

# Initialization for robust inverse synthesis of phase-shifting masks in optical projection lithography

Stanley H. Chan,<sup>1</sup> Alfred K. Wong,<sup>2</sup> and Edmund Y. Lam<sup>1,\*</sup>

<sup>1</sup>*Imaging Systems Laboratory,  
Department of Electrical and Electronic Engineering,  
The University of Hong Kong, Pokfulam Road, Hong Kong.*

<sup>2</sup>*Magma Design Automation Inc.,  
Santa Clara, CA 95054, USA.  
[elam@eee.hku.hk](mailto:elam@eee.hku.hk)*

**Abstract:** The continuous shrinkage of minimum feature size in integrated circuit (IC) fabrication incurs more and more serious distortion in the optical projection lithography process, generating circuit patterns that deviate significantly from the desired ones. Conventional resolution enhancement techniques (RETs) are facing critical challenges in compensating such increasingly severe distortion. In this paper, we adopt the approach of inverse lithography in the mask design, which is a branch of design methodology to treat it as an inverse mathematical problem. We focus on using pixel-based algorithms to design alternating phase-shifting masks with minimally distorted output, with the goal that the patterns generated should have high contrast and low dose sensitivity. This is achieved with a dynamic-programming-based initialization scheme to pre-assign phases to the layout when alternating phase-shifting masks are used. Pattern fidelity and worst case slopes are shown to improve with this initialization scheme, which are important for robustness considerations.

© 2008 Optical Society of America

**OCIS codes:** (100.3190) Inverse problems; (100.2980) Image enhancement; (110.5220) Photolithography.

---

## References and links

1. J. Plummer, M. Deal, and P. Griffin, *Silicon VLSI Technology — Fundamentals, Practice and Modeling* (Prentice Hall, 2000).
2. C. A. Mack, "30 years of lithography simulation," *Proc. SPIE* **5754**, 1-12 (2004).
3. F. Schellenberg, "Resolution enhancement technology: The past, the present, and extensions for the future," *Proc. SPIE* **5377**, 1-20 (2004).
4. A. K.-K. Wong, *Resolution enhancement techniques in optical lithography* (SPIE Press, Bellingham, Washington, 2001).
5. K. Nashold and B. Saleh, "Image construction through diffraction-limited high-contrast imaging systems: An iterative approach," *J. Opt. Soc. Am. A* **2**, 635–643 (1985).
6. B. Saleh and S. Sayegh, "Reductions of errors of microphotographic reproductions by optical corrections of original masks," *Opt. Eng.* **20**, 781–784 (1981).
7. S. Sherif, B. Saleh, and R. Leone, "Binary image synthesis using mixed linear integer programming," *IEEE Trans. Image Process.* **4**, 1252–1257 (1995).

8. Y. Liu and A. Zakhor, "Binary and phase-shifting image design for optical lithography," *IEEE Trans. Semicond. Manuf.* **5**, 138–151 (1992).
9. Y. Liu and A. Zakhor, "Optimal binary image design for optical lithography," *Proc. SPIE* **1264**, 410-412 (1990).
10. N. Cobb and A. Zakhor, "Fast sparse aerial image calculation for OPC," *Proc. SPIE* **2621**, 534-545 (1995).
11. Y. C. Pati and T. Kailath, "Phase-shifting masks for microlithography automated design and mask requirements," *J. Opt. Soc. Am. A* **11**, 2438–2452 (1994).
12. D. S. Abrams and L. Pang, "Fast inverse lithography technology," *Proc. SPIE* **6154**, 534-542 (2006).
13. C. Hung, B. Zhang, E. Guo, L. Pang, Y. Liu, K. Wang, and G. Dai, "Pushing the lithography limit: Applying inverse lithography technology (ILT) at the 65nm generation," *Proc. SPIE* **6154**, 61541M (2006).
14. L. Pang, Y. Liu, and D. Abrams, "Inverse lithography technology (ILT): What is the impact to the photomask industry?" *Proc. SPIE* **6283**, 62830X (2006).
15. A. Poonawala and P. Milanfar, "Prewarping techniques in imaging: Applications in nanotechnology and biotechnology," *Proc. SPIE* **5674**, 114-127 (2005).
16. A. Poonawala and P. Milanfar, "OPC and PSM design using inverse lithography: A nonlinear optimization approach," *Proc. SPIE* textbf6154, 61543H (2006).
17. S. H. Chan, A. K. Wong, and E. Y. Lam, "Inverse synthesis of phase-shifting mask for optical lithography," in "OSA Topical Meeting in Signal Recovery and Synthesis," (2007), p. SMD3.
18. X. Ma and G. R. Arce, "Generalized inverse lithography methods for phase-shifting mask design," *Opt. Express* **15**, 15066–15079 (2007).
19. J. W. Goodman, *Statistical Optics* (Wiley-Interscience, 1985).
20. J. W. Goodman, *Introduction to Fourier Optics* (Roberts and Company Publisher, Englewood, Colo, 2005), 3rd ed.
21. Y. Granik, "Solving inverse problems of optimal microlithography," *Proc. SPIE* **5754**, 506-526 (2004).
22. A. Poonawala and P. Milanfar, "Mask design for optical microlithography — an inverse imaging problem," *IEEE Trans. Image Process.* **16**, 774–788 (2007).
23. S. H. Chan and E. Y. Lam, "Inverse image problem of designing phase shifting masks in optical lithography," in "IEEE International Conference on Image Processing," (2008).
24. P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization* (Academic Press, London, 1986).
25. W. Rudin, *Principles of Mathematical Analysis* (McGraw-Hill, 1976).
26. M. Minoux, *Mathematical programming theory and algorithms* (John Wiley and Sons, Chichester, 1986).
27. P. Berman, A. Kahng, D. Vidhani, H. Wang, and A. Zelikovsky, "Optimal phase conflict removal for layout of dark field alternating phase shifting masks," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems* **19**, 175–187 (2000).
28. A. Moniwa and T. Terasawa, "Heuristic method for phase-conflict minimization in automatic phase-shift mask design," *Jpn. J. Appl. Phys.* **34**, 6584–6589 (1995).
29. L. G. Shapiro and G. C. Stockman, *Computer Vision* (Prentice Hall, 2001).
30. D. Halliday, R. Resnick, and K. S. Krane, *Physics*, (John Wiley and Sons, New York, 2002), 2nd ed., Vol. 2.

## 1. Introduction

Optical lithography is one of the most challenging steps in semiconductor fabrication. Undoubtedly, decades of advances in lithographic technology have enabled the continuation of Moore's Law [1]. However, when one uses a 193nm source to obtain a feature size as small as 90nm, 65nm or even 45nm, pattern distortions associated with diffraction effects are no longer secondary, but have become the primary issues [2, 3]. Conventional optical proximity correction (OPC) techniques, including rule-based and model-based OPCs [4], face increasing challenges in providing desirable solutions in mask design.

For many years, people have explored ways to efficiently treat the mask design as an inverse problem through mathematical modeling of the pattern distortion. In the 1980s, lithographers started to consider these inverse lithography techniques (ILTs) [5, 6]. However, due to the slow run time and complicated mask design, this approach was not practical in manufacturing. In the following decade, there were several important milestones in ILT. Sherif and Leone applied iterative methods to generate binary masks [7]; Liu and Zakhor used simulated annealing to synthesize masks [8, 9]; Cobb and Zakhor parameterized the mask pattern using edges and corners, and simulated the output pattern with changes in these geometrical elements [10]; and Pati and Kailath proposed to use Gercberg-Saxton algorithms to generate phase-shifting masks (PSM) [11].

In recent years, increase in computational power (attributable to more densely packed circuit patterns made feasible in part by advances in lithography) causes ILT to be revisited as an important research area [12, 13, 14]. In particular, a fast optimization framework to synthesize binary and phase-shifting masks has been demonstrated [15, 16] and refined [17, 18]. It is well understood that the optimization is non-convex, and iterative methods can easily lead to a local rather than global solution. Thus, a good initialization scheme is essential. While many methods choose to use the ideal mask pattern as the initialization [16], in this paper we show that a simple method of phase assignment with dynamic programming can greatly improve the mask design for alternating phase-shifting masks. In particular, pattern fidelity and worst case slopes improve with this initialization scheme, which are important for robustness considerations.

The flow of the paper is as follows. We first give a brief description about optical lithography system and the mathematical models in Section 2, as well as an outline of existing and our optimization algorithms used to find the mask. This prepares us for Section 3, devoted to our initialization scheme, where we demonstrate a simple way to efficiently find an initial guess that can further improve the performance of the algorithm. Examples and analyses are given in Section 4, followed by some concluding remarks in Section 5.

## 2. Optical lithography

### 2.1. Lithography systems modeling

Optical lithography consists of four basic elements: a source, a mask, a lens and a wafer. As shown in Fig. 1, when light coming from the source reaches the mask, it is essentially transmitted only through the transparent regions. The layout pattern on the mask is replicated onto the photoresist-coated wafer by an exposure system. Exposed photoresist layer then undergoes a series of chemical reactions, resulting in regions of high and low exposure. Finally, these latent images are developed, creating openings in the resist for subsequent processing.

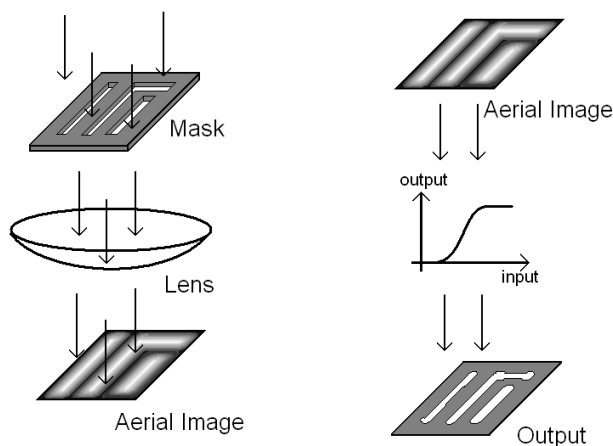


Fig. 1. A simplified diagram of an optical projection lithography system. There are two processes: projection optics, and photoresist development, and four basic elements: source, mask, lens, and wafer.

For simplicity we first assume that the source is a monochromatic coherent point source. With Köhler's illumination method [19] incident light illuminates the mask uniformly. Let  $O(x, y)$  be the mask pattern, and its spectrum  $\tilde{O}(f, g)$ . Similarly,  $H(x, y)$  is the transfer function describing

the projection optics, while its Fourier transform  $\tilde{H}(f, g)$  is the pupil function. For unity magnification systems,  $\tilde{H}(f, g) = 1$  for  $\sqrt{f^2 + g^2} \leq \text{NA}/\lambda$ , which is the transmitted region, and zero otherwise. Here, NA is the numerical aperture and  $\lambda$  is the wavelength of the source. Thus,

$$H(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{H}(f, g) \exp\{j2\pi(fx + gy)\} df dg = \frac{J_1(2\pi r \text{NA}/\lambda)}{2\pi r \text{NA}/\lambda}, \quad (1)$$

where  $r = \sqrt{x^2 + y^2}$ , and  $J_1(x)$  is the Bessel function of the first kind, order 1 [20]. To maintain a unity gain, often we need to normalize  $H(x, y)$ . The total E-field at the image plane is  $E(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{H}(f, g) \tilde{O}(f, g) \exp\{-j2\pi(fx + gy)\} df dg$ , and the intensity of the aerial image equals magnitude squared of the field, i.e.

$$I_{\text{aerial}}(x, y) = |E(x, y)|^2 = |H(x, y) * O(x, y)|^2. \quad (2)$$

Note that while in this paper we have used coherent illumination in our discussions for simplicity, the principle can be applied to partially coherent systems and incoherent systems, because the former can be decomposed into a sum of coherent sources [11] and the latter can be modeled as convolution between  $|H(x, y)|^2$  and  $|O(x, y)|^2$  [20].

The next step in lithography system is photoresist development, where the photoresist layer changes its chemical profile after exposed to light. As shown in Fig. 2, the resist development proceeds vertically downward from the resist surface. In reality, the process will proceed laterally as well, but to simplify the modeling, we will only consider the vertical development. This can be modeled as a thresholding. Given a threshold intensity  $t_r$ , any aerial image intensity higher than  $t_r$  will cause the development. It has been suggested that one can reasonably model the resist action by a sigmoid function  $\text{sig}\{\cdot\}$  as [15]

$$I(x, y) = \text{sig}\{I_{\text{aerial}}(x, y)\} = \frac{1}{1 + \exp\{-a(I_{\text{aerial}}(x, y) - t_r)\}}, \quad (3)$$

where  $a > 0$  is the parameter determining the contrast of the sigmoid function.

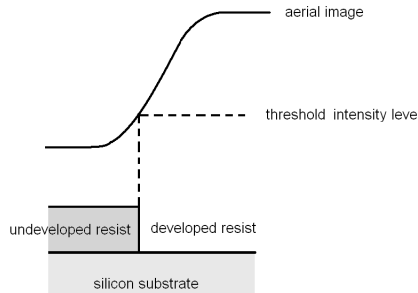


Fig. 2. Development of an infinite-contrast resist can be defined by the threshold.

## 2.2. Optimization process in inverse lithography

We described above the formation of the image  $I(x, y)$  with a certain mask pattern  $O(x, y)$ . For conventional chromium on glass (COG) masks, we can assume that  $O(x, y)$  takes on binary values  $\{0, 1\}$ . Apart from such masks, the most commonly used ones nowadays are phase-shifting masks (PSM) [4], which contain two or more phases. They are further divided into

attenuated PSM and alternating PSM. For the latter, which is the focus of this paper, we can assume that  $O(x,y)$  takes on ternary values  $\{-1, 0, 1\}$ .

Thus, the error  $F$  between a desired circuit pattern  $\hat{I}(x,y)$  and the output image due to an input mask  $O(x,y)$  is given by

$$F = \sum_{x,y} (I(x,y) - \hat{I}(x,y))^2 = \sum_{x,y} \left[ \frac{1}{1 + e^{-a(|H(x,y)*O(x,y)|^2 - t_r)}} - \hat{I}(x,y) \right]^2. \quad (4)$$

We set up a minimization problem to find an optimal mask pattern with the smaller error, i.e.

$$O_{\text{opt}}(x,y) = \arg \min_{O(x,y) \in \{-1,0,1\}} F. \quad (5)$$

Note that this is an inverse problem that is ill-posed, because small changes in  $\hat{I}(x,y)$  can lead to large changes in  $O(x,y)$  that can minimize  $F$ . There can also be multiple solutions, and hence the optimal mask pattern is not necessarily unique.

In general, analytical approach to inverse lithography problems is not possible [14], because although the model of projection is well established, the model of resist development is empirical. Here we approximate the resist development as hard threshold, but in reality this is much more complicated. Therefore, an inverse lithography problem does not imply that the solution must be in closed form. Often the solutions are found by iterative approaches [21].

Another remark is the existence of unique global optimum. Again because Eq. (5) is an ill-posed problem, there is no guarantee for a unique global minimum. In most situations, there are multiple global minima [14]. Hence, it is usually the decision of a designer to accept which solution. This is also a reason why a local search algorithm is applicable to solve the problem.

It has been suggested that to solve Eq. (5) iteratively, one can relax the constraint to  $-1 \leq O(x,y) \leq 1$ , add a penalty term  $R = \sum_{x,y} (-4.5O(x,y)^4 + O(x,y)^2 + 3.5)$ , and minimize  $F + \gamma R$  where  $\gamma$  is a weighting constant for the penalty  $R$ . The optimization is then solved iteratively by steepest descent [18, 22]. Our implementation consists of two differences: the use of active set method and conjugate gradient, to speed up the iterations [23]. Nevertheless, it should be mentioned that the initialization scheme proposed in this paper should work for different variations of the inverse imaging approach.

### 2.2.1. Active set method

The constraint  $-1 \leq O(x,y) \leq 1$  is a simple bound on the variables. It was suggested to use the trigonometric substitution  $O(x,y) = \cos \theta(x,y)$  to transform the optimization problem into an unconstrained one [15, 16]. However, this type of trigonometric substitution is generally not recommended [24]. First, trigonometric substitution introduces periodicity to the solution. The new variable  $\theta$  has a period of  $2\pi$  so for every  $2\pi$  we have a repeated solution. Another problem is the increment of nonlinearity. The objective function itself is nonlinear. Using a trigonometric substitution, we may bring additional local minima and stationary points. Non-linearity also makes the gradient algorithm difficult to search, because often it affects stability and convergence negatively.

Noting that the constraint is an upper-lower bound, in this work we use the active set method [24]. The key idea behind this technique is to project all out-of-range variables to its closest bound. Thus, we set  $O(x,y) = 1$  if  $O(x,y) > 1$ , and  $O(x,y) = -1$ , if  $O(x,y) < -1$ . Active set method guarantees convergence, because the constraints form a  $k$ -dimensional interval (or  $k$ -cell [25]), which is convex. Hence the restriction to this  $k$ -cell is a projection onto convex set (POCS). Another advantage of using active set method is that we can stop searching at  $(x_i, y_i)$  if  $O(x_i, y_i)$  is a boundary point. Thus the number of variables will decrease monotonically.

### 2.2.2. Conjugate gradient

We tackle each iteration using the Fletcher-Reeves conjugate gradient algorithm [26]. Conjugate gradient is a first-order derivative based algorithm. It searches better than steepest descent, while keeping the implementation manageable. In our algorithm, we normalize all gradient vectors, and add a line search algorithm, so that at every step we seek the best step size for better convergence.

Let  $O^k(x, y)$  be the variable at  $k$ -th iteration, and let  $G = F + \gamma R$ . The following pseudo-code for conjugate gradient is then applicable:

**Stage 0** Choose a starting point  $O^0(x, y)$ , and set  
 $d_0 = -\nabla G(O^0(x, y))$ .

**Stage  $k$**

- Line search:  
 Find  $\lambda_k = \operatorname{argmin} G(O^k(x, y) + \lambda d_k)$ .
- Set  $O^{k+1}(x, y) = O^k(x, y) + \lambda_k d_k$ .
- $d_{k+1} = -\nabla G(O^{k+1}(x, y)) + \beta_k d_k$ .
- $\beta_k = \frac{\|\nabla G(O^{k+1}(x, y))\|^2}{\|\nabla G(O^k(x, y))\|^2}$ .

**Stop** If stopping criteria is satisfied, then END. Otherwise,  $k \leftarrow k + 1$  and proceed Stage  $k + 1$ .

The computation of  $\nabla G$  requires  $\nabla F$  and  $\nabla R$ , which equal to:

$$\nabla F = 4a \left\{ H(x, y) * \left[ (\hat{I}(x, y) - I(x, y)) \cdot I(x, y) \cdot (1 - I(x, y)) \cdot (H(x, y) * O(x, y)) \right] \right\}, \quad (6)$$

and

$$\nabla R = -18O(x, y)^3 + 2O(x, y). \quad (7)$$

The “ $\cdot$ ” above indicates pointwise multiplication. The derivations are omitted for brevity, but are similar in spirit to those in [22] with the use of chain rules in calculating derivatives.

### 3. Initialization

Finding the initial estimate is a critical part of the optimization. It affects both the quality of the solution and the run time to achieve it. A poor initialization may lead to an undesirable local minimum in our optimization operation, and possibly a lot more computation to get out of the local minimum, if at all.

A trivial initial guess for mask design algorithm can be the desired pattern itself (i.e. a binary mask without the phase component). However, we observe that such initialization may not be the best choice, because the algorithm tends to search for a local optimum around the desired pattern. The optimal mask so generated often has good fidelity. However, it may suffer from poor image slope if two objects are closely located, because the gradient-based algorithm is a local searching algorithm — it cannot make an abrupt  $0-\pi$  phase alternation among the objects if the initial guess is all 0-phase.

Consider Fig. 3(a). Assuming that an alternating phase-shifting mask fabrication is used in the manufacturing process, we can have zero intensity at the middle of two strips if one strip uses 0 phase while the other one uses  $\pi$  phase. Therefore, given a complicated mask pattern,

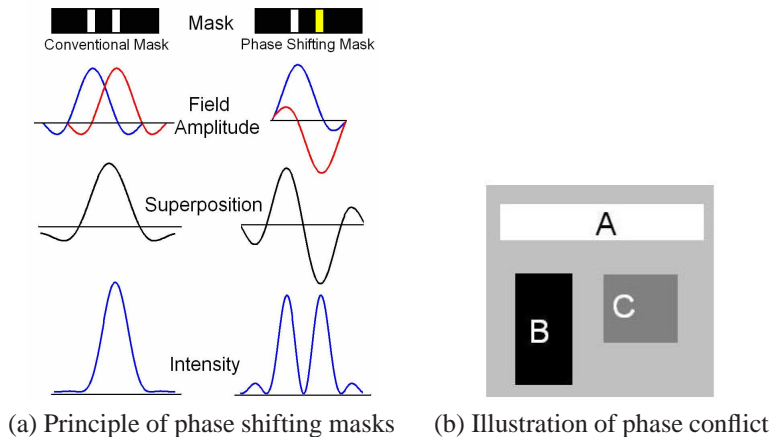


Fig. 3. Principle of phase shifting masks and the difficulty of phase assignment.

if we can assign phases to objects as alternating as possible, we can improve the slope significantly. This is the motivation of our methodology in this paper. However, it is very common that phase conflicts would exist if we enforce this rule rigidly. For example, in Fig. 3(b), if we assign Object A with phase 0, and Object B with phase  $\pi$ , then we will have trouble in assigning phase to Object C. This type of phase conflict problem is well known in the industry, and there are plenty of researches done [27, 28]. However, their methodologies are relatively sophisticated and time-consuming, as they are designed for large layouts. In contrast, we seek a simple way that requires a small amount of time to find the initial guess compared with the iterations. Our problem also involves significantly fewer objects, as pattern size in inverse lithography is generally relatively small.

Our proposed phase assignment scheme is fast and simple. It consists of three parts: (i) Extract pattern information; (ii) Build a distance matrix; (iii) Use dynamic programming to find the optimal path for assignment.

### 3.1. Extracting vertices

All mask patterns have rectangles as the elementary building blocks so they have right-angled corners. Hence, one efficient way to represent a pattern is to use vertices. Knowing the vertices we can know the length and location of every edge. We extract the vertices with a  $3 \times 3$  all 1's kernel to filter the image. Note that the output pixel value can be 4 or 8 if and only if it is a vertex. Then we label the objects on a pattern by using a connected component algorithm [29]. With the labeled pattern, we make a cross referencing between labels and vertices so that the vertices can be grouped.

### 3.2. Distance matrix

Knowing the vertices of objects we can find the degree of interaction between two objects. However, we still need an appropriate function  $d(i, j)$  to measure the degree of interaction between Object  $i$  and Object  $j$ . There are various distance functions that can be use for such degree measurement, and we discuss three reasonable designs here. Let  $W$  be the width of overlapping region of two edges, and  $G$  be the gap between two objects, as shown in Fig. 4(a).

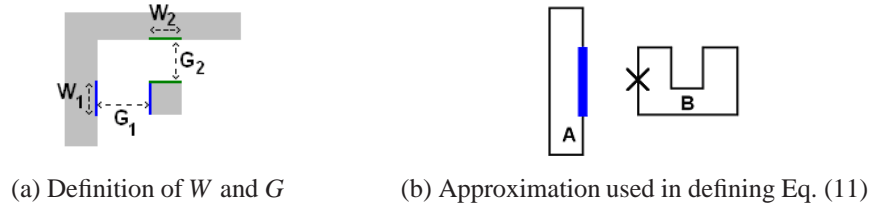


Fig. 4. Definitions and approximations to define various distance functions

The first possible distance metric is

$$d_1(i, j) = \sum_k \frac{G_k}{W_k}, \quad (8)$$

where the index  $k$  refers to the  $k$ -th edge pair under comparison. In the figure,  $k = 2$ , as there are two pairs of edges being interacting with each other. Equation (8) measures the sum of ratios between  $G$  and  $W$ . Intuitively, if the gap  $G$  is small, then image slope of  $I_{\text{aerial}}(x, y)$  will be low. If the width  $W$  is large, then there are more parts of the edge suffering from poor slope. Clearly, Eq. (8) satisfies these two criteria.

A second function is

$$d_2(i, j) = \left( \sum_k \frac{W_k}{G_k^2} \right)^{-1}. \quad (9)$$

We take the squared contribution of  $G$ , because optical illumination is electromagnetic wave propagation. The E-field at a point is inversely proportional to the square of distance from the source. A summation is taken to represent superposition, and a reciprocal is taken because we want to reverse the orderings of the magnitude.

The third function is modified from E-field at a point from a rod of charge. Suppose we have a rod, with length  $l$ , and charge distribution  $\lambda$ , then the E-field at a point  $y$  from the mid-point of the rod is [30]

$$E = \frac{1}{4\pi\epsilon_0} \frac{2\lambda}{y} \frac{l/2}{\sqrt{y^2 + (l/2)^2}}. \quad (10)$$

In our problem, there is no isolated pixel in the layout. If we want an accurate value of the interaction between two objects, then we need to sum over all pixels. However, this causes too much calculation and we seek an approximation instead. We consider only the overlapping regions of one object, and treat another object as a point as shown in Fig. 4(b). Dropping the constants, we have

$$d_3(i, j) = \left( \sum_k \frac{W_k/2}{G_k \sqrt{G_k^2 + (W_k/2)^2}} \right)^{-1}. \quad (11)$$

We test these three functions by comparing the relative magnitudes of  $d_1(i, j)$ ,  $d_2(i, j)$  and  $d_3(i, j)$ . Our results show that if we arrange them in a descending order, the order will be very similar with each other despite the differences in the specific formulas used. Since the dynamic programming in the next Subsection depends on  $d(i, j)$ , this implies relative insensitivity to the choice among the formulas above.



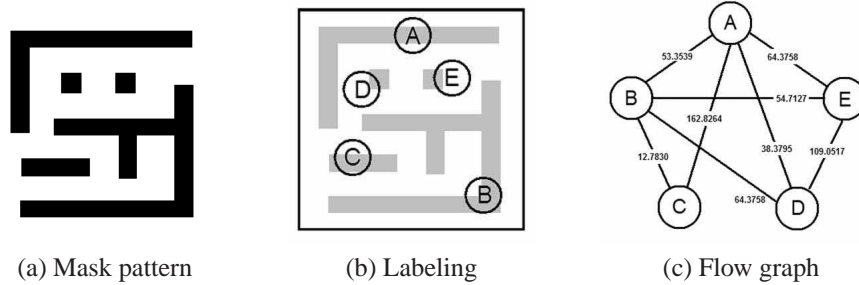


Fig. 5. Example pattern for conversion of a mask pattern into a graph.

### 3.3. Dynamic programming

Dynamic programming is a powerful optimization tool to search for the shortest path going from one point to another via several intermediate nodes. In our phase initialization problem, the goal is to alternate the phase as abruptly as possible. If we assign phase 0 to an Object A, then we should assign  $\pi$  to the one with the largest interaction with A. We observe that this is equivalent to arranging the objects in an order such that the overall interaction is maximized. Using our terminology discussed above, this is equivalent to minimizing the cost.

Figure 5 shows a system of five objects as an example. Interacting objects are joined by lines in the flow graph. The numbers beside the lines are the cost of going from one point to another. We want to find the cheapest path to pass all nodes without repeating. In the following we use  $X_k$  to denote the node at stage  $k$ . Suppose we start from node  $X_0 = D$  in stage 1 (this is arbitrary). Then at stage 2, the possible destinations are A, B, and E. So we can compute the cost to reach these nodes:

$$\begin{aligned} X_1X_2 = DA &= 38.3795 \\ X_1X_2 = DB &= 64.3758 \\ X_1X_2 = DE &= 109.0517. \end{aligned}$$

Starting from stage 3, we shall refer to the tree diagram Fig. 6. As shown, if we want to reach A, we only have one choice (through DE), and the cost is

$$X_1X_2X_3 = DEA = 109.0517 + 64.3758 = 173.4275.$$

If we want to reach B, we have two choices:

$$\begin{aligned} X_1X_2X_3 = DAB &= 38.3795 + 53.3539 = 91.7334 \\ X_1X_2X_3 = DEB &= 109.0517 + 54.7127 = 163.7644, \end{aligned}$$

But by principle of dynamic programming, we shall pick the one with lower cost. So to reach B at stage 3, we shall choose path DAB.

Similarly, to reach C, we have two choices:

$$\begin{aligned} X_1X_2X_3 = DAC &= 38.3795 + 162.8364 = 201.2059 \\ X_1X_2X_3 = DBC &= 64.3758 + 12.7830 = 77.1588, \end{aligned}$$

and so we should choose DBC.

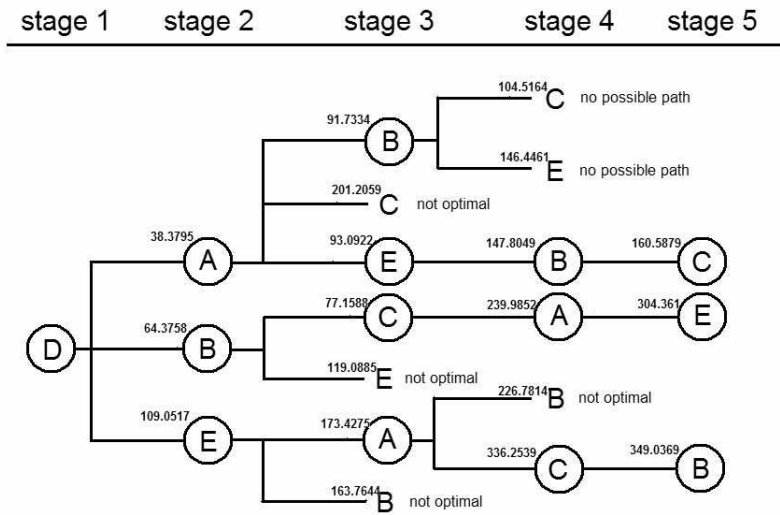


Fig. 6. Illustration of DP computation. Numbers represent the accumulated costs to reach the node. Circled nodes are nodes with lowest accumulated cost at their stages.

For  $E$ , we have

$$X_1X_2X_3 = DAE = 38.3795 + 64.3758 = 93.0922$$

$$X_1X_2X_3 = DBE = 64.3758 + 54.7127 = 119.0885,$$

and so we should choose  $DAE$ .

Now we go to stage 4. The logic is the same as before: we first pick a destination at stage 4 (say  $B$ ), then we list out all possible ways to go to  $B$ , and choose the one with lowest *accumulated* cost. Repeating for other destinations, we can complete stage 4. See circled nodes at stage 4 in Fig. 6. Finally at stage 5, we can see that there are three feasible paths:

$$X_1X_2X_3X_4X_5 = DAEBC = 160.5879$$

$$X_1X_2X_3X_4X_5 = DBCAE = 304.3610$$

$$X_1X_2X_3X_4X_5 = DEACB = 349.0369.$$

Choosing the one with lowest accumulated cost, we see that the optimal path is  $DAEBC$  for the whole graph.

Note that among all the paths considered from stage  $k - 1$  to stage  $k$ , we only need to consider the *survivors*. For example from stage 2 to stage 3, we had two paths to go to  $B$  ( $DAB$  and  $DEB$ ). By comparing the accumulated cost we chose path  $DAB$ , and discarded  $DEB$ . So starting from stage 3 onwards we only considered the path  $DAB$  (as circled in Fig. 6).

In the above discussion we choose the starting point to be  $D$ . However, in order to search for the best possible path, we should start the dynamic programming at all possible nodes. Thus, we loop over all starting nodes. With each starting node, we run through the above mentioned algorithm, and give the best path. Among all these paths (best with respect to different starting points) we choose the one with the lowest cost.

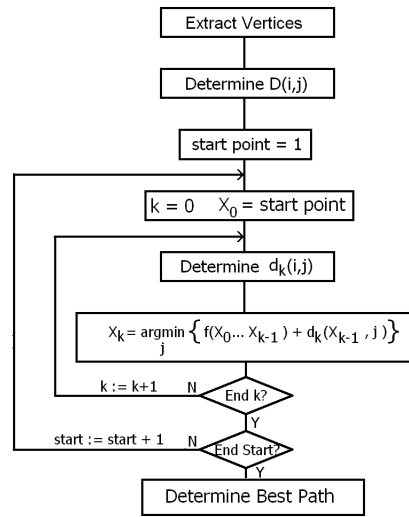


Fig. 7. Flow chart of the phase initialization algorithm, where function  $f$  denotes the accumulated cost.

Having the optimal path, we assign phases alternatively  $(0, \pi, 0, \pi, \dots)$  to the objects according to the optimal sequence. Figure 7 summarizes the above in a flow chart for the phase initialization algorithm.

#### 4. Results

Here we discuss some results using the above phase initialization algorithm. Suppose Fig. 5(a) is a pattern to be initialized. As we have just calculated, the best path is *DAEBC*. We assign phases according to this order:  $D(0)$ ,  $A(\pi)$ ,  $E(0)$ ,  $B(\pi)$ ,  $C(0)$ . The resulting pattern is shown in Fig. 8(a). Several other examples can be found in (b) to (f). In these figures, we display white = amplitude 1 phase 0, black = amplitude 1 phase  $\pi$ , gray = amplitude 0. We can see that for complicated patterns, the dynamic program indeed finds reasonable solutions as well. If there is no phase conflict, such as in Fig. 8(b), the algorithm assigns phases from left to right, and up to down. If there are phase conflicts, the algorithm can assign phases so as to minimize the conflict, as demonstrated in Fig. 8(c) to (f).

The importance of this initialization scheme is the improvement of the search space. Since the algorithm is a local searching algorithm, by initializing the phases we provide a better starting point to search. Therefore, even though the algorithm searches locally, the solution is significantly improved compared with not having initialization. We demonstrate this in Fig. 9. This is an XOR gate of size  $1700\text{nm} \times 1400\text{nm}$ , and with critical dimension  $90\text{nm}$ . We simulate the source as monochromatic laser with wavelength  $\lambda = 193\text{nm}$ , numerical aperture  $\text{NA} = 0.85$ . As in numerical implementation, we represented the point spread function  $H(x, y)$  as an array of size  $121 \times 121$ , which is adequate to cover 98% energy. For resist development, we set the sigmoid function cut-off threshold as  $t_r = 0.3$ , and the sharpness of the function  $a = 25$ . The weighting constant in the algorithm is  $\gamma = 0.01$ . The initial guess follows from Section 3.

Figure 9(a) shows the layout pattern. If we do not perform any correction to the layout pattern and use it as the mask, the aerial image is seriously blurred and the resulting pattern deviates substantially from the desired one, as shown in (b) and (c). Some objects even merge together.

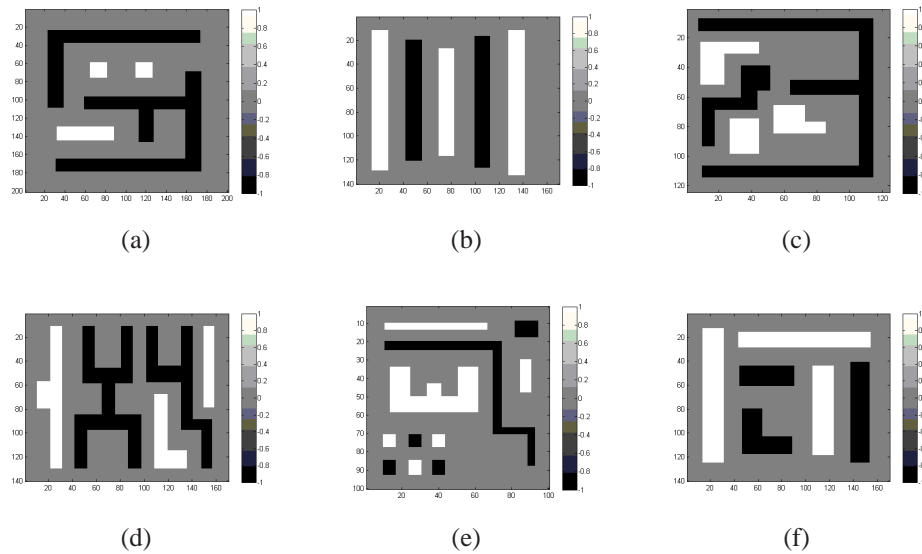


Fig. 8. Results of phase initialization. We used Eq. (11) as the distance function.

Now if we use inverse imaging without phase initialization, the 3D view of the aerial image is shown in (d). This can be compared with the 3D view of the aerial image when phase initialization has been used (with phase assignment shown in Fig. 8(d)). We can see that image slope is improved, because using initialization scheme the intensity can reach zero at pixels in between two objects. Besides, the intensity of undesired ripples are suppressed, so that we have better fidelity. Intensity view of the aerial image with phase initialization is shown in (f) for an alternative perspective. The resulting circuit pattern is given in (g). We can see that it shows a remarkable improvement in fidelity with the desired mask pattern, compared with (c) and the case without phase initialization. For the latter, we also count the number of mismatch pixels with the desired pattern. There are 373 pixel errors with phase initialization, compared to 494 without the initialization scheme.

We can further look at the use of the initialization scheme by comparing step-by-step whether it is used for a particular pattern. This is shown in Fig. 10. In (a), the pattern is initialized with alternating phases in the four bars. This is certainly intuitive to an experienced engineer. However, without a proper algorithm the computer may simply initialize the four bars to the same phase, as shown in (b). They can then lead to very different results. The optimal mask patterns for the case with our initialization and without are given in (c) and (d), respectively. Both are three-toned images because they are alternating phase-shifting masks. The resulting aerial contours after the imaging are shown in (e) and (f). Alternatively, we also plot the intensity at a particular row of these figures in (g) and (h), respectively. It is clear that with our initialization scheme, the intensity of the bars can reach more extreme values, and the intensity slopes are steeper, which are desirable.

Indeed, image robustness is of increasing concern in lithography, as pattern-printing well under nominal conditions may become out of specifications with process fluctuations. Two important metrics of robustness are the contrast and dose sensitivity of our design. Using a two-rectangle example, we make a cross section cut of the aerial image at the middle row, and plot the intensity curves, as shown in Fig. 11. We can then numerically evaluate the contrast,

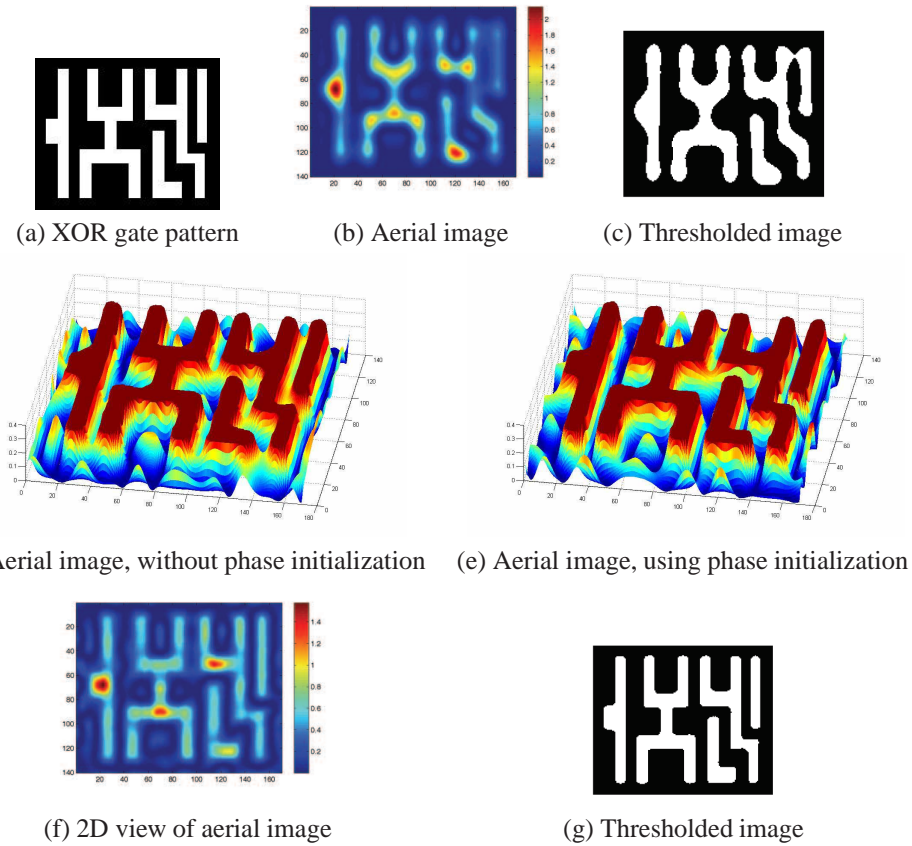


Fig. 9. Inverse imaging applied to an XOR gate pattern.

which is defined as

$$V = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \times 100\%. \quad (12)$$

It is the measure of the difference between the largest and the smallest intensity value. High contrast is always desired because it implies sharper edges.

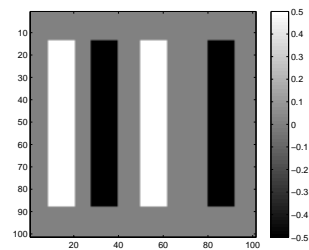
In this example, the contrast of the aerial image using original layout as mask is

$$V_0 = \frac{0.7413 - 0.6566}{0.7413 + 0.6566} \times 100\% = 6.06\%,$$

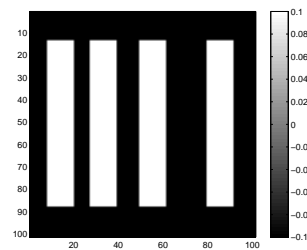
where the values are taken at Distance = 490nm, which is the falling edge of the first rectangle. Without performing any OPC, the aerial images of the two rectangles merge because of the wide point spread function  $H(x,y)$  (large  $\lambda$ ). Therefore, the contrast is very low. On the other hand, the contrast of the aerial image using optimal PSM generated by inverse imaging is

$$V_{\text{PSM}} = \frac{2 - 0}{2 + 0} \times 100\% = 100\%.$$

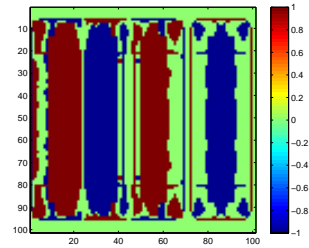
Since we use PSM, the alternating phases cause the intensity to drop to zero at the trough. This is the reason why contrast can be 100%.



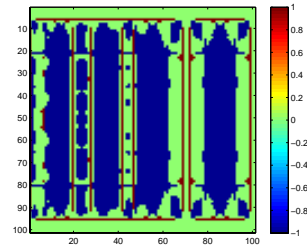
(a) With phase initialization



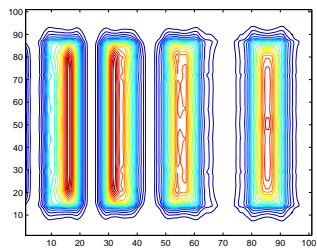
(b) Without phase initialization



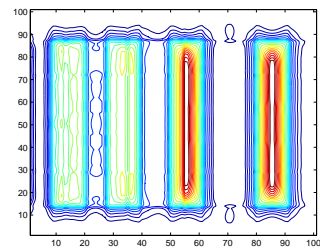
(c) Optimal mask pattern resulting from (a)



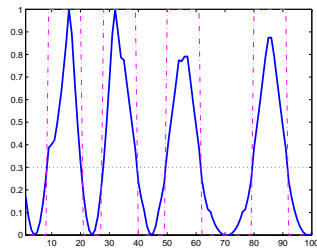
(d) Optimal mask pattern resulting from (b)



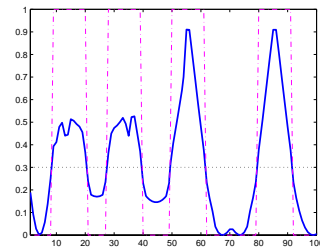
(e) Aerial contour resulting from (a)



(f) Aerial contour resulting from (b)



(g) Intensity curve at row 50 from (a)



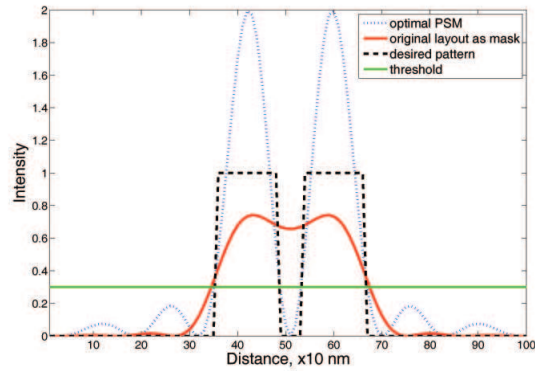
(h) Intensity curve at row 50 from (b)

Fig. 10. Four-bar example.

Another measure that we can compare is the dose sensitivity, which is usually quantified as the normalized image log slope [4]:

$$\text{NILS} = \left| \frac{\text{CD}}{I_{\text{threshold}}} \frac{dI}{dx} \Big|_{I_{\text{threshold}}} \right|, \quad (13)$$

where  $I_{\text{threshold}}$  is the threshold intensity (i.e.  $t_r$ ), CD is the critical dimension. NILS measures



(d) Intensity curve using a two-rectangle example

Fig. 11. Two-rectangle example.

the tolerance of the aerial image when facing fluctuation in dose concentration. Therefore, a higher value is preferred.

In this example, if we use the original layout as the mask, NILS can be found as

$$\text{NILS} = \left| \frac{50 \times 10^{-9}}{0.3} \times \frac{0.681 - 0.668}{10 \times 10^{-9}} \right| = 0.2167,$$

where we have set  $t_r = 0.3$ ,  $\text{CD} = 50\text{nm}$ , and use forward difference with  $\Delta x = 10\text{nm}/\text{pixel}$ . On the other hand, if we use the optimal PSM, the NILS is

$$\text{NILS} = \left| \frac{50 \times 10^{-9}}{0.3} \times \frac{0.5475 - 0.2581}{10 \times 10^{-9}} \right| = 0.4823.$$

It is clear that the algorithm also improves the tolerance to dose fluctuation significantly.

## 5. Conclusion

We demonstrated an initialization scheme for a pixel-based optimization algorithm to synthesis phase-shifting masks in optical lithography. The synthesis is an inverse problem where existence of solutions is not always guaranteed, and we used conjugate gradient and active set method in the iterations. For alternating phase-shifting mask design, we proposed a phase initialization scheme to further improve the pattern fidelity and image slope. Results showed that the aerial images generated by our optimal phase-shifting masks generate fewer pattern error, and have good contrast and tolerance to dose variation.

## Acknowledgment

This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China under Project HKU 7139/06E.