Mask Optimization for Directed Self-Assembly Lithography: Inverse DSA and Inverse Lithography

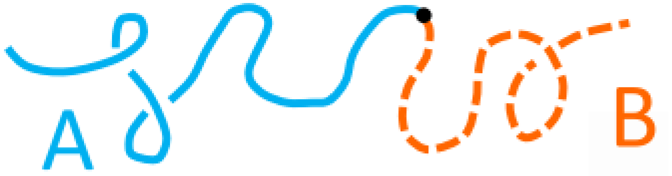
Seongbo Shim and Youngsoo Shin

School of EE, KAIST

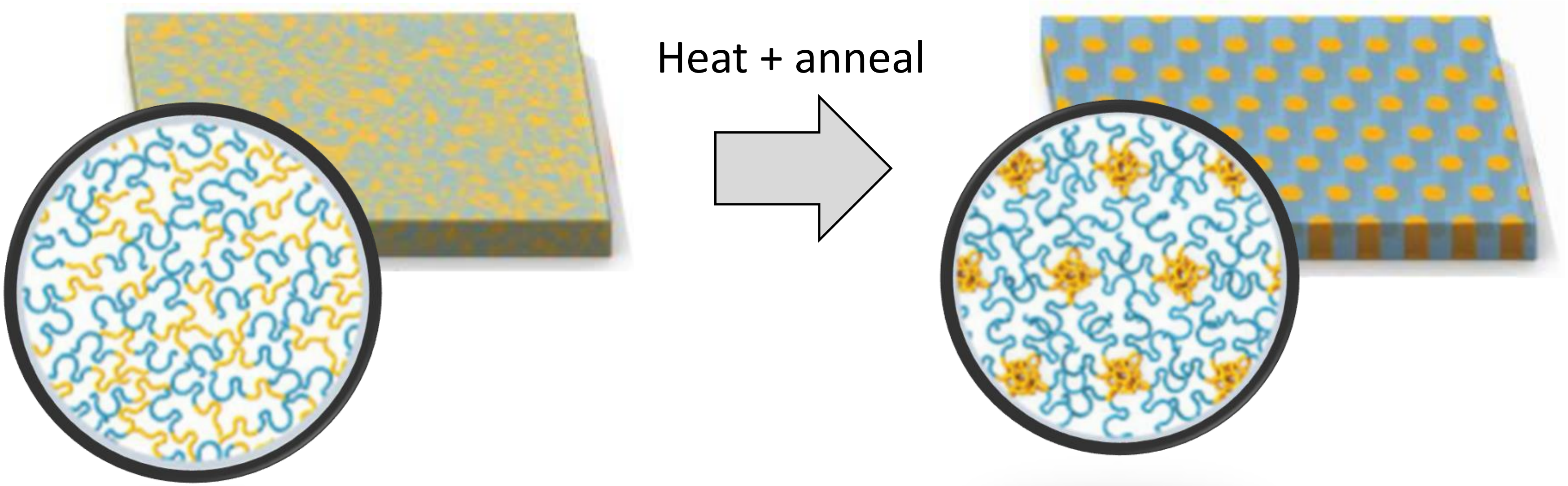
Outline

- Introduction
  - DSAL, Guide pattern (GP), Mask synthesis process
- **Inverse DSA**: Synthesizing ideal GP shape
  - Parameterizing GP, Synthesis algorithm
- **Inverse lithography**: Synthesizing GP mask image
  - Approximate cost gradient, Synthesis algorithm
- **DSAL mask design with lithography variations**
  - GP error tolerance, SRAF insertion
- Conclusion

Directed Self-Assembly (DSA)



Diblock copolymer (BCP)

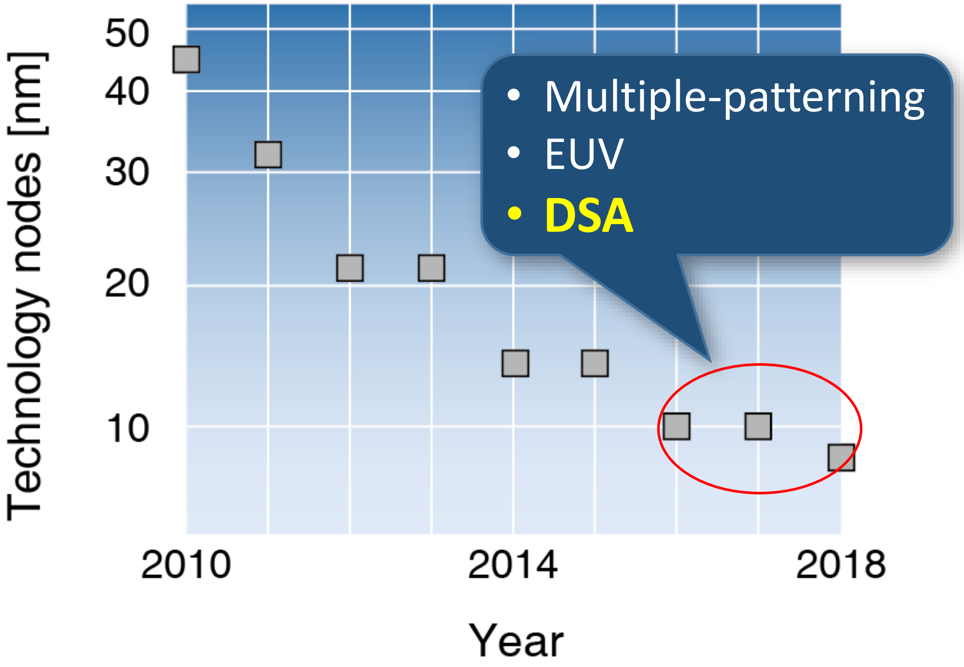


Heat + anneal

[Courtland, IEEE spectrum 2012]

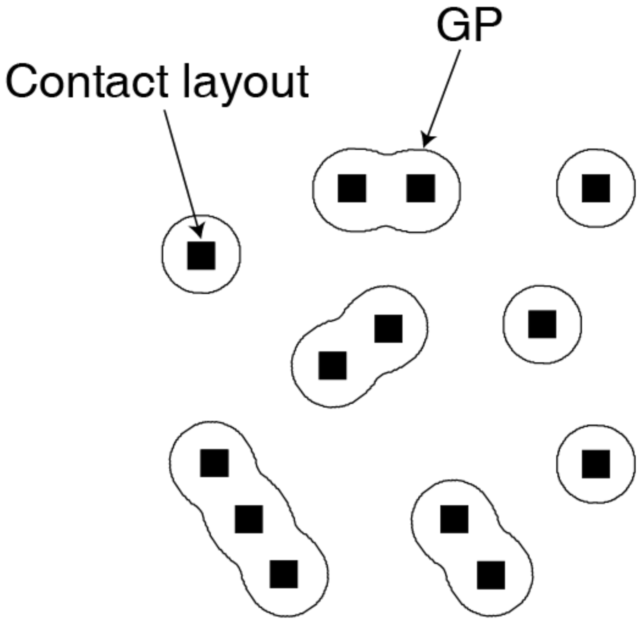
DSA using Guide Patterns (GPs)

- Contact and via in sub 10-nm cannot be patterned using optical lithography
- In DSAL: Local contacts are grouped → GP is patterned through optical litho → contacts are patterned through DSA (2 step process)

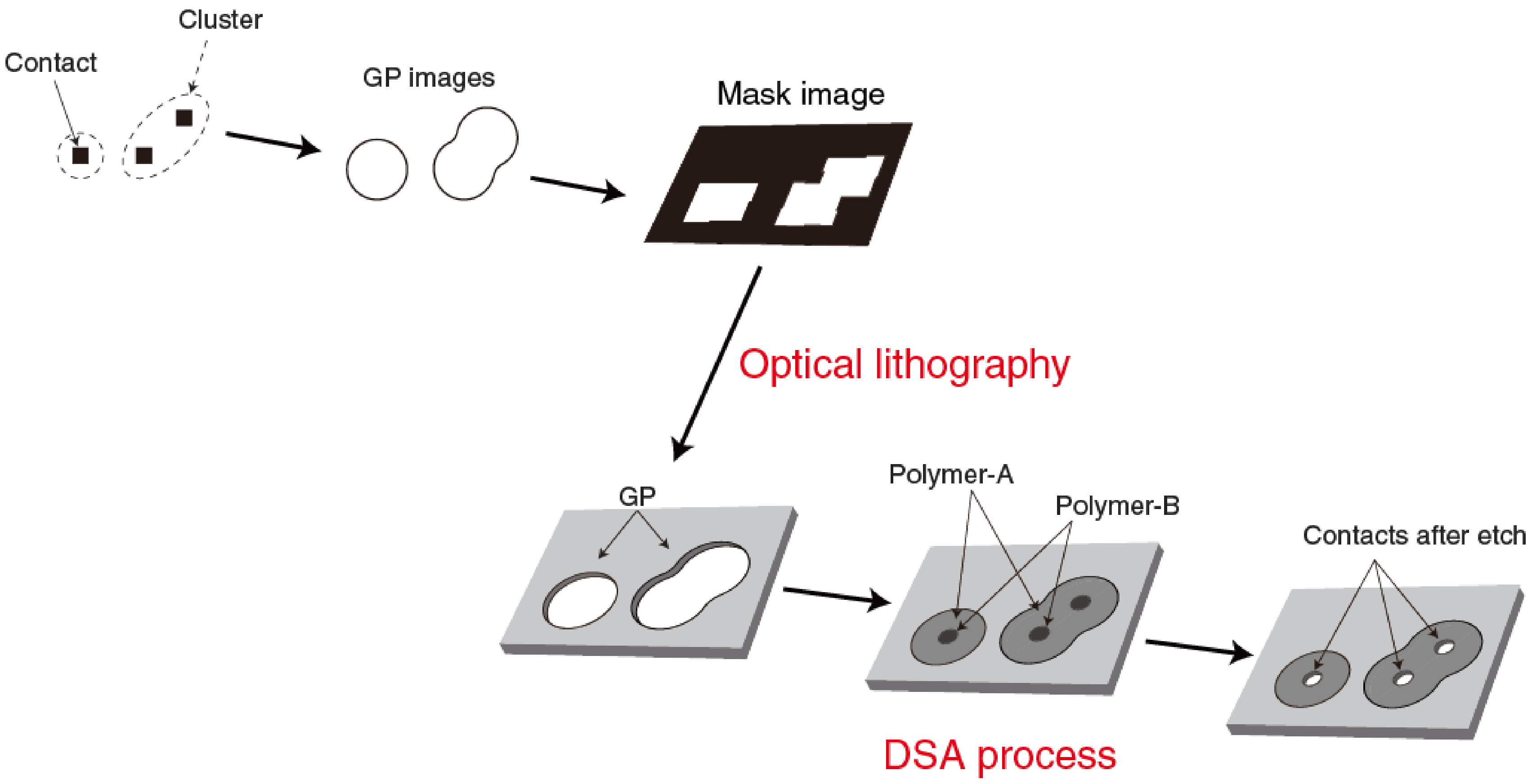


- Multiple-patterning
- EUV
- **DSA**

[Intel developer forum 2014]



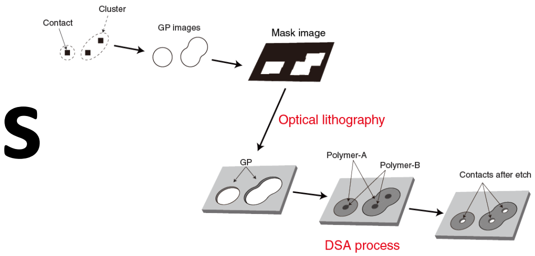
DSA Lithography (DSAL)

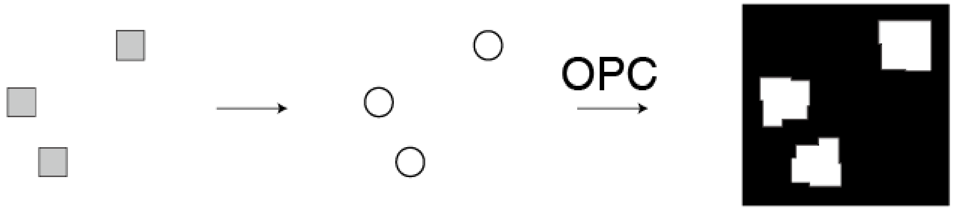


Optical lithography

DSA process

DSAL Mask Synthesis





Contact layout

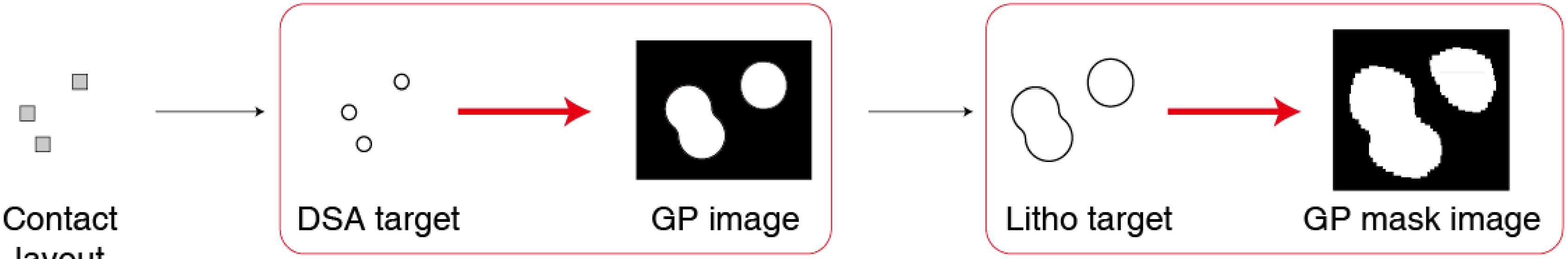
Litho target

Mask image

Mask synthesis for optical lithography

Inverse DSA

Inverse litho



Contact layout

DSA target

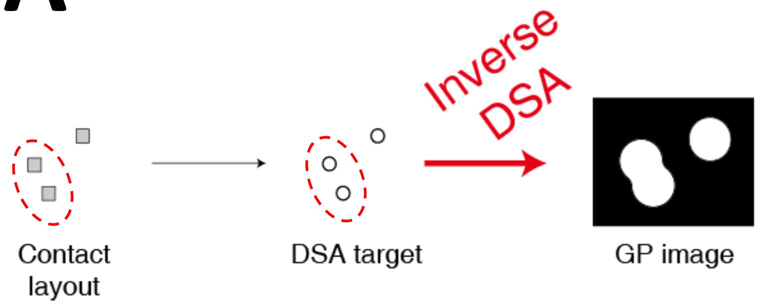
GP image

Litho target

GP mask image

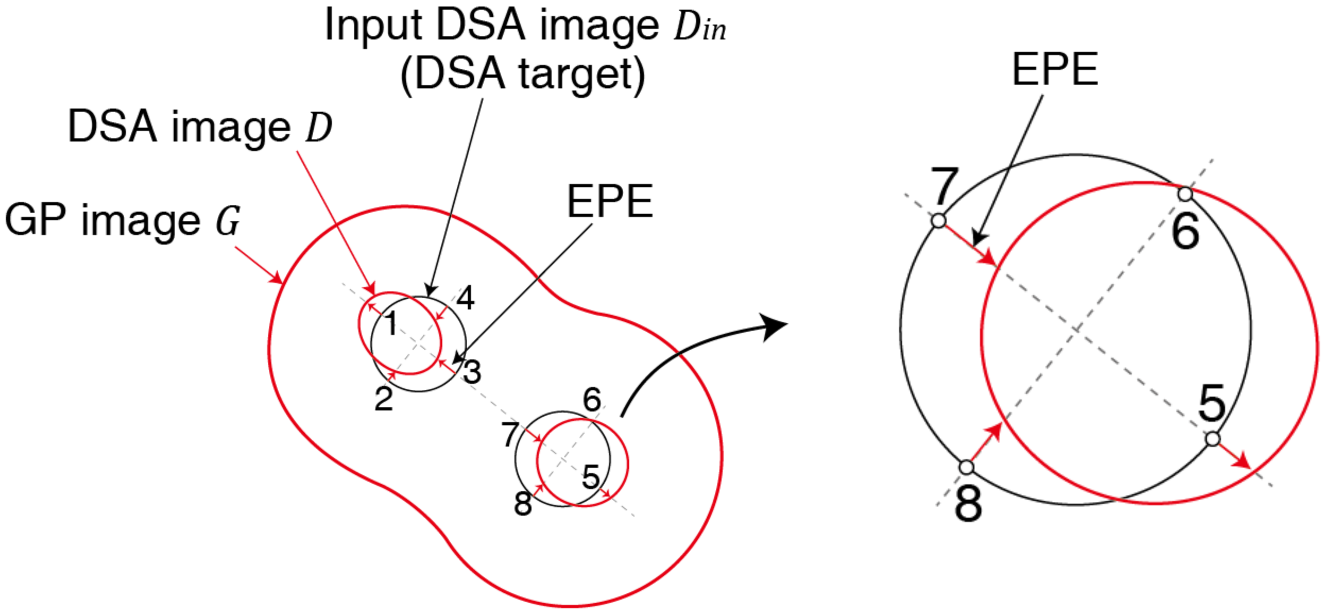
Mask synthesis for DSAL

Inverse DSA



Problem definition

- **Input**: A contact cluster
- **Output**: Ideal GP image
- **Objective**: Minimize max edge placement error (EPE)



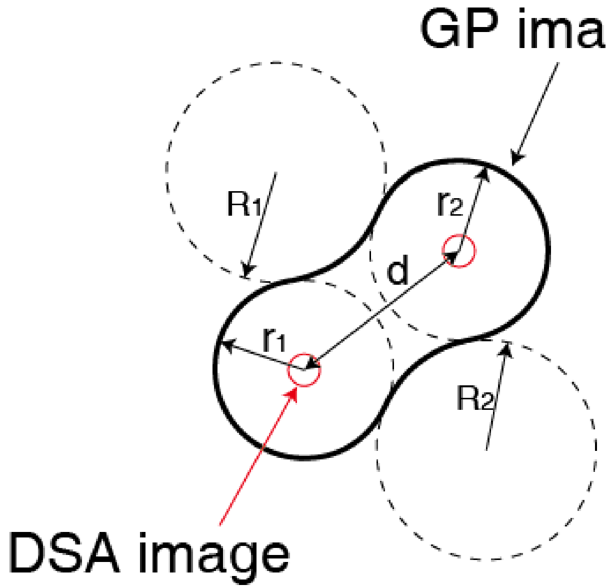
Inverse DSA

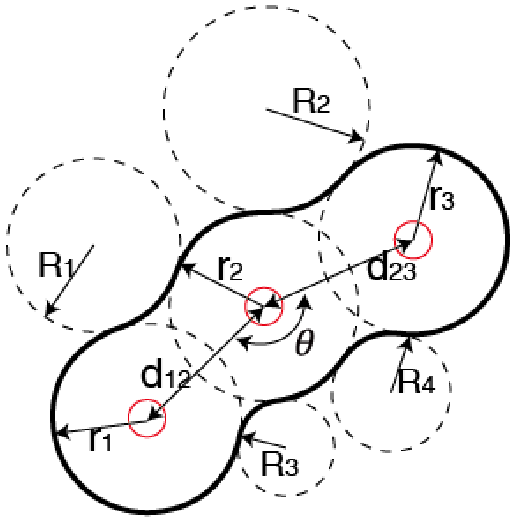
Parameterizing GP

- Represent a GP as a function of a few geometry parameters

$$\mathcal{G} = f(\mathbf{g}) = f(g_1, g_2, g_3, \cdots, g_n)$$

- Reduce complexity of inverse DSA





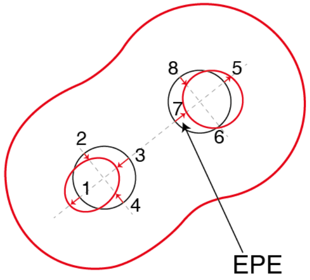
Inverse DSA

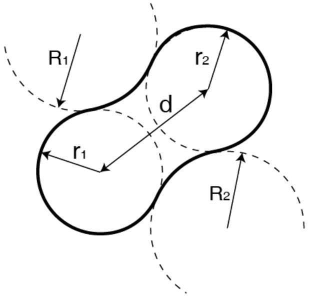
Algorithm

Input: a DSA image $\mathcal{D}_{in}$ of a cluster of contacts

Output: a GP image $\mathcal{G}$

```
$\mathcal{G} \leftarrow$ an initial GP image
$\mathcal{D} \leftarrow$ DSA_Simulation($\mathcal{G}$)
$\mathbf{e} \leftarrow$ Measure_EPE($\mathcal{D}_{in}$, $\mathcal{D}$)
repeat for max_iterations
    $\mathbf{M} \leftarrow$ Calc_Matrix($\mathcal{D}_{in}$, $\mathcal{G}$)
    $\mathcal{G} \leftarrow f(\mathbf{g})$, where $\mathbf{g}^{\mathrm{T}} \leftarrow \mathbf{g}^{\mathrm{T}} - \mathbf{M}^{-1} \times \mathbf{e}^{\mathrm{T}}$
    $\mathcal{D} \leftarrow$ DSA_Simulation($\mathcal{G}$)
    $\mathbf{e} \leftarrow$ Measure_EPE($\mathcal{D}_{in}$, $\mathcal{D}$)
    if $\max_{\forall i}(|e_i|) \leq$ max_EPE then Exit loop
return $\mathcal{G}$
```



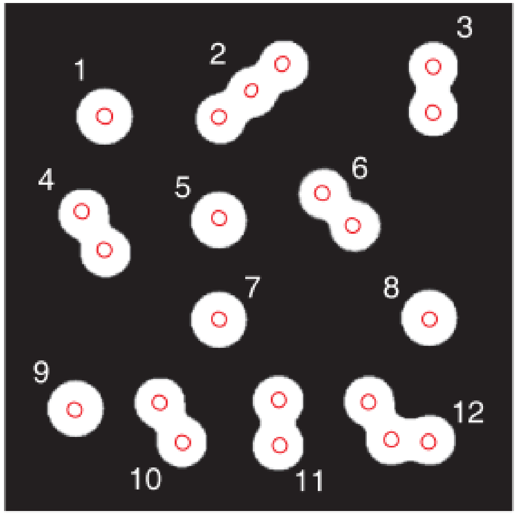


$$\mathbf{M} = \begin{bmatrix} \frac{\partial e_1}{\partial g_1} & \frac{\partial e_1}{\partial g_2} & \cdots & \frac{\partial e_1}{\partial g_n} \\ \frac{\partial e_2}{\partial g_1} & \frac{\partial e_2}{\partial g_2} & \cdots & \frac{\partial e_2}{\partial g_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_m}{\partial g_1} & \frac{\partial e_m}{\partial g_2} & \cdots & \frac{\partial e_m}{\partial g_n} \end{bmatrix}$$

Inverse DSA

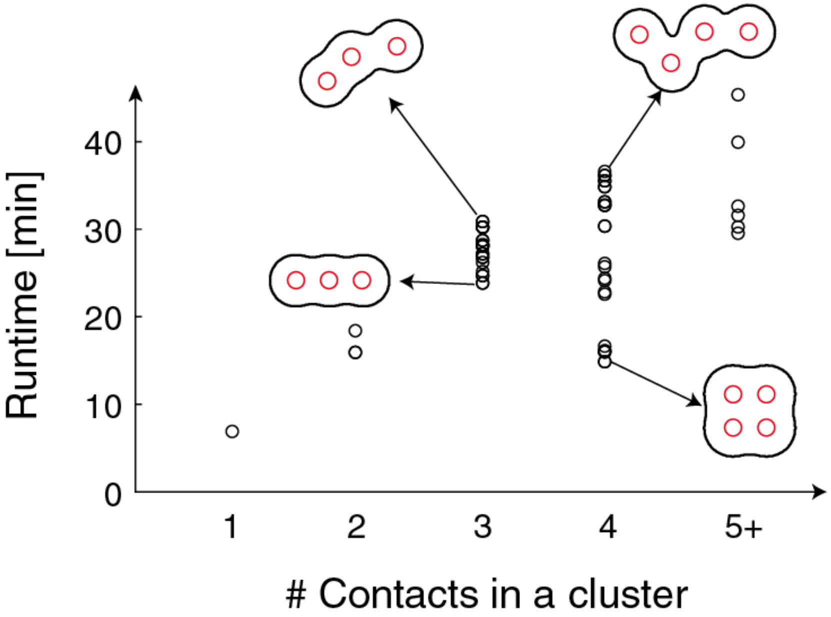
Experimental observations

- Inverse DSA is applied to each group of congruent clusters one by one
- Runtime increases with larger and more complex clusters



{1, 5, 7, 8, 9} {3, 4, 6, 10, 11} {2} {12}

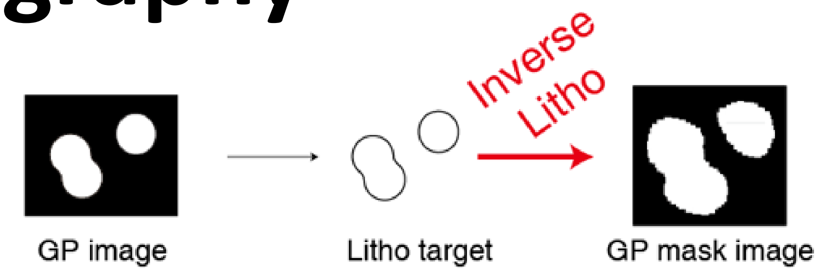
Groups of congruent clusters (GPs)

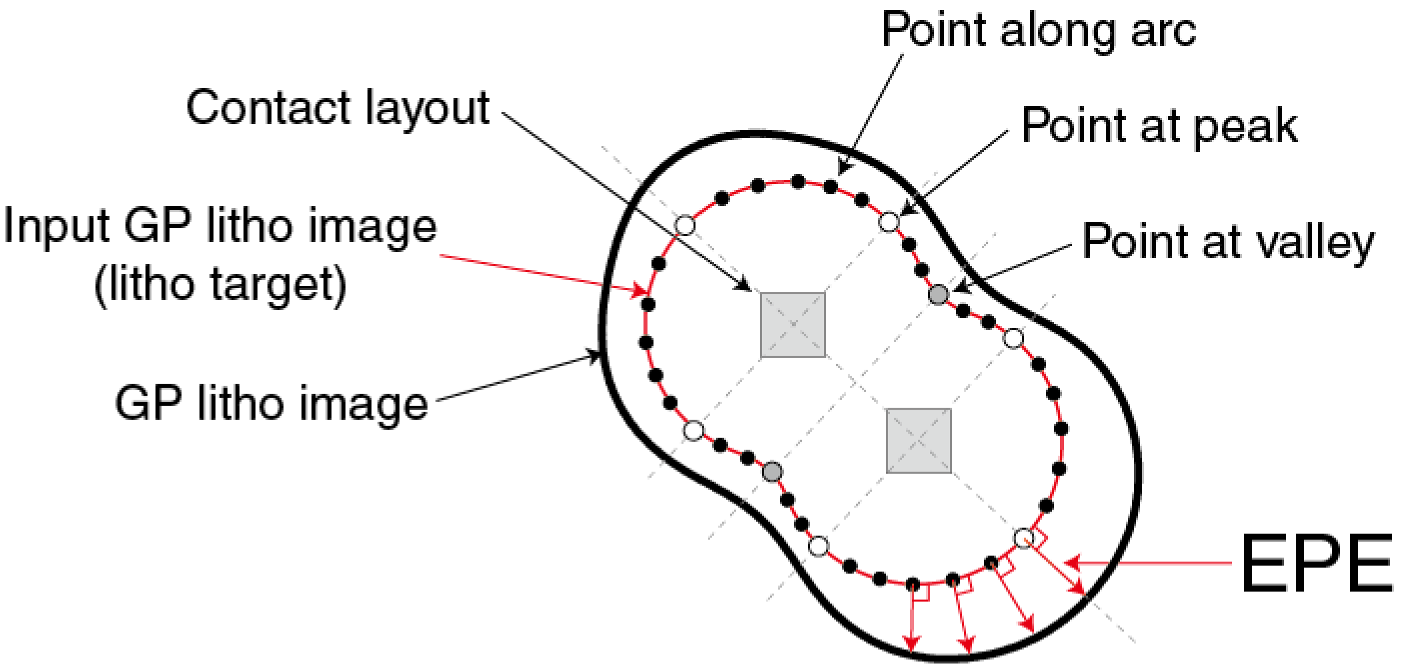


Inverse Lithography

Problem definition

- **Input**: an ieal GP image
- **Output**: a GP mask image
- **Objectives**: $minimize \sum_k |EPE_k|^2$





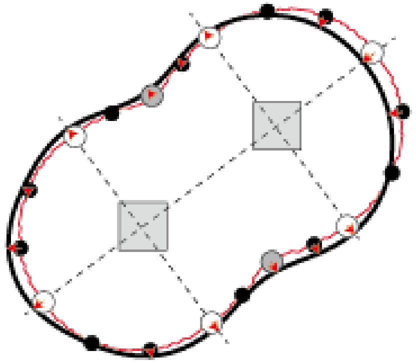
Inverse Lithography

Algorithm

Input: a GP litho image $\mathcal{L}_{in}$

Output: a GP mask image $\mathcal{M}$

```
$\mathcal{M} \leftarrow$ an initial GP mask image
$\mathcal{L} \leftarrow$ Litho_Simulation($\mathcal{M}$)
$C \leftarrow$ Cost($\mathcal{L}_{in}$, $\mathcal{L}$)
repeat for max_iterations
    $\mathcal{M} \leftarrow \mathcal{M} - k\nabla C$
    $\mathcal{M} \leftarrow$ Convert $\mathcal{M}$ to a binary mask
    $\mathcal{L} \leftarrow$ Litho_Simulation($\mathcal{M}$)
    $C \leftarrow$ Cost($\mathcal{L}_{in}$, $\mathcal{L}$)
    if $C$ increases OR $|\nabla C| \leq \epsilon$ then
        Roll back $\mathcal{M}$; exit loop
return $\mathcal{M}$
```



$$\nabla C = \left(\frac{\partial C}{\partial g_1}, \frac{\partial C}{\partial g_2}, \frac{\partial C}{\partial g_3}, \cdots, \frac{\partial C}{\partial g_n} \right)$$

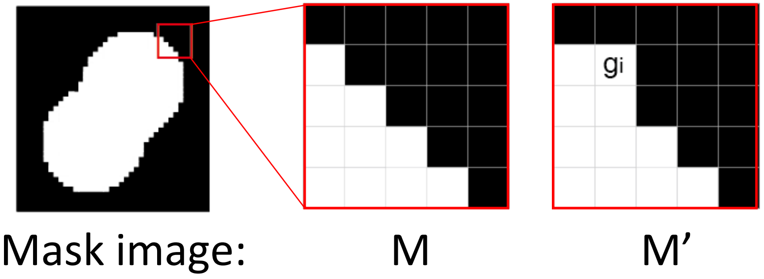
Inverse Lithography

Approximate cost gradient

- Calculation of ∇C

$$\frac{\partial C}{\partial g_i} = \sum_k |EPE'_k|^2 - \sum_k |EPE_k|^2$$

n times convolutions for n pixels



- **Approximate** ∇C

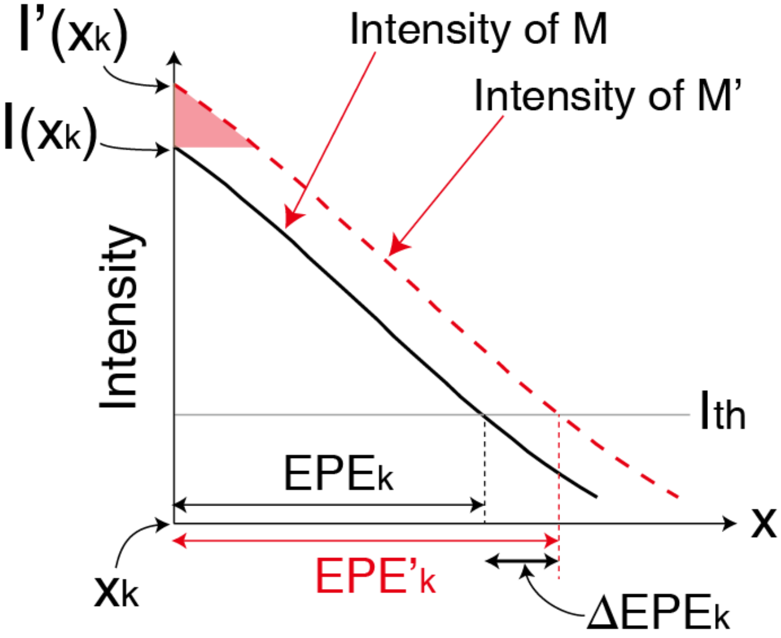
$$\frac{\partial C}{\partial g_i} = \sum_k (\Delta EPE_k)(2EPE_k + \Delta EPE_k)$$

$$\Delta EPE_k = \frac{I'(x_k) - I(x_k)}{dI(x_k)/dx}$$

From *I(x)*

From *I(x)*

Much faster than convolution

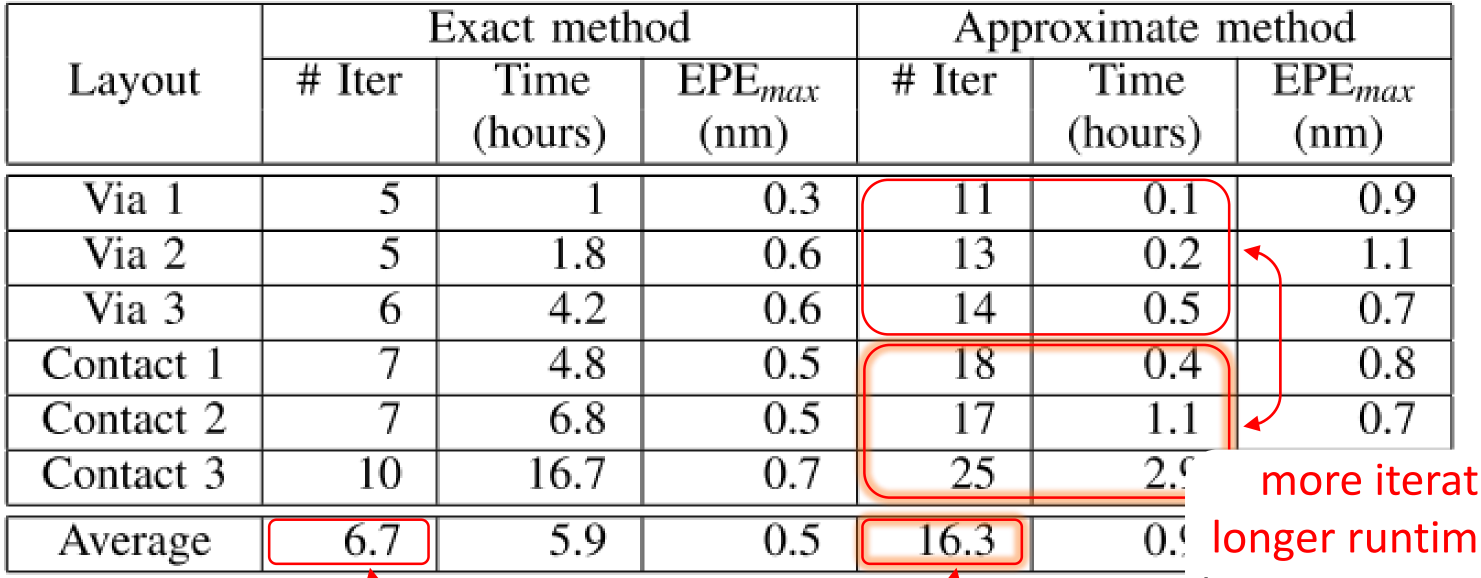


Inverse Lithography

Experiments: compare 2 methods

- **Exact method:** perform explicit litho simulations for computing ∇C
- **Our (approximate) method**: 6X faster, comparable accuracy

COMPARISON OF EXACT AND APPROXIMATE INVERSE LITHOGRAPHY



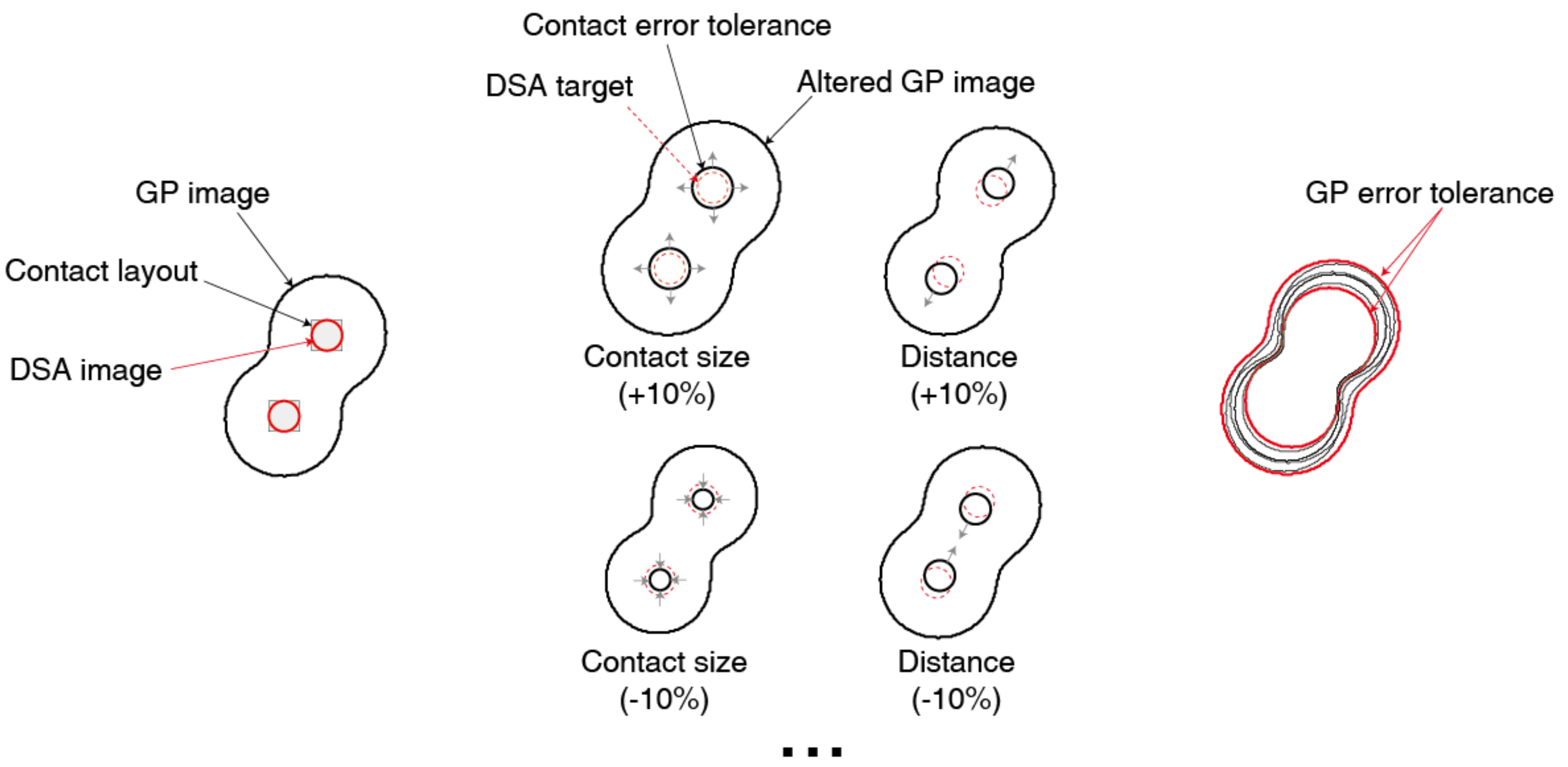
Layout	Exact method			Approximate method		
	# Iter	Time (hours)	EPE$_{max}$ (nm)	# Iter	Time (hours)	EPE$_{max}$ (nm)
Via 1	5	1	0.3	11	0.1	0.9
Via 2	5	1.8	0.6	13	0.2	1.1
Via 3	6	4.2	0.6	14	0.5	0.7
Contact 1	7	4.8	0.5	18	0.4	0.8
Contact 2	7	6.8	0.5	17	1.1	0.7
Contact 3	10	16.7	0.7	25	2.	
Average	6.7	5.9	0.5	16.3	0.	

more iterations & longer runtime due to less accuracy of approx.

2X more iterations due to approximation

Mask Design with Litho Variations

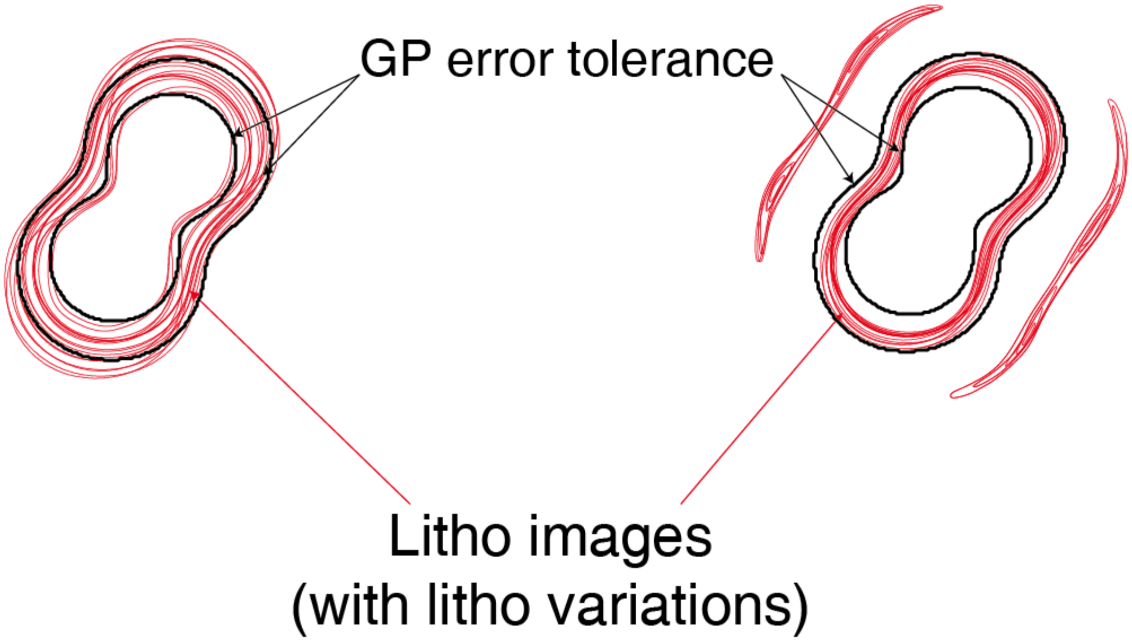
- GP may have errors due to litho variations → final contact error
- **Contact error tolerance**: DSA images should reside within some tolerance (e.g. ±10% contact size)
- **GP error tolerance**: GP litho images should reside within some tolerance



Mask Design with Litho Variations

DSAL SRAF

- **SRAF:** sub-resolution assist feature for constructive light interference
- **DSAL SRAF:** no problem of SRAF printing if no residue after DSA process
- # GPs with violation: 6.6% (no SRAF) → 0% (with SRAF)



Conclusion

- **Inverse DSA**: Synthesizing ideal GP shape
  - Parameterizing GP
- **Inverse lithography**: Synthesizing GP mask image
  - Approximate cost gradient
- **DSAL mask design with lithography variations**
  - GP error tolerance, SRAF insertion

Thank you