An Efficient Algorithm for Stencil Planning and Optimization in E-Beam Lithography(EBL)

Jiabei Ge[1] , Changhao Yan[1] , Hai Zhou[2] , Dian Zhou[3] , Xuan Zeng[1]

1. State Key Laboratory of ASIC & Systems, Microelectronics Department, Fudan University, China
2. Electrical Engineering and Computer Science Department, Northwestern University, U.S.A.
3. Electrical Engineering Department, University of Texas at Dallas, U.S.A.

Outline

- Introduction
- Problem Formulation
- Geometry-Frequency Balanced Algorithm
- Experimental Results
- Conclusion

Outline

- Introduction
- Problem Formulation
- Geometry-Frequency Balanced Algorithm
- Experimental Results
- Conclusion

Next generation lithography for sub-22 nm

Multiple patterning lithography (MPL) :

- widely used, expensive,
- difficult on the alignment requirement

Extreme ultra violet (EUV) lithography :

- not ready for volume production

Electron beam lithography (EBL) :

- maskless, low throughput

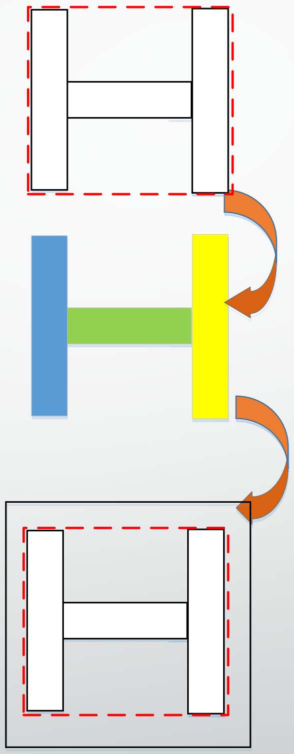
Electron beam lithography (EBL)

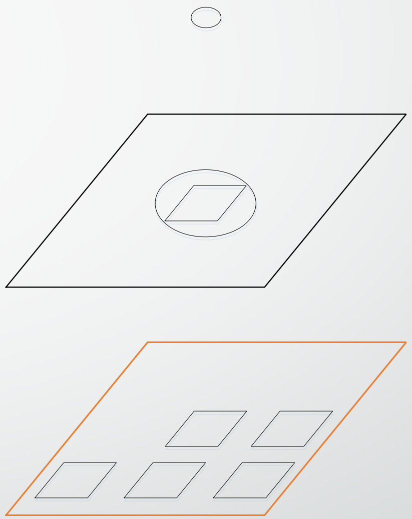
Variable Shaped Beam (VSB):

- Print one **rectangle** by one shot

Character Projection (CP) :

- Print one **character** by one shot
- A designed stencil





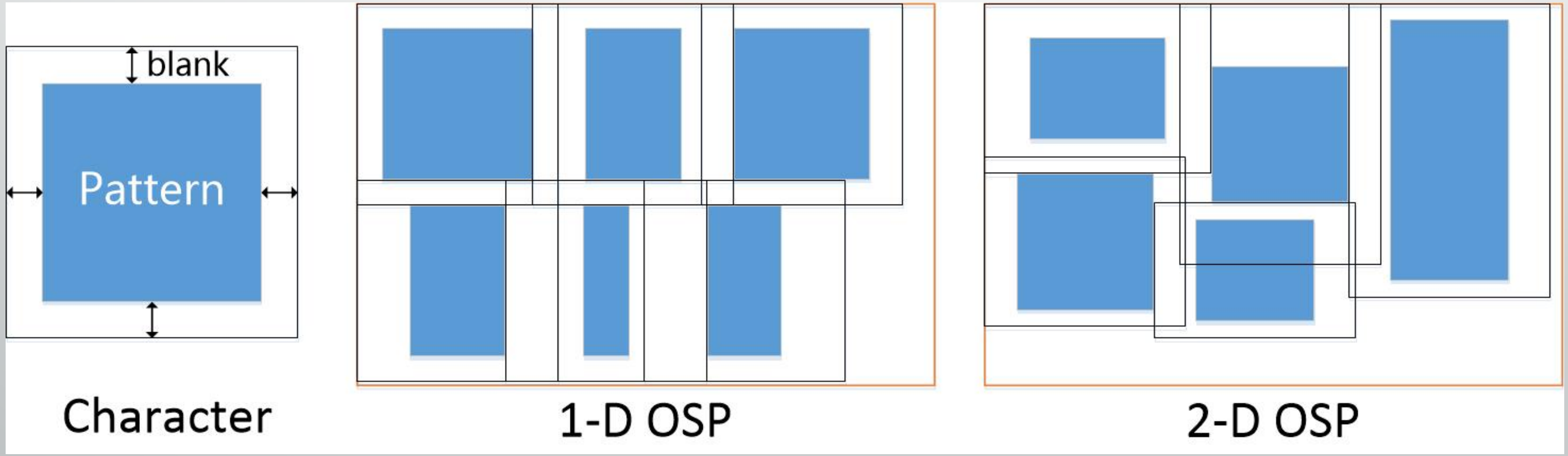
Overlapping-aware Stencil Planning (OSP)

1-D OSP :

- the character height and vertical blanks are equal
- often used in standard cell designs

2-D OSP :

  - horizontal and vertical blanks of a character may be different



Outline

- Introduction
- **Problem Formulation**
- Geometry-Frequency Balanced Algorithm
- Experimental Results
- Conclusion

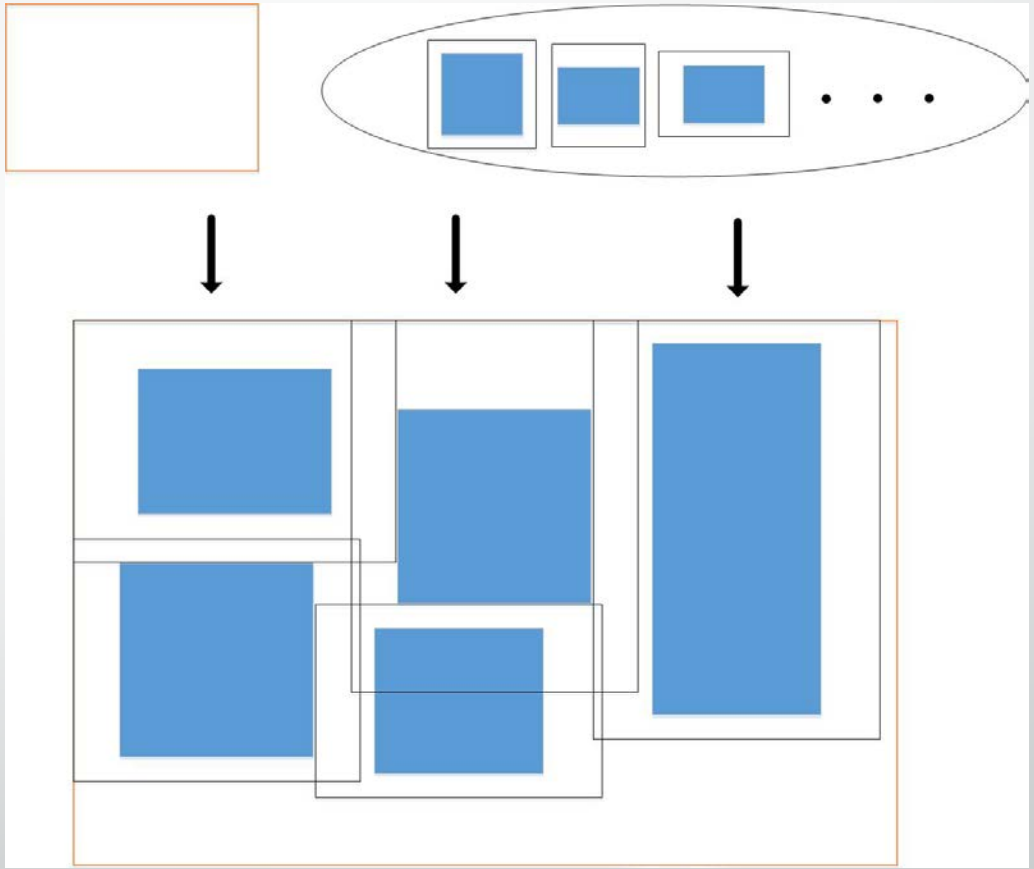
Problem Formulation(1/2)

Given:

- An empty stencil
- A character set

Output :

- A designed stencil with characters on it



Problem Formulation(2/2)

Given:

- An empty stencil
- A character set C

Object:

- Obtain a subset $C_s \in C$ to be placed on the stencil
- Minimize the total shooting number S

Satisfy:

- Any character $c_i \in C_s$ can be located in the boundary of the stencil
- Any two characters c_i and c_j are not overlapped except at the blank area.

$$S = \sum_{c_i \in (C - C_s)} \boxed{t_i \cdot vsb_i} + \sum_{c_i \in C_s} \boxed{t_i \cdot 1}$$

- S - the sum of shots by VSB and CP
- C_s – the set of characters on the stencil
- c_i - the character i
- t_i - the count of c_i appeared in layout
- vsb_i - the shot number of the character ci by VSB(the number of rectangles inside the pattern)

Outline

- Introduction
- Problem Formulation
- **Geometry-Frequency Balanced Algorithm**
- Experimental Results
- Conclusion

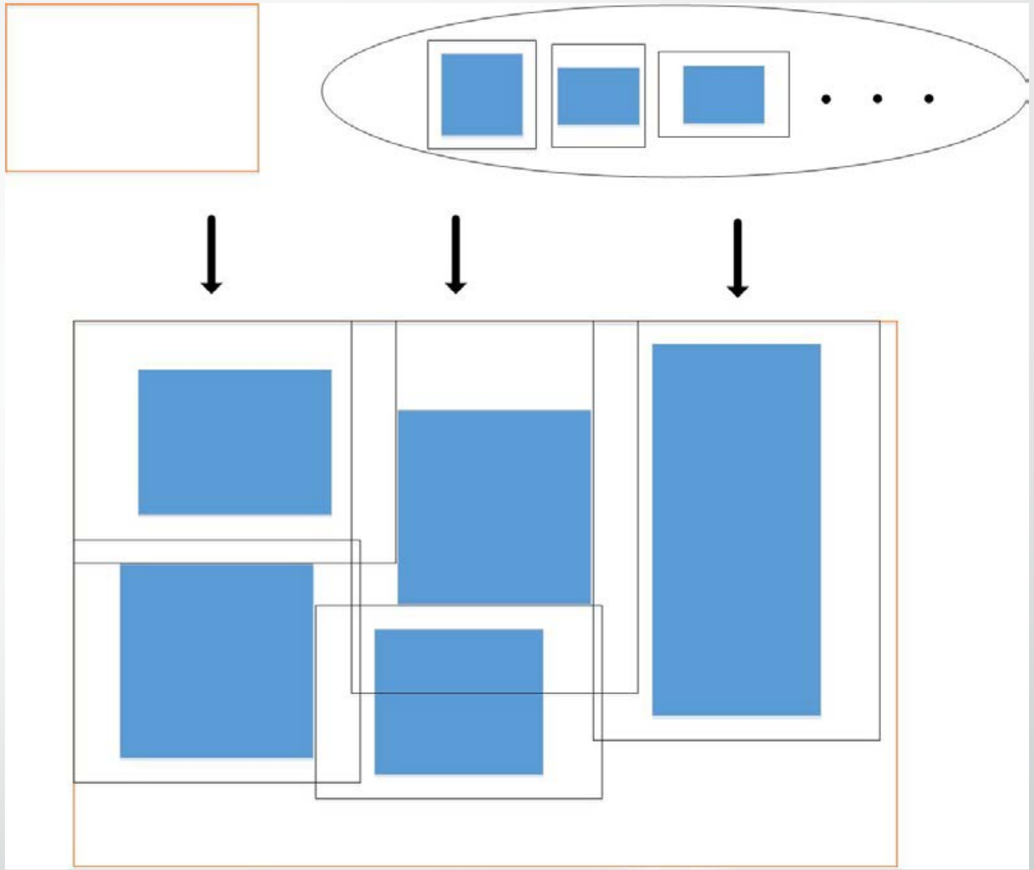
Properties in 2D-OSP

Geometry

- More overlapping
- More characters on stencil
- Better design

Frequency

- Complex pattern inside
- More reduction of shots
- Better design



$$S = \sum_{c_i \in (C - C_s)} t_i \cdot vsb_i + \sum_{c_i \in C_s} t_i \cdot 1 = \underbrace{\sum_{c_i \in C} t_i \cdot vsb_i}_{\text{constant}} - \underbrace{\sum_{c_i \in C_s} t_i \cdot (vsb_i - 1)}_{\text{the reduction of shot number}}$$

Previous Works

[3] B. Yu, et al. "E-blow: E-beam lithography overlapping aware stencil planning for mcc system", DAC, 2013.

Sequence Pair(SP) model and simulated annealing framework

[4] J. Kuang and E.F.Y. Young. "Overlapping-aware throughput-driven stencil planning for e-beam lithography", ICCAD, 2014.

A novel optimization flow

Both of them put emphasis on the **Geometry** to improve the number of characters on the stencil, its effectiveness of saving shot number is limited.

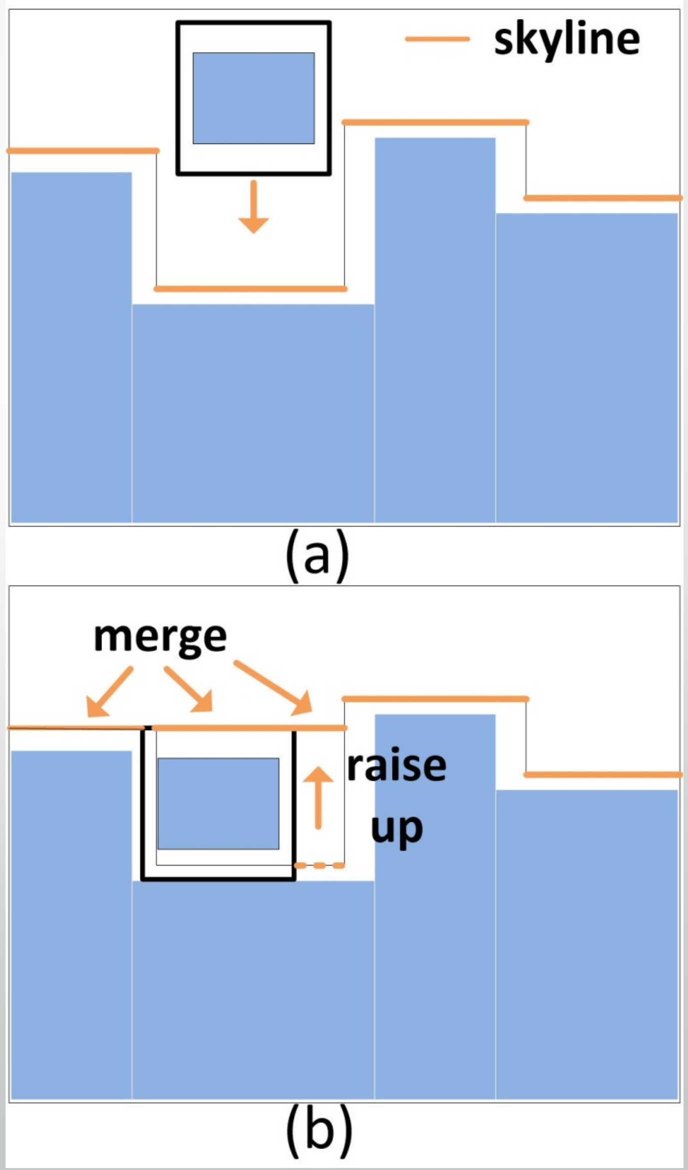
The work in [4]

For geometry property

- Pre-packing for the similar characters
- Bin-packing algorithm

Skyline:

- Top bound of the characters
- Parallel to the horizontal axis



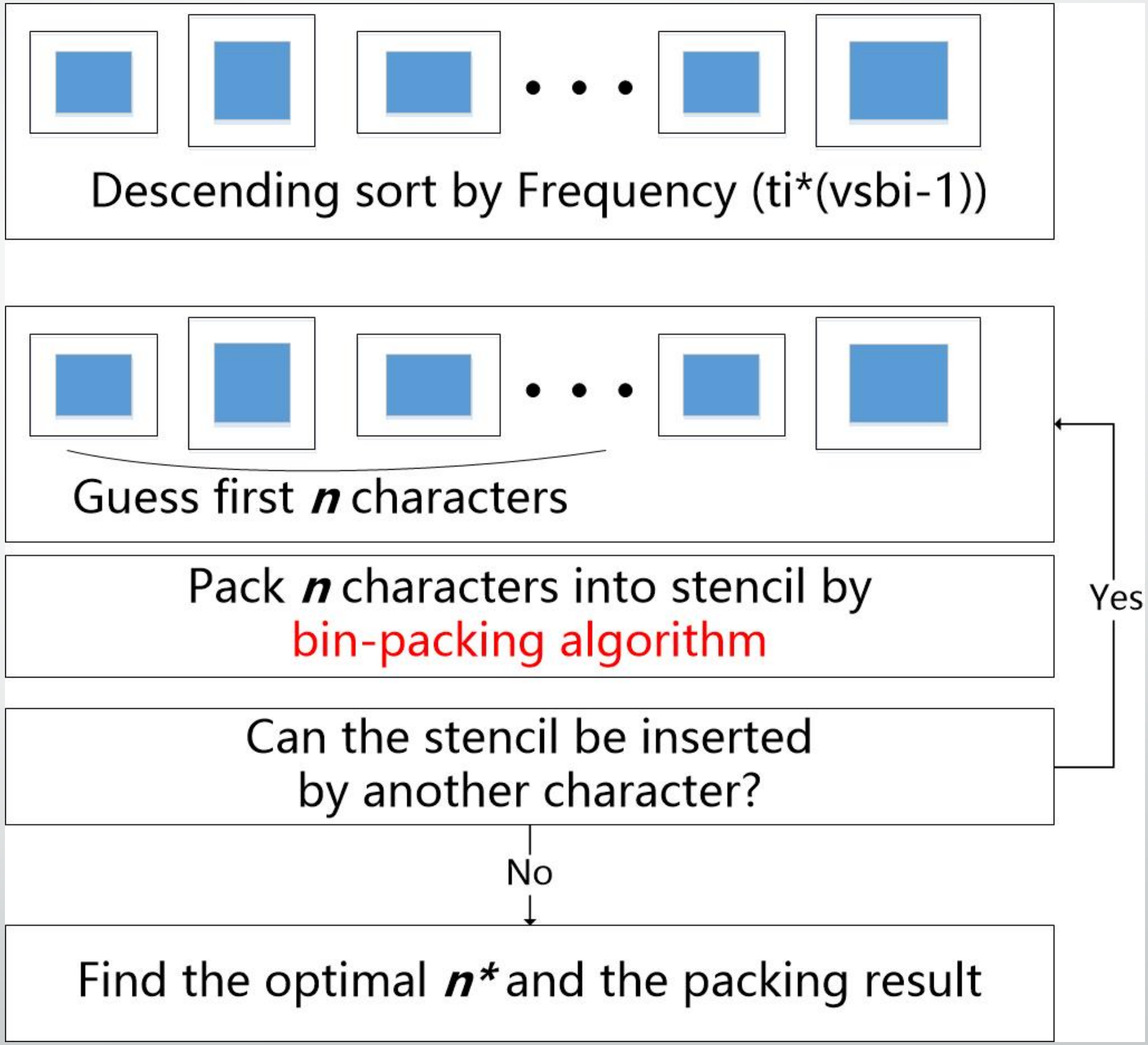
(a)

(b)

The work in [4]

For frequency property

- An outside loop to change the number of candidate characters

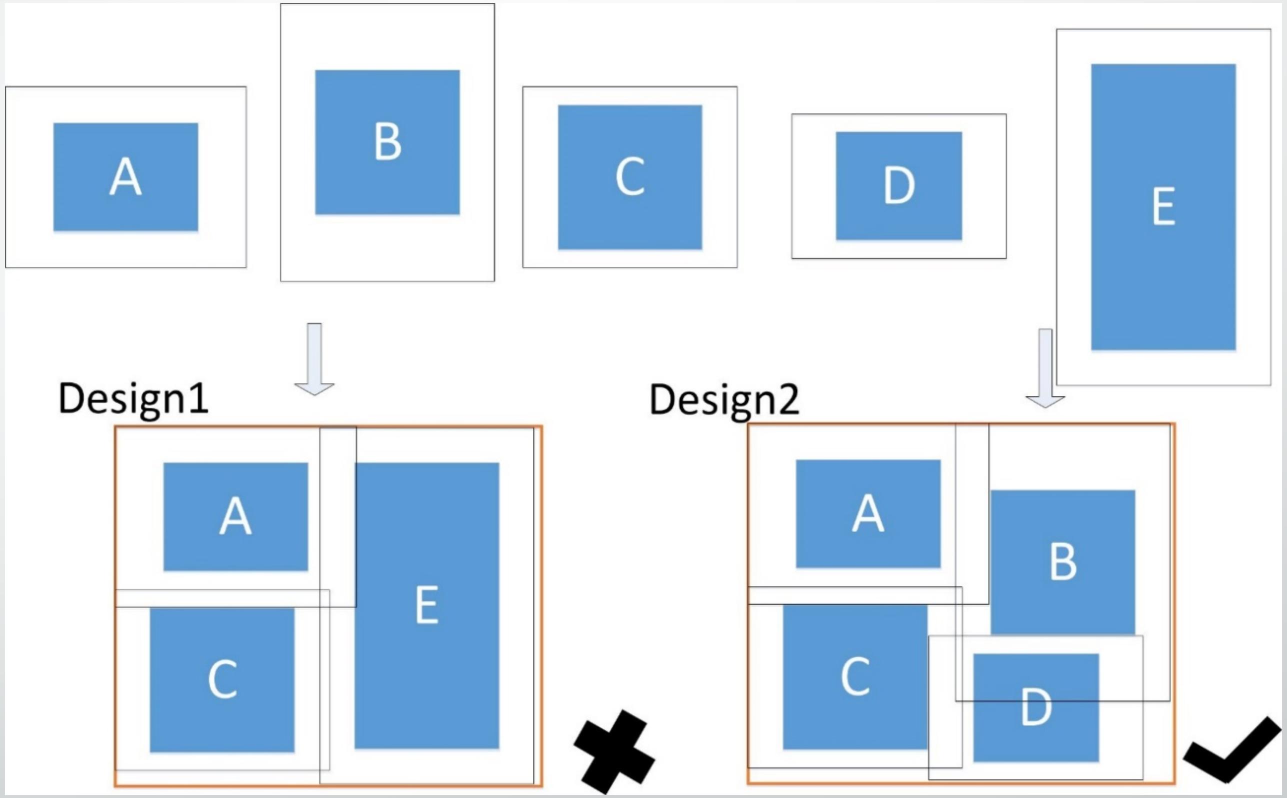


An Example in [4]

Design 2 is better than 1 as more characters is selected.

Character A, B, C, D and E are selected by the outside loop.

By bin-packing, the solution by [4] is design 2.



The solution by [4]

An Example in [4]

Design 1 is better than 2 as shot number of 1 is less than 2 by the given t_i and vsb_i. Design 2 is the more reasonable solution.

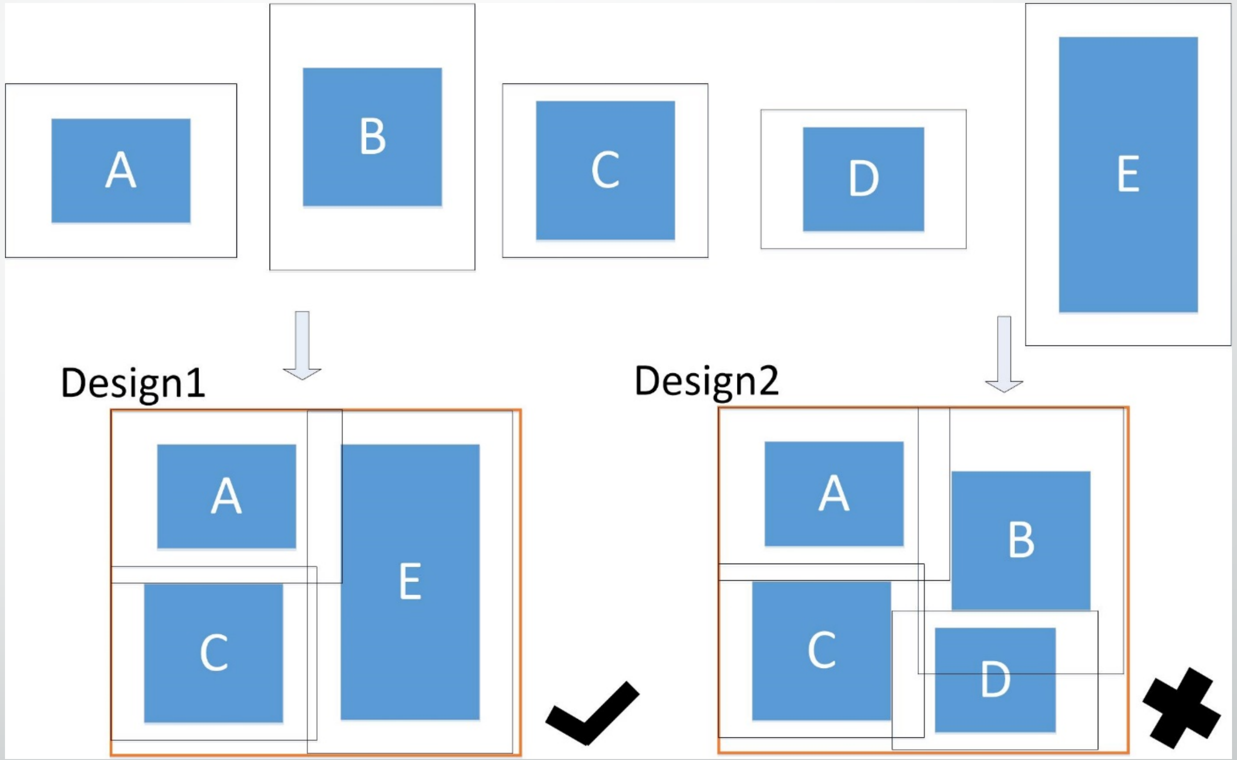
	t_i	vsb_i
A	1	2
B	1	3
C	1	2
D	1	4
E	1	7

S1=10

S2=11

S1<S2

$$S = \sum_{c_i \in (C - C_s)} t_i \cdot vsb_i + \sum_{c_i \in C_s} t_i \cdot 1$$



The more reasonable solution

Motivation

Defects in [4]

- A loop to combine the character properties of geometry and frequency may cause unreasonable solutions.
- Pre-packing characters is suspectable.

How to do better on 2-D OSP?

Geometry-Frequency Balanced Algorithm

- Use a new merit f/A to select best character without the outside loop
- Use 2-D bin-packing algorithm to insert each selected character into the stencil without pre-packing

Merit f/A for choosing character

f/A:

- The higher frequency per unit area, the more shots will be saved
- Frequency :

  f – the given frequency of each character

$$f_i = t_i \cdot (vsb_i - 1)$$

- Geometry :

  A - the occupied area of each selected character

  - environmental related and influenced by characters nearby

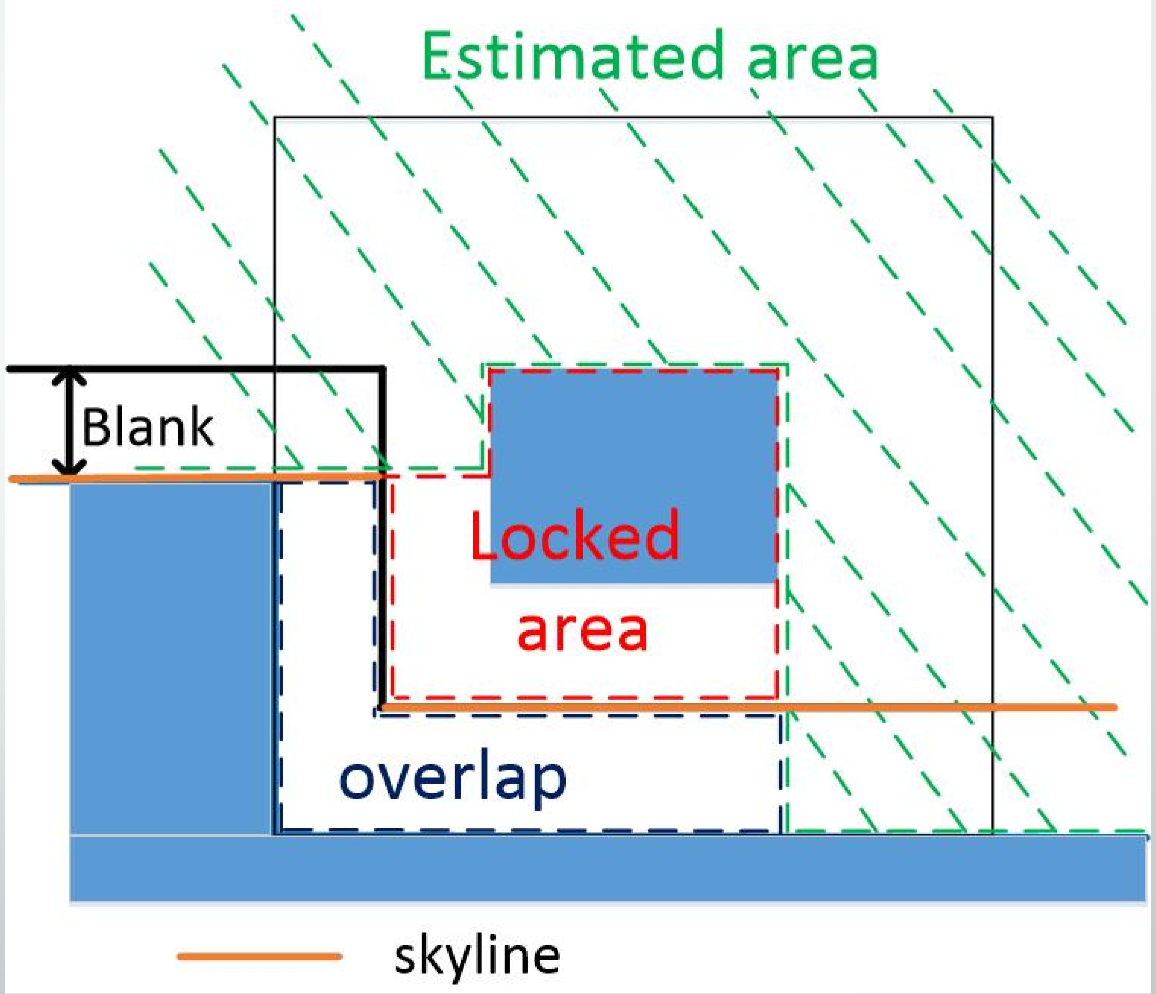
  - estimated accurately by the geometry property of the characters

Merit f/A for choosing character

Occupied area:

- Overlap - A_O
- Locked area - A_L
- Estimated area - A_E

$$A = \frac{A_O}{2} + A_L + A_E$$

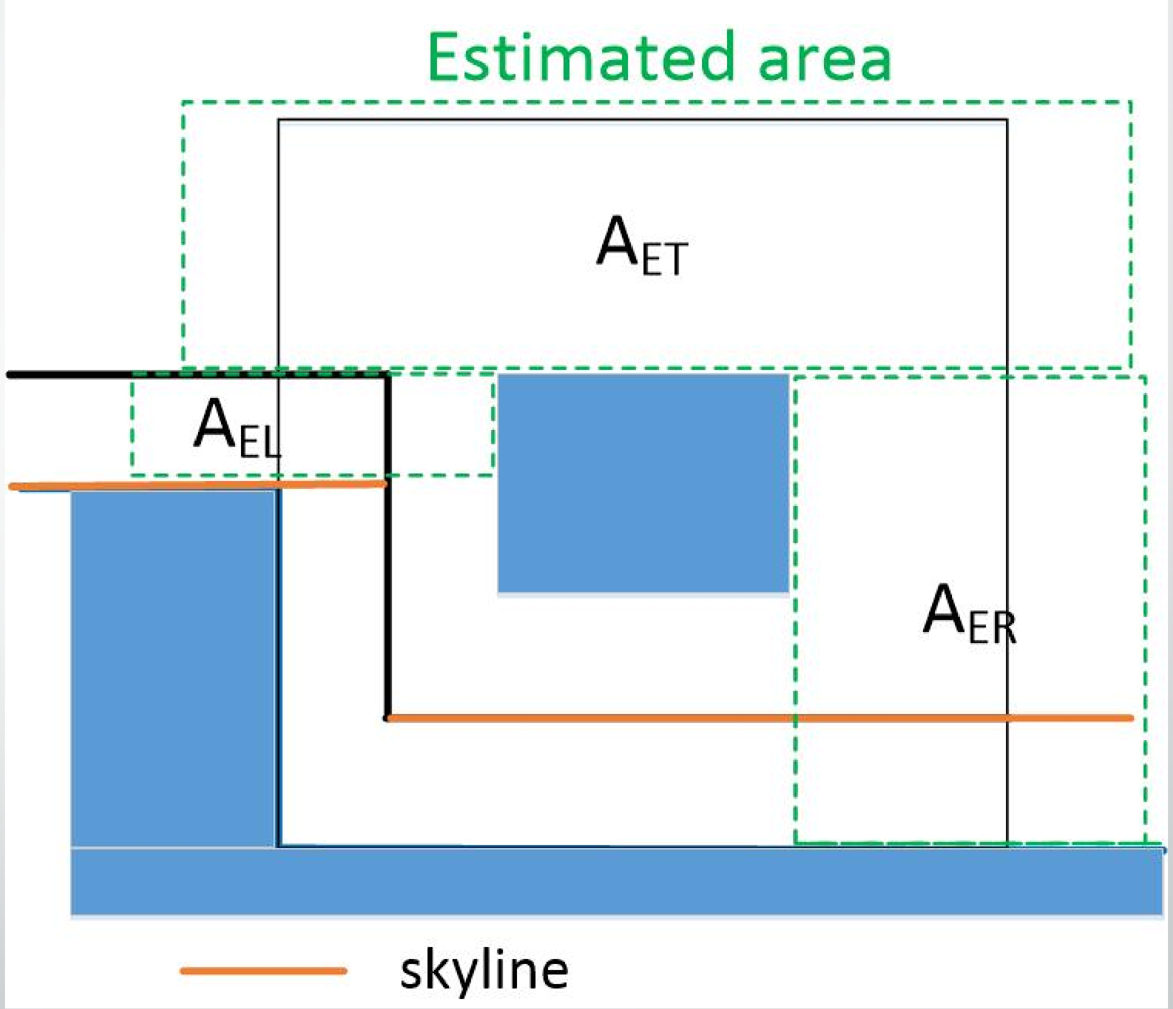


Estimate A_E by skyline

A_E :

- Left area - A_{EL}
- Top area – A_{ET}
- Right area - A_{ER}

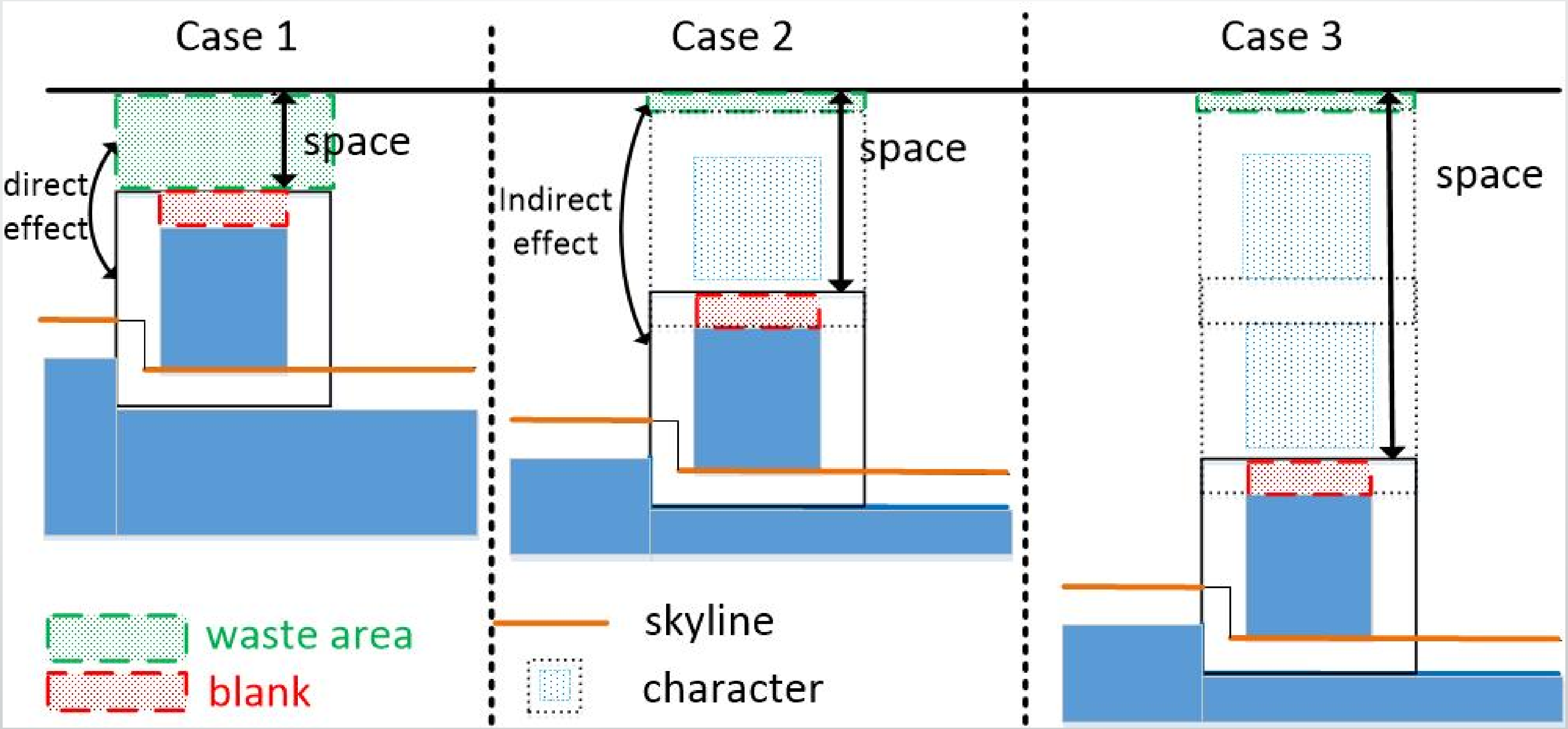
$$A_E = A_{EL} + A_{ET} + A_{ER}$$



Estimate top area A_{ET} by skyline

Top area - A_{ET} :

- Top blank
- Waste area



Top blank + Waste area

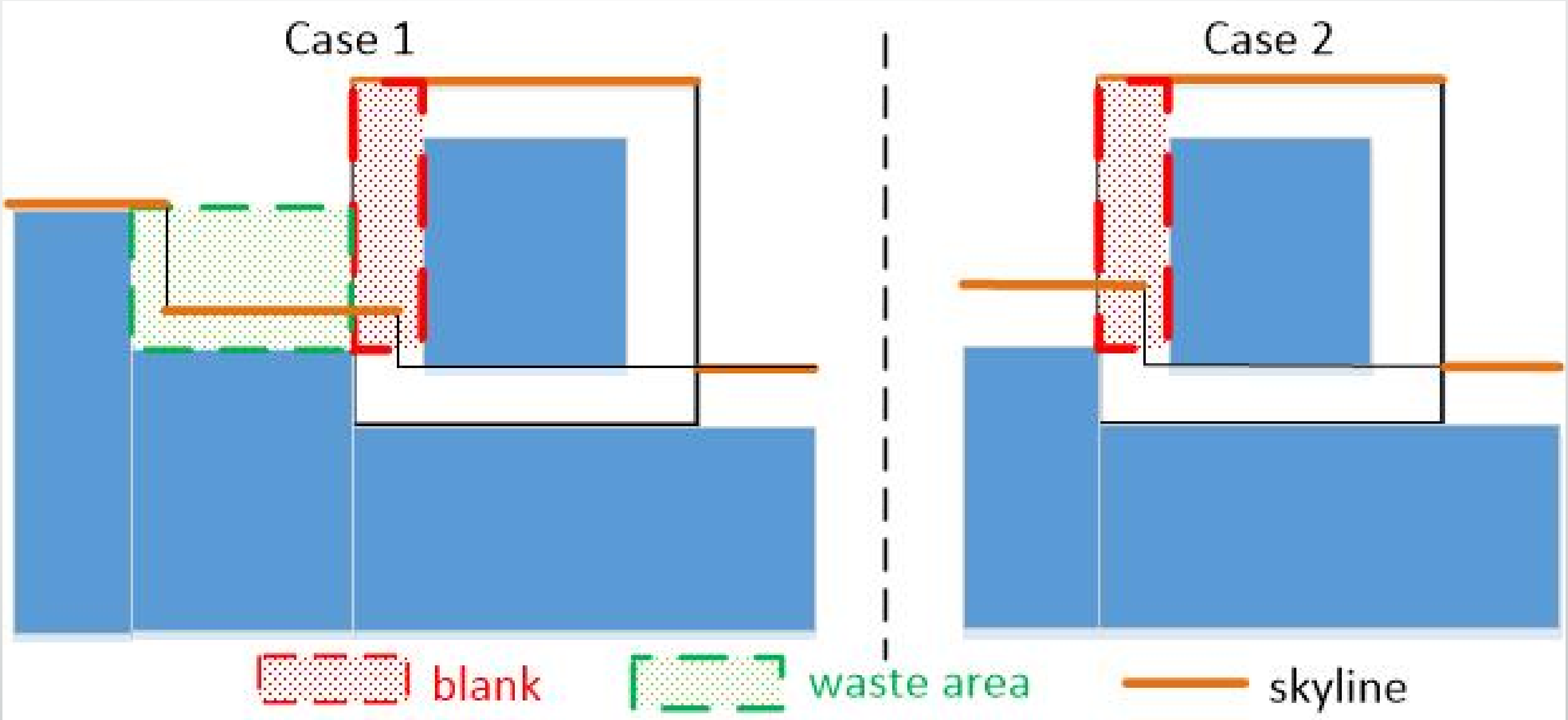
Top blank

Top blank/2

Estimate left area A_{EL} by skyline

Left area - A_{EL} :
- Left blank
- Waste area

Right area - A_{ER} :
- Right blank
- Waste area



Left blank + Waste area

Left blank/2

Merit f/A for choosing character

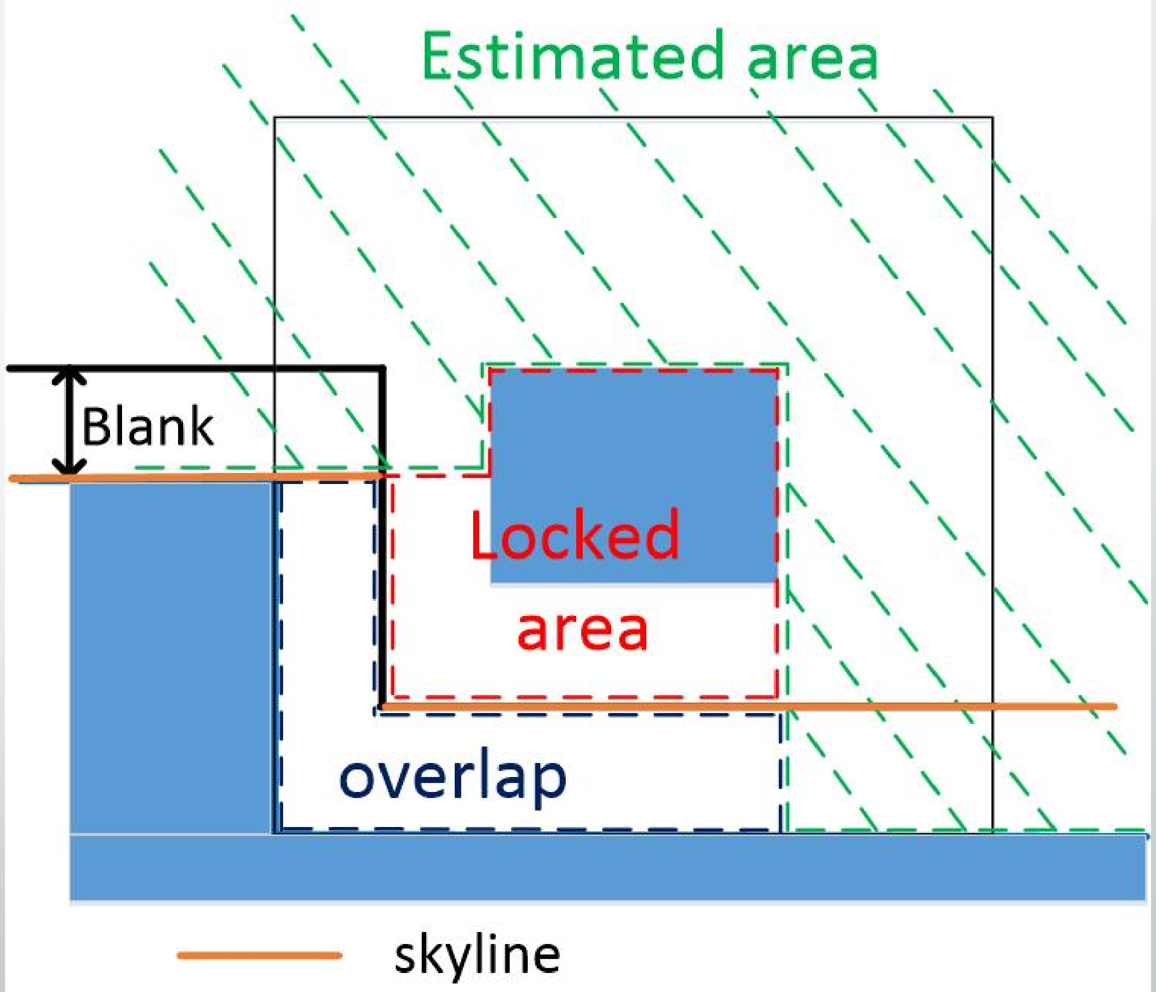
$$f_i = t_i \cdot (vsb_i - 1)$$

Occupied area A:

- Overlap - A_O
- Locked area - A_L
- Estimated area - A_E

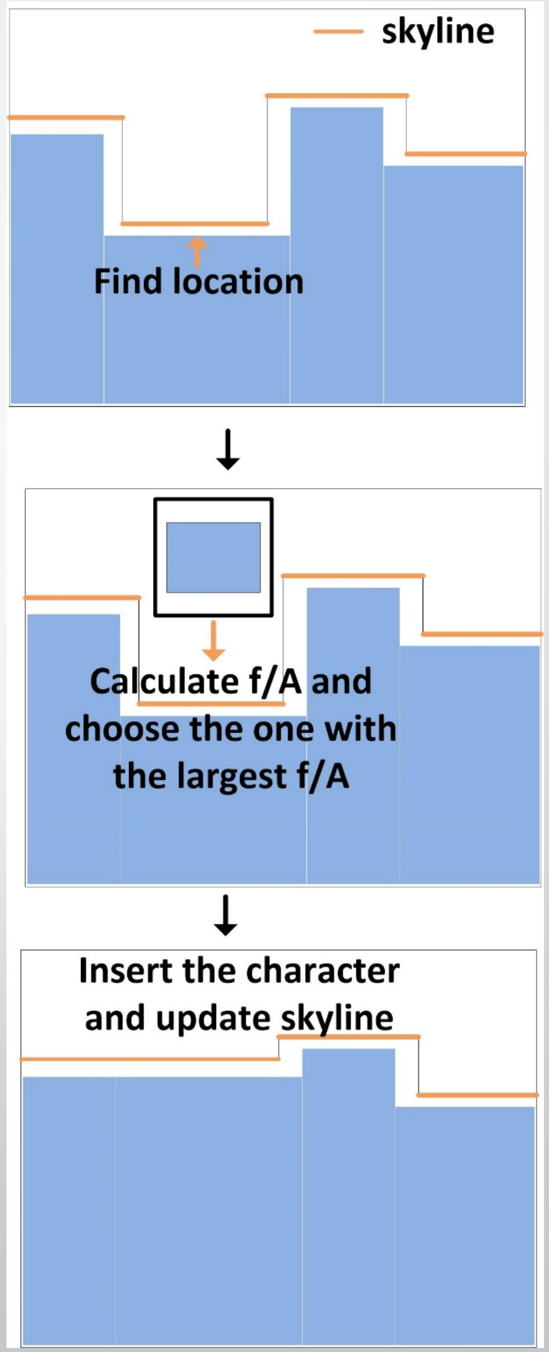
$$A = \frac{A_O}{2} + A_L + A_E$$

$$A_E = A_{ET} + A_{EL} + A_{ER}$$



Flow of our method

1. **Find location:**
   - Choose the lowest skyline
2. **Choose the best character :**
   - The best character is the one with largest f/A
3. **Place the character into the stencil:**
   - Use bin-packing algorithm to insert the character and update the skyline for the next insert
   - Go back to step 1 until the stencil cannot be inserted any other character



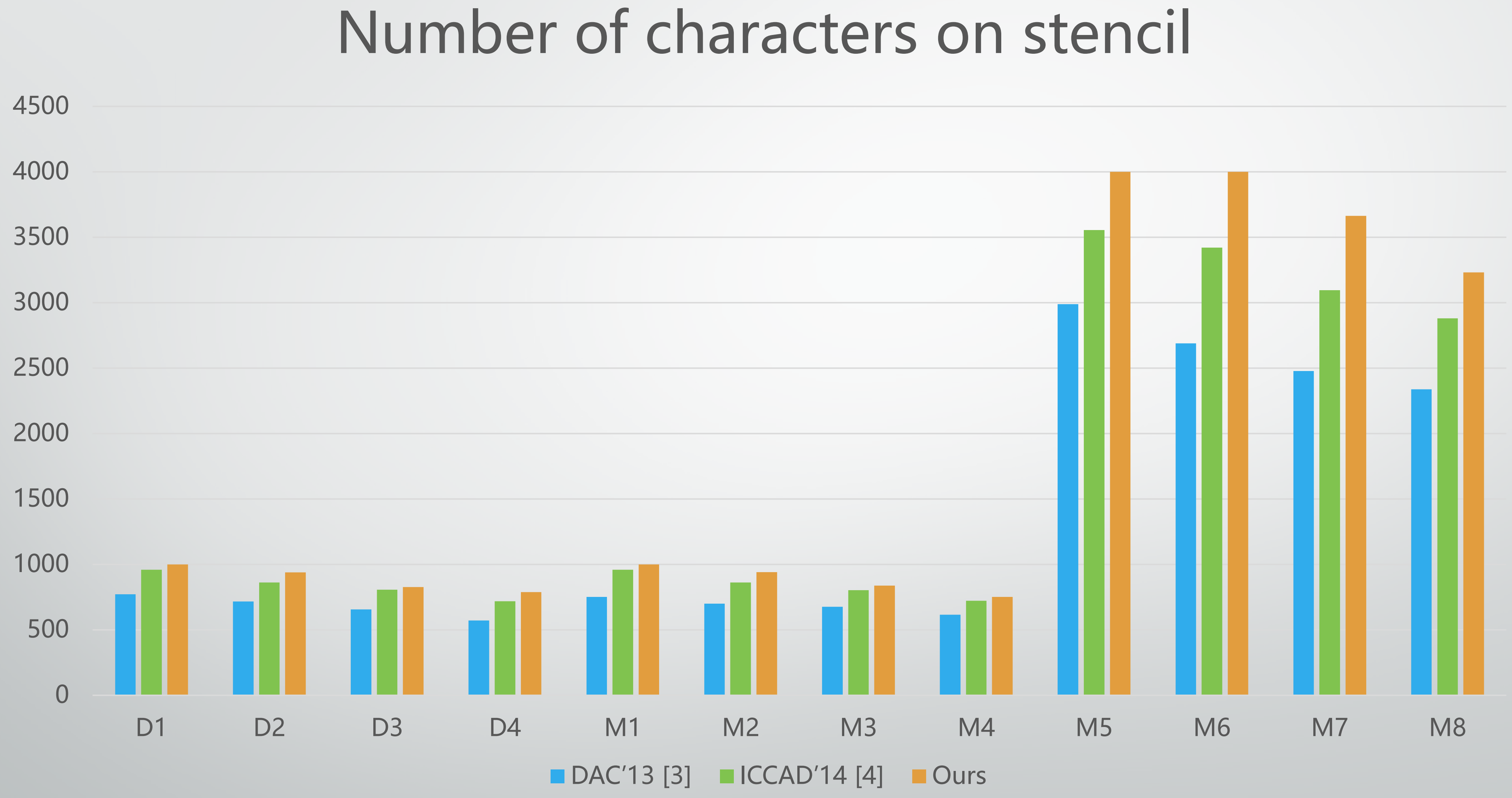
Outline

- Introduction
- Problem Formulation
- Geometry-Frequency Balanced Algorithm
- **Experimental Results**
- Conclusion

Experimental Results

- Implemented in C++
- Performed on a Linux machine with 2.5GHz CPU and 64GB memory
- Benchmarks are from [4]

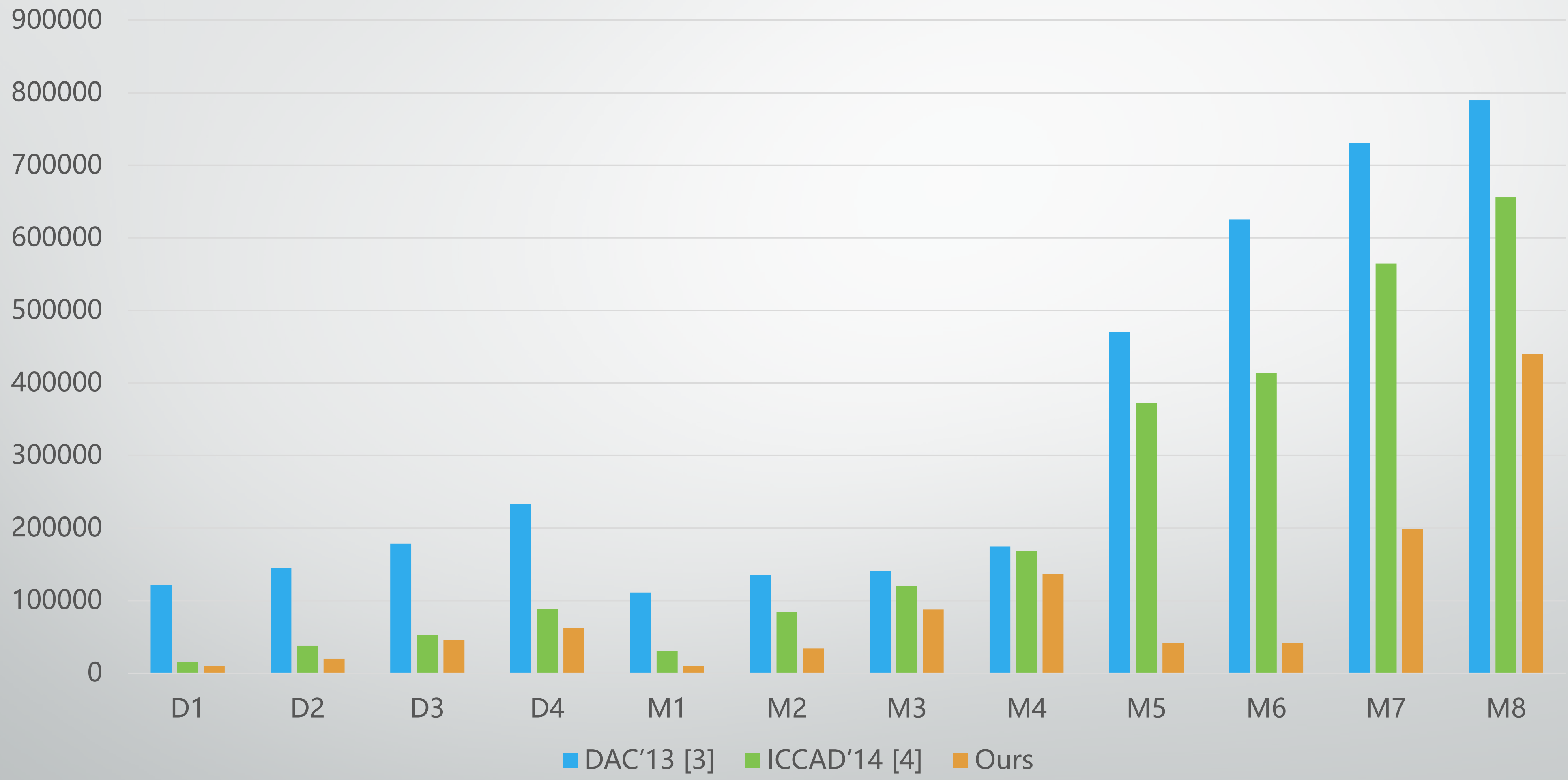
Experimental Results



Our method is 27% better than the solver of reference [3] and 11% better than the solver of reference [4] on average.

Experimental Results

Number of shots



Our method achieves a significant saving over [3] and [4], for their results are 3.41 and 2.30 times that of ours on average.

Conclusion

Simple yet Effective method

- The new merit have a strong connection with the properties of 2-D OSP
- Effective approach to estimate the occupied area of the candidate character
- Achieved better solution for all benchmarks

THX

Q&A