Kinetic Shape Reconstruction (Supplementary Material)

JEAN-PHILIPPE BAUCHET and FLORENT LAFARGE, Université Côte d'Azur, Inria, France

This supplementary material presents a pseudo-code describing the kinetic partitioning algorithm and a comparative study with surface reconstruction and surface approximation methods on 42 datasets.

CCS Concepts: • **Computing methodologies** → *Mesh models*; *Parametric curve and surface models.*

ACM Reference Format:
Jean-Philippe Bauchet and Florent Lafarge. 2020. Kinetic Shape Reconstruction (Supplementary Material). *ACM Trans. Graph.* 39, 5, Article 1 (January 2020), 18 pages. https://doi.org/10.1145/3376918

A PSEUDO-CODE OF THE KINETIC PARTITIONING ALGORITHM

We start by defining some notations. Let:

- $\boldsymbol{P} = (P_1, .., P_N)$, be the input set of N convex polygons;
- $\overline{\boldsymbol{P}} = (\overline{P}_1, .., \overline{P}_n)$, be the set of n (infinite) supporting planes of polygons $\boldsymbol{P}$ with $n \leq N$;
- $\mathcal{P} = (\mathcal{P}_1, .., \mathcal{P}_n)$, be the set of 2D polygonal partitions with $\mathcal{P}_i$ defined in the supporting plane $\overline{P_i}$ by the triplet (L_i, T_i, S_i) :
  - L_i is the set of intersection lines $L_{ij} = \overline{P_i} \cap \overline{P_j}$;
  - T_i is the set of intersection-free polygons that propagate on the plane $\overline{P_i}$;
  - S_i is the set of line-segments $s_{ij} = \overline{P_i} \cap T_j$;
- Q, be the a global priority queue;
- C, be the output partition of polyhedra;
- $\mathcal{F}$, be the set of polygonal facets contained in C.

The pseudo-code of the kinetic partitioning algorithm is composed of three successive steps: Initialization of the kinetic data structure, processing of the events, and finalization of the data structure. The first step is described in Algorithm 1 while the two other steps are detailed in Algorithm 2.

Some comments on the pseudo-code. An illustration of the initialization procedure can be found in Figure 3 of the paper. On the left side, the blue polygon is an input convex polygon P_i in the plane $\overline{P_i}$. On the right side, the dashed lines, the line-segments and the four blue intersection-free polygons are elements of L_i, S_i and T_i, respectively. Among the collision types illustrated in Figure 4 of the paper, note that the initial polygon decomposition avoids considering collision type (d) in practice.

Algorithm 1 Kinetic partitiong (part 1/2)

$\boldsymbol{P} \leftarrow (P_1, P_2, \ldots, P_n)$
$\overline{\boldsymbol{P}} \leftarrow (\overline{P_1}, \overline{P_2}, \ldots, \overline{P_n})$
$\mathcal{P} \leftarrow \emptyset$
$Q \leftarrow \emptyset$

procedure Initialization

  for $\overline{P_i} \in \overline{\boldsymbol{P}}$ **do**
    $L_i \leftarrow \bigcup_{j \neq i} L_{ij}$ where $L_{ij} = \overline{P_i} \cap \overline{P_j}$
    $T_i \leftarrow \{P_i\}$
    $S_i \leftarrow \emptyset$
    $\mathcal{P}_i \leftarrow (L_i, T_i, S_i)$
    Add $\mathcal{P}_i$ to $\mathcal{P}$
  end for

  for $\mathcal{P}_i \in \mathcal{P}$ **do**
    for $L_{ij} \in L_i$ **do**
      for $P \in T_i$ **do**
        if $L_{ij} \cap P \neq \emptyset$ **then**
          Create sliding or frozen vertices along L_{ij}
          Split P into subpolygons P_1, P_2
          Add P_1, P_2 to T_i
          Remove P from T_i
          Add line-segment $s = P_1 \cap P_2$ to S_j
        end if
      end for
    end for

    for $P \in T_i$ **do**
      for each non-frozen vertex $v \in P$ **do**
        Determine the 3 next events $e_{v,j} = \{t \mid v(t) \cap L_{ij} \neq \emptyset\}$ where $L_{ij} \in L_i$
        Add these events to Q
      end for
    end for
  end for

end procedure

B ROBUSTNESS TO NOISE

Figure 1 shows how our algorithm digests a noisy point cloud representing an object with details and curving features. In this example, the defect-free point cloud of Tower of Pi (presented in Figure 9 of the paper) has been perturbed by a random noise both on point coordinates (0.2% of the bounding box diagonal of the entire object, or equivalently 3% of the bounding box diagonal of a digit) and on point normals (30-degree additive noise). Such a perturbation corresponds to the limit where our algorithm can still produce a

Authors' address: Jean-Philippe Bauchet; Florent Lafarge, Université Côte d'Azur, Inria, Sophia Antipolis, France, Firstname.Lastname@inria.fr.

0730-0301/2020/1-ART1
https://doi.org/10.1145/3376918

Algorithm 2 Kinetic partitioning (part 2/2)

procedure PROCESSING_EVENTS

  while $Q \neq \emptyset$ **do**
    Pop the vertex v intersecting the line L_{ij} from Q
    Get the polygon P of T_i containing v
    Determine the collision case (see Figure 4)
    Update P with sliding and/or frozen vertices
    Determine the 3 next events of the new vertices of P
    Add these events to Q
    Add line-segment $s = \overline{P}_j \cap P$ to S_j

    if CROSSING_CONDITION(P, L_{ij}) = TRUE **then**
      Add a new polygon P' to T_i
      Determine the 3 next events of the vertices of P'
      Add these events to Q
      Add line-segment $s' = \overline{P}_j \cap P'$ to S_j
    end if

    if v is not frozen **then**
      if Q has no more events involving v **then**
        Determine the 3 next events involving v
        Add these events to Q
      end if
    else
      Remove all future events of v from Q
    end if

  end while

end procedure

procedure FINALIZATION
  $\mathcal{F} \leftarrow \emptyset$
  $\mathcal{C} \leftarrow \emptyset$
  for $\mathcal{P}_i \in \mathcal{P}$ **do**
    Assemble adjacent polygons of T_i into facets $\mathcal{F}_i$
    Add $\mathcal{F}_i$ to $\mathcal{F}$
  end for
  Add facets on the bounding box to $\mathcal{F}$
  Assemble adjacent facets of $\mathcal{F}$ into polyhedra $\mathcal{C}$
end procedure

result with visually recognizable digits. The complexity of the output mesh is twice higher than the one obtained from the defect-free version of the point cloud, resulting from a higher number of detected shapes. Similarly, geometric accuracy is decreased as shapes are detected with a higher ϵ value. Visually, the output mesh is less structure-aware even if the digits can still be recognized.

C COMPARATIVE STUDY ON 42 DATASETS

This section extends the paragraphs *Comparisons with surface reconstruction methods* and *Comparisons with surface approximation pipelines* of Section 6 with a more complete comparative study on a collection of 42 datasets (the 34 models used in our paper and the 8 models of Figure 4 of Polyfit paper). This collection, which includes 11 point clouds generated from multiview stereo, 19 Laser scans and 12 synthetic datasets sampled from CAD models, has been divided into three groups: Simple (< 100 shapes), Intermediate (between 100 and 500 shapes and the Hilbert Cube models) and Advanced (> 500 shapes). We compared the efficiency of our algorithm on this dataset collection with two surface reconstruction methods (Polyfit and Chauve's algorithm) and four surface approximation pipelines (Screened Poisson+QEM, Screened Poisson+VSA, Screened Poisson+SAMD and Screened Poisson+SAMD-CP which is an alternative version of SAMD where corners are preserved) under the experimental conditions detailed in Section 6. The screened Poisson algorithm was run with an octree depth set to 9. Tables 1, 2 and 3 give quantitative results in terms of output quality (SMH error, AMH error, simplicity score) and performances (processing time, memory consumption) for simple, intermediate and advanced models respectively. Figures 2-13 show the corresponding qualitative results (output mesh and error map).

Some comments on the quantitative and qualitative results.

- Our algorithm outclasses other methods in terms of accuracy for 33 of the 42 models. The gain is particularly high against the two surface approximation pipelines which fail to recover the structure of objects. These two pipelines are of interest on advanced models for quickly returning output mesh with low accuracy. The scenarios where our algorithm does not necessarily deliver the most accurate results concern some simple models with a very low number of planar shapes (ie BUILDING A, BUILDING C, FERTILITY COARSE, FOAM BOX, ROOM A, ROOM B). For such very simple cases, the use of a kinetic partitioning does not bring any particular benefit, apart from a lower memory consumption and a better simplicity score.
- From a visual point of view, our algorithm gives the most detailed and structured descriptions of objects and scenes. The visual differences are especially strong with approximation pipelines on urban and indoor scenes.
- The simple models include the 8 datasets from Polyfit paper. Polyfit was not able to process all the simple models. It fails to return valid results on BARN MVS and BUNNY after several hours of computing, its integer programming solver being not scalable enough to operate from exhaustive partitions generated from approximately one hundred planar shapes. Chauve 's method scales better with its visibility-based graph-cut solver, but still, the intermediate models with more than 300 planar shapes could not be processed.
- The idea of subdividing partitions before extracting surface from the union of these partitions cannot be applied efficiently to Polyfit and Chauve's method. In order to deliver coherent results, the subsequent surface extraction needs to operate on the reunified partition. However, the union of exhaustive sub-partitions still gives a very high number of polyhedra (for instance, several millions of polyhedra from 500 shapes only with a 4^3 decomposition). Unfortunately, the integer programming solver of Polyfit and the visibility-based

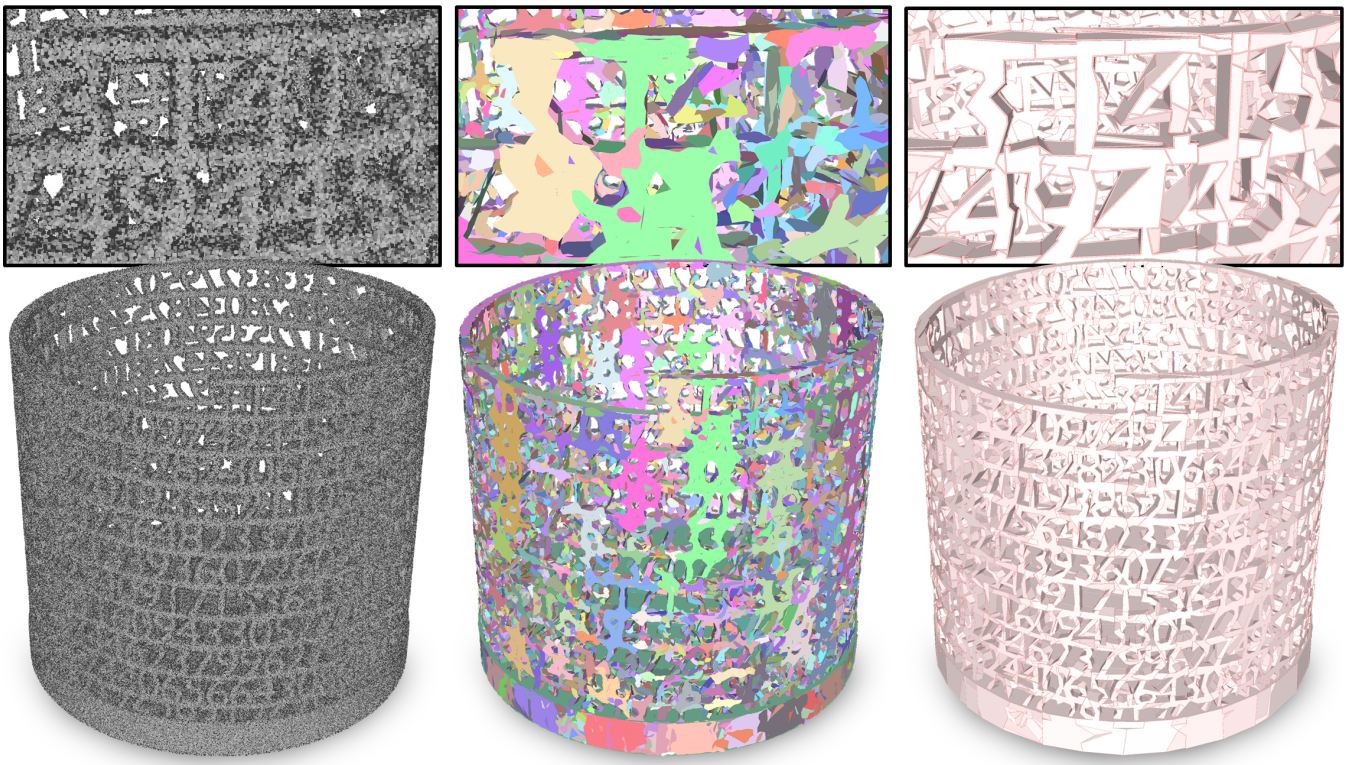

Fig. 1. Reconstruction of a noisy version of Tower of Pi. $17.1K$ planar shapes (middle) are detected from the $2.9M$ noisy input points (left) and assembled by our algorithm into a concise polygonal mesh of $26.5K$ facets (middle). In contrast to the result obtained on the defect-free version (presented in Figure 9 of the paper), the output mesh has a higher complexity and a lower geometric accuracy. Although the reconstruction of the digits is not as accurate and regularized as for the defect-free version, they remain recognizable from a visual point of view (see closeups).

graph-cut solver of Chauve cannot process millions of polyhedra under reasonable time or without exceeding memory consumption partitions. Note also the integer programming solver of Polyfit does not process the facets located on the sides of bounding box: this leads to miss border components as ground from some datasets.

- The number of output facets returned by the SP+VSA and SP+SAMD-CP pipelines is not easily controllable and can differ from the target number while always remaining higher. In particular, SAMD-CP is an alternative version of SAMD that preserves the corner vertices. This alternative version detects sharp features (based on planar shape intersections) and corner vertices, keeps corner vertex positions unchanged, leaves the collapse vertices on the sharp edges and avoids the collapse between two corner vertices. This algorithm however limits the maximal number of possible collapses and does not allow us in practice to reach the targeted number of facets in our experiments. The reason is that we rarely detect the ideal configuration of corner vertices from a real-world Screen Poisson reconstruction, with frequent parts where corner vertices are missing and other parts where they are overly-detected.

		Barn MVS	Building A	Building B	Building C	Bunny	Chair	Cottage	Couch	Fertility-Coarse	Foam box	Hand	Lans-Coarse	Rocker arm	Rooms A	Rooms B	Temple
Type		U	U	U	U	F	S	U	S	F	S	F	U	S	I	I	S
Origin		MVS	MVS	MVS	MVS	Laser	Laser	MVS	Laser	Laser	Laser	Laser	Laser	Laser	Laser	Laser	MVS
#**p**		619K	101K	73K	577K	146K	756K	143K	911K	242K	382K	369K	1.22M	733K	186K	176K	621K
#**s**		95	20	34	32	98	23	23	21	75	61	75	24	73	48	28	69
ϵ		0.75	5	2.5	0.4	0.4	1.6	0.7	2.2	1.75	1.75	0.8	1	1.1	0.8	0.3	1.7
σ		0.08	0.6	0.25	1.2	0.15	0.5	0.2	0.7	0.2	0.2	0.07	0.8	0.04	0.3	0.6	0.3
Ours	$\mathbf{e_A}$	**0.231**	**0.951**	**0.605**	**0.372**	**0.454**	**0.557**	**0.401**	**0.560**	0.350	0.247	**0.423**	0.491	**0.257**	0.677	0.553	**0.381**
	$\mathbf{e_S}$	**0.179**	1.522	**0.735**	0.344	**0.540**	**0.580**	**0.471**	**1.681**	0.498	0.281	**0.480**	**0.726**	**0.382**	0.603	0.538	**0.481**
	s	**2.5**	**0.690**	1.478	**0.941**	**0.856**	**1**	**0.821**	**0.84**	**0.789**	**0.968**	**0.815**	**0.96**	**1**	**1.043**	**1.077**	0.726
	#f	38	29	23	34	111	23	28	25	95	63	92	25	73	46	26	95
	t	28.9	3.4	2.5	18.4	53.3	25	4.2	25.1	25	15	31	33.4	43.4	7.9	4.8	35
	m	53	10	13	15	97	34	15.7	11.6	61.2	29	76.6	11.5	53.8	19.1	8.8	48.1
Polyfit	$\mathbf{e_A}$	-	1.347	2.097	0.461	-	1.447	1.491	6.385	0.388	0.221	0.460	3.977	0.361	**0.590**	0.680	0.548
	$\mathbf{e_S}$	-	**1.170**	1.385	0.343	-	1.089	1.259	3.325	0.500	0.297	0.537	2.592	0.524	**0.556**	0.572	0.683
	s	-	1.429	**1.36**	1.067	-	1.438	1.278	1.235	0.5	**0.968**	0.765	0.923	0.507	0.941	1.474	**0.767**
	#f	-	14	25	30	-	16	18	17	150	63	98	26	144	51	19	90
	t	-	2.1	2.5	10.2	-	16.9	5.3	18.8	1070	672	296	26.6	9343	4.5	3.1	1943
	m	-	144	169	252	-	241	177	299	1919	1748	1600	331	1896	221	135	900
Chauve	$\mathbf{e_A}$	1.042	9.585	2.626	0.407	0.545	1.422	1.429	3.931	**0.328**	**0.207**	0.462	**0.421**	0.357	0.790	**0.526**	0.459
	$\mathbf{e_S}$	0.772	5.172	1.704	**0.308**	0.555	1.082	0.823	2.137	**0.407**	**0.261**	0.488	1.195	0.411	0.671	**0.465**	0.911
	s	3.8	0.625	0.330	0.571	0.375	0.92	1.278	1.5	0.142	0.570	0.399	0.667	0.361	0.384	0.519	0.099
	#f	25	32	103	56	261	25	18	14	528	107	188	36	202	125	54	700
	t	131.2	0.4	1.9	5.5	94	9.7	1.5	6.7	53.1	31.5	115	4.4	93.6	8.9	1.7	76.7
	m	3189	10	103	125	2924	501	56	308	1037	1131	1400	96	2141	196	56	1673
SP + QEM	$\mathbf{e_A}$	2.815	4.081	6.344	2.873	1.435	4.287	2.373	2.922	0.722	2.969	0.724	2.814	2.124	2.847	2.871	2.127
	$\mathbf{e_S}$	1.575	2.482	3.856	1.934	1.069	3.000	1.513	1.971	0.756	1.933	0.777	1.831	1.509	1.984	1.645	1.498
	#f	39	28	22	26	110	24	27	25	100	62	92	25	72	46	26	92
	t	36	17.4	14.2	21.4	21	70.1	22.7	22.8	38.5	55	29.1	31.6	38.1	34.7	29.9	43.5
	m	232	162	111	458	195	495	156	406	289	518	241	311	142	377	399	453
SP + VSA	$\mathbf{e_A}$	0.752	4.222	2.673	1.593	1.002	1.906	1.181	1.762	1.724	1.404	1.058	2.894	1.201	2.066	0.911	0.894
	$\mathbf{e_S}$	0.538	3.107	1.921	1.316	1.031	1.664	0.958	1.806	1.287	0.960	0.918	2.355	1.091	1.422	0.704	0.806
	#f	717	213	157	947	175	420	528	113	93	682	127	31	83	568	123	270
	t	45.3	27.5	15.1	24.3	27.1	96.7	26.4	27.1	28.8	78.8	39.6	31.8	54.4	41.2	40.2	69.3
	m	675	428	347	480	352	1572	473	556	361	1341	740	311	142	901	758	1013
SP + SAMD-CP	$\mathbf{e_A}$	1.164	1.138	0.806	0.387	1.042	1.797	1.341	1.965	0.366	2.363	0.745	0.842	0.907	0.704	0.760	1.361
	$\mathbf{e_S}$	0.675	0.988	0.781	0.307	0.909	1.302	0.833	1.556	0.383	1.480	0.789	0.761	0.744	0.604	0.552	0.998
	#f	141	95	92	267	162	40	75	35	352	105	120	102	137	125	98	125
	t	30.3	14.8	10.7	18.5	18.1	52.2	18.2	17.5	30.2	43.3	24.6	31.4	48.6	26	21.7	35.4
	m	232	162	150	458	196	793	232	406	361	518	338	311	142	377	399	453
SP + SAMD	$\mathbf{e_A}$	1.428	1.524	1.071	1.885	0.585	2.326	0.801	1.483	0.657	1.546	0.508	1.686	0.800	1.240	1.907	0.686
	$\mathbf{e_S}$	0.902	1.059	1.240	1.359	0.820	1.394	0.776	1.840	0.907	1.481	0.806	1.721	0.971	0.940	1.186	0.766
	#f	39	28	23	32	112	21	29	26	96	64	92	25	72	46	26	93
	t	31	14.8	11.3	25.6	18.1	52.2	18.9	17.5	34.9	43.2	25.1	32.2	53.7	25.3	22.3	35.5
	m	232	162	150	458	196	793	232	406	361	518	338	311	142	377	399	453

Table 1. Quantitative results on simple models. Possible types of objects are urban (U), freeform (F), indoor (I) and structured (S). Possible point cloud origin include multiview stereo (MVS), laser scanning (Laser) and point sampling from CAD models (CAD). For these simple models, the number of input points #**p** and the number of planar shapes #**s** range from $101K$ to $1.2M$ points and from 20 to 98 shapes respectively. The fitting tolerance ϵ and the minimal shape size σ for shape detection are expressed in percent of the bounding box diagonal and in percent of the total number of input points respectively. The evaluation metrics $\mathbf{e_S}$, $\mathbf{e_A}$, **s**,#**f**, **t** and **m** correspond to the symmetric mean Hausdorff error (in % of the bounding box diagonal), the mean Hausdorff error from input points to output model, the simplicity score, the number of output facets, the running time (in sec) and the memory peak (in MB) respectively. Note that Polyfit fails to return a result on Barn MVS and Bunny after several hours of computing.

		Barn Laser	Building Block	Capron	Hilbert Cube	Hilbert Cube (N 0.5)	Hilbert Cube (N 1)	Hilbert Cube (N 1.5)	Hilbert Cube (O 500)	Hilbert Cube (O 1500)	Hilbert Cube (D 25)	Hilbert Cube (D 75)	Horse	Ignatius	Lans-Fine	M60	Sphere
Type		U	U	F	S	S	S	S	S	S	S	S	F	F	U	S	F
Origin		Laser	MVS	MVS	CAD	CAD	CAD	CAD	CAD	CAD	CAD	CAD	MVS	MVS	Laser	MVS	CAD
#**p**		2.1M	793K	168K	144K	144K	144K	144K	864K	2.3M	36K	108K	788K	1.4M	1.2M	2.8M	100K
#**s**		222	160	146	986	1135	2601	480	1003	1184	531	965	274	294	306	362	304
ϵ		0.2	0.5	0.6	1	1	2	1	1	1	1	1	0.6	0.8	0.2	0.4	0.4
σ		0.01	0.05	0.09	0.01	0.01	0.01	0.07	0.002	0.001	0.05	0.02	0.04	0.04	0.02	0.04	0.05
Ours	$\mathbf{e}_A$	**0.133**	**0.232**	0.282	**0**	**0.030**	**0.122**	**0.741**	**0.009**	**0.537**	4.755	**0.021**	**0.249**	**0.212**	0.112	**0.442**	**0.054**
	$\mathbf{e}_S$	**0.128**	**0.222**	**0.290**	**0.124**	**0.140**	**0.229**	**0.929**	**0.128**	**0.448**	2.513	**0.135**	**0.237**	0.288	**0.355**	**0.348**	**0.119**
	#**f**	142	142	156	986	997	2087	894	986	626	720	1032	351	443	179	471	319
	t	93.5	107.5	106.8	16.1	16.8	121.2	21.4	18.2	19.9	7.4	13.8	712.8	201.8	576.7	281.1	138.9
	m	54	147	194	105	66	451	62	59	50	79	126	1018	372	824	378	256
Chauve	$\mathbf{e}_A$	0.465	1.115	**0.223**	-	-	-	-	-	-	-	-	-	-	**0.083**	-	**0.054**
	$\mathbf{e}_S$	0.627	0.419	0.466	-	-	-	-	-	-	-	-	-	-	0.491	-	0.387
	#**f**	814	589	10k	-	-	-	-	-	-	-	-	-	-	3719	-	1256
	t	2475	333.4	555.7	-	-	-	-	-	-	-	-	-	-	6510	-	294
	m	>32GB	5361	6080	-	-	-	-	-	-	-	-	-	-	>32GB	-	8616
SP + QEM	$\mathbf{e}_A$	0.375	1.094	0.855	1.501	1.583	0.858	1.382	3.794	10.89	**2.172**	1.465	0.471	0.330	0.255	0.867	0.174
	$\mathbf{e}_S$	0.339	0.690	0.587	1.044	1.060	0.640	1.156	2.240	6.073	**1.439**	1.025	0.341	0.277	0.386	0.627	0.214
	#**f**	143	143	155	986	998	2088	894	978	568	721	1030	350	443	175	470	318
	t	28.8	27.8	34.5	26.8	29	27.7	27.7	161.4	494.4	7.8	22.5	38.4	35.8	31.6	43.3	24.6
	m	307	391	98	155	179	367	350	1427	4576	137	156	339	422	311	287	198
SP + VSA	$\mathbf{e}_A$	0.388	0.566	0.738	4.884	2.173	2.002	2.962	3.459	0.815	3.655	5.722	1.208	0.725	0.354	0.533	0.060
	$\mathbf{e}_S$	0.332	0.454	0.5	2.594	1.288	1.252	1.807	2.286	0.975	2.223	3.096	0.818	0.625	0.383	0.427	0.141
	#**f**	501	158	712	2486	2737	2049	1938	55K	489K	887	1081	451	468	164	1694	311
	t	33.8	35.3	44	36	40	36.5	35.6	210.3	806.5	10.1	30.4	52	47.7	37	58.6	40
	m	579	551	608	507	577	617	666	4617	15571	179	492	825	773	311	772	487
SP + SAMD-CP	$\mathbf{e}_A$	0.227	0.305	0.533	0.317	0.310	0.354	1.213	1.109	1.262	1.312	0.381	0.509	0.332	0.206	0.598	0.197
	$\mathbf{e}_S$	0.184	0.233	0.394	0.292	0.309	0.373	0.946	1.237	1.186	0.877	0.328	0.364	0.303	0.424	0.466	0.303
	#**f**	495	654	367	2747	3149	3903	1652	3379	4475	1830	3760	350	442	180	500	318
	t	23.4	22.8	28.3	22.1	23.1	21.3	21.2	94.2	238.7	6.5	21.6	32.8	31.7	31.4	36.4	38.1
	m	307	391	491	130	338	367	350	1147	2443	137	254	339	422	311	405	198
SP + SAMD	$\mathbf{e}_A$	0.134	0.397	0.423	2.520	2.898	0.657	2.483	2.704	3.150	3.926	2.819	0.475	0.253	0.203	0.677	0.199
	$\mathbf{e}_S$	0.211	0.424	0.399	1.546	1.696	0.565	1.635	2.016	2.092	2.258	1.676	0.363	**0.259**	0.460	0.521	0.296
	#**f**	141	141	157	985	997	2088	895	995	631	719	1033	351	442	180	471	318
	t	38.4	38.9	29.5	195.0	1347	35.8	32.7	2143.4	3022.6	61.8	218.4	32.6	31.7	31.5	37.4	38.3
	m	307	391	491	130	338	367	350	1147	2443	137	254	339	422	311	405	198

Table 2. Quantitative results on intermediate models.

		Asian dragon	Castle	Church	Courthouse	Euler	Fertility-Fine	Full thing	Meeting Room	Navis	Tower of Pi
Type		F	U	I	U	I	F	S	I	I	F
Origin		Laser	CAD	Laser	Laser	Laser	Laser	CAD	Laser	Laser	CAD
#p		3.6M	737K	31.1M	1.9M	2.7M	242K	1.4M	3.1M	3.6M	2.9M
#s		2712	735	1188	2716	887	812	1648	1655	749	10834
ϵ		0.25	0.1	0.8	0.8	0.1	0.2	0.2	0.1	0.1	0.1
σ		0.004	0.001	0.005	0.002	0.004	0.03	0.004	0.003	0.005	0.001
Ours	$\mathbf{e_A}$	**0.065**	**0.006**	0.460	0.134	**0.069**	**0.061**	**0.064**	0.198	**0.061**	**0.020**
	$\mathbf{e_S}$	**0.069**	**0.038**	**0.292**	**0.183**	**0.108**	**0.095**	**0.064**	**0.146**	**0.057**	**0.039**
	#f	3132	711	394	1795	1317	998	1807	1491	457	12059
	t	1099	268.8	1040	375.4	354.2	1467	462.1	268.4	962.1	2074
	m	1667	321	324	862.4	711.4	1466	552	642.5	1569	3864
SP + QEM	$\mathbf{e_A}$	0.071	0.044	**0.292**	0.155	0.148	0.082	0.170	0.299	0.167	0.135
	$\mathbf{e_S}$	0.072	0.082	0.345	0.245	0.237	0.117	0.141	0.204	0.146	0.098
	#f	3132	712	394	1795	1318	1000	1806	1490	458	12050
	t	27.6	100.5	42.8	44.3	40.5	41.3	67.3	65.5	36.4	205.8
	m	300	1127	558	352	639	361	595	869	646	1422
SP + VSA	$\mathbf{e_A}$	0.125	0.098	0.569	0.254	0.242	0.183	0.378	0.202	0.192	0.295
	$\mathbf{e_S}$	0.108	0.110	0.377	0.262	0.240	0.243	0.245	0.160	0.164	0.241
	#f	3351	1894	494	1818	4070	982	4368	4949	522	27285
	t	49.9	146.1	47.1	67.8	39.7	30	104	105.8	46.3	365.1
	m	594	1975	588	876	870	361	1455	1328	692	4176
SP + SAMD-CP	$\mathbf{e_A}$	0.114	0.065	0.187	0.154	0.169	0.150	0.117	0.330	0.098	0.104
	$\mathbf{e_S}$	0.101	0.078	0.214	0.195	0.166	0.152	0.094	0.215	0.079	0.110
	#f	3672	1600	1924	1835	1450	1709	10046	1552	1898	30654
	t	30.4	77.5	40.5	41.6	38.1	30.7	64.8	55.3	35.8	166.8
	m	300	1127	558	352	639	361	595	869	646	1422
SP + SAMD	$\mathbf{e_A}$	0.088	0.073	0.238	**0.113**	0.128	0.128	0.371	**0.185**	0.113	0.194
	$\mathbf{e_S}$	0.094	0.095	**0.274**	0.223	0.199	0.163	0.258	0.168	0.114	0.205
	#f	3132	733	396	1800	1318	999	1809	1491	457	12060
	t	39	90.7	121.9	44.7	41	37.7	495.2	58	66.4	731.1
	m	300	1127	558	352	639	361	595	869	646	1422

Table 3. Quantitative results on advanced models.

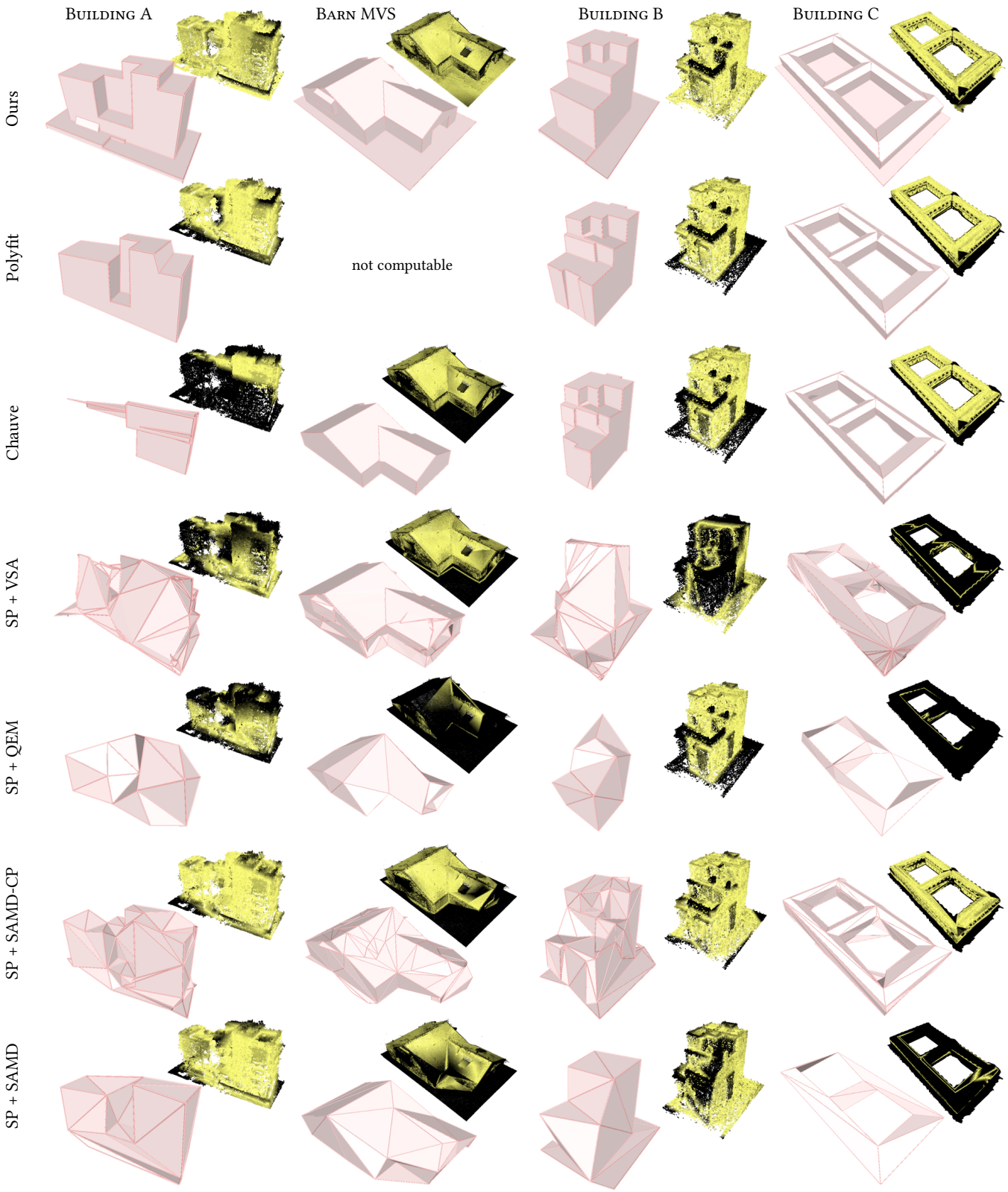

Fig. 2. Qualitative comparisons on simple models (part 1/4). The error maps correspond to the Hausdorff error from input points to output models and ranges from 0 (yellow) to ϵ or higher (black), ϵ being the fitting tolerance parameter for the detection of planar shapes (fixed similarly for each model).

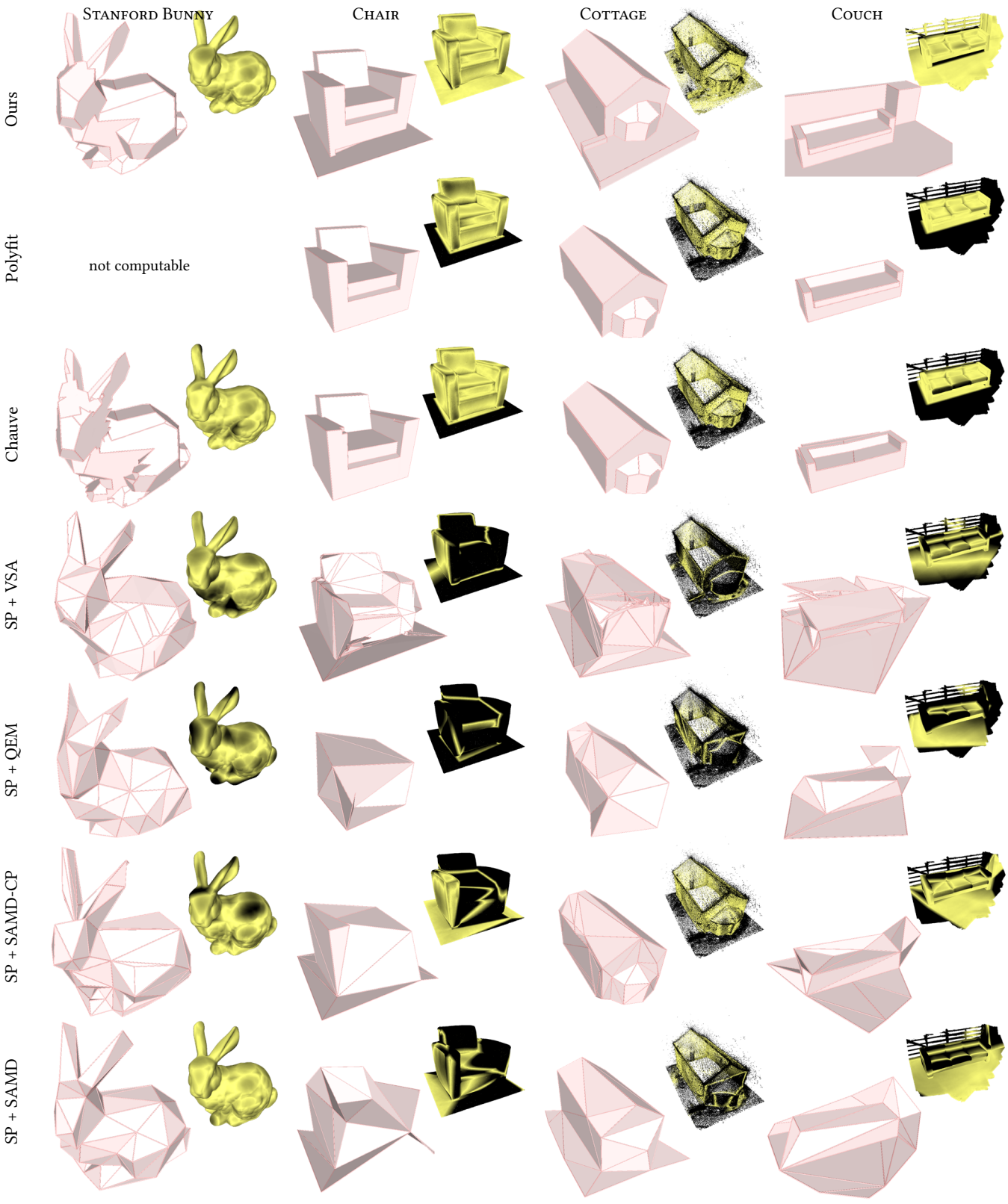

Fig. 3. Qualitative comparisons on simple models (part 2/4).

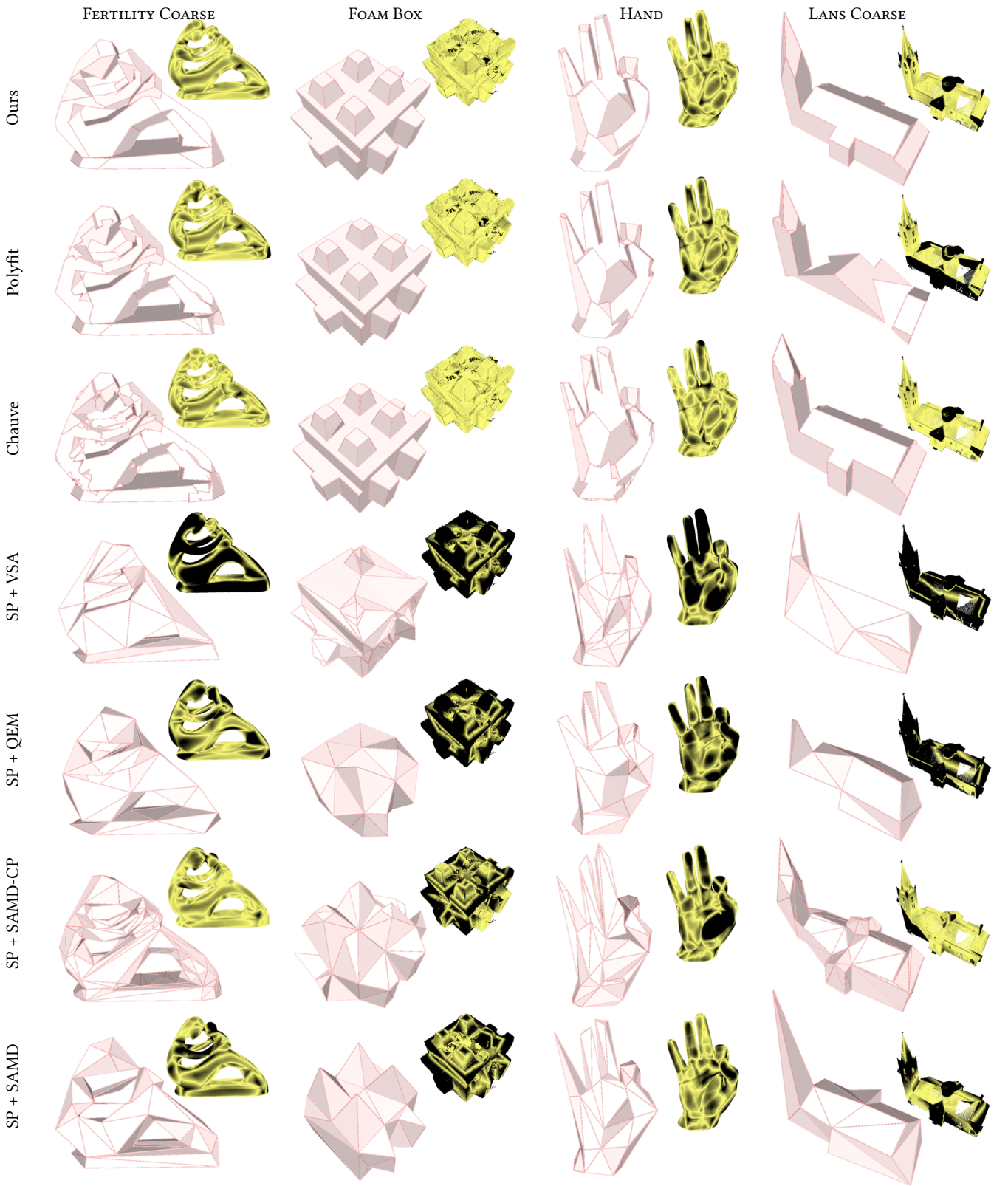

Fig. 4. Qualitative comparisons on simple models (part 3/4).

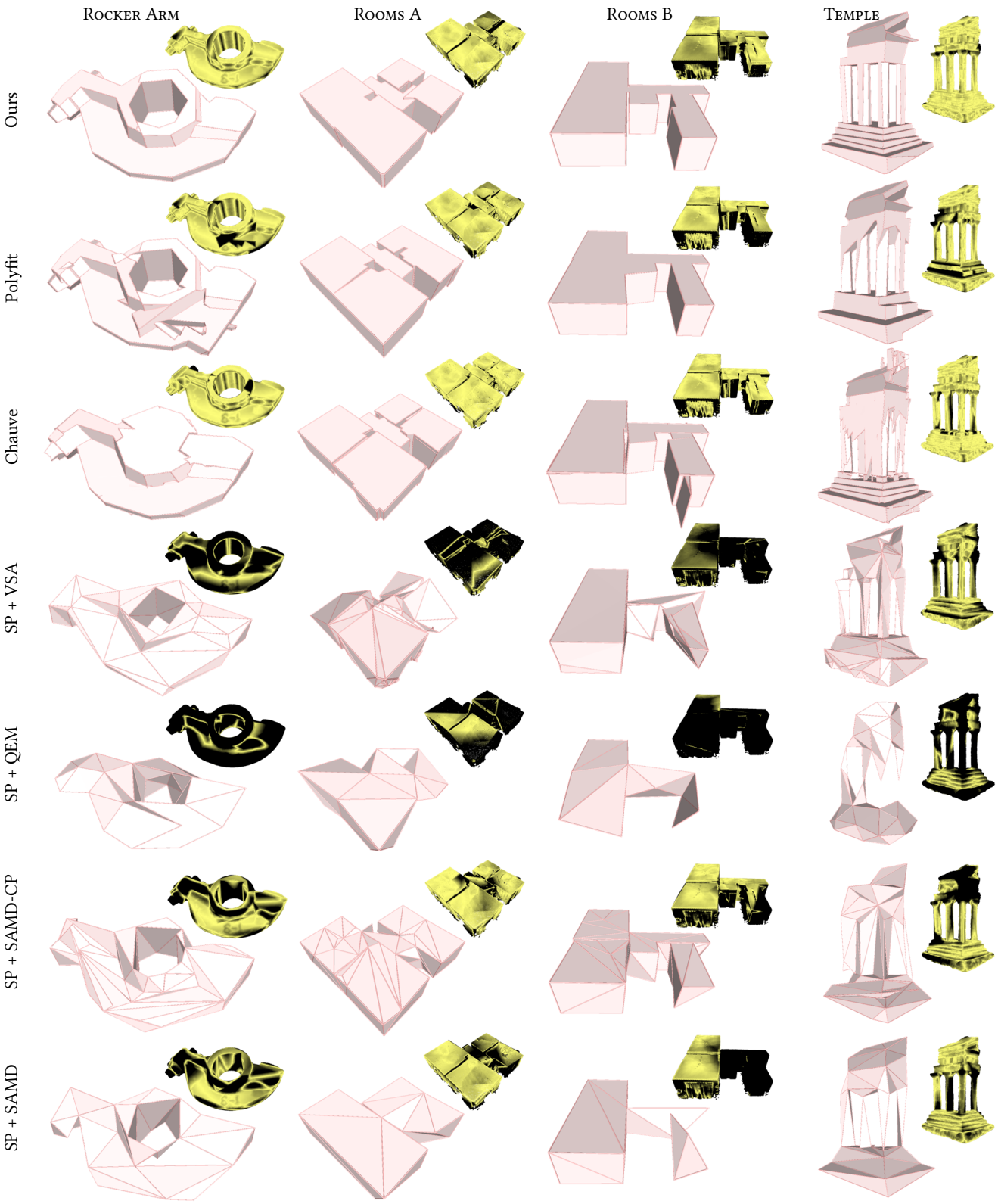

Fig. 5. Qualitative comparisons on simple models (part 4/4).

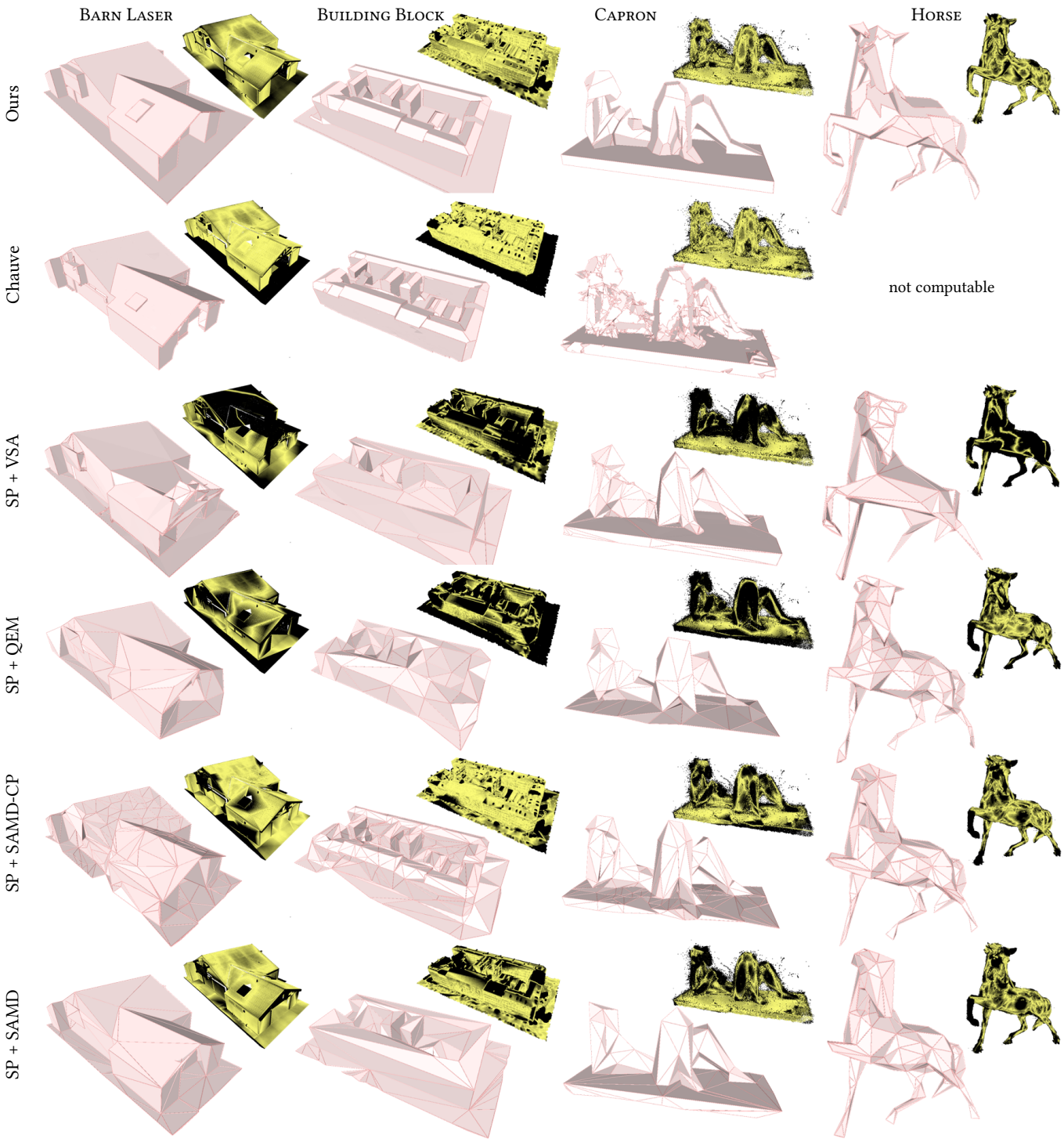

Fig. 6. Qualitative comparisons on intermediate models (part 1/4).

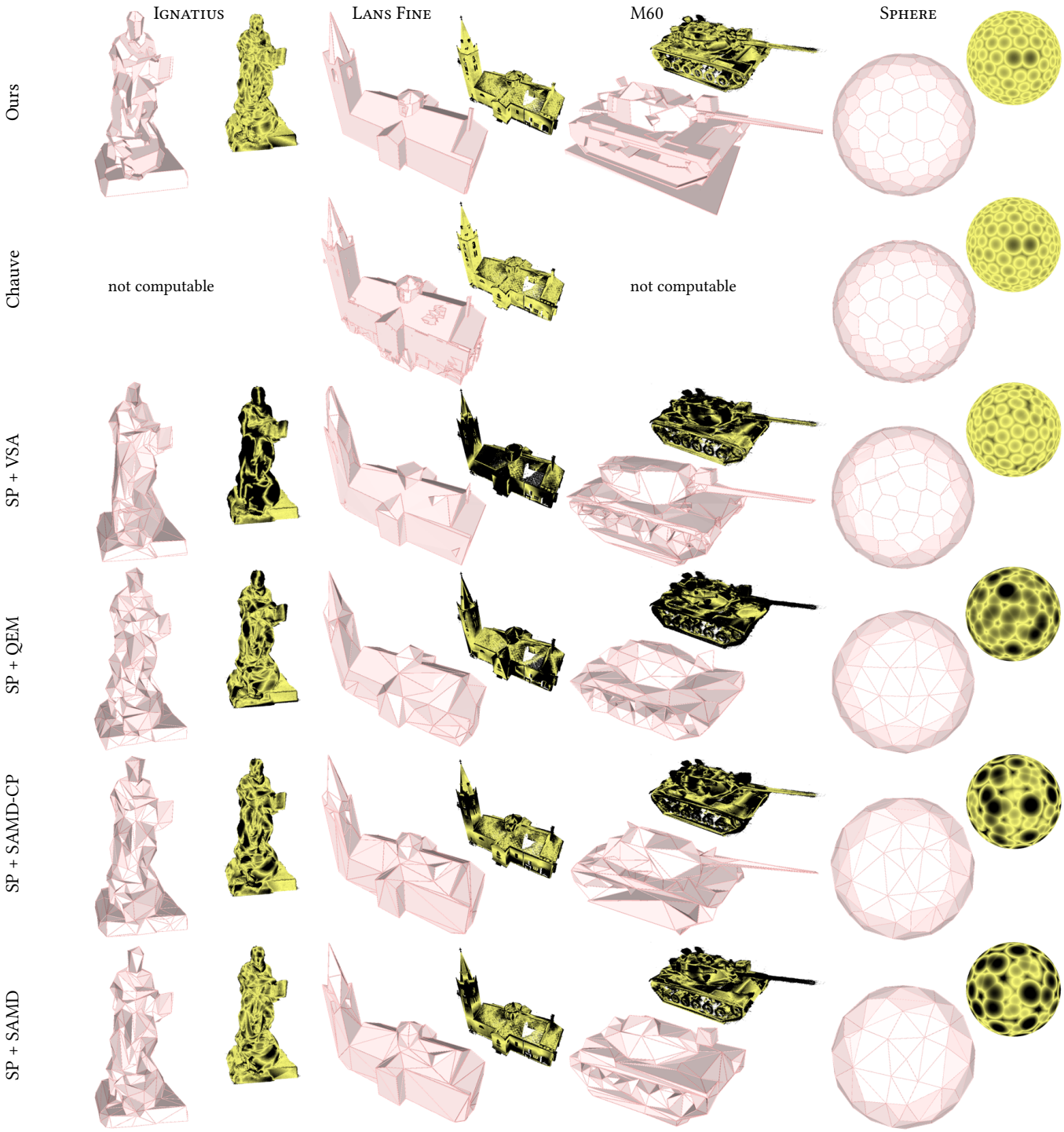

Fig. 7. Qualitative comparisons on intermediate models (part 2/4).

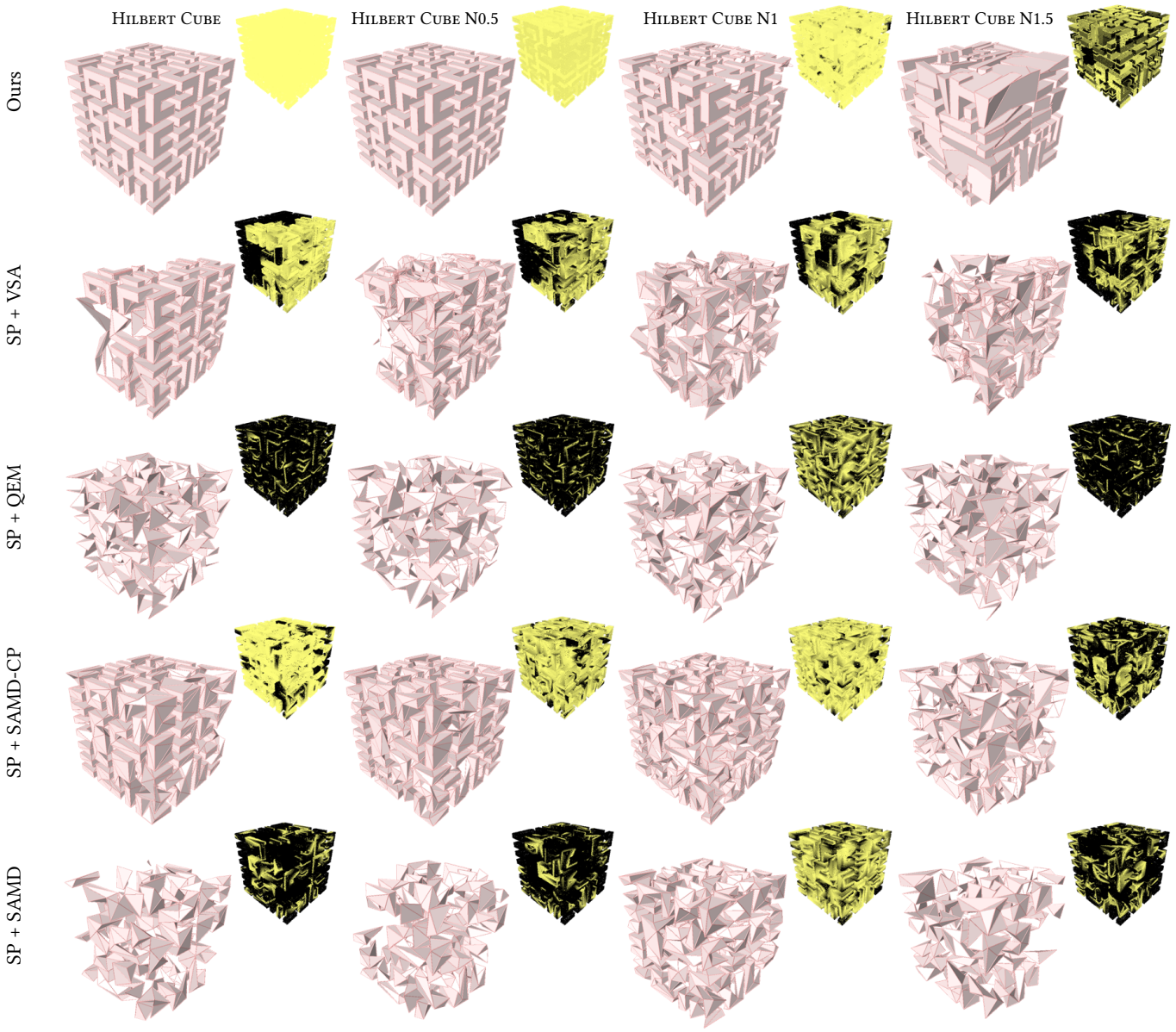

Fig. 8. Qualitative comparisons on intermediate models (part 3/4).

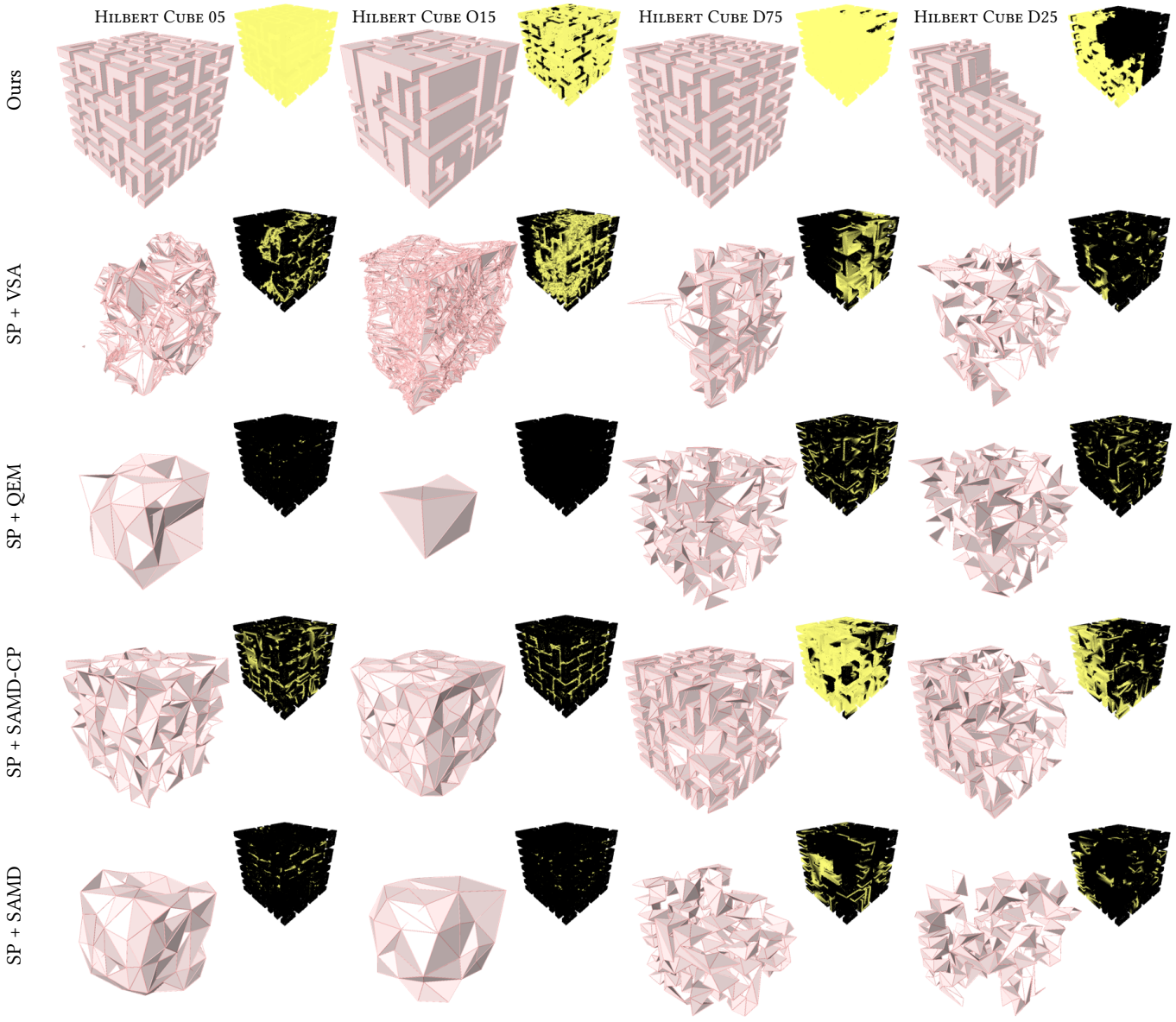

Fig. 9. Qualitative comparisons on intermediate models (part 4/4).

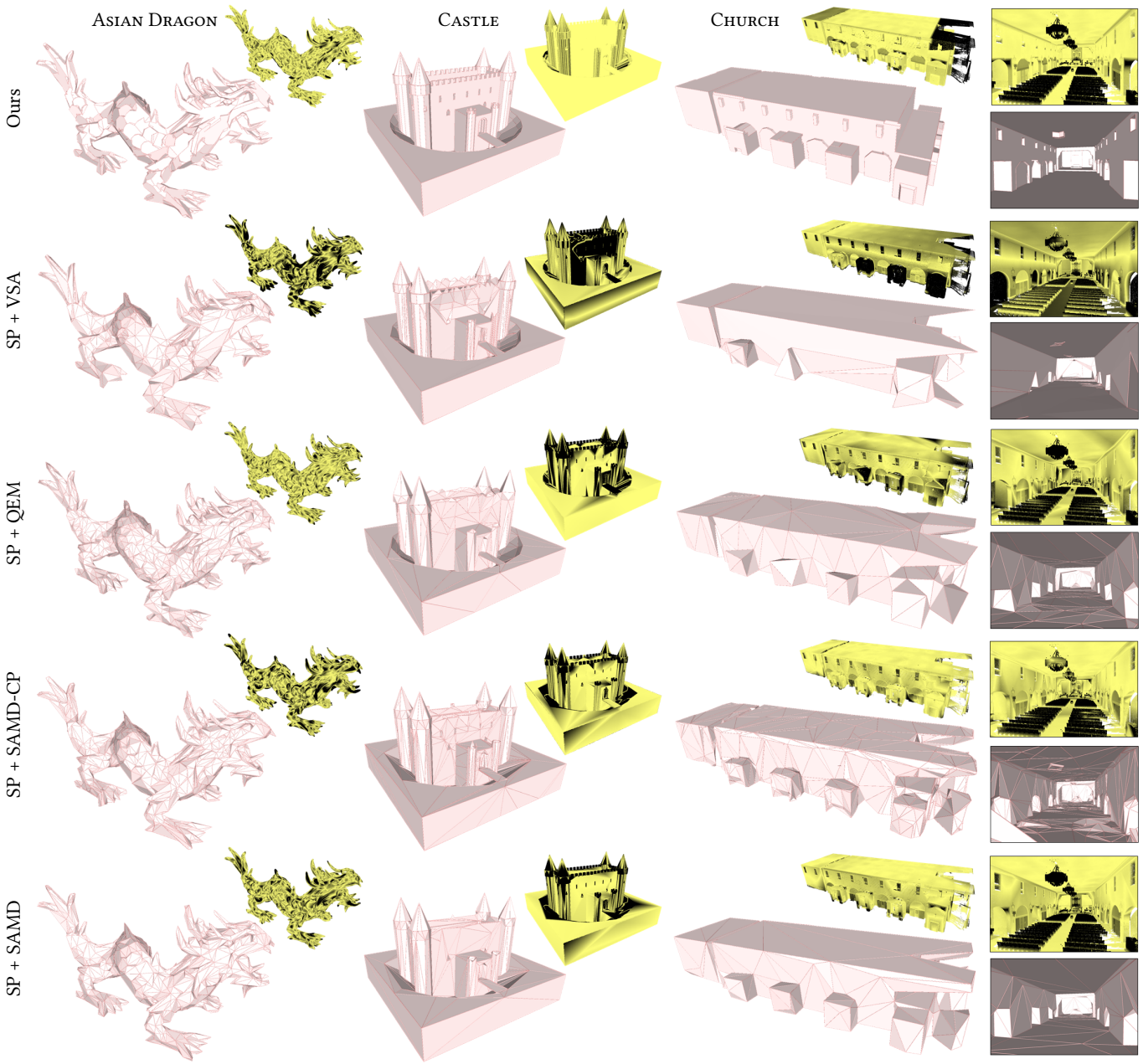

Fig. 10. Qualitative comparisons on advanced models (part 1/4).

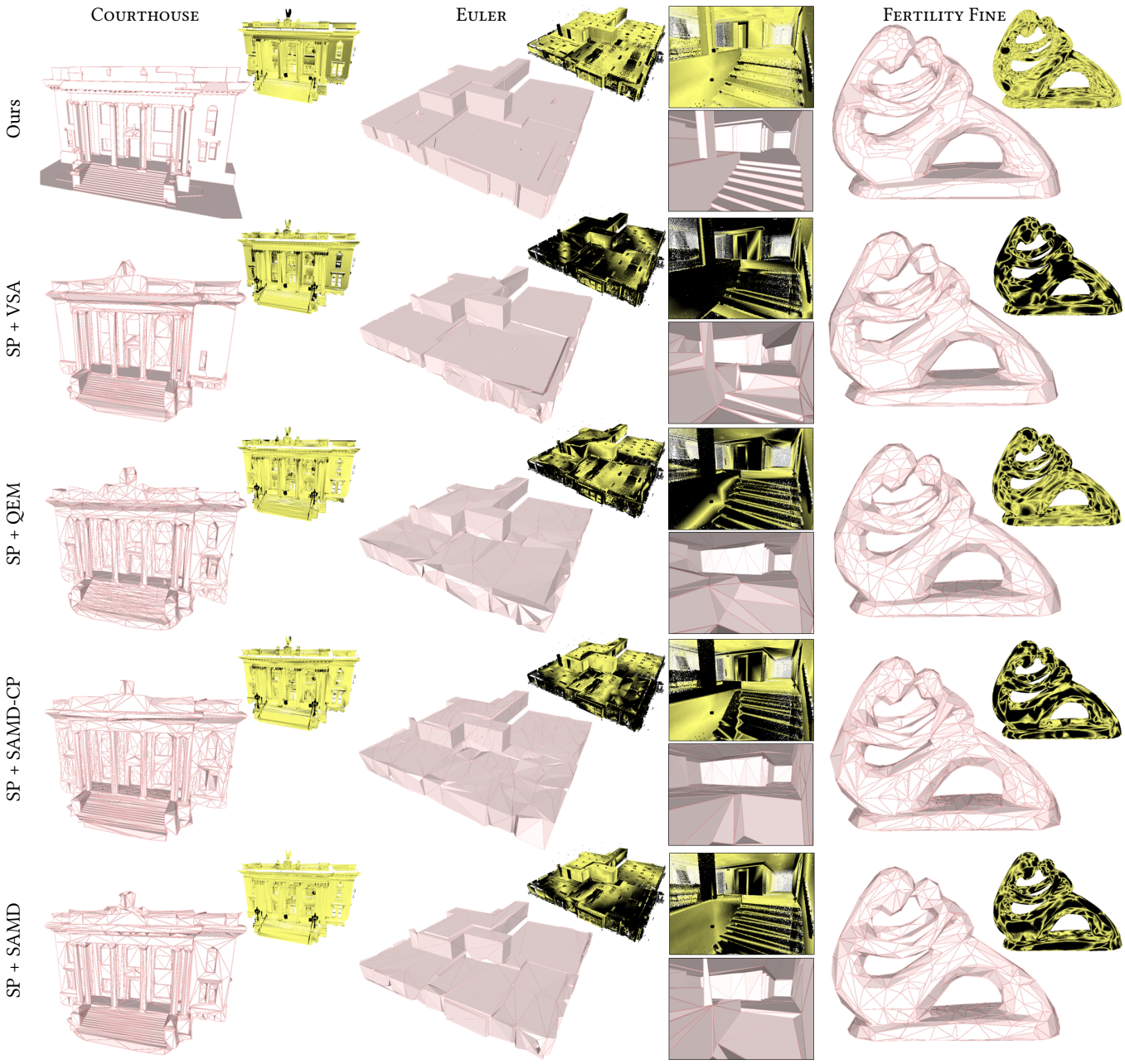

Fig. 11. Qualitative comparisons on advanced models (part 2/4).

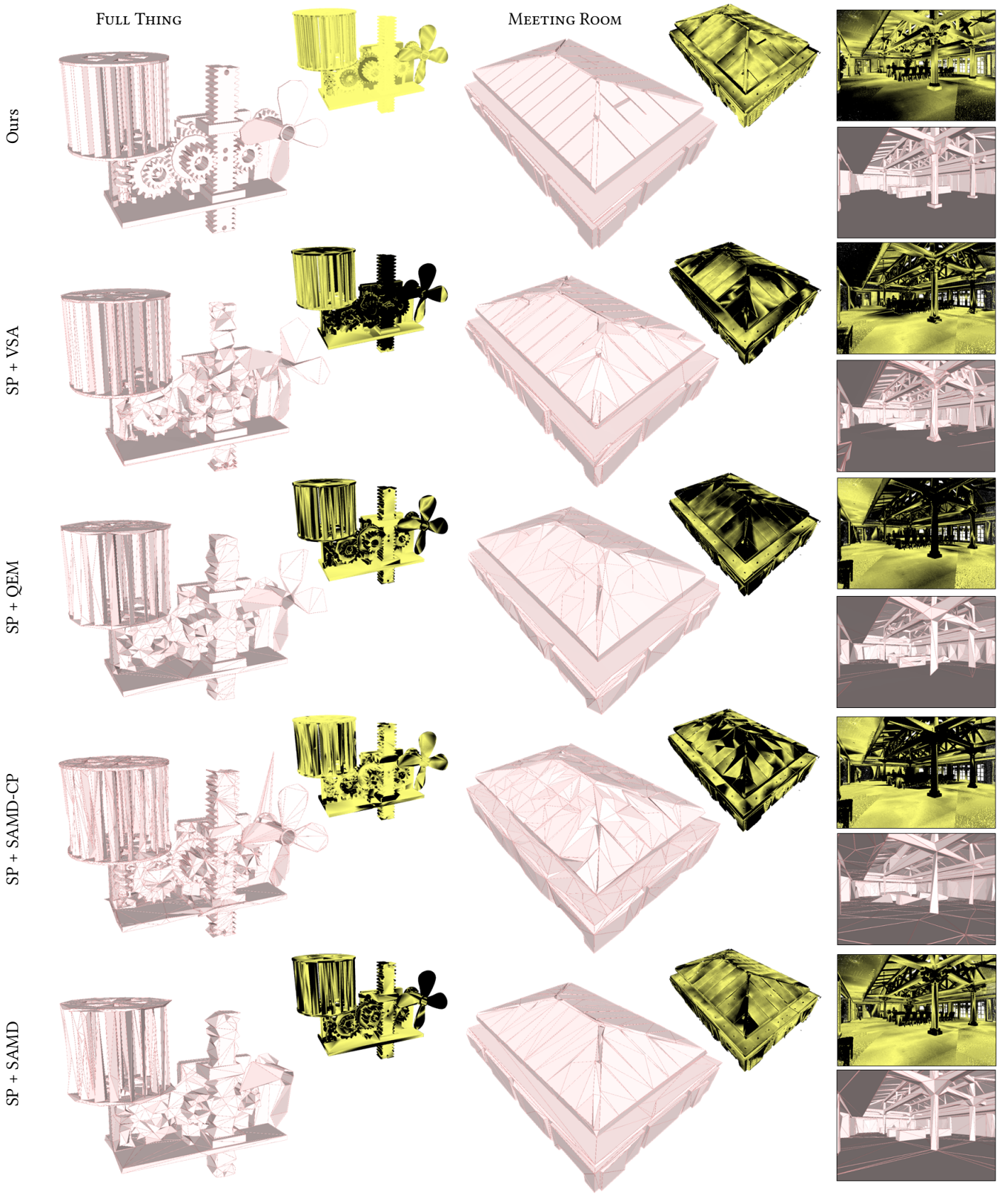

Fig. 12. Qualitative comparisons on advanced models (part 3/4).

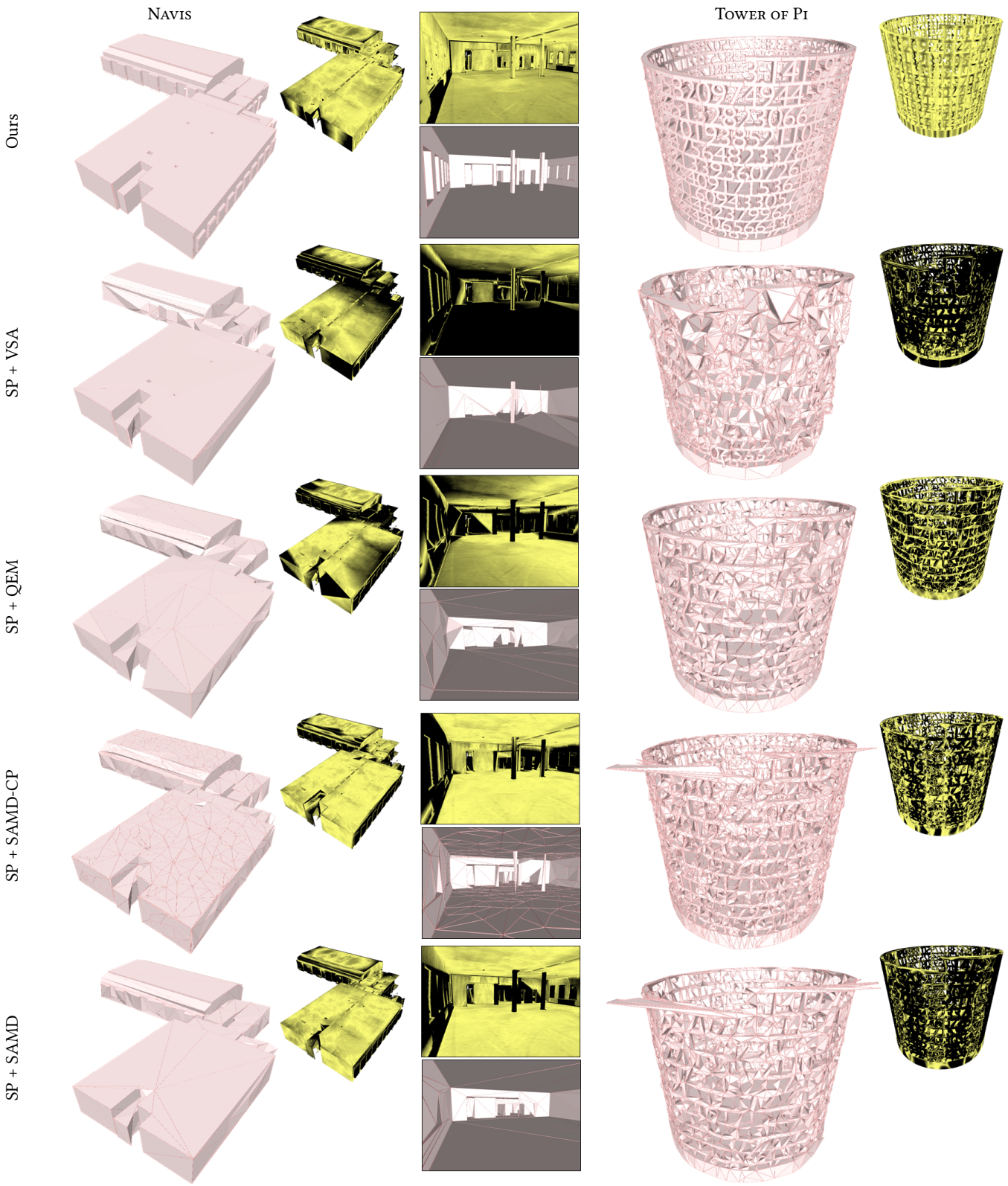

Fig. 13. Qualitative comparisons on advanced models (part 4/4).