IOWA STATE UNIVERSITY
Digital Repository

Graduate Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2012

Line search based inverse lithography technique for mask design

Xin Zhao
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

 Part of the Computer Engineering Commons

Recommended Citation
Zhao, Xin, "Line search based inverse lithography technique for mask design" (2012). *Graduate Theses and Dissertations*. 12607.
https://lib.dr.iastate.edu/etd/12607

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

www.manaraa.com

Line search based

inverse lithography technique for mask design

by

Xin Zhao

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:

Chris Chu, Major Professor

Zhao Zhang

Gary Tuttle

Iowa State University

Ames, Iowa

2012

Copyright © Xin Zhao, 2012. All rights reserved.

www.manaraa.com

DEDICATION

To my wife and parents

www.manaraa.com

TABLE OF CONTENTS

www.manaraa.com

LIST OF TABLES

www.manaraa.com

LIST OF FIGURES

www.manaraa.com

www.manaraa.com

ACKNOWLEDGMENTS

I would like to use this opportunity to thank those who helped me on my research and this thesis.

First of all, thank Dr. Chris Chu to give me this great opportunity to study my Master Degree in Iowa State University. And thank him to bring me such a advanced research topic in VLSI CAD area. During these years, his very kind guidance shows me the way how to setup a research topic, how to model a research problem, and how to solve a problem with a creative method. He gave me so many advices that would be very helpful for the rest of my life.

Furthermore, I would also like to express my thanks to my other committee members, Dr. Zhao Zhang and Dr. Gary Tuttle, for their insightful advices. I learned much useful knowledge for my research topic from them including Dr. Zhao Zhang's course, CPRE581 Computer System Architecture, and Dr. Gary Tuttle's EE532 Microelectronic Fabrication Techniques. It is a great honor to study from and working along with them.

Finally, I would like to dedicate this thesis to my wife without whose support I would not have been able to complete this work.

www.manaraa.com

ABSTRACT

Following moore's law, microelectronic fabrication techniques have been developed to fabricate deep-submicron devices. Device feature size on wafer turns to be much smaller than the illumination source of nowadays widely used lithography equipments, which is 193 nm wavelength of UV(ultraviolet) light. Diffraction effects can not be avoided when transfer patterns from masks to wafers in the process of lithography because of the extremely small size of features. So the patterns transferred from masks to wafers surface are distorted very much, and it causes many problems, such as poly line end shortening or bridging which result in leakage or short circuit.

The industry has been investigating various alternatives, such as EUV(extreme ultraviolet) illumination source. However, the next generation of illumination source, EUV with a wavelength of about 13.5 nm, still has a long way to be put into practice. As a result, Resolution Enhancement Technology (RET) has been increasingly relied upon to minimize image distortions. In advanced process nodes, pixelated mask becomes essential for RET to achieve an acceptable resolution. In this thesis, we investigate the problem of pixelated binary mask design in a partially coherent imaging system. Similar to previous approaches, the mask design problem is formulated as a nonlinear program and is solved by gradient based search. Our contributions are four novel techniques to achieve significantly better image quality than state-of-the-art technology. First, to transform the original bound-constrained formulation to an unconstrained optimization problem, we propose a new non-cyclic transformation of mask variables to replace the well-known cyclic one. As our transformation is monotonic, it enables a better control in flipping pixels. Second, based on this new transformation, we propose a highly efficient

www.manaraa.com

line search based heuristic technique to solve the resulting unconstrained optimization problem. Third, we introduce a jump technique. As gradient based search techniques will get trapped at a local minimum, we introduce a new technique named jump in order to jump out of the local minimum and continue the search. It increases the chance to achieve a better result. Fourth, to simplify the optimization, instead of using widely used discretization regularization penalty technique, we directly round the optimized gray mask into binary mask for pattern error evaluation. Experiment results show that the results of state-of-the-art algorithm implemented by Ma and Arce [5] are 8.55% to 358.8% higher than ours.

www.manaraa.com

CHAPTER 1. OVERVIEW

1.1 Microelectronic Manufacture

A typical CMOS circuit structure includes PMOS, NMOS, metal wire, via, and dielectric parts etc. Fig. 1.1 shows a cross section view of a typical CMOS circuit.

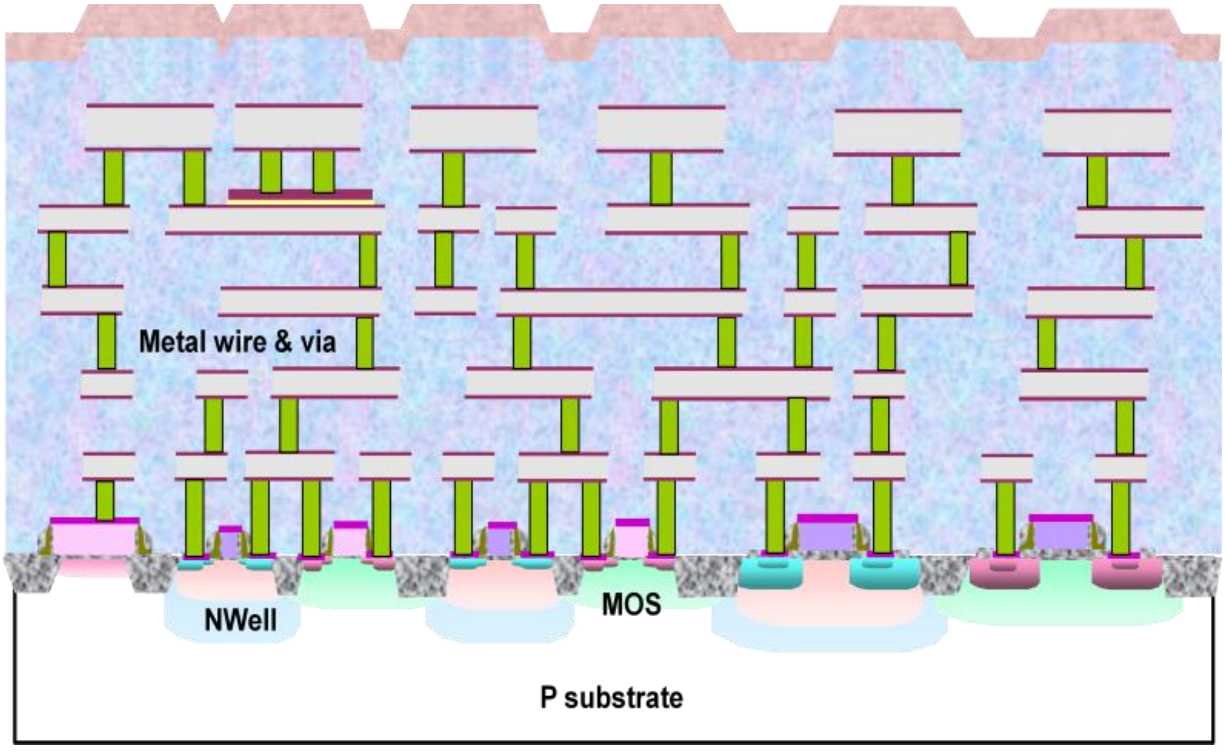


Figure 1.1 CMOS circuit cross section view.

For the manufacturing, as shown in the Fig. 1.2, a typical microelectronic manufacture flow starts from a bare wafer, then includes a serial of physical and chemical process such as diffusion, photo lithography, etching, thin film, chemical mechanical polishing,

www.manaraa.com

and implantation. Finally, microelectronic circuits are made on the wafer. After that, a testing(WAT and wafer sorting) is proceeded on the wafer with dies on it.

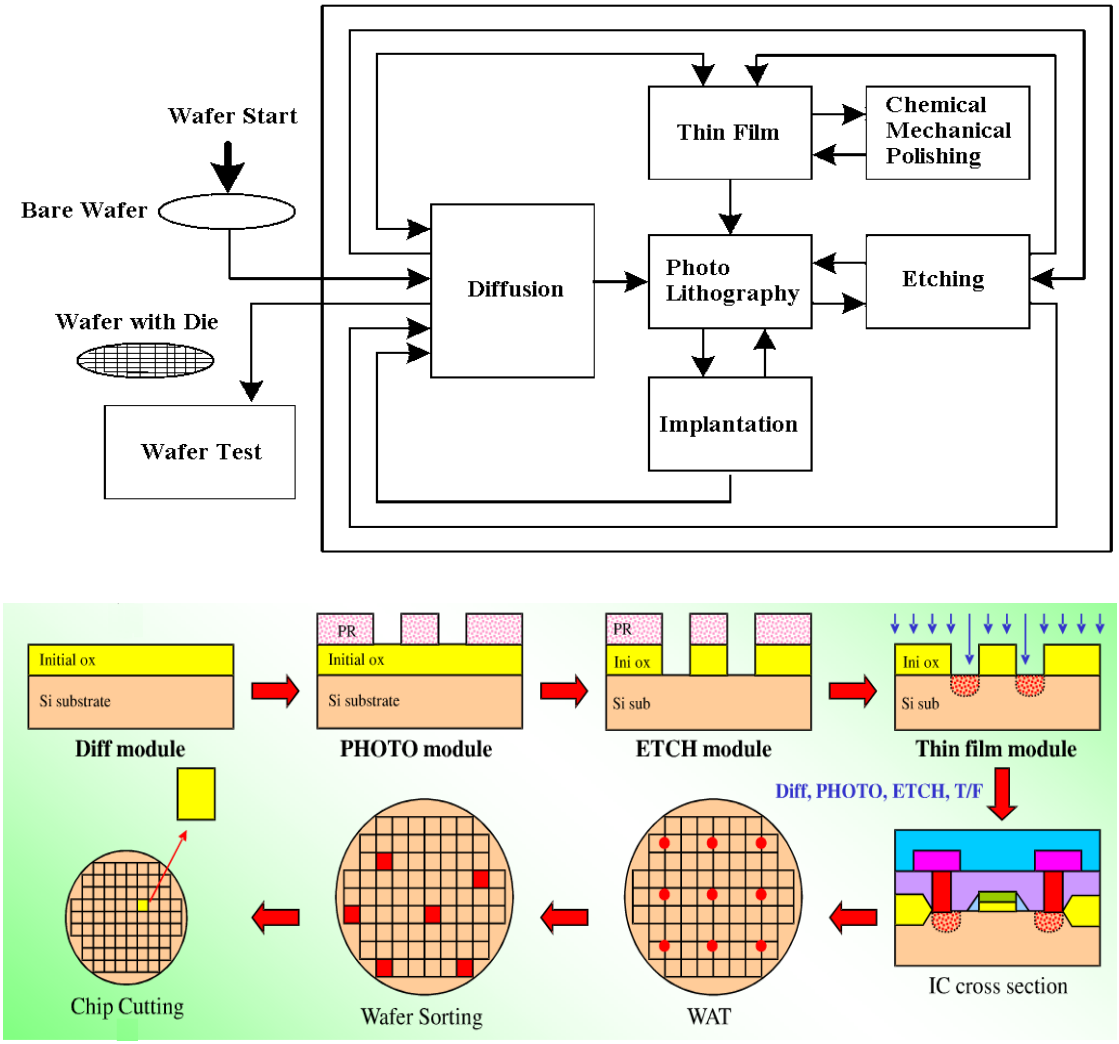


Figure 1.2 Microelectronic manufacture flow.

This thesis focuses on the problem of photo lithography process. For manufacturing microelectronic circuits on the wafer, the structure information of a circuit need to be transferred on to wafer surface in photo lithography process. Fig. 1.3 shows a detailed PMOS device with cross section view and top view. The top view information will be converted into layout then made into mask.

A typical photo lithography processing system is shown in Fig. 1.4. A photo mask is

www.manaraa.com

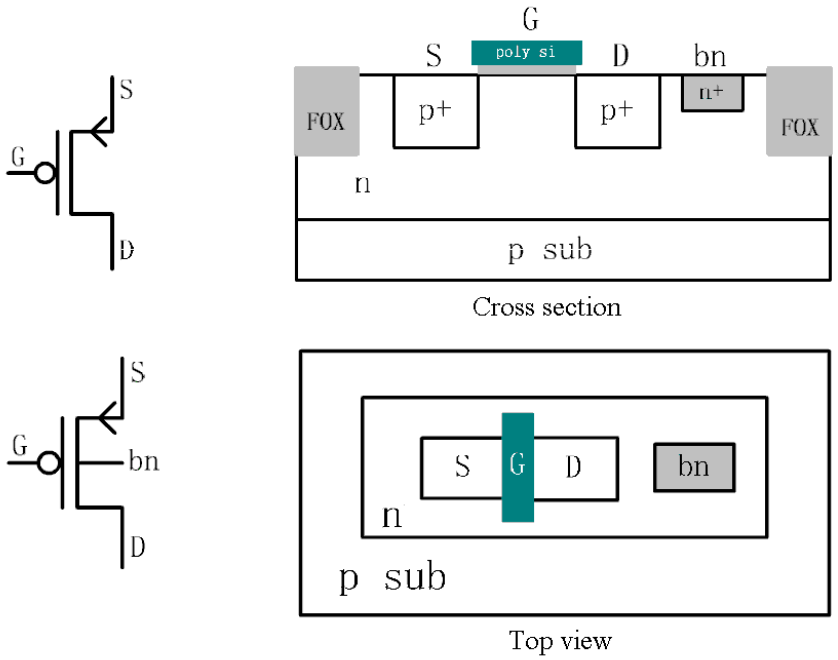


Figure 1.3 PMOS device.

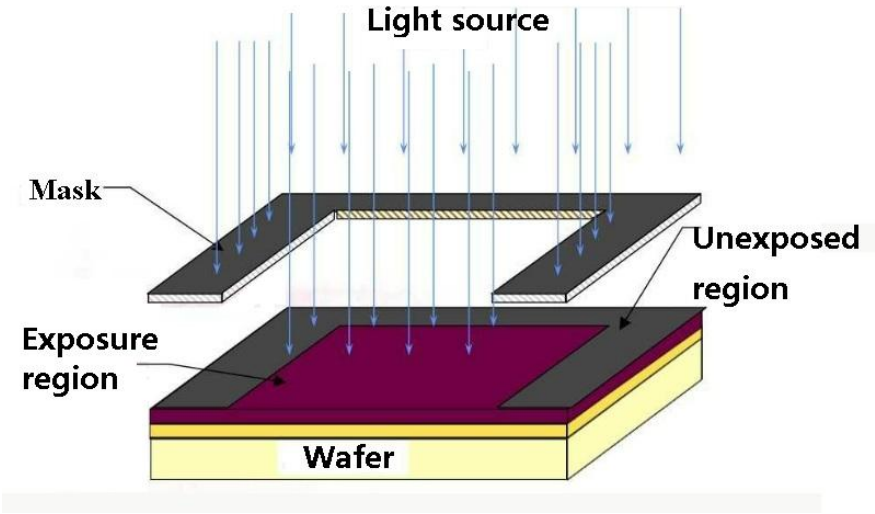


Figure 1.4 Typical photo lithography system.

www.manaraa.com

projected to the wafer which is covered by photoresists. Photoresists is a photosensitive materials. After photo-imaging and in developing process, it resists the action of certain chemicals in desired areas. In some area on wafer, if the light intensity received is below the threshold of the photoresist, the patterns will not be printed after development. Otherwise, the patterns can be printed on wafers.

1.2 Optical Proximity Correction (OPC)

As semiconductor manufacturers move to advanced process nodes (especially 45 nm process and below), lithography has become a great challenge due to the fundamental constraints of optical physics. Because feature size is much smaller than the wavelength of illumination source (currently 193 nm), the image formed on wafer surface is distorted more and more seriously due to optical diffraction and interference phenomena. In a typical optical lithography processing system, as illustrated in Fig. 1.5, if (a) is a target mask pattern, after projecting, (b) will be the aerial image(image intensity) on the wafer. Fig. 1.6, illustrates the model of photoresist with the threshhold value at 0.33 when diffraction happens through a pin hole at 0.0. The light intensity spreaded out of the desired region, a pin hole. Therefore, with this kind of threshold, the pattern in Fig. 1.6(c) would be printed on the wafer surface instead of Fig. 1.6(a).

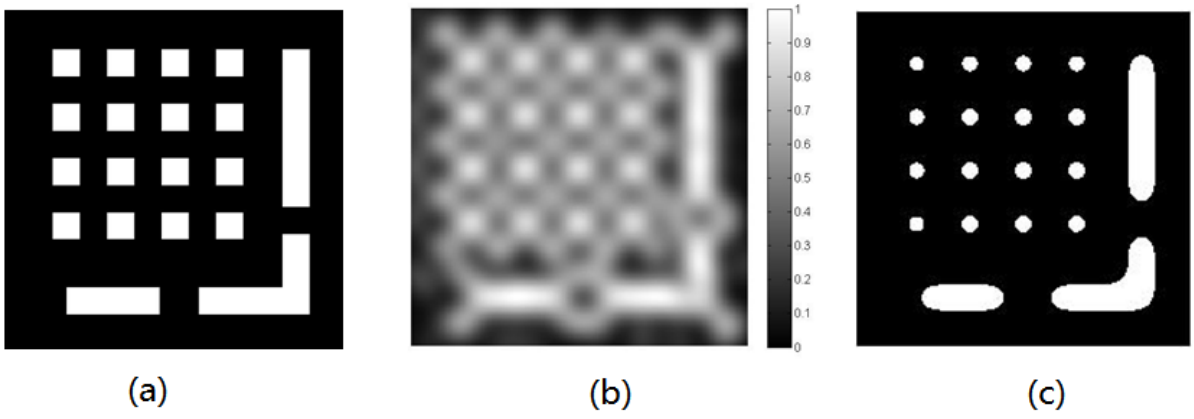


Figure 1.5 Target mask, image intensity and pattern after development

www.manaraa.com

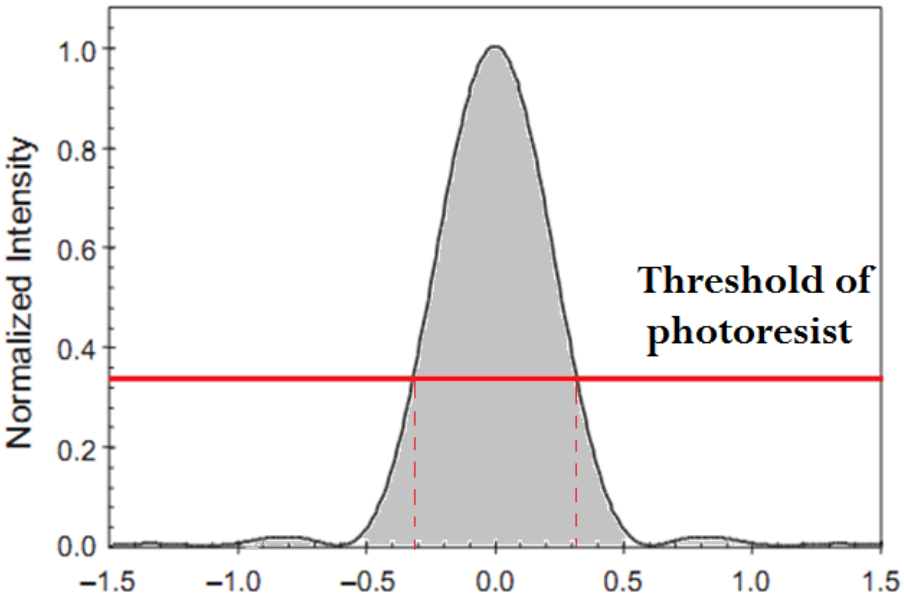


Figure 1.6 Photoresist with threshold of 0.33

The image transformed from a mask onto a wafer surface is distorted seriously due to diffraction of the light. As well known that diffraction occurs whenever the wavelength is on the order of or smaller than the diffracting objects' size. So when the light passes the small features on the mask which are a scale similar to or smaller than its wavelength, diffraction makes the result image on the wafer blurred, and the light intensity seems to be sprayed out of the original region where it should be printed on the wafer surface. The light does not travel in "ray" paths, as we usually assume, but can bend around the obstruction. Fig 1.7 illustrates the diffraction phenomena. Fig 1.8 shows a Matlab simulation result of a diffraction pattern of a pin hole in both black-and-white and color image. From Fig. 1.8, the light intensity distributed can be observed in rings. If this pattern is partitioned into a matrix, it can be called point spread function **h**, which means the light through a pin hole point will spread out by diffraction. And this matrix is also called spatial kernel for convolution. It can be calculated by look up table of Bessel function of the 1st kind.

While nowadays using 193nm wavelength of light source in the optical lithography processing system, the size of the features on the mask reached 45nm or even 22nm for

www.manaraa.com

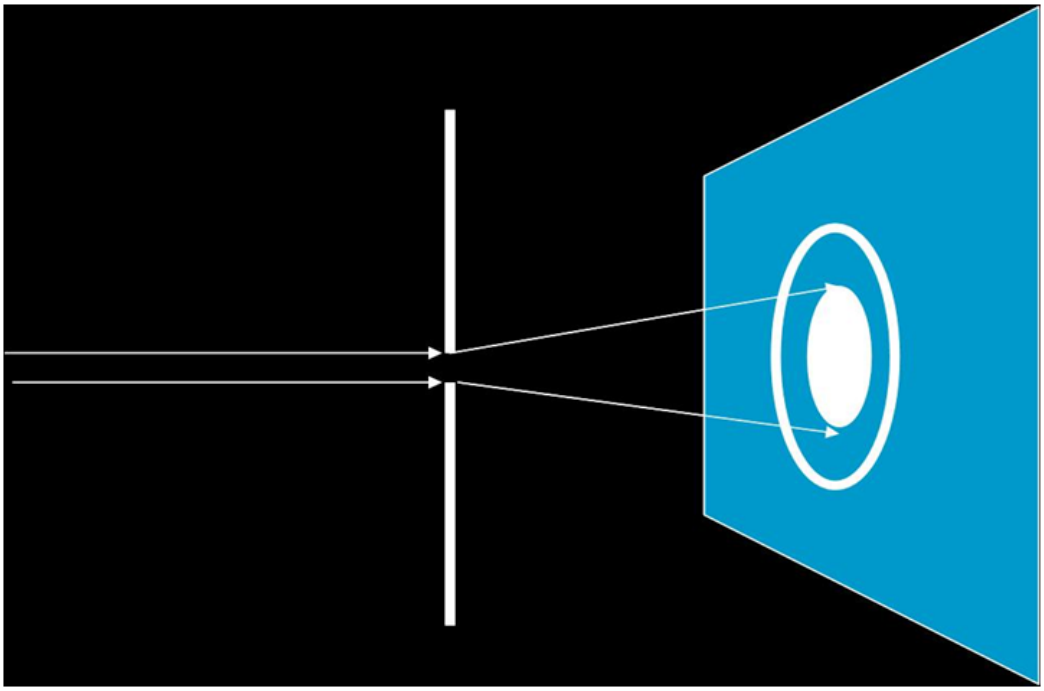


Figure 1.7 Diffraction phenomena of a pin hole

nowadays microelectronic manufacturing. The image transformed from a mask onto a wafer surface is distorted seriously due to diffraction of the light. This kind of problem bring a disaster to the microelectronic manufacturing by introducing line end shortening and corner rounding effect. These are instant damages or potential damages to the integrated circuits and a great threat to the reliability of the integrated circuits.

The industry has been investigating various alternatives (e.g., EUV lithography, E-beam lithography) but none of them is ready in foreseeable future. As a result, semiconductor manufacturers have no choice but to keep using the existing equipments in patterning the progressively smaller features.

Given the limitation of lithography equipments, and because the character of diffraction in optical lithography processing system can not be avoided, a method of modify the input mask before projecting is a most promising technology to improve the shape of the resulting image on the wafer.

The Resolution Enhancement Technology (RET) such as optical proximity correction (OPC), phase shift mask (PSM) and double patterning has been increasingly relied upon

www.manaraa.com

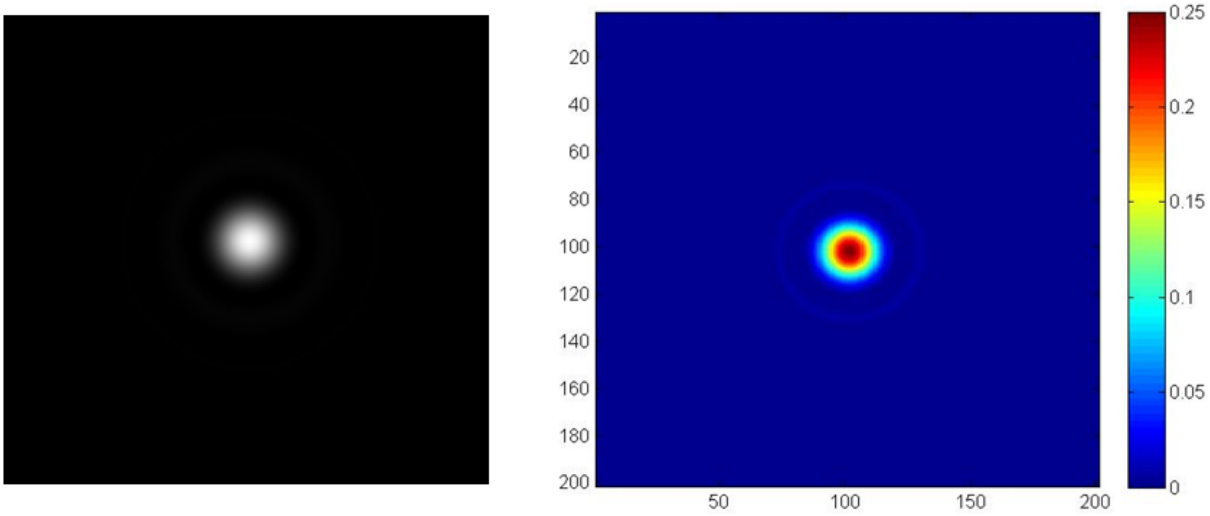


Figure 1.8 Diffraction pattern of a pin hole,or called spacial kernel **h**

to minimize image distortions [12]. In recent years, pixelated mask, which allows great flexibility in the mask pattern, has become essential for OPC to achieve better resolution. Fig. 1.9 [5] illustrates the OPC optimization. In the Fig. 1.9, the upper left one is the original target mask. When using this original mask, the projected result image pattern on the wafer will have 154 pixels pattern error because of diffraction, as shown in the bottom left one. That means, the original mask and its image on the wafer have 154 different pixel values. However, when using upper right one which is OPC optimized mask although it far away looks like the original mask, the pattern error of projected result will decrease to 40 pixels, shown in bottom right one. So OPC is a way to minimize image distortions. For the design of pixelated optimized mask, the most popular and successful approach is to formulate it as a mathematical program and solve it by gradient-based search [2, 9, 10, 3, 4, 6, 5]. Granik [2] considered a constrained nonlinear formulation. Poonawala and Milanfar [9, 8, 10] proposed a unconstrained nonlinear formulation and employed a regularization framework to control the tone and complexity of the synthesized masks. Ma and Arce [3, 5] presented a similar unconstrained nonlinear formulation targeting PSM. Ma and Arce [4, 5] focused on partially coherent illumination and used singular value decomposition to expand the partially coherent imaging equation

www.manaraa.com

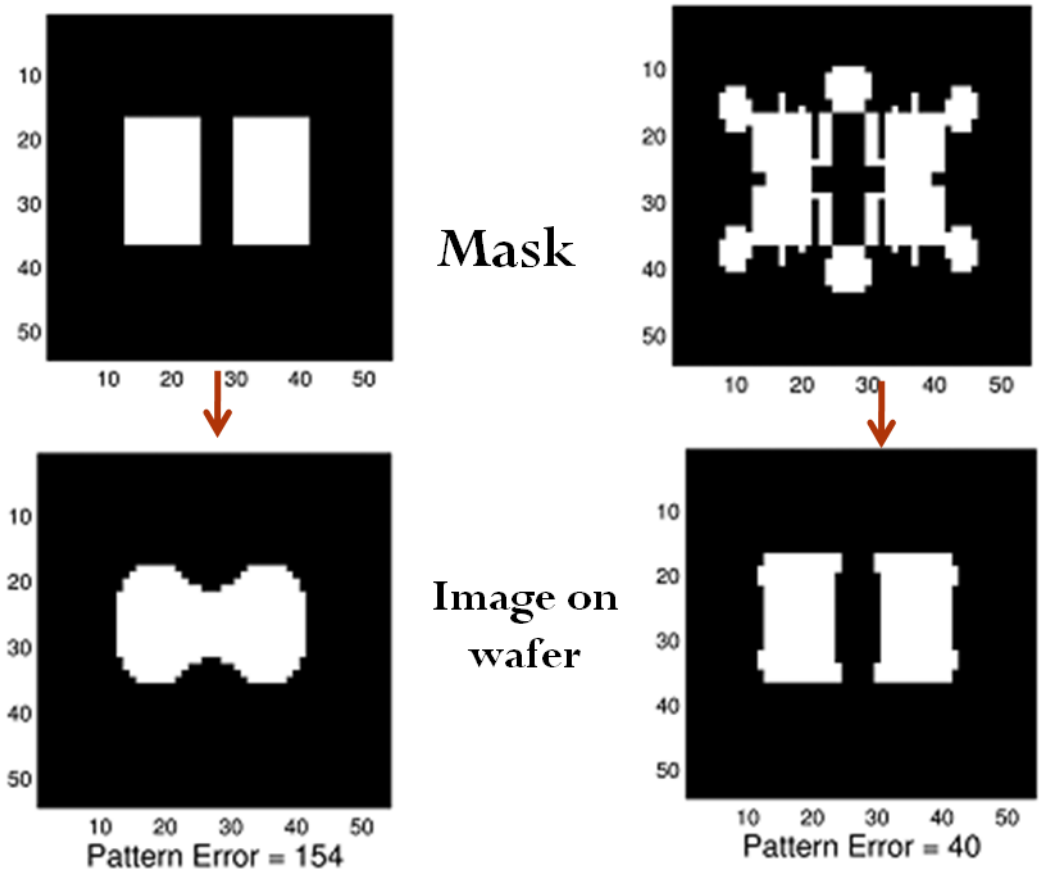


Figure 1.9 Optical Proximity Correction (OPC)

by eigenfunctions into a sum of coherent systems (SOCS). All works discussed above utilized the steepest descent method to solve the nonlinear programs, Ma and Arce [6] demonstrated that the conjugate gradient method is more efficient. Yu and Pan [13] is an exception to the mathematical programming approach. Instead, a model-based pixel flipping heuristic is proposed.

www.manaraa.com

1.3 Contributions of This Thesis

The inverse lithography technique for OPC mask design has been proposed in [2] and [8] in 2006, and has been widely discussed in recent years as semiconductor manufacturers move to advanced process nodes. But so far there is not an effective search method proposed because of the complicated solution space of this problem. We introduce a novel transformation for mask pixel, which enables an effective line search technique.

In this thesis, we focus on design of pixelated binary mask in a partially coherent imaging system[1]. Similar to previous approaches, we formulate the problem as an unconstrained non-linear program and solve it by iterative gradient-based search. The main contributions of this paper are listed below:

- To transform the problem formulation from a bounded optimization to an unconstrained one, we propose a new non-cyclic transformation of mask variables to replace the widely-used cyclic one. Our transformation is monotonic and allows a better control of flipping pixels.

- Based on this new transformation, we present a highly efficient line search based technique to solve the resulting unconstrained optimization. Because of the non-cyclic nature of the transformation, it is not easy for our line search to be trapped at local minimum. Therefore our algorithm can find much better binary masks for the inverse lithography problem.

- We introduce a jump technique. As gradient based search techniques will be trapped at a local minimum, we introduce a new technique named jump in order to jump out of the local minimum and continue the search.

- We apply a direct rounding technique to regularize gray masks into binary ones instead of adding a discretization regularization penalty to the cost function as

[1] The techniques proposed in this thesis can all be easily extended to PSM and other imaging systems.

www.manaraa.com

in [9] and [5]. This simplifies the computation and achieves better results as the experiment results show.

The rest of this thesis is organized as follows: The formulation of the inverse lithography for OPC problem is explained in Chapter 2. Chapter 3 describes in details the flow of our algorithm and the four novel techniques that we proposed. Chapter 4 presents the experimental results. The thesis is concluded in Chapter 5.

www.manaraa.com

CHAPTER 2. Problem Formulation

In an optical lithography system, a photo mask is projected to a silicon wafer through optical lens. An aerial image of the mask is then formed on the wafer surface, which is covered by photoresist. After developing and etching, a pattern similar to the one on the mask is formed on the wafer surface. To simulate the pattern formation on the wafer surface for a given mask, we first describe a projection optics model and a photoresist model below. After that, we present the formulation of the mask design problem. In this thesis, we use pixelated binary mask.

2.1 Projection Optics Model

The projection optics system is modeled by three kinds of imaging system. They are coherent imaging system, incoherent imaging system and partially coherent imaging system.

For the coherent imaging system, the illumination source is simplified to be just a single point. The intensity distribution of this kind of imaging system is modeled by Eq. (2.1), in which $\mathbf{I}$ is the aerial image, which means light intensity distribution when projecting a pixelated mask $\mathbf{M}$, $\mathbf{h}$ is the spatial kernel for convolution, and $\otimes$ indicates convolusion operation. This computation can also be illustrated in Fig.2.1, in which [.] indicates convolution. As mentioned in Chapter 1.2, the spatial kernel $\mathbf{h}$ can be calculated by look up table of Bessel function of the 1st kind and it physically is a diffraction image pattern of a pin hole illustrated by Fig. 1.8. For computation, it is also

www.manaraa.com

partitioned into pixelated image pattern, so that both the mask $\mathbf{M}$ and spatial kernel $\mathbf{h}$ turn to be matrixes for convolution. For the square operation, it means each entry of the matrix is squared.

$$\mathbf{I}(\mathbf{M}) = |\mathbf{h} \otimes \mathbf{M}|^2. \tag{2.1}$$

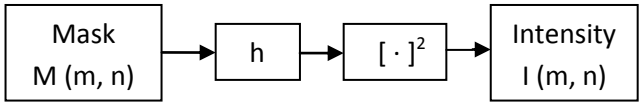


Figure 2.1 Computation model for coherent imaging system

Because the spatial kernel $\mathbf{h}$ is central symmetry, and if the mask is partitioned into $M_{(m,n)}$, the original convolution method is a dot product of all the neighbors of source pixel in $M_{(m,n)}$ with spatial kernel $\mathbf{h}$. As an example illustrated in Fig. 2.2, the entry with value of 4 in the result image matrix $I_{(m,n)}$ is calculated by

$$0*2+0*0+0*2+0*0+1*4+1*0+0*2+1*0+0*2=4 \tag{2.2}$$

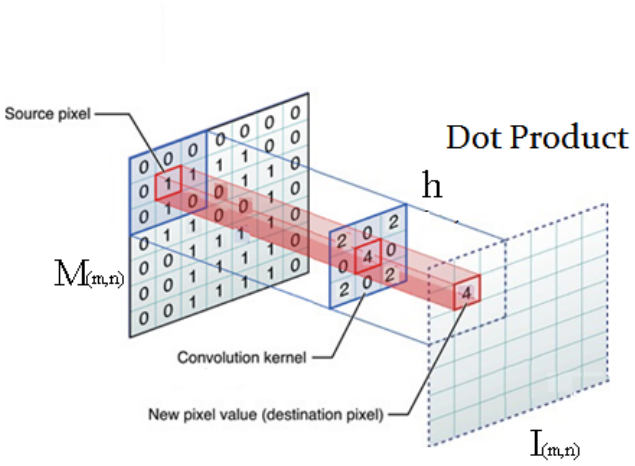


Figure 2.2 Original convolution method

In an incoherent imaging system, the illumination source is of infinite extent. The intensity distribution is modeled by Eq. (2.3)

www.manaraa.com

$$\mathbf{I}(\mathbf{M}) = |\mathbf{h}|^2 \otimes |\mathbf{M}|^2. \tag{2.3}$$

Finally, in a partially coherent imaging system, the illumination source has a nonzero but limited area. Common partially coherent illumination sources include dipole, quadrupole, and annular illumination, shown as Fig. 2.3

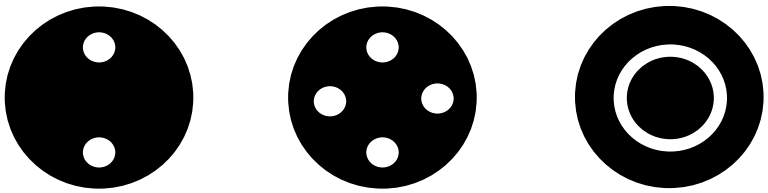


Figure 2.3 Dipole, quadrupole, and annular illumination source

The Hopkins diffraction model[1] is widely used to approximate this kind of optics systems. To reduce the computational complexity of the Hopkins diffraction model, the Fourier series expansion model [11] is a common approach. In this thesis, we followed this model.

The Fourier series expansion model approximates the partially coherent imaging system as a sum-of-coherent system (SOCS). Based on this model, the computation of the aerial image $\mathbf{I}$ of a pixelated mask $\mathbf{M}$ is given in Eq. (2.4) and illustrated in Fig. 2.4 [5]. In here, the dimensions of the pixelated mask and the image are $m \times n$. The illumination source is partitioned into $N \times N$ sources. u and $\mathbf{h}$ are the Fourier series coefficients and spatial kernels, respectively.

$$\mathbf{I}(\mathbf{M}) = \sum_{p=1}^{N \cdot N} u_p |\mathbf{h_p} \otimes \mathbf{M}|^2. \tag{2.4}$$

Note that the convolution $\mathbf{h} \otimes \mathbf{M}$ can be achieved in frequency domain using Fast Fourier transform $\mathcal{FFT}$ and inverse Fast Fourier transform $\mathcal{FFT}^{-1}$ as shown below

$$\mathbf{I}(\mathbf{M}) = \sum_{p=1}^{N \cdot N} u_p |\mathcal{FFT}^{-1}\{\mathcal{FFT}(\mathbf{h_p}) \cdot \mathcal{FFT}(\mathbf{M})\}|^2. \tag{2.5}$$

www.manaraa.com

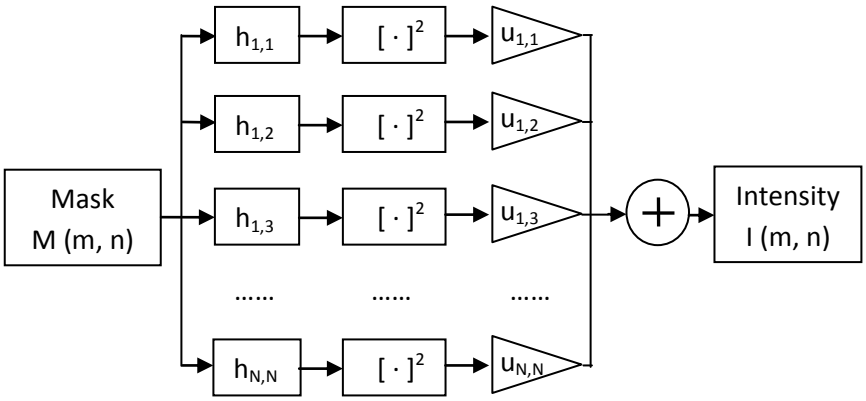


Figure 2.4 Fourier series expansion model for partially coherent system.

2.2 Photoresist Model

To model the reaction of the photoresist to the intensity of light projected on it, we use the constant threshold model below

$$z_i = \begin{cases} 0 & \text{if } I_i \leq t_r \\ 1 & \text{if } I_i > t_r \end{cases} . \tag{2.6}$$

where I_i and z_i are the light intensity and the corresponding reaction result of the photoresist at pixel i on the wafer surface, respectively, and t_r is the threshold of the photoresist. This means if the light intensity received at pixel i is higher then the threshold t_r, the photoresist will react and the image pattern will be printed on this pixel.

Thus the pattern $\mathbf{z}$ formed on the wafer surface can be expressed as a function of the mask $\mathbf{M}$ based on Eq. (2.5) and Eq. (2.6). In order to make $\mathbf{z}$ differentiable so that gradient based search can be applied, we approximate the constant threshold model above with the sigmoid function

$$sigmoid(x) = \frac{1}{1 + e^{-a(x-v)}}. \tag{2.7}$$

where the parameter a determines the steepness of the sigmoid function around $x = v$.

www.manaraa.com

The larger value of a is, the steeper and hence the closer to the constant threshold model the sigmoid function will be. The sigmoid function with $a = 10$, $v = 0$ is illustrated in Fig. 2.5.

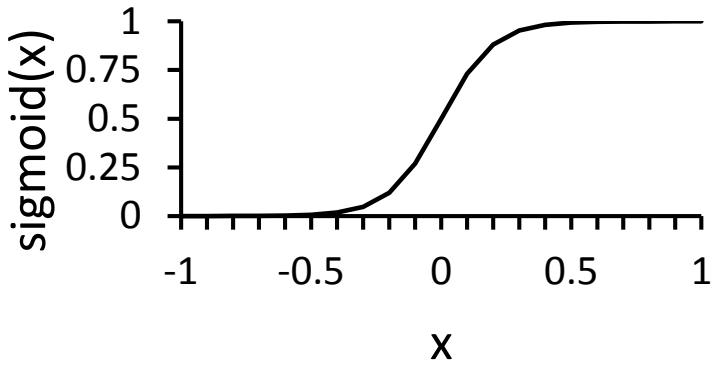


Figure 2.5 The sigmoid function with $a = 10$ and $v = 0$.

Using the sigmoid function, the reaction of the photoresist at pixel i for a mask $\mathbf{M}$ is

$$z_i(\mathbf{M}) \approx \frac{1}{1 + e^{-a(I_i(\mathbf{M}) - t_r)}}. \tag{2.8}$$

2.3 Our Inverse Lithography Problem Formulation

Inverse lithography treats mask design as an inverse problem of imaging. Given a target pattern $\hat{\mathbf{z}}$, the problem is to find a mask $\mathbf{M}^*$ such that the corresponding pattern $\mathbf{z}(\mathbf{M}^*)$ on the wafer surface is as close to $\hat{\mathbf{z}}$ as possible [7].

The error between the target pattern $\hat{\mathbf{z}}$ and the generated pattern $\mathbf{z}(\mathbf{M})$ for any mask $\mathbf{M}$ is commonly defined as

$$E(\mathbf{M}) = \sum_{i=1}^{m \cdot n} (\hat{z}_i - z_i(\mathbf{M}))^2. \tag{2.9}$$

So the inverse lithography problem is to find a mask to minimize the error between the target pattern and the generated pattern. It is formulated as

$$\mathbf{M}^* = \underset{\forall i, M_i \in \{0,1\}}{\arg\min} \; E(\mathbf{M}). \tag{2.10}$$

www.manaraa.com

Combining Eq. (2.8) and Eq. (2.9) with Eq. (2.10), the problem is written as

$$\mathbf{M}^* = \underset{\forall i, M_i \in \{0,1\}}{\arg\min} \sum_{i=1}^{m \cdot n} \left(\hat{z}_i - \frac{1}{1 + e^{-a(I_i(\mathbf{M}) - t_r)}} \right)^2 . \tag{2.11}$$

where $I_i(\mathbf{M})$ is the light intensity at pixel location i calculated by Eq. (2.5).

www.manaraa.com

CHAPTER 3. Line Search Based Inverse Lithography Technique

As the value of each pixel M_i should be 0 or 1, the inverse lithography problem is an integer non-linear program. To make it easier to solve, a common approach is to relax the integer constraints to $0 \leq M_i \leq 1$ for all i [2, 9, 10, 3, 4, 6]. Therefore, the problem becomes a bounded non-linear program. To further simplify the program, it is also common to convert it into an unconstrained non-linear program [2, 9, 10, 3, 4, 6]. It is achieved by a transformation $M_i = T(\beta_i)$ which maps an unbounded variable β_i into the range $[0, 1]$. (We will discuss this transformation in Section 3.1.) The program is then solved with respect to $\boldsymbol{\beta}$'s domain.

This unconstrained non-linear program can be solved by an iterative gradient-based search method. Starting from some point $\boldsymbol{\beta}$ in the solution space, a search direction, which can be decided based on the gradient of Eq. (2.9), is first found. Then a step of a certain size along the search direction is taken. Thus a new point, which hopefully has less pattern error, is reached. The search is repeated until the error cannot be further reduced.

In this thesis, we apply this iterative gradient-based search method, which is outlined in Fig. 3.1 [1]. Our contributions are four novel techniques as described in Sections 3.1, 3.2, 3.3 and 3.4 to reduce pattern error over previous works.

In particular, we use the steepest descent method, i.e., the search direction is the negative of the local gradient of Eq. (2.9). But our techniques are not limited only to

[1] Note that specific details about our algorithm are given in the comments in Fig. 3.1.

www.manaraa.com

steepest descent method. It should be applicable to other iterative gradient-based search approaches like conjugate gradient method.

1. Transform initial mask into $\boldsymbol{\beta}$ // Sec. 3.1
2. Repeat
3. Find the search direction $\mathbf{d}$ at $\boldsymbol{\beta}$ // Eq. (3.4)
4. Determine the step size S // Sec. 3.2
5. $\boldsymbol{\beta}^{new} = \boldsymbol{\beta} + S \times \mathbf{d}$
6. Generate *gray* mask $\mathbf{M} = T(\boldsymbol{\beta}^{new})$ // Eq. (3.3)
7. // Round $\mathbf{M}$ to binary as described in Sec. 3.4
8. Evaluate pattern error $\mathbf{E}(\mathbf{M})$ // Eqs. (2.5), (2.6) & (2.9)
9. $\boldsymbol{\beta} = \boldsymbol{\beta}^{new}$, may have a jump // Sec. 3.3
10. Until stop criteria is reached

Figure 3.1 Generic framework for iterative gradient-based search.

3.1 Novel Transformation for Mask Pixel

As explained above, to convert the inverse lithography problem into an unconstrained optimization problem, we need a transformation $T : \mathcal{R} \to [0, 1]$. Then we can use an unbounded variable β_i to represent each pixel based on $M_i = T(\beta_i)$.

One such transformation is proposed by Poonawala and Milanfar [9]:

$$M_i = \frac{1 + \cos \beta_i}{2}. \tag{3.1}$$

This idea is widely adopted by later works [10, 3, 4, 6]. We call it the cosine transformation.

In gradient-based search, a line search along the search direction is typically performed to determine the step size S to get to a local minimum (step 4 in Fig. 3.1). The line search will be more effective if the function $\mathrm{E}(\mathbf{M})$ along the search direction is smooth and, better yet, convex. However, the cosine transformation is a cyclic function. It is clearly not an one-to-one transformation. By increasing the value of β_i, M_i changes

www.manaraa.com

its value between 0 and 1 periodically. As a result, when $\boldsymbol{\beta}$ is moving along any direction, E($\mathbf{M}$) may keep jumping up and down as M_i keeps switching between 0 and 1.

To illustrate this, we consider the algorithm described in Chapter 7 of Ma and Arce [5], which solved the same problem formulation as our paper. It also applied the steepest descent method but it used the cosine transformation. The pattern error function Eq. (2.9) is turned into Eq. (3.2) below

$$E(\boldsymbol{\beta}) = \sum_{i=1}^{m \cdot n} \left(\hat{z}_i - \frac{1}{1 + e^{-a(\sum_{p=1}^{N \cdot N} u_p |\mathbf{h_p} \otimes \frac{1+\cos \boldsymbol{\beta}}{2}|^2 - t_r)}} \right)^2. \tag{3.2}$$

Using the software code and the target pattern (as shown in Fig. 3.2) provided by [5], when $\boldsymbol{\beta}$ is moved along the negative gradient direction of Eq. (3.2), the function E($\boldsymbol{\beta}$) is illustrated in Fig. 3.3 and Fig. 3.4. It shows that the function changes in a very chaotic manner. We have observed a large number of experiments on different target patterns and different current masks. The function E($\boldsymbol{\beta}$) always shows a similar chaotic behavior. It makes line search very difficult. In theory, the negative gradient points out the direction each pixel should be adjusted to achieve the minimal value of pattern error. However, the gradient only provides the direction of change at the local point. Because of the cyclic property of Eq. (3.1), the pixels on the mask may be flip to the wrong direction if step size S is not set appropriately. This makes the gradient-based search method very ineffective. In fact, the common practice in previous works [9, 10, 3, 4, 6] is to set the step size to some fine-tuned constant instead of computed by line search.

We propose a new transformation for M_i based on the sigmoid function (Eq. (2.7)):

$$M_i = \frac{1}{1 + e^{-A(\beta_i - T_R)}}. \tag{3.3}$$

where A is the steepness control parameter and T_R specifies the transition point of the function. A larger A will cause the pixel values to be closer to 0 or 1. T_R can be set to any value and is set to 0 in this paper. We call this the sigmoid transformation. As the sigmoid transformation is a strictly increasing function, when $\boldsymbol{\beta}$ is moved along any direction, each mask pixel is flipped at most once.

www.manaraa.com

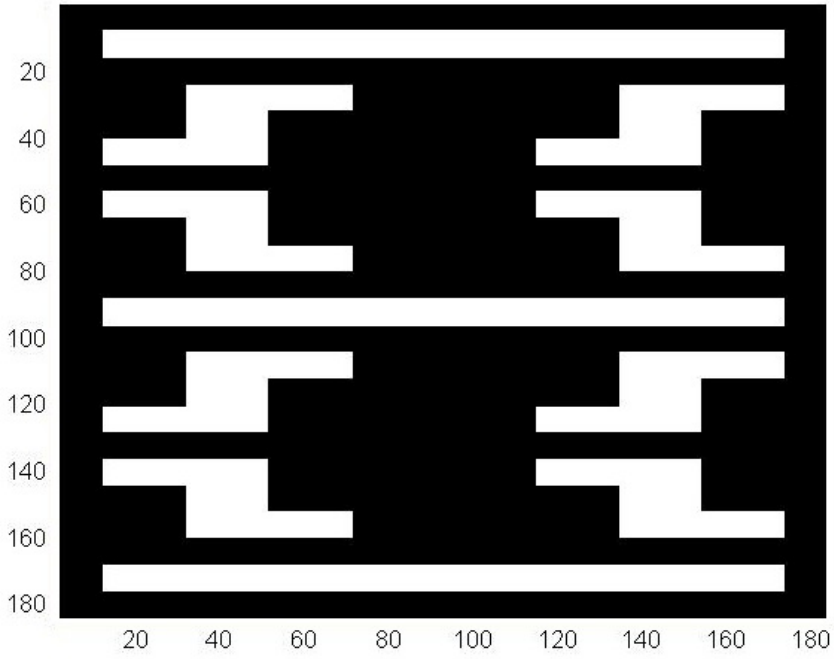


Figure 3.2 Target pattern from [5] with 184×184 pixels.

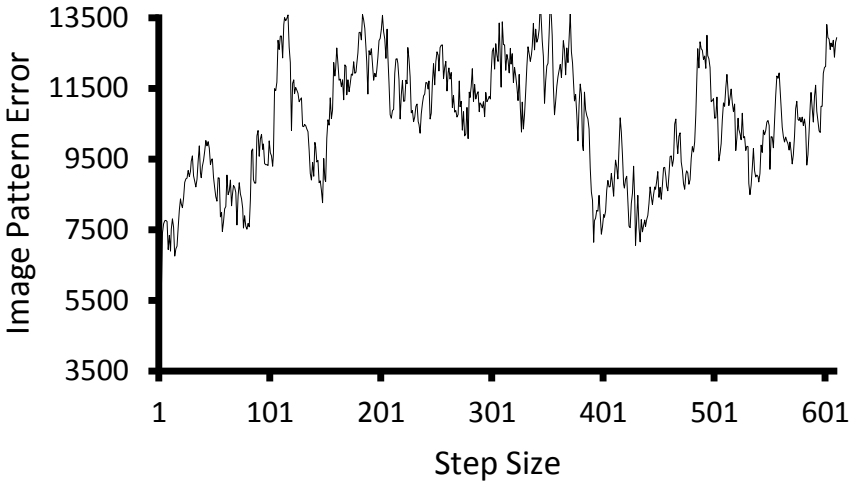


Figure 3.3 Pattern error based on cosine transformation.

www.manaraa.com

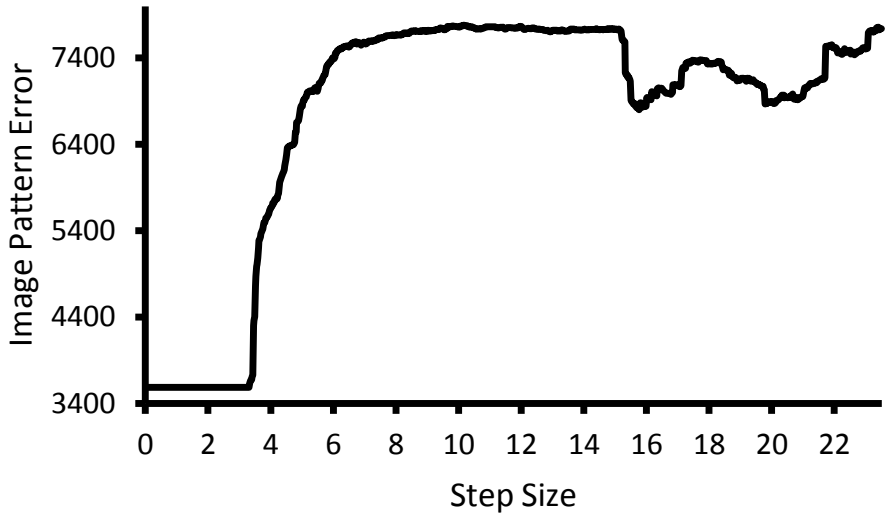


Figure 3.4 Enlarged version of Fig. 3.3 with step size from=0 to 25.

Based on the sigmoid transformation, the gradient of Eq. (2.9) is

$$\begin{aligned}\frac{\partial E(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -a \cdot A \cdot \{\sum_{p=1}^{N \cdot N} u_p[(\mathbf{h_p} \otimes \mathbf{M}) \odot (\hat{\mathbf{z}} - \mathbf{z}) \odot \mathbf{z} \\ \odot (1 - \mathbf{z})] \otimes {\mathbf{h_p}}^{*T}\} \odot \mathbf{M} \odot (1 - \mathbf{M}) - a \cdot A \\ \cdot \{\sum_{p=1}^{N \cdot N} u_p[(\mathbf{h_p}^* \otimes \mathbf{M}) \odot (\hat{\mathbf{z}} - \mathbf{z}) \odot \mathbf{z} \odot (1 - \mathbf{z})] \\ \otimes {\mathbf{h_p}}^{T}\} \odot \mathbf{M} \odot (1 - \mathbf{M}) \end{aligned} \tag{3.4}$$

where $1 = [1, ..., 1]^T \in \mathcal{R}^{m \times n}$, $\odot$ is the element-by-element multiplication operator and $\mathbf{h_p^*}$ is the conjugate of $\mathbf{h_p}$. We have performed a large number of experiments on different target patterns and different current masks. When $\boldsymbol{\beta}$ is moved along the negative gradient direction, the function E($\boldsymbol{\beta}$) is almost always unimodal. One typical example is shown in Fig. 3.5 and Fig. 3.6. (Note that not every pixel can be flipped along the negative gradient direction, as we will explain in Section 3.2.) This makes it feasible to apply line search to heuristically minimize the pattern error. Note that gradient calculation is very expensive due to the number of convolutions in Eq. (3.4). Hence once a gradient is calculated, it is desirable to perform line search to minimize the pattern error as much as possible in order to reduce the number of iterations (i.e., gradient calculations) of the

www.manaraa.com

gradient-based search algorithm. As Fig. 3.6 shows, by performing line search along the negative gradient direction, the image pattern error can be effectively reduced from around 3600 to below 3100, about 14%, in one iteration.

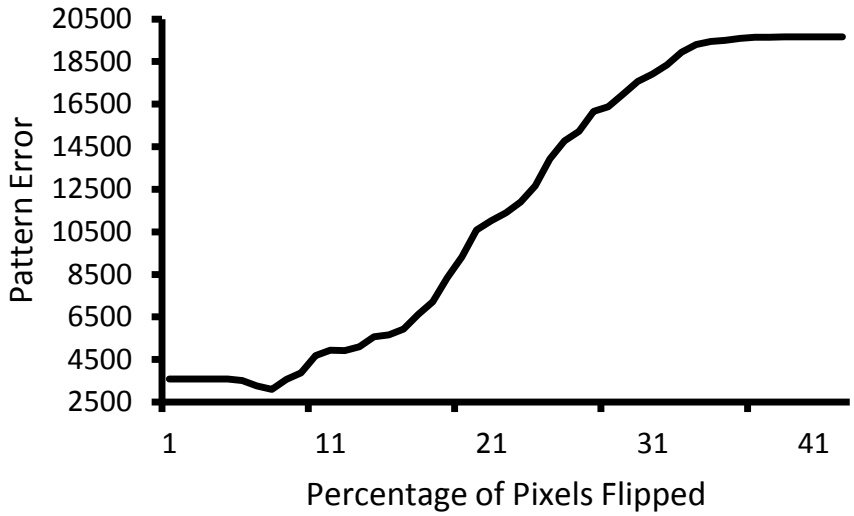


Figure 3.5 Pattern error based on sigmoid transformation.

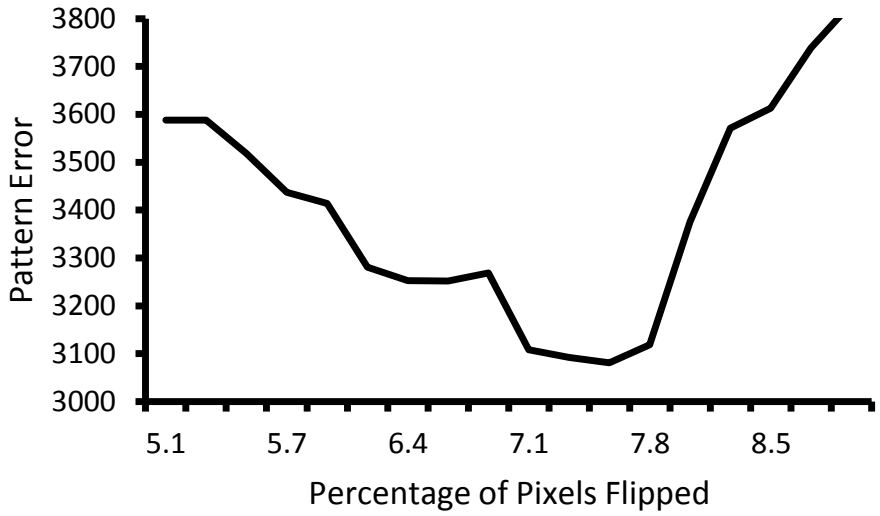


Figure 3.6 Enlarged version of Fig. 3.5 with percentage from 5.1% to 9.0%.

www.manaraa.com

3.2 Highly Efficient Line Search Technique

In this section, we present a highly efficient line search technique to determine the step size in Step 4 of the algorithm in Fig. 3.1 to minimize pattern error. We observe that in each iteration, the shape of the function E($\boldsymbol{\beta}$) along the direction of negative gradient is almost always like the curve shown in Fig. 3.5. We employ golden section method for line search. Golden section search is an iterative technique which successively narrows the search range.

Because the final optimized mask should be a binary one, we need to round the gray mask, which is given by Eq. (3.3), to binary according to some rounding threshold t_m. In other words,

$$M_i^{binary}(\beta_i) = \begin{cases} 0, & M_i(\beta_i) < t_m \\ 1, & M_i(\beta_i) \geq t_m \end{cases}, \tag{3.5}$$

where $\mathbf{M^{binary}}$ is the resulting binary mask. Here we simply set t_m to 0.5.

When moving along the negative gradient direction, as the value of each pixel M_i is changed monotonically due to our new transformation, by setting a step size S, we can easily control the number of pixels flipped (i.e., changed from below t_m to above t_m or vice versa) during line search. This idea is explained below.

Given the current mask specified by $\boldsymbol{\beta}$ and the negative gradient direction $\mathbf{d}$, Eq. (3.3) can be written as a function of S as

$$M_i(S) = \frac{1}{1 + e^{-A(\beta_i + S \times d_i - T_R)}}. \tag{3.6}$$

By substituting Eq. (3.6) into Eq. (3.5) and rearranging, we get

If $d_i > 0$

$$M_i^{binary}(S) = \begin{cases} 0, & S < S_i \\ 1, & S \geq S_i \end{cases}, \tag{3.7}$$

www.manaraa.com

If $d_i < 0$

$$M_i^{binary}(S) = \begin{cases} 0, & S > S_i \\ 1, & S \leq S_i \end{cases}, \tag{3.8}$$

where

$$S_i = \frac{\beta_i - \frac{lg(1/t_m - 1)}{-A} - T_R}{-d_i}. \tag{3.9}$$

is the threshold on step size for flipping pixel i. At the current mask, if a pixel's value is less than t_m and its negative gradient is positive, or if a pixel's value is larger than t_m and its negative gradient is negative, then the pixel will be flipped when we apply a step size S larger than S_i. Other pixels are un-flippable no matter how large step size S we use. So it is easy to determine how many pixels can be flipped. To control the number of pixels flipped during golden section search, we first mark all flippable pixels along the negative gradient direction. Then we calculate the threshold on step size, S_i, for each flippable pixel i. By sorting these thresholds from smallest to largest, the number of pixels flipped can be controlled by setting the value of S. For example, by using the 50th value of the sorted thresholds as the step size S, 50 pixels will be flipped along the negative gradient. In golden section search, the minimum and maximum sorted thresholds can be used to define the search region. In this thesis, we use a segment in the region from the minimum to maximum sorted thresholds as our search region. The detail will be discussed in Section 4

3.3 Jump Technique

By comparing the Fig. 3.3 and Fig. 3.5, it is so obvious that by applying our new sigmoid transformation, along the negative gradient direction, our efficient algorithm of

www.manaraa.com

line search technique is not so easy to be trapped into local minima. Because of the non-cyclic nature of our transformation, the solution space is not so rugged. However, the solution space is still extremely complicated with many local minima because of the complexity of the problem itself. As that gradient based search techniques will be trapped at a local minimum, we introduce a new technique named jump in order to jump out of the local minimum and continue the search. A simple illustration in 1-dimension is shown in Fig. 3.7. That is, during the line search process, if the algorithm cannot find a better solution along the search direction (i.e., be trapped at some local minimum), instead of terminating, it will jump along the search direction with a large step size to a probably worse solution. Then the algorithm will continue the gradient based search starting from the new solution. If the step size is large enough, it is likely that the algorithm will not converge to the previous local minimum. At the end, the algorithm will return the best local minimum that it has found.

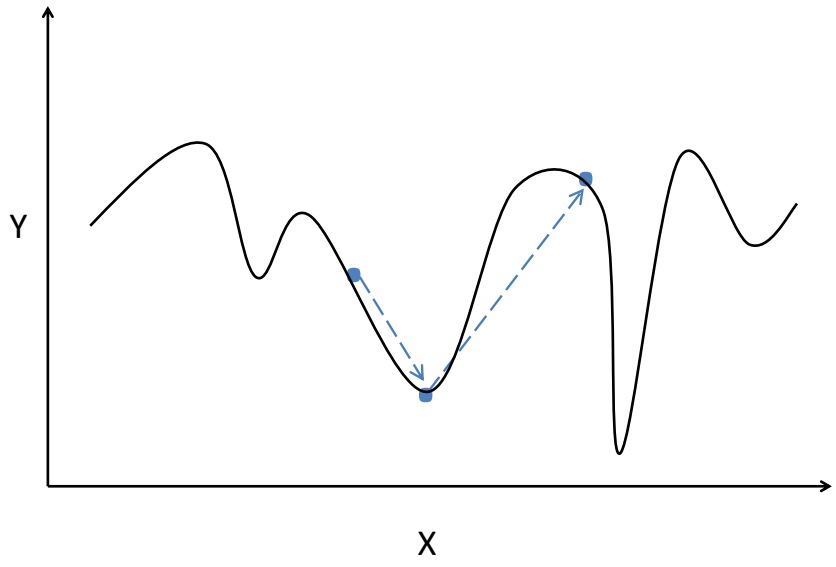


Figure 3.7 Jump technique illustrated in 1-dimension

www.manaraa.com

3.4 Directly Rounding of Gray Mask

In order to apply gradient-based approach, it is unavoidable to relax the integer constraints as discussed at the beginning of Chapter 3. As a result, the optimized mask becomes a gray one. Because our goal is to generate a binary mask, the optimized gray mask has to be rounded to a binary one at last. A regularization framework was proposed in Section IV.A of [9] and also in Chapter 6.1 of [5] to bias the output gray mask to be closer to binary. This regularization approach adds to the objective function (i.e., Eq. (2.9)) a quadratic penalty term for each pixel. However, it is still hard to control the change in the image pattern error caused by the rounding of the gray mask at the end. The optimized gray mask may achieve a low pattern error. However, after rounding the gray mask into binary, the pattern error often increases dramatically. Instead of using the quadratic penalty regularization framework, we propose to directly round the optimized gray mask into a binary one *in each iteration* before evaluating the pattern error. In this way, we simplify the objective function and also guarantee that our search will not be misled by inaccurate pattern error values.

www.manaraa.com

CHAPTER 4. Experimental Results

We compare an implementation of our algorithm with the program developed by Ma and Arce [5]. Both of the programs are coded in Matlab and executed on an Intel Xeon(R) $X5650$ 2.67GHz CPU. The runtime reported is CPU time and the programs are restricted to use a single core when running in Matlab.

In [5], the program uses cosine transformation and a preset step size of 2. Besides, it applies regularization with a quadratic penalty term as mentioned in Section 3.4 and complexity penalty term which can be found in Section IV.B of [9] and also in Chapter 6.2 of [5]. To have a fair comparison, our program applies the complexity penalty regularization too. But as we mentioned in Section 3.4, our program does not apply the quadratic penalty regularization, but directly rounds the optimized gray mask into binary one instead whenever pattern error is evaluated. In [5], the stopping criteria of gradient-based search is set in an ad hoc manner according to the target pattern. In order to fairly compare the two programs on various masks, the same stopping criteria is applied to both programs. The criteria is that if the average pattern error over the last 30 iterations is larger than the average pattern error over the 30 iterations before that, the program will stop. For evaluation of pattern error in both programs, in each iteration, the optimized gray mask is rounded using Eq. (3.5). We use the same convolution kernel $\mathbf{h}$ as the Matlab program of [5].

For the photoresist model, we use $a = 25$ and $tr = 0.19$ for the sigmoid function in Eq. (2.8). For the transformation of mask variables from $\boldsymbol{\beta}$, we use A=4 and T_R=0 for the sigmoid function in Eq. (3.3). The threshold t_m in Eq. (3.5) is set to 0.5.

www.manaraa.com

For our program, based on our observation of many experiments, for the first iteration of gradient-based search, the minimum pattern error can almost always be achieved in the region by flipping less than 10% of all pixels. One example is showed in Fig. 3.5 and Fig. 3.6, where the minimum pattern error is at about 7.7%. Then in the later iterations, this region remains about the same or keeps shrinking. So for the first 2 iterations, we set the initial search region of golden section search to be the region in which the first 10% of overall pixels can be flipped along the negative gradient direction. Our program keeps recording the minimum location which is found in each iteration to guide the search region for next iteration. For example, if in the current iteration, the minimum error is found at 5% of the overall pixels flipped, to be on the safe side, the search region of the next iteration will be automatically set as 1.5 times of 5%, which is 7.5%, of the pixels flipped. To prevent this search region to shink too small, we set a minimum as 2%. For the stop criteria of the golden section search, we set it as 0.25%, which means when the search region shrinks to or below 0.25%, our program will stop searching. As mentioned above for the jump technique, if our program cannot find a better solution along the search direction until reach the stop searching criteria(i.e., be trapped at some local minimum), our program will take the best solution except the starting point of that line search as a new solution, although it is actually a worse solution. This means one jump. Our program temporally accepts this worse solution as a new solution and continue the gradient based search starting from this new solution. At the end, our grogram will return the best local minimum that it has found.

The comparison of pattern error of optimized binary masks and runtime between our program and that of [5] is shown in Table 4.1. We use 9 binary image patterns as predefined targets. For the illumination source, both programs use annular illumination source as shown in Fig. 2.3. The outer and inner partial coherence factors for 184×184 target pattern are set to 0.4 and 0.3 respectively, for 400×400 target pattern, are set to 0.975 and 0.8 respectively, for all 2000×2000 target patterns, are set to 0.3 and 0.299

www.manaraa.com

respectively, and for all 4000×4000 target patterns, are set to 0.2 and 0.1995 respectively.

The pattern errors reported are calculated according to the best binary mask generated for both programs. The runtimes listed in the last two columns are based on the stopping criteria mentioned above. As the table shows, our program always generates better optimized binary mask which has significantly less pattern error. The pattern errors of [5] are higher than ours by 8.55% to 358.80%, with an average of 97.61%. Moreover, [5]'s program uses 4.49% more runtime than our program on average.

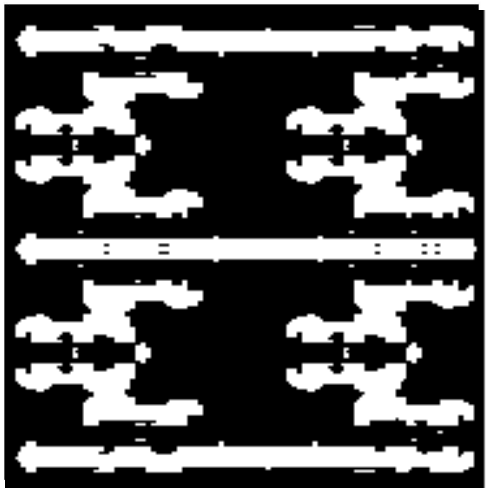
We report the pattern error of the *final* binary mask generated for [5]'s program with our program's runtime in column 6 of the table. For target patterns No.1, No.3 and No.7, [5]'s program stops earlier than ours. To see if [5]'s program will converge to better solutions if more runtime is allowed, we change the stopping criteria to let it run until the same runtime as our program. The result shows that the error gets worse in all 3 cases. If the pattern error of the *best* binary mask generated is reported instead, the result will be exactly the same as column 5. It indicates that the program fails to get out of the local minima even with more time.

Target pattern No.1 is obtained from [5] as shown in Fig. 3.2. We illustrate the optimized binary masks and the corresponding image patterns for both programs in Fig. 4.1. The pattern error convergence curves are shown in Fig. 4.2.

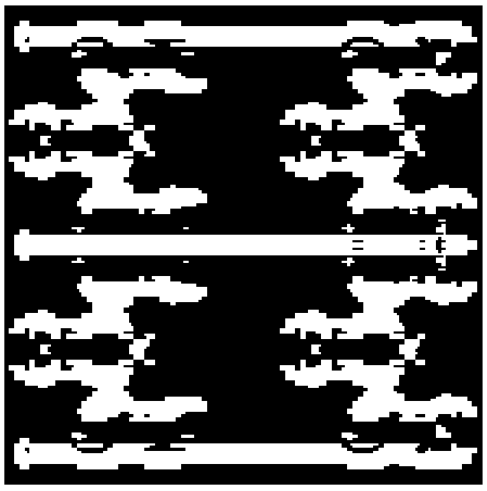
Because the program we obtained from [5] is fine tuned for the target pattern No.1 which is also obtained from [5], the experiment result of our algorithm is not so much better than [5]'s. However, based on the experiment results of other target patterns which cover multiple mask sizes, pixel sizes, and feature sizes, our algorithm has overwhelming advantage due to the application of line search engine which is enabled by our novel transformation for mask pixel.

More comparison of experiment results are shown in Fig. 4.3, Fig. 4.4, Fig. 4.5, Fig. 4.6, Fig. 4.7, Fig. 4.8, Fig. 4.9, and Fig. 4.10. And the pattern error convergence curves are shown in Fig. 4.11, Fig. 4.12, Fig. 4.13, Fig. 4.14, Fig. 4.15, Fig. 4.16, Fig. 4.17,

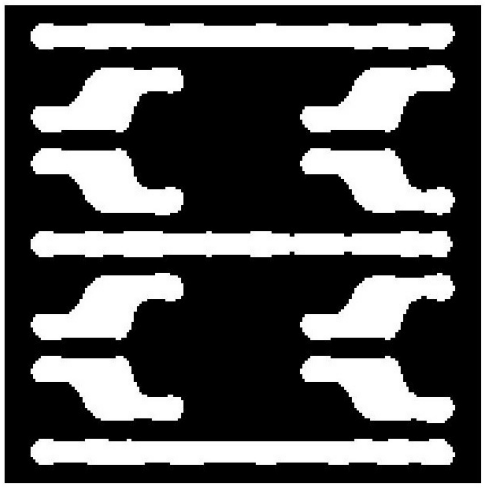
www.manaraa.com



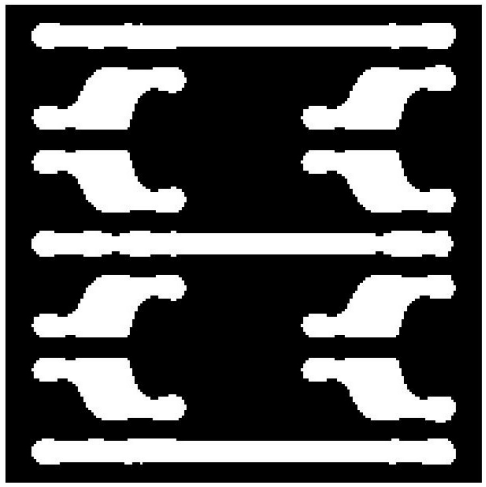
(a) [5]'s optimized binary mask



(b) Our optimized binary mask



(c) Image pattern of [5]'s optimized binary mask, error=1512



(d) Image pattern of our optimized binary mask, error=1376

Figure 4.1 Comparison for target pattern No.1.

www.manaraa.com

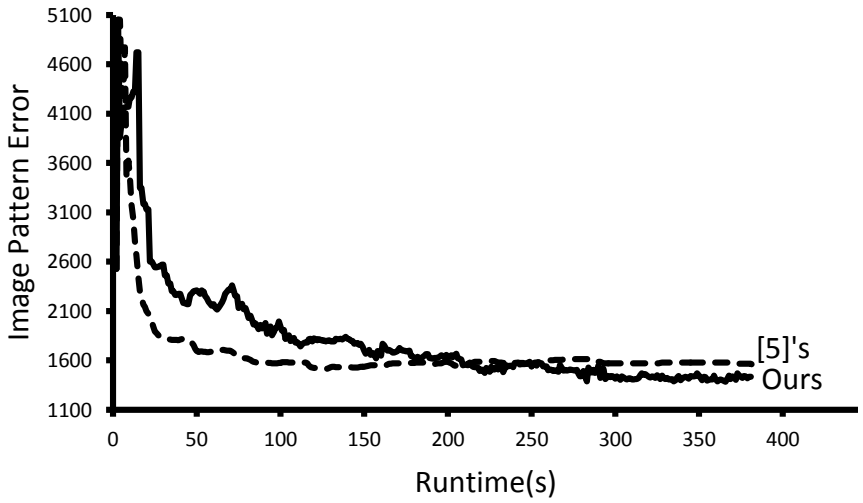


Figure 4.2 The Convergence Curves for target pattern No.1

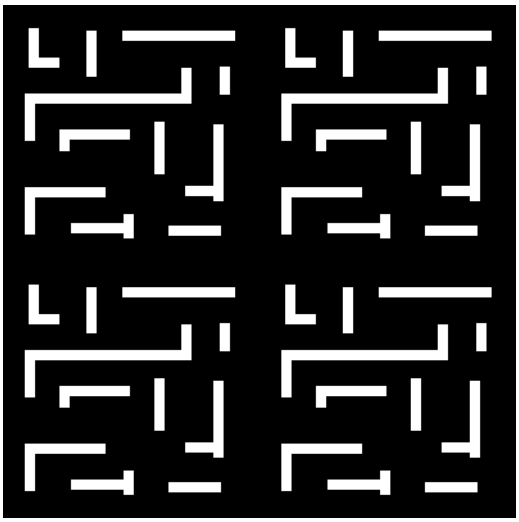
Fig. 4.18, Fig. 4.19, and Fig. 4.20. Except for target pattern No.1, and No.2, the size of the other patterns are too large to be shown. So we just illustrate a small part of the patterns and they are enlarged to show the details.

Target pattern No. 6 is a very challenging one which contains double snakes and many small contacts. According to the comparison in the Fig. 4.7, the image result of the optimized binary mask of [5]'s program even lost some of small contacts because it is very easy to be trapped in a local minimum and can not find a way out duo to the line search is not applied and a fixed step size is used. On the other hand, benefited from line search and jump technique, our program has better performance. As shown in Fig. 4.16, when our program is trapped in a local minimum which has almost the same error level as the local minimum [5]'s program found, the jump technique enables the algorithm to jump out and continue the search. To further compare the performance, we extended the size of target pattern No. 6 and name it to be target pattern No. 8. The comparison of enlarged a part of patterns are shown in Fig. 4.9. Because the pattern size is larger and there are more pixels which means more variables, the program has more degree of freedom to explore the solution space. We found that there are no missing contacts in the image result of the optimized binary mask of [5]'s program. But from the enlarged

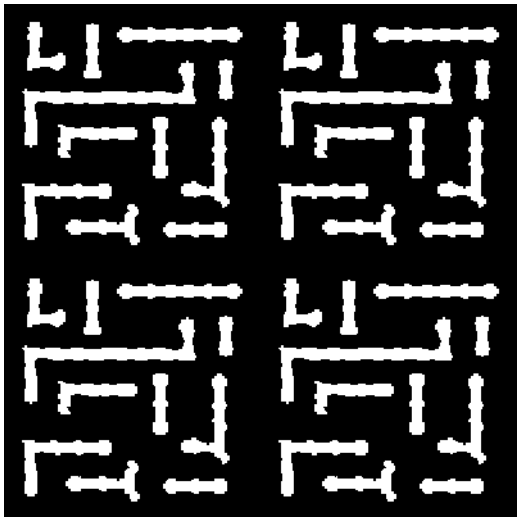
www.manaraa.com

figure, we can see that the small contacts formed by the optimized binary mask of [5]'s program are round while ours are more like square, which is more close to the target pattern.

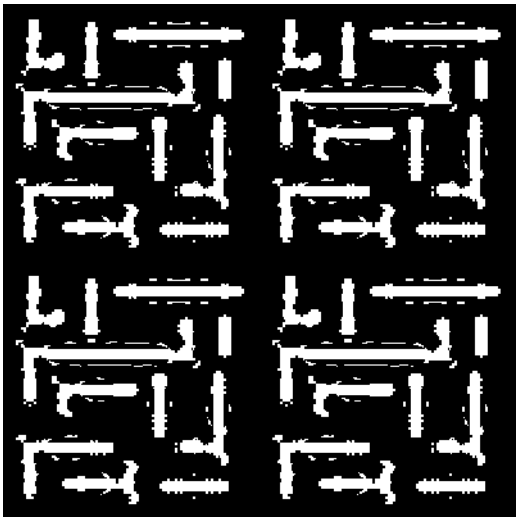
www.manaraa.com



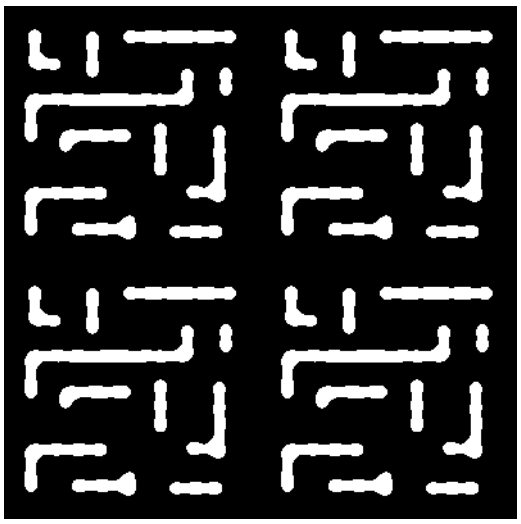
(a) Target pattern



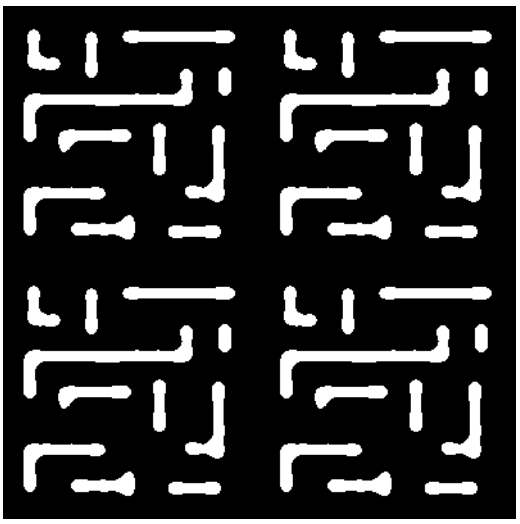
(b) [5]'s optimized binary mask



(c) Our optimized binary mask



(d) Image pattern of [5]'s optimized binary mask, error=3308

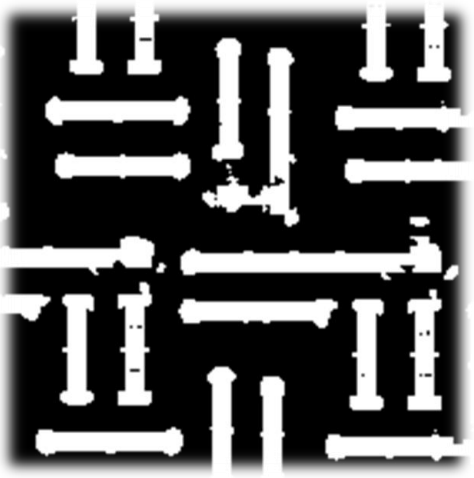


(e) Image pattern of our optimized binary mask, error=2664

Figure 4.3 Comparison for target pattern No.2.

www.manaraa.com

(a) Target pattern (a part)



(b) [5]'s optimized binary mask(a part)

(c) Our optimized binary mask (a part)

(d) Image pattern (a part) of [5]'s optimized binary mask, error=108144

(e) Image pattern (a part) of our optimized binary mask, error=99624

Figure 4.4 Comparison for target pattern No.3.

www.manaraa.com

(a) Target pattern (a part)

(b) [5]'s optimized binary mask(a part)

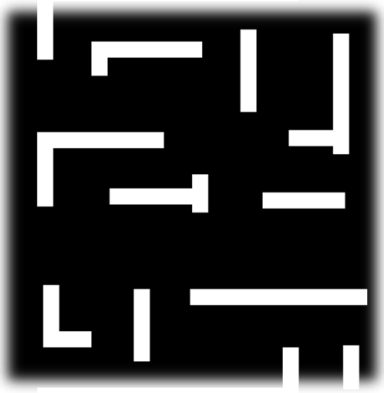
(c) Our optimized binary mask(a part)

(d) Image pattern (a part) of [5]'s optimized binary mask, error=61350

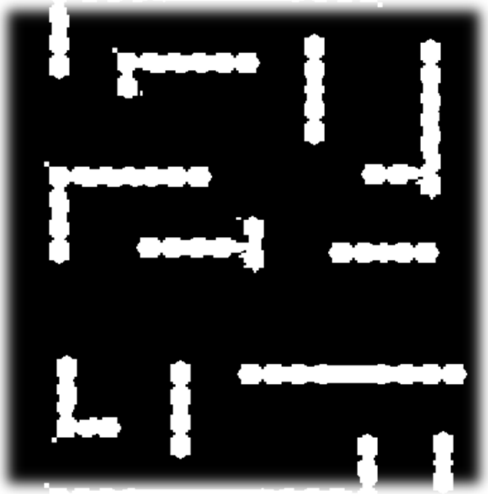
(e) Image pattern (a part) of our optimized binary mask, error=55409

Figure 4.5 Comparison for target pattern No.4.

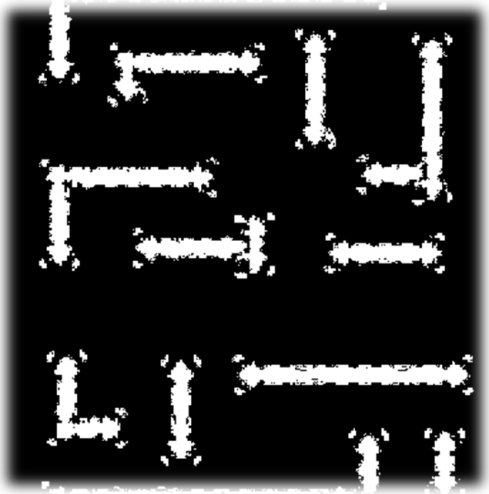
www.manaraa.com



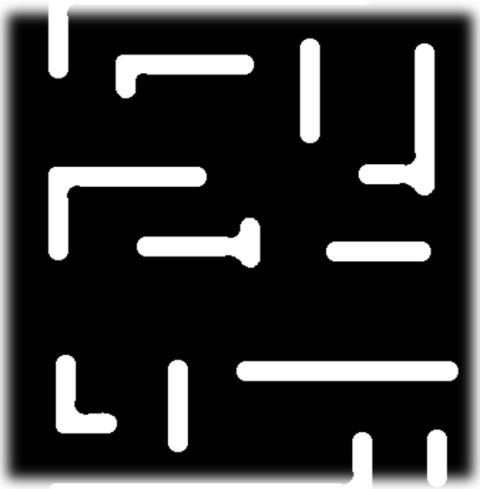
(a) Target pattern (a part)



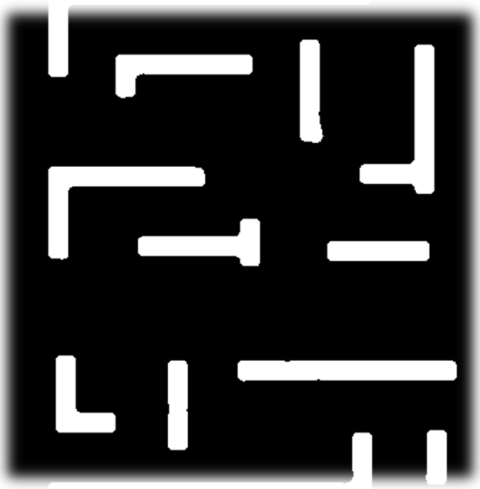
(b) [5]'s optimized binary mask(a part)



(c) Our optimized binary mask (a part)



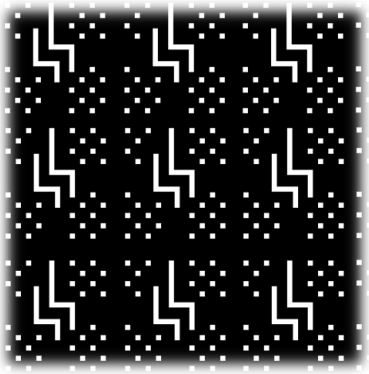
(d) Image pattern (a part) of [5]'s optimized binary mask, error=58410



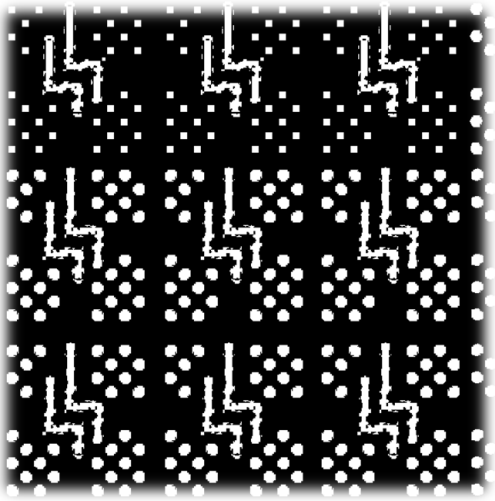
(e) Image pattern (a part) of our optimized binary mask, error=19575

Figure 4.6 Comparison for target pattern No.5.

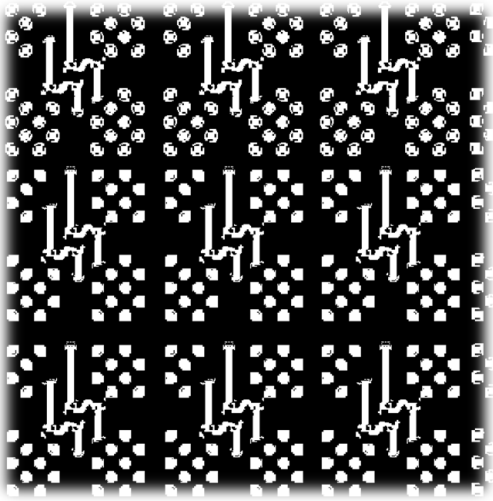
www.manaraa.com



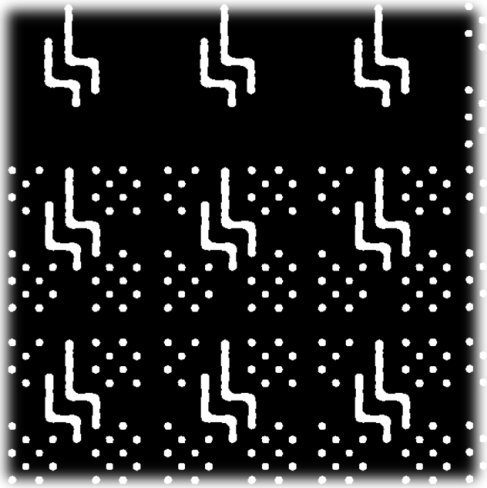
(a) Target pattern (a part)



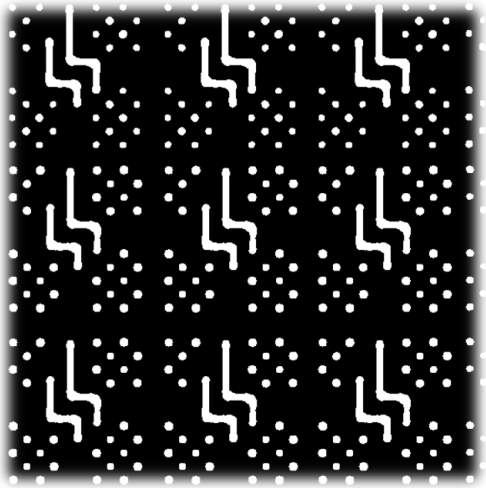
(b) [5]'s optimized binary mask(a part)



(c) Our optimized binary mask (a part)



(d) Image pattern (a part) of [5]'s optimized binary mask, error=101785



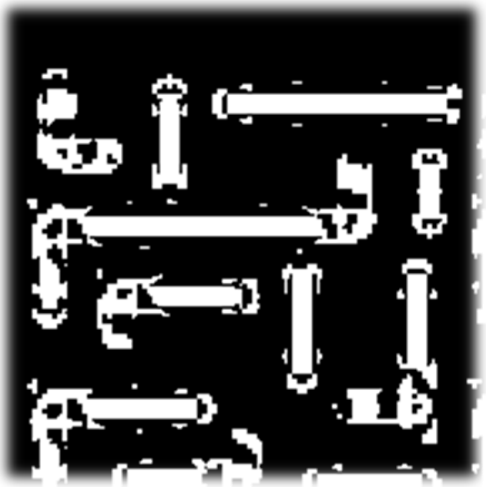
(e) Image pattern (a part) of our optimized binary mask, error=58104

Figure 4.7 Comparison for target pattern No.6.

www.manaraa.com

(a) Target pattern (a part)

(b) [5]'s optimized binary mask(a part)



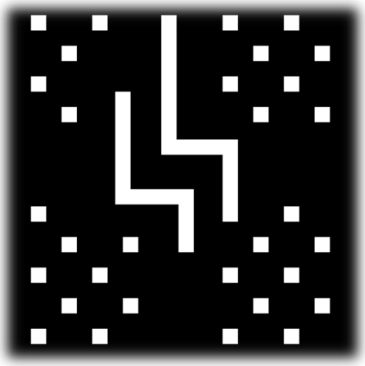
(c) Our optimized binary mask (a part)

(d) Image pattern (a part) of [5]'s optimized binary mask, error=64252

(e) Image pattern (a part) of our optimized binary mask, error=46063

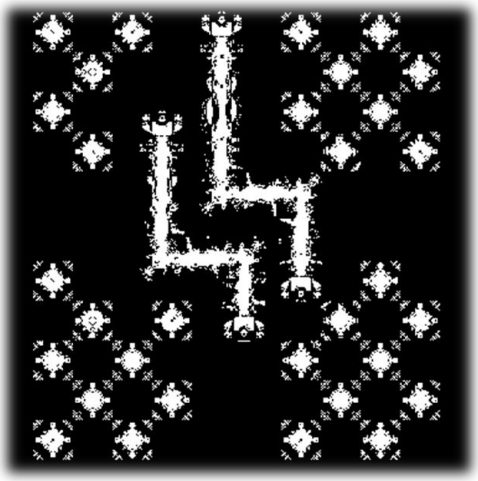
Figure 4.8 Comparison for target pattern No.7.

www.manaraa.com

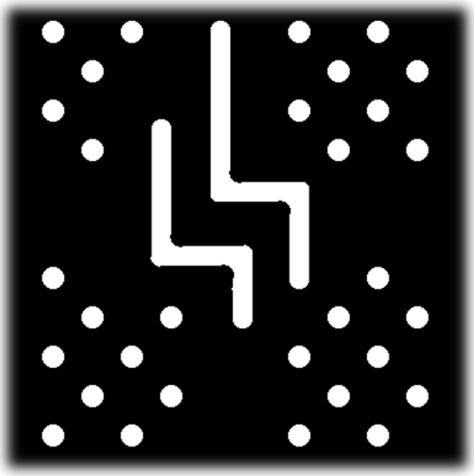


(a) Target pattern (a part)

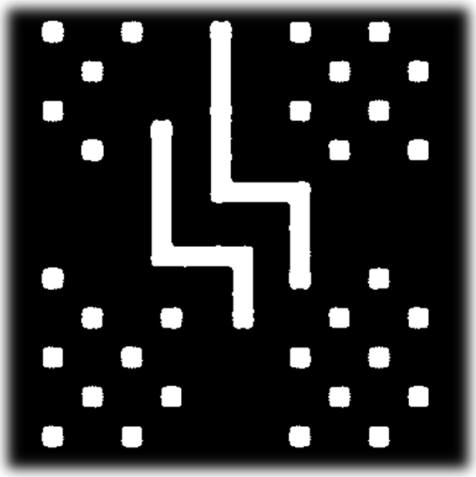
(b) [5]'s optimized binary mask(a part)



(c) Our optimized binary mask (a part)



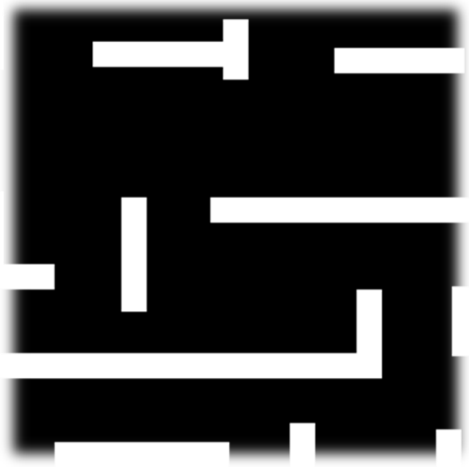
(d) Image pattern (a part) of [5]'s optimized binary mask, error=148356



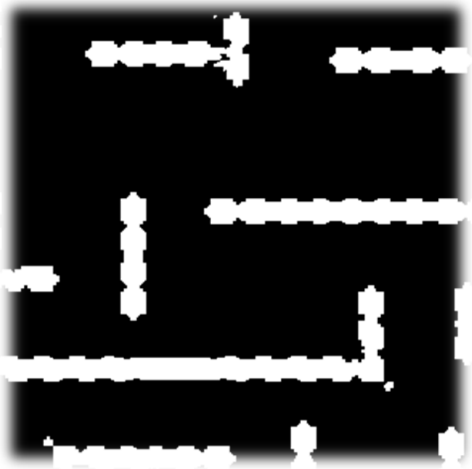
(e) Image pattern (a part) of our optimized binary mask, error=32336

Figure 4.9 Comparison for target pattern No.8.

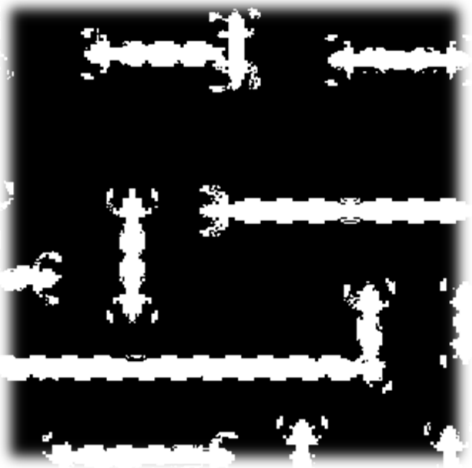
www.manaraa.com



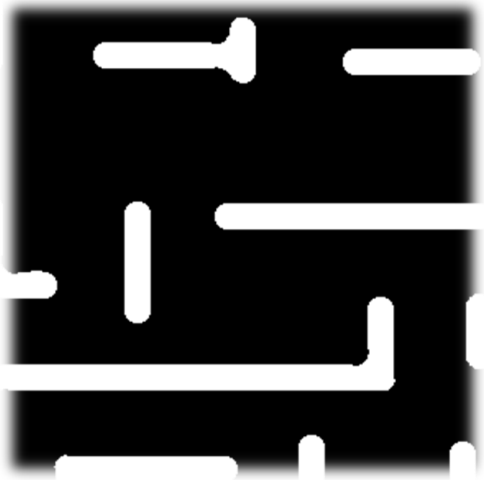
(a) Target pattern (a part)



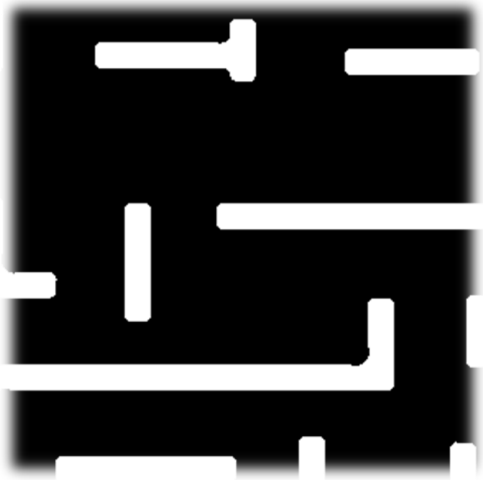
(b) [5]'s optimized binary mask(a part)



(c) Our optimized binary mask (a part)



(d) Image pattern (a part) of [5]'s optimized binary mask, error=52160



(e) Image pattern (a part) of our optimized binary mask, error=20592

Figure 4.10 Comparison for target pattern No.9.

www.manaraa.com

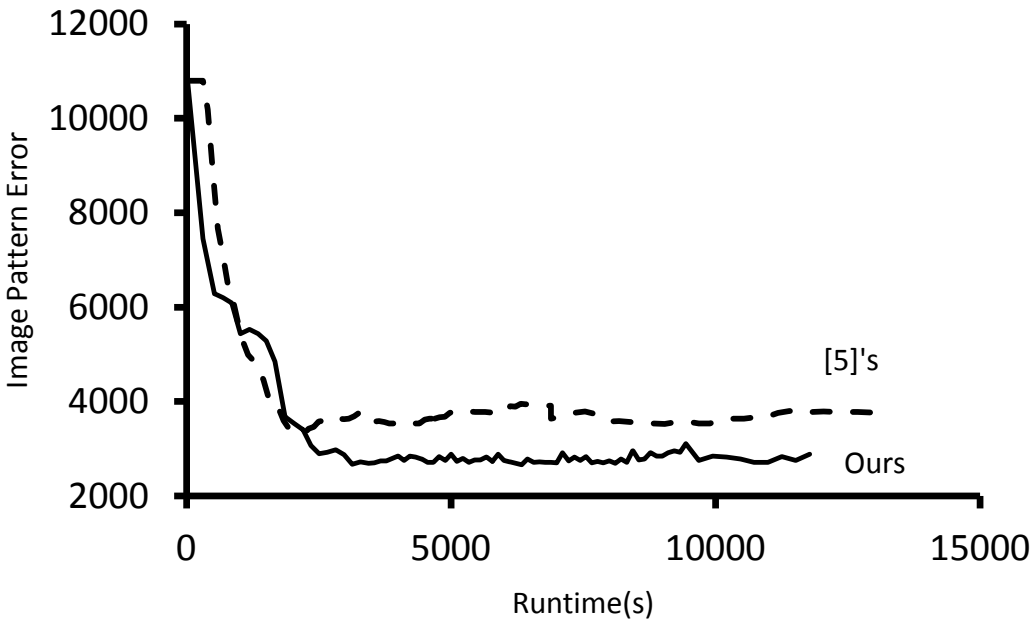


Figure 4.11 The Convergence Curve for target pattern No.2

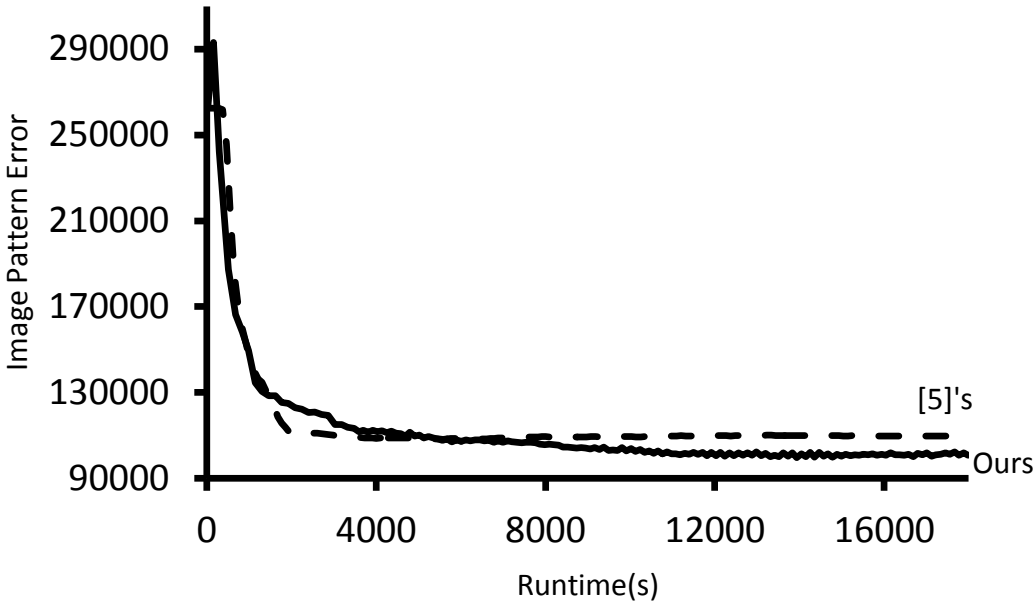


Figure 4.12 The Convergence Curve for target pattern No.3

www.manaraa.com

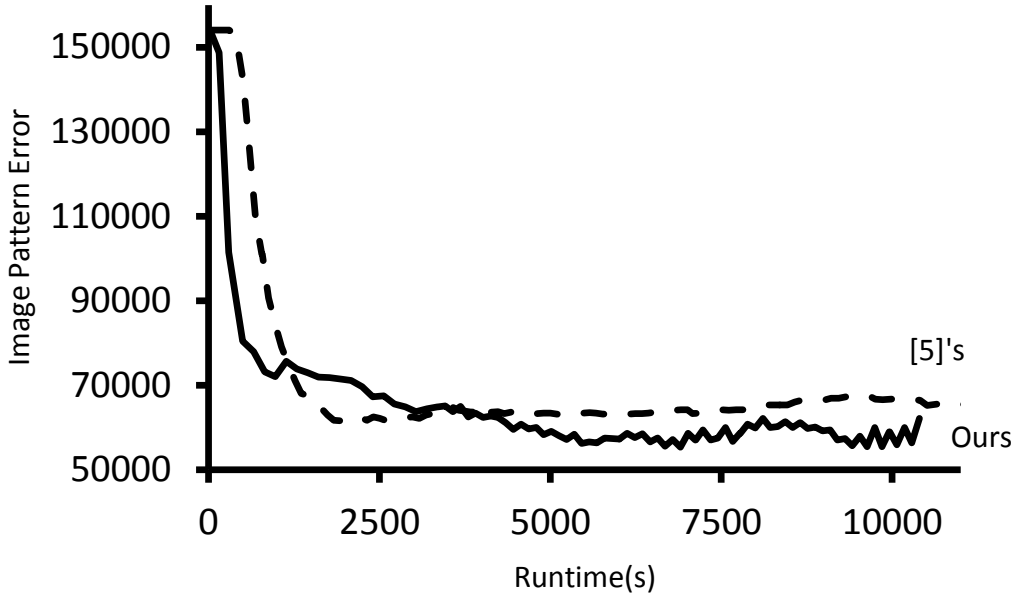


Figure 4.13 The Convergence Curve for target pattern No.4

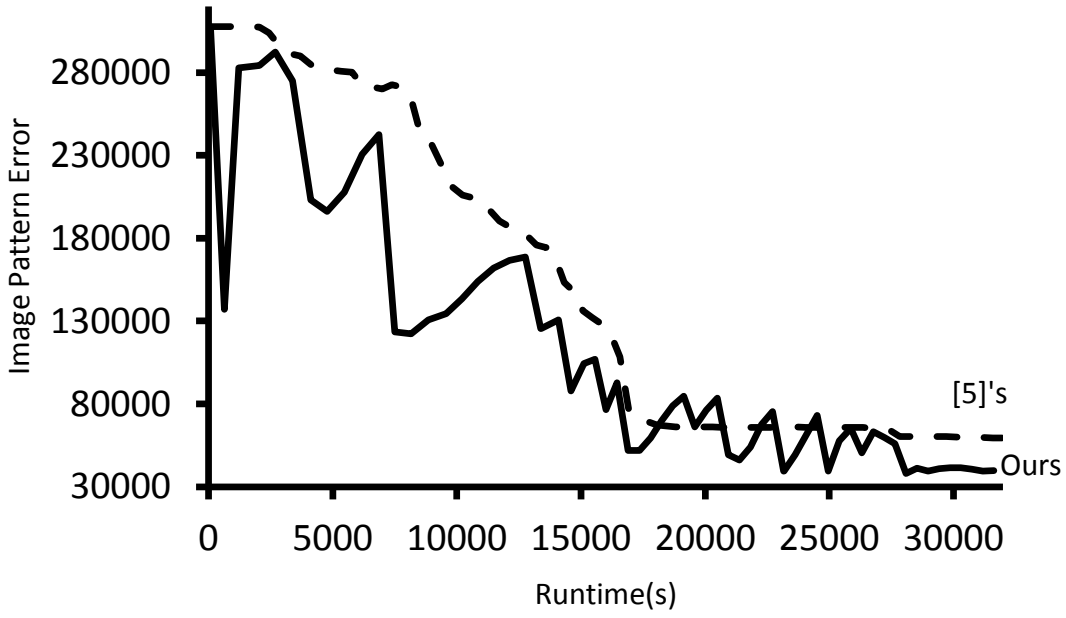


Figure 4.14 The Convergence Curve for target pattern No.5 part 1

www.manaraa.com

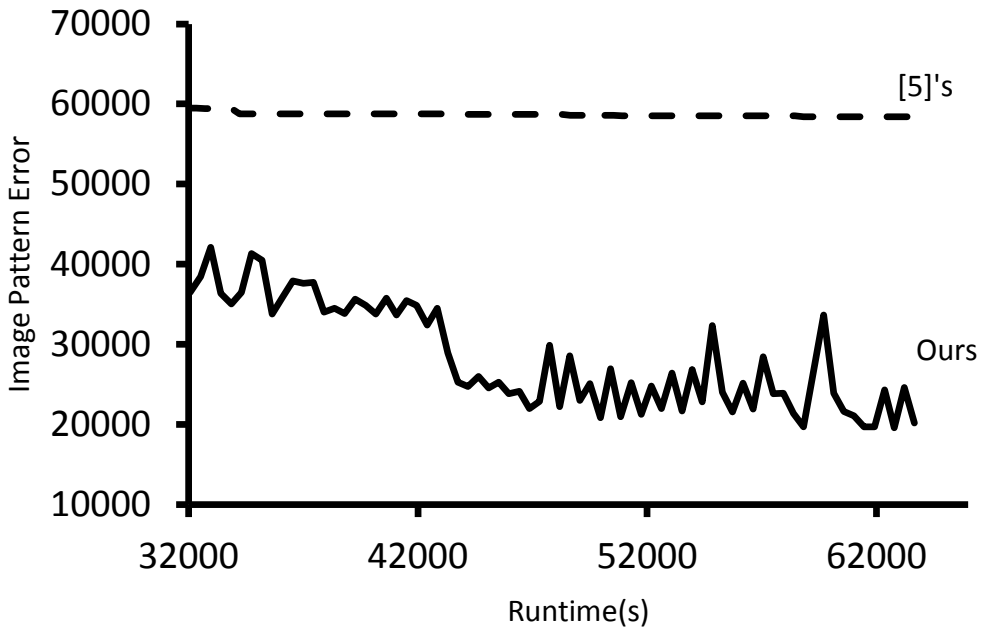


Figure 4.15 The Convergence Curve for target pattern No.5 part 2

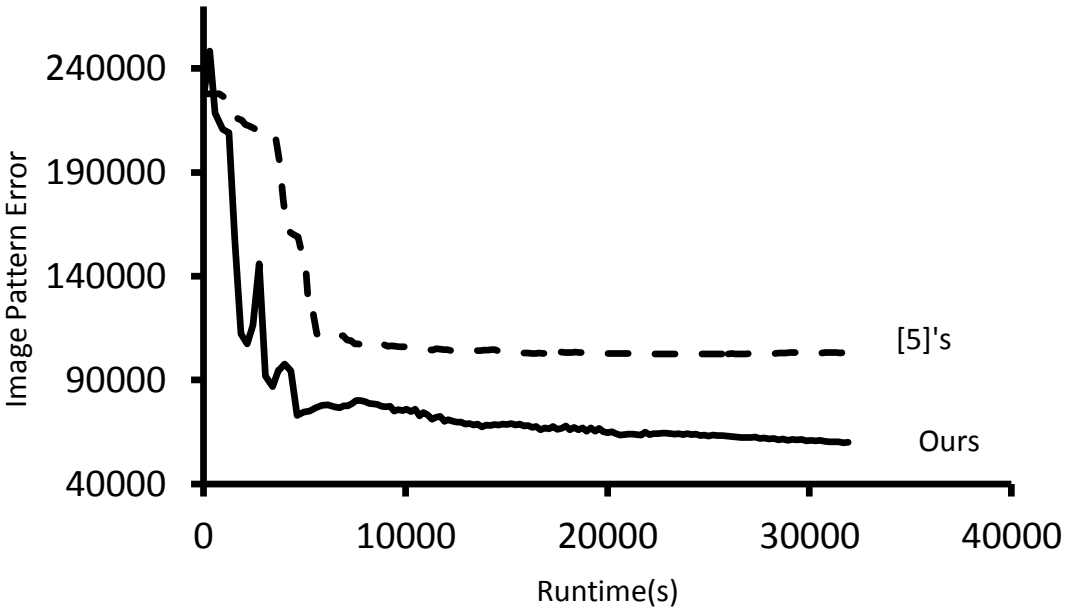


Figure 4.16 The Convergence Curve for target pattern No.6

www.manaraa.com

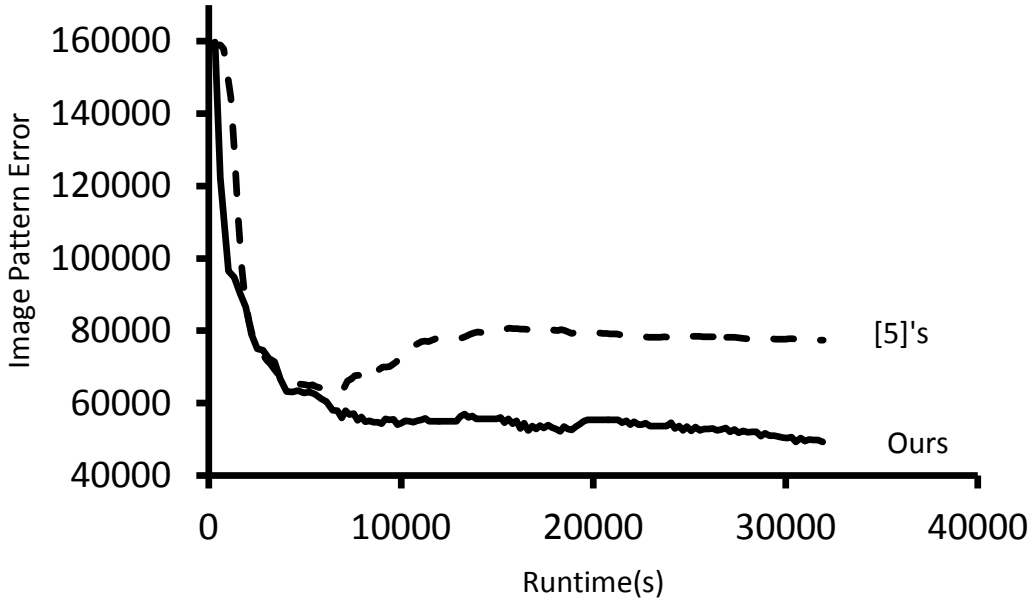


Figure 4.17 The Convergence Curve for target pattern No.7

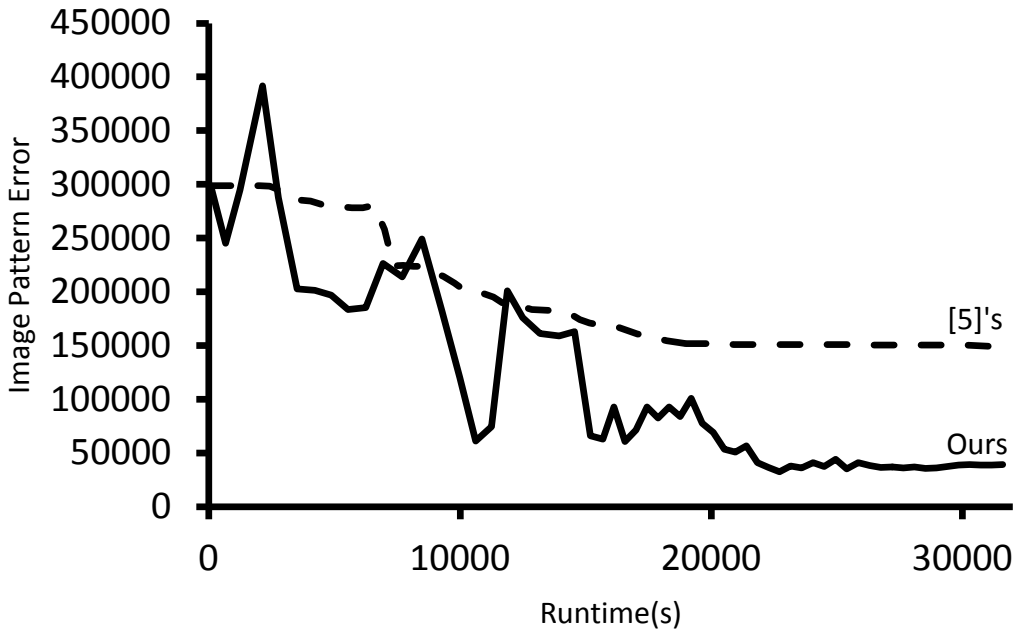


Figure 4.18 The Convergence Curve for target pattern No.8

www.manaraa.com

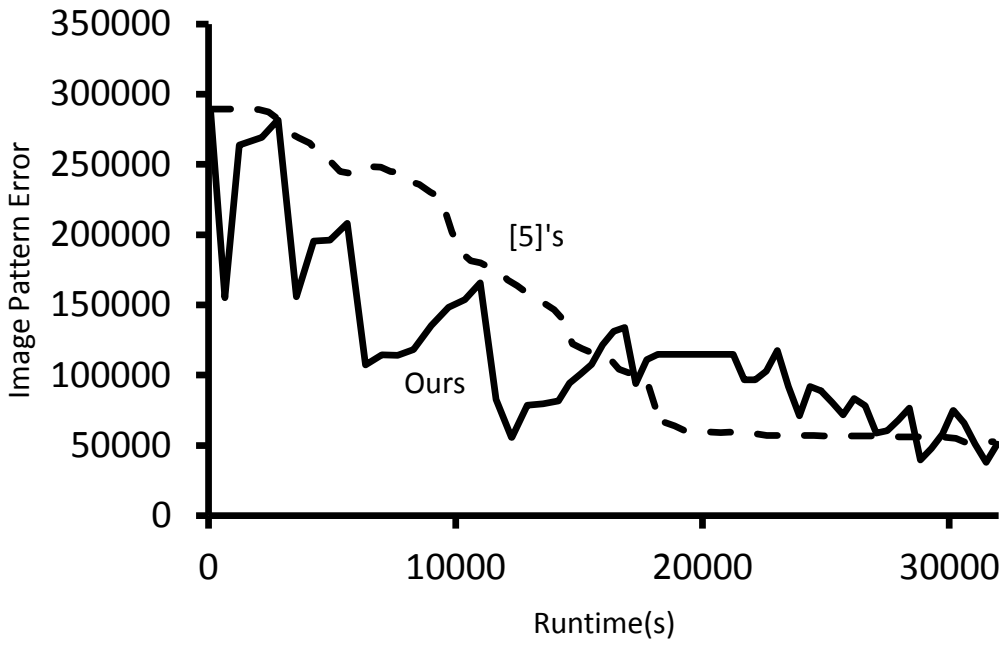


Figure 4.19 The Convergence Curve for target pattern No.9 part 1

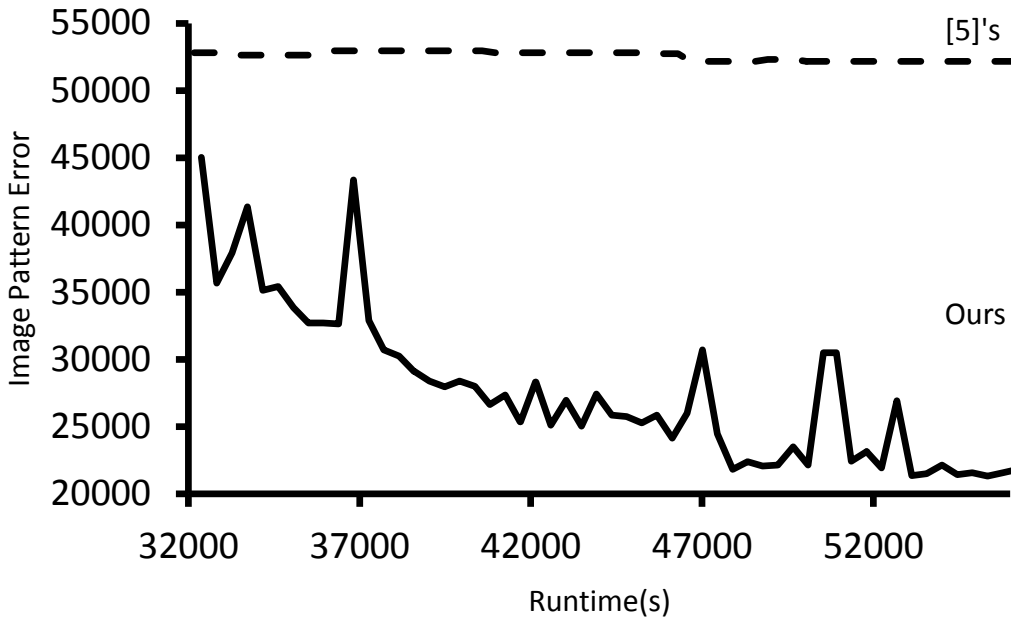


Figure 4.20 The Convergence Curve for target pattern No.9 part 2

www.manaraa.com

Table 4.1 Pattern error and Runtime Comparison between [5] and Ours.

No.	Mask Size (pixel)	Pixel Size (nm)	Feature Size (nm)	Pattern Error			Runtime (s)	
				[5]	[5] with our runtime	Ours	[5]	Ours
1	184×184	5.625	45	1512 (9.88%)	1566(13.81%)	1376	192 (-49.61%)	381
2	400×400	5.625	45	3308 (24.17%)	3780 (41.89%)	2664	13337(13.24%)	11778
3	2000×2000	5.625	45	108144 (8.55%)	109267 (9.68%)	99624	17918 (-0.38%)	17986
4	2000×2000	5.625	45	61350 (10.72%)	66516 (20.05%)	55409	12457 (19.81%)	10397
5	4000×4000	5.625	45	58410 (198.39%)	58410 (198.39%)	19575	78652 (16.34%)	67603
6	2000×2000	4	32	101785 (75.18%)	102100 (75.72%)	58104	49333 (13.45%)	43485
7	2000×2000	4	32	64252 (39.49%)	75300 (63.47%)	46063	51856 (-21.57%)	66117
8	4000×4000	4	32	148356 (358.80%)	148420 (358.99%)	32336	62008 (43.35%)	43255
9	4000×4000	4	32	52160 (153.30%)	52160 (153.30%)	20592	71489 (5.77%)	67588
Average				(97.61%)	(103.92%)		(4.49%)	

www.manaraa.com

CHAPTER 5. Conclusion

In this thesis, we introduced a highly efficient gradient-based search technique to solve the inverse lithography problem. We propose a new non-cyclic transformation of mask variables to replace the well-known cyclic one. Our transformation is monotonic, and it enables a much better control in flipping pixels and the use of line search to minimize the pattern error. we introduced a new technique named jump in order to jump out of the local minimum and continue the search. We used direct rounding technique to simplify the optimization. The experimental results showed that our technique is significantly more effective than the state-of-the-art techniques. It produces better binary masks in a similar runtime. The four techniques we proposed should be applicable to other iterative gradient-based search approaches like conjugate gradient method. We plan to incorporate our techniques into other search methods in the future.

www.manaraa.com

Bibliography

[1] Born, M. and Wolf, E. (1999). Principles of optics. In *Cambridge U.Press*.

[2] Granik, Y. (2006). Fast pixel-based mask optimization for inverse lithography. *Journal of Microlithography, Microfabrication and Microsystems*, 5:043002.

[3] Ma, X. and Arce, G. R. (2007). Generalized inverse lithography methods for phase-shifting mask design. *Opt. Express*, 15(23):15066–15079.

[4] Ma, X. and Arce, G. R. (2008). PSM design for inverse lithography with partially coherent illumination. *Opt. Express*, 16(24):20126–20141.

[5] Ma, X. and Arce, G. R. (2010). Computational lithography. *wiley*.

[6] Ma, X. and Arce, G. R. (2011). Pixel-based OPC optimization based on conjugate gradients. *Opt. Express*, 19:2165–2180.

[7] Pang, L., Liu, Y., and Abrams, D. (Jun. 2006). Inverse lithography technology (ILT): What is the impact to the photomask industry? *In Proc. SPIE 6283*.

[8] Poonawala, A. and Milanfar, P. (2006). OPC and PSM design using inverse lithography: a nonlinear optimization approach. *In Proc. SPIE Optical Microlithography*, (6514):1159–1172.

[9] Poonawala, A. and Milanfar, P. (2007a). Mask design for optical microlithography – an inverse imaging problem. *IEEE Transactions on Image Processing*, 16:774–788.

www.manaraa.com

[10] Poonawala, A. and Milanfar, P. (2007b). A pixel-based regularization approach to inverse lithography. *Microelectron. Eng*, 12:84.

[11] Saleh, B. E. A. and Rabbani, M. (1982). Simulation of partially coherent imagery in the space and frequency domains and by modal expansion. *Appl. Opt*, 21:2770–2777.

[12] Wong, A. K. (2001). Resolution enhancement techniques. In *SPIE Press*.

[13] Yu, P. and Pan, D. Z. (2007). Tip-opc: A new topological invariant paradigm for pixel based optical proximity correction. *Computer-Aided Design, ICCAD. IEEE/ACM International Conference on*, pages 847–853.

www.manaraa.com