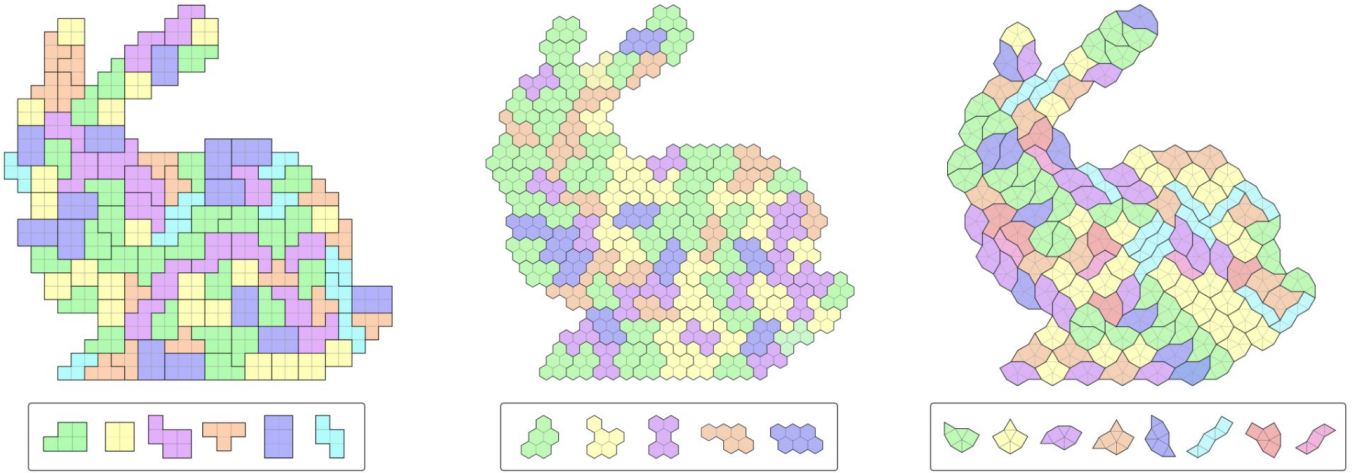
Inverse Tiling of 2D Finite Domains

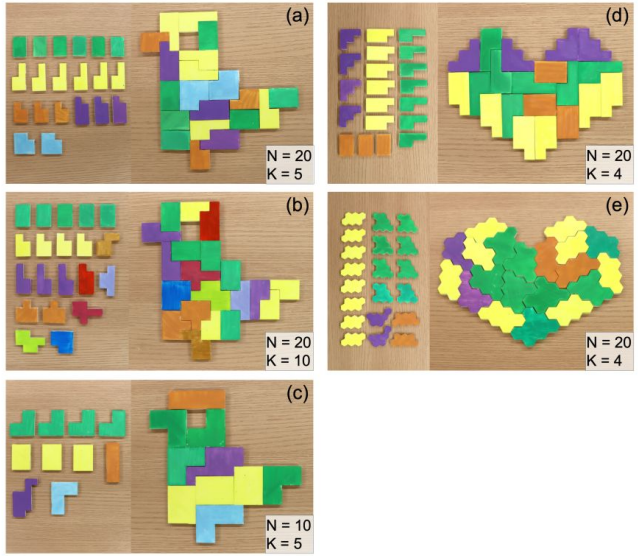
RULIN CHEN, Singapore University of Technology and Design (SUTD), Singapore and Beijing Normal–Hong Kong Baptist University, China
XUYANG MA, Singapore University of Technology and Design (SUTD), Singapore
PRAVEER TEWARI, Singapore University of Technology and Design (SUTD), Singapore
CHI-WING FU, The Chinese University of Hong Kong, China
PENG SONG, Singapore University of Technology and Design (SUTD), Singapore

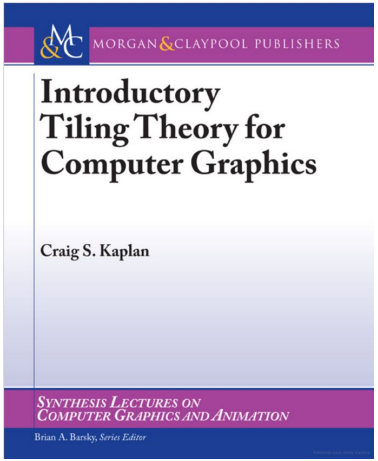


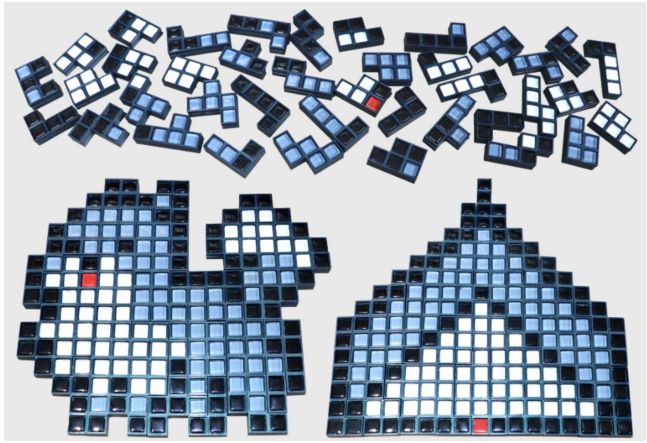
GDL – 4 mars 2026 – Mattéo Clémot

Tilings in graphics

Texture synthesis, sampling, decorative pattern generation, manufacturing, recreational puzzles



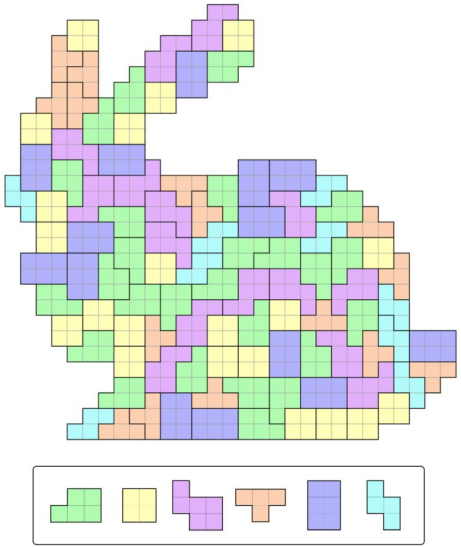




Definitions

Everything in the 2D plan

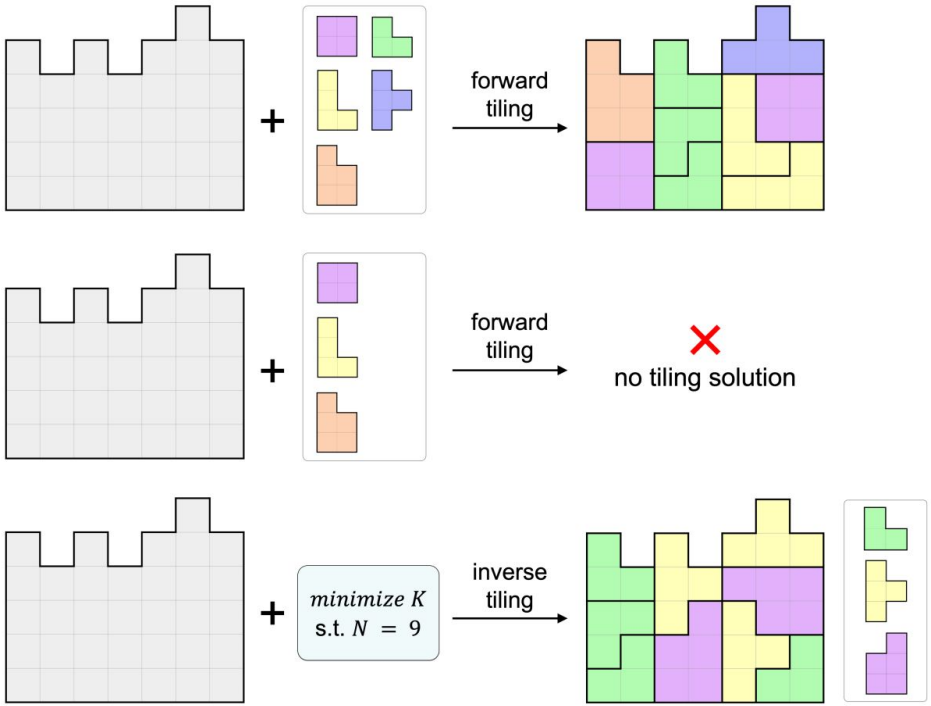
- ***Tiling*** of a domain: a countable set of regions called ***tiles***, with no overlaps and no gaps
- ***K-hedral tiling***: every tile congruent to one of K distinct shapes called ***prototiles***



Problems

Forward tiling: set of prototiles given

Inverse tiling: set of prototiles not given, and should be minimized (K) (with fixed number N of instances, range [C_{min},C_{max}] of prototiles size)



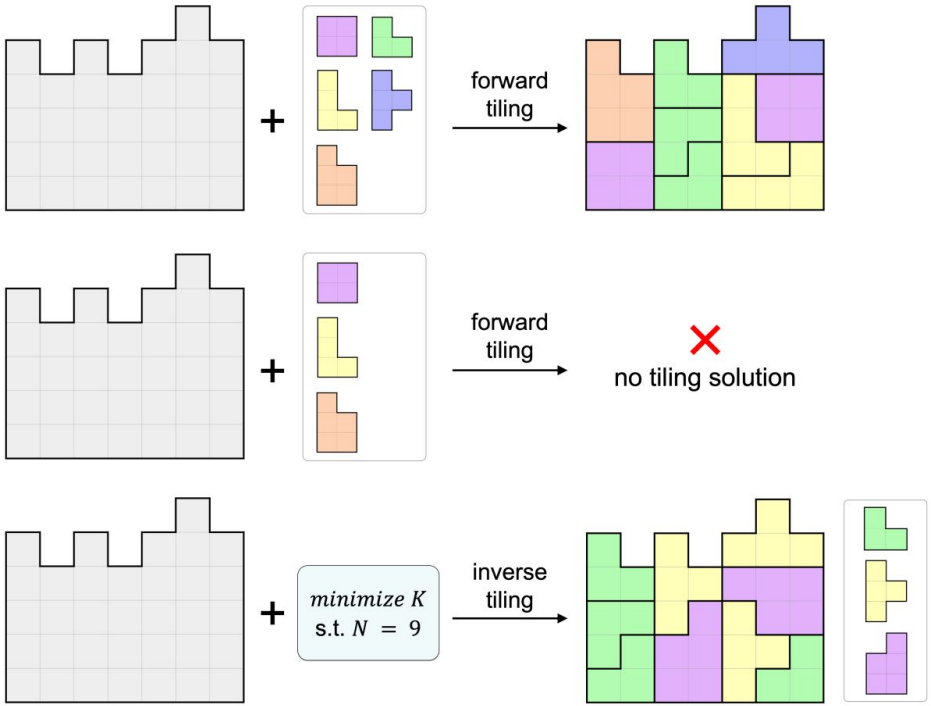
Problems

Forward tiling: set of prototiles given

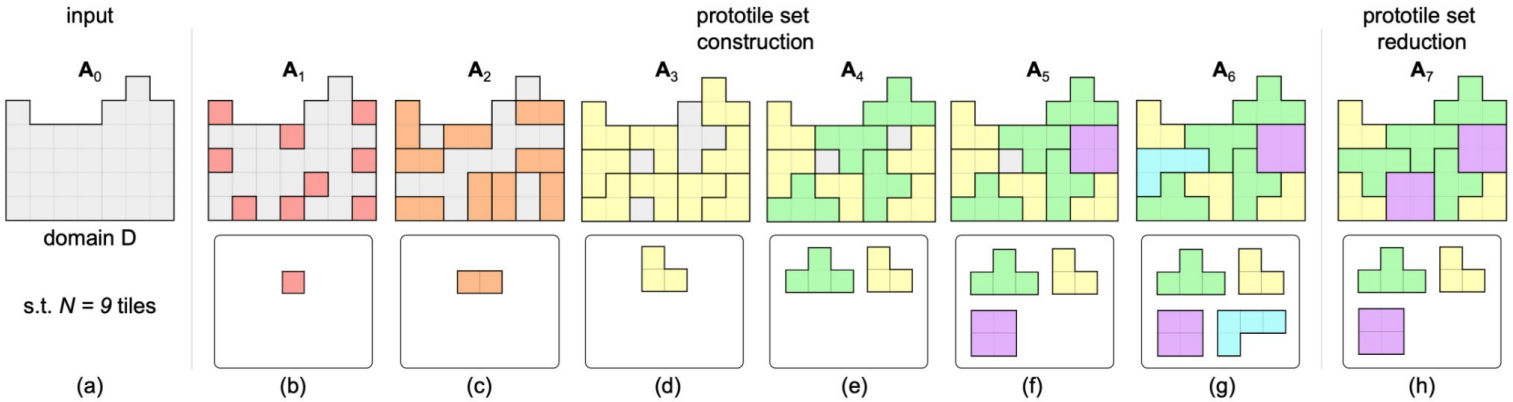
NP-complete

Inverse tiling: set of prototiles not given, and should be minimized (K) (with fixed number N of instances, range [C_{min},C_{max}] of prototiles size)

NP-hard



Approach



1. Construction
   a. N random monominoes (K=1) **(b)**
   b. Grow the instances of a prototile congruently, maintaining K **(c)**, **(d)**
   c. If not possible, grow a subset of the instances of a prototile congruently (increments K) **(e)**, **(f)**, **(g)**
   d. Until domain is covered
2. Reduction
   a. Try to eliminate prototiles with few instances via local re-tiling

Search tree

- Search tree to deal with *abortive tiling states*
- For each state, at most *m* child states
- Ranking of these child states (number of prototiles, number of uncovered cells)
- Backtracking if abortive states

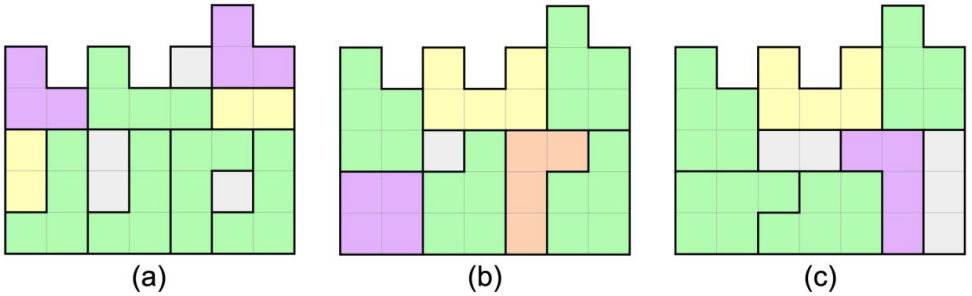
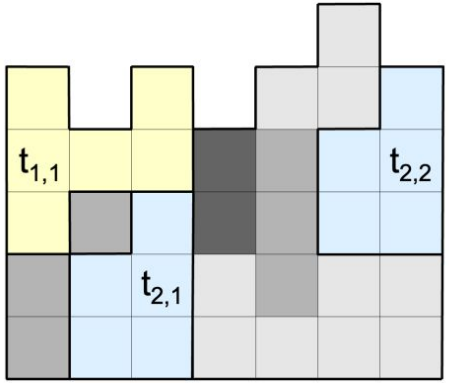


Fig. 4. Three example abortive tiling states. (a) The two yellow tiles cannot be enlarged. Their number of grid cells is less than the minimum allowed number of grid cells $C_{\text{min}} = 3$. (b) The uncovered grid cell (in gray) cannot be assigned to any tile without violating the maximum allowed number of grid cells $C_{\text{max}} = 5$. (c) We cannot assign the uncovered grid cells without violating the bounding box size constraint $W_{\text{bbox}} = 3$ and $H_{\text{bbox}} = 2$.

Heuristics

- Favor enlargeability
  - Initial seed distributed evenly
  - Enlarge tiles so that they still have uncovered adjacent cells (idea: minimize the potential blocking impact)
- Minimize K
  - Try to match existing prototiles in priority when enlarging the instances of a smaller prototile
  - Try to enlarge into a shape that matches part of a larger prototile

Construction



Blockability of uncovered grid cells

b( ) = 1

b( ) = 2

b( ) = 3

Enlargeability of tiles

$e(t_{1,1}) = 1.33$

$e(t_{2,1}) = 3.83$ $e(t_{2,2}) = 4.0$

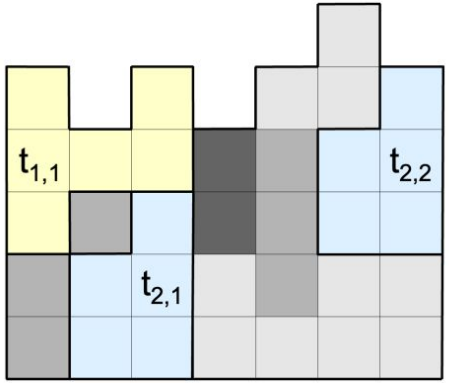
Enlargeability of prototiles

$e(t_1) = 1.33$

$e(t_2) = 3.915$

- ***blockability*** of a grid cell x: $b(x) =$ number of tiles intersecting with I-ring grid cell neighborhood of x
- ***enlargeability*** of a tile: $e(t_{k,j}) = \sum_{x_u \in \text{adjacent\_uncovered}(t_{k,j})} \frac{1}{b(x_u)}$
- ***enlargeability*** of a prototile: $e(t_k) = \frac{1}{n_k} \sum_j e(t_{k,j})$ (average on the instances)

Construction



Blockability of uncovered grid cells

b( ) = 1

b( ) = 2

b( ) = 3

Enlargeability of tiles

$e(t_{1,1}) = 1.33$

$e(t_{2,1}) = 3.83$ $e(t_{2,2}) = 4.0$

Enlargeability of prototiles

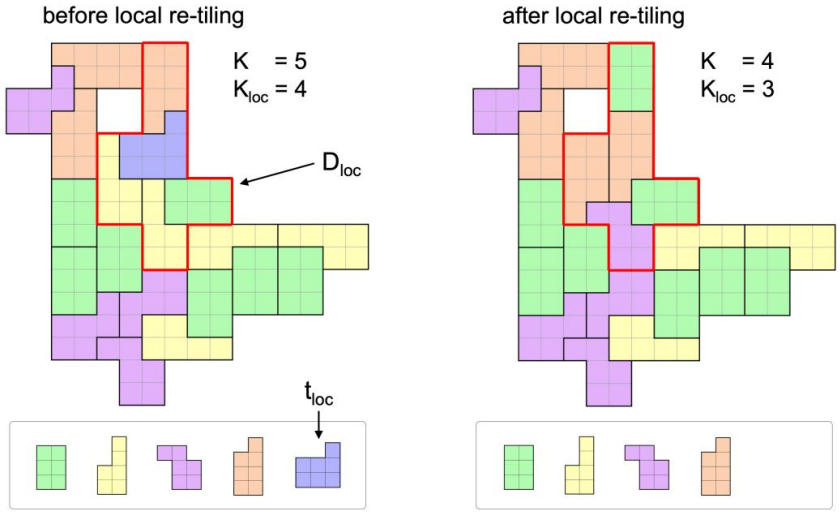
$e(t_1) = 1.33$

$e(t_2) = 3.915$

1. select a prototile: small size, high enlargeability
2. select a reference prototile instance
3. choose a target
   a. try to match an existing prototile
   b. otherwise, assign a uncovered cell with low blockability
4. enlarge the other prototile instances

Reduction

1. select a prototile: few instances
2. identify the associated region D_{loc}
3. re-tile the region with the construction method (try to match the other prototiles)
4. accept if K smaller



Timings

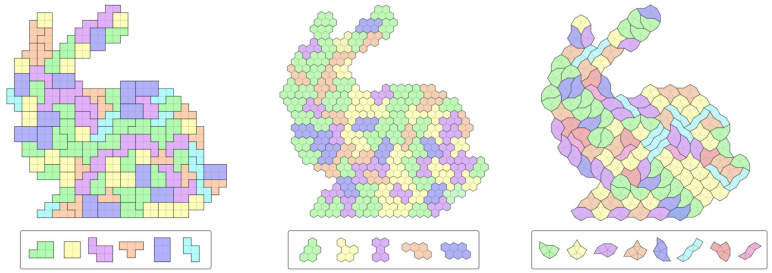
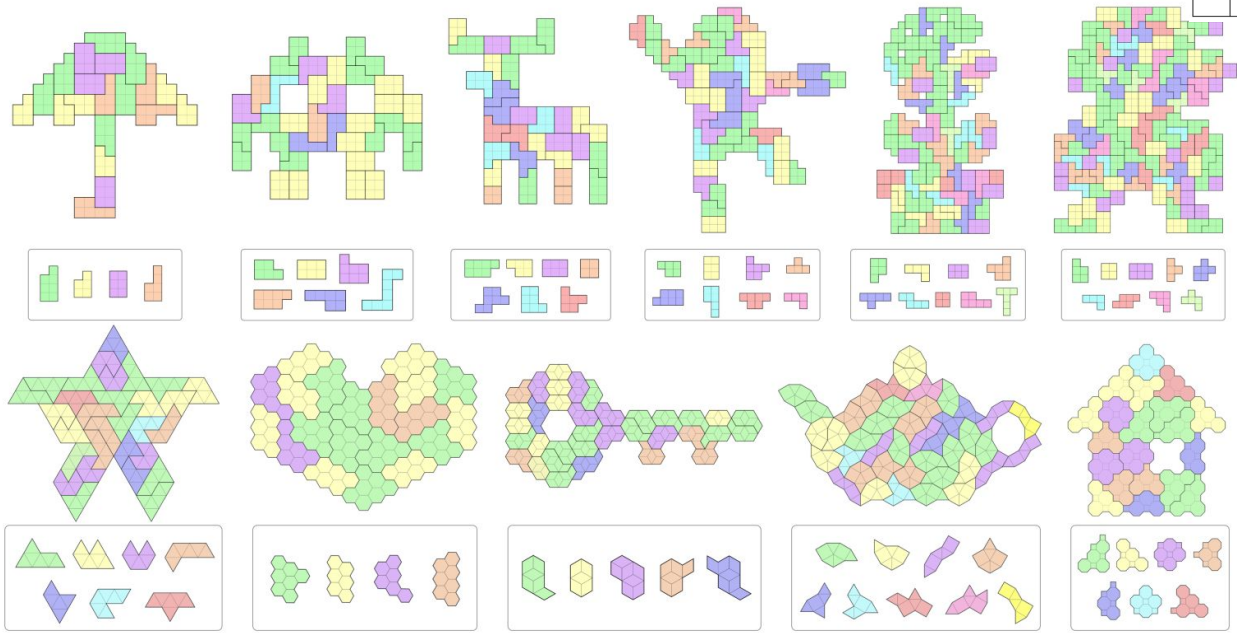


Fig	Input Domain	Grid	M	N	C_{min}	C_{max}	K_{constr}	K	N/K	T_{constr} (mins)	T_{reduce} (mins)	T_{total} (mins)
1	Bunny	Squ	484	100	4	6	9	6	16.7	32.6	12.7	45.3
		Hex	509	100	5	6	12	5	20.0	43.0	9.7	52.7
		Squ-Tri	756	125	5	7	20	8	15.6	49.7	21.6	71.2
7	Umbrella	Squ	120	20	5	7	6	4	5.0	0.6	0.3	0.9
	Robot		202	35	5	7	9	6	5.8	13.3	1.8	15.1
	Deer		180	30	4	7	9	7	4.3	7.2	0.9	8.1
	Mega Man		274	50	4	8	16	8	6.3	35.8	13.9	49.7
	Flower		408	80	4	6	14	9	8.9	55.0	23.4	78.4
	Mario		604	120	4	6	18	9	13.3	119.6	16.1	135.7
8	Kite	Tri	172	30	5	6	9	7	4.3	10.6	2.1	12.6
	Heart	Hex	130	20	6	7	6	4	5.0	8.5	1.0	9.5
	Key	Rho	190	35	5	7	8	5	7.0	12.6	5.3	17.8
	Teapot	Squ-Tri	242	42	5	6	12	9	4.7	37.4	8.2	45.6
	House	Oct-Squ	134	20	6	8	8	7	2.9	28.2	6.8	34.9



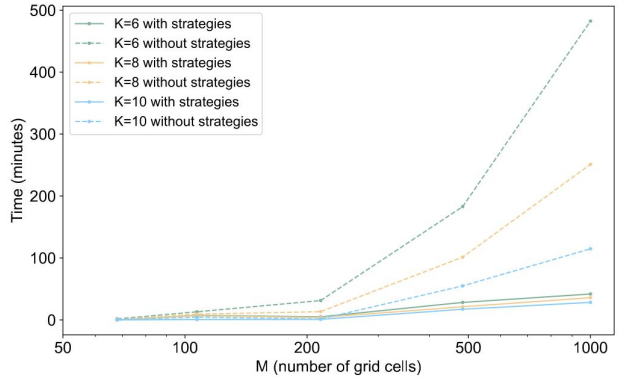
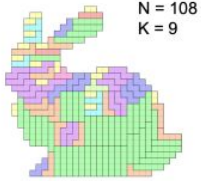
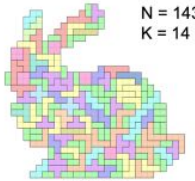
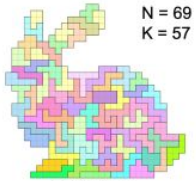
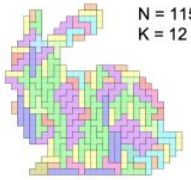
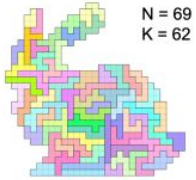
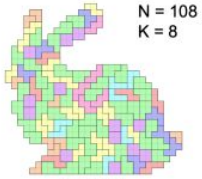
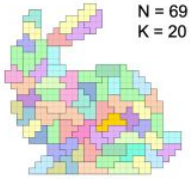
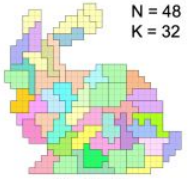


Fig. 10. Experiment to evaluate the scalability of our inverse tiling approach for tiling Bunny in different resolutions. We compare our approach (in solid curves) with a baseline approach (in dashed curves) that uses our computational framework without our proposed strategies.

Minimization of K

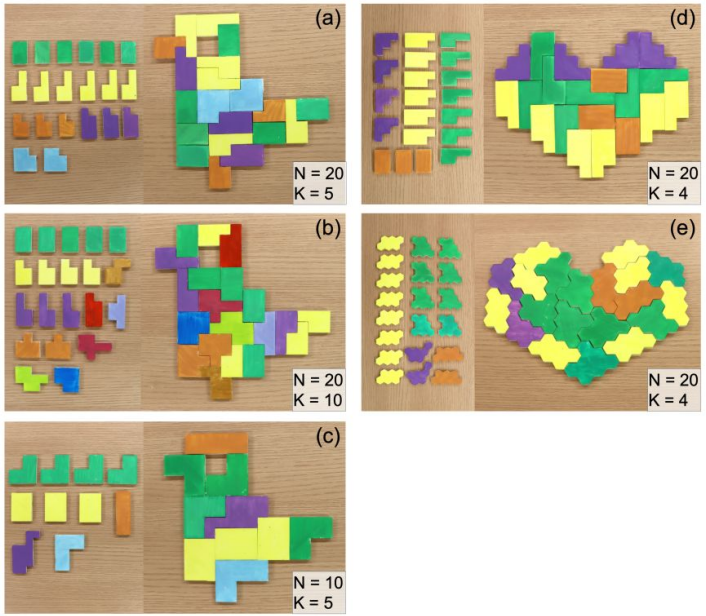
- Compared with forward tiling approaches

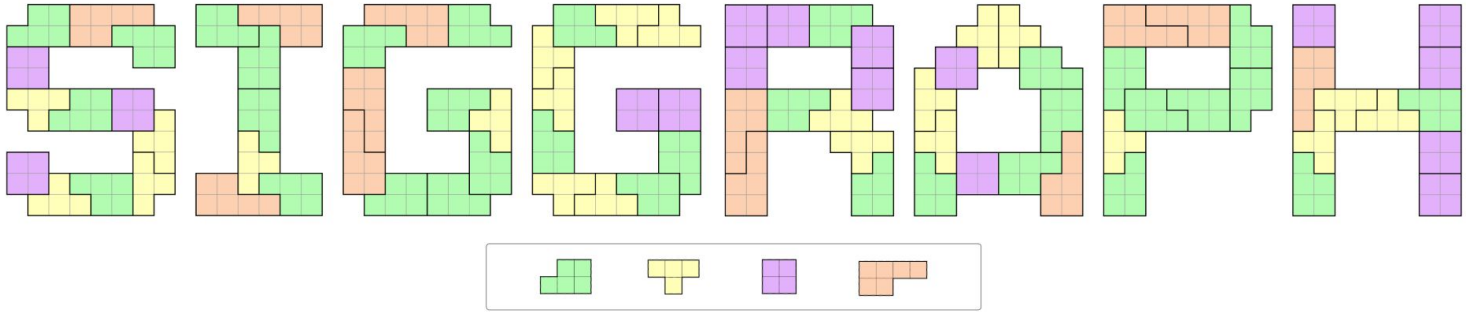
	Task #1	Task #2	Task #3
	$[C_{\min}, C_{\max}] = [2, 5]$	$[C_{\min}, C_{\max}] = [5, 8]$	$[C_{\min}, C_{\max}] = [8, 11]$
BT	 N = 108 K = 9 0.26 min		
ILP	 N = 143 K = 14 0.36 min	 N = 69 K = 57 909.97 min	
SAT	 N = 115 K = 12 0.093 min	 N = 69 K = 62 7.66 min	
Ours	 N = 108 K = 8 8.72 min	 N = 69 K = 20 47.61 min	 N = 48 K = 32 105.67 min

Application to puzzles: user study

breakthrough: puzzle's difficulty related to N, and K, and the grid (!!!)

	Bird N=20, K=5	Bird N=20, K=10	Bird N=10, K=5	Heart (squ) N=20, K=4	Heart (hex) N=20, K=4
User 1	10.55	/	21.34	36.41	/
User 2	13.42	26.47	6.5	14.95	39.71
User 3	18.94	37.54	3.41	21.96	17.56
User 4	22.84	/	17.46	57.51	/
User 5	19.62	/	5.61	17.35	45.78
User 6	22.45	41.28	10.54	27.46	52.68
$\bar{t}$	17.97	35.10	10.81	29.27	38.93




SIGGRAPH