Three-dimensional Floorplan Representations by Using Corner Links and Partial Order

ILGWEON KANG, University of California at San Diego, USA
FANG QIAO, Microsoft, USA
DONGWON PARK and DANIEL KANE, University of California at San Diego, USA
EVANGELINE FUNG YU YOUNG, The Chinese University of Hong Kong, China
CHUNG-KUAN CHENG and RONALD GRAHAM, University of California at San Diego, USA

Three-dimensional integrated circuit (3D IC) technology offers a potential breakthrough to enable a paradigm-shift strategy, called "more than Moore," with novel features and advantages over the conventional 2D process technology. By having three-dimensional interconnections, 3D IC provides substantial wirelength reduction and a massive amount of bandwidth, which gives significant performance improvement to overcome many of the nontrivial challenges in semiconductor industry. Moreover, 3D integration technology enables to stack disparate technologies with various functionalities into a single system-in-package (SiP), introducing "true 3D IC" design.

As the first physical design (PD) step, IC floorplanning takes a crucial role to determine IC's overall design qualities such as footprint area, timing closure, power distribution, thermal management, and so on. However, lack of efficient 3D floorplanning algorithms that practically implement advantages of 3D integration technology is a critical bottleneck for PD automation of 3D IC design and implementation. 3D floorplanning (or packing, block partitioning) is a well-known NP-hard problem, and most of 3D floorplanning algorithms rely on heuristics and iterative improvements. Thus, developing complete and efficient 3D floorplan representations is important, since floorplan representation provides the foundation of data structure to search the solution space for 3D IC floorplanning. A well-defined floorplan representation provides a well-organized and cost-effective methodology to design high-performance 3D IC.

We propose a new 3D IC floorplan representation methodology using *corner links* and *partial order*. Given a fixed number of cuboidal blocks and their volume, algorithmic 3D floorplan representations describe topological structure and physical positions/orientations of each block relative to the origin in the 3D floorplan space. In this article, (1) we introduce our novel 3D floorplan representation, called *corner links representation*, (2) we analyze the equivalence relation between the corner links representation and its corresponding *partial order representation*, and (3) we discuss several key properties of the corner links representation and partial order representation. The corner links representation provides a complete and efficient structure to assemble the original 3D mosaic floorplan. Also, the corner links representation for the non-degenerate 3D mosaic floorplan can be equivalently expressed by the *four trees representation*. The partial order representation defines the topological structure of the 3D floorplan with three transitive closure graphs (TCG) for each

Authors' addresses: I. Kang, University of California at San Diego, La Jolla, CA, 92093, USA; email: igkang@ucsd.edu; F. Qiao, Microsoft, Redmond, WA, 98052, USA; email: fmarkqiao@gmail.com; D. Park, University of California at San Diego, La Jolla, CA, 92093, USA; email: dwp003@ucsd.edu; D. Kane, University of California at San Diego, La Jolla, CA, 92093, USA; email: dakane@ucsd.edu; E. F. Y. Young, The Chinese University of Hong Kong, Shatin, N.T., China; email: fyyoung@cse.cuhk.edu.hk; C.-K. Cheng, University of California at San Diego, La Jolla, CA, 92093, USA; email: ckcheng@ucsd.edu; R. Graham, University of California at San Diego, La Jolla, CA, 92093, USA; email: graham@ucsd.edu.

1084-4309/2018/12-ART13 $15.00
https://doi.org/10.1145/3289179

direction and captures all stitching planes in the 3D floorplan in the order of their respective directions. We demonstrate that the corner links representation can be reduced to its corresponding partial order representation, indicating that the corner links representation shares well-defined and -studied features/properties of 3D TCG-based floorplan representation. If the partial order representation describes relations between any pairs of blocks in the 3D floorplan, then the floorplan is a valid floorplan. We show that the partial order representation can restore the absolute coordinates of all blocks in the 3D mosaic floorplan by using the given physical dimensions of blocks.

CCS Concepts: • **Hardware → 3D integrated circuits**; **Partitioning and floorplanning**;

Additional Key Words and Phrases: 3D integrated circuit, 3D IC design, 3D floorplan, floorplan representation

ACM Reference format:
Ilgweon Kang, Fang Qiao, Dongwon Park, Daniel Kane, Evangeline Fung Yu Young, Chung-Kuan Cheng, and Ronald Graham. 2018. Three-dimensional Floorplan Representations by Using Corner Links and Partial Order. *ACM Trans. Des. Autom. Electron. Syst.* 24, 1, Article 13 (December 2018), 33 pages.
https://doi.org/10.1145/3289179

1 INTRODUCTION

Three-dimensional (3D) integration circuit (IC) technology offers a potential breakthrough with novel features and advantages over the conventional 2D integration technology. A 3D IC (as shown in Figure 1) is an integrated circuit manufactured by vertically stacking wafers, dies, or devices, and interconnecting them by using inter-layer vias and/or microbumps such as through-silicon vias (TSVs), monolithic inter-tier vias (MIVs), and face-to-face (F2F) vias. Design's hierarchical level and vertical interconnection's type (as shown in Figure 1) classify 3D IC integration technologies into interposer-based 2.5D packages, TSV-based stacked 3D ICs, monolithic 3D ICs and 3D devices, 3D system-on-chip (SoC), system-in-package (SiP), and so on.

The 3D IC technology provides strong momentum to overcome many of nontrivial challenges that arise from the semiconductor industry's relentless push into the deep nanoscale regime [27]. Recently, a feature editor of *Nature* announced the paradigm shift from shrinking the devices to a new strategy called "more than Moore" [39]. The author [39] declares that the 3D IC technology is a potential option to extend the trend. With the three-dimensional interconnections, substantial wirelength reduction and massive amounts of bandwidth can be achieved between device layers without incurring the usual latency penalties. Also, the 3D IC technology introduces new architectures that can have much better performance for the future applications. For example, the 3D IC technology enables us to stack and integrate disparate technologies into a single system-in-package (SiP) implementation. Fabrication technologies to specific functions such as RF circuits, memories, or optoelectronic devices are often incompatible with the normal IC integration processes for high-performance logic devices. The 3D IC technology suggests a flexible and promising way to include device/gate layers with distinct functionalities (as shown in Figure 1), toward "true 3D IC."

A number of studies have validated the benefits of 3D IC integration technology and have explored the design space for 3D architectures and physical implementations, which maximize the advantages of 3D ICs. Bernstein et al. [2] confirm the benefits of 3D integration in the scope of architecture and performance. In [43], the authors demonstrate that a multicore processor chip design could reduce the number of interconnections between critical intercore components by an order of magnitude through 3D IC implementation. Hybrid memory cube (HMC) consortium [21] is another example of efforts from industrial leaders to drive 3D ICs into mainstream production.

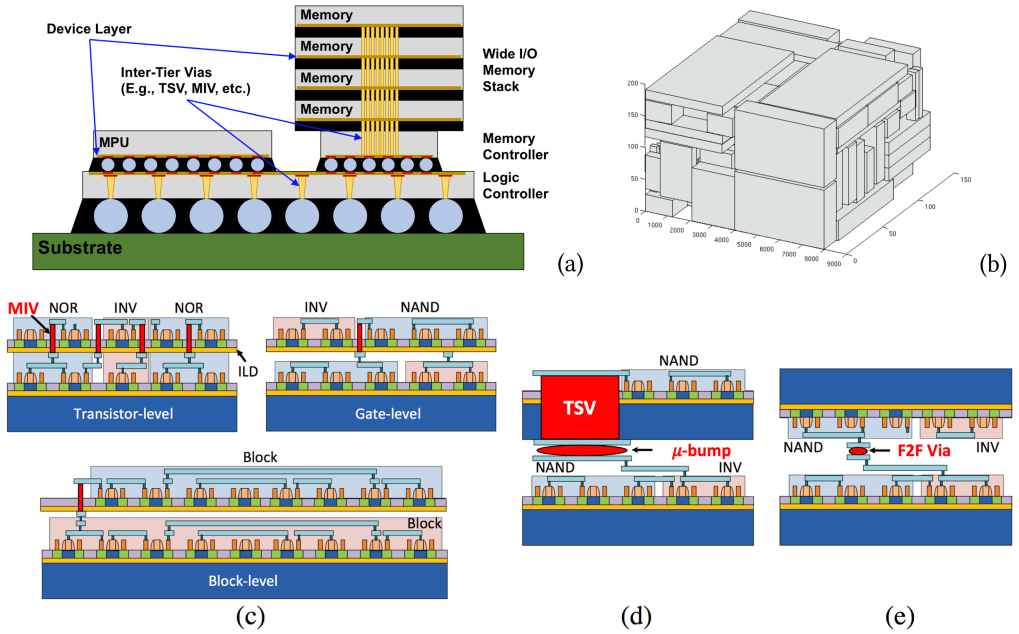

Fig. 1. An example of 3D IC integration. (a) A 3D IC package, re-illustrated based on the original figure in Ref. [47]. (b) A general 3D floorplan to map a design into physical space of hardware [42]. (c) Transistor-, gate-, and block-level monolithic 3D ICs [35]. (d) Gate-level TSV-based and (e) face-to-face (F2F) 3D IC [35].

Despite the recent emphases on the necessity of 3D IC designs, a multitude of challenges has so far obstructed large-scale transition from the classical 2D ICs to stacked 3D ICs [26]. Knechtel and Lienig discuss the most relevant aspects of automating the physical design (PD) process for 3D ICs by highlighting how 3D IC design becomes increasingly difficult and demanding as compared to well-engineered design automation for 2D ICs. Major design challenges for 3D ICs include 3D IC-specific challenges such as 3D fabrication technologies, 3D architecture exploration, system-level interconnection design, 3D stacking-aware partitioning, 3D vertical interconnection's parasitic analysis, and so on, as well as traditional challenges such as placement, clock-tree synthesis, thermal management, reliability, power distribution, and so on.

For a past decade, 3D IC design automation has been intensively studied by both industry and academia [4–6, 9, 16, 22, 24, 31, 35, 36, 38] but not yet sufficient. Therefore, more sophisticated and practical design methodologies are urgently required to fully exploit the benefits of 3D IC technology. For example, for power distribution, IC designers encounter many potential design choices of the packaging technologies such as on-chip and off-chip voltage regulators, decoupling capacitors, number of dies, wire pitches and allocation of vertical vias (i.e., through-silicon via (TSV) or monolithic inter-tier via (MIV)) for millions of nodes [19, 20, 37, 52]. The difficulties of 3D IC designs and implementations call for powerful and efficient design automation algorithms to optimize the 3D IC design.

In this work, we focus on the 3D floorplan representations to enable effective 3D IC design automation. Floorplanning is the first step of IC's physical design (PD) procedure, which determines (absolute or relative) locations and/or dimensions (or aspect ratios) of functional and/or hierarchical blocks/clusters/constraints. Therefore, with good spatial structures, IC practitioners can have better opportunities to obtain high-performance ICs. The solution quality of IC floorplanning directly impacts overall layout design quality in terms of footprint area, die utilization,

timing closure, clock distribution, routing congestion, power consumption, and so on. On top of the discussion above, the spatial and topological structuring of non-overlapping blocks (or block packing/partitioning) is an important theoretical/practical research topic for PD engineers.

Floorplan representation is an important topic for PD engineers as a foundation of data structure, facilitating to envision the early design perspectives for the full-chip level solution quality. Without complete and efficient floorplan representation algorithms, designing high-performance and cost-effective IC layout becomes extremely hard due to the lack of computational tools for IC designers to explore the IC floorplanning solution space. While conventional 2D IC floorplanning researches have published a large amount of literature including notable results, the extension to 3D IC floorplanning remains open with many nontrivial challenges to explore. A complete and efficient 3D floorplan representation enables effective and efficient 3D IC layout design automation. General 3D floorplan representations are also applicable for the classical temporal 2D floorplanning problems (by having the time component as the additional dimension to 2D space) such as dynamically reconfigurable FPGA design [44, 49, 50].

In this work, we introduce the new 3D floorplan representation by using "corner links" and "partial order." We also propose "four trees" representation, which equivalently expresses the corner links of the non-degenerate 3D mosaic floorplan. The corner links representation is a complete and efficient 3D floorplan representation, meaning that (1) a 3D mosaic floorplan can be represented by the corner links representation, and (2) a corner links representation corresponds to a unique 3D floorplan and restores the 3D floorplan in $O(m)$ time. We show that the corner links representation can be reduced to the corresponding partial order representation, indicating that the corner links representation can utilize well-defined and -studied features/properties of 3D TCG-based floorplan representation [50]. We describe and demonstrate several key properties of, and relations between the corner links representation and partial order representation. Our contributions are as follows.

- We introduce a new 3D floorplan representation methodology by using **corner links** and **partial order** representations.
  - —We define a *corner link* as a set of 1/8 corners that intersect at the same coordinate and belong to the adjacent blocks next to each other in X, Y, Z, or diagonal direction. A corner link consists of pair(s) of neighboring corners with odd number of the Hamming distance (i.e., the number of positions at which the corresponding symbols are different [30]).
  - —The corner links representation is a set of the entire corner links in the 3D floorplan.
  - —The partial order defines the topological structure of the 3D floorplan with three transitive closure graphs (TCG) for each direction. We show that the corner links can be reduced to its corresponding partial order, indicating that the corner links representation can utilize well-defined and -studied features/properties of 3D TCG-based floorplan representation.
- We analyze the properties of, and relations between the corner links representation and the partial order representation.
  - —The corner links representation is a complete and efficient 3D floorplan representation, indicating that (1) every 3D mosaic floorplan can be represented by the corner links representation, and (2) a corner links representation corresponds to a single 3D floorplan and restores the 3D floorplan in $O(m)$ time.
  - —The corner links representation can be obtained in $O(m \log m)$ time complexity.
  - —For the non-degenerate 3D mosaic floorplan, the corner links can be equivalently expressed by a set of *four trees* representation.
  - —We introduce two partial-order-related representations, (i) *face partial order* and (ii) *block partial order*, equivalent to the TCG-based partial order representation.

  - —If the partial order representation describes relations between all pairs of blocks in the 3D floorplan, then the partial order representation produces the valid floorplan.
  - —The partial order representation captures all stitching planes in the 3D floorplan, in sorted order by their respective directions.
- We present three algorithms for the following transformations:
  - —From the 3D mosaic floorplan to the corner links representation.
  - —From the corner links representation to the partial order representation.
  - —From the partial order representation to the absolute coordinates of the entire blocks.

The remainder of this article is organized as follows. Section 2 reviews general foundations of classes of floorplans and floorplan representations. Section 3 introduces our new 3D floorplan representation, i.e., corner links, and partial order representation. Section 4 discusses the properties of, and relations between the corner links, four trees, and face and block partial orders. Then, we demonstrate these properties. Section 5 describes the algorithms to obtain corner links from the given 3D floorplan, to convert the corner links representation to the partial order representation, and to restore the absolute coordinates of all blocks from the partial order representation. Section 6 concludes this article and presents future works.

2 FLOORPLAN OVERVIEW

In this section, we first introduce the basis of floorplanning and the related terms such as block overlap, block adjacency, and valid floorplan. Next, we describe three floorplan classifications. In this work, we focus on a class of mosaic floorplans. We present the fundamental floorplan representations and several key characteristics of the mosaic floorplan. Then the foundations of the 2D and 3D transitive closure graphs (TCG) and the adjacent constraint graph (ACG) are described.

2.1 Basis of Floorplanning

In a d-dimensional (d-D) floorplanning (sometimes called as box partitioning) problem, suppose we have a d-D floorplan space P. The floorplan space P is the rectangular or block that covers the entire block components. We partition the given floorplan space P into m blocks (or more generally find an arrangement of m non-overlapping blocks within P). Each block B occupies the space spanning in an interval $[\min B^i,\ \max B^i]$ for each dimension i where B^i is the coordinates of block B in dimension i. In this article, we refer to $\min B^i$ and $\max B^i$ as B^{i-} and B^{i+}, respectively. Following this convention, for 3D floorplanning, we use symbols X, Y, and Z to represent each dimension i in the 3D floorplan space P. Obviously, the strict inequality should hold for all blocks in P, i.e., $\min B^i < \max B^i$ (or $B^{i-} < B^{i+}$) for all dimensions i so that all the blocks have positive volume. Note that, in the later sections of this article, we use superscripts for indicating block's dimensional information (e.g., B^{X+}) and subscripts for indicating block's index (e.g., B_1). We define four terminologies, frequently used in this work as follows.

2.1.1 Interval Overlap. For two blocks B_1 and B_2, we define *interval overlap* in a dimension i if two intervals $[B_1^{i-},\ B_1^{i+}]$ (i.e., $[\min B_1^i,\ \max B_1^i]$) and $[B_2^{i-},\ B_2^{i+}]$ (i.e., $[\min B_2^i,\ \max B_2^i]$) hold two inequalities: (i) $B_2^{i-} < B_1^{i+}$ (i.e., $\min B_2^i < \max B_1^i$) and (ii) $B_1^{i-} < B_2^{i+}$ (i.e., $\min B_1^i < \max B_2^i$). The strict inequality ensures that the overlap has positive length.

2.1.2 Block Overlap. We define *block overlap* of two blocks B_1 and B_2 if intervals $[B_1^{i-},\ B_1^{i+}]$ (i.e., $[\min B_1^i,\ \max B_1^i]$) and $[B_2^{i-},\ B_2^{i+}]$ (i.e., $[\min B_2^i,\ \max B_2^i]$) overlap for all dimensions i in the given d-D floorplan space P. In other words, two blocks B_1 and B_2 are not overlapped if there exists a dimension i that does not hold the interval overlap.

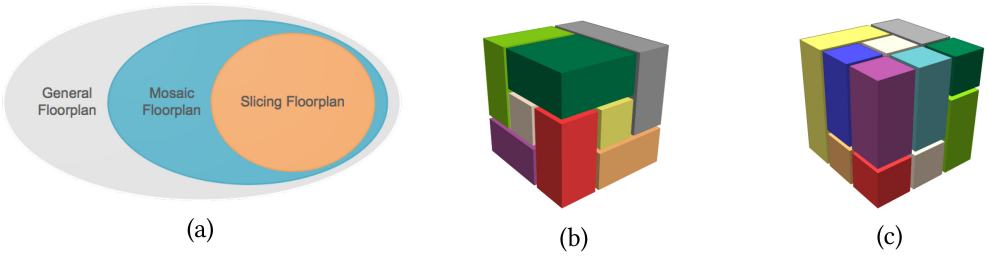

Fig. 2. Classification of floorplans. (a) Set relation chart of floorplan classifications. (b) An example of the mosaic floorplan. (c) An example of the slicing floorplan where we can partition the floorplan into single blocks by cutting blocks recursively.

2.1.3 Block Adjacency. A pair of blocks B_1 and B_2 are *adjacent* if there exists a dimension i such that $B_1^{i-} = B_2^{i+}$ (i.e., $\min B_1^i = \max B_2^i$) or $B_2^{i-} = B_1^{i+}$ (i.e., $\min B_2^i = \max B_1^i$); and for the remaining dimension(s) j where $j \neq i$, all the intervals $[B_1^{j-}, B_1^{j+}]$ (i.e., $[\min B_1^j, \max B_1^j]$) and $[B_2^{j-}, B_2^{j+}]$ (i.e., $[\min B_2^j, \max B_2^j]$) are overlapped. For 3D floorplanning, we define two blocks B_1 and B_2 are *adjacent* if these two blocks are touching the surface of one another.

2.1.4 Valid Floorplan. In the d-D floorplan space P, a floorplan is *valid* if any pair of blocks are not overlapped with each other.

2.2 Classification of Floorplanning

We classify floorplans into three categories, (i) general floorplan, (ii) mosaic floorplan, and (iii) slicing floorplan. Figure 2(a) shows the set relation of three floorplan classifications, and Figures 2(b) and (c) present the examples of mosaic and slicing floorplan, respectively. A set of slicing floorplans is a subset of mosaic floorplans and a set of mosaic floorplans is a subset of general floorplans. In this article, we focus on the mosaic floorplan and its representation methodology.

2.2.1 General Floorplan. A *general floorplan* is a collection of non-overlapping blocks (i.e., non-intersecting blocks), where all blocks are contained within the given floorplan space P. The general floorplan allows empty spaces within the given floorplan space P. To reduce the problem complexity, we often focus on the case of a *compact floorplan*, where all blocks are pushed toward the origin such that no block can be further shifted closer toward the origin without overlapping or moving other blocks.

2.2.2 Mosaic Floorplan. A set of *mosaic floorplans* is a subset of the general floorplans, obtained by partitioning the floorplan space P into multiple blocks (as shown in Figure 2(b)). In the mosaic floorplan, the blocks cover all available floorplan space P, indicating that there are no empty spaces within the given floorplan space P. While nonexistence of empty space is a substantial restriction compared to the general floorplans, the mosaic floorplan provides much simpler thus more efficient structure to study and analyze. This allows designers to reduce possible combinations of the solution range. Also, the mosaic floorplan can be easily extended to the general floorplan by adding empty blocks with a controlled number of gaps (i.e., holes). In addition, the mosaic floorplan gives a more general overview than the slicing floorplan so that physical design engineers can obtain deeper insights of the IC floorplanning problems. We further discuss the mosaic floorplan in Section 2.4.

2.2.3 Slicing Floorplan. A set of *slicing floorplans* is a subset of the mosaic floorplans, obtained by recursively bi-partitioning (i.e., slicing) a block in the floorplan space P into two blocks with

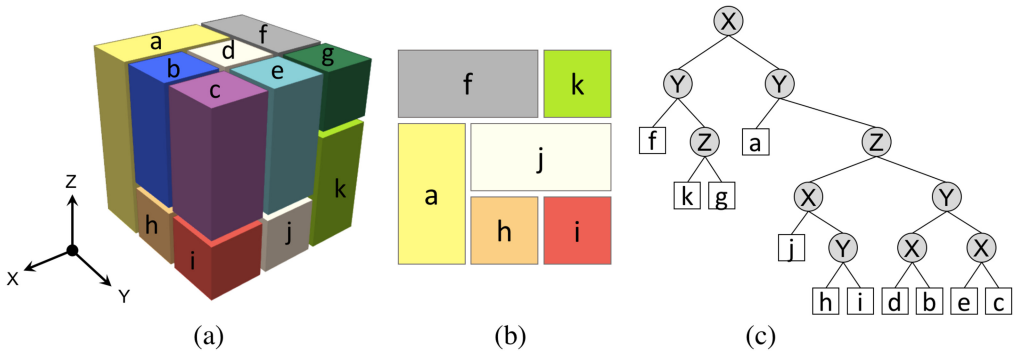

Fig. 3. An example of 3D slicing floorplan with 11 blocks, i.e., blocks *a*-*k*. (a) The 3D slicing floorplan from Figure 2(c) and its coordinate system. (b) The bottom layer layout of the 3D slicing floorplan in Figure 3(a). (c) The slicing tree representation of the 3D slicing floorplan in Figure 3(a).

new boundaries perpendicular to a dimensional axis. The slicing floorplan starts with assuming the floorplan space P as a single block (as shown in Figure 2(c)). With this in mind, the slicing floorplan can be represented with a hierarchical tree structure (i.e., slicing tree) [7] as shown in Figure 3. By its nature, a slicing tree is a full binary tree. The root shows the first cut of the floorplan. Each node of the tree represents one of the three cutting directions in the 3D floorplan space. Each internal node represents the further cuts on the partial subspace, derived through the path from the root to the node. The leaf contains each individual block.

2.3 Fundamental Floorplan Representations

Floorplan representations are fundamental and important research topics in IC physical design. Many floorplanning problems for 2D and 3D ICs are NP-hard [41]. Most floorplanning algorithms are heuristic and rely on perturbations with random searches and iterative improvements. Thus, researches on floorplan representations are crucial, because a floorplan representation (1) provides a foundation of the data structure, (2) decides the algorithm's complexity, and (3) determines the solution space size. Without a complete and efficient methodology for the floorplan representation, designing high-performance and cost-effective IC layout becomes extremely hard due to the lack of computational tools for IC designers to explore the IC floorplanning solution space. Therefore, floorplan representation methodologies that express geometrical information of functional/hierarchical blocks/clusters/constraints directly impact turnaround time of IC floorplan/placement as well as overall layout design quality of results (QoR) with respect to footprint area, routability, timing closure, power consumption, and so on.

Examples of simple floorplan representation methodologies include floorplan specifications based on (i) absolute coordinate, (ii) corner-stitching, and (iii) partial ordering. The absolute coordinate-based floorplan specification is a naïve representation method that directly gives the physical locations of blocks. These can be used, e.g., for visualization purposes to specify the physical position and dimensions of each circuit block in a fixed space. While the absolute coordinate floorplan specification is usefully explicit, it is a high-entropy representation, which makes the design problems more difficult to work with and unsuitable for solving various optimization problems. Corner-stitching is a well-known representation for 2D IC floorplanning [33, 34]. During floorplanning, we track all neighboring blocks through corner stitches connecting opposite corners (e.g., lower-left and upper-right corners in Figure 4) of every block. The stitches combined all together form the data structure for the given floorplan. In this article, our new 3D floorplan

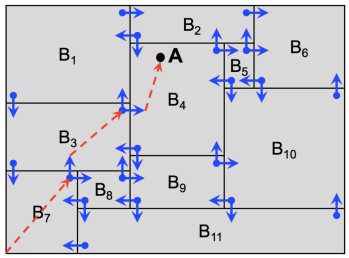

Fig. 4. An example of corner-stitching-based 2D floorplan representation. Each blue arrow depicts stitch, which is pointer. Red dotted arrows show the searching path for the block containing point A by following the pre-defined pointers.

representation methodology, i.e., corner links, is deeply inspired by the corner-stitching representation (see Section 3.1). The partial ordering provides the topological structure of a floorplan with transitive closure graphs per each dimension. There are two partial order-related representations; face partial order and block partial order. The details are described in Section 3.2.

2.4 Mosaic Floorplan

In this section, we discuss the key features of the 2D and 3D mosaic floorplan.

2.4.1 Key Features. The 2D mosaic floorplan is well-described in Reference [18], and the authors summarize the key features of the 2D mosaic floorplan as follows:

- No empty space exists within the floorplan space, i.e., each coordinate is assigned at-least one and only one block.
- The segment intersection forms a "*T-junction*," except the corners of the given floorplan space. The T-junctions (0°, 90°, 180°, 270°) are defined as the point where one end of the non-crossing segment contacts the crossing segment.
- There is no *degenerate* case where two distinct T-junctions meet at the same location. We define the *non-degenerate floorplan* as the floorplan that does not have any degenerate cases.
- The topological structure of the mosaic floorplan remains the same even after adjusting block sizes by shifting T-junction corners, unless the neighboring corners are non-degenerate.

2.4.2 2D Mosaic Floorplan. The mosaic floorplan has been well-studied in past decades [3]. Also, many advances have been made in 2D floorplan representations. In particular, our research on the new 3D floorplan representation, i.e., corner links, is inspired by corner stitching [33, 34] and twin binary trees [46, 48] representations. For the 2D mosaic floorplans, Yao et al. [46] present the precise number of mosaic floorplans with the given-fixed number of blocks. The number of distinct mosaic floorplans with m blocks is equal to the mth Baxter number $B(m)$ [1, 8] as shown in Equation (1). Dulucq and Guibert [12] also prove that the exact number of distinct twin binary trees with m nodes is equal to the Baxter number $B(m)$. Finally, the authors of Reference [46] demonstrate a bijective mapping between the pair of twin binary trees and the 2D mosaic floorplan:

$$B(m) = \binom{m+1}{1}^{-1} \binom{m+1}{2}^{-1} \sum_{k=1}^{m} \binom{m+1}{k-1} \binom{m+1}{k} \binom{m+1}{k+1}. \tag{1}$$

In this work, we define the corners of each block as $1/2^d$ where d is the dimension of the given floorplan space P. Thus, in a 2D floorplan, we refer to each corner of the four corners of a block as

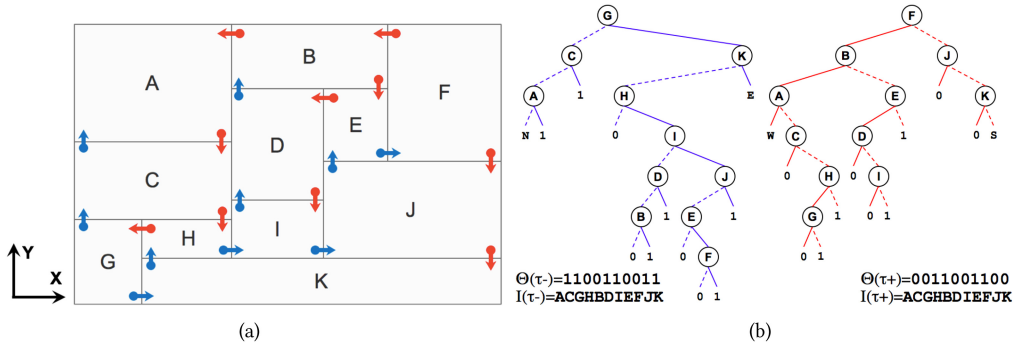

Fig. 5. (a) An example of the 2D mosaic floorplan, which has the same topological structure of the 2D floorplan in Figure 4. (b) The twin binary trees for the floorplan of Figure 5(a). The pair of trees represents the up-right and down-left corner relations.

1/4 corner. Note that the two edges of the corner (i.e., a non-crossing segment at a T-junction) divide the space into a 90-degree angle. Then, a straight line (i.e., a crossing segment at a T-junction) is considered as a 1/2 corner. Therefore, except for the four boundary corners of the 2D floorplan space P, the tip of each 1/4 corner is always connected to either four 1/4 corners or two 1/4 corners and a 1/2 corner. As we discussed in the previous Section 2.4.1, we refer to the four-1/4-corner-junction configuration as a degenerate case. For the 2D mosaic floorplan, we eliminate these configurations by shifting the segment (forming a degenerate case) by a small distance, so that we have no degenerate cases in our consideration. Without degenerate cases, i.e., in the non-degenerate 2D floorplan, the segment intersection always forms a T-junction.

In each dimension, we denote the 1/8 corner closer to the origin as $-$ corner and the 1/8 corner farther from the origin as $+$ corner. And we denote each corner of block B as B^{xy} (or B^{xyz} in 3D) where x, y, and z, respectively, indicate the relative location of the corner as $-$ or $+$ for each dimension. E.g., $B_1^{--} = B_1^{X-Y-}$, indicating block B_1's corner at the minimum X and Y directional coordinate, B_1^{X-} (i.e., $\min B_1^X$) and B_1^{Y-} (i.e., $\min B_1^Y$). For block A in Figure 5, the lower-left, upper-left, lower-right, and upper-right corners are denoted as A^{--}, A^{-+}, A^{+-}, and A^{++}, respectively. For each block in the 2D mosaic floorplan of Figure 5(a), the $--$ corners are linked by the forward arrows (blue arrows) while the $++$ corners are linked by the backward arrows (red arrows).

Figure 5 illustrates (a) an example of 2D mosaic floorplan that has the same topological structure of Figure 4's floorplan and (b) the corresponding twin binary trees of the floorplan in Figure 5(a). A pair of trees represents the up-right and down-left corner relations together. Blue and red arrows in Figure 5(a) indicate up-right and down-left corner relations, and these are depicted by blue and red edges (left and right trees, respectively) in Figure 5(b). Solid and dotted edges in Figure 5(b), respectively, represent horizontal and vertical relations. In Figure 5, $\tau-$ and $\tau+$, respectively, denote twin binary trees for up-right (i.e., blue) and down-left (i.e., red) corner relations. Θ is the leaf label sequence of the tree ignoring the first and the last bits, and I is the node sequence of the tree from the in-order traversal. Each in-order traversal of two trees (i.e., $I(\tau-)$ and $I(\tau+)$) produces the same sequence, $ACGHBDIEFJK$. We extend the two trees into binary leaves with label 0 at left and 1 at right. Traversing the leaves from left to right, the complement relation of the binary strings is observed (i.e., 1100110011 ($\tau-$) vs. 0011001100 ($\tau+$)). Combined Θ and I information together, we can decode and restore the physical structure of 2D mosaic floorplan [46].

2.4.3 3D Mosaic Floorplan. Recently, 3D floorplans have been intensively studied. Research on 3D floorplan representation methodology include sequences of blocks to encode the topology of

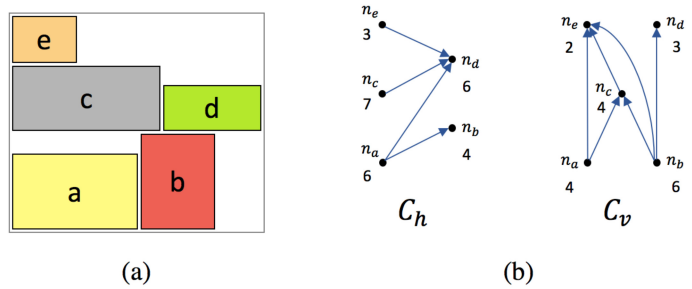

Fig. 6. An illustration for the transitive closure graph (Figure 2 of Reference [29]). (a) An example 2D floorplan, and (b) its horizontal (C_h) and vertical (C_v) transitive closure graphs.

3D packing [45], representation for the 3D slicing floorplan [7], 3D corner block lists (CBL) [32], 3D bonded slice-surface grid (BSG) [51], T-tree data structure for the spatial and temporal relations of blocks in 3D space [49], graph-based topological 3D floorplan representation (3D transitive closure graph) [50], topological structure using weighted directed graph [23], and tree-based approaches and sequences to extend twin binary tree [42, 40, 15]. Fischbach et al. present a comparative study of most of the aforementioned 3D floorplanning methods and more in Reference [14].

As discussed in the previous Section 2.4.2, we define the corners of each block as $1/2^d$ where d is the dimension of the given d-D floorplan space. With this in mind, for the 3D floorplan, we define a point on a vertex of a block as 1/8 corner, since the block occupies $\frac{1}{8}$ of the space at the point. Similarly, we define a point on an edge as 1/4 corner and a point on a face as 1/2 corner, since the block occupies $\frac{1}{4}$ of the space at the point on the edge and $\frac{1}{2}$ of the space at the point on the face. The sum of the fractional corner parts at any internal junction points is always 1.

2.5 Transitive Closure Graph and Adjacent Constraint Graph

2.5.1 Transitive Closure Graph. Suppose a directed-acyclic graph $G = (V, E)$, with vertex set $V = \{1, 2, \ldots, n\}$. We define the *transitive closure* of G as the graph $G^* = (V, E^*)$, where $E^* = \{(i, j)$: there is a path from vertex i to vertex j in $G\}$ [11]. The transitive closure graph (TCG) representation describes the geometric relations among blocks based on two graphs, i.e., a horizontal transitive closure graph C_h and a vertical transitive closure graph C_v [29]. Figure 6 shows (a) an example 2D floorplan and (b) its horizontal (C_h) and vertical (C_v) transitive closure graphs. Transitive closure graph has the following properties: (1) C_h and C_v are acyclic, ensuring a block cannot be placed both left and right (or below and above). (2) Each pair of nodes must be connected by exactly one edge either in C_h or in C_v, which guarantees that no overlaps between any blocks. And (3) the transitive closure of C_h (C_v) is equal to itself, eliminating redundant solutions.

2.5.2 Adjacent Constraint Graph. Proposed in Reference [53], the adjacent constraint graph (ACG) has advantages of both adjacency graph and constraint graph, indicating that the ACG can represent both interconnects and areas. (1) Since edges in an ACG only connect blocks close to each other, the physical distance of two blocks can be measured directly in the graph. (2) Since an ACG is a constraint graph, the floorplan area and block positions can be found by longest path computations. Figure 7 shows (a) an example 2D floorplan, its (b) constraint graph, (c) adjacency graph, and (d) adjacent constraint graph (ACG). ACG is complete and non-redundant floorplan representation by virtue of its symmetrical data structure from the union of horizontal and vertical edges. As preventing two or more edges between any two blocks, ACG has no overlaps between any blocks.

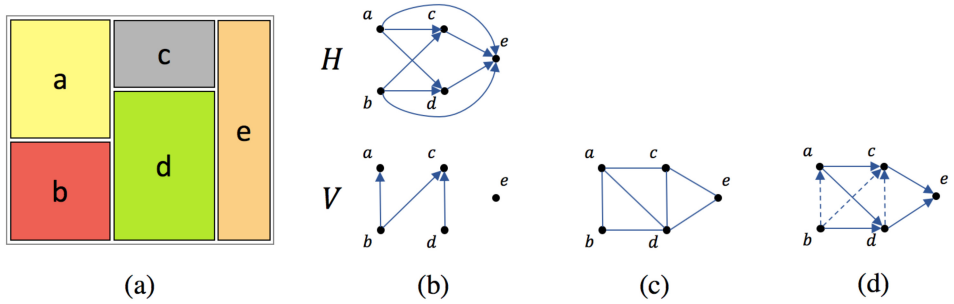

Fig. 7. An illustration for the adjacent constraint graph (Figure 1 of Reference [53]). (a) An example 2D floorplan. (b) Constraint graph, (c) adjacency graph, and (d) adjacent constraint graph (ACG) of Figure 7(a).

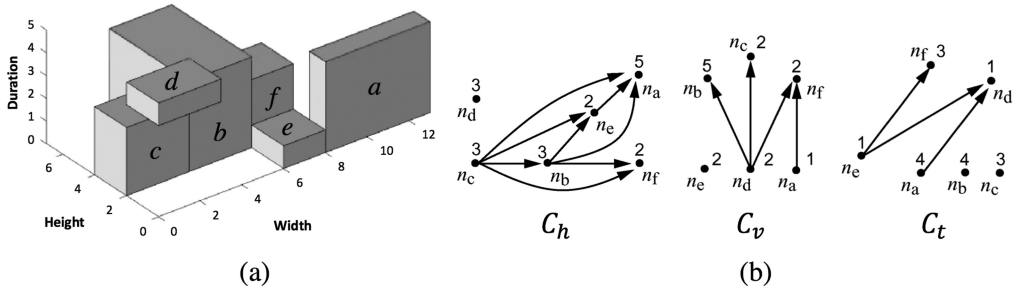

Fig. 8. 3D transitive closure graph (Figures 8 and 9 of Reference [50]). (a) An example 3D floorplan with duration (i.e., temporal information). (b) The corresponding 3D TCG of Figure 8(a).

2.5.3 3D Transitive Closure Graph and our 3D Floorplan Representation. Based on the 2D TCG-based floorplan representation [29] in Section 2.5.1, the authors of Reference [50] introduce the new graph C_t to model the temporal relations among tasks, since they consider the temporal and spatial relations at the same time. The 3D transitive closure graph (3D TCG) [50] is an extension of the 2D TCG [29], which contains three transitive graphs, C_h, C_v, and C_t. Figure 8 shows (a) an example 3D floorplan, and (b) the corresponding 3D TCG of Figure 8(a). 3D TCG has the following properties: (1) C_h, C_v, and C_t are acyclic. (2) Each pair of nodes must have exactly one edge in either C_h, C_v, or C_t. And (3) there must exist an edge (i, j) if there is a path from block i to block j in one graph and there exist no edges between i and j in other graphs. (1) and (2) ensure solution's feasibility, and (3) eliminates the redundant solutions.

3D TCG is a complete and fully topological representation for 3D design. Since the relation between each pair of tasks is defined in the representation, the geometric relation of each pair of tasks is transparent to both the representation and the induced operations [50]. Given a 3D TCG, a 3D floorplan can be obtained in $O(m^2)$ time by performing a well-known longest path algorithm [28] on TCG, where m is the number of blocks. To facilitate the implementation of the longest path algorithm, each closure graph is required to add two special nodes with zero weights, the source n_s and the sink n_t, and edges from n_s to each node with in-degree equal to zero and also from each node with out-degree equal to zero to n_t [29, 50]. In addition, operations for solution perturbation are introduced in Reference [50]. For instance, Rotation, Swap, Reverse, and Move can be performed in respective $O(1)$, $O(1)$, $O(m^2)$, and $O(m^2)$ time. In our study, we adopt the 3D transitive closure graph-based (3D TCG-based) 3D floorplan representation [50] as our partial

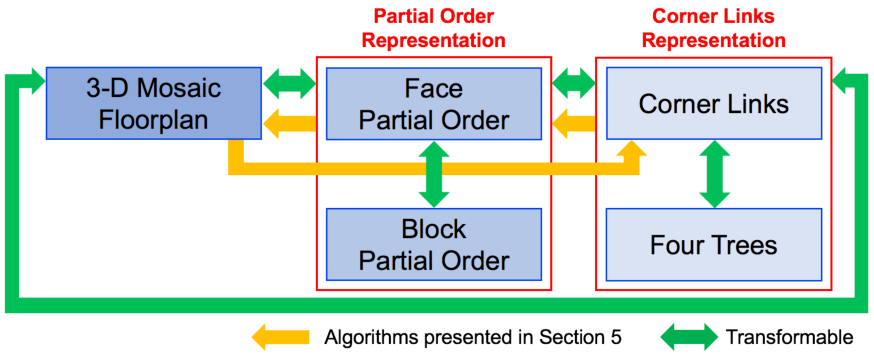

Fig. 9. Relations of the 3D floorplan representations in this work. The corner links representation and four trees representation and face/block partial order representation. Green arrows depict transformable relations. Yellow arrows show transformations with Algorithms presented in Section 5.

order representation for bridging our corner links representation to its application and valid 3D mosaic floorplan.

3 NEW 3D FLOORPLAN REPRESENTATIONS: CORNER LINK AND PARTIAL ORDER

In this work, we develop novel 3D floorplan representations and present their properties. In this section, we introduce the new 3D floorplan representation methodology, *corner links representation,* using *corner link*, composed of a set of 1/8 corners (i.e., vertices) that belong to the adjacent blocks next to each other in X, Y, Z, and/or diagonal direction in the 3D floorplan space. Then, we connect our new 3D floorplan representation to the *partial order representation* [50] that shows the topological structure of the 3D floorplan with three transitive closure graphs for each dimension. Figure 9 summarizes the relations of the 3D mosaic floorplan, corner links and four trees representations, and face and block partial orders. Each arrow describes the transformable relations through our proposed Lemmas and Theorems in Section 4 and Algorithms in Section 5.

3.1 Corner Links Representation

Corner links representation is our new representation methodology for the 3D mosaic floorplan.

3.1.1 Corner Links. Inspired by corner stitching [33, 34], 2D mosaic floorplan representation with T-junction [18], and twin binary trees representation [46, 48], we define *corner links* as the spatial relations that describe all of the corner relations among the entire blocks in the 3D mosaic floorplan. We define a *corner link* as a set of 1/8 corners (i.e., vertices) that (1) intersect at the same coordinate and (2) belong to the adjacent blocks next to each other in X, Y, Z, and/or diagonal direction. For each corner link, we link pairs of 1/8 corners in the adjacent blocks together that intersect at the same point, which we call *neighboring corners*. Since we consider 3D mosaic floorplans, each corner link has at least two 1/8 corners, i.e., we have at least one pair of neighboring corners in every corner link. If a corner has multiple 1/8 neighboring corners, then we consider neighboring corners with the odd number of the Hamming distance.[1] Each corner link can have up to eight 1/8 corners, i.e., up to 16 pairs of neighboring corners in a corner link. By linking neighboring corners of the Hamming distance 1 or 3, the corner links representation captures all pairs

[1]The Hamming distance between two same-length vector/string is defined as the number of positions at which the corresponding components/symbols are different [30]. For example, the Hamming distance between + + + and + − + is 1.

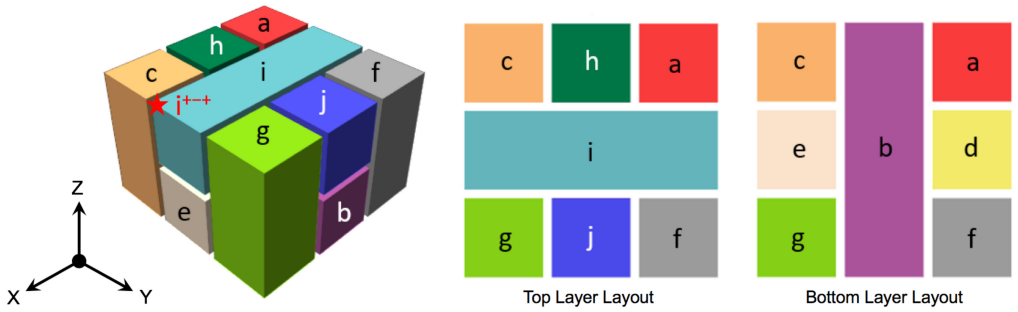

Fig. 10. A 3D mosaic floorplan and layouts of the top and bottom layers. There are ten blocks, a–j. A vertex of block i depicted by the red star indicates i^{+-+} corner (i.e., i^{X+Y-Z+}), meaning that the corner is located at maximum X-, minimum Y-, and maximum Z-coordinates of block i.

of such neighboring corners (i.e., all links between adjacent blocks) in the 3D floorplan, indicating the completeness of the corner links representation.

Figure 10 shows an example of the 3D mosaic floorplan and layouts of the top and bottom layers. The 3D floorplan space P (i.e., the outermost block that covers the entire blocks in the 3D floorplan) has the origin at the farthest point from the reader's point of view (as shown in the left-most arrows depicting X-, Y-, and Z-direction). There are ten blocks, a–j. For the 3D floorplan space P and each block B (where B is either P or a–j), we denote vertices (i.e., 1/8 corners) of each block as B^{xyz} where x, y, and z are either + or −. In each dimension, the symbols + and −, respectively, indicate the relative position of the 1/8 corners in each block on X-, Y-, or Z-direction.

For example, block i in Figure 10 has eight 1/8 corners. And one of block i's vertices (i.e., 1/8 corner) depicted by the red star means that the corner is located at maximum X- (i^{X+}, i.e., $\max i^X$), minimum Y- (i^{Y-}, i.e., $\min i^Y$), and maximum Z-coordinates (i^{Z+}, i.e., $\max i^Z$) of block i. This corner is expressed as i^{X+Y-Z+}, and simplified to i^{+-+}. Corner i^{+-+} is linked to block c's 1/8 corner c^{+++} in Y-direction (i.e., through XZ-place). Two corners i^{+-+} and c^{+++} are a pair of neighboring corners, and these two corners compose a corner link as there are no other neighboring corners at the given location in this 3D floorplan.[2] Between two neighboring corners, the complement relation for Y-direction is observed, since two neighboring corners are linked in Y-axis. Similarly, corner i^{++-} is linked to corner e^{+++} in Z-direction, corner e^{-++} is linked to corner j^{+--} in diagonal direction, and so on.

Each link specifies a pair of 1/8 neighboring corners. A corner link is a set of pairs of 1/8 neighboring corners intersecting at the same coordinate, and the corner links representation is a set of all corner links in the 3D floorplan. A set of pairs of neighboring corners is a particular set of all corners in the 3D floorplan that are pair-wise equal by the *corner equation*. Thus, the corner links represents all stitching planes in the 3D floorplan. We write corner equations to describe each pair of neighboring corners at a point. For the 3D floorplan in Figure 10, $e^{-++} = j^{+--}$ is a corner equation equivalent to $e^{X-Y+Z+} = j^{X+Y-Z-}$, meaning that $e^{X-} = j^{X+}$ (i.e., $\min e^x = \max j^x$),

[2]For the 2D mosaic floorplan, the authors of Reference [18] define the non-degenerate floorplan and exclude degenerate cases, implying that every corner has one neighboring 1/4 corner at most (as discussed in Section 2.4). For the 3D floorplan, it is nontrivial to define non-degenerate floorplan by the number of 1/8 corners in a corner link, because there are non-degenerate 3D floorplans with corner links having multiple pairs of neighboring corners (as shown in Figure 20). Instead, we define the non-degenerate 3D floorplan as a floorplan that we cannot perturb the topological structure by adjusting any block's volume. The proposed corner links representation, nonetheless, can completely abstract all 3D mosaic floorplans regardless of non-degenerateness, i.e., the corner links can represent both degenerate and non-degenerate 3D mosaic floorplans.

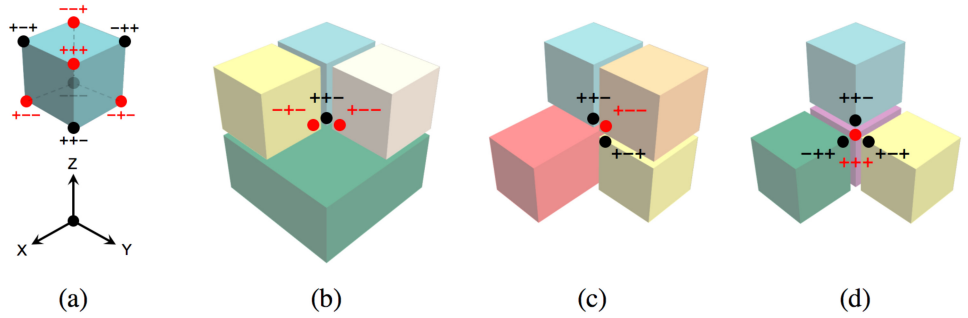

Fig. 11. Illustrations of opposite corners and neighboring corners. (a) Each red dot is one of four opposite corners in 3D floorplan, forming the root for each of trees in four trees representation (as shown in Figure 12). (b–d) Red (respectively, black) corners have outgoing (respectively, incoming) links (i.e., edges in the four trees representation) to (respectively, from) any neighboring corners in the corner link.

$e^{Y+} = j^{Y-}$ (i.e., $\max e^y = \min j^y$), and $e^{Z+} = j^{Z-}$ (i.e., $\max e^z = \min j^z$). In the 3D mosaic floorplan of Figure 10, there are 36 pairs of neighboring corners thus 36 corner equations, and 36 sets of corner links. More generally, if there are m blocks in the 3D mosaic floorplan, the number of total neighboring-corner pairs are $4 \times (m - 1)$.

3.1.2 Four Trees. Corner links for 3D mosaic floorplan are conceptually matched to T-junctions for 2D mosaic floorplan as they define the neighboring corners of the entire blocks. A 2D mosaic floorplan can be equivalently represented by twin binary trees by using T-junctions. Similarly, the corner links representation is capable of building trees for representing the 3D mosaic floorplan. However, two trees are insufficient to represent all spatial corner relations defined by corner links. Instead, a 3D mosaic floorplan requires four distinct trees that include all corner links. Particularly, if the given 3D mosaic floorplan is non-degenerate floorplan, the corner links representation is equivalently expressed by the *four trees* representation. The *non-degenerate 3D floorplan* is a floorplan that we cannot perturb the topological structure by adjusting any block's volume. The non-degenerate 3D mosaic floorplan must have its corresponding four trees representation.[3]

For the four trees representation of the given 3D floorplan, each of four trees is rooted at each of four opposite corners of the 3D floorplan space P, i.e., P^{+++}, P^{--+}, P^{+--}, and P^{-+-} corners or P^{---}, P^{++-}, P^{-++}, and P^{+-+} corners. Figure 11 illustrates (a) opposite corners as red (or black) dots and (b) neighboring corners to create four trees. For the discussions below, we use the set of opposite corners depicted by red dots, forming roots of four trees representation as shown in Figure 12. When the 3D floorplan has m blocks, each of four trees has m nodes and $m - 1$ edges, thus the total number of edges is $4 \times (m - 1)$. Each edge represents a link between a pair of neighboring corners, describing the spatial relationship of the pair of blocks (containing each of 1/8 neighboring corners) in X, Y, Z, or diagonal direction. By traversing the entire edges in the four trees, we can construct the original 3D floorplan, indicating that the four trees representation has $O(m)$ complexity.

Figure 12 shows the four trees representation, which equivalently expresses the 3D mosaic floorplan in Figure 10. In Figures 12(a)–12(d), each tree is, respectively, rooted at the four opposite corners of the 3D floorplan space P, i.e., (a) P^{+++}, (b) P^{--+}, (c) P^{+--}, and (d) P^{-+-}. Instead of P^{xyz}, each root of four trees is, respectively, denoted as g^{+++}, a^{--+}, c^{+--}, and f^{-+-}, because the four opposite corners of the 3D floorplan space P share the corners of g^{+++}, a^{--+}, c^{+--}, and f^{-+-},

[3]Degenerate floorplans can be converted into non-degenerate floorplans after applying a very tiny adjustment.

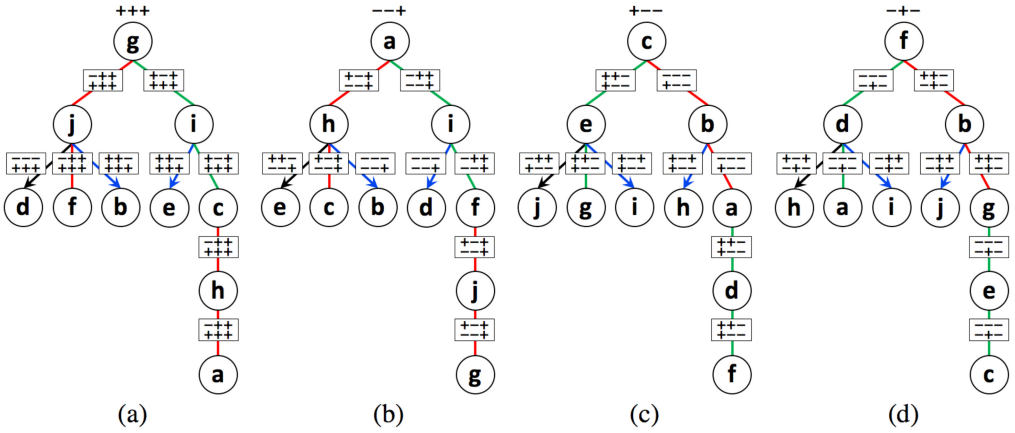

Fig. 12. Four trees representation, which equivalently expresses the 3D mosaic floorplan in Figure 10. Each of the four trees is, respectively, rooted at the opposite corners of the 3D floorplan space P, i.e., (a) P^{+++}, (b) P^{--+}, (c) P^{+--}, and (d) P^{-+-}. Each tree has nine edges, since the total number of blocks is 10.

respectively. Each node denotes blocks. Each edge represents the spatial relationship of each pair of neighboring corners in diagonal (black), X (red), Y (green), and Z (blue) directions. Since we consider the pair of neighboring corners that has the odd number of the Hamming distance, every diagonal relationship has the Hamming distance 3, and the others have 1. For example, there is an edge showing diagonal direction neighboring corners between blocks j and d (j^{---} and d^{+++}, the Hamming distance is 3) in Figure 12(a). We have 36 neighboring corners in the 3D floorplan, so the total number of edges in four trees is equal to 36.

On each edge of the four trees in Figure 12, there is a rectangle showing the relative locations of the 1/8 neighboring corners, i.e., one corner from the parent node (upper xyz symbols where x, y, z are either $+$ or $-$) and another corner from the child node (lower xyz symbols). For example, the rectangle $\left[\begin{smallmatrix} - & + & + \\ + & + & + \end{smallmatrix}\right]$ between nodes (i.e., blocks) g and j in Figure 12(a) presents that two corners g^{-++} and j^{+++} fabricate a pair of neighboring corners. In other words, blocks g and j are adjacent next to each other in X-axis direction and share the same coordinate at g^{-++} and j^{+++} corners. We refer the corner from the parent node (e.g., g^{-++}) and the corner from the child node (e.g., j^{+++}) as the *linking* and *linked corners*, respectively. Note that all linked neighboring corners in each tree have the same relative corner locations of each block. This is because each child node represents the linked and adjacent block to the direction from the root through its parent node. In Figure 12, edges depicted by arrows are stitching together the stitching plane p_3 for Figure 18. In four trees representation, every edge is mapped into a specific stitching plane as each link is perpendicular to corresponding stitching plane(s). The stitching plane takes a crucial role for the partial order representation and will be discussed in Section 3.2.

Figures 13(a) and 13(b) present the examples of constructing trees from the roots P^{+++} and P^{--+} (i.e., g^{+++} and a^{--+}), respectively. (a) Red and (b) blue arrows indicate the neighboring corners. Black dots are four opposite corners of the 3D floorplan space P. For Figure 13(a) from the root g^{+++}, the linking neighboring corner g^{-++} links its linked neighboring corner j^{+++} in X direction. Block j has three neighboring corners from j^{-++} to f^{+++} in X direction, from j^{++-} to b^{+++} in Z direction, and from j^{---} to d^{+++} in diagonal direction. From the root, block i is linked with g^{+-+} in Y direction. Then block i links corners c^{+++} and e^{+++} in Y and Z directions, respectively. Block c links corner h^{+++}, and block h links corner a^{+++}. Figure 13(b) shows another example to construct

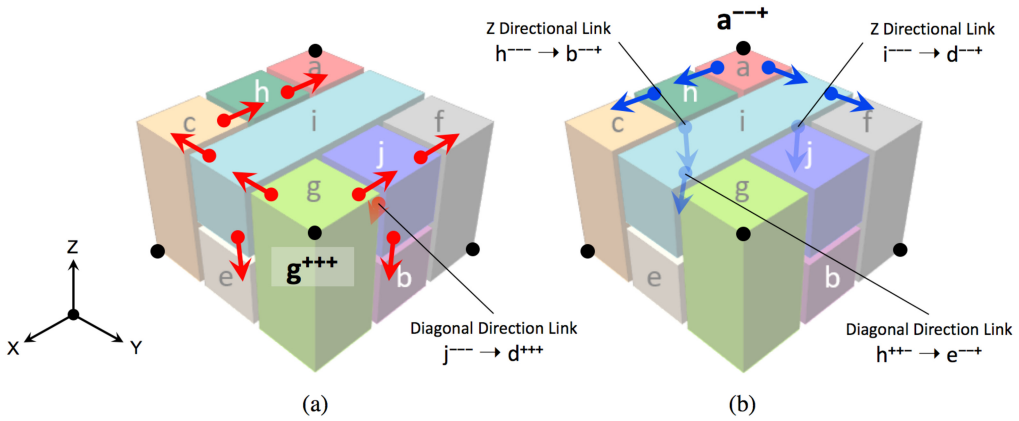

Fig. 13. Examples of constructing trees in Figures 12(a) and 12(b). Figures (a) and (b) correspondingly illustrate traversing procedures from each root, i.e., (a) a root of corner g^{+++} and (b) a root of corner a^{--+}. Blocks j in (a) and h in (b) have three neighboring corners to diagonal, X, and Z directions.

tree from the root a^{--+}. Note that we must consider diagonal-direction link otherwise the tree may not traverse all blocks in the 3D floorplan. For example, the tree in Figures 12(a) and 13(a) misses block d without the diagonal-direction link from j^{---} to d^{+++}, resulting in incomplete 3D floorplan representation.

3.2 Partial Order Representation

In this section, we describe the partial order representation for 3D floorplan. The partial order defines the topological structure of the 3D mosaic floorplan with three transitive closure graphs (TCGs) per each dimension [50]. In this work, we have two partial-order-related representations; (i) face partial order and (ii) block partial order representations. The face partial order is our novel representation method that is equivalent to the block partial order. Both of them can produce transitive closure graphs for each dimension.

3.2.1 Face Partial Order. For each dimension in the 3D floorplan space, there exists a *face partial order* through the faces of the blocks perpendicular to the dimensional direction. We determine the face partial order by defining *face (in)equalities*, letting touching faces of adjacent blocks be equal and relating opposite faces of the same block based on their coordinate in the appropriate dimension. By taking the transitive closure graphs of each face partial order relation for each dimension, we obtain a *face partial order representation* of the floorplan. The face partial order provides complete abstraction for the topological structure of the 3D mosaic floorplan. We determine the face partial order based on the minimum (i.e., B^{i-}, $\min B^i$) and maximum (i.e., B^{i+}, $\max B^i$) coordinate of each block B in each dimension i. That is, the face partial order in X direction is a transitive closure graph composed of these relations by defining any of two touching YZ faces of blocks as equal, and the two YZ faces of the same block as naturally ordered. Therefore, every coordinate on YZ planes in the 3D floorplan has and belongs to the corresponding face partial order.

Figure 14 is the face partial order representation for the 3D mosaic floorplan in Figure 10. Figures 14(a)–14(c) present the face partial orders in X (red), Y (green), and Z (blue) directions, respectively. The $-$ (respectively, $+$) sign indicates the minimum (respectively, the maximum) coordinate of blocks (i.e., the closest (respectively, the farthest) faces of blocks from the origin in the dimension). For example, in Figure 14(a), B^{X-} (equivalent to $\min B^X$) denotes the YZ face of block B that has the smallest X coordinate where B is either the 3D floorplan space P or blocks a–j. For any

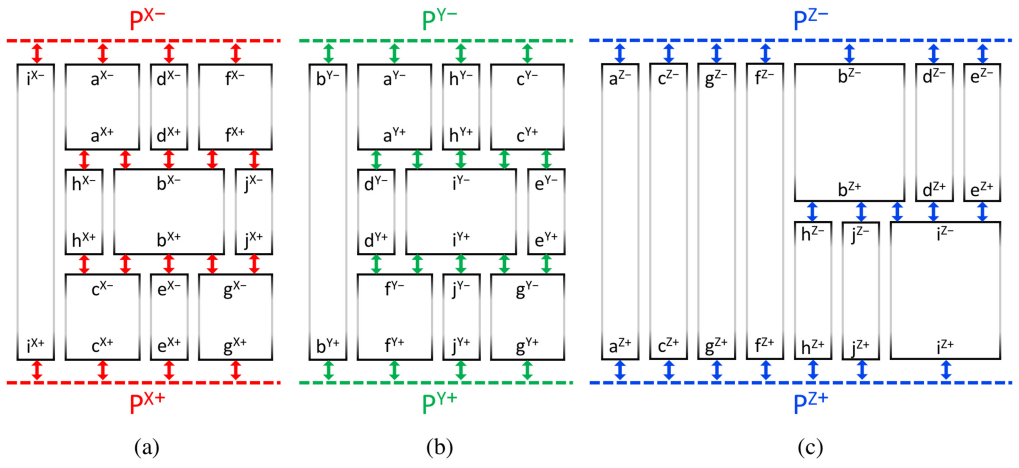

Fig. 14. The face partial order representation for the 3D mosaic floorplan in Figure 10. Figures (a), (b), and (c) present the face partial orders in X (red), Y (green), and Z (blue) directions, respectively.

point in YZ plane of block i, there are no other blocks in the X directional face partial order, since block i directly touches both YZ faces of the given 3D mosaic floorplan space P (i.e., P^{X-} and P^{X+}) . In the 3D floorplan of Figure 10, there are four blocks a, d, f, and i touching P^{X-} plane. Block a is facing block h and b, blocks h and b are commonly facing block c, and block c is facing P^{X+}. Starting with P^{X-}, finding the entire YZ-plane face partial order relations among all blocks constructs the complete X-directional face partial order representation as shown in Figure 14(a). Similarly, we obtain Y- and Z-directional face partial order representations as shown in Figures 14(b) and 14(c).

In Figure 14(a), we obtain the following face equations by the definition of the face partial order representation: $a^{X+} = h^{X-} = b^{X-} = d^{X+} = f^{X+} = j^{X-}$, meaning that $X+$ faces of blocks a, d, and f and $X-$ faces of blocks h, b, and j are on the same YZ plane. This YZ plane fabricates a YZ *stitching plane*, i.e., p_1, which we define as a equivalence class through the corner equations. Similarly, we have the following relations: $h^{X+} = c^{X-} = e^{X-} = b^{X+} = j^{X+} = g^{X-}$, describing another stitching plane p_2. Intuitively, the stitching plane consists of two sets of block faces touching in a particular axis, which must have the same footprints (which we prove in Section 4). The stitching planes provide key concept to assemble 3D mosaic floorplan based on the partial order representation.

3.2.2 Block Partial Order. We determine a *block partial order* on the blocks by defining $A > B$ (where A and B are distinct blocks in the 3D floorplan) if each properly oriented face of A is greater than or equal to each properly oriented face of B in the face partial order of corresponding direction. For example, in X direction, $A > B$ if the minimum X-coordinate face of A (i.e., A^{X-}) is at least as large as the maximum X-coordinate face of B (i.e., B^{X+}) along with the X-directional face partial order. The block partial order and the face partial order are equivalent. And both are equivalently capable of constructing the partial order representation, which abstracts the topological structure of the given 3D floorplan. Figures 15(a)–15(c) present the equivalence relations of (a) the X-directional face partial order (from Figure 14(a)), (b) the X-directional block partial order, and (c) the X-directional partial order representation, the X-directional transitive closure graph for the topological structure of the 3D floorplan of Figure 10. Two YZ planes p_1 ($p_1 = p_1^{X-} = p_1^{X+}$) and p_2 ($p_2 = p_2^{X-} = p_2^{X+}$) fabricate two distinct YZ stitching planes (i.e., p_1 and p_2) in X direction, which we observe in Figure 10. For two blocks a and j, by analyzing the partial order representation, we

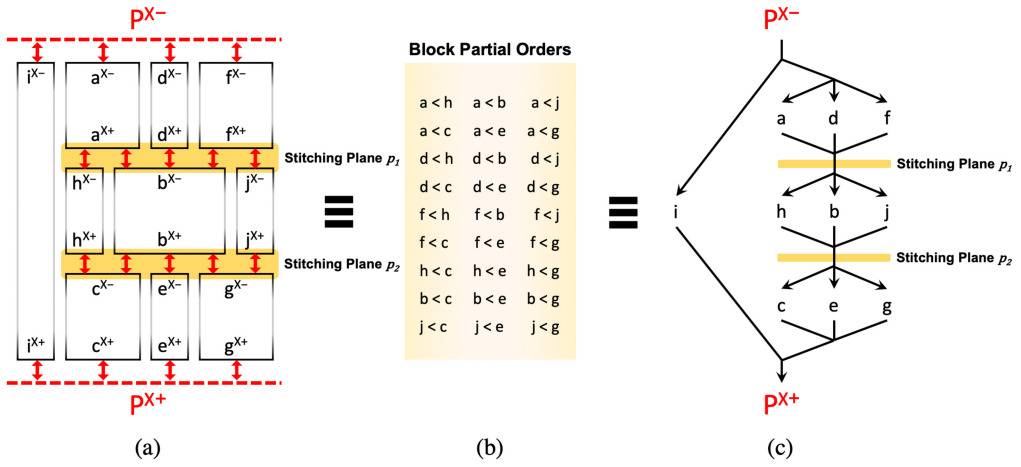

Fig. 15. The equivalent relations of (a) the X-directional face partial order (from Figure 14(a)), (b) the X-directional block partial orders, and (c) the X-directional partial order representation presented as a TCG.

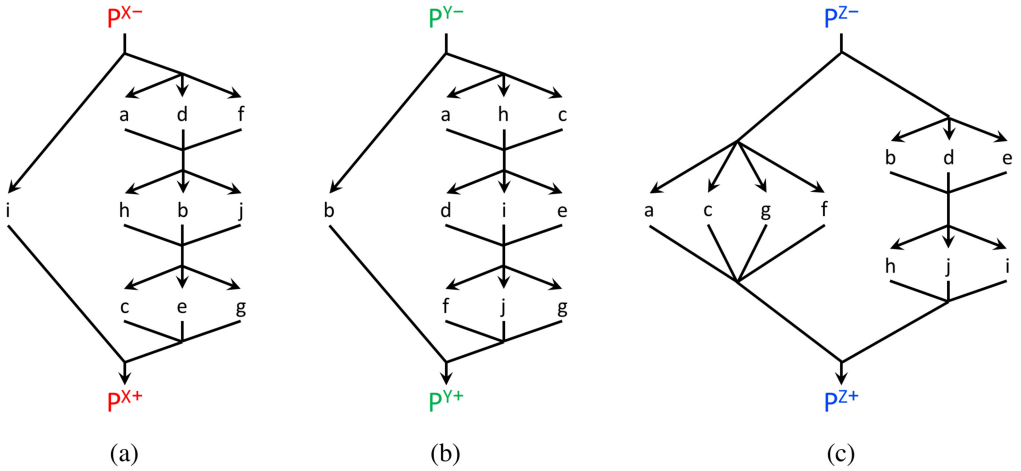

Fig. 16. The X-, Y-, and Z-directional partial orders for the 3D mosaic floorplan in Figure 10.

notice that the two blocks are facing the same stitching plane p_1 although these two blocks are not directly adjacent each other. There are no corner links to stitch them together, however, the partial order representation provides intuitive overview for the stitching planes, containing key joints to assemble blocks for the given 3D floorplan. Finally, Figures 16(a)–16(c), respectively, show the X-, Y-, and Z-directional partial order representations with three transitive closure graphs.

4 3D FLOORPLAN REPRESENTATION PROPERTIES

In this section, we present several key properties for the corner links and partial order representations. Theorems and Lemmas with proofs demonstrate characteristics and relationships of the corner links, four trees, partial order representations, and their corresponding 3D mosaic floorplan. We show that (i) the corner links representation can be reduced to the partial order representation, (ii) the corner links representation for the non-degenerate 3D mosaic floorplan can be equivalently expressed by four tree representation, (iii) a 3D floorplan is a valid floorplan if the partial order

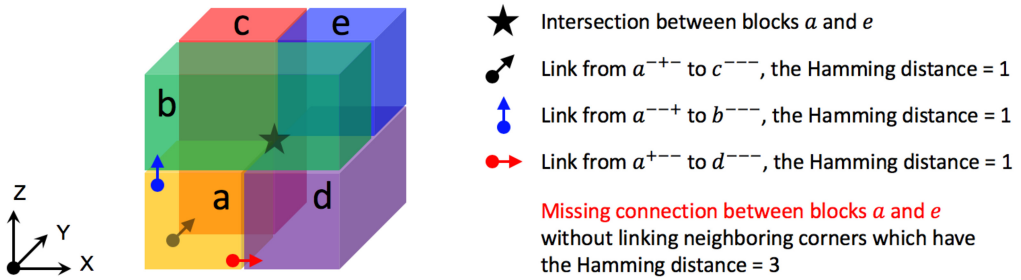

Fig. 17. A 3D mosaic floorplan with five blocks, a–e. Without linking neighboring corners that have the Hamming distance 3, i.e., a^{+++} and e^{---}, we cannot describe the spatial relationship between blocks a and e.

representation describes relations between all pairs of blocks in the 3D mosaic floorplan, and (iv) a partial order representation can restore the absolute coordinates of all blocks in the 3D mosaic floorplan by using the given physical dimensions of blocks.

4.1 Corner Links Representation Properties

The corner links representation is a complete representation for a 3D mosaic floorplan. The corner links representation captures all pairs of 1/8 neighboring corners (i.e., all links between adjacent blocks) in the 3D floorplan. If there is a corner link with multiple neighboring corners (more than two), then we link 1/8 neighboring corners of the Hamming distance 1 or 3, and the neighboring corners of the Hamming distance 2 are automatically covered (see Algorithm 1 in Section 5.1).

Theorem 4.1 (Completeness of the Corner Links for 3D Mosaic Floorplan). *For a 3D mosaic floorplan, the corner links representation (obtained by Algorithm 1) can always completely build four trees by linking all pairs of* 1/8 *neighboring corners of the Hamming distance 1 or 3.*

Proof.

(1) All pairs of 1/8 neighboring corners having the Hamming distance 1 are linked together.
(2) By (1), all pairs of neighboring corners having the Hamming distance 2 are linked together, since there must exist a block[4] that links each block of neighboring-corner pair together by the Hamming distance 1.
(3) Consider a 3D mosaic floorplan with five blocks, a-e, as shown in Figure 17.
(4) For block a and corner a^{+++} depicted by block star, if all of the Hamming distance 1 neighboring corners are unavailable to link corner e^{---}, then corner a^{+++} does not have any neighboring corners (i.e., contradiction to Lemma 4.2 described in Section 4.2).
(5) By Lemma 4.2, corner a^{+++}'s the only 1/8 neighboring corner is e^{---}, which has the Hamming distance 3.
(6) Therefore, by linking all pairs of 1/8 neighboring corners of the Hamming distance 1 or 3, the corner links completely represent the given 3D mosaic floorplan. □

The corner links representation has $4(m-1)$ links fabricating the given 3D mosaic floorplan with m blocks, since every block in the 3D floorplan will be linked exactly once from each opposite corner (there are four opposite corners). By traversing the entire links in the corner links representation, we can assemble the original 3D floorplan, indicating that the corner links representation has $O(m)$ time complexity to restore the given 3D floorplan.

[4]If there does not exist such a block, then the pair of blocks are separated by two 1/4 corners, indicating that the pair of blocks do not have neighboring relation.

For the given 3D mosaic floorplan, the corner links representation can be obtained in $O(m \log m)$ time complexity. We first sort the block list per each X, Y, and Z coordinate. For each block B, we search the adjacent blocks that have 1/8 neighboring corners with B, among the blocks having the larger (respectively, smaller) minimum (respectively, maximum) coordinate than block B's maximum (respectively, minimum) coordinate in the dimension. Algorithm 1 in Section 5.1 shows the procedure to obtain the corner links representation from the given 3D floorplan.

The solution space size of the corner links representation is determined by the number of 4-ary labelled trees with m nodes where $m \geq 1$, since each node can have up to four children nodes. The number of k-ary unlabelled trees with m nodes is a generalization of the Catalan numbers [17, 25] as shown in Equation (2):

$$ {}_kC_m = \frac{1}{km+1}\binom{km+1}{m} = \frac{1}{(k-1)m+1}\binom{km}{m} = \frac{1}{m}\binom{km}{m-1}. \tag{2} $$

Since every unlabelled tree with m nodes creates $m!$ distinct labelled trees by assigning different permutations of labels to all nodes, the number of k-ary labelled trees with m nodes is $m!{}_kC_m$. Based on Stirling's approximation and [13], the asymptotic expression of ${}_kC_m$ is

$$ {}_kC_m \approx \left(1 + O\left(\frac{1}{m}\right)\right)\left(\frac{k}{2\pi}\right)^{1/2}\Big((k-1)m\Big)^{-\frac{3}{2}}\left(\frac{k^k}{(k-1)^{k-1}}\right)^m, \tag{3} $$

where $k \geq 2$. Therefore, the number of the 4-ary labelled trees with m nodes is $m!{}_4C_m$. Since the corner links representation requires four 4-ary labelled trees, the size of solution space is

$$ (m!{}_4C_m)^4, \tag{4} $$

and it is approximately expressed as

$$ O\left((m!)^4\left(\frac{4^4}{3^3}\right)^{4m}\Big/ m^6\right). \tag{5} $$

As a result, for a 3D mosaic floorplan with m blocks, Equation (5) shows the upperbound of the solution space size for the corner links representation.

4.2 Corner Links and Partial Order Representations

The corner links representation is capable of composing the partial order representation, indicating that the information from the corner links representation is sufficient to abstract the topological structure of a 2D or 3D mosaic floorplan. The key benefit of converting to the partial order representation is that the corner links representation can adopt useful features of the 3D transitive closure graph (TCG)-based representation [50], such as operations for solution perturbation.

4.2.1 Corner Links and 2D Floorplan. Corner links representation for 2D mosaic floorplan is equivalent to twin binary trees representation when the mosaic floorplan is non-degenerate. The twin binary trees include all T-junctions (i.e., neighboring corners) in the 2D mosaic floorplan except at the four corners of the 2D floorplan space P. Since every corner link is composed of a single pair of neighboring corners, twin binary trees and corner links representations are equivalent to each other for 2D mosaic floorplan. In Figure 5 above, (a) illustrates the example of the corner links (red and blue arrows) for the 2D floorplan, and (b) illustrates the corresponding twin binary trees based on the corner links in (a).

4.2.2 Corner Links to Partial Order Representation. In this section, we show that the corner links representation can be reduced to the partial order representation. Converting the corner links representation to the partial order representation (i.e., 3D TCG-based representation, 3D TCG) enables

to utilize several key features of 3D TCG-based representation. Since many of floorplan representations are usually used in simulated annealing-based frameworks, it is important to provide several floorplanning operations perturbing the solutions. In Reference [50], five operations are defined to perturb a 3D floorplan represented in 3D TCG as follows:

- *Rotation.* Rotate a block.
- *Swap.* Swap two nodes in X-, Y-, and Z-directional TCGs.
- *Reverse.* Reverse a reduction edge[5] in X-, Y-, or Z-directional TCG.
- *Move.* Move a reduction edge from one TCG to another TCG.
- *Transpositional Move.* Move a reduction edge from one TCG to another TCG, and then transpose the two nodes associated with the edge.

As discussed in Section 2.4.3, we define the corners of each block as $1/2^d$ where d is the dimension of the floorplan space P. For 3D floorplan, we define a vertex, edge, and face of each block as a 1/8, 1/4, and 1/2 corners, respectively. Given any internal junction point in the 3D mosaic floorplan, the total sum of the fractional corner parts at the junction point is always 1, since mosaic floorplans do not allow any empty spaces. Consequently, there are always even number of 1/8 corners at any internal junction point in the 3D mosaic floorplan. From Lemma 4.2, we claim that the structure of corner links is a connected and acyclic directed graph.

Lemma 4.2 (Odd Number of Neighboring Corners). *Every* 1/8 *corner in a 3D mosaic floorplan, other than those at the eight outermost* 1/8 *corners of the 3D floorplan space P, must have an odd number of neighboring corners.*

Proof.

(1) The size of a single 1/8 corner (i.e., vertex) of each block in 3D space is $\frac{1}{8}$.
(2) Except the eight outermost corners of the 3D floorplan space P, each 1/8 corner at a certain point is contained in $\frac{1}{4}$, $\frac{1}{2}$, or 1 of space.
(3) To fill $\frac{1}{4}$, $\frac{1}{2}$, or 1 of the space at the point, even number of 1/8 corners are required.
(4) Therefore, each 1/8 corner requires an odd number of 1/8 neighboring corners to round up to $\frac{1}{4}$, $\frac{1}{2}$, or 1. □

The XY plane between blocks b, d, e and h, j, i in Figure 10 is a stitching plane p_3 in Figure 18(a). The stitching plane p_3 is the only stitching plane in Z direction. The stitching plane p_3-related edges (between neighboring corners) are depicted by blue and black arrows in Figures 12(a)–12(d). Figures 18(b) and 18(c) present two cross-sectional views of stitching plane p_3 toward the bottom ($Z-$) direction (i.e., p_3^{Z-} in (b)) and the top ($Z+$) direction (i.e., p_3^{Z+} in (c)), respectively. Grey blocks go through stitching plane p_3 along with Z axis. By using the relevant corner links information, it is possible to show that all of non-grey-color faces in Figures 18(b) and 18(c) have the same Z-coordinates. Furthermore, we can find the symmetric difference between the set of blocks in p_3^{Z-} and the set of blocks in p_3^{Z+}. Blocks b, d, e and h, j, i from two sets make the symmetric difference. We refer the two sets as $S-$ and $S+$, respectively. Figure 19 shows another example of two cross-sectional views from stitching plane p_k, highlighting the symmetric difference by purple blocks in

[5] An edge (n_i, n_j) is a *reduction edge* if there are no other paths from n_i to n_j except the edge (n_i, n_j) itself [50].

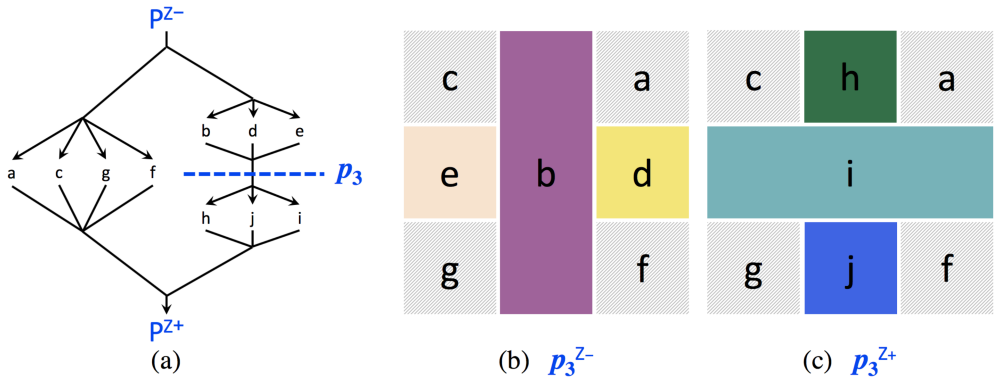

Fig. 18. (a) Stitching plane p_3 of the 3D floorplan example in Figure 10 and its Z-directional partial order in Figure 16(c). In Figure 12, stitching plane p_3-related edges are depicted by arrows. (b) and (c) are cross-sectional views of stitching plane p_3 toward $Z-$ (i.e., p_3^{Z-}) and $Z+$ (i.e., p_3^{Z+}) directions, respectively. Grey blocks go through stitching plane p_3 along with Z axis.

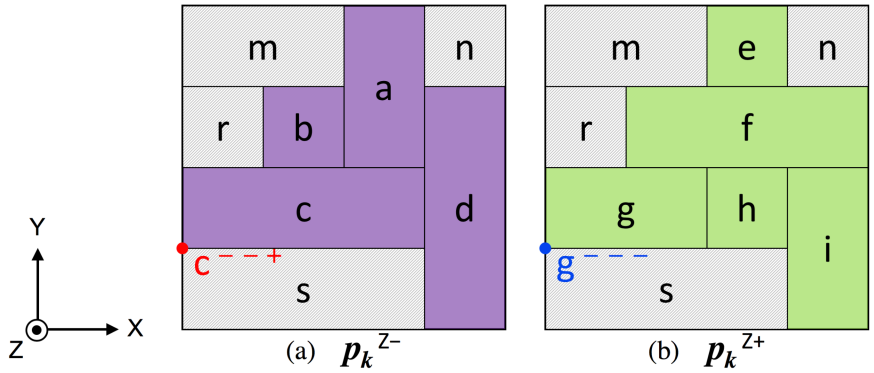

Fig. 19. The symmetric difference for footprints across stitching plane p_k. Grey blocks go through stitching plane p_k along with Z axis. By Theorem 4.3, corner c^{---+} must have a neighboring corner, i.e., g^{---} in this example. Note that $S-$ and $S+$ are sets of blocks in p_k^{Z-} and p_k^{Z+}, respectively.

p_k^{Z-} (i.e., (a)) and green blocks in p_k^{Z+} (i.e., (b)). Note that $S-$ and $S+$ are sets of blocks in p_k^{Z-} and p_k^{Z+}, respectively.

Theorem 4.3 (Restoring Partial Order from Corner Links). *For a 3D mosaic floorplan, the partial order representation can be restored from the corner links representation by enumerating all blocks in p_k^{j-} and p_k^{j+} for every **stitching plane** p_k in each dimension j. Stitching plane p_k is a plane characterized by the transitive closure of face equalities obtained from faces that touch a corner, in each of the three dimensions.*

We start with adjacent blocks that have a common surface (i.e., stitching plane p_k) perpendicular to dimension j. We consider one of these blocks, B_1, looking at T that is a set of all blocks that can be shown by using the corner links (thus the corner equations) and have a face along the stitching plane p_k. Note that if any block $B_2 \in T$ shares an 1/8 corner at a point along the stitching plane p_k with any other block B_3, then block B_3 should be in T. Also, note that set T can be partitioned into two subsets according as to whether a block in T is above or below stitching plane p_k in dimension

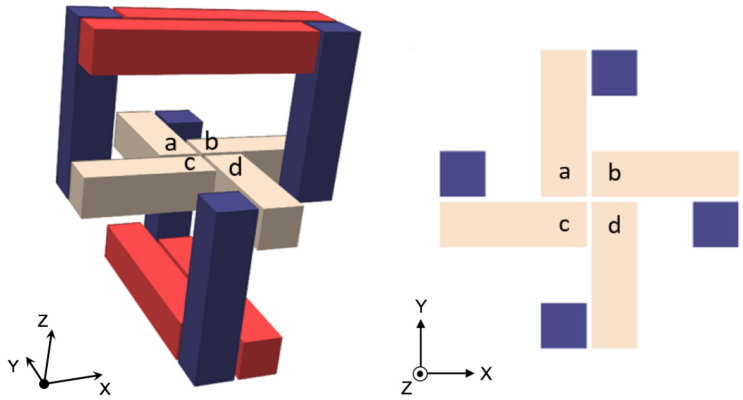

Fig. 20. A special example for the non-degenerate 3D floorplan, which has a corner link with more than one pair of neighboring corners. Blocks a, b, c, and d are intersecting at the same coordinate. However, we cannot shift the corners to split the intersecting point because of the alignments made by the blue and red blocks above and below blocks a, b, c, and d, thus non-degenerate.

j. The two subsets give two footprints E and F, from sets of blocks in p_k^{j-} and p_k^{j+}, respectively. To prove Theorem 4.3, it is enough to show the shapes of the two footprints are equal, i.e., $E = F$.

Proof. (by contradiction)

(1) Suppose that we have a 3D mosaic floorplan. We use corner links representation to identify the stitching plane p_k. T is a set of all blocks that can be shown by using the corner links and that have a face along the stitching plane p_k.
(2) Suppose that at least one block in the floorplan is not linked in the stitching plane p_k (i.e., the block is crossing the stitching plane p_k). And $S-$ and $S+$ are sets of blocks in p_k^{j-} and p_k^{j+} where j is perpendicular to the stitching plane p_k.
(3) Assume footprints of $S-$ and $S+$ are not equal, i.e., $E \neq F$. Then, we take a corner of the symmetric difference (e.g., a bottom-left corner c^{--+} as shown in Figure 19).
(4) The corner (e.g., c^{--+}) lies on the stitching plane p_k^{j-} and belongs to $S-$. And $S- \in T$.
(5) The corner is a 1/8 corner of an odd number of 1/8 corners in T as $E \neq F$ in (3).
(6) However, the 1/8 corner must be in a set of an even number of 1/8 corners by Lemma 4.2. Thus, there must be block B_i that has the corner (i.e., c^{--+}) as its neighboring corner.
(7) This would force $B_i \in T$ by (1), a contradiction. Therefore, $E = F$. □

4.3 Corner Links and Four Trees Representation

For a non-degenerate 3D mosaic floorplan, the corner links representation can be equivalently expressed by the four trees representation. Suppose that each corner has only one neighboring corner except at the corner of the 3D floorplan space P. Then the corner links representation forms trees by corner equations. Also, there are 3D floorplans with 1/8 corners having more than one pair of neighboring corners, implying that we may have multiple sets of four trees representations but capable of constructing four trees. Figure 20 shows a special example for the non-degenerate 3D floorplan with a corner having more than one pair of neighboring corners. Blocks a, b, c, and d are intersecting at the same coordinate. For 2D floorplan, when we have corners that have more than one neighboring corner (i.e., two segments are crossing against each other), we can slide one segment to split the crossing into two T-junctions. Then the floorplan becomes the mosaic floorplan, and its corresponding twin binary trees representation gives an exact one-to-one mapping

to the mosaic floorplan. However, the configuration of the 3D floorplan in Figure 20 indicates that adjusting block size is inapplicable due to the alignments of red and blue blocks above and below blocks a, b, c, and d, thus the 3D floorplan is non-degenerate.

Theorem 4.4 (Conjecture of Four Trees Representation to 3D Floorplan). *For 3D mosaic floorplan, four trees are sufficient for a non-degenerate floorplan representation.*

Proof.

(1) Each face of a block has two corner links on two opposite corners, determining the block's boundary.
(2) By Theorem 4.3, all faces on a stitching plane are linked together.
(3) Based on (1) and (2), the statement of Theorem 4.4 is true. □

More generally, we can conjecture that every 3D mosaic floorplan can be expressed by its corresponding four trees representation. Note that degenerate floorplans can be converted into non-degenerate floorplans after applying very tiny adjustment. Conversely, each four trees representation describes a single 3D mosaic floorplan, since both corner links representation and four trees representation are based on corner equations.

4.4 Partial Order Representation to Valid Floorplan

From the corner links, we derive an important property of the partial order representation. Based on Lemma 4.5, we show that the partial order representation can describe a valid 3D floorplan. Each corner link determines relations (either descendent or sibling) between every pair of blocks in the corner link. The partial order representation yields all stitching planes in the 3D mosaic floorplan, in sorted order by their respective dimensions. If the partial order representation describes relations between all pairs of blocks in the 3D floorplan, then the partial order representation produces a valid 3D floorplan.

Lemma 4.5 (Partial Order based on Corner Link in 3D Mosaic Floorplan). *Given a corner link between blocks B_1 and B_2 in a 3D mosaic floorplan, let the locations of the two neighboring corners be $B_1^{x_1y_1z_1}$ and $B_2^{x_2y_2z_2}$ where $x_1, x_2, y_1, y_2, z_1, z_2$, respectively, indicate the relative location of the corner as $-$ or $+$ for each dimension. If $x_1 \neq x_2$, then the two blocks B_1 and B_2 are in descendant relation in the partial order representation for X dimension. Otherwise, the two blocks are in sibling relation under the stitching plane on the $B_1^{x_1}$ side. Similarly, Y and Z dimensional relations are obtained.*

Proof. By the definition of the corner links and the corner equations, the statement of Lemma 4.5 is true for all 3D mosaic floorplans. □

For example, from the mosaic 3D floorplan in Figure 10 and the four trees representation in Figure 12, we have a corner link composed of two neighboring corners g^{-++} and j^{+++}, describing the relation between two blocks g and j in X dimension. By Lemma 4.5, block j is the parent of block g in the X-dimensional partial order. In the meantime, the two blocks are siblings under the stitching plane on the + sides (i.e., $g^{Y+} = j^{Y+}$ and $g^{Z+} = j^{Z+}$) in the Y- and Z-dimensional partial order representation (as shown in Figure 16). Based on Lemma 4.5, we derive Theorem 4.6.

Theorem 4.6 (Valid 3D Mosaic Floorplan and Partial Order Representation). *Given a valid and non-degenerate 3D mosaic floorplan, any pair of distinct blocks are related under at least one of the partial orders.*

Proof. (by enumeration)

(1) The proof is by enumeration on the number of non-overlapping coordinates of the two blocks, A and B.

(2) If the two blocks A and B overlap in two coordinates (e.g., assume X and Y dimensions in this proof), then tracing a straight line from one to the other will provide a chain of face relations between one and the other, proving comparability in that coordinate. Note that there are no blocks overlapping in three coordinates, since we consider valid 3D floorplans.

(3) If the two blocks A and B do not overlap in two coordinates (e.g., assume X and Y dimensions) but overlap in the remaining coordinate (e.g., assume Z dimension), then consider the intersection with a plane (e.g., XY plane) that covers both block A and block B. Assume (without loss of generality) that A's coordinates are larger than B's coordinates in each dimension, e.g., $A^{X-} > B^{X+}$, $A^{Y-} > B^{Y+}$. Then, we consider the intersecting XY plane as a 2D floorplan. We produce a chain of blocks $A_0,\ A_1,\ A_2, \ldots,\ A_n$ where $A = A_0$ and block A_{i+1} has an 1/4 corner that shares the corner with block A_i's $--$ corner (i.e., A_i^{--}). Note that such a corner must exist for every block A_i, since the corner A_i^{--} coincides with an odd number of other 1/4 corners (similarly, by Lemma 4.2). Continue this procedure until we reach block A_n that overlaps block B in at least one of X or Y coordinate, which must happen eventually.

(4) If the two blocks A and B do not overlap in any of three coordinates, then assume (without loss of generality) that A's coordinates are larger than B's coordinates in each dimension, e.g., $A^{X-} > B^{X+}$, $A^{Y-} > B^{Y+}$, and $A^{Z-} > B^{Z+}$. We produce a chain of blocks $A_0, A_1, A_2, \ldots, A_n$ where $A = A_0$ and block A_{i+1} has an 1/8 corner that shares the corner with block A_i's $---$ corner (i.e., A_i^{---}). Note that such a corner must exist for every block A_i, since the corner A_i^{---} coincides with an odd number of other 1/8 corners (by Lemma 4.2). Continue this procedure until we reach block A_n that overlaps block B in at least one of X, Y, or Z coordinate, which must happen eventually.

(5) Based on (2), (3), or (4), we obtain $A_0 \geq A_1 \geq A_2 \geq \ldots \geq A_n$ under at least one of dimensions. These inequalities hold for all the partial orders. Since $A = A_0$ and $A_n \geq B$, $A_0 \geq A_n \geq B$. Therefore, we have the relation $A \geq B$ under at least one of the partial order representation. □

4.5 Partial Order Representation to Blocks' Absolute Coordinates

A partial order representation can restore the absolute coordinates of all blocks in the 3D mosaic floorplan by using the given physical dimensions of blocks. Based on the relative locations of stitching planes per each dimension, the partial order of each dimension provides information on the relative orders of the blocks in the corresponding dimension. For direction d where d is one of X, Y, and Z, (1) we start from the minimum d-directional coordinate of the floorplan space P, i.e., P^{d-}. (2) We traverse and iterate through the stitching planes, incrementing a "layer counter" for each stitching plane. (3) We finish the procedure when we reach P^{d+}. (4) Then, we obtain the minimum and maximum coordinates of each block in d direction. Examples are shown in Figure 21 and Table 1 for the 3D mosaic floorplan in Figure 10. The relative orders for stitching planes are $x_0 < x_1 < x_2 < x_3$, $y_0 < y_1 < y_2 < y_3$, and $z_0 < z_1 < z_2$ according to the X-, Y-, and Z-directional partial orders, respectively. Suppose that block B's X-directional physical dimension is $B(X)$. Then, we have Equations (6) to obtain the absolute coordinates of the entire blocks in X direction. We assume $x_0 = y_0 = z_0 = 0$. Similarly, Y- and Z-directional absolute coordinates can be obtained. In Table 1, $Block^{d+}$ and $Block^{d-}$ denote $Block$'s d-directional maximum and minimum coordinates, respectively, where d is X, Y, or Z. Since the corner links representation is completely expressible for the given 3D mosaic floorplan and equivalently convertible to the 3D TCGs, all 3D floorplans represented by the corner links representation can obtain the absolute coordinates of the entire blocks in the floorplan. Therefore, the corner links representation provides sufficient information

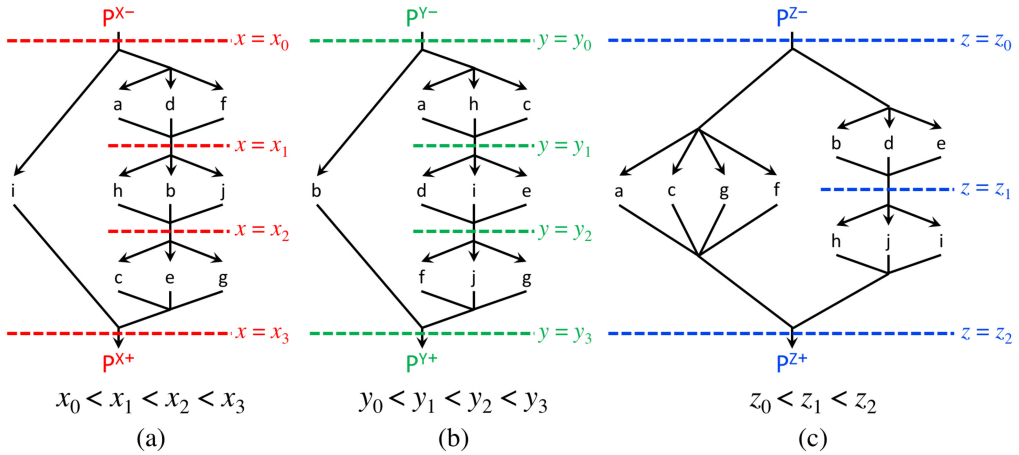

Fig. 21. Obtaining X-, Y-, and Z-dimensional relative orders of all blocks in the 3D floorplan based on the partial order representation in Figure 16.

Table 1. X-, Y-, and Z-directional Absolute Coordinates of All Blocks of the 3D Floorplan in Figure 10, Based on the Relative Orders of Stitching Planes in Figure 21

Blocks	X Coordinate		Y Coordinate		Z Coordinate	
	$Block^{X-}$	$Block^{X+}$	$Block^{Y-}$	$Block^{Y+}$	$Block^{Z-}$	$Block^{Z+}$
a	x_0	x_1	y_0	y_1	z_0	z_2
b	x_1	x_2	y_0	y_3	z_0	z_1
c	x_2	x_3	y_0	y_1	z_0	z_2
d	x_0	x_1	y_1	y_2	z_0	z_1
e	x_2	x_3	y_1	y_2	z_0	z_1
f	x_0	x_1	y_2	y_3	z_0	z_2
g	x_2	x_3	y_2	y_3	z_0	z_2
h	x_1	x_2	y_0	y_1	z_1	z_2
i	x_0	x_3	y_1	y_2	z_1	z_2
j	x_1	x_2	y_2	y_3	z_1	z_2

With the given physical dimensions of blocks and the 3D floorplan space P, we can obtain the entire coordinates as described in Equations (6).

to abstract the geometrical/topological structure of the 3D floorplan.

$$\begin{gathered} a(X) = d(X) = f(X);\ x_1 = a(X); \\ h(X) = b(X) = j(X);\ x_2 = x_1 + h(X); \\ c(X) = e(X) = g(X);\ x_3 = x_2 + c(X); \\ x_3 = i(X). \end{gathered} \tag{6}$$

4.6 Comparison with Other 3D Floorplan Representations

There are several 3D floorplan representation methodologies and shows different features and properties. In this section, we compare the corner links representation with several 3D floorplan representations. Table 2 presents comparisons of the corner links representations with various 3D floorplan representations. The entries for the other representations, e.g., Sequence Triple (ST) and Sequence Quintuple (SQ) [45], 3D slicing tree [7], 3D corner block list (3D-CBL) [32],

Table 2. Comparisons of the Corner Links Floorplan Representations with Several 3D Floorplan Representations

Representation	Completeness	Complexity of Floorplan Construction	Move Complexity	Floorplan Category	Solution Space
ST [45]	Incomplete	$O(m^2)$	$O(1)$	General but not all	$m!^3$
SQ [45]	Complete	$O(m^2)$	$O(1)$	All 3-D floorplans	$m!^5$
3D Slicing Tree [7]	Complete	$O(m)$	$O(1)$	All Slicing	$O(m!3^{m-1}2^{2m-2}/m^{1.5})$
3D-CBL [32]	Incomplete	$O(m)$	$O(1)$	Mosaic	$O(m!3^{m-1}2^{4m-4})$
3D-subTCG [50]	Complete	$O(m^2)$	$O(m^2)$	General but not all	$m!^3$
Corner Links	Complete	$O(m)$	$O(m^2)$	Mosaic	$O\left(m!^4\left(\frac{4^4}{3^3}\right)^{4m}/m^6\right)$

The entries of the corner links representation are our analysis, while the entries for the other representations, e.g., Sequence Triple (ST) and Sequence Quintuple (SQ) [45], 3D slicing tree [7], 3D corner block list (3D-CBL) [32], 3D-subTCG [50], are from Table 4.2 of Reference [10]. m is the number of blocks.

3D-subTCG [50], are referred from Table 4.2 of Reference [10]. ST and 3D-CBL cannot completely represent 3D floorplans [10], while the other methodologies including the corner links representation are able to represent all 3D floorplans in its targeting category. The complexities to assemble the original 3D floorplan from the given representation is $O(m)$ for 3D slicing tree, 3D-CBL, and the corner links, and $O(m^2)$ for ST, SQ, and 3D-subTCG, where m is the number of blocks in the 3D floorplan. The corner links representation has $4(m-1)$ edges, thus we can assemble the original floorplan by traversing all edges. The corner links representation and 3D-subTCG have the same time complexity for the perturbation operations, e.g., $O(m^2)$ for Move operation, because the corner links representation shares the properties of the operations defined 3D-subTCG after converting (as discussed in Section 4.2.2). The corner links representation covers the 3D mosaic floorplan category with 3D-CBL. As there are no empty spaces in a 3D mosaic floorplan, it provides much simpler thus more efficient structure to study by virtue of the smaller possible solution space. Also, the mosaic floorplan can be easily extended to the general floorplan by adding empty blocks with a controlled number of gaps (i.e., holes). Overall, the corner links representation provides (1) complete abstraction of the given 3D floorplan and (2) efficient to construct the original floorplan.

5 3D FLOORPLAN REPRESENTATION ALGORITHMS

In this section, we describe three algorithms: (i) to obtain the corner links representation from the given 3D mosaic floorplan (Algorithm 1 in Section 5.1), (ii) to convert corner links representation to partial order representation (Algorithm 2 in Section 5.2), and (iii) to restore the absolute coordinates of the entire blocks in the 3D floorplan by using the relative orders of stitching planes described by the partial order representation (Algorithm 3 in Section 5.3).

5.1 3D Mosaic Floorplan to Corner Links Representation

Algorithm 1 describes the procedure to obtain the corner links representation from the given 3D mosaic floorplan. Based on Theorem 4.1 and Lemma 4.2, we can obtain a complete set of all the corner links in a 3D mosaic floorplan consisting of a set of blocks described with eight corners' coordinates for each block. Starting with sorting, every 1/8 corner of each block searches adjacent blocks that have 1/8 neighboring corners intersecting at the same coordinate, indicating that Algorithm 1 performs in $O(m \log m)$ where m is the number of blocks. As discussed in the previous Sections 3.1 and 4.1, we consider 1/8 neighboring corners of the Hamming distance 1 or 3 only

ALGORITHM 1: 3D Mosaic Floorplan to Corner Links Representation

Input: A 3D floorplan as a set of the entire blocks, composed of eight corners' coordinates per each block
Output: A complete set of the corner links in the given 3D floorplan

Procedure getCornerLinks (sBlock)

/* Initialization */
$sCorners \leftarrow \{---, -++, +-+, ++-, --+, -+-, +--, +++\}$;
$sOppositeCorners \leftarrow$ either $\{---, -++, +-+, ++-\}$ or $\{--+, -+-, +--, +++\}$; // linked corners
$sLinkingCorners \leftarrow \{sCorners \setminus sOppositeCorners\}$; // linking corners
CornerLinks.clear();

/* Sort the set of all blocks in the given 3D mosaic floorplan */
$sSortedBlock \leftarrow \emptyset$;
for *each dimension* $d \in \{X, Y, Z\}$ **do**
  $sSortedBlock \leftarrow$ sortBlock($sBlock$, d);
end

/* Find neighboring corners for each block */
for *each block* $A_i \in sSortedBlock$ **do**
  for *each* 1/8 *corner* v *of* A_i *where* $v \in sLinkingCorners$ **do**
    /* Find 1/8 neighboring corners of the Hamming distance = 1 */
    for *each block* $A_j \in sSortedBlock$ *where* $i \neq j$ **do**
      for *each* 1/8 *corner* w *of* A_j *where* $w \in sOppositeCorners$ **do**
        if *v.coordi == w.coordi && getHamDist(v,w) == 1 && cornerLink (v,w)* $\notin$ *CornerLinks* **then**
          CornerLinks.add (v,w);
        end
      end
    end
    /* Find 1/8 neighboring corners of the Hamming distance = 3 */
    for *each block* $A_j \in sSortedBlock$ *where* $i \neq j$ **do**
      for *each* 1/8 *corner* w *of* A_j *where* $w \in sOppositeCorners$ **do**
        if *v.coordi == w.coordi && getHamDist(v,w) == 3 && cornerLink (v,w)* $\notin$ *CornerLinks* **then**
          CornerLinks.add (v,w);
        end
      end
    end
  end
end

Procedure getHamDist (v, w)

$distHamming \leftarrow 0$;
for *each dimension* d *of corner* v *where* $d \in \{X, Y, Z\}$ **do**
  if $v.d \neq w.d$ **then**
    $distHamming$ + +;
  end
end
return $distHamming$;

ALGORITHM 2: Corner Links Representation to Partial Order Representation

Input: Complete set of corner links in the 3D floorplan, composed of a set of equivalent block corners
Output: Three partial orders for each dimension X, Y, and Z (i.e., three lists of stitching planes that are ordered to each dimension)

Procedure getPartialOrders (CornerLinks)

/* Note: ith cornerLink $cL_i \in CornerLinks$ is composed of a set of neighboring corners nC_j,
which includes all pairs of 1/8 corners in the adjacent blocks that intersect at the same coordinate.
jth neighboring corner $nC_j \in cL_i$ is defined as $A^{xyz} = B^{xyz}$
where A and B denote blocks and x, y, $z$indicate the relative location of the corner as + or −. */

for *each dimension* $d \in \{X, Y, Z\}$ **do**
  $sStitchingPlanes(d) \leftarrow \emptyset$;
end
for *each cornerLink* $cL_i \in CornerLinks$ **do**
  for *each neighboringCorners* $nC_j \in cL_i$ **do**
    for *each dimension* $d \in \{X, Y, Z\}$ *of the corner equation of* nC_j **do**
      Obtain d-dimensional equivalent corner relation, i.e., stitching plane $cP_i(d)$;
      $sStitchingPlanes(d) \leftarrow sStitchingPlanes(d) \cup$ stitchingPlane $cP_i(d)$;
    end
  end
end
for *each* $sStitchingPlanes(d)$ *where* $d \in \{X, Y, Z\}$ **do**
  $partialOrder(d) \leftarrow$ acyclicDirectedGraph($sStitchingPlanes(d)$);
end

as described in Algorithm 1, since 1/8 neighboring corners of the Hamming distance 2 should be covered after we take 1/8 neighboring corners of the Hamming distance 1. With this in mind, Algorithm 1 searches neighboring corners of the Hamming distance 1 first, then searches 3. Algorithm 1 ends up with a complete set of the corner links that uniquely describe the topological structure of the original floorplan, resulting in $4(m-1)$ corner links.

5.2 Corner Links Representation to Partial Order Representation

Algorithm 2 describes a procedure to convert the corner links representation to the partial order representation. Based on Theorem 4.3, a complete list of the corner links in the 3D floorplan can be reduced to the three partial orders for each dimension of the 3D space by using stitching planes. From the inputs of the complete set of corner links in floorplan, Algorithm 2 develops three partial orders for each dimension X, Y, and Z as its outputs. The resulting partial orders are acyclic directed graphs, i.e., topologically ordered three transitive closure graphs per each dimension.

Note that corner links are composed of a set of neighboring corners, and each corner link has at least two and the even number of 1/8 corners as shown in Lemma 4.2, i.e., we have at least a pair of neighboring corners in the corner link. Also, all 1/8 neighboring corners are pair-wise equal by the corner equations. In Algorithm 2, ith corner link cL_i in the set of the corner links $CornerLinks$ is composed of a set of neighboring corners nC_j, which includes all pairs of 1/8 corners in the adjacent blocks that intersect at the same coordinate. And jth neighboring corner nC_j is defined with corner equation as $A^{xyz} = B^{xyz}$ where A and B denote blocks determining nC_j and x, y, z indicate the relative location of the corner within the blocks as + or −. Through the first FOR-loop, we initialize sets of stitching planes per each dimension, $sStitchingPlanes(d)$, as empty sets. We then proceed the second FOR-loop for each corner link cL_i in a set $CornerLinks$ that includes the entire corner

ALGORITHM 3: Partial Order Representation to Absolute Coordinate

Input: Three partial orders for each dimension X, Y, and Z
Output: Absolute coordinates of $---$ and $+++$ corners of each block

Procedure getCoordinateForAllBlocks (partialOrders)

for *each* $partialOrder(d) \in partialOrders$ **do**
  /* i.e., for each dimension d where $d \in \{X, Y, Z\}$ */
  initialPlane $iP \leftarrow$ floorplanSpace.$d-$;
  $iP.d-$.coordinate $\leftarrow 0$;
  $sBlock \leftarrow \{A_i \mid$ blocks that are connectedfrom iP plane in $partialOrder(d)\}$;
  for *each block* $A_i \in sBlock$ **do**
    /* Block A_i is ith block of $sBlock$ */
    $A_i.d-$.coordinate $\leftarrow iP.d-$.coordinate;
  end
  while $sBlock \neq \emptyset$ **do**
    getPlusCoordinate ($sBlock$);
    updateSetBlock (sBlock);
  end
end

Procedure getPlusCoordinate (sBlock)

for *each block* $A_i \in sBlock$ **do**
  $A_i.d+$.coordinate $\leftarrow A_i.d-$.coordinate + $A_i.d$.dimension;
end

Procedure updateSetBlock (sBlock)

for *each block* $A_i \in sBlock$ **do**
  if *($A_i.d+$ plane touches floorplanSpace.$d+$)* **then**
    $sBlock \leftarrow sBlock \setminus A_i$;
    continue;
  end
  $sNewBlock \leftarrow \{B_j \mid$ blocks that are faced with $A_i.d+$ plane in $partialOrder(d)\}$;
  for *each block* $B_j \in sNewBlock$ **do**
    $B_j.d-$.coordinate $\leftarrow A_i.d+$.coordinate;
  end
  $sBlock \leftarrow sBlock \cup sNewBlock$;
  $sBlock \leftarrow sBlock \setminus A_i$;
end

links of the given 3D floorplan. For each corner link cL_i, we traverse every neighboring corner nC_j in cL_i and obtain each dimensional equivalent corner relation from the corner equation of nC_j. This equivalent corner relation determines the stitching plane $cP_{ij}(d)$, which will be added into the set of stitching planes for each dimension d. After traversing all corner links through the second FOR-loop, we have complete sets of stitching planes for each dimension. As every stitching plane defines the relative location of blocks, we can generate each dimensional partial order $partialOrder(d)$ by obtaining the acyclic directed graph of stitching planes in $sStitchingPlanes(d)$. Traversing the entire corner links yields the partial order representation, showing $O(m)$ time complexity where m is the number of blocks.

5.3 Partial Order Representation to Absolute Coordinates

Algorithm 3 describes the procedure to restore the absolute coordinates of all blocks in the 3D floorplan from the partial order representation. As discussed in Section 4.4, the partial order representation can restore the absolute coordinates of all blocks in the 3D floorplan, since the partial orders give the topological-relative orders of all blocks for each dimension. For each dimensional partial order $partialOrder(d)$, Algorithm 3 starts with the initialPlane iP as the minimum plane of the floorplan space floorplanSpace.$d-$, i.e., the root node of each dimensional partial order. We define the absolute coordinates of these root nodes as 0. In Algorithm 3, $sBlock$ is a set of blocks to be considered in each WHILE-loop iteration, and is initialized as a set of blocks that are connected from iP plane in $partialOrder(d)$. For each block $A_i \in sBlock$, we determine the minimum coordinate of block A_i (i.e., $A_i.d-$.coordinate) as the minimum coordinate of the 3D floorplan space (i.e., $iP.d-$.coordinate). We execute WHILE-loop until $sBlock$ becomes empty set. WHILE-loop has two subprocedures. Procedure **getPlusCoordinate (sBlock)** obtains the maximum coordinates of each block A_i in $sBlock$, which adds A_i's d dimension to block A_i's minimum coordinate. Then Procedure **updateSetBlock (sBlock)** is performed for each block $A_i \in sBlock$. The procedure first removes block A_i from $sBlock$ if block A_i touches the maximum coordinate of the given 3D floorplan. Otherwise, the procedure separately creates a set of blocks $sNewBlock$ containing all blocks faced with the maximum coordinate of block A_i. For each block $B_j \in sNewBlock$, we define the minimum coordinates of block B_j as the maximum coordinate of block A_i. Then the procedure adds all blocks in $sNewBlock$ into $sBlock$, and removes block A_i from $sBlock$. We obtain the absolute coordinates of all blocks when we terminate WHILE-loop. Traversing the entire blocks in three transitive closure graphs gives sufficient information to restore the absolute coordinates of the entire blocks in the original 3D floorplan, showing $O(m)$ time complexity where m is the number of blocks.

6 CONCLUSION

In this article, we have presented our new 3D floorplan representation, the *corner links representation*. We define the corner links representation as the spatial relationships describing all corner relations of the entire blocks in the 3D mosaic floorplan. A *corner link* as a set of 1/8 corners that intersect at the same coordinate and belong to the adjacent blocks next to each other in X, Y, Z, or diagonal direction. A corner link consists of pair(s) of neighboring corners with odd number of the Hamming distance. We have analyzed the corner links representation's key properties with lemmas, theorems, and their proofs along with the corresponding *partial order representation*, i.e., three transitive closure graphs (TCGs) for each dimension. We have demonstrated the followings: (1) The corner links is a complete representation for 3D floorplan. (2) The corner links representation can be reduced to the partial order representation. (3) A non-degenerate 3D mosaic floorplan can be equivalently expressed by the corresponding four trees representation. (4) A partial order representation with three transitive closure graphs captures all stitching planes in the 3D mosaic floorplan, in order of their respective dimensions. (5) The 3D floorplan is a valid floorplan if the partial order representation describes relations between all pairs of blocks in the 3D mosaic floorplan. (6) The partial order representation can restore the absolute coordinates of all blocks in the 3D mosaic floorplan by using the given physical dimensions of blocks. We leave to our future work a couple of directions: (i) more encoding schemes for the corner links representation and four trees representation (ii) further reduce the total number of combinations while preserving the completeness of the encoding, and (iii) extension of our representation methodology to four (4D) or even higher dimensions for the mapping of dynamic re-programmable 3D devices, the thermal/power management minimizing the peak temperature, and so on.

REFERENCES

[1] G. Baxter. 1964. On fixed points of the composite of commuting functions. *Proc. Amer. Math. Soc.* 15, 6 (1964), 851–855.

[2] K. Bernstein, P. Andry, J. Cann, P. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, R. Puri, and A. Young. 2007. Interconnects in the third dimension: Design challenges for 3D ICs. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'07)*. 562–567.

[3] H. H. Chan, S. N. Adya, and I. L. Markov. 2005. Are floorplan representations important in digital design? In *Proceedings of the ACM/IEEE International Symposium on Physical Design (ISPD'05)*. 129–136.

[4] W.-T. J. Chan, Y. Du, A. B. Kahng, S. Nath, and K. Samadi. 2015. 3D-IC benefit estimation and implementation guidance from 2D-IC implementation. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'15)*. 1–6.

[5] K. Chang, K. Acharya, S. Sinha, B. Cline, G. Yeric, and S. K. Lim. 2017. Impact and design guideline of monolithic 3D IC at the 7-nm technology node. *IEEE Trans. Very Large Scale Integr. Syst.* 25, 7 (2017), 2118–2129.

[6] C.-K. Cheng, P. Du, A. B. Kahng, and S.-H. Weng. 2012. Low-power gated bus synthesis for 3D IC via rectilinear shortest-path steiner graph. In *Proceedings of the ACM/IEEE International Symposium on Physical Design (ISPD'12)*. 105–112.

[7] L. Cheng, L. Deng, and M. D. F. Wong. 2005. Floorplanning for 3D VLSI design. In *Proceedings of the ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC'05)*. 405–411.

[8] F. R. K Chung, R. L. Graham, V. E. Hoggatt Jr., and M. Kleiman. 1978. The number of Baxter permutations. *J. Combinator. Theory, Ser. A* 24, 3 (1978), 382–394.

[9] J. Cong, G. Luo, J. Wei, and Y. Zhang. 2007. Thermal-aware 3D IC placement via transformation. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'07)*. 780–785.

[10] J. Cong and Y. Ma. 2010. Thermal-aware 3D floorplan. In *Three-Dimensional Integrated Circuit Design—EDA, Design and Microarchitectures*, Y. Xie, J. Cong, and S. Sapatnekar (Eds.). Springer, New York, NY, Chap. 4, 63–102.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. 2009. *Introduction to Algorithms* (3rd ed.). The MIT Press, Cambridge, MA.

[12] S. Dulucq and O. Guibert. 1998. Baxter permutations 1. *Discrete Math.* 180, 1 (1998).

[13] J. A. Fill and R. P. Dobrow. 1997. The number of m-ary search trees on n keys. *Combinator. Probabil. Comput.* 6, 4 (1997), 435–453.

[14] R. Fischbach, J. Lienig, and M. Thiele. 2010. Solution space investigation and comparison of modern data structures for heterogeneous 3D designs. In *Proceedings of the IEEE International Conference on 3D System Integration (3DIC'10)*. 1–8.

[15] K. Fujiyoshi, H. Kawai, and K. Ishihara. 2009. A tree-based novel representation for 3D-block packing. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 28, 5 (2009), 759–764.

[16] B. Goplen and S. S. Sapatnekar. 2007. Placement of 3D ICs with thermal and interlayer via considerations. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'07)*. 626–631.

[17] P. Hilton and J. Pedersen. 1991. Catalan numbers, their generalization, and their uses. *Math. Intell.* 13, 2 (1991), 64–75.

[18] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C.-K. Cheng, and J. Gu. 2000. Corner block list: An effective and efficient topological representation of non-slicing floorplan. In *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD'00)*. 8–12.

[19] X. Hu, P. Du, J. F. Buckwalter, and C.-K. Cheng. 2013. Modeling and analysis of power distribution networks in 3D ICs. *IEEE Trans. Very Large Scale Integr. Syst.* 21, 2 (2013), 354–366.

[20] G. Huang, M. Bakir, A. Naeemi, H. Chen, and J. D. Meindl. 2007. Power delivery for 3D chip stacks: Physical modeling and design implication. In *Proceedings of the IEEE Electrical Performance of Electronic Packaging and Systems (EPEPS'07)*. 205–208.

[21] Hybrid Memory Cube. 2016. Retrieved from http://www.hybridmemorycube.org/.

[22] M. Jung, T. Song, Y. Wan, Y.-J. Lee, D. Mohapatra, H. Wang, G. Taylor, D. Jariwala, V. Pitchumani, P. Morrow, C. Webb, P. Fischer, and S. K. Lim. 2013. How to reduce power in 3D IC designs: A case study with OpenSPARC T2 core. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC'13)*. 1–4.

[23] A. K. Khan, R. Vatsa, S. Roy, and B. Das. 2014. A new efficient topological structure for floorplanning in 3D VLSI physical design. In *Proceedings of the IEEE International Advance Computing Conference (IACC'14)*. 696–701.

[24] D. H. Kim and S. K. Lim. 2015. Physical design and CAD tools for 3D integrated circuits: Challenges and opportunities. *IEEE Design Test Comput.* 32, 4 (2015), 8–22.

[25] D. A. Klarner. 1970. Correspondences between plane trees and binary sequences. *J. Combinator. Theory* 9, 4 (1970), 401–411.

[26] J. Knechtel and J. Lienig. 2016. Physical design automation for 3D chip stacks—Challenges and solutions. In *Proceedings of the ACM/IEEE International Symposium on Physical Design (ISPD'16)*. 3–10.

[27] J. U. Knickerbocker, P. S. Andry, B. Dang, R. R. Horton, C. S. Patel, R. J. Polastre, K. Sakuma, E. S. Sprogis, C. K. Tsang, B. C. Webb, and S. L. Wright. 2008. 3D silicon integration. In *Proceedings of the IEEE Electronic Components and Technology Conference*. 538–543.

[28] E. Lawler. 1976. *Combinatorial Optimization: Networks and Matroids* (1st ed.). Holt, Rinehart and Winston, New York, NY.
[29] J.-M. Lin and Y.-W. Chang. 2001. TCG: A transitive closure graph-based representation for non-slicing floorplans. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'01)*. 764–769.
[30] S. Lin and D. J. Costello, Jr. 2004. *Error Control Coding* (2nd ed.). Pearson, Upper Saddle River, NJ.
[31] J. Lu, H. Zhuang, I. Kang, and C.-K. Cheng. 2016. ePlace-3D: Electrostatics-based placement for 3D-ICs. In *Proceedings of the ACM/IEEE International Symposium on Physical Design (ISPD'16)*. 11–18.
[32] Y. Ma, X. Hong, S. Dong, and C.-K. Cheng. 2005. 3D CBL: An efficient algorithm for general 3D packing problems. In *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'05)*. 1079–1082.
[33] J. K. Ousterhout. 1984. Corner stitching: A data-structuring technique for VLSI layout tools. *IEEE Trans. Comput. Aided Design Integr. Circ. Syst.* 3, 1 (1984), 87–100.
[34] J. K. Ousterhout, G. T. Hamachi, R. N. Mayo, W. S. Scott, and G. S. Taylor. 1985. The magic VLSI layout system. *IEEE Design Test Comput.* 2, 1 (1985), 19–30.
[35] S. Panth, K. Samadi, Y. Du, and S. K. Lim. 2008. Design and CAD methodologies for low power gate-level monolithic 3D ICs. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED'08)*. 171–176.
[36] S. S. Sapatnekar. 2009. Addressing thermal and power delivery bottlenecks in 3D circuits. In *Proceedings of the ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC'09)*. 423–428.
[37] A. Shayan, X. Hu, H. Peng, C.-K. Cheng, W. Yu, M. Popovich, T. Toms, and X. Chen. 2009. Reliability aware through silicon via planning for 3D stacked ICs. In *Proceedings of the ACM/IEEE Design, Automation and Test in Europe (DATE'09)*. 288–291.
[38] T. Song, A. Nieuwoudt, Y. S. Yu, and S. K. Lim. 2015. Coupling capacitance in face-to-face (F2F) bonded 3D ICs: Trends and implications. In *Proceedings of the IEEE Electronic Components and Technology Conference*. 529–536.
[39] M. M. Waldrop. 2016. The chips are down for Moore's law. *Nature* 530, 7589 (2016), 144–147.
[40] R. Wang, E. F. Y. Young, and C.-K. Cheng. 2009. Representing topological structures for 3D floorplanning. In *Proceedings of the IEEE International Conference of Communications, Circuits and Systems (ICCCAS'09)*. 1098–1102.
[41] R. Wang, E. F. Y. Young, and C.-K. Cheng. 2010. Complexity of 3D floorplans by analysis of graph cuboidal dual hardness. *ACM Trans. Design Automat. Electron. Syst.* 15, 4 (2010), article 33.
[42] R. Wang, E. F. Y. Young, Y. Zhu, F. C. Graham, R. Graham, and C.-K. Cheng. 2008. 3D floorplanning using labeled tree and dual sequences. In *Proceedings of the ACM/IEEE International Symposium on Physical Design (ISPD'08)*. 54–59.
[43] R. Widialaksono, R. B. R. Chowdhury, Z. Zhang, J. Schabel, S. Lipa, E. Rotenberg, R. Davis, and P. Franzon. 2016. Physical design of a 3D-stacked heterogeneous multi-core processor. In *Proceedings of the IEEE International Conference on 3D System Integration (3DIC'16)*. 1–5.
[44] G.-M. Wu, J.-M. Lin, and Y.-W. Chang. 2001. An algorithm for dynamically reconfigurable FPGA placement. In *>Proceedings of IEEE International Conference on Computer Design (ICCD'01)*. 501–504.
[45] H. Yamazaki, K. Sakanushi, S. Nakatake, and Y. Kajitani. 2000. The 3D-packing by meta data structure and packing heuristics. *IEICE Trans. Fund. Electron. Commun. Comput. Sci.* 83, 4 (2000), 639–645.
[46] B. Yao, H. Chen, C.-K. Cheng, and R. Graham. 2003. Floorplan representations: Complexity and connections. *ACM Trans. Design Automat. Electron. Syst.* 8, 1 (2003), 55–80.
[47] Yole Développement. 2015. Equipment & Materials for 3DIC & WLP Applications. Retrieved from http://www.yole.fr/2014-galery-3D.aspx#I000350e1.
[48] E. F. Y. Young, C. C. N. Chu, and Z. C. Shen. 2003. Twin binary sequences: A nonredundant representation for general nonslicing floorplan. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 22, 4 (2003), 457–469.
[49] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. 2004. Temporal floorplanning using the T-tree formulation. In *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD'04)*. 300–305.
[50] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. 2007. Temporal floorplanning using the three-dimensional transitive closure subgraph. *ACM Trans. Design Automat. Electron. Syst.* 12, 4 (2007), article 37.
[51] L. Zhang, S. Dong, X. Hong, and Y. Ma. 2007. A fast 3D-BSG algorithm for 3D packing problem. In *Proceedings of the IEEE International Symposium of Circuits and Systems (ISCAS'07)*. 2044–2047.
[52] W. Zhang, W. Yu, X. Hu, A. Shayan, A. E. Engin, and C.-K. Cheng. 2009. Predicting the worst-case voltage violation in a 3D power network. In *Proceedings of the ACM/IEEE System Level Interconnect Prediction (SLIP'09)*. 93–98.
[53] H. Zhou and J. Wang. 2004. ACG-adjacent constraint graph for general floorplans. In *Proceedings of IEEE International Conference on Computer Design (ICCD'04)*. 572–575.

Received April 2018; revised September 2018; accepted October 2018