# Floorplanning by A Revised 3-D Corner Block List with sub-$C^+$-tree*

Sheqin Dong Rensheng Wang Fan Guo Jun Yuan Xianlong Hong

Department of Computer Science & Technology, Tsinghua University, Beijing China

## Abstract

In this paper we present a 3-D floorplan representation based on the methodologies of Corner Block List (CBL). Our model is an extension and generalization of the original 2-D CBL and inherits its most crucial advantages including: linear time operations and evaluations, relatively small solution space ($<n!3^{4n-4}$) with complex non-slicing structures, etc. Although like other sequence based representations, the solution space of 3-D CBL is not complete, floorplanning algorithms using this 3-D representation can be highly efficient. Our conclusions are confirmed by experimental results.

**Keywords:** 3D Floorplanning, Corner Block List

## 1. Introduction

In our persistent advances towards faster and smaller VLSI circuits, CMOS technology has scaled into deep submicron regime. Interconnection delay became a dominant factor of the circuits' performances and the size of wires is consuming an increasingly large portion of design budgets. While the techniques in planer wafers are approaching the limits, three-dimensional integrated circuits is a new way to achieve our goals. So design automation tools for 3-D ICs are becoming a prominent demand. Besides, some new-generation reconfigurable FPGAs allow several tasks to share a same physical location at different times and the temporal floorplanning of these tasks is close to a 3-D placement problem. All these demands call for 3-D floorplan representations, which are much more complicated than 2-D ones.

Sequence-Triple and Sequence-Quintuple introduced in [1] are the earliest 3-D floorplan representations. They are based on the successful 2-D representation Sequence-Pair. Recently, T-tree [6] and 3-D subTCG [7] were proposed for temporal floorplanning of FPGAs. Another representation of slicing structure was proposed in [8].

In this paper, we introduce a revised 3-D representation based on the Corner Block List (CBL) proposed in [2]. This new representation has inherited most of the advantages of CBL such as the evaluation of any instance is TIME(n), the size of the solution space has an upper bound $n!3^{4n-4}$, which is relatively low and enables efficient heuristic search algorithms, etc.

Based on these properties of 3-D CBL representation, the simulated annealing algorithm can give high quality solutions with multiple objectives such as volume, wire length, thermal distribution, etc. For instance, our experiments show that the 3-D CBL based SA scheme can obtain a low ratio of dead space.

The rest of this paper is organized as follows. Section 2 introduces some concepts about 3-D mosaic floorplan, which is extended from [2]. Section 3 describes the 3-D Corner Block List representation. Section 4 discusses the properties of the solution space and the transformation algorithms. Section 5 reports the experimental results. Finally, section 6 concludes the paper.

## 2. Preliminaries

### 2.1 3-D Mosaic Floorplan

A floorplan divides a 3-D chip into cuboidal rooms with rectangular slices. Each room is assigned to no more than one module. The floorplan is defined to be "mosaic" if and only if it observes the following properties:

- Floorplan of no empty space. Each room is assigned to one module. The internal slices intersect and form 3-D T-junctions. A T-junction is composed of a non-crossing slice and a crossing slice where the non-crossing slice has one edge touching the crossing slice.
- Topological equivalence on slice sliding. The topology is defined to be equivalent before and after the non-crossing slice of a T-junction slides. The slice positions are to be determined by the lengths, widths and heights of the modules.

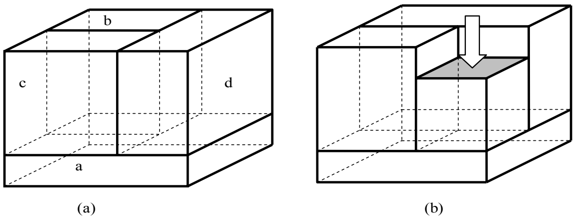


**Figure 1. (a) 3-D mosaic floorplan (b) Corner block insertion**

*This work is supported by NSFC 60473126

### 2.2 Corner Block Insertion

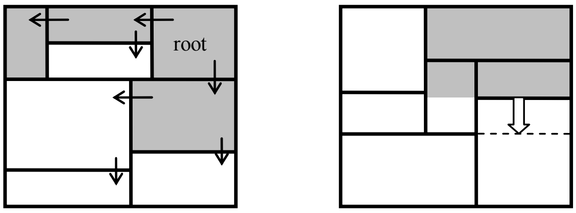
For a room whose three edges each pointing to the higher boundary of its direction, we define it to be a "corner block". A mosaic floorplan has one and unique corner block. The corner block can be deleted while the revised floorplan remains to be mosaic. And inversely, we can insert corner blocks in to a mosaic floorplan and form larger floorplans.

In a 3-D mosaic floorplan, we have a front, a right and a top surface, perpendicular to the x-axis, y-axis and z-axis respectively. Each surface is a 2-D mosaic floorplan. When inserting a new room, we push a rectangular slice at the up-right corner of the surface into the cuboid and form a new 3-D corner block, as Figure 1(b) shows. The figure means that the new added module covers and only covers the rooms under the grey slice at z+ direction.

## 3. 3-D Corner Block List

The key of the 3-D floorplan representation is to represent the rectangular slice at the up-right corner of a 2-D mosaic floorplan.

In [3], a Twin Binary Tree representation for 2-D mosaic floorplans was introduced. In this representation, a binary tree is constructed by connecting each room to its $C^+$-neighbor (the adjacent room by the non-crossing segment of the up-right T-junction) except for the corner block. The root of the tree (called $C^+$-tree) is at the up-right corner block of the floorplan. So a sub-tree rooted at the corner block is likely to be able to represent the covered area for the 3-D corner block insertion. But not all the sub-trees can be used as rectangular covering.



(a) (b)

**Figure 3. An impossible rectangular covering (a) and a possible one (b)**

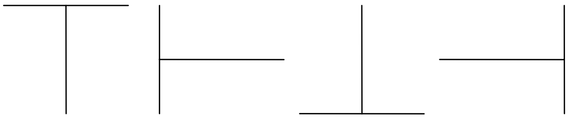
Figure 3(a) shows a sub-tree where it is impossible to cover the four grey rooms by a rectangle while not covering any other rooms. Another case in Figure 3(b) allows the sub-tree to be covered by a rectangle. We can slide a segment downwards and form a new corner block by pushing the grey part into the 3-D floorplan.

In order to ensure feasible corner block insertions, we propose a revised sub-$C^+$-tree, represented by a sequence of "0", "1" and "2" called sequence T.

**Definition 1: Sub-$C^+$-tree by sequence T.** The sequence of covered rooms starts from the corner block, which is always to be covered. The following rooms are added according to the digits in sequence T: At "0" of T, we move to the left branch, i.e. the adjacent room by the up-left 0° T-junction (see Figure 4). If the up-left T-junction is 90°, then continue moving up until a 0° one is met (there is at least one 0° T-junction at the top boundary), and if all the rooms passed by have been covered, we can move to the room at the left side of the 0° T-junction. And at "1" of sequence T, we move to the right branch, symmetric to the process of moving to the left branch. Lastly, "2" means end of the sequence.

**Proposition 1**: For any sub-$C^+$-tree defined above, there exists a rectangular slice which covers the rooms in the sequence of the sub-$C^+$-tree and does not cover any room not in the sequence.

*Proof*: omit here.



(a) 0° T-junction (b) 90° T-junction (c) 180° T-junction (d) 270° T-junction

**Figure 4. Four types of T-junctions**

We now have feasible corner block insertions by a representation of sub-$C^+$-tree. To make this sub-$C^+$-tree reasonable, we need to prove its strength in representing rectangular coverings.

**Proposition 2**: When false modules [5] can be added into a 2-D mosaic floorplan, any rectangular covering from the up-right corner can be represented by sub-$C^+$-tree.

Based on this representation of covered areas on surfaces, we propose a 3-D representation of mosaic floorplans constructed by corner block insertions.

**Definition 2: Corner block list.** The tuple (S, L, T) is called a corner block list, with sequence S of module names, L of the orientations of corner block insertions, and T of the sub-$C^+$-trees indicating the covered area of each corner block insertion.

## 4. The Solution Space

### 4.1 Size of the Solution Space

Assume there are n modules in a 3-D mosaic floorplan. The tuple (S, L, T) decides the configuration of the floorplan. Sequence S has length n-1. And for sequence T, the total length of T is at most (3n-3). Since the digits in sequence L and T have three values, the upper bound of the solution space of 3-D corner block list is $n!3^{4n-4}$.

### 4.2 From Placement to 3-D CBL

Given a packing placement of modules, we try to find a 3-D corner block list which represents it as a mosaic floorplan.

The first problem is whether such a corner block list exists. Though the strength of the representation of corner block insertion is shown in Proposition 2, the solution space of 3-D CBL is still not complete. The placement in Figure 5 cannot be represented by corner block list even when false modules can be added. We have a counter example of Figure 6 which is common for other sequence based representations except [1].

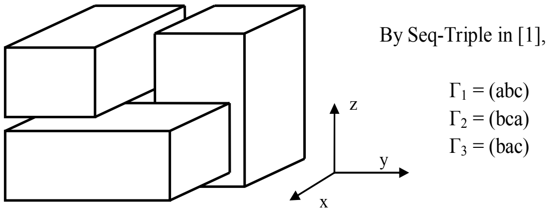


**Figure 5. A placement with a "constraint loop"**

Being conscious of the incompleteness of the solution space, we still can transform a packing placement with NO constraint loops into a 3-D corner block list. From the conclusions in Section 3 and Proposition 2, we can construct a 3-D mosaic floorplan by inserting corner blocks which contain modules, adding empty rooms (false modules [5]) when necessary.

**Proposition 3**: Given a 3-D packing placement without constraint loops, it can be represented by a corner block list tuple by adding no more than 2n empty rooms.

So we have a linear upper bound of the number of empty rooms needed by a placement without "constraint loops". In this sense, our 3-D CBL is stronger than Seq-Triple in [1] which cannot represent placement containing double-constraint.

Though proposition 3 does not provide the exact method to insert empty rooms, we have such an algorithm which performs in time O(n$^2$). The trivial algorithm details are omitted here.

### 4.3 From 3-D CBL to Placement

The algorithm of transforming a corner block list to a 3-D placement is critical for floorplan optimization. We maintain three surfaces of the 3-D floorplan, each facing +x, +y, or +z and each is a 2-D corner block list. Each room of these 2-D CBLs has its position.

Assume the corner block is inserted from the surface facing +z. Obviously, the z-value of the new module is no less than all the covered rooms. And among the covered rooms, we pick out a row of "left side rooms" and a row of "bottom side rooms". A left side room is covered by the sub-C$^+$-tree, and if we move to its left branch as in definition 1, the next room is NOT covered or there is no next room. So the left side rooms are adjacent to the new corner block at the left edge of the pushed slice. And the x-value of the new module is no less than all the left side rooms. Symmetrically, the "bottom side rooms" are defined as the adjacent rooms of the new room at the bottom edge and the y-value of the module is no less than all the bottom side rooms.

As Figure 6 shows, there are four cases of T-junctions between the two adjacent left side rooms. The up-left T-junction of the lower room and the bottom-left T-junction of the upper room are respectively (a) 0° and 180°, (b) 0° and 90°, (c) 90° and 90°, (d) 90° and 180°. In all these four cases, the left boundary can be aligned so that the new corner block can be inserted as a cuboidal room. In case (a), the positions of the two left side rooms should be checked by their y-values. Because like 10 and 18 in Figure 5, the up-down relation may be not satisfied by the modules which are already placed in the floorplan, and adjustment of y-values is necessary for avoiding module overlapping. And in case (b), we extend the inside room to maintain the 2-D mosaic floorplan of the surface. The room is already covered under +y surface, but it may appear at the top boundary of +z surface again so the new inserted corner block must have y, z-values no less than the inside room in order to replace it on the mosaic floorplan of +y surface.

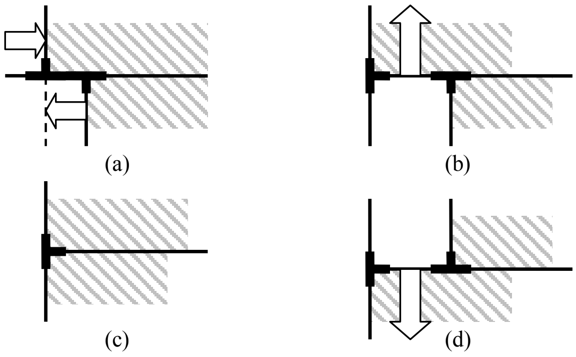


**Figure 6. Four cases of two adjacent left side rooms.**

Besides, in case (b) and (d), the x-value of the extended room should also be checked.

For the bottom side rooms, we can align the lower boundary symmetrically. Thus a corner block insertion is completed, and so is the placement of a module.

**Proposition 4**: The algorithm of transforming a 3-D corner block list to its placement is in TIME(n).

## 5. Experimental Results

Based on simulated annealing, we implement the 3-D Corner Block List representation in our 3-D floorplanning algorithm. We perturb a corner block list tuple into another one by following operations.

- Exchange two modules in sequence S.
- Choose a digit in L and modify it (0/1/2).
- Choose a digit in T and modify it (0/1/2). We have redundancy in our solution space because there may be too much "0"/"1"s for the sub-$C^+$-tree on a surface.
- Rotate a module to one of the 24 directions.
- For floorplan problems with soft modules, there is another operation which chooses a module and give it an alternative size.

The algorithm is implemented in C++ language on a Pentium4 CPU (2.82GHz) with 1GB memory. Both 3-D CBL and 3-D ECBL(3) are tested on the benchmark of Beasley where the values of pieces are used as the third dimension. ECBL(3) [5] means 2n empty rooms are added into the mosaic floorplan. The size of its solution space is therefore $C(3n,n)n!3^{12(n-1)}$. The results shown in Table 1 prove that our approach is very effective for cost optimizations. Besides, Figure 7 shows some results of packing placement produced by the program.

**Table 1 Results of volume optimization using 3-D benchmarks**

| Test case | Number of Modules | Sum of Volume | 3-D CBL | | | 3-D ECBL(3) | | |
|---|---|---|---|---|---|---|---|---|
| | | | Volume | Usage | Time (sec) | Volume | Usage | Time (sec) |
| Beasley1 | 10 | 6218 | 8772 | 70.8846% | 4 | 7820 | 79.5141% | 8 |
| Beasley2 | 17 | 11497 | 13920 | 82.5934% | 6 | 13910 | 82.6528% | 11 |
| Beasley3 | 21 | 10362 | 13608 | 76.1464% | 7 | 13430 | 77.1556% | 14 |
| Beasley4 | 7 | 7365 | 10125 | 72.7407% | 4 | 9568 | 76.9753% | 6 |
| Beasley5 | 14 | 16734 | 20520 | 81.5497% | 5 | 19140 | 87.4295% | 10 |
| Beasley6 | 15 | 11040 | 15554 | 70.9785% | 5 | 14520 | 76.0331% | 11 |
| Beasley7 | 8 | 17168 | 20640 | 83.1783% | 4 | 20574 | 83.4451% | 7 |
| Beasley8 | 13 | 86404 | 111600 | 77.4229% | 5 | 101520 | 85.1103% | 9 |
| Beasley9 | 18 | 138928 | 193450 | 71.8152% | 6 | 179928 | 77.2131% | 12 |
| Beasley10 | 13 | 493746 | 596970 | 82.7087% | 5 | 596970 | 82.7087% | 9 |
| Beasley11 | 15 | 383391 | 494914 | 72.7234% | 5 | 473110 | 81.0363% | 10 |
| Beasley12 | 22 | 646158 | 911232 | 70.9104% | 7 | 843360 | 76.6171% | 15 |
| Okp1 | 50 | 124358256 | 190175424 | 65.3913% | 11 | 185468640 | 67.0508% | 33 |
| Okp2 | 30 | 85445223 | 123675000 | 69.0885% | 8 | 117855000 | 72.5003% | 20 |
| Okp3 | 30 | 123808466 | 175537000 | 70.5313% | 8 | 169089024 | 73.2209% | 20 |
| Okp4 | 61 | 238860881 | 336526344 | 70.9784% | 13 | 328069700 | 72.8080% | 40 |
| Okp5 | 97 | 189874755 | 293910000 | 64.6030% | 22 | 279360000 | 67.9678% | 68 |

## 6. Conclusions

We introduce a class of 3-D mosaic floorplans and one of its representations as 3-D Corner Block List. The four propositions in our paper reveal the strong expressivity and high efficiency of 3-D CBL. And together with the supportive experimental results, we conclude that this 3-D representation is suitable for 3-D IC design and other potential applications.

## 7. References

[1] H. Yamazaki, K. Sakanushi, S. Nakatake, Y. Kajitani. "The 3D-Packing by Meta Data Structure and Packing Heuristics", *IEICE Trans. Fundamentals*, Vol. E83-A, No.4, pp. 639-645, April 2000

[2] X. Hong, G. Huang, S. Dong, Y. Ma, Y. Cai, C.-K. Cheng. "Corner Block List: An Efficient Topological Representation of Non-Slicing Floorplan", *Proc. ICCAD*, pp. 8–12, 2000

[3] B. Yao, H, Chen, C.-K. Cheng, R. Graham. "Revisiting Floorplan Representations", *Proc. ISPD*, pp. 138-143, 2001

[4] X. Zhang, Y. Kajitani. "Space-Planning: Placement of Modules with Controlled Empty Area by Single-Sequence", *Proc. ASP-DAC*, pp. 25-30, 2004

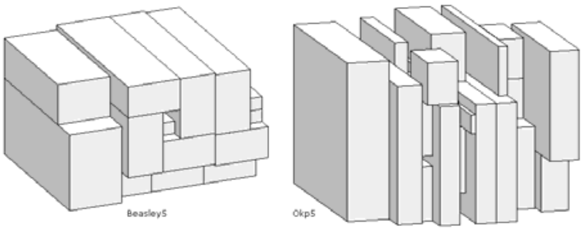
[5] S. Zhou, S. Dong, X. Hong, Y. Cai, C.-K. Cheng. "ECBL: An Extended Corner Block List with Solution Space Including Optimum Placement", *Proc. ISPD*, pp. 156-161, 2001

[6] P.-H. Yuh, C.-L. Yang, Y.-W. Chang. "Temporal Floorplanning Using the T-tree Formulation", *Proc. ICCAD*, pp. 300-305, 2004

[7] P.-H. Yuh, C.-L. Yang, Y.-W. Chang, H.-L. Chang. "Temporal Floorplanning using 3D-subTCG", *Proc. ASP-DAC*, pp. 725-730, 2004

[8] L. Cheng, L. Deng, D. F. Wong. "Floorplanning for 3-D VLSI Design", *Proc. ASP-DAC*, pp. 405-411, 2005

[9] R. Wang, S. Dong, X. Hong. "An Improved P-admissible Floorplan Representation Based on Corner Block List", *Proc. ASP-DAC*, pp. 1115-1118, 2005



(a) Beasley5 (b) Okp5

**Figure 7. Packing Placements**