Fast Inverse Lithography using Machine Learning

Tapan Shah TAPANSHAH@TIFR.RES.IN
Onkar Dabeer ONKAR@TCS.TIFR.RES.IN
School of Technology and Computer Science,
Tata Institute of Fundamental Research,
Mumbai, India, 400005

Abstract

In this paper, we use Random Forests to learn the inverse lithography function. We choose a suitable feature vector based on the structure of the lithography transfer function and use an ensemble of decision trees as the classifier. While our method yields outputs comparable (in visual quality as well as BER) to previous work, it is 10 folds faster.

1. Inverse Lithography

In semiconductor industry, optical lithography (OL) is a method to transfer desired circuit patterns from a mask on to a semiconductor wafer using light. Due to the low pass properties of an OL system, the etched circuit patterns do not resemble the masks and can be severely distorted (for example, see Figure 1). This necessitates methods for suitable mask design, especially for very small feature dimensions, and one such method is to pose it as an inverse imaging problem (Poonawala & Milanfar, 2007)

2. System Model and Problem formulation

An equivalent model for the inverse lithography problem is stated as follows. Let $\mathbf{I} \in \{0,1\}^{N\times N}$ be the discrete input binary image. Let $F : \{0,1\}^{N\times N} \rightarrow \{0,1\}^{N\times N}$ be the forward model (i.e. the model for the lithography system) which defines a non-linear mapping between two binary matrices. Thus the output image is given by $\mathbf{O} = F(\mathbf{I})$. The forward model $F(\cdot)$ is composed of a convolution with a low pass filter $\mathbf{H} \in \mathcal{R}^{N\times N}$ followed by a threshold operator $\Lambda_\tau(\cdot)$

Appearing in Proceedings of the 1st Indian Workshop on Machine Learning, IIT Kanpur, India, 2013. Copyright 2013 by the author(s).

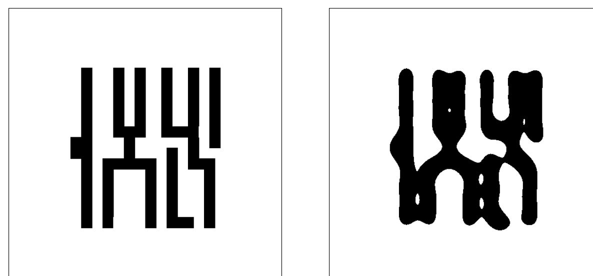

Figure 1. First figure is the desired layout of a XOR gate. The next figure shows the pattern etched on the wafer if the desired pattern is used as mask for a typical OL system.

defined by

$$\Lambda_\tau(z) = \begin{cases} 0 & \text{if} \quad 0 \le |z|^2 \le \tau \\ 1 & \text{if} \quad \tau < |z|^2. \end{cases}$$

It is common to assume an ideal low pass filter (Sherif et al., 1995). Therefore, the output image $\mathbf{O}$ is given by

$$\mathbf{O} = \Lambda_\tau(\mathbf{H} * \mathbf{I}), \tag{1}$$

where $*$ is the 2-D convolution operator and $\Lambda_\tau(\cdot)$ is applied element-wise.

Suppose we are given a desired output binary image $\mathbf{O}^* \in \{0,1\}^{N\times N}$. Then the goal is to find an inverse mapping $G : \{0,1\}^{N\times N} \rightarrow \{0,1\}^{N\times N}$ such that $F(G(\mathbf{O}^*)) = \mathbf{O}^*$. No results for existence of an inverse mapping for binary input masks are available. Consequently, most previous methods used in inverse lithography pose the inverse problem as an optimization problem - they make the best effort as per a suitable cost. We reformulate the above problem as a classification problem. For each pixel (x, y) of $\mathbf{O}^*$, we build a suitable feature vector $\mathbf{f}(x, y) \in \mathcal{F}$ - a suitable feature space. A classifier $C : \mathcal{F} \rightarrow \{0,1\}$ yields the value of the desired mask $\mathbf{I}^*(x, y) = C(\mathbf{f}(x, y))$.

3. Machine Learning based Inverse Lithography (ML-ILT)

3.1. Feature Selection

The forward map preserves rotation and translation. Therefore, the inverse map should also satisfy these properties. For each pixel (x, y), we ensure preservation of translation by picking the feature vector $\mathbf{f}(x, y)$ so that it depends on the pixel values in a neighborhood of (x, y) but not on the location indices (x, y) and the same classifier $C(\cdot)$ is used for all pixels irrespective of their location. To preserve rotation, we slice the remaining pixels in D radial directions and sample R pixels in each direction. The binary string of length R is converted into an integer between 0 and $2^R - 1$. The resultant D length vector $\mathbf{p}$ consists of integers between 0 and $2^R - 1$ and we compute the histogram of this vector.

3.2. Dimensionality reduction:

We observe that for typical values of R, only a few of these bins are populated indicating scope for dimensionality reduction. Locality sensitive hashing (LSH) (Gionis et al., 1999) is one such technique and we use this to convert the R length string into a K length string. The resulting D length vector $\hat{\mathbf{p}}$ consists of integers between 0 and 2^{K-1}. We take $\mathbf{f}(x, y)$ to be the histogram of $\hat{\mathbf{p}}$ and it has the desired invariance properties. We show that $K = \log R$ is a good choice.

3.3. Training and choice of classifier

We choose 256×256 images with random layout generated with Manhattan geometry and apply the forward map $F(\cdot)$, extract features $\mathbf{f}(x, y)$ from the output images, and learn classifiers to predict the known input pixels. The inherent hardness (proved to be NP-hard in (Granik, 2006)) of solving the inverse problem forces us to use non-parametric classifiers like Random Forest (Breiman, 2001).

3.4. Scalability

The fastest known pixel based algorithms scale as $O(N^2 \log N)$ (Poonawala & Milanfar, 2007; Granik, 2006). In comparison, we show that our algorithm scales linearly with number of pixels ($O(N^2)$).

4. Performance Evaluation

We evaluate our algorithm in terms of bit error rates (BER) and running time and compare it with the gradient based optimization (GBO) described in (Poonawala & Milanfar, 2007). For XOR, we show the mask and resulting output using our method in Figure 2. In Table 1, we tabulate the performance averaged over 10 random layouts with Manhattan geometry for different image sizes.

Table 1. Comparing ML-ILT and GBO for with an edge layer of 2.

N	ML-ILT		GBO	
	BER	Average runtime (s)	BER	Average runtime (s)
256	0.0480	37	0.0487	370
512	0.0490	154	0.0493	1562
1024	0.0520	620	0.0585	6001

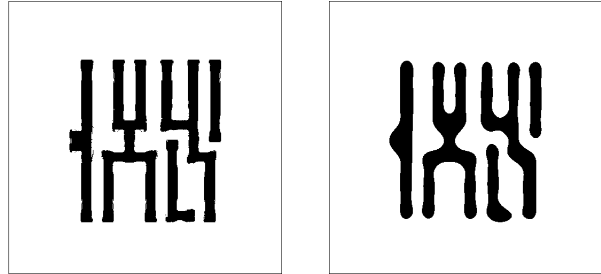

Figure 2. BER=0.0280 and running time=64 s.

References

Breiman, Leo. Random forests. *Machine learning*, 45 (1):5–32, 2001.

Gionis, A., Indyk, P., and Motwani, R. Similarity search in high dimensions via hashing. In *Proceedings of the International Conference on Very Large Data Bases*, pp. 518–529, 1999.

Granik, Yuri. Fast pixel-based mask optimization for inverse lithography. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 5(4): 043002–043002–13, 2006.

Poonawala, A. and Milanfar, P. Mask design for optical microlithography mdash;an inverse imaging problem. *IEEE Transactions on Image Processing*, 16(3):774 –788, March 2007.

Sherif, S., Saleh, B., and De Leone, R. Binary image synthesis using mixed linear integer programming. *IEEE Transactions on Image Processing*, 4(9):1252 –1257, sep 1995. ISSN 1057-7149. doi: 10.1109/83.413169.