation function is discontinuous, which leads to the difficulty of calculating its gradient. So we will approximate it by sigmoid function

$$\text{sig}(x) = \frac{1}{1 + \exp(-a(x - tr))},$$

where a controls the approximation accuracy. Given the target intensity distribution function $I_{target}(x_i, y_i)$ in a binary way, then the mask optimization problem can be reduced to

an unconstrained optimization problem

$$\min_{P_1, \dots, P_n} J := \|\text{sig}(I) - I_{\text{target}}\|_2^2 = \iint_{\Omega} (\text{sig}(I) - I_{\text{target}})^2 dx dy,$$

where domain Ω is the whole image plane. Further we will calculate the derivative of the objective function with respect to the horizontal and vertical coordinates $P_{1x}, P_{2x}, \dots, P_{nx}$ and $P_{1y}, P_{2y}, \dots, P_{ny}$ of the control points P_1, P_2, \dots, P_n

$$\left(\begin{array}{c} \frac{\partial J}{\partial P_{kx}} \\ \frac{\partial J}{\partial P_{ky}} \end{array} \right) = \left(\begin{array}{c} \iint_{\Omega} 2(\text{sig}(U\bar{U}) - I_{\text{target}}) \text{sig}'(U\bar{U}) \left(\bar{U} \frac{\partial U}{\partial P_{kx}} + U \frac{\partial \bar{U}}{\partial P_{kx}} \right) dx dy_i \\ \iint_{\Omega} 2(\text{sig}(U\bar{U}) - I_{\text{target}}) \text{sig}'(U\bar{U}) \left(\bar{U} \frac{\partial U}{\partial P_{ky}} + U \frac{\partial \bar{U}}{\partial P_{ky}} \right) dx dy_i \end{array} \right), \quad k=1, 2, \dots, n.$$

The integral is carried out in the image plane. Now we discretize the integrals equidistantly on the image plane by $N_x \times N_y$ sampling points $(x_i^m, y_i^n), m=1, 2, \dots, N_x, n=1, 2, \dots, N_y$

$$\begin{aligned} & \sum_{m,n=1}^{N_x, N_y} 2 \left(\text{sig} \left(U(x_i^m, y_i^n) \overline{U(x_i^m, y_i^n)} \right) - I_{\text{target}}(x_i^m, y_i^n) \right) \text{sig}' \left(U(x_i^m, y_i^n) \overline{U(x_i^m, y_i^n)} \right) \\ & \quad \times \left(\overline{U(x_i^m, y_i^n)} \frac{\partial U(x_i^m, y_i^n)}{\partial P_{kx}} + U(x_i^m, y_i^n) \frac{\partial \overline{U(x_i^m, y_i^n)}}{\partial P_{kx}} \right) \Delta x_i \Delta y_i, \\ & \sum_{m,n=1}^{N_x, N_y} 2 \left(\text{sig} \left(U(x_i^m, y_i^n) \overline{U(x_i^m, y_i^n)} \right) - I_{\text{target}}(x_i^m, y_i^n) \right) \text{sig}' \left(U(x_i^m, y_i^n) \overline{U(x_i^m, y_i^n)} \right) \\ & \quad \times \left(\overline{U(x_i^m, y_i^n)} \frac{\partial U(x_i^m, y_i^n)}{\partial P_{ky}} + U(x_i^m, y_i^n) \frac{\partial \overline{U(x_i^m, y_i^n)}}{\partial P_{ky}} \right) \Delta x_i \Delta y_i. \end{aligned} \quad (3.21)$$

Considering that conjugated functions can be directly transformed into each other, the problem can be reduced to calculating the explicit expression of $\partial U / \partial P_{kx}$ and $\partial U / \partial P_{ky}$.

We come back to the formula (3.7) that computes the intensity on the image plane

$$U(x_i, y_i) = \sum_{p=1}^{N_T} \sum_{q=1}^{N_G} W_{G,q} H(x_i - x_o^{pq}, y_i - y_o^{pq}) |S_p|, \quad I(x_i, y_i) = |U(x_i, y_i)|^2. \quad (3.22)$$

In this formula, H and $|S_p|$ are related to the coordinates of the control points. The expression of derivative can be written as

$$\begin{aligned} \frac{\partial U(x_i^m, y_i^n)}{\partial P_{kx}} &= \sum_{p=1}^{N_T} \sum_{q=1}^{N_G} W_{G,q} \left[\frac{\partial H(x_i^m - x_o^{pq}, y_i^n - y_o^{pq})}{\partial P_{kx}} |S_p| + H(x_i^m - x_o^{pq}, y_i^n - y_o^{pq}) \frac{\partial |S_p|}{\partial P_{kx}} \right], \\ \frac{\partial U(x_i^m, y_i^n)}{\partial P_{ky}} &= \sum_{p=1}^{N_T} \sum_{q=1}^{N_G} W_{G,q} \left[\frac{\partial H(x_i^m - x_o^{pq}, y_i^n - y_o^{pq})}{\partial P_{ky}} |S_p| + H(x_i^m - x_o^{pq}, y_i^n - y_o^{pq}) \frac{\partial |S_p|}{\partial P_{ky}} \right]. \end{aligned} \quad (3.23)$$

The problem can be further reduced to the calculation of the following four derivatives:

$$\frac{\partial |S_p|}{\partial P_{kx}}, \quad \frac{\partial |S_p|}{\partial P_{ky}}, \quad \frac{\partial H(x_i - x_o^{pq}, y_i - y_o^{pq})}{\partial P_{kx}}, \quad \frac{\partial H(x_i - x_o^{pq}, y_i - y_o^{pq})}{\partial P_{ky}}.$$

We start from the calculation of $\partial |S_p| / \partial P_{kx}, \partial |S_p| / \partial P_{ky}$. If we assume that the three vertexes of the p -th triangle are $R_{p,a}, R_{p,b}, R_{p,c}$ and they have been arranged in counter-clockwise order, then the absolute value operation in the area formula can be omitted and it becomes

$$|S_p| = \frac{1}{2} ((R_{p,bx} - R_{p,ax})(R_{p,cy} - R_{p,ay}) - (R_{p,by} - R_{p,ay})(R_{p,cx} - R_{p,ax})). \quad (3.24)$$

In this way, the derivatives of $|S_p|$ become

$$\begin{aligned} \frac{\partial |S_p|}{\partial P_{kx}} &= \frac{1}{2} \left(\left(\frac{\partial R_{p,bx}}{\partial P_{kx}} - \frac{\partial R_{p,ax}}{\partial P_{kx}} \right) (R_{p,cy} - R_{p,ay}) - (R_{p,by} - R_{p,ay}) \left(\frac{\partial R_{p,cx}}{\partial P_{kx}} - \frac{\partial R_{p,ax}}{\partial P_{kx}} \right) \right), \\ \frac{\partial |S_p|}{\partial P_{ky}} &= \frac{1}{2} \left((R_{p,bx} - R_{p,ax}) \left(\frac{\partial R_{p,cy}}{\partial P_{ky}} - \frac{\partial R_{p,ay}}{\partial P_{ky}} \right) - \left(\frac{\partial R_{p,by}}{\partial P_{ky}} - \frac{\partial R_{p,ay}}{\partial P_{ky}} \right) (R_{p,cx} - R_{p,ax}) \right). \end{aligned} \quad (3.25)$$

There are six derivatives connecting the vertexes and the control points. Here we illustrate the calculation of derivatives using $\partial R_{p,ax} / \partial P_{kx}$ and $\partial R_{p,ay} / \partial P_{ky}$ as an example.

As mentioned above, in the process of generating matrix \mathbf{R} , we will obtain $\tilde{\mathbf{R}} = \mathbf{WNP}$ and connectivity list matrix \mathbf{C} . $R_{p,ax}$ is the $(p,1,1)$ -th entry in \mathbf{R} , which comes from the $(\mathbf{C}(p,1),1)$ -th entry in $\tilde{\mathbf{R}}$. And $R_{p,ay}$ is the $(p,1,2)$ -th entry in \mathbf{R} , which comes from the $(\mathbf{C}(p,1),2)$ -th entry in $\tilde{\mathbf{R}}$. If we note $\mathbf{T} = \mathbf{WN} \in \mathbb{R}^{K \times n}$, then we can get

$$\frac{\partial R_{p,ax}}{\partial P_{kx}} = \frac{\partial R_{p,ay}}{\partial P_{ky}} = \mathbf{T}(\mathbf{C}(p,1),k), \quad k=1,2,\dots,n. \quad (3.26)$$

Similarly, we can obtain

$$\frac{\partial R_{p,bx}}{\partial P_{kx}} = \frac{\partial R_{p,by}}{\partial P_{ky}} = \mathbf{T}(\mathbf{C}(p,2),k), \quad \frac{\partial R_{p,cx}}{\partial P_{kx}} = \frac{\partial R_{p,cy}}{\partial P_{ky}} = \mathbf{T}(\mathbf{C}(p,3),k), \quad k=1,2,\dots,n. \quad (3.27)$$

Now we have finished the calculation of $\partial |S_p| / \partial P_{kx}, \partial |S_p| / \partial P_{ky}$. Then we consider the calculation of the last two derivatives. In the preliminary section, we have already obtained the formula of optical system transfer function in Eq. (2.5). Here we also omitted the $\hat{\cdot}$ symbol that represents the normalization and we get

$$H(x_i - x_o, y_i - y_o) = \frac{J_1(2\pi\rho)}{\rho} = \frac{J_1(2\pi\sqrt{(x_i - x_o)^2 + (y_i - y_o)^2})}{\sqrt{(x_i - x_o)^2 + (y_i - y_o)^2}}. \quad (3.28)$$

Then the two derivatives can be directly written as

$$\begin{aligned}\frac{\partial H(x_i - x_o^{pq}, y_i - y_o^{pq})}{\partial P_{kx}} &= \frac{1}{\rho^2} \left[\frac{\partial J_1(2\pi\rho)}{\partial \rho} \frac{\partial \rho}{\partial P_{kx}} \rho - J_1(2\pi\rho) \frac{\partial \rho}{\partial P_{kx}} \right], \\ \frac{\partial H(x_i - x_o^{pq}, y_i - y_o^{pq})}{\partial P_{ky}} &= \frac{1}{\rho^2} \left[\frac{\partial J_1(2\pi\rho)}{\partial \rho} \frac{\partial \rho}{\partial P_{ky}} \rho - J_1(2\pi\rho) \frac{\partial \rho}{\partial P_{ky}} \right].\end{aligned}\quad (3.29)$$

In the formula, terms that we need to further explain are $\partial J_1(2\pi\rho)/\partial \rho, \partial \rho/\partial P_{kx}, \partial \rho/\partial P_{ky}$. Using the recursion property of the first kind of Bessel function we can get

$$J_{n-1}(x) - J_{n+1}(x) = 2J'_n(x).$$

Then the first term can be directly calculated by

$$\frac{\partial J_1(2\pi\rho)}{\partial \rho} = 2\pi J'_1(2\pi\rho) = \pi[J_0(2\pi\rho) - J_2(2\pi\rho)].\quad (3.30)$$

As for the latter two terms, we can get

$$\frac{\partial \rho}{\partial P_{kx}} = \frac{x_o^{pq} - x_i^m}{\rho} \frac{\partial x_o^{pq}}{\partial P_{kx}}, \quad \frac{\partial \rho}{\partial P_{ky}} = \frac{y_o^{pq} - y_i^n}{\rho} \frac{\partial y_o^{pq}}{\partial P_{ky}}.\quad (3.31)$$

Recall Eq. (3.10), in which we have

$$\begin{pmatrix} x_o^{p1} & x_o^{p2} & \cdots & x_o^{pN_G} \\ y_o^{p1} & y_o^{p2} & \cdots & y_o^{pN_G} \end{pmatrix} = \begin{pmatrix} R_{p,ax} & R_{p,bx} & R_{p,cx} \\ R_{p,ay} & R_{p,by} & R_{p,cy} \end{pmatrix} \mathbf{L}.\quad (3.32)$$

It can be derived directly that:

$$\begin{pmatrix} \frac{\partial x_o^{p1}}{\partial P_{kx}} & \frac{\partial x_o^{p2}}{\partial P_{kx}} & \cdots & \frac{\partial x_o^{pN_G}}{\partial P_{kx}} \\ \frac{\partial y_o^{p1}}{\partial P_{ky}} & \frac{\partial y_o^{p2}}{\partial P_{ky}} & \cdots & \frac{\partial y_o^{pN_G}}{\partial P_{ky}} \end{pmatrix} = \begin{pmatrix} \frac{\partial R_{p,ax}}{\partial P_{kx}} & \frac{\partial R_{p,bx}}{\partial P_{kx}} & \frac{\partial R_{p,cx}}{\partial P_{kx}} \\ \frac{\partial R_{p,ay}}{\partial P_{ky}} & \frac{\partial R_{p,by}}{\partial P_{ky}} & \frac{\partial R_{p,cy}}{\partial P_{ky}} \end{pmatrix} \mathbf{L}.\quad (3.33)$$

Now we have obtained all the derivatives that need to be calculated, and finally obtained the gradient of the original optimization problem. The method can be extended to multiple separate regions. The only point need to be considered is that the amplitude U should be the sum of amplitude produced by each region, but the derivatives $\partial U/\partial P_{kx}, \partial U/\partial P_{ky}$ are only influenced by the region that the control points belong to.

4 Algorithm

With the information of $\partial J/\partial P_{kx}, \partial J/\partial P_{ky}, k=1,2,\dots,n$, in fact we have obtained the gradient information ∇J of target function J and so that we can solve the optimization problem by various gradient-based methods, e.g. steepest descent method and conjugate gradient

method. In this article we apply steepest descent (SD) method to set the descent direction and golden section method to set the step size α_k . The basic iteration formula is

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \alpha_k \nabla J(\mathbf{P}_k).$$

One thing needs to be explained in particular is that, in the process of optimization, the shape of the domain could twist violently, and the mesh will probably intersect. So after every step, we should regenerate the mesh and corresponding matrices. Based on these ideas, we introduced the procedure of our method in Algorithm 1.

Algorithm 1. Inverse Lithography Based on Periodic B-splines.

Require:

- Maximum number of iterations I_M .
- Accuracy level ϵ .
- Wave length λ_0 .
- Numerical aperture NA .
- Lateral magnification M .
- Initial control points \mathbf{P}_0 .
- Target mask pattern I_{target} .
- Curve parameter t_1, t_2, \dots, t_m .

Ensure: Optimized control points \mathbf{P} .

Normalize the control points \mathbf{P}_0 by Eq. (2.1).

$\mathbf{P} \leftarrow \mathbf{P}_0$.

Generate discrete points on the periodic B-spline curve by Eq. (3.11) and obtain \mathbf{Q} , \mathbf{N} defined in Eq. (3.12).

CDT and refinement to obtain $\tilde{\mathbf{R}}$, \mathbf{C} and \mathbf{W} by Eqs. (3.13)-(3.18).

Reshape $\tilde{\mathbf{R}}$ into \mathbf{R} by Eqs. (3.19), (3.20).

$k \leftarrow 1$.

while $J = \|\text{sig}(I) - I_{target}\|_2^2 > \epsilon$ && $k < I_M$ **do**

$k \leftarrow k + 1$.

Compute ∇J by Eqs. (3.21)-(3.33).

Determine the optimal step size α_k .

if $\alpha_k < \epsilon$ **then**

Inverse normalize \mathbf{P} by Eq. (2.2).

Return \mathbf{P} .

else if $\alpha_k \geq \epsilon$ **then**

$\mathbf{P} \leftarrow \mathbf{P} - \alpha_k \nabla J$.

Regenerate $\mathbf{N}, \mathbf{Q}, \mathbf{C}, \mathbf{W}, \tilde{\mathbf{R}}, \mathbf{R}$.

end if

end while

Inverse normalization \mathbf{P} by Eq. (2.2).

Return \mathbf{P} .

5 Numerical results

In this section, we present some numerical examples for a variety of mask patterns to show the efficiency of our proposed algorithm. We implement Algorithm 1 by Matlab R2024b. During the numerical calculation, the common parameters that we selected are $a = 90$, $tr = 0.3$, $M = -1$, $\lambda_0 = 193 \text{ nm}$, $NA = 0.93$. In order to ensure that the boundary has a certain smoothness, the order of periodic B-spline we selected is $p=3$. The stop criterion is set by $\epsilon = 1e-4$ and $I_M = 100$. As we highlighted in Remark 3.1, here we choose a 4 point triangular Gaussian quadrature formula with precision of degree 3. In order to see the difference between the target and the output more clearly, in each example, we provide the information of edge placement error(EPE) which is defined by

$$EPE = |\text{sig}(I) - I_{\text{target}}|.$$

After regarding EPE as a matrix, we can further define the error by Frobenius norm of EPE

$$\text{error} = \|EPE\|_F.$$

Example 1. In Fig. 6, we use the simplest square target pattern as our first example. In this example, we arrange 40 control points on the boundary, 100 discrete points on the spline curve, and set the image size by 100×100 pixels. The resolution in this example is 4 nm/pixel . It took 5144 seconds to finish the 100-step optimization process. In Fig. 6(a) we show the target square pattern. In Fig. 6(b) we show how the objective function decreases as the iteration goes on. We can see from the error decreasing graph that an almost optimal mask can be obtained in about twenty steps, so in fact, we only need to spend about 1500 seconds to get the optimized mask. Fig. 6(c) shows the initial mask pattern, in which we indicate the control points, curvilinear mask boundary, and the triangulation of the initial mask. The initial locations of control points are determined by the boundary of target pattern. Fig. 6(d) shows the output patterns with initial mask pattern as input. It is almost a circle, which differs a lot from the target. Fig. 6(e) shows the edge placement error map of initial mask. Figs. 6(f)-6(h) are corresponding pictures for the optimized mask. Among them, Fig. 6(f) shows the optimized mask with control points, curvilinear mask boundary and triangulation. It is quite different from the initial one in Fig. 6(c). From Figs. 6(g) and 6(h), we can find that the image pattern is better than that in Fig. 6(d) and the edge placement error is reduced significantly, which confirms the feasibility of our method.

The arrangements of the following Figs. 7, 8, 9, 10 are similar. For each target pattern, we present the initial mask along with its output, the optimized mask and its corresponding output. To more comprehensively assess the imaging quality, we also supply the edge placement error patterns for both the initial and optimized masks. Moreover, we provide a figure to illustrate the reduction of the error during the optimization process.

Example 2. In Fig. 7 we use a less symmetrical L-shaped pattern as an example to verify the adaptability of the model to patterns with inside and outside corners. We arrange

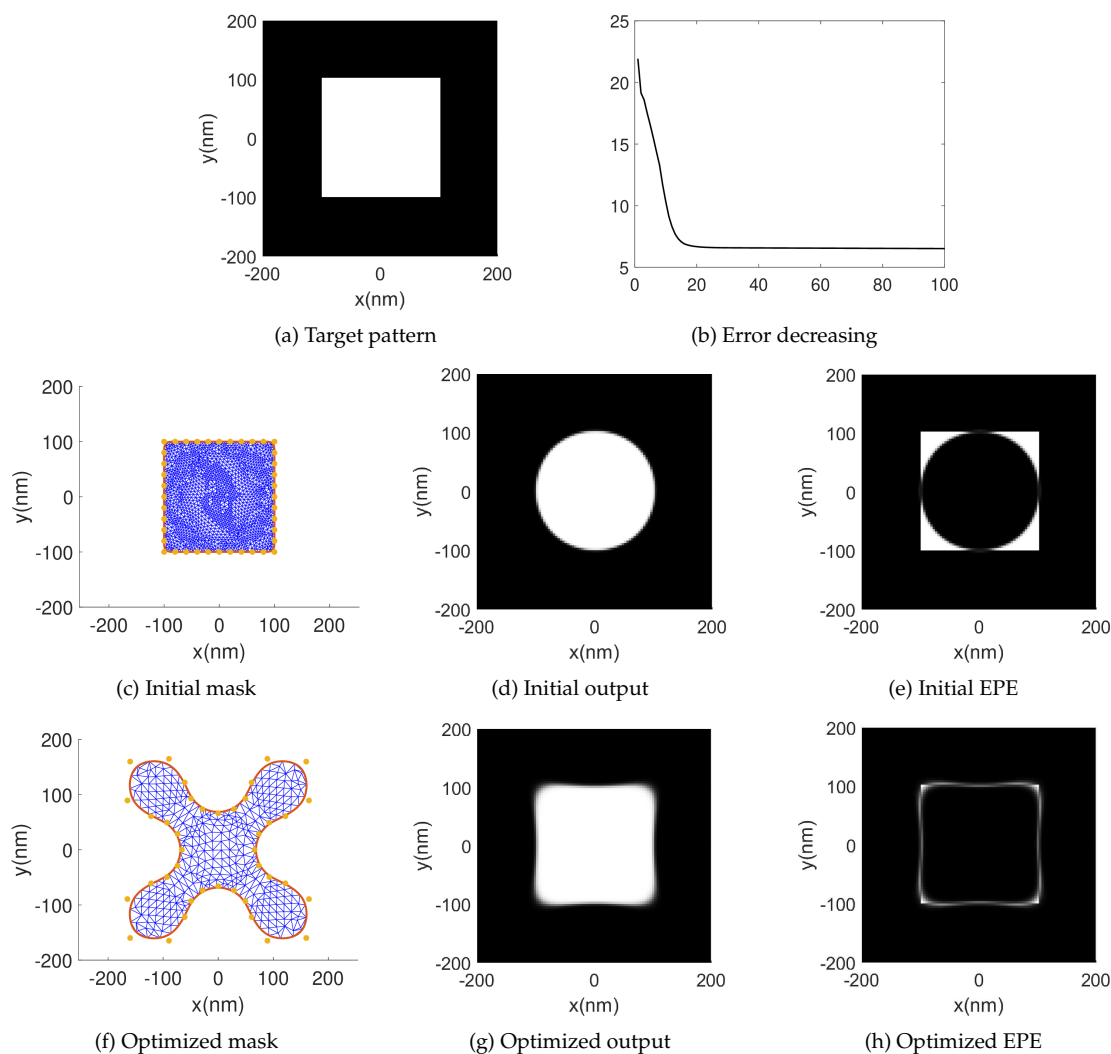


Figure 6: Example of square pattern.

68 control points on the boundary, 100 discrete points on the spline curve, and set the image size by 100×100 pixels. The resolution in this example is 5 nm/pixel. It took 7845 seconds to finish the 100-step optimization process. In the initial output, the output pattern shrinks significantly at each end and the inner corner is smoothed. Our method improves the imaging performance at each corner and makes the image more similar to the target pattern, which verifies the adaptability to the asymmetrical pattern of our method.

Example 3. In Fig. 8, we use a pattern of two separate rectangles as an example. Here we arrange 40 control points for each region on the boundary, 80 discrete points on each spline curve, and set the image size by 100×100 pixels. The resolution in this example

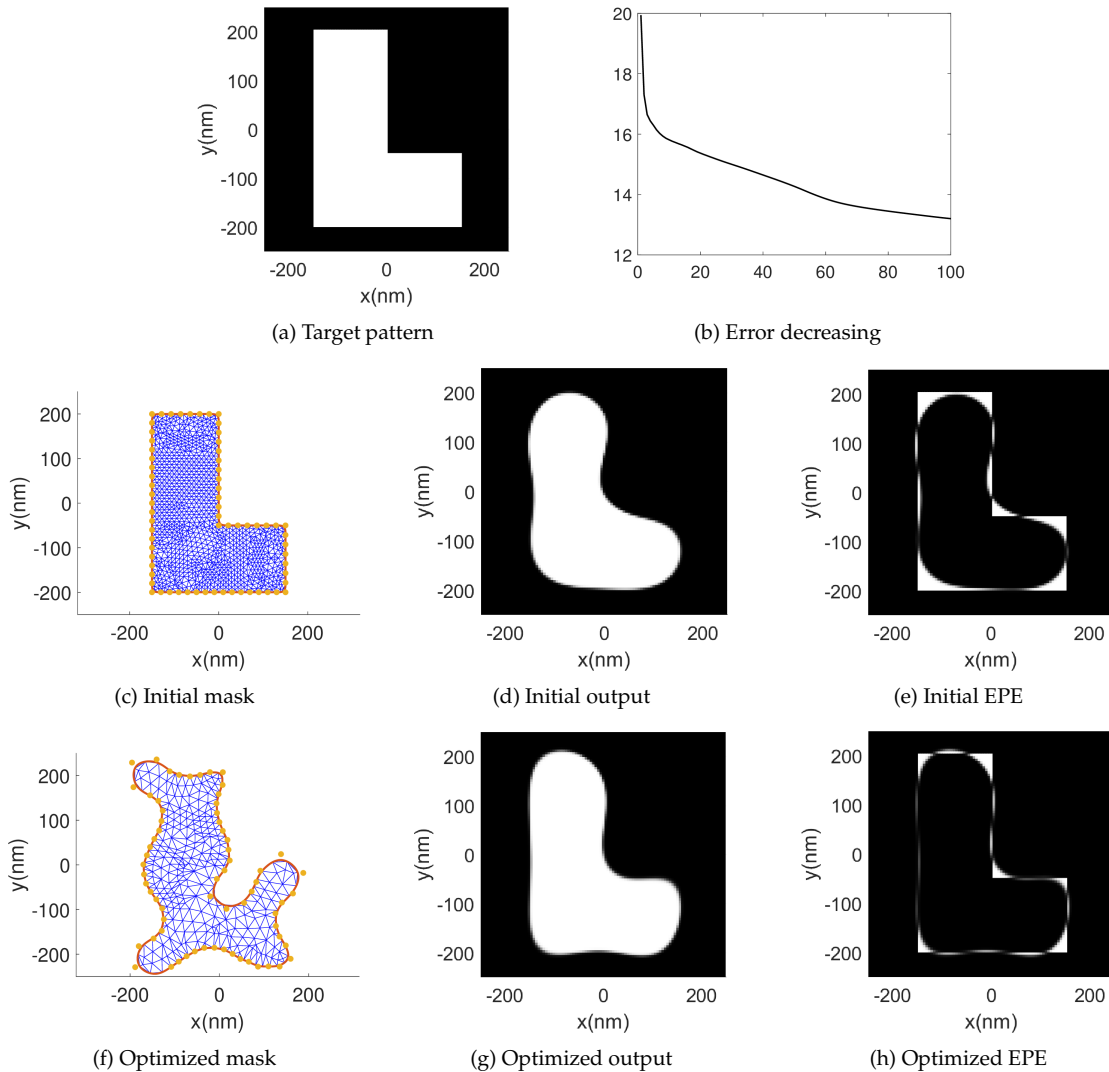


Figure 7: Example of L-shape pattern.

is 4 nm/pixel. It took 14040 seconds to finish the 100-step optimization process. At each end, the output pattern of the initial mask shrinks significantly and the position of output is away from the original design. This means that the initial mask is not suitable for generating the desired target pattern. After optimization by our method, both the shortening and offset of the initial mask are compensated. This example means that our method can be successfully used for multiple regions.

Example 4. In Fig. 9, we use a relatively complex reversal pattern as an example. In this example, we arrange 171 control points on the boundary, 300 discrete points on the spline curve, and set the image size by 150×150 pixels. The resolution in this example is about

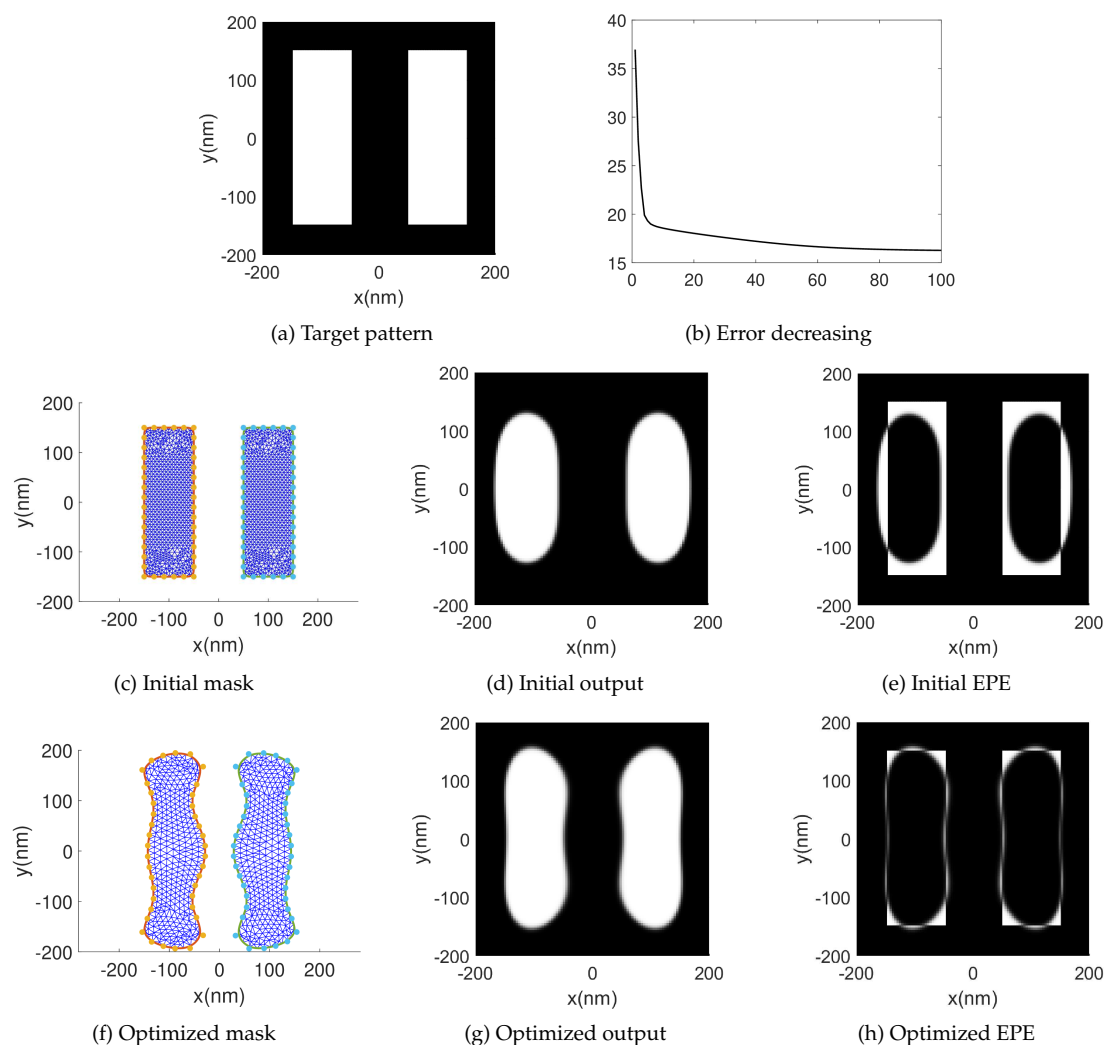


Figure 8: Example of two rectangle pattern.

6.67nm/pixel. It took 38583 seconds to complete the 100-step optimization process. From the initial EPE in Fig. 9(e) we can see some obvious shortcomings of the output. At the top left of the Fig. 9(e), we can see the shortening of the pattern. At the center of the same picture, we can see a severe narrowing, which is unacceptable in optical lithography. Here is still some offset at the center part. With the improvement of our method, the shortcomings mentioned above are noticeably remedied. The shrinkage at each end of the line is compensated. The severe narrowing of line width at the center of the pattern is corrected greatly, and the offset of the pattern is also corrected in the final result.

Example 5. In this example we use big-small-feature & corner test mask (BigMaC) [24] pattern as an example. It is a mask with a large number of features coexisting simulta-

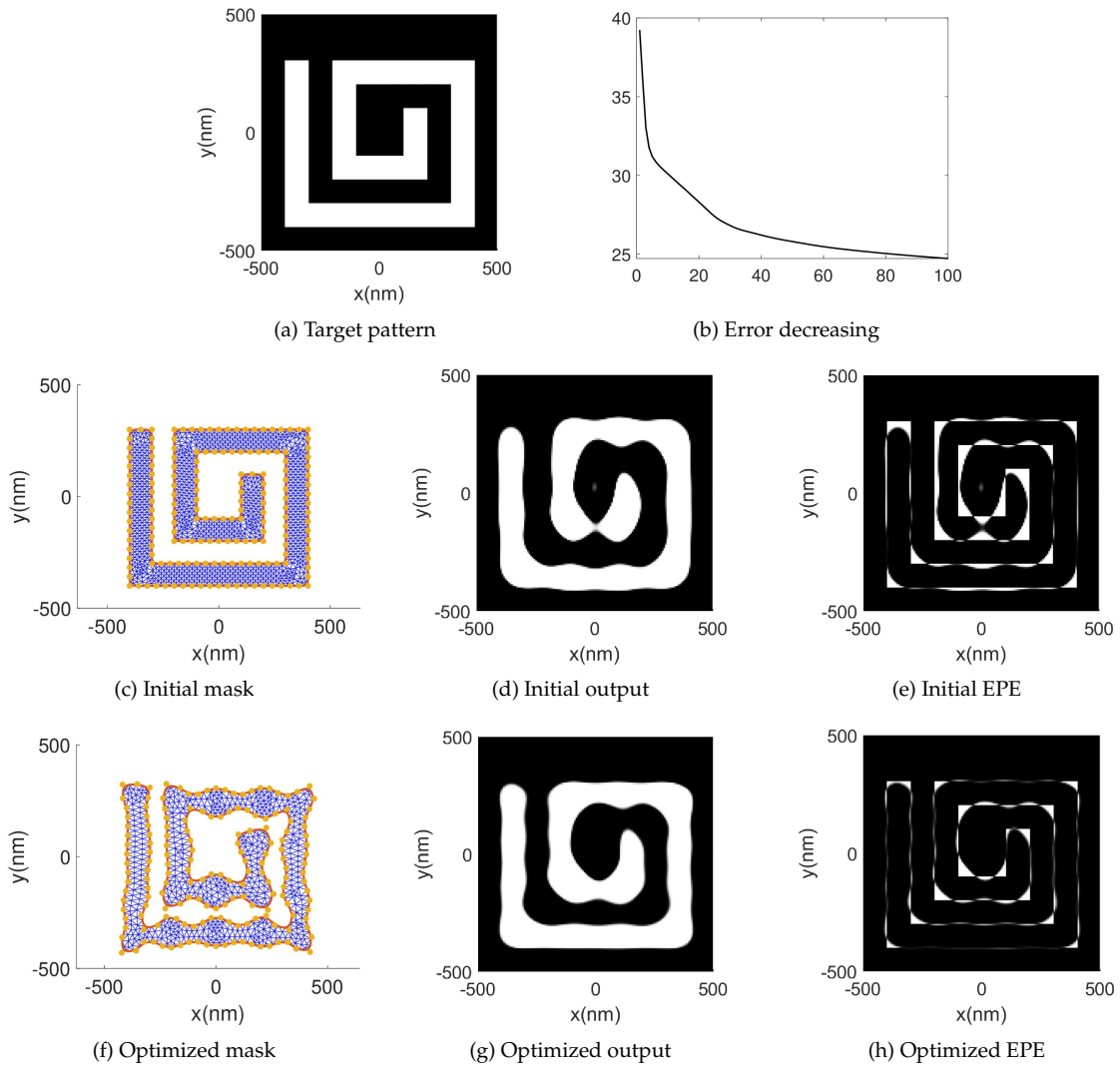


Figure 9: Example of reversal pattern.

neously, including both large-scale and small-scale features, long-scale and short-scale features, inner and outer corners, as well as vertical and horizontal features. The detailed results are listed in Fig. 10. Here we extend the size of the pattern to 1.3 times. The number of control points for the biggest region is 202, discrete points on this spline curve is 200. And for each small region, we arrange 20 control points on the boundary, 20 discrete points on the curve. We set the image size by 150×150 pixels. The resolution in this example is about 14.39 nm/pixel. It took 41765 seconds to complete the 100-step optimization process. From the results in the last row of Fig. 10, we can find that the optimization algorithm brings a satisfactory improvement on the narrowing, shortening and offset. This example demonstrates that our method still exhibits excellent optimization

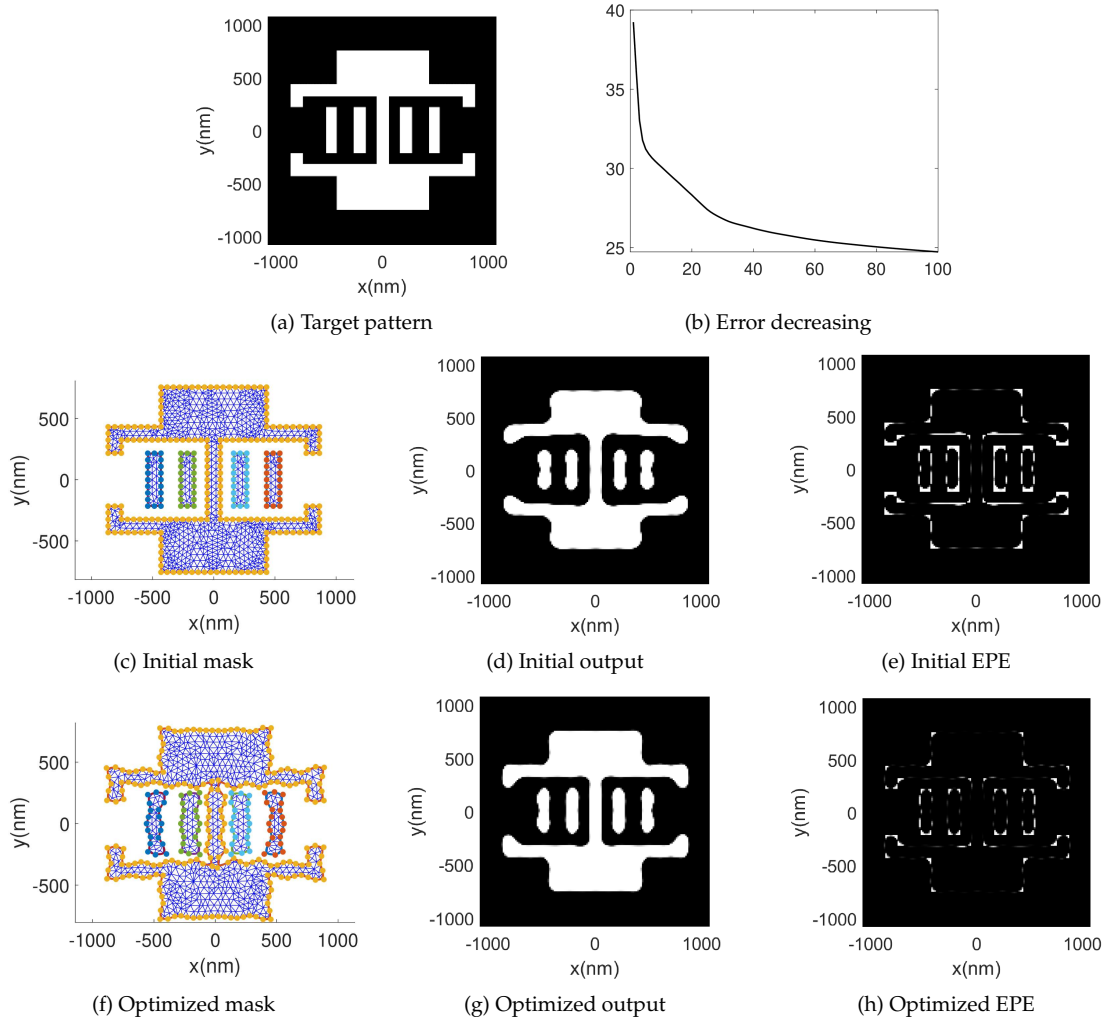


Figure 10: Example of BigMaC pattern.

capabilities for complex masks and has the potential to be applied to the production of complex masks.

Example 6. We compare our method with two state-of-the-art methods in this example. One of the method is an curvilinear optical proximity correction method using non-uniform B-spline curves and tabu search algorithm optimizer (we refer to it as NUBTS in the following), which can be found in [40]. The other is the conventional pixel-based optimization methods together with regularization techniques in [27]. In order to make the results comparable, we do not use the symmetric techniques introduced in [40] and we also pixelate our curve mask in Fig. 11(a). The results are exhibited in Fig. 11. The first row shows the results of our method, the second row shows the results of method based

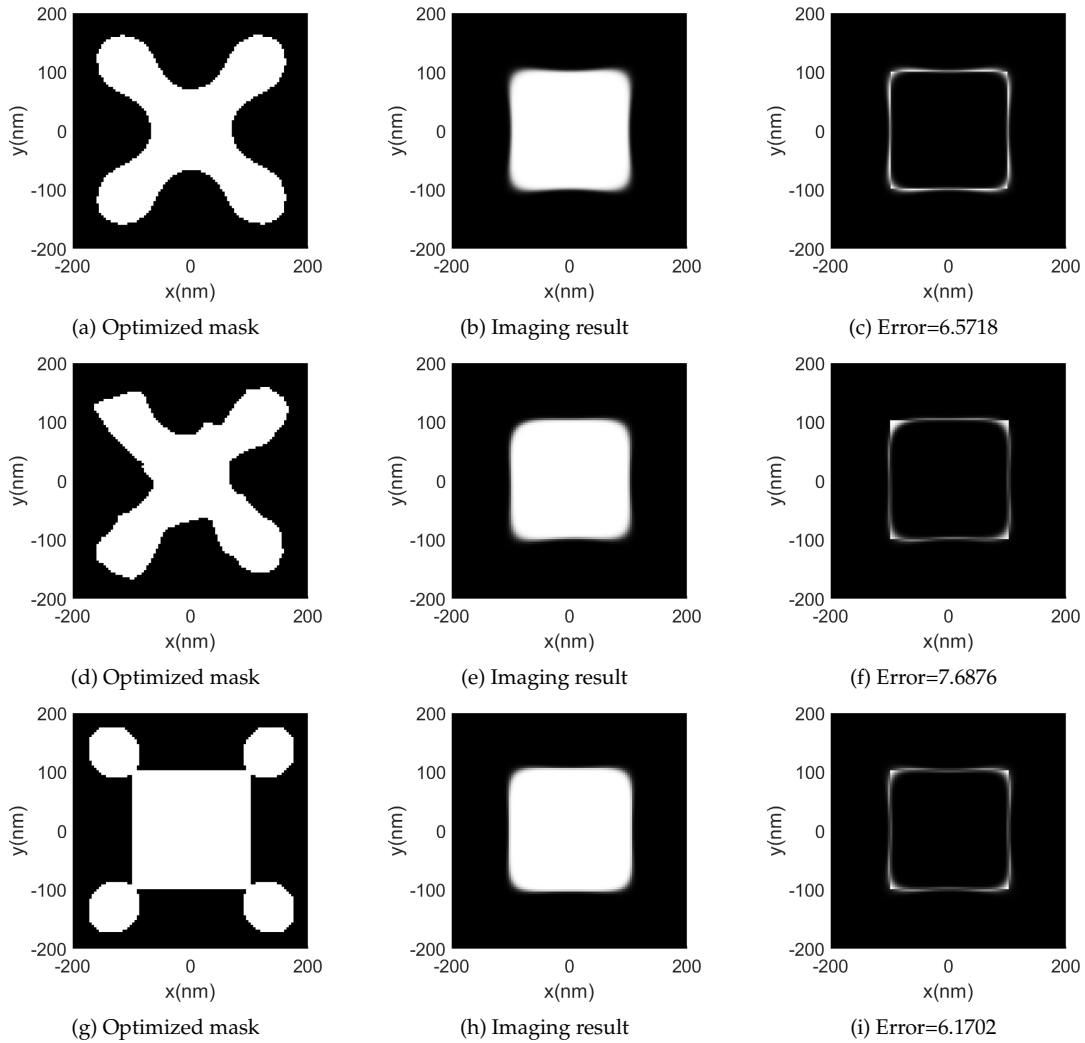


Figure 11: Comparison between different methods. First row: our method; second row: NUBTS method; third row: pixel based method.

on NUBTS and the third row shows the results based on pixel mask and steepest descent method. In these cases, the first column shows the optimized masks, the second column shows the images formed by the corresponding optimized masks, and the third column shows EPE patterns and the corresponding errors. In the three cases, we all set the mask size by 100×100 pixels. The resolution in these examples is 4 nm/pixel. From these results, we can see that, compared with the pixel-based method and NUBTS method, without causing significant loss of accuracy, our method has obtained a mask pattern with a smoother boundary, which is beneficial for mask production. Besides, without extra symmetric techniques, our method can automatically capture the symmetry of the image.

6 Conclusion

In this paper, the curvilinear mask boundary is represented by the periodic B-splines. The CDT and numerical integration are used to simulate the optical lithography imaging process. We establish an explicit relationship between the integral points and the control points of the spline curves. Based on the relationship, we further establish an explicit formula to calculate the gradient of objective functional for inverse lithography, and propose a steepest descent optimization algorithm based on the gradient. Various numerical experiments illustrate that our method exhibits excellent adaptability to the mask optimization problem, regardless of whether the mask has a single connected region or multiple separate regions. In the future, we will consider how to arrange initial control points more flexibly to give more freedom to the regions that need to be described more precisely. This will be helpful in improving the qualities of the inverse lithography problem.

Acknowledgments

The work was supported by the National Key R&D Program of China (Grant Nos. 2019YFA0709600, 2019YFA0709602).

References

- [1] W. Böhm, *Cubic B-spline curves and surfaces in computer aided geometric design*, Computing, 19:29–34, 1977.
- [2] I. Bork, P. Buck, B. Durvasula, V. Liubich, N. Rao, R. Sharma, and M. Zuo, *Curvature based fragmentation for curvilinear mask process correction*, in: Photomask Technology 2021, Proceedings of the SPIE, Vol. 11855, SPIE, 118550R, 2021.
- [3] I. Bork, K. Mishra, R. Sharma, and M. Zuo, *Curvilinear mask process correction – status quo and outlook*, in: 2022 International Workshop on Advanced Patterning Solutions (IWAPS), IEEE, 7–13, 2022.
- [4] I. Bork, A. Tritchkov, S. Shang, E. Levine, and M. Zuo, *MRC for curvilinear mask shapes*, in: Photomask Technology 2020, Proceedings of the SPIE, Vol. 11518, SPIE, 115180R-8, 2020.
- [5] M. Born and M. Wolf, *Principles of Optic*, Cambridge University Press, 1980.
- [6] Q. Cao, P. Xu, S. Sun, J. He, J. Wei, J. Shi, and Y. Wei, *Curvilinear mask optimization with re-fined generative adversarial nets*, Journal of Micro/Nanopatterning, Materials, and Metrology, 22:013201, 2023.
- [7] T. Cecil, D. Peng, D. Abrams, S. J. Osher, and E. Yablonovitch, *Advances in inverse lithography*, ACS Photonics, 10:910–918, 2023.
- [8] J. Chen and H. Liu, *An alternating direction method of multipliers for inverse lithography problem*, Numer. Math. Theory Methods Appl., 16:820–846, 2023.
- [9] Y. Y. Chen, K. H. Chang, W. L. Cheng, and Y. P. Tang, *Curvilinear mask handling in OPC flow*, Journal of Micro/Nanopatterning, Materials, and Metrology, 23:011203, 2023.
- [10] Z. Chen and H. Wu, *Selected Topics in Finite Element Methods*, Science Press, 2010.

- [11] Y. Choi, A. Fujimura, and A. Shendre, *Curvilinear masks: An overview*, in: Photomask Technology 2021, Proceedings of the SPIE, Vol. 11855, 118550U-16, 2021.
- [12] S. K. Choy, N. Jia, C. S. Tong, M. L. Tang, and E. Y. Lam, *A robust computational algorithm for inverse photomask synthesis in optical projection lithography*, SIAM J. Imaging Sci., 5:625–651, 2012.
- [13] N. B. Cobb, *Fast Optical and Process Proximity Correction Algorithms for Integrated Circuit Manufacturing*, PhD Thesis, University of California, 1998.
- [14] L. B. da Veiga, A. Buffa, G. Sangalli, and R. Vázquez, *An Introduction to the Numerical Analysis of Isogeometric Methods*, in: IsoGeometric Analysis: A New Paradigm in the Numerical Approximation of PDEs. Lecture Notes in Mathematics, Vol. 2161, Springer, 87–154, 2016
- [15] C. de Boor, *A Practical Guide to Spline*, Springer, 1978.
- [16] D. A. Dunavant, *High degree efficient symmetrical Gaussian quadrature rules for the triangle*, Internat. J. Numer. Methods Engrg., 21:1129–1148, 1985.
- [17] J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, 1969.
- [18] Y. Granik, *Fast pixel-based mask optimization for inverse lithography*, Journal of Micro-Nanolithography MEMS and MOEMS, 5:043002, 2006.
- [19] K. Hooker, G. Xiao, Y. P. Tang, Y. Zhang, M. Jeong, J. Valadez, and K. Lucas, *Curvilinear mask solutions for full-chip EUV lithography*, in: Novel Patterning Technologies 2022, Proceedings of the SPIE, Vol. 12054, SPIE, 1205407, 2022,
- [20] J. Hu, A. Lyons, C. Spence, K. Wampler, M. Mohsen, Y. W. Lu, R. Gupta, Y. Zhang, R. Howell, J. Chang, J. Moon, and J. Ye, *Curvilinear data representation and its impact on file size and lithographic performance*, Journal of Micro/Nanopatterning, Materials, and Metrology, 23:011204, 2024.
- [21] W. Huang, Y. Li, H. Yang, M. Yuan, Z. Zhang, Z. Li, and Z. Li, *Fast and high-fidelity EUV curvilinear mask optimization by distance-versus-angle signature*, Opt. Express, 32(15):26292–26305, 2024.
- [22] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Dover Publications, 1994.
- [23] S. Kim, S. Zhang, and Y. Shin, *ML-guided curvilinear OPC: Fast, accurate, and manufacturable curve correction*, IEEE Trans. Semicond. Manuf., 38(1):19–28, 2025.
- [24] B. J. Lin, *Off-axis illumination – Working principles and comparison with alternating phase-shifting masks*, in: Optical/Laser Microlithography VI, Proceedings of the SPIE, Vol. 1927, SPIE, 89–100, 1993.
- [25] Y. Liu and A. Zakhor, *Optimal binary image design for optical lithography*, in: Optical/Laser Microlithography III, Proceedings of the SPIE, Vol. 1264, SPIE, 401–412, 1990.
- [26] Y. Liu, A. Zakhor, and M. A. Zuniga, *Computer-aided phase shift mask design with reduced complexity*, IEEE Trans. Semicond. Manuf., 9:170–181, 1996.
- [27] X. Ma and G. R. Arce, *Computational Lithography*, John Wiley & Sons, 2010.
- [28] X. Ma, L. Dong, C. Han, J. Gao, Y. Li, and G. R. Arce, *Gradient-based joint source polarization mask optimization for optical lithography*, Journal of Micro-Nanolithography MEMS and MOEMS, 14:023504, 2015.
- [29] X. Ma, Y. Li, and L. Dong, *Gradient-based resolution enhancement optimization methods based on vector imaging model*, in: Optical Microlithography XXV, Proceedings of the SPIE, Vol. 8326, SPIE, 83262B, 2012.
- [30] X. Ma, Y. Li, and L. Dong, *Mask optimization approaches in optical lithography based on a vector imaging model*, J. Opt. Soc. Am. A, 29:1300–1312, 2012.
- [31] L. Pang, *Inverse lithography technology: 30 years from concept to practical, full-chip reality*, Journal of Micro/Nanopatterning, Materials, and Metrology, 20:030901, 2021.

- [32] L. Pang and A. Fujimura, *Why the mask world is moving to curvilinear*, in: DTCO and Computational Patterning III, Proceedings of the SPIE, Vol. 12954, SPIE, 1295416, 2024.
- [33] L. Pang, D. Seal, T. Boettiger, N. Shirali, G. Dai, and A. Fujimura, *Curvilinear masks: Motivations and metrology*, in Photomask Technology 2024, Proceedings of the SPIE, Vol. 13216, SPIE, 132161K, 2024.
- [34] M. L. Rieger and J. P. Stirniman, *Using behavior modeling for proximity correction*, in Optical/Laser Microlithography VII, Proceedings of the SPIE, Vol. 2197, SPIE, 371–376, 1994.
- [35] B. E. A. Saleh and S. I. Sayegh, *Reduction of errors of microphotographic reproductions by optimal corrections of original masks*, Opt. Eng., 20(5):205781, 1981.
- [36] L. Schumaker, *Spline Functions: Basic Theory*, Cambridge University Press, 2007.
- [37] X. Shi, Y. Yan, C. Li, M. Xia, B. Pan, Y. Gao, and W. Yuan, *Can optical proximity correction solution be learned? The learning limit and a general learning framework*, Journal of Micro/Nanopatterning, Materials, and Metrology, 21:043203, 2022.
- [38] P. Wei, Y. Li, Z. Li, M. Yuan, and H. Yang, *Implicit function characterization of the curvilinear mask to realize parametric optical proximity correction with a neighborhood parallel tabu search*, Appl. Opt., 62:4848–4859, 2023.
- [39] A. K. Wong, *Optical Imaging in Projection Microlithography*, SPIE Press, 2005.
- [40] H. Yang, Y. Li, M. Yuan, Z. Li, W. Huang, and Z. Li, *Efficient curvilinear optical proximity correction using non-uniform b-spline curves*, Appl. Opt., 63:8818–8826, 2024.
- [41] H. Yang, Z. Li, M. Yuan, Z. Li, and Y. Li, *Fast curvilinear optical proximity correction adopting quasi-uniform B-spline curves*, in: Eighth International Workshop on Advanced Patterning Solutions (IWAPS 2024), Proceedings of the SPIE, Vol. 13423, SPIE, 134230S, 2024.