Ultrafast Density Gradient Accumulation in 3D Analytical Placement with Divergence Theorem

Peiyu Liao*, Yuxuan Zhao*, Siting Liu, Bei Yu
{pyliao,byu}@cse.cuhk.edu.hk

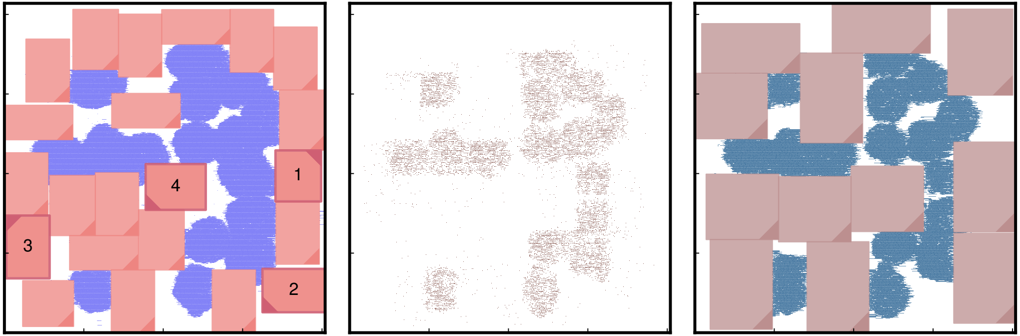
Department of Computer Science and Engineering
The Chinese University of Hong Kong

October 28, 2025

Face-to-Face bonded 3D ICs: How We Place Them

Top die, bottom die, and hybrid bonding terminals (HBTs).

- ➔ Enables direct metal-to-metal and dielectric-to-dielectric bonding.
- ➔ Achieve ultra-high vertical integration density.

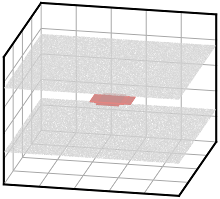


- An example layout of `ICCAD'23 case3h` from [TCAD'24][1].

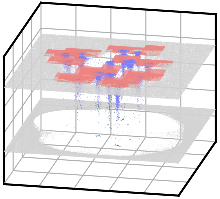
[1] Yuxuan Zhao et al. "Analytical Heterogeneous Die-to-Die 3D Placement with Macros". In: *IEEE TCAD* 44.2 (2024), pp. 402–415.

Analytical 3D Placement

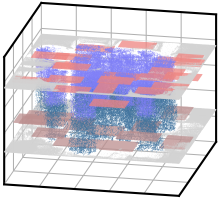
True 3D placers[2][3][4][5] build upon the methodology of ePlace-3D[6] to model the 3D density.



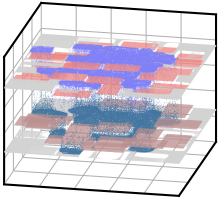
Iter.	WL (10^7)	#HBTs	OVFL
0	3.60	88507	1.00



Iter.	WL (10^7)	#HBTs	OVFL
1000	4.49	1254	0.89



Iter.	WL (10^7)	#HBTs	OVFL
1800	10.30	30961	0.34



Iter.	WL (10^7)	#HBTs	OVFL
2142	10.51	13798	0.07

[2] Yan-Jen Chen et al. "Late breaking results: Analytical placement for 3D ICs with multiple manufacturing technologies". In: *Proc. DAC*. IEEE. 2023, pp. 1–2.

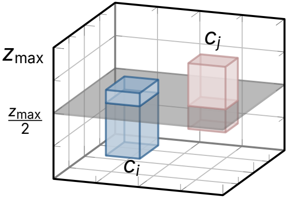
[3] Peiyu Liao et al. "Analytical Die-to-Die 3D Placement with Bistratal Wirelength Model and GPU Acceleration". In: *IEEE TCAD* 43.6 (2023), pp. 1624–1637.

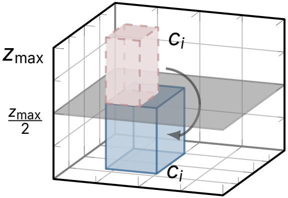
[4] Yan-Jen Chen et al. "Mixed-size 3D analytical placement with heterogeneous technology nodes". In: *Proc. DAC*. 2024, pp. 1–6.

[5] Yuxuan Zhao et al. "Analytical Heterogeneous Die-to-Die 3D Placement with Macros". In: *IEEE TCAD* 44.2 (2024), pp. 402–415.

[6] Jingwei Lu et al. "ePlace-3D: Electrostatics based placement for 3D-ICs". In: *Proc. ISPD*. 2016, pp. 11–18.

Analytical 3D Placement

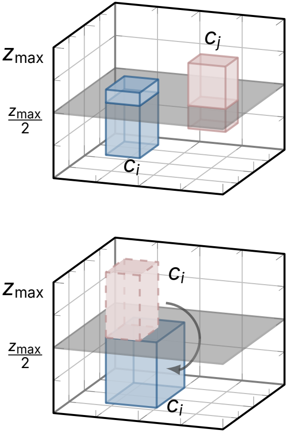


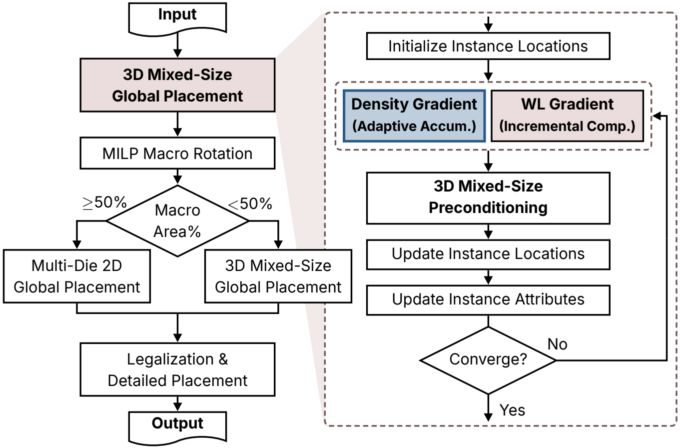


Every node has a **depth**.

[7] Yuxuan Zhao et al. "Analytical Heterogeneous Die-to-Die 3D Placement with Macros". In: *IEEE TCAD* 44.2 (2024), pp. 402–415

Analytical 3D Placement





Every node has a **depth**.

➔ The overall flow of [TCAD'24] [7].

[7] Yuxuan Zhao et al. "Analytical Heterogeneous Die-to-Die 3D Placement with Macros". In: *IEEE TCAD* 44.2 (2024), pp. 402–415

Density Gradient Accumulation

➔ What is density gradient accumulation?

Generally, it is a process to calculate the gradient of a node by accumulating gradients on all its covered bins.

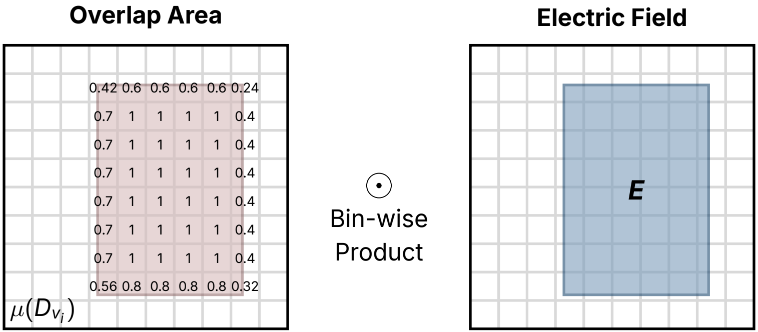
$$\frac{\partial U}{\partial x_i} \approx \sum_{b \in B(D_{v_i})} -\mu(D_{v_i} \cap b) E_x(b),$$

Density Gradient Accumulation

➔ What is density gradient accumulation?

Generally, it is a process to calculate the gradient of a node by accumulating gradients on all its covered bins.

$$\frac{\partial U}{\partial x_i} \approx \sum_{b \in B(D_{v_i})} -\mu(D_{v_i} \cap b) E_x(b),$$



➔ Vanilla Algorithm: Compute gradients for macros in 3D placement (in every iteration) is rather more expensive!

Observation: Divergence Theorem

Any Better Idea?

Observation: Divergence Theorem

Any Better Idea?

Consider the fundamental theorem of calculus:

$$\int_a^b f'(t)\mathrm{d}t = f(b) - f(a).$$

What happens inside can be understood by looking at the boundary.

Observation: Divergence Theorem

Any Better Idea?

Consider the fundamental theorem of calculus:

$$\int_a^b f'(t)\mathrm{d}t = f(b) - f(a).$$

What happens inside can be understood by looking at the boundary.

Divergence Theorem.

$$\iiint_\Omega \nabla \cdot \mathbf{F}\, \mathrm{d}\Omega = \oiint_{\partial\Omega} \mathbf{F} \cdot \hat{\mathbf{n}}\, \mathrm{d}S.$$

Total divergence inside a region is the total outward flux across the boundary

Observation: Divergence Theorem

Any Better Idea?

Consider the fundamental theorem of calculus:

$$\int_a^b f'(t)\mathrm{d}t = f(b) - f(a).$$

What happens inside can be understood by looking at the boundary.

Divergence Theorem.

$$\iiint_\Omega \nabla \cdot \mathbf{F}\, \mathrm{d}\Omega = \oiint_{\partial\Omega} \mathbf{F} \cdot \hat{\mathbf{n}}\, \mathrm{d}S.$$

Total divergence inside a region is the total outward flux across the boundary ➔ **Dimension Reduction?**

Apply Divergence Theorem

💡 How to apply divergence theorem to density gradient computation?

Apply Divergence Theorem

How to apply divergence theorem to density gradient computation?

Gradient Theorem

The **Gradient Theorem** between electric potential ϕ and electric field $\boldsymbol{E}$ states that $-\boldsymbol{E} = \nabla\phi$.

Apply Divergence Theorem

How to apply divergence theorem to density gradient computation?

Gradient Theorem

The **Gradient Theorem** between electric potential ϕ and electric field $\boldsymbol{E}$ states that $-\boldsymbol{E} = \nabla\phi$.

The following is then intuitive, with $\mathbf{F}_x := (\phi, 0, 0)$.

$$\iiint_{D_{v_i}} E_x \, \mathrm{d}D_{v_i} = \iiint_{D_{v_i}} \nabla \cdot \mathbf{F}_x \, \mathrm{d}D_{v_i} = \oiint_{\partial D_{v_i}} \hat{n}_x \phi \, \mathrm{d}S,$$

where $\hat{n}_x$ is the x component of normal vector $\hat{\boldsymbol{n}}$.

Apply Divergence Theorem

How to apply divergence theorem to density gradient computation?

Gradient Theorem

The **Gradient Theorem** between electric potential ϕ and electric field $\boldsymbol{E}$ states that $-\boldsymbol{E} = \nabla\phi$.

The following is then intuitive, with $\mathbf{F}_x := (\phi, 0, 0)$.

$$\iiint_{D_{v_i}} E_x \, \mathrm{d}D_{v_i} = \iiint_{D_{v_i}} \nabla \cdot \mathbf{F}_x \, \mathrm{d}D_{v_i} = \oiint_{\partial D_{v_i}} \hat{n}_x \phi \, \mathrm{d}S,$$

where $\hat{n}_x$ is the x component of normal vector $\hat{\boldsymbol{n}}$.

It is a dimension reduction technique if ϕ is easier to compute!

Is ϕ easier to compute?

Apply Divergence Theorem

Forward Pass

$$\rho(x, y, z) = \sum_{i=1}^{n} \chi_{D_{v_i}}(x, y, z)$$

Density Map

$$a_{jkl} = \frac{1}{|B|} \sum_{x,y,z} \rho \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Coefficients: one DCT_3D required.

$$\phi = \sum_{j,k,l} \frac{a_{jkl}}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Potential Map: one IDCT_3D required.

$$E_x = \sum_{j,k,l} \frac{a_{jkl}\omega_j}{\omega_j^2 + \omega_k^2 + \omega_l^2} \sin(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Field Maps: 3 IDCT/IDXST combinations required.

$$g_i = \iiint_{D_{v_i}} E_x \, \mathrm{d}D_{v_i}$$

Density Gradient Accumulation

Backward Pass

Apply Divergence Theorem

Forward Pass

$$\rho(x, y, z) = \sum_{i=1}^{n} \chi_{D_{v_i}}(x, y, z)$$

Density Map

$$a_{jkl} = \frac{1}{|B|} \sum_{x,y,z} \rho \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Coefficients: one DCT_3D required.

$$\phi = \sum_{j,k,l} \frac{a_{jkl}}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Potential Map: one IDCT_3D required.

$$E_x = \sum_{j,k,l} \frac{a_{jkl}\omega_j}{\omega_j^2 + \omega_k^2 + \omega_l^2} \sin(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Field Maps: 3 IDCT/IDXST combinations required.

$$g_i = \oiint_{\partial D_{v_i}} \hat{n}_x \phi \, dS$$

Density Gradient Accumulation

Backward Pass

Apply Divergence Theorem

Forward Pass

$$\rho(x, y, z) = \sum_{i=1}^{n} \chi_{D_{v_i}}(x, y, z)$$

Density Map

$$a_{jkl} = \frac{1}{|B|} \sum_{x,y,z} \rho \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Coefficients: one `DCT_3D` required.

$$\phi = \sum_{j,k,l} \frac{a_{jkl}}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Potential Map: one `IDCT_3D` required.

$$g_i = \oiint_{\partial D_{v_i}} \hat{n}_x \phi \, dS$$

Density Gradient Accumulation

Backward Pass

Outcomes:

- The computation of ***E*** can be saved!
- Three `IDCT/IDXST` operations (e.g. `IDXST_IDCT_IDCT`) to compute ***E*** can be saved in each **forward** computation.
- The density gradient accumulation in each **backward** computation can be accelerated by reducing the dimension of integral region.

Apply Divergence Theorem

Forward Pass

$$\rho(x, y, z) = \sum_{i=1}^{n} \chi_{D_{v_i}}(x, y, z)$$

Density Map

$$a_{jkl} = \frac{1}{|B|} \sum_{x,y,z} \rho \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Coefficients: one `DCT_3D` required.

$$\phi = \sum_{j,k,l} \frac{a_{jkl}}{\omega_j^2 + \omega_k^2 + \omega_l^2} \cos(\omega_j x) \cos(\omega_k y) \cos(\omega_l z)$$

Potential Map: one `IDCT_3D` required.

$$g_i = \oiint_{\partial D_{v_i}} \hat{n}_x \phi \, dS$$

Density Gradient Accumulation

Backward Pass

Outcomes:

- The computation of ***E*** can be saved!
- Three `IDCT/IDXST` operations (e.g. `IDXST_IDCT_IDCT`) to compute ***E*** can be saved in each **forward** computation.
- The density gradient accumulation in each **backward** computation can be accelerated by reducing the dimension of integral region.

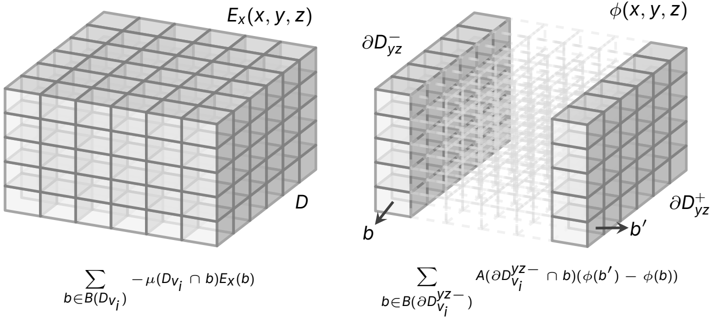
How to compute the integral?

Apply Divergence Theorem

The new rule of calculating g_i (the x-direction density derivative w.r.t. node v_i).

$$g_i = \oiint_{\partial D_{v_i}} \hat{n}_x \phi \, \mathrm{d}S \approx \iint_{\partial D_{v_i}^{yz-}} (\phi(x, y, z) - \phi(x + w_i, y, z)) \, \mathrm{d}S.$$

➜ An example illustrating the density gradient accumulation via the potential map ϕ.



➜ It is the difference of two 2D integral of potential values on a pair of faces.

Experimental Results

Benchmarks	7 mixed-size designs from ICCAD'23 contest
Metrics	die-to-die HPWL and runtime
Platform	Intel(R) Xeon(R) Platinum 8480C CPUs (Max 3.8 GHz) H800 GPU (80G VRAM)

➔ Compare to SOTA analytical 3D placer[8]: more than 3× speedup on CPU, and 4× speedup on GPU, without quality loss.

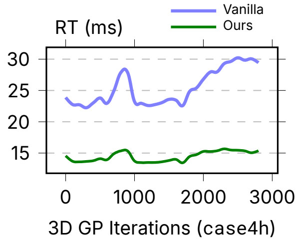
Bench.	Statistics		[TCAD'24][7]-CPU		Ours-CPU		[TCAD'24][7]-GPU		Ours-GPU	
	$\|V\|$	$\|V_M\|$	Score	RT	Score	RT	Score	RT	Score	RT
case2	14K	6	15540090	76	15639790	26	15635352	38	15443965	15
case2h1	14K	6	16719713	80	16706960	20	16569703	35	16775356	9
case2h2	14K	6	16759058	80	16773536	20	16820960	36	16721757	9
case3	124K	34	97388944	236	100684926	78	98206238	92	99997072	21
case3h	124K	34	109518959	239	109096510	73	108166770	86	107974746	20
case4	740K	32	1041523590	3070	1040956218	964	1037676163	335	1038139180	77
case4h	740K	32	635991850	2640	631972730	980	635259476	361	630601092	76
AVERAGE			0.996	3.300	1.000	1.000	1.000	4.029	1.000	1.000

[8] Yuxuan Zhao et al. "Analytical Heterogeneous Die-to-Die 3D Placement with Macros". In: *IEEE TCAD* 44.2 (2024), pp. 402–415.

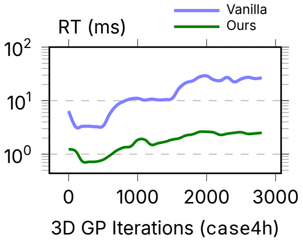
Ablation Study: Results of different algorithms on the same machine: about 36% faster on CPU and 43% faster on GPU (**end-to-end** runtime).

Bench.		Vanilla Score	Vanilla RT	Ours Score	Ours RT
CPU	case2	15657924	39	15639790	26
	case2h1	16723968	29	16706960	20
	case2h2	16748488	29	16773526	20
	case3	101738290	96	100684926	78
	case3h	109580966	86	109096510	73
	case4	1037251354	1315	1040956218	964
	case4h	632423856	1310	631972730	980
AVERAGE		1.002	1.359	1.000	1.000
GPU	case2	15587229	21	15443965	15
	case2h1	16850092	12	16775356	9
	case2h2	16801938	12	16721757	9
	case3	101196054	34	99997072	21
	case3h	107621590	33	107974746	20
	case4	1035731150	102	1038139180	77
	case4h	632611980	103	630601092	76
AVERAGE		1.004	1.431	1.000	1.000

Backward runtime on **CPU**



Backward runtime on **GPU**



Q&A

Reference I

[1] Yan-Jen Chen et al. “Late breaking results: Analytical placement for 3D ICs with multiple manufacturing technologies”. In: *Proc. DAC*. IEEE. 2023, pp. 1–2.

[2] Yan-Jen Chen et al. “Mixed-size 3D analytical placement with heterogeneous technology nodes”. In: *Proc. DAC*. 2024, pp. 1–6.

[3] Peiyu Liao et al. “Analytical Die-to-Die 3D Placement with Bistratal Wirelength Model and GPU Acceleration”. In: *IEEE TCAD* 43.6 (2023), pp. 1624–1637.

[4] Jingwei Lu et al. “ePlace-3D: Electrostatics based placement for 3D-ICs”. In: *Proc. ISPD*. 2016, pp. 11–18.

[5] Yuxuan Zhao et al. “Analytical Heterogeneous Die-to-Die 3D Placement with Macros”. In: *IEEE TCAD* 44.2 (2024), pp. 402–415.