Supplementary Material: Joint 3D Object and Layout Inference from a single RGB-D Image

Andreas Geiger[1] Chaohui Wang[1,2]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]Université Paris-Est, Marne-la-Vallée, Paris, France

Abstract. In this supplementary **document** we illustrate the 3D CAD model database we use, give details on the sparse factor-to-variable message calculation and provide additional quantitative results in terms of reconstruction, segmentation and 2D/3D detection performance. Furthermore, we show additional qualitative results and comparisons with respect to Lin et al. [10] on randomly selected scenes from the NYUv2 dataset.
In addition, the supplementary **video** visualizes the proposals, the inference process and our estimates in the form of wireframe projections and 3D popups for several scenes. To illustrate how inference progresses, we show the estimated max-marginal configuration at different iterations of the belief propagation algorithm. The video can be viewed with the VLC media player, available at: http://www.videolan.org/vlc/.

1 Related Work and 3D CAD Model Database

Table 1 shows a table comparing related methods for scene understanding to the proposed method in terms of their properties. Fig. 1 shows scaled (i.e., normalized) versions of the 3D CAD models we collected from Google 3D Warehouse[1]. For consistency with the object annotations from [3], we selected the same 3D models for the object classes *bed*, *table*, *shelf*, *sofa* and *chair*. Note that this doesn't affect our evaluation as we compute the intersection-over-union of the 3D bounding box associated with the estimated 3D models with respect to the corresponding ground truth 3D bounding box.

[1] https://3dwarehouse.sketchup.com/

	Zhang et al. [16]	Choi et al. [2]	Pero et al. [11]	Hedau et al. [4]	Schwing et al. [14]	Lim et al. [9]	Aubry et al. [1]	Satkin et al. [13]	Kim et al. [7]	Jiang et al. [6]	Jia et al. [5]	Lin et al. [10]	Our model
Input	360°	2D	2D	2D	2D	2D	2D	2D	3D	3D	3D	3D	3D
#Object classes	16	6	8	1	1	8	1	2	9	1	11	21	21
#Layout classes	1	5	5	5	5	–	–	–	4	–	–	–	3
#Scene classes	–	3	–	–	–	–	–	–	–	–	–	12	–
Context model	✓	✓	✓	✓	✓			✓			✓	✓	✓
Visibility model			✓	✓	✓				✓	✓	✓		✓
Occlusion model				✓	✓								✓
CAD models						✓	✓	✓					✓
Beyond MHW		✓				✓	✓			✓	✓	✓	✓

Table 1. Related Work. This table gives an overview over the most related work on indoor scene understanding with 3D models. "Beyond MHW" expresses the fact that the object orientations are not limited to the Manhattan world assumption.

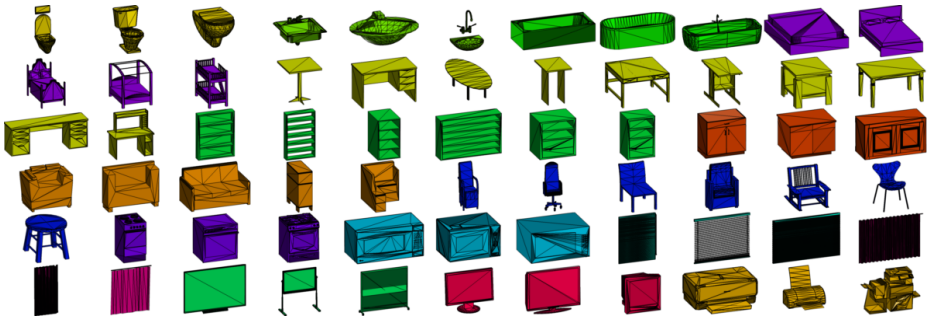


Fig. 1. 3D Model Database. For each object category which can't be well represented by a cuboid model we use one out of 66 CAD models collected from Google 3D Warehouse. The colors correspond to the different semantic categories.

2 Sparse Factor-to-Variable Message Calculation

Finding the minimum of $E(\mathbf{x}|\mathbf{I})$ with respect to $\mathbf{x}$ in the main paper is an NP hard problem and we employ max-product loopy belief propagation to find an approximate solution. For numerical stability we follow common practice and make use of the equivalent min-sum formulation where messages between variable i and the factor corresponding to clique c are passed as follows:

$$m_{c\to i}^{t}(x_i) = \min_{\mathbf{x}_c^{-i}} \left(f_c(\mathbf{x}_c) + \sum_{j\in c\setminus\{i\}} m_{j\to c}^{t-1}(x_j) \right) \tag{1}$$

$$m_{i\to c}^{t}(x_i) = \sum_{c'\in\mathcal{N}_i\setminus\{c\}} m_{c'\to i}^{t-1}(x_i) \tag{2}$$

Here, $\mathcal{N}_i$ denotes all factors involving variable i and $\mathbf{x}_c^{-i}$ denotes all variables involved in clique c except variable i. For high-order cliques the computational complexity of computing the factor-to-variable message in Eq. 1 is $O(D^N)$ in general, where N is the number of variables in the clique and D denotes the number of labels[2]. As the size of our high-order cliques easily exceeds $N = 20$ or $N = 30$, brute force calculation is clearly intractable. We approach this problem by taking advantage of the sparsity in our high-order potentials which can be written as[3]

$$f(\mathbf{x}) = \begin{cases} \xi(\mathbf{x}) & \text{if } \mathbf{x}\in\mathcal{S} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\xi(\mathbf{x}) : D^N \to \mathbb{R}$ is an arbitrary function and $\mathcal{S}$ is a set of *special states* with $|\mathcal{S}| \ll 2^N$. Note that in contrast to the high-order potentials considered in [8,12], we allow negative as well as positive values for $\xi(\mathbf{x})$. The latter case is required by our superpixel-object consistency potentials (see paper, Eq. 5).

In Algorithm 1, we provide an efficient recursion (an illustration of which is given below using a simple example) to exactly compute factor-to-variable messages (Eq. 1) for the class of potentials specified in Eq. 3 given a limited computational budget. Without loss of generality, we assume that the message is passed to the first node in the clique. For other cases we simply switch the target variable with the first variable in the clique. Moreover, we assume that the variables in the clique under consideration are indexed from 1 to N for clarity of presentation.

To illustrate the idea, let us look at the simple case of a clique c with $N = 3$ binary variables (i.e., $D = 2$). We first consider the trivial case where no special state is present, i.e., $|\mathcal{S}| = 0$ and $f(\cdot) \equiv 0$. In this case, Eq. 1 can be computed efficiently by swapping the min and the sum operators: we obtain $m_{c\to 1}^{t}(\cdot) \equiv$

[2] For clarity, here we assume that all variables have the same number of labels D while our algorithm has no such restriction.

[3] We define $f(\mathbf{x}) = 0$ for non-special states as we can always add a constant to $f(\mathbf{x})$ without changing the optimization problem.

Algorithm 1 Sparse Factor-to-Variable Message Passing

Input: Factor: $f_c(\cdot)$, Target variable $i = 1$, Input messages: $m_{i\to c}^{t-1}(\cdot)\ \forall i \in \{2,\dots,N\}$
Output: Factor-to-variable message: $m_{c\to 1}^{t}(\cdot)$
$\forall i \in \{2,\dots,N\} : \mu_i \leftarrow \min_x(m_{i\to c}^{t-1}(x))$ ▷ precompute all min-messages
$\forall x : m_{c\to 1}^{t}(x) \leftarrow +\infty$ ▷ initialize output message to $+\infty$
$\mathbf{y} \leftarrow vector(N)$ ▷ $\mathbf{y}$ will be set during the recursion
UpdateMessage($\mathbf{y}$,0) ▷ start recursion
procedure UpdateMessage($\mathbf{y}$,k)
  if $k < N$ **then** ▷ partition contains ≥ 1 state
    if $k = 0 \ \vee \ \mathcal{T}_k(\mathbf{y}) \cap \mathcal{S} \neq \emptyset$ **then** ▷ $k = 0$ or $\mathcal{T}_k(\mathbf{y})$ contains ≥ 1 special state
      for all $d \leftarrow 1,\dots,D$ **do** ▷ for all sub-partitions
        $y_{k+1} \leftarrow d$ ▷ specify sub-partition
        UpdateMessage($\mathbf{y}$,k+1) ▷ continue recursion
    else ▷ compute minimum for partition $\mathcal{T}_k(\mathbf{y})$
      $m_{c\to 1}^{t}(y_1) = \min\left(m_{c\to 1}^{t}(y_1), g_k(\mathbf{y})\right)$ ▷ $g_k(\cdot)$ is defined in Eq. 5
  else ▷ update minimum for single state $T_N(\mathbf{y})$
    $m_{c\to 1}^{t}(y_1) = \min\left(m_{c\to 1}^{t}(y_1), f(\mathbf{y}) + \sum_{i\in\{2,\dots,N\}} m_{i\to c}^{t-1}(y_i)\right)$

$\sum_{i\in\{2,3\}} \mu_i$ with min-messages defined as $\mu_i = \min_x\left(m_{i\to c}^{t-1}(x)\right)$. Next, let us consider the presence of a single special state: $\mathcal{S} = \{\mathbf{s}\}$, $\mathbf{s} = (1,1,1)^T$. Let

$$\mathcal{T}_k(\mathbf{y}) = \{\mathbf{x} | x_i = y_i, i \leq k\} \subset \{1,\dots,D\}^N \tag{4}$$

denote the subset of states for which the first k variables in $\mathbf{x}$ are equal to those in $\mathbf{y}$. For $N = 3$ this allows us to partition the state space into 4 disjoint sets $\mathcal{T}_1(0) \cup \mathcal{T}_2(1,0) \cup \mathcal{T}_3(1,1,0) \cup \{\mathbf{s}\}$ and we obtain $m_{c\to 1}^{t}(\cdot)$ as

$$m_{c\to 1}^{t}(x) = \begin{cases} g_1(0) & \text{if } x = 0 \\ \min\left(g_2(1,0), g_3(1,1,0), \xi(\mathbf{s}) + \sum_{i\in\{2,3\}} m_{i\to c}^{t-1}(s_i)\right) & \text{otherwise} \end{cases}$$

where $g_k(\mathbf{y})$ denotes the minimum of partition $\mathcal{T}_k(\mathbf{y})$,

$$g_k(\mathbf{y}) = \sum_{i\in\{2,\dots,k\}} m_{i\to c}^{t-1}(y_i) + \sum_{i\in\{k+1,\dots,N\}} \mu_i \tag{5}$$

which can be computed in linear time. This partitioning naturally suggests a recursive implementation which we specify in Algorithm 1: we iterate over the variables involved in the factor, update the minimum using the pre-computed min-messages $\{\mu_i\}$ if no special state is included in the current partition, and continue the recursion otherwise. By adaptively exploring the state space, this reduces the time complexity for computing sparse high-order factor-to-variable messages from $O(D^N)$ to $O(DN^2)$. In the presence of multiple special states the complexity depends on the distribution of these states and the worst-case complexity is $O(|\mathcal{S}|DN^2)$.

3 Additional Quantitative Results

2D Object Detection In addition to the *3D object detection* results shown in Table 2 of the paper submission we investigate the performance of our method on *2D object detection*, i.e., we project all 3D bounding boxes into the image and evaluate detection performance using the traditional 2D intersection-over-union (IOU) criterion. Due to the large degree of clutter and occlusion in the scenes, we use a IOU threshold of 30% for considering an object detection as correct. As shown in Table 3 (unclipped case) and Table 3 (clipped case), our method performs about equally for CAD models and cuboids, due to the perspective "flattening" of the results. However, note that for both, the clipped as well as the unclipped case, our method compares favorably with respect to the approach of Lin et al. [10].

	mantel	counter	toilet	sink	bathtub	bed	headboard	table	shelf	cabinet	sofa	chair	chest	refrigerator	oven	microwave	blinds	curtain	board	monitor	printer	overall
#obj	11	127	30	40	25	170	23	466	253	544	229	751	143	43	30	45	112	98	52	100	30	3322
[10] - 8 Proposals	0	28	42	**17**	0	36	7	25	29	32	37	25	26	7	0	11	32	28	20	**34**	**11**	27.46
[10] - 15 Proposals	**17**	32	41	15	7	32	5	28	25	30	36	26	32	12	10	25	27	24	**21**	28	6	27.48
[10] - 30 Proposals	0	29	36	16	29	28	6	27	23	29	33	26	30	16	6	**26**	24	22	21	28	0	26.40
Base-Det-Cuboid	0	20	12	3	17	14	23	12	7	11	12	10	19	21	**12**	9	7	8	8	12	3	11.40
Base-NMS-Cuboid	0	29	49	6	7	67	**25**	20	28	38	46	29	28	20	6	15	**32**	33	15	28	7	31.07
NoOcclusion-Cuboid	0	29	49	9	22	65	13	27	32	43	46	**38**	27	29	6	0	26	30	19	19	11	**34.54**
NoContext-Cuboid	0	**36**	51	3	32	62	13	29	**32**	42	43	33	28	**34**	11	4	10	10	10	21	6	32.33
FullModel-Cuboid	0	27	49	9	23	66	14	27	**32**	**44**	46	34	31	32	6	0	27	31	18	23	5	34.29
Base-Det-CAD	0	20	13	4	13	11	23	12	8	11	12	13	19	21	**12**	9	12	8	8	12	3	12.05
Base-NMS-CAD	0	26	47	8	0	63	**25**	23	28	38	43	26	29	24	11	14	22	**33**	15	26	7	30.02
NoOcclusion-CAD	0	26	52	8	31	67	0	28	26	42	53	30	28	32	6	4	9	31	18	20	0	32.40
NoContext-CAD	0	32	54	6	**33**	61	14	31	27	42	50	33	31	30	6	4	12	10	10	20	0	32.60
FullModel-CAD	0	27	**57**	7	31	**73**	15	**32**	27	44	**55**	29	**32**	30	6	4	10	33	16	23	0	33.97
FullModel-CAD-GT	60	59	86	24	70	91	41	61	69	78	89	47	78	83	76	68	20	63	68	51	51	63.92

Table 2. 2D Detection Unclipped. Evaluation in terms of F1 score (%). See text for details.

	mantel	counter	toilet	sink	bathtub	bed	headboard	table	shelf	cabinet	sofa	chair	chest	refrigerator	oven	microwave	blinds	curtain	board	monitor	printer	overall
#obj	10	126	30	36	25	169	23	455	242	534	228	703	137	42	29	40	111	91	50	81	25	3187
[10] - 8 Proposals	0	29	45	**17**	0	36	20	26	**28**	32	36	25	27	7	6	16	**30**	25	**21**	**36**	**13**	27.54
[10] - 15 Proposals	**18**	**33**	43	16	7	32	22	28	24	30	35	26	33	9	5	**31**	25	22	19	27	6	27.31
[10] - 30 Proposals	0	29	38	16	29	28	**22**	27	21	29	32	26	31	13	6	29	23	20	18	28	0	26.25
FullModel-CAD	0	26	**61**	8	**31**	**74**	7	**32**	26	**41**	**53**	**29**	**33**	**30**	**6**	5	7	**30**	17	24	0	**33.35**

Table 3. 2D Detection Clipped. Evaluation in terms of F1 score (%). See text for details.

4 Additional Qualitative Results

The following pages show additional qualitative results on the NYUv2 test set. We compare our estimates against ground truth and the results of Lin et al. [10]. Each subfigure shows: Ground truth, results of [10], our results (top-to-bottom) in form of 3D popups, re-projections, depth maps and semantic segmentations (left-to-right) using the semantic color coding scheme from Fig. 1 in the main paper. For the ground truth we directly show the depth channel of the RGB-D image in the corresponding row of the reconstruction column. A legend illustrating what is shown in the individual subplots is given in Fig. 2

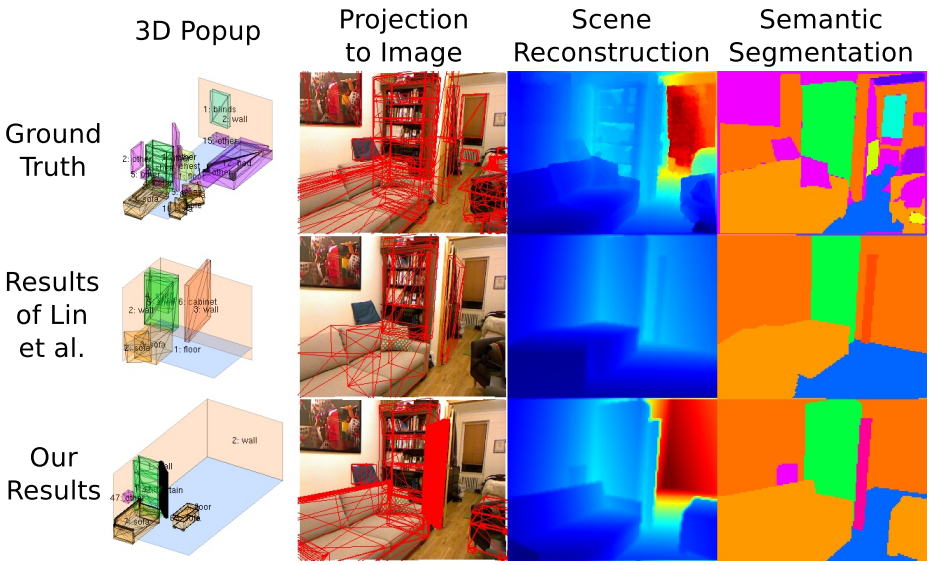
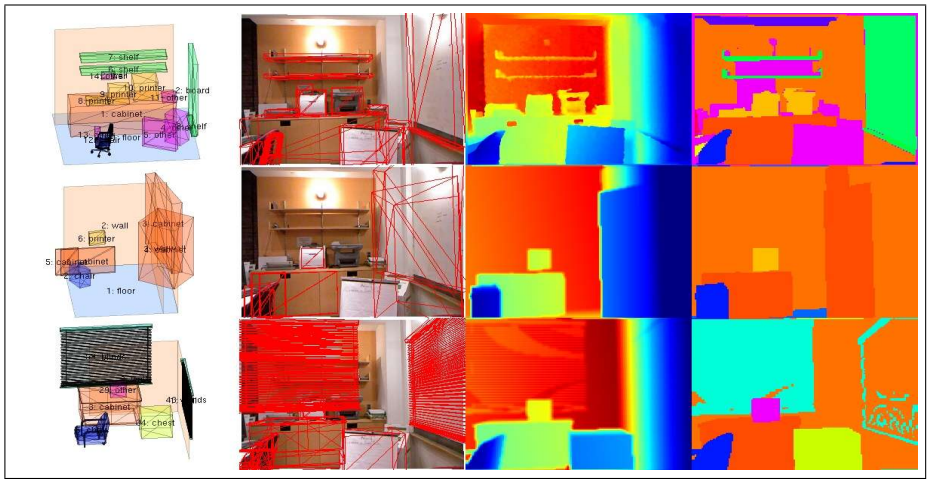
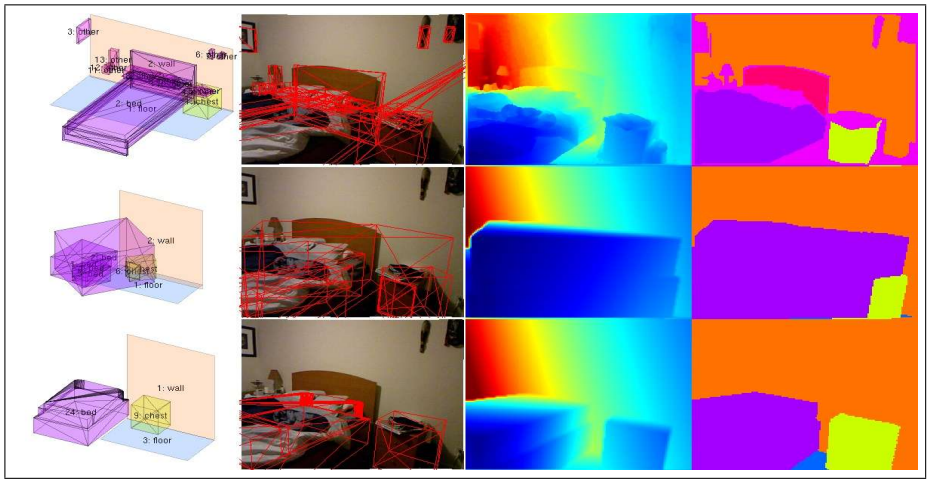
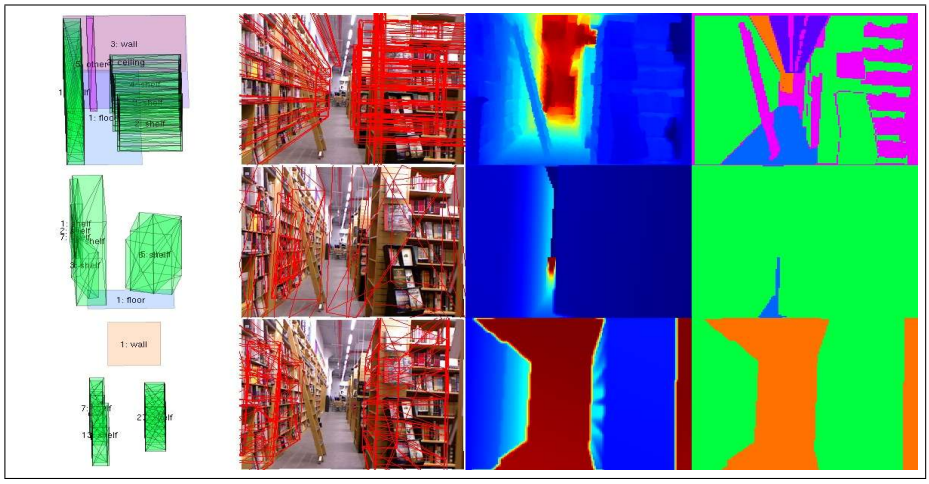
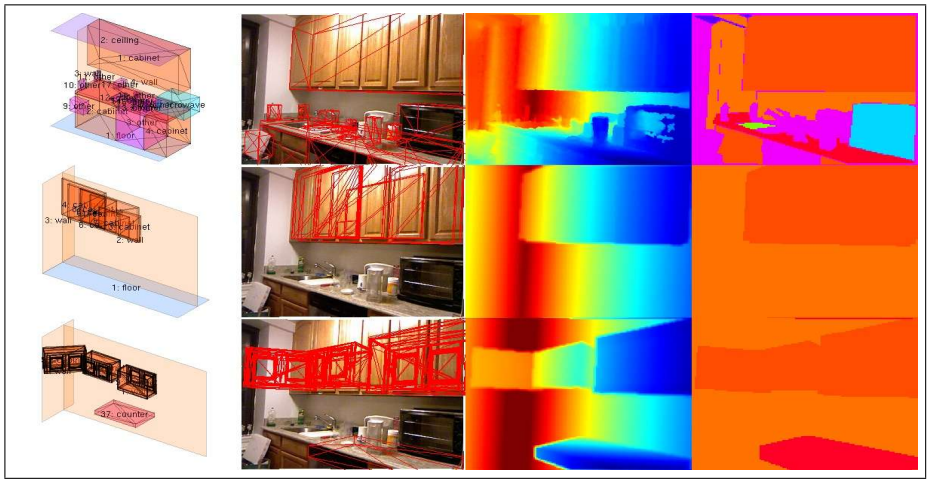


Fig. 2. Additional Qualitative Results Legend.

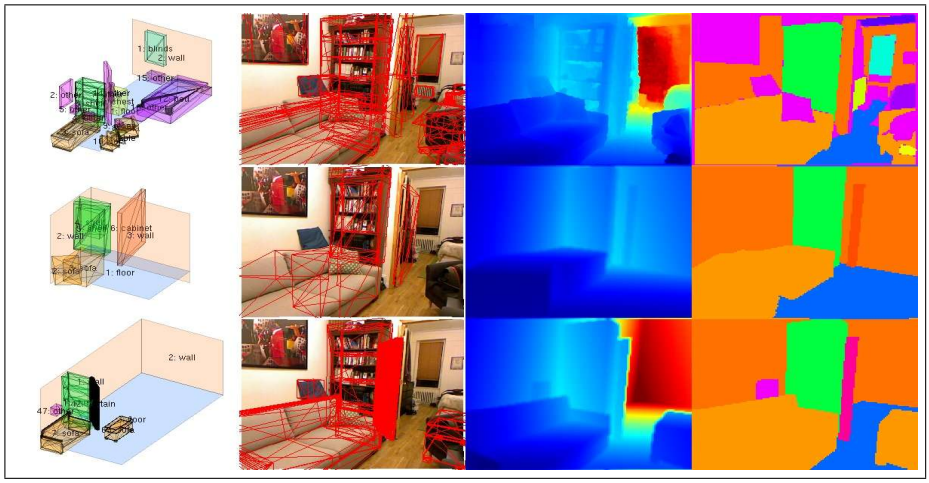

7: shelf
10: printer
2: board
9: printer
11: other
8: printer
1: cabinet
4: night_shelf
12: floor
6: other
2: wall
3: cabinet
6: printer
4: night_shelf
5: cabinet
2: chair
1: floor
29: stool
3: cabinet
34: chest

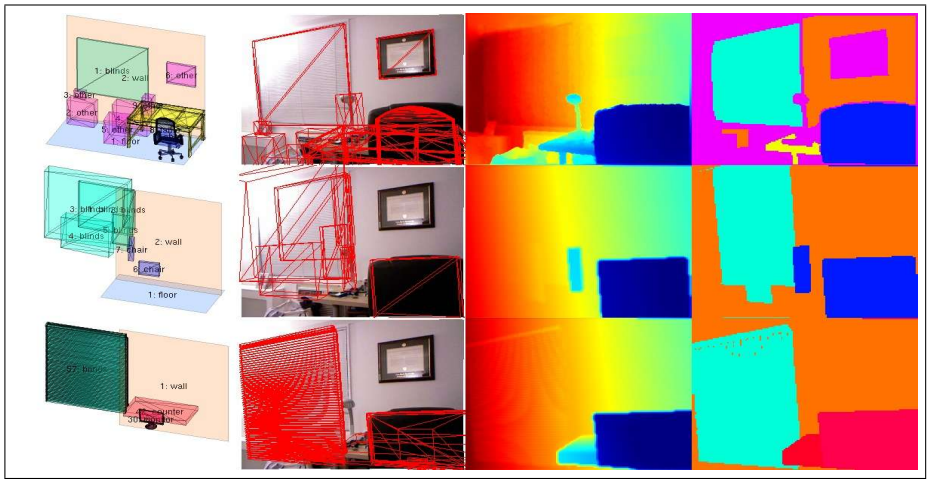

3: other
6: other
13: other
2: wall
2: bed
1: floor
2: wall
1: floor
1: wall
24: bed
9: chest
3: floor

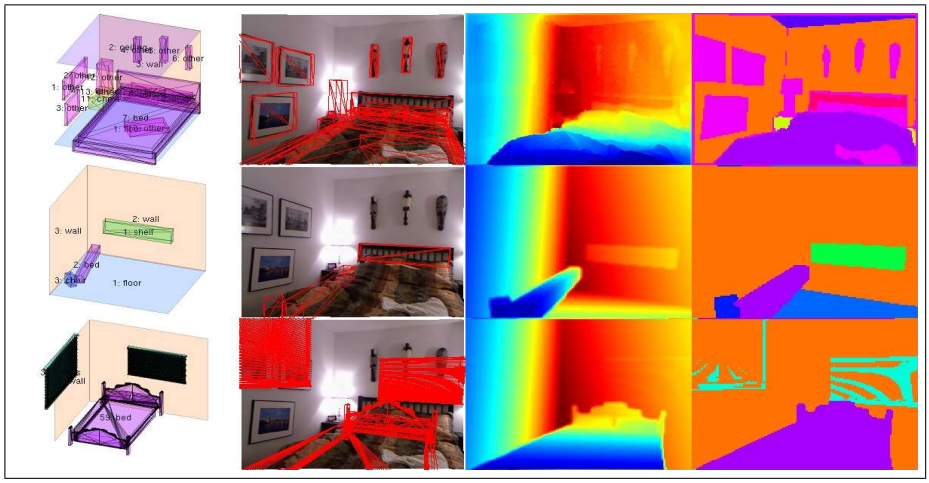

3: wall
2: ceiling
1: floor
2: shelf
1: floor
1: wall

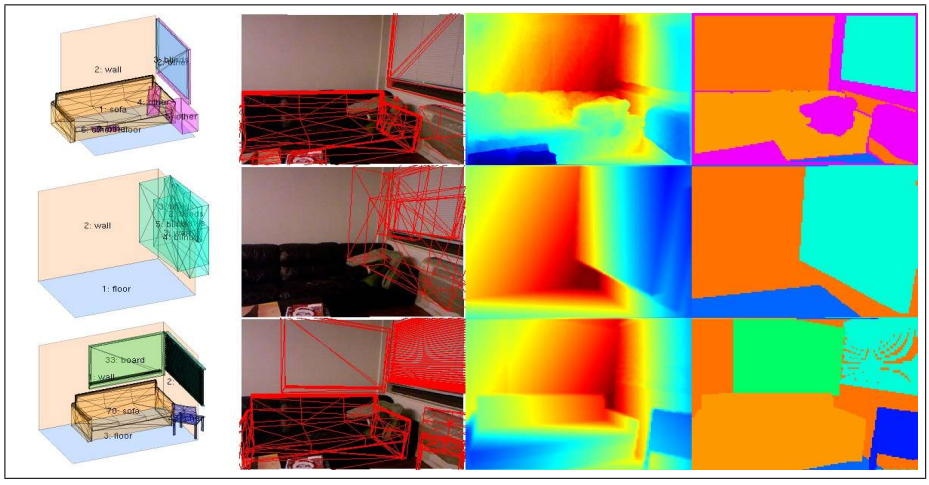

2: ceiling
1: cabinet
1: floor
3: wall
2: wall
1: floor
37: counter

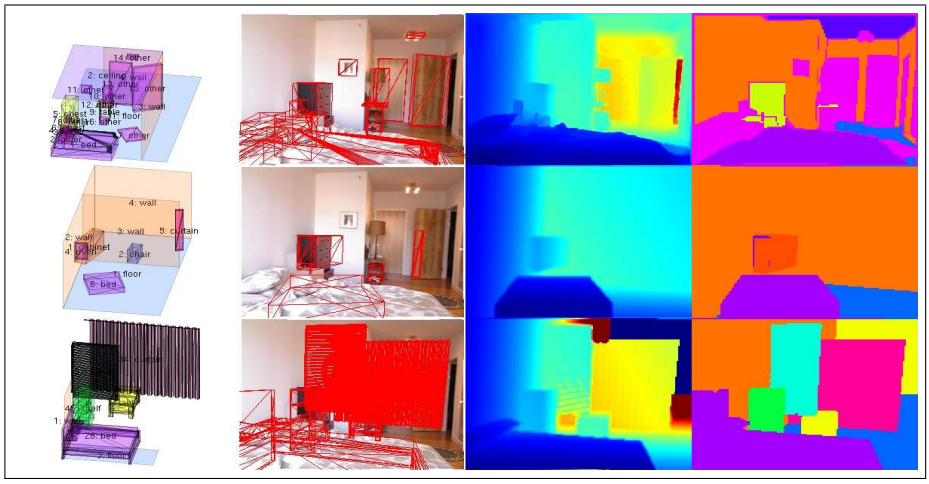

2: monitor
3: other
4: other
2: wall
5: other
1: cabinet
6: other
1: floor
5: chair
1: floor
1: monitor
2: wall
3: floor

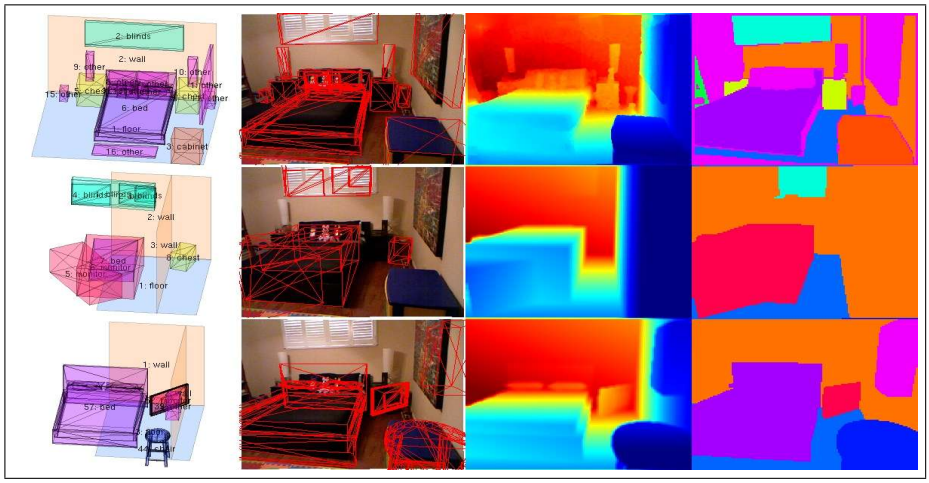

1: blinds
2: wall
15: others
2: wall
6: cabinet
3: wall
1: floor
2: wall
1: wall
1: floor

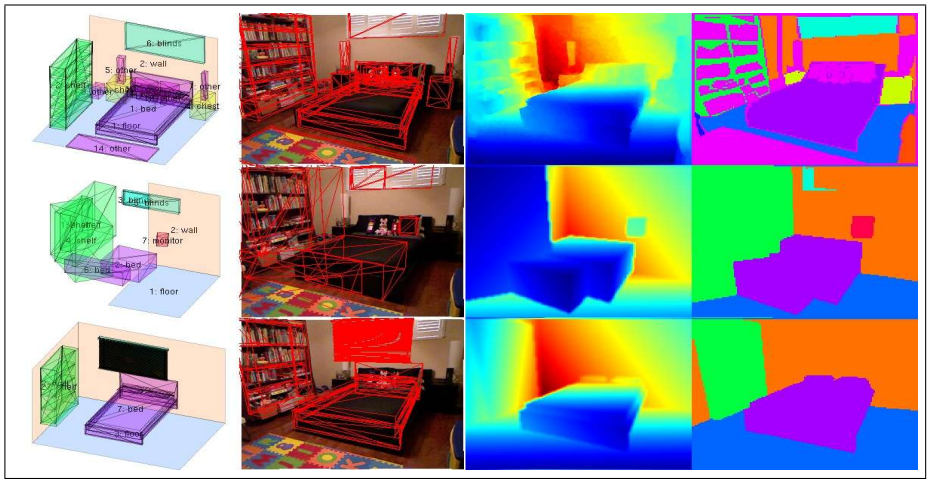

1: blinds
2: wall
6: other
3: other
2: other
1: floor
3: blinds
4: blinds
2: wall
7: chair
8: chair
1: floor
1: wall
1: floor

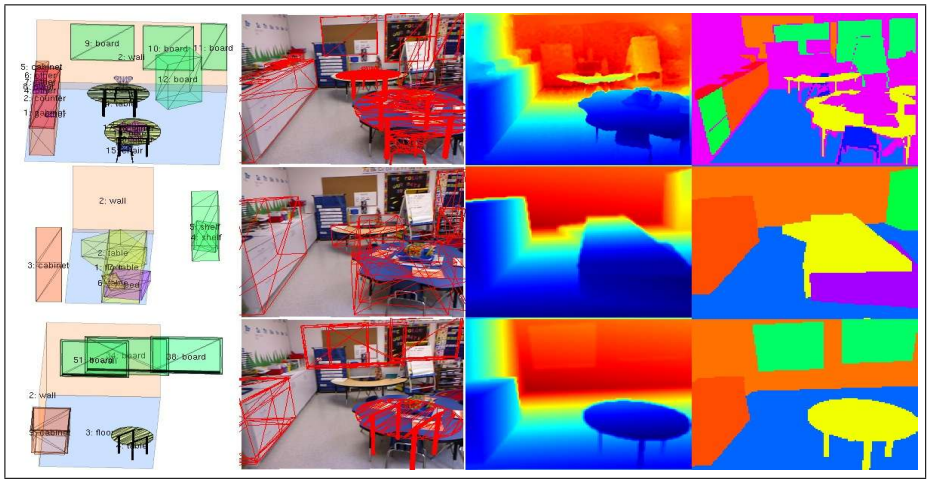

2: wall
3: wall
1: shelf
2: bed
1: floor
wall
bed

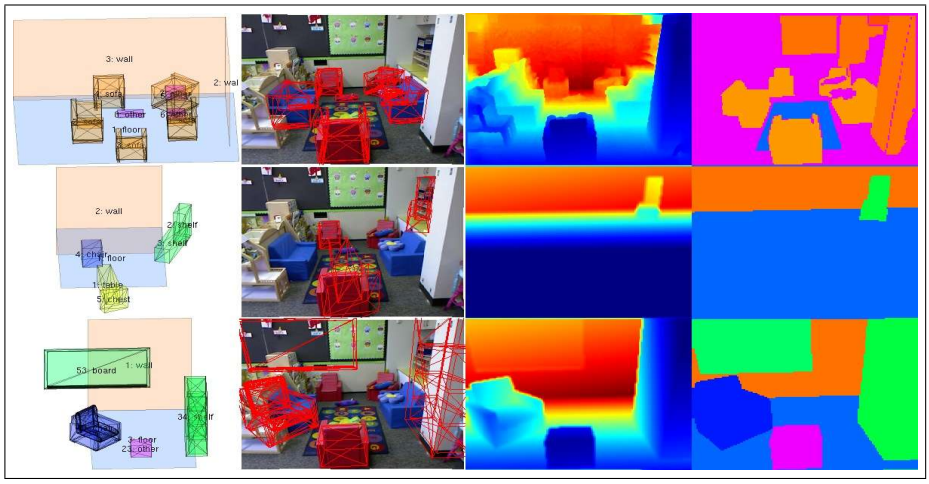


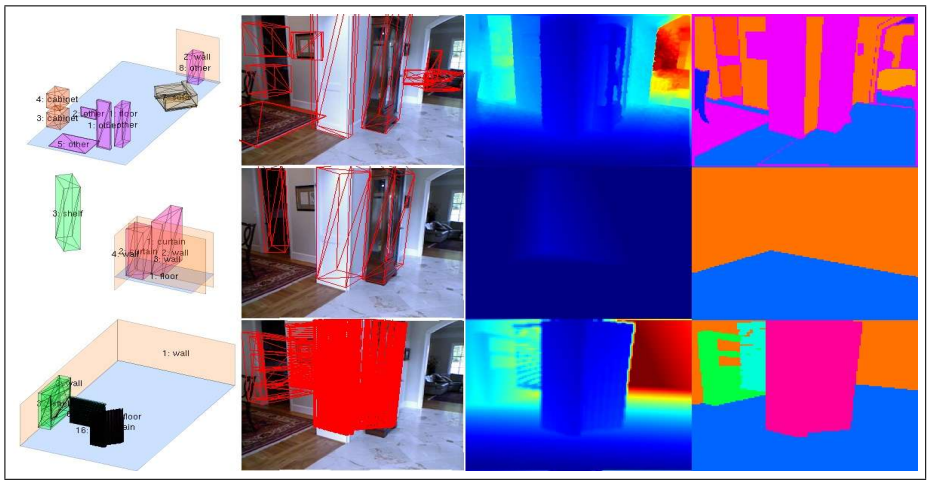


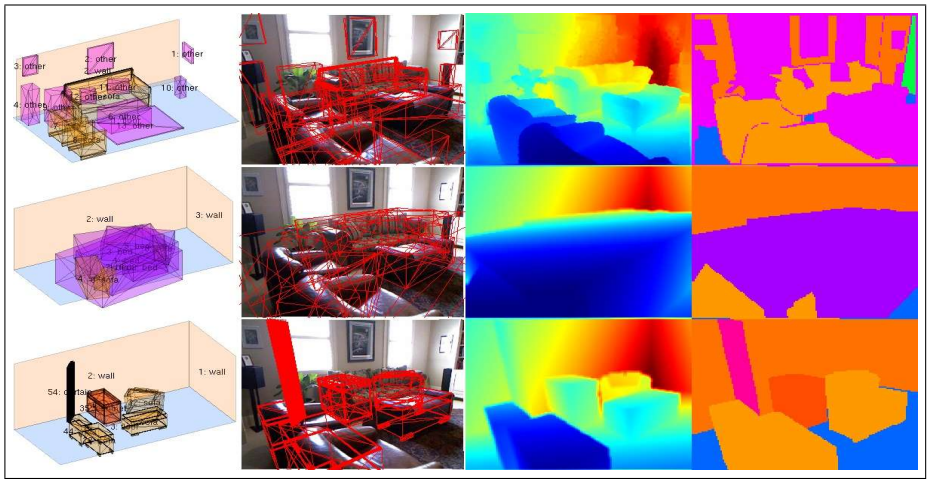


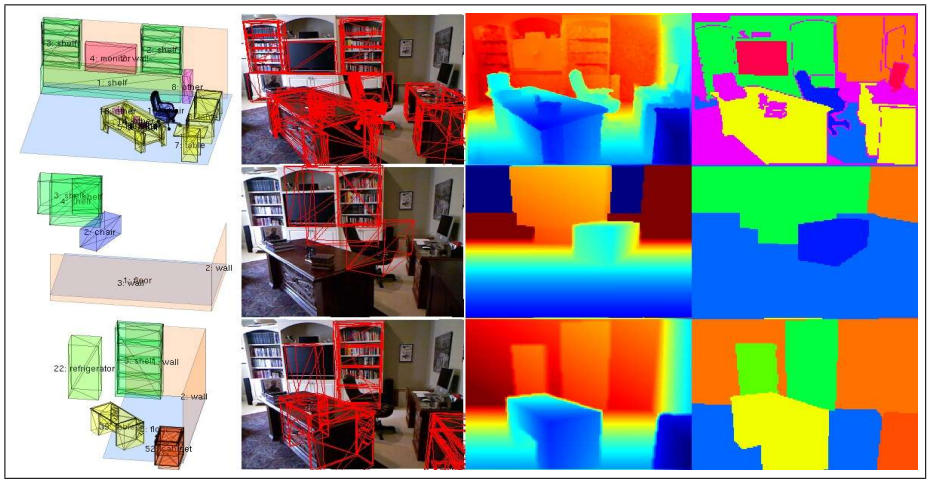


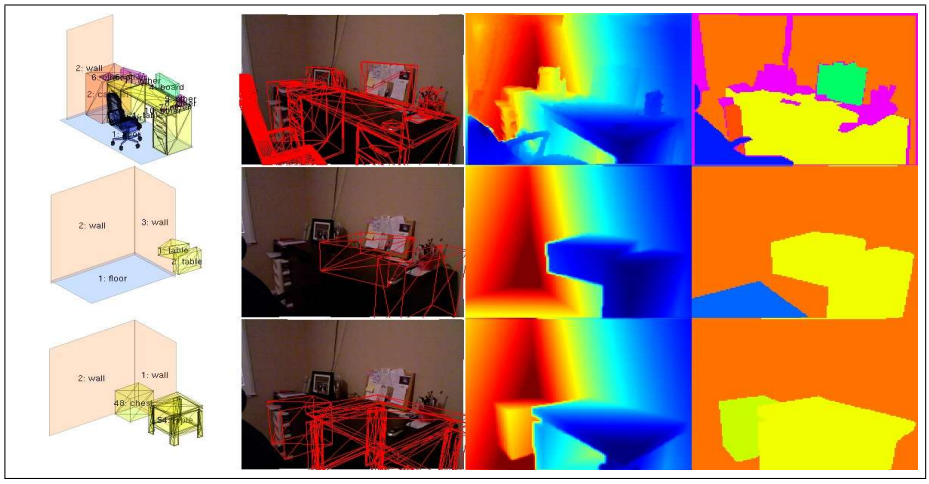


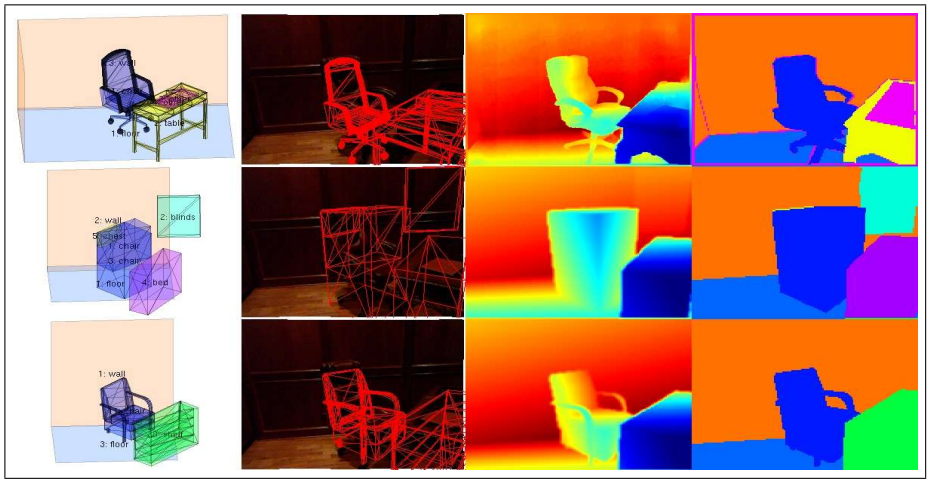

3: wall
2: wall
1: other
1: floor
2: wall
2: shelf
3: shelf
4: shelf
1: floor
1: table
5: chest
53: board
1: wall
34: shelf
3: floor
23: other

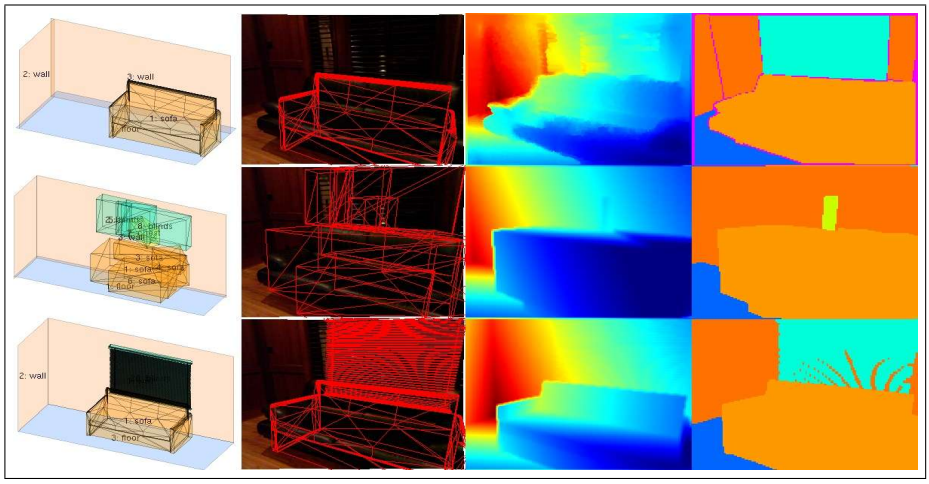

2: wall
8: other
4: cabinet
3: cabinet
1: floor
1: other
5: other
3: shelf
4: wall
1: curtain
2: wall
3: wall
1: floor
1: wall
1: floor

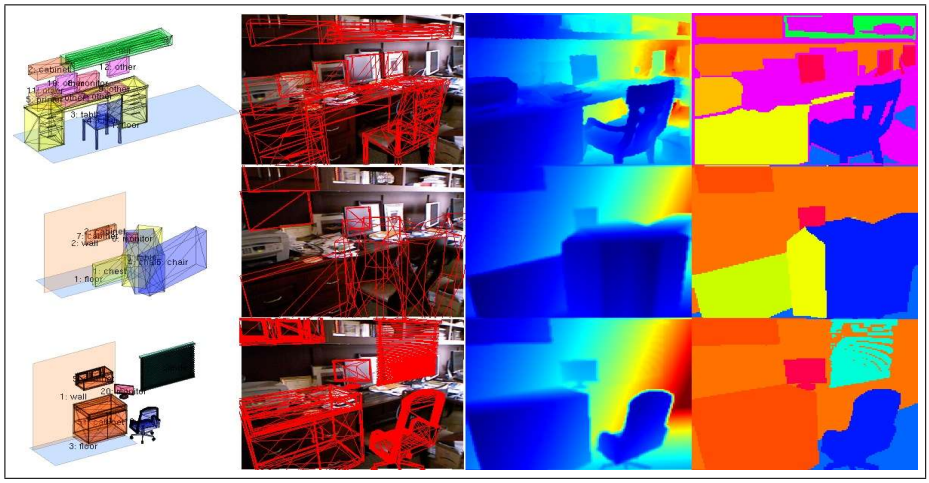

3: other
2: other
2: wall
1: other
10: other
4: other
5: other
2: wall
3: wall
2: wall
1: wall

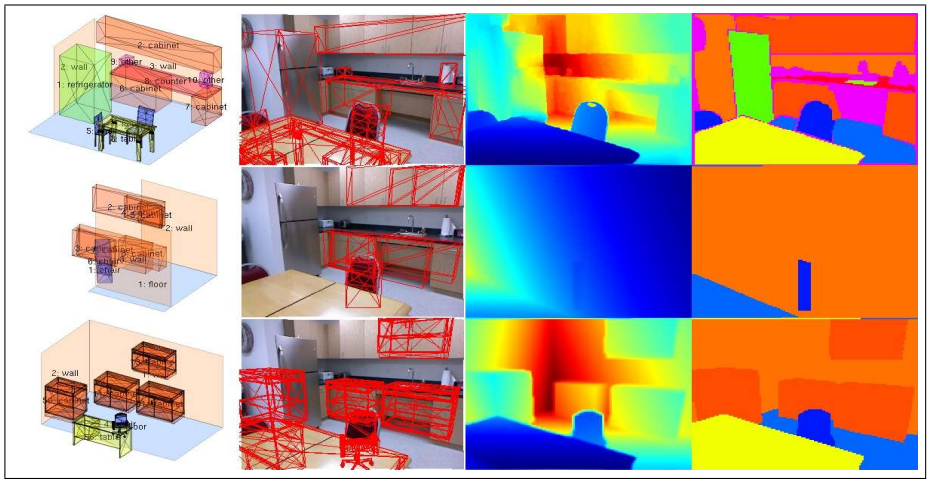

4: monitor
1: shelf
8: other
7: other
2: chair
2: wall
3: wall
22: refrigerator
5: shelf
2: wall

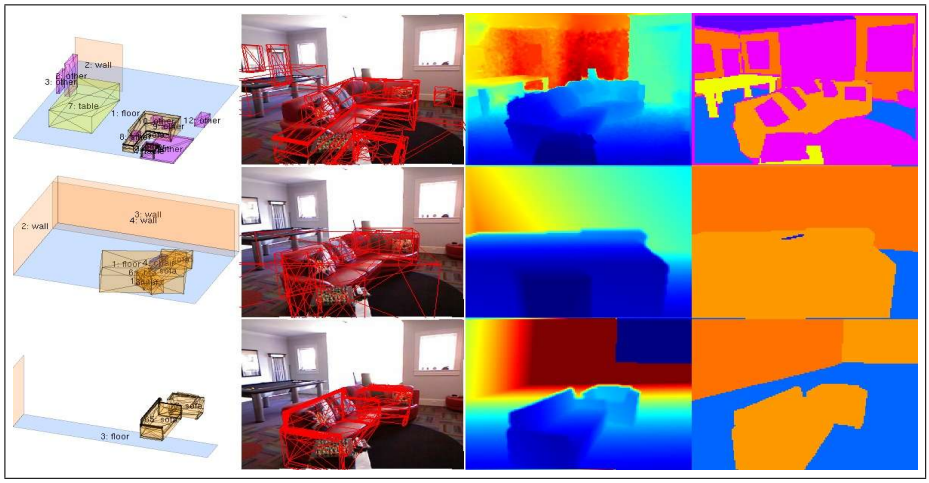

2: wall
1: floor
2: wall
3: wall
1: table
2: table
2: wall
1: wall

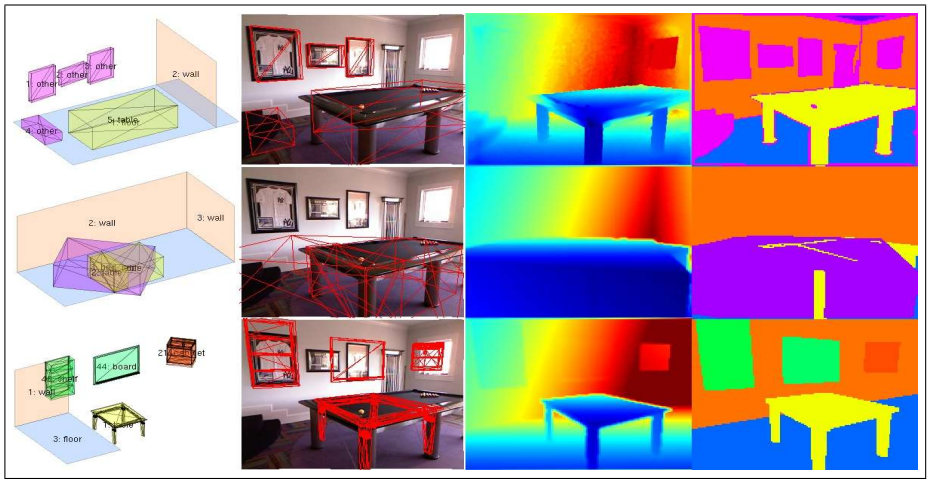

1: floor
2: wall
2: blinds
1: chair
3: chair
1: floor
4: bed
1: wall
3: floor

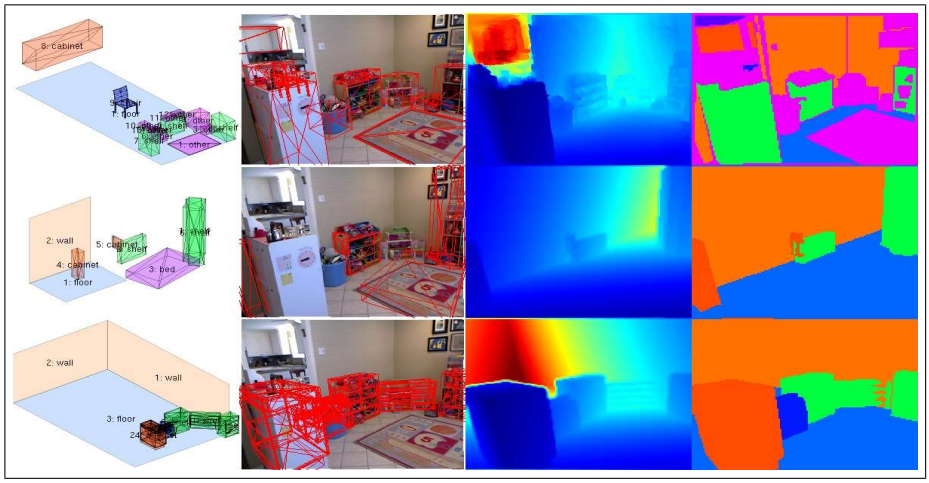

2: wall
3: wall
1: sofa
floor
2: wall
1: sofa
3: floor

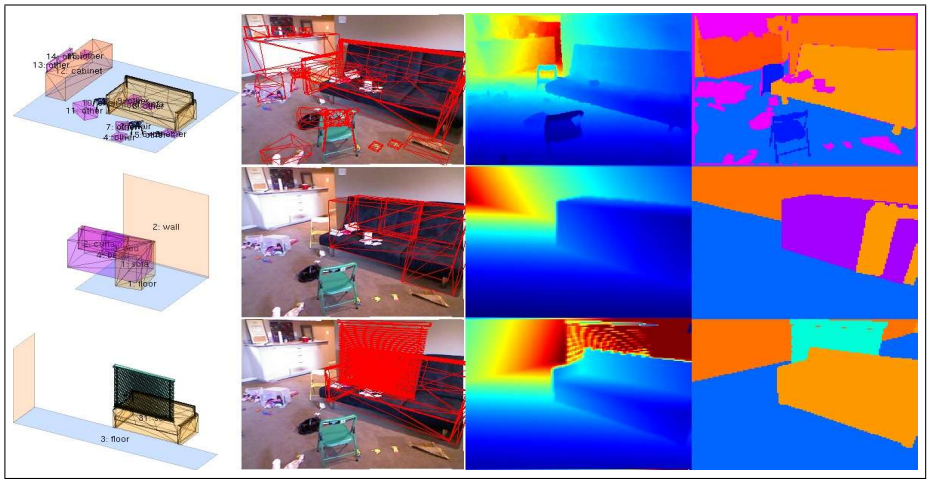

2: cabinet
1: other
3: table
floor
2: wall
1: floor
1: wall
3: floor

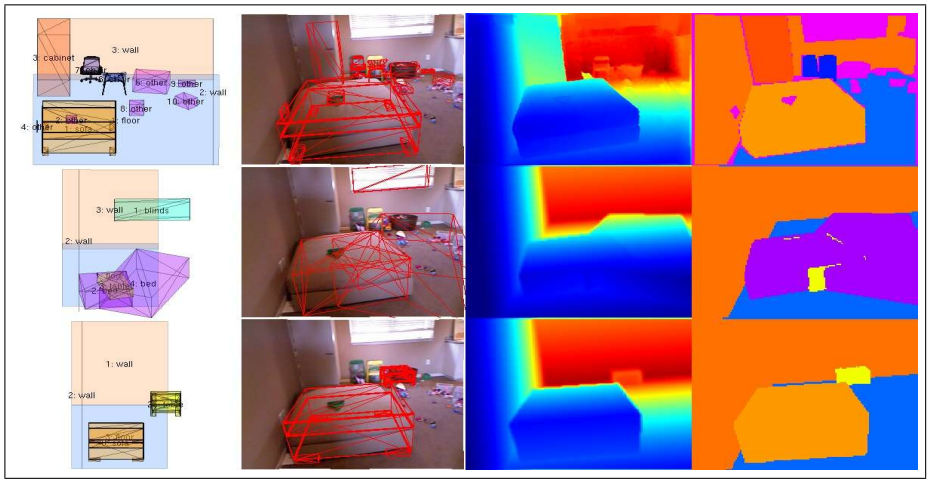


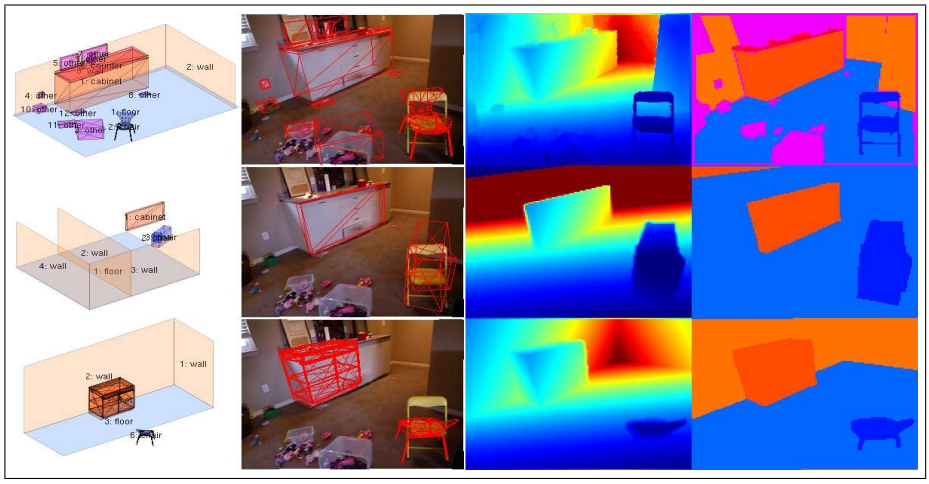


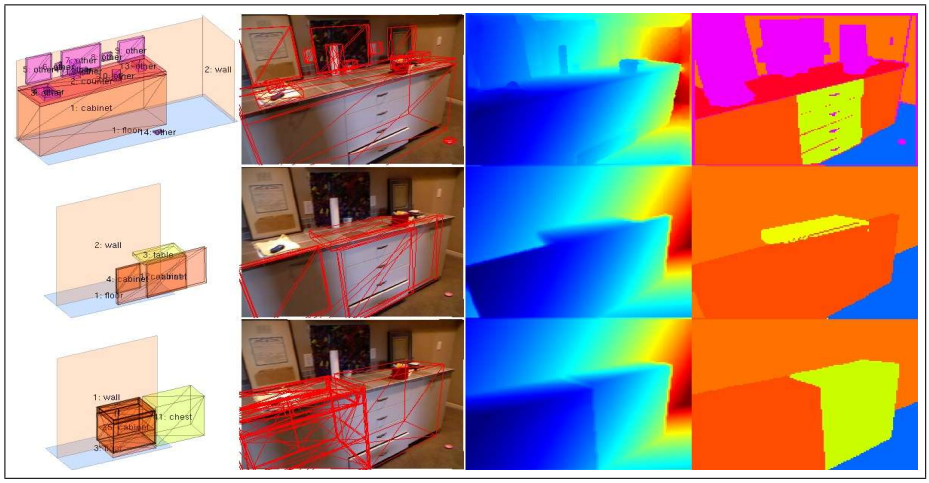

2: wall
1: other
2: other
3: other
4: other
5: table
2: wall
3: wall
5: table
44: board
1: wall
3: floor

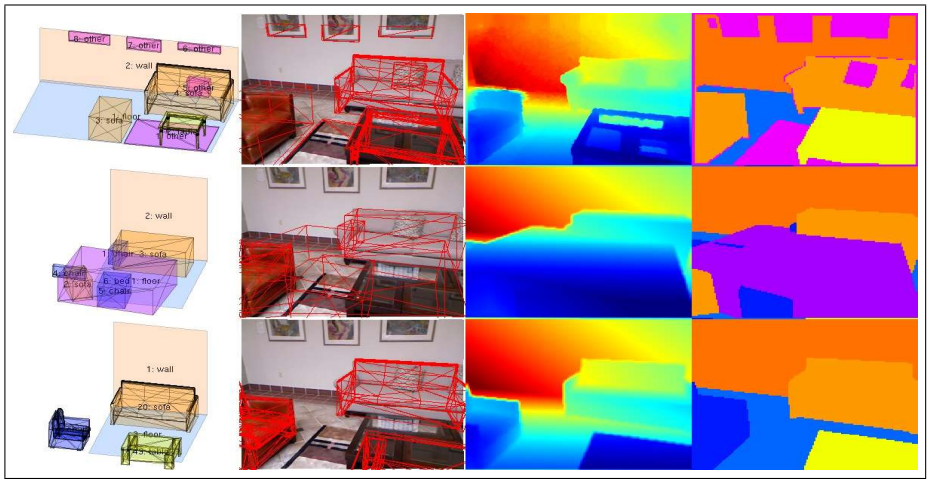

8: cabinet
1: floor
1: other
2: wall
5: cabinet
4: cabinet
1: floor
3: bed
2: wall
1: wall
3: floor

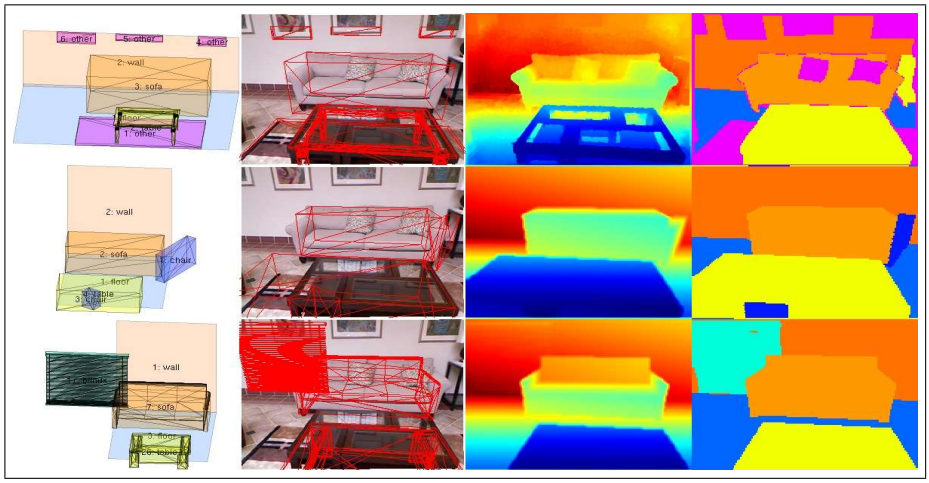

2: wall
1: floor
3: floor

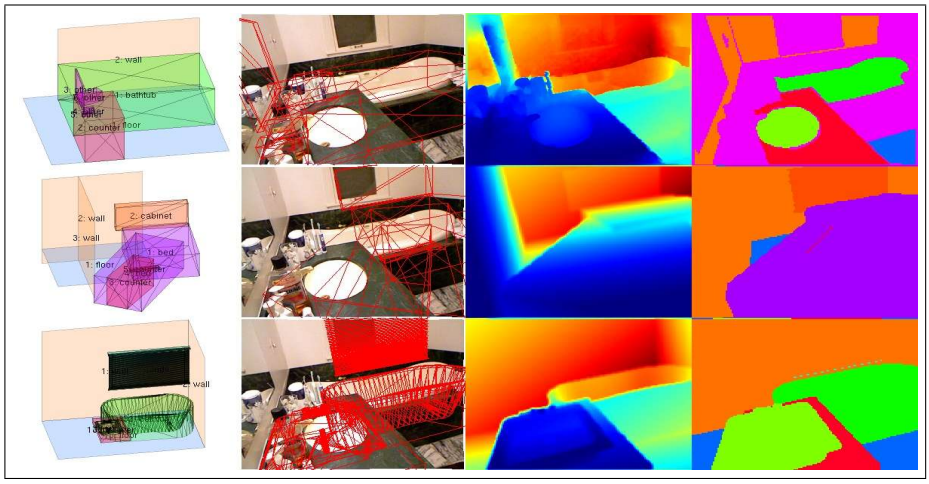

3: wall
2: wall
floor
3: wall
1: blinds
2: wall
1: wall
2: wall

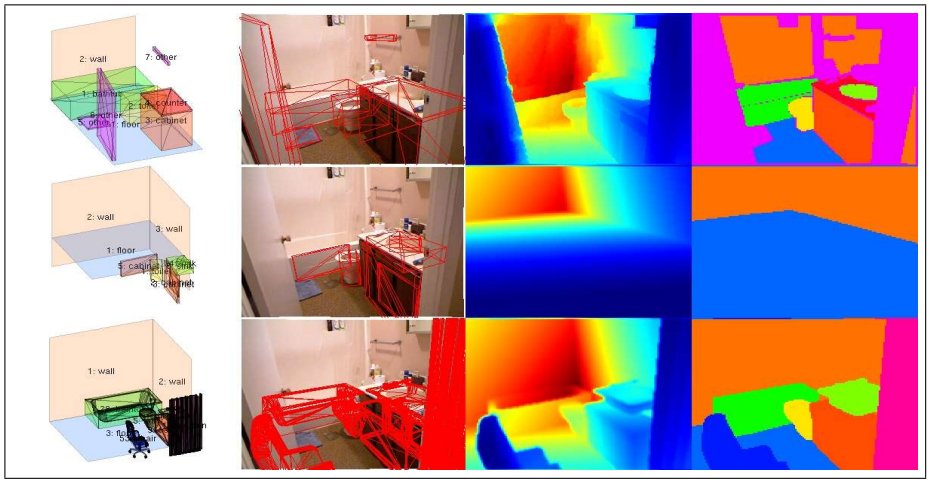

2: wall
1: cabinet
1: floor
1: cabinet
2: wall
4: wall
1: floor
3: wall
2: wall
1: wall
3: floor

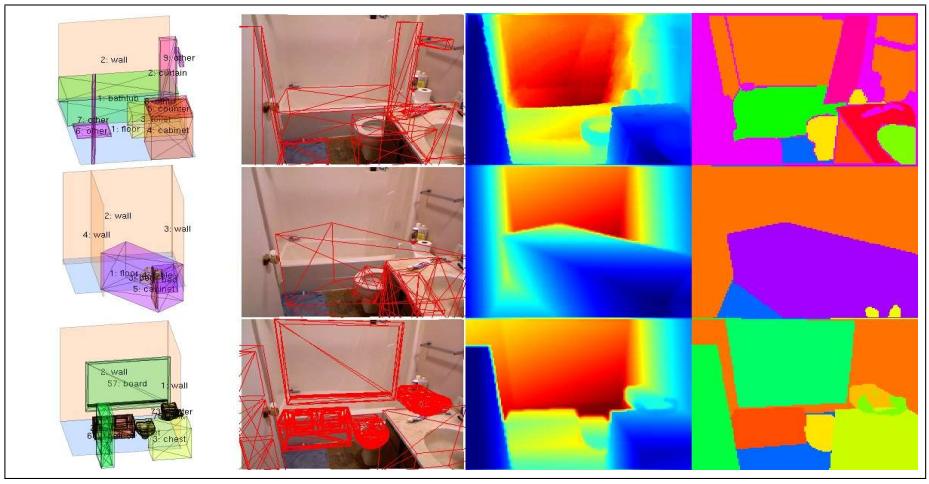

2: wall
1: cabinet
1: floor
2: wall
3: table
1: floor
1: wall
1: chest

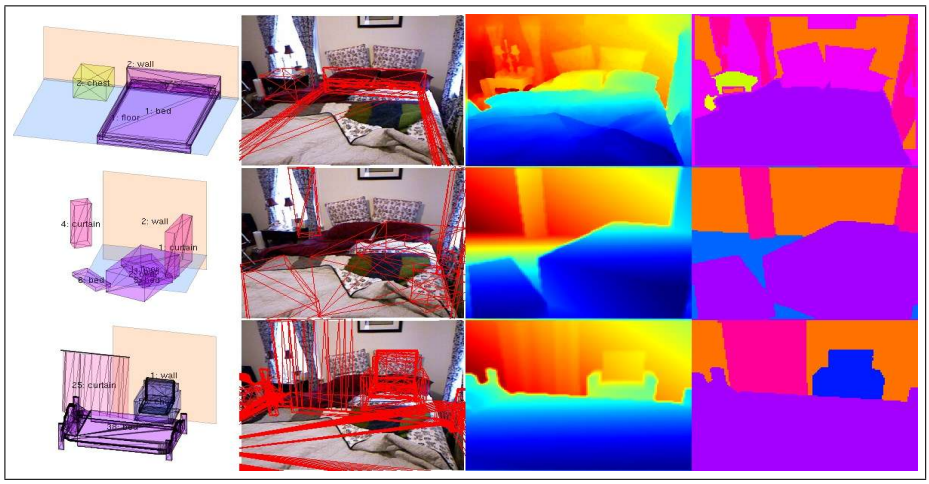


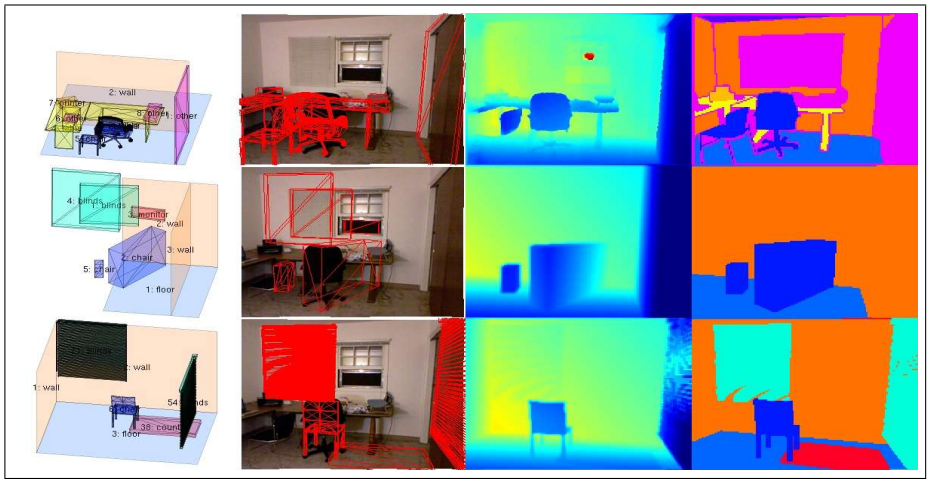


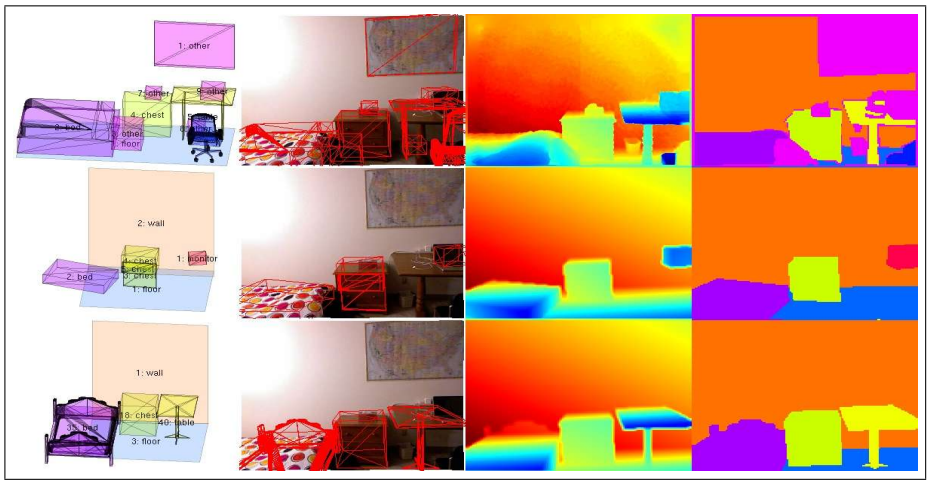

2: wall
1: bathtub
2: counter floor
2: wall
2: cabinet
3: wall
1: floor
1: bed
3: counter
2: wall

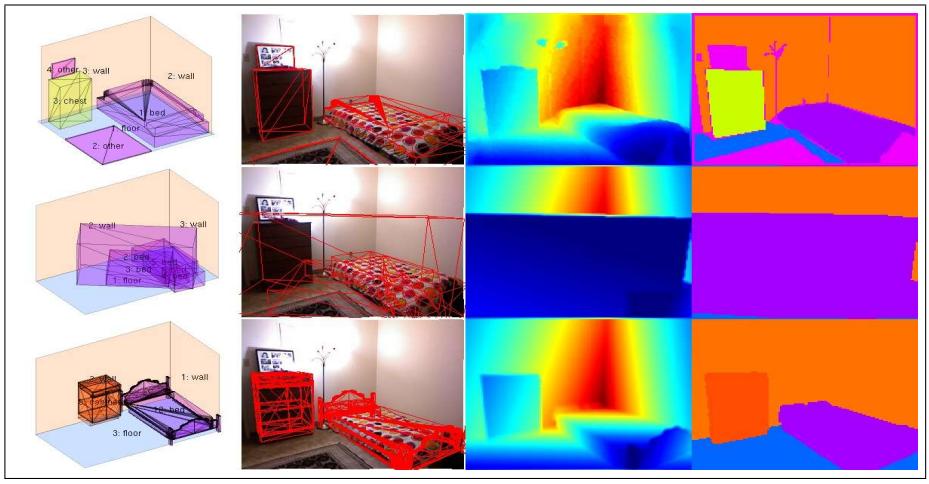

2: wall
7: other
1: bathtub
2: counter
3: cabinet
2: wall
3: wall
1: floor
1: wall
2: wall

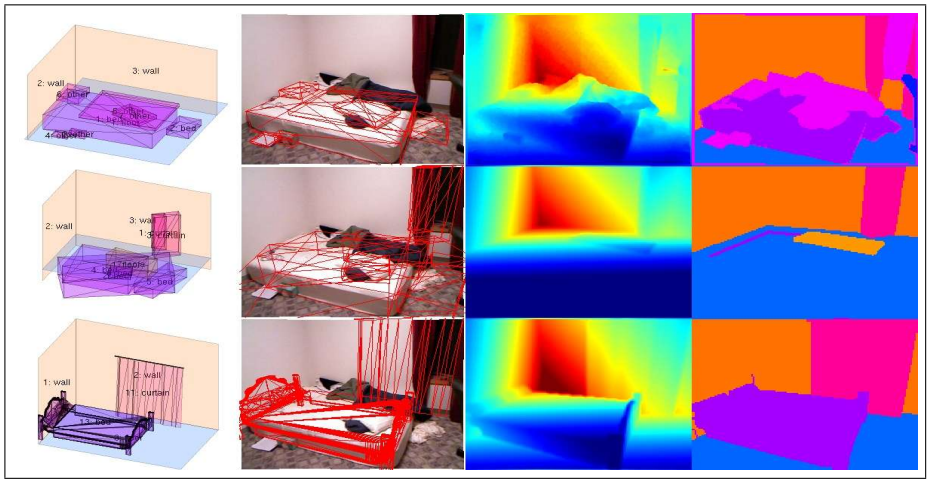

2: wall
9: other
2: curtain
1: bathtub
6: counter
7: other
3: toilet
6: other
1: floor
4: cabinet
2: wall
4: wall
3: wall
1: floor
5: cabinet
2: wall
57: board
1: wall
3: chest

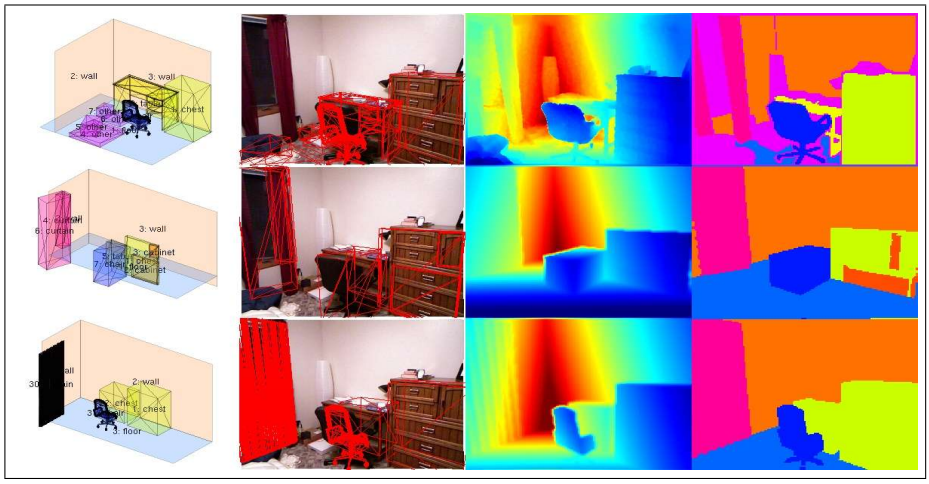

2: wall
2: chest
1: bed
floor
4: curtain
2: wall
1: curtain
floor
6: bed
25: curtain
1: wall

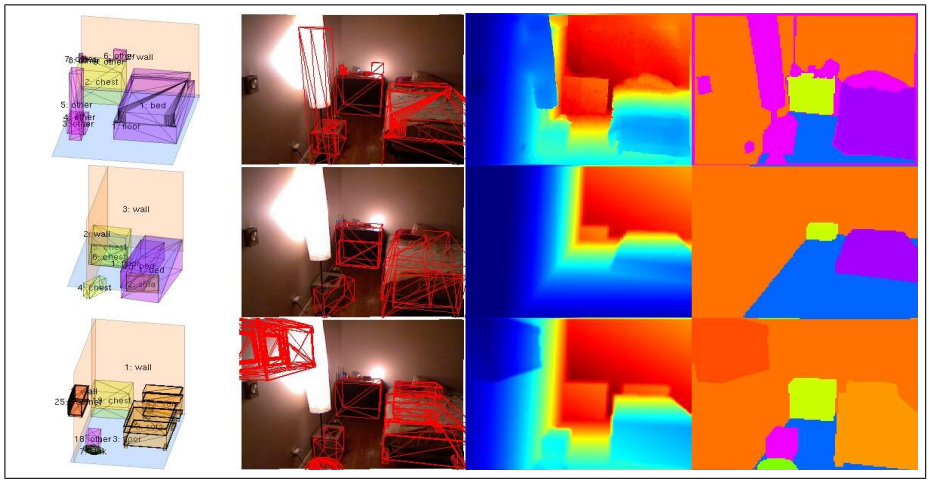

2: wall
7: other
8: other
1: other
4: blinds
3: monitor
2: wall
2: chair
3: wall
5: chair
1: floor
1: wall
2: wall
3: floor

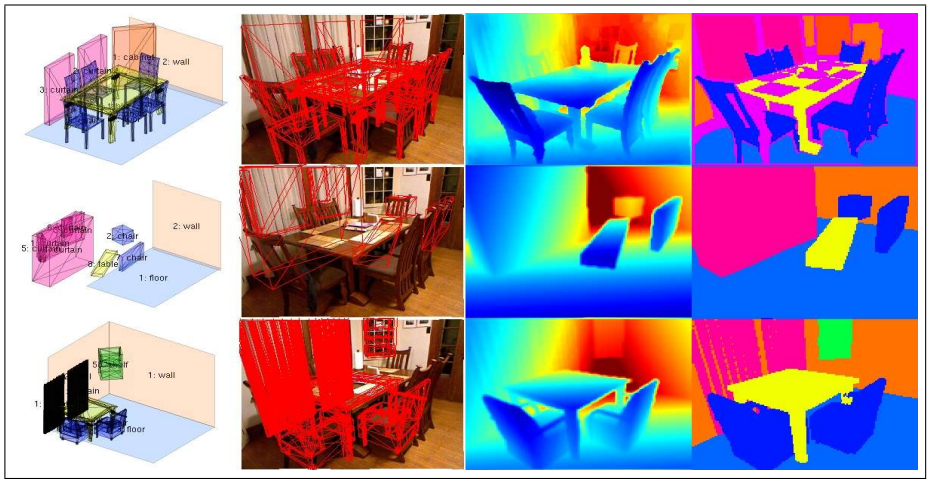

1: other
7: other
8: other
4: chest
3: other
1: floor
2: wall
1: monitor
2: bed
1: floor
1: wall
3: bed
3: floor

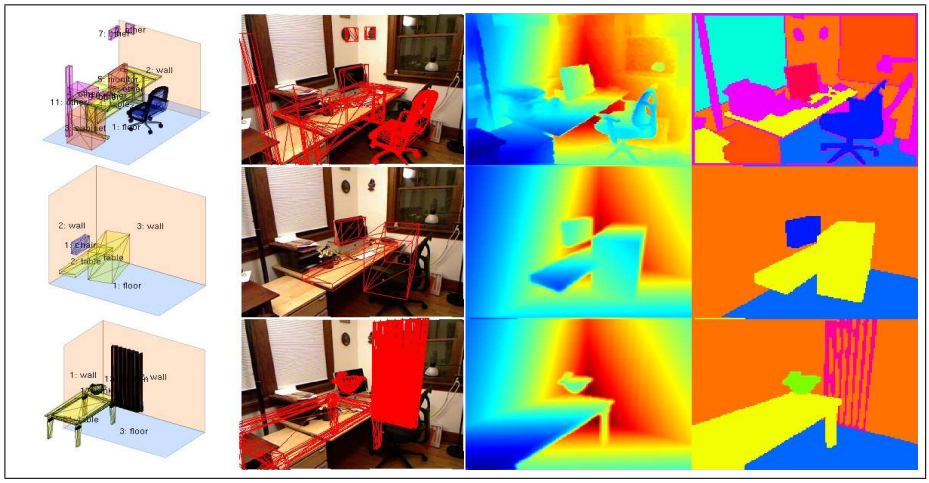

4: other
3: wall
2: wall
3: chest
1: bed
floor
2: other
2: wall
3: wall
2: bed
3: bed
1: floor
1: wall
3: floor

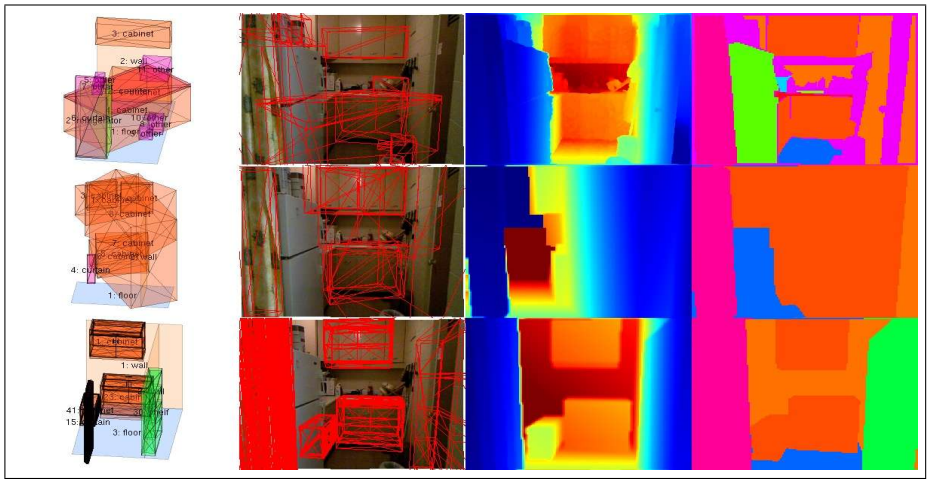

3: wall
2: wall
2: bed
2: wall
3: wall
1: wall
2: wall
11: curtain
1: bed

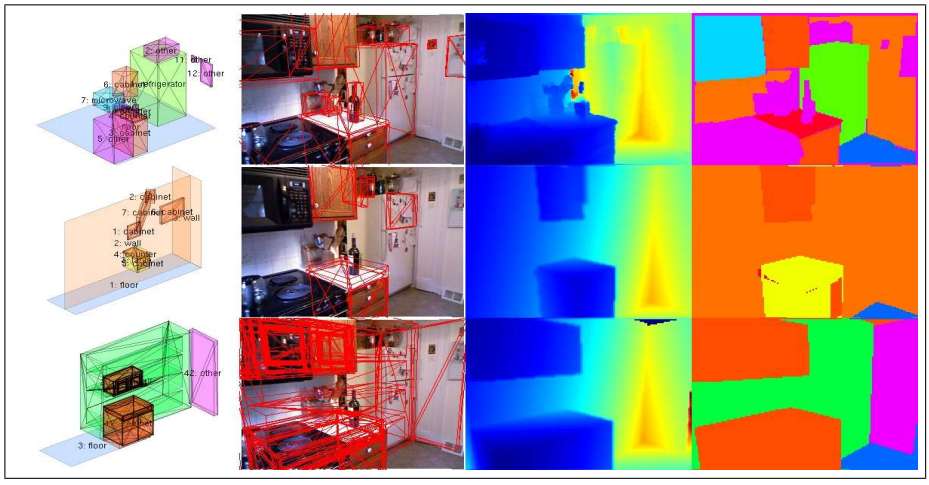

2: wall
3: wall
4: chest
3: wall
2: cabinet
2: wall
1: chest
3: floor

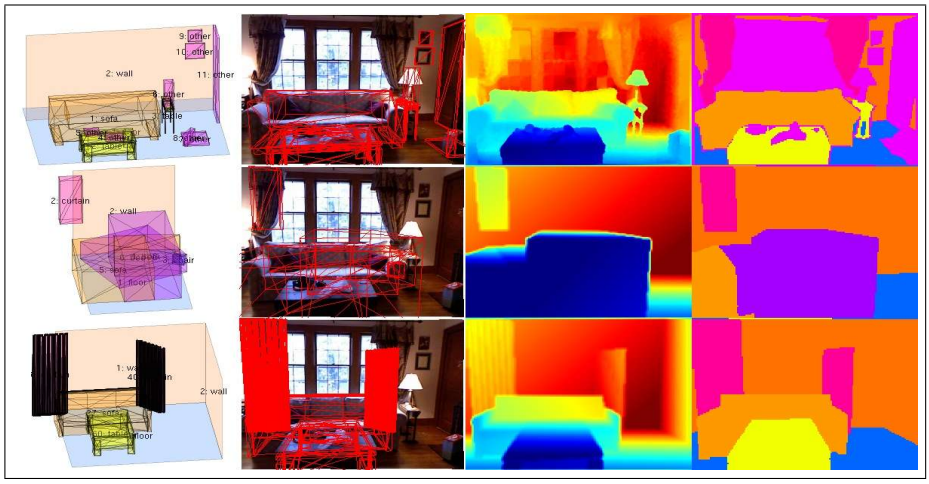

2: chest
1: bed
1: floor
3: wall
2: wall
1: wall
1: floor

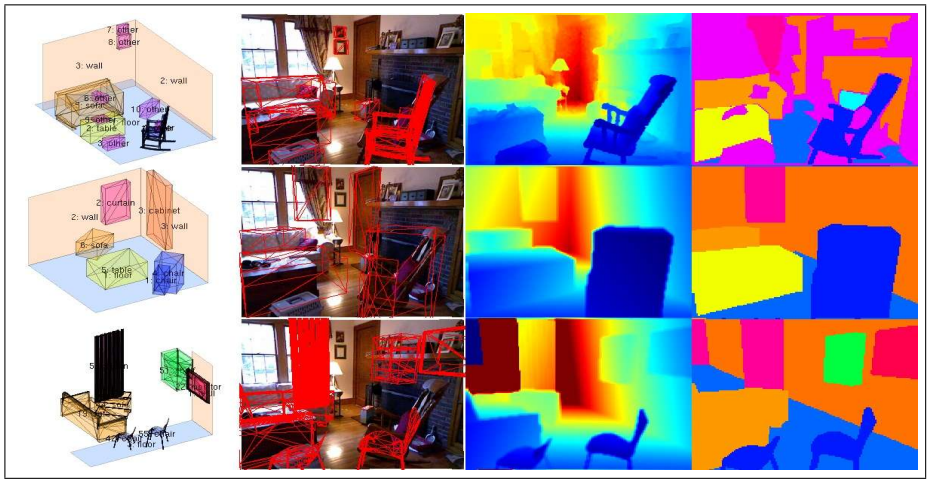

1: cabinet
2: wall
3: curtain
2: chair
2: wall
5: curtain
7: chair
8: table
1: floor
1: wall
floor

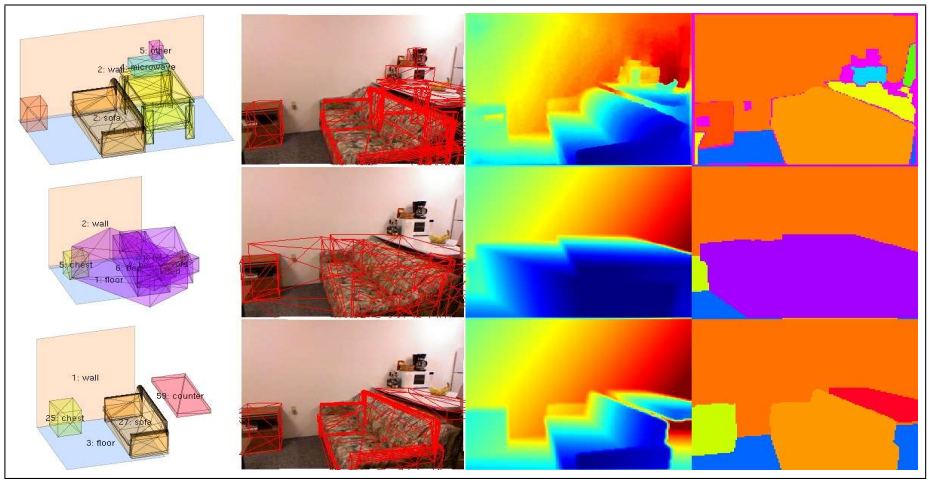

2: wall
2: wall
3: wall
1: floor
1: wall
3: floor

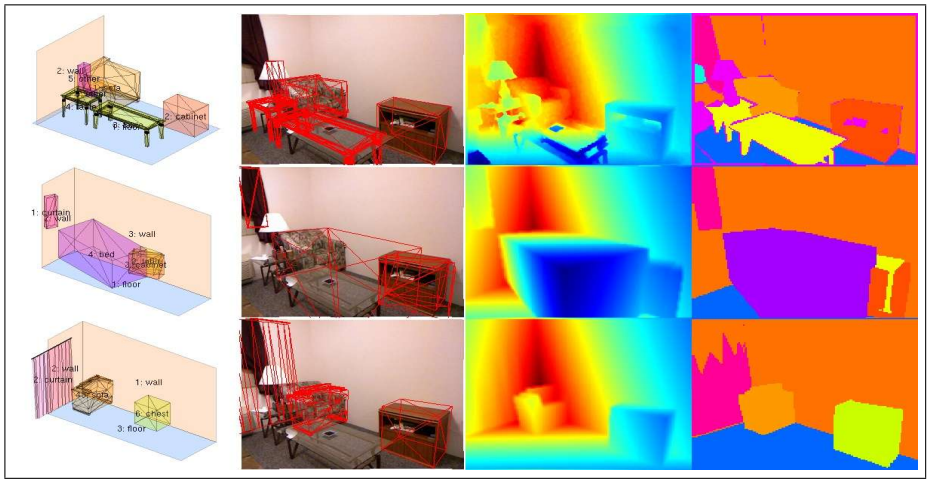


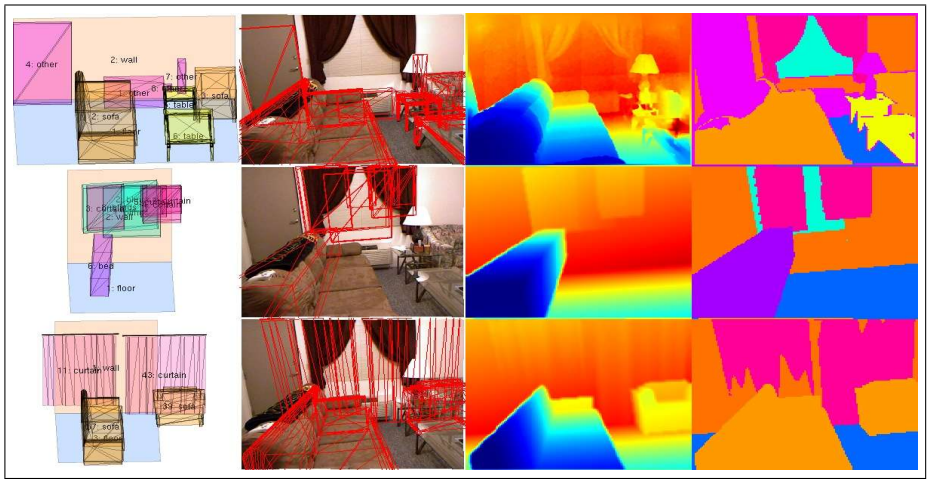


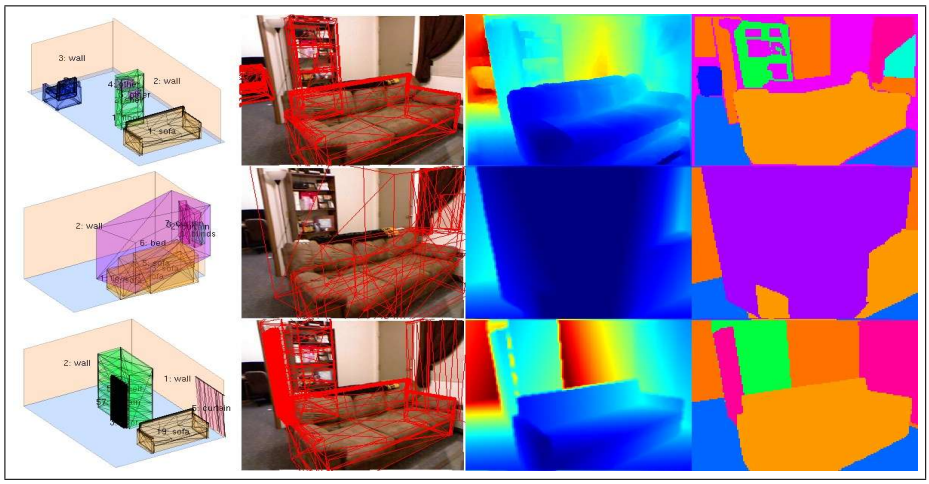


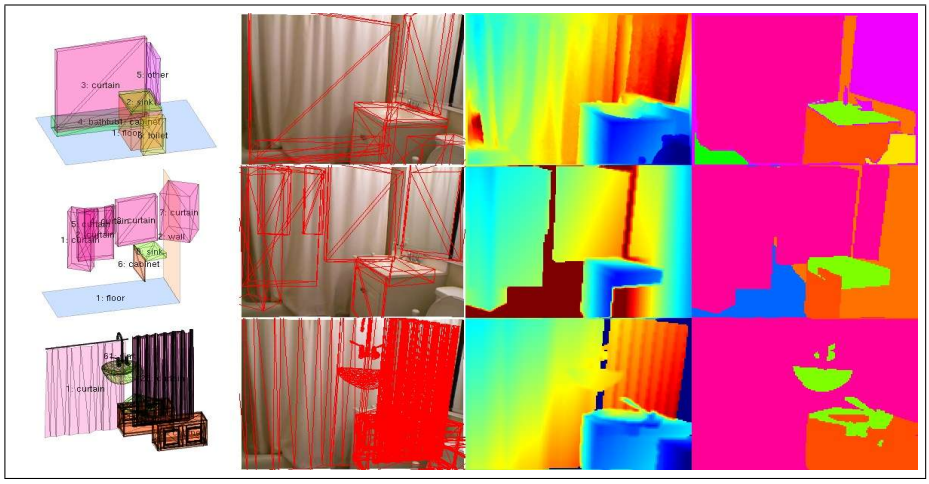


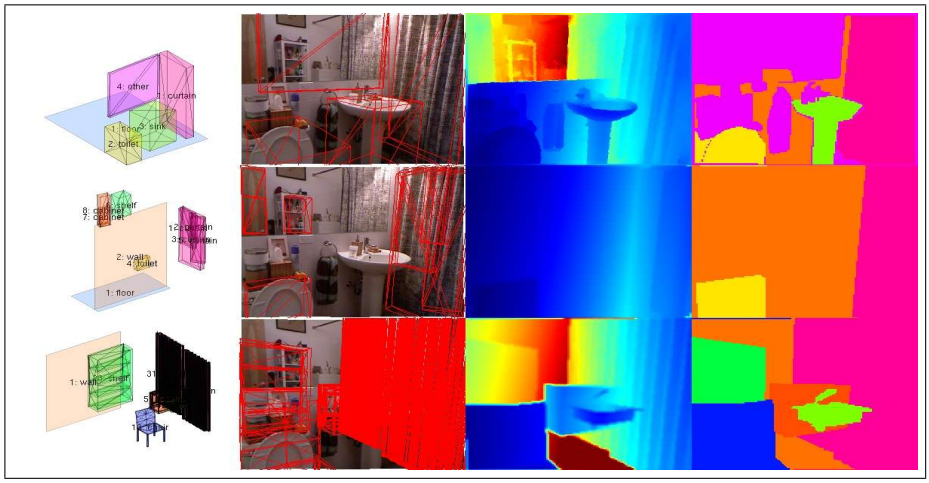

5: other
2: wall
2: sofa
2: wall
5: chest
1: floor
1: wall
59: counter
25: chest
27: sofa
3: floor

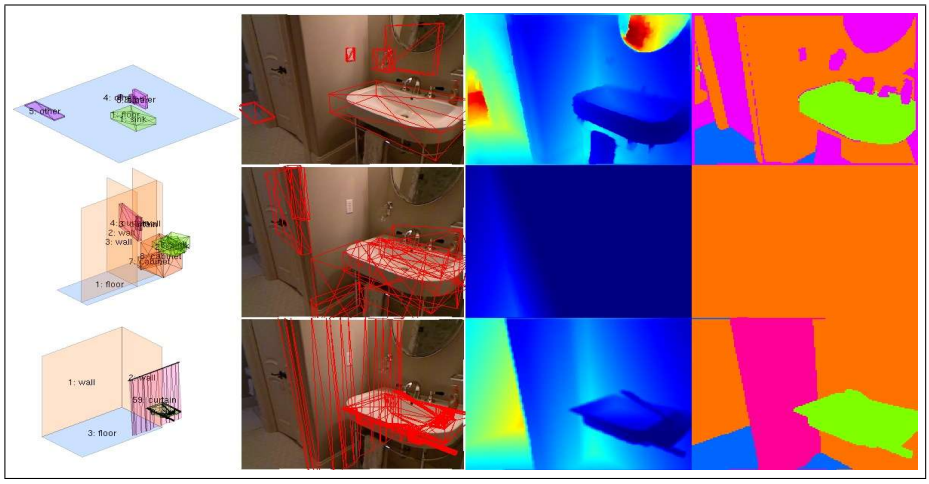

2: wall
2: cabinet
1: curtain
2: wall
3: wall
4: bed
3: floor
2: wall
2: curtain
1: wall
6: chest
3: floor

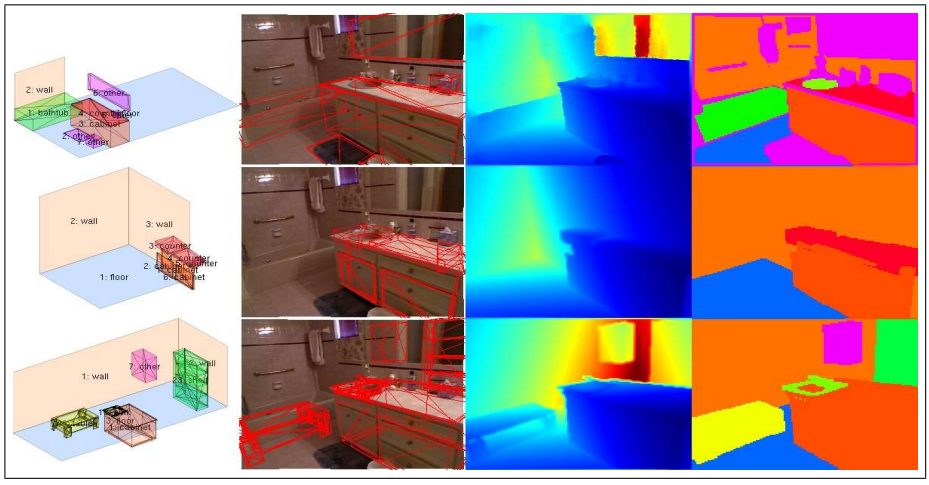

4: other
2: wall
7: other
6: other
4: sofa
3: table
2: sofa
1: floor
5: table
3: curtain
2: wall
6: bed
1: floor
11: curtain
2: wall
43: curtain
33: table
17: sofa
1: floor

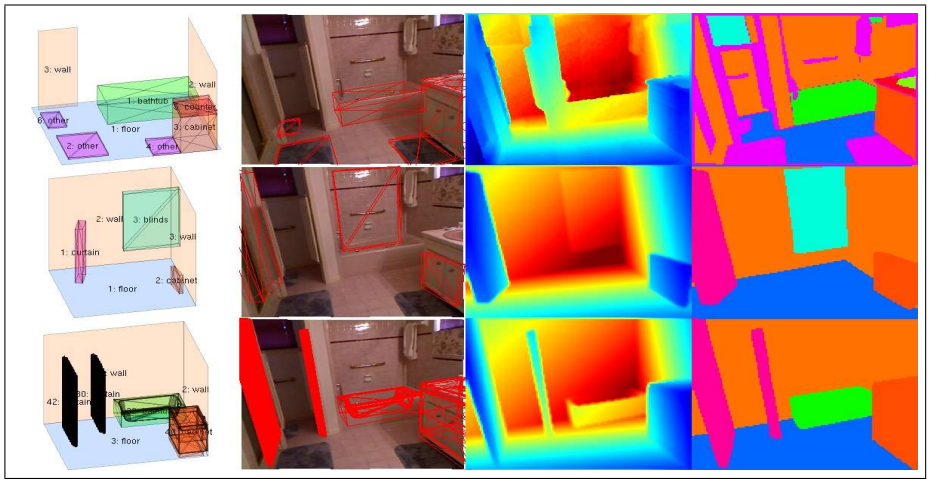

3: wall
4: other
2: wall
1: floor
1: sofa
2: wall
6: bed
2: wall
1: wall
6: curtain
19: sofa

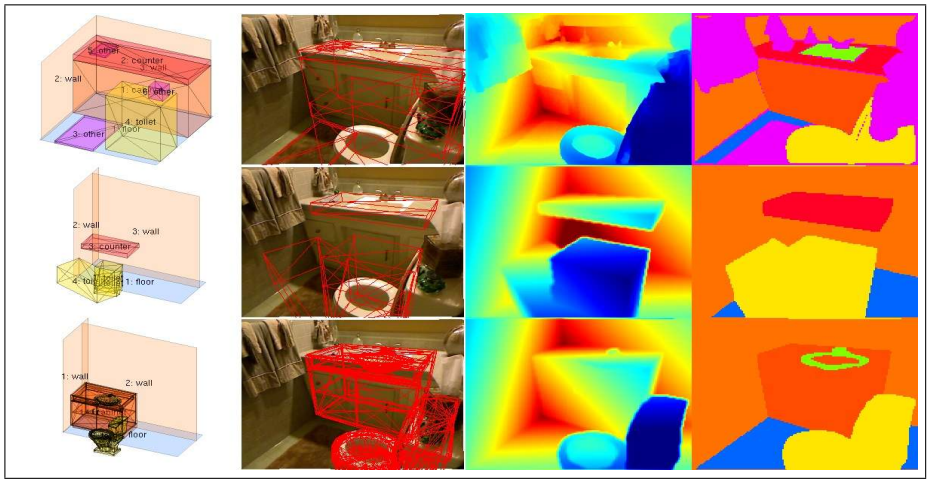

3: curtain
5: other
2: sink
4: bathtub
3: cabinet
1: floor
6: toilet
7: curtain
2: wall
8: sink
6: cabinet
1: floor
1: curtain

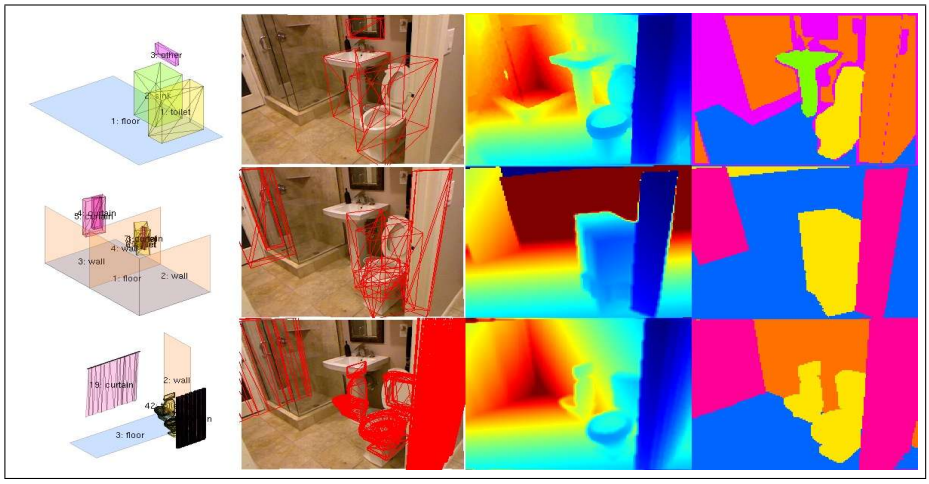

4: other
1: curtain
1: floor
3: sink
2: toilet
2: wall
4: toilet
1: floor
1: wall
3: shelf

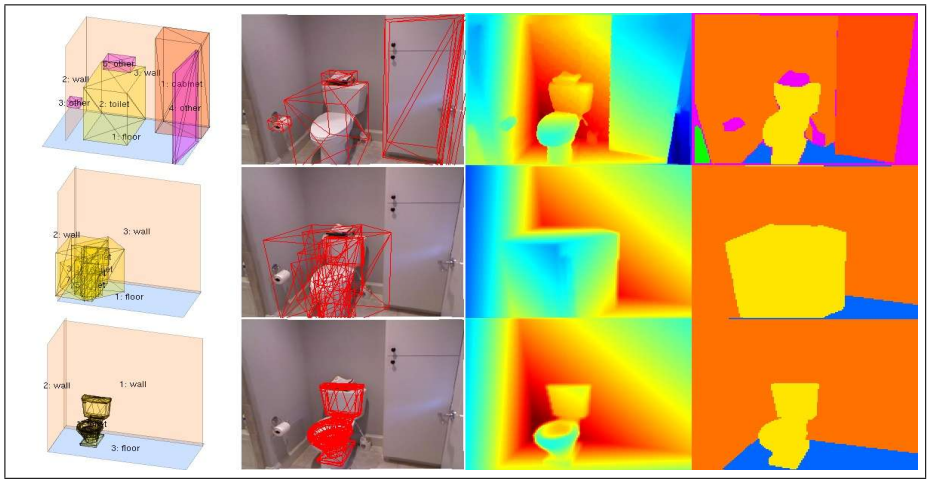


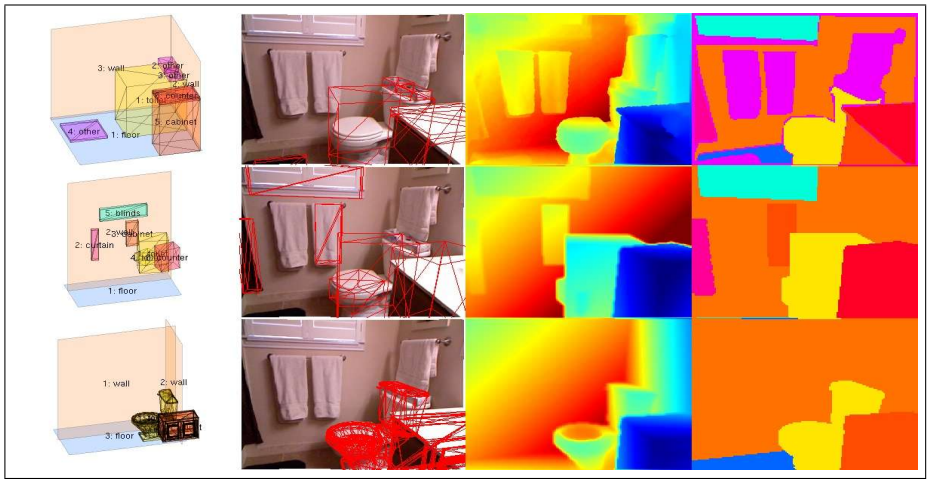


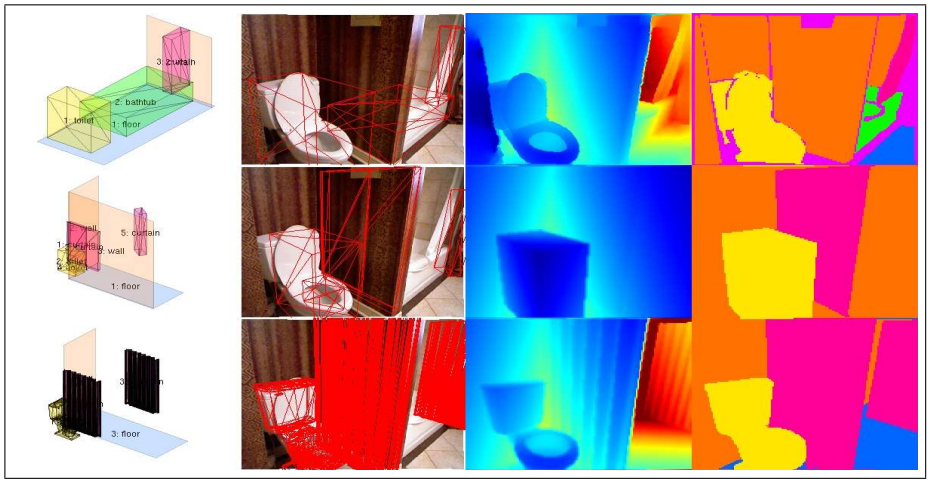

3: wall
2: wall
1: bathtub
5: counter
6: other
1: floor
3: cabinet
2: other
4: other
2: wall
3: blinds
3: wall
1: curtain
1: floor
2: cabinet
1: wall
2: wall
3: floor

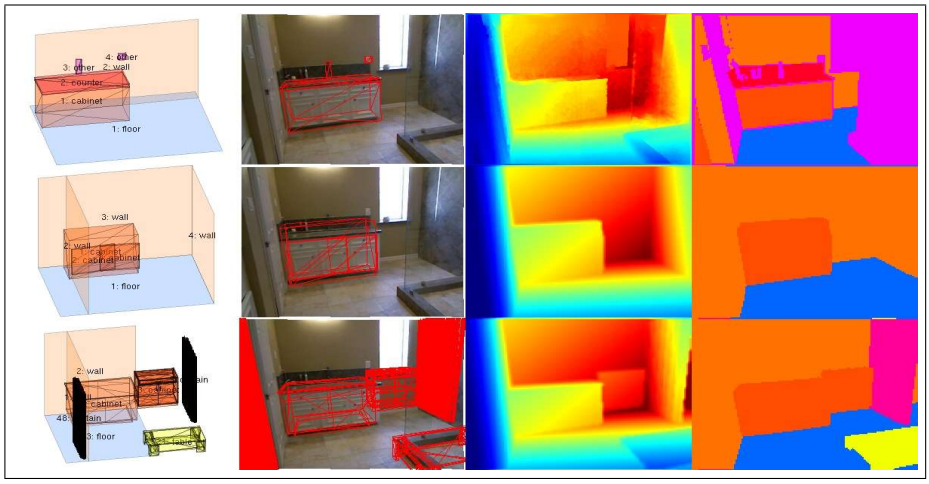

5: other
2: counter
3: wall
2: wall
1: cabinet
4: toilet
1: floor
3: other
2: wall
3: wall
1: floor
1: wall
2: wall
floor

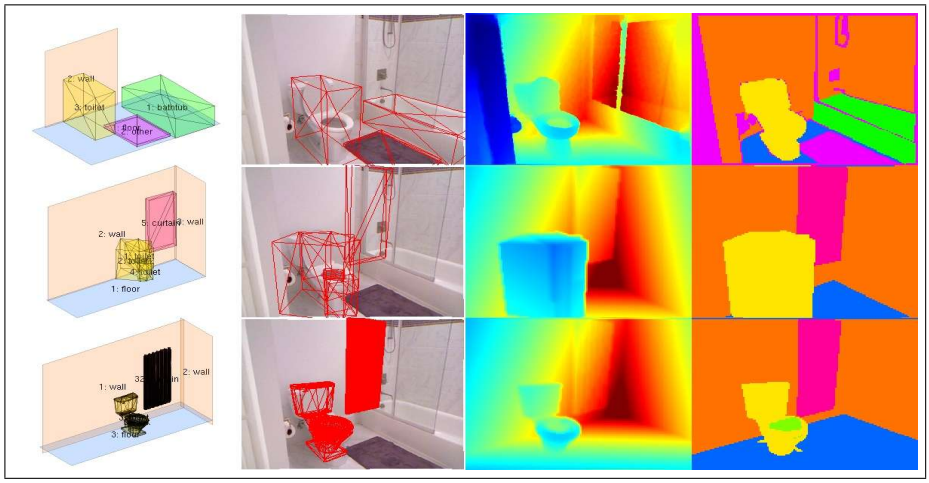

3: other
2: toilet
1: floor
1: toilet
4: wall
3: wall
1: floor
2: wall
15: curtain
2: wall
3: floor

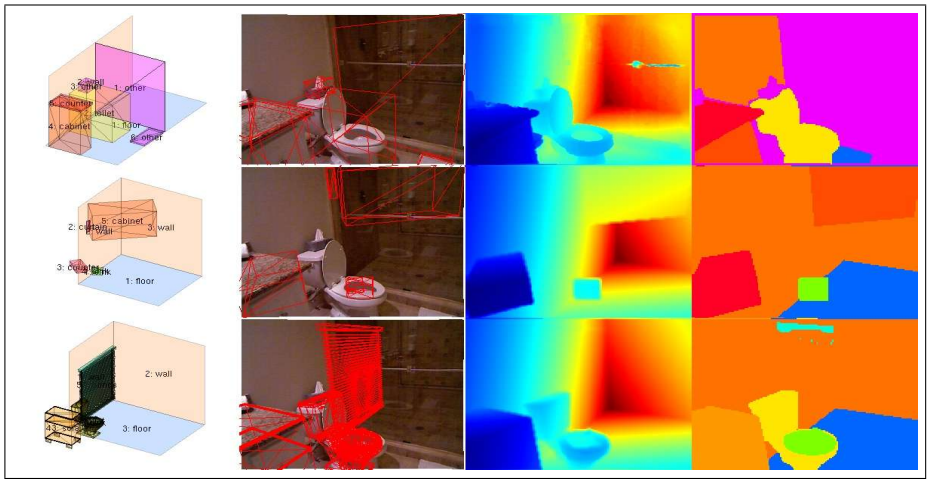

5: other
2: wall
3: wall
1: cabinet
3: other
2: toilet
4: other
1: floor
2: wall
3: wall
1: floor
2: wall
1: wall
3: floor

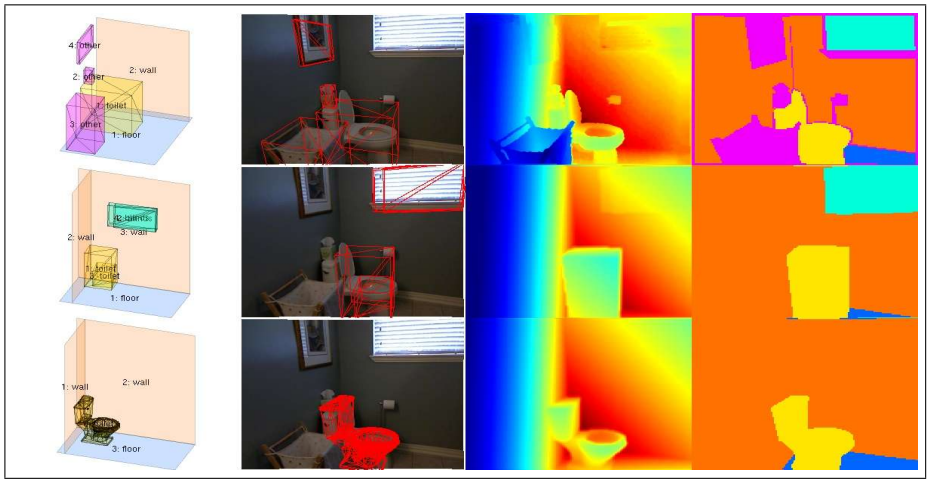

3: wall
4: other
1: floor
5: cabinet
5: blinds
2: curtain
1: floor
1: wall
2: wall
3: floor

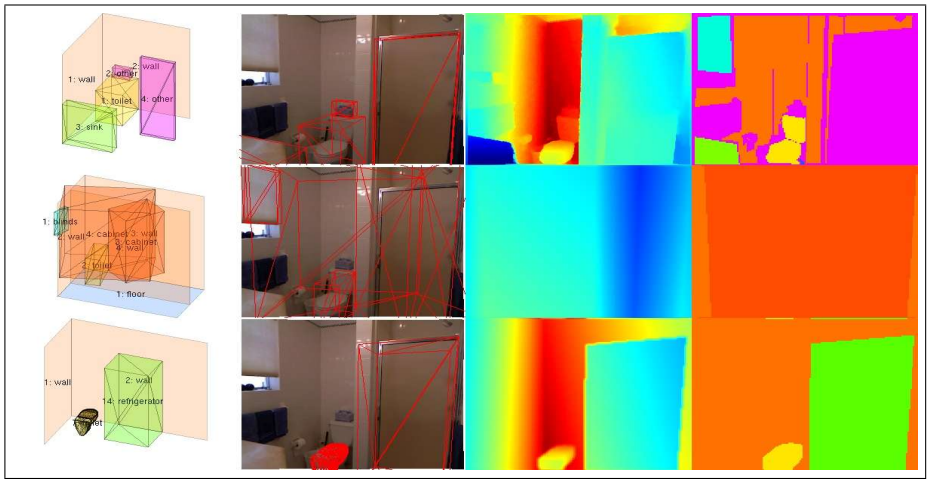

3: curtain
2: bathtub
1: toilet
1: floor
5: curtain
1: floor
3: floor

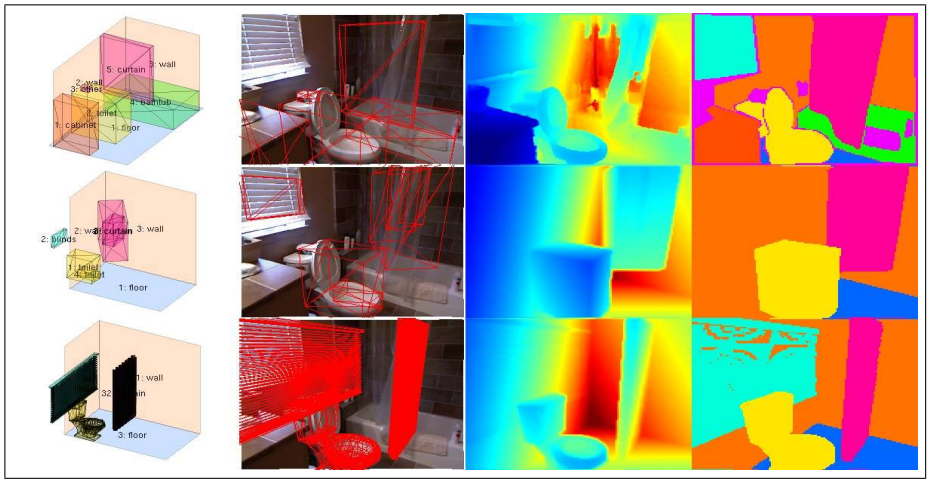

4: other
3: other
2: wall
2: counter
1: cabinet
1: floor
3: wall
4: wall
2: wall
1: cabinet
1: floor
2: wall
3: floor

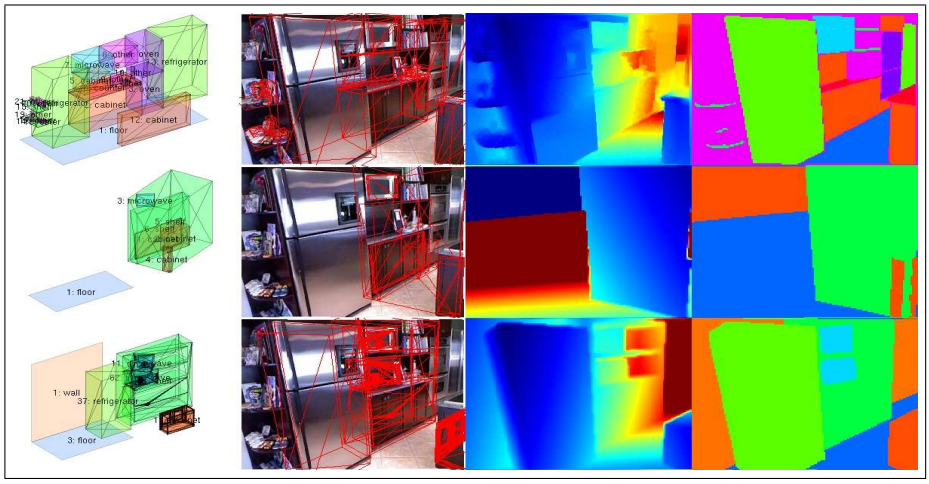

2: wall
3: floor
1: bathtub
floor
other
2: wall
5: curtain
3: wall
1: floor
1: wall
2: wall
3: floor

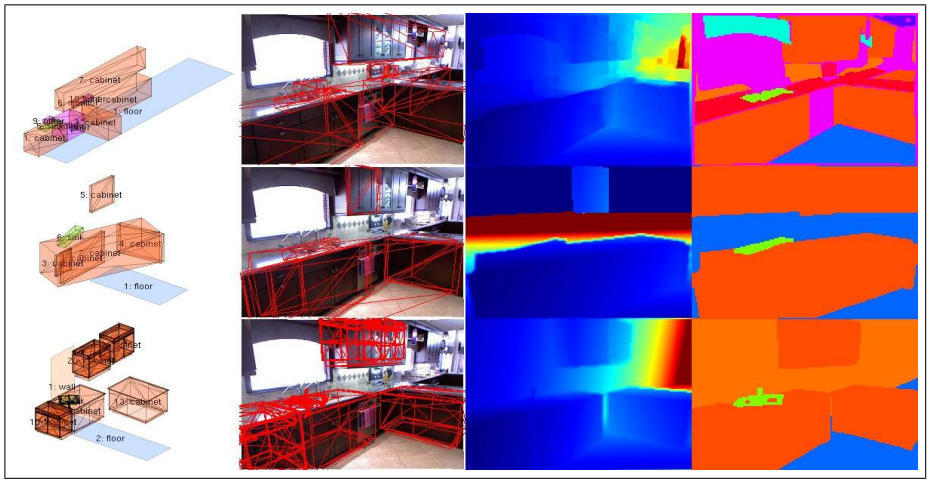

1: other
4: cabinet
1: floor
6: other
5: cabinet
3: wall
1: floor
2: wall
3: floor

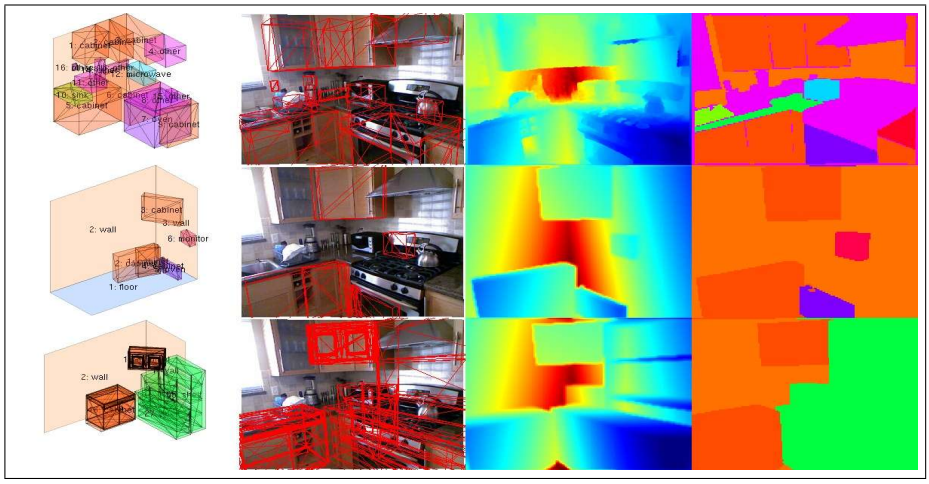

4: other
2: other
2: wall
1: toilet
3: other
1: floor
2: wall
3: wall
1: toilet
2: toilet
1: floor
1: wall
2: wall
3: floor

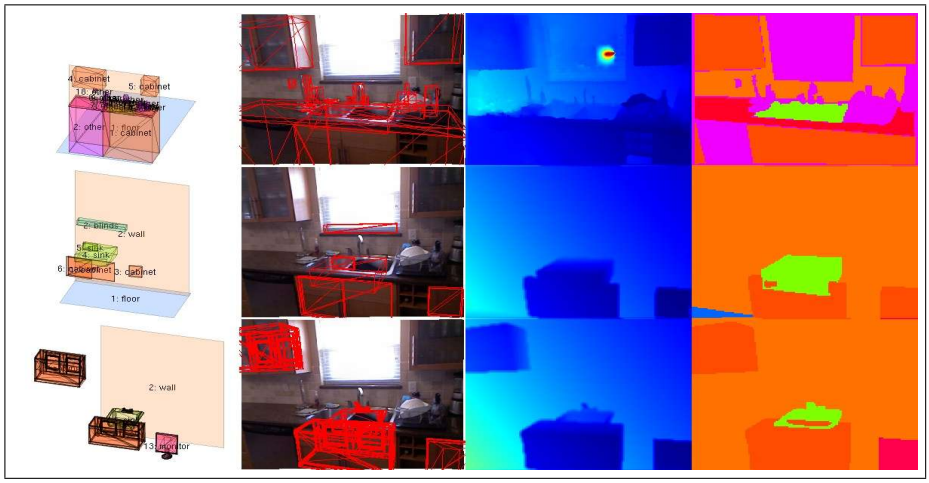

1: wall
2: other
3: wall
1: toilet
4: other
3: sink
1: blinds
2: wall
4: cabinet
3: wall
3: cabinet
3: wall
2: toilet
1: floor
1: wall
2: wall
4: refrigerator

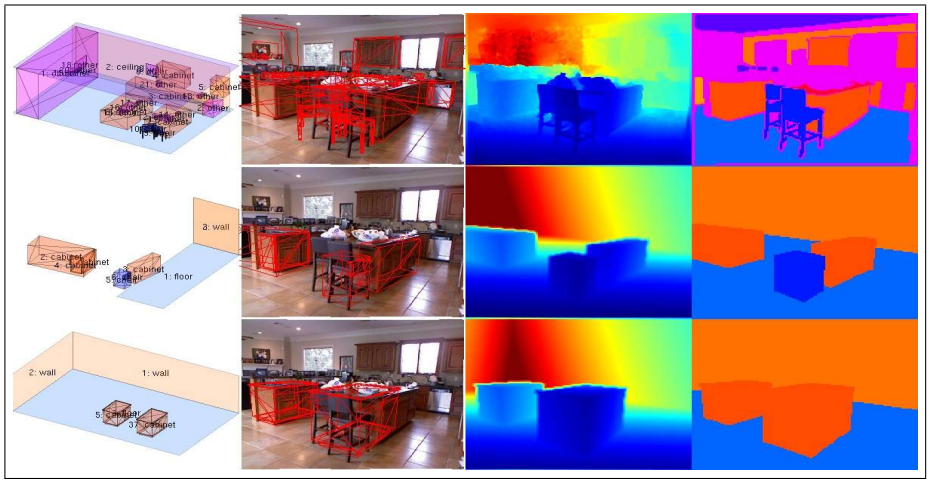

5: curtain
3: wall
2: wall
3: other
4: bathtub
3: toilet
1: cabinet
1: floor
2: blinds
2: wall
5: curtain
3: wall
1: toilet
1: floor
1: wall
3: floor

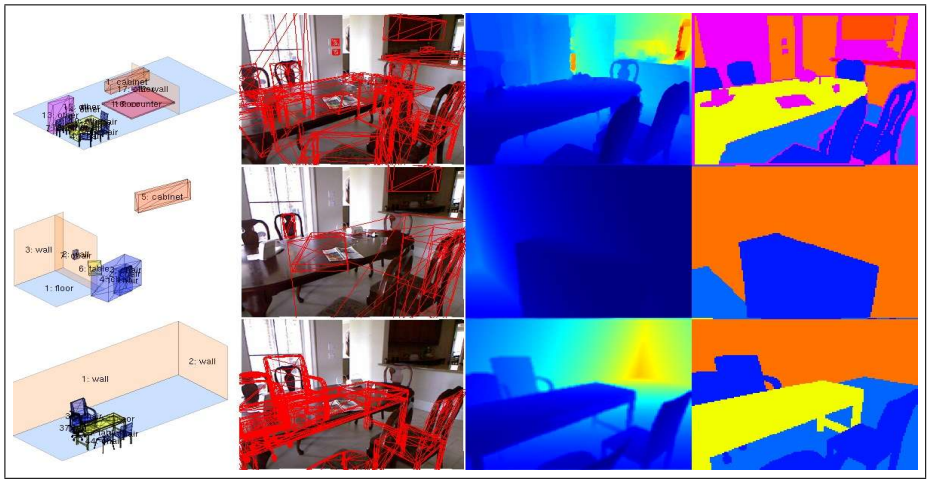

1: floor
3: microwave
4: cabinet
1: floor
1: wall
37: refrigerator
3: floor

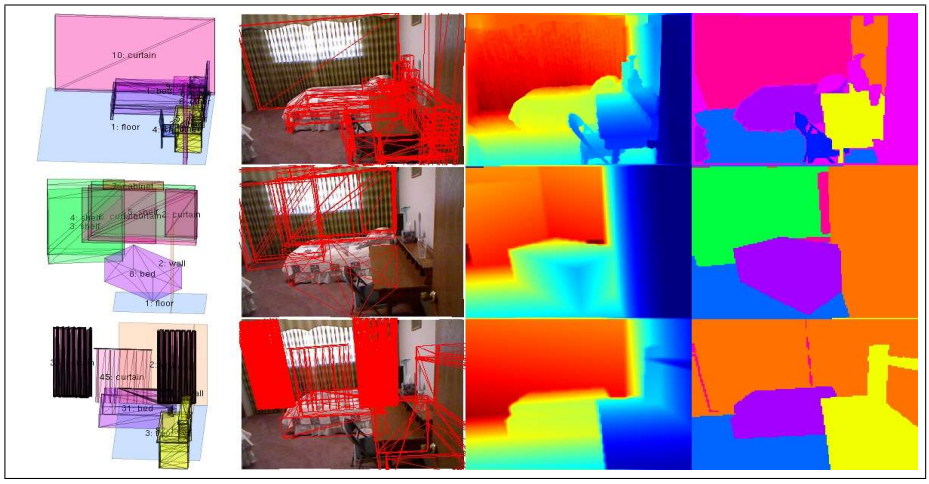

7: cabinet
1: floor
5: cabinet
8: sink
4: cabinet
1: floor
1: wall
2: floor

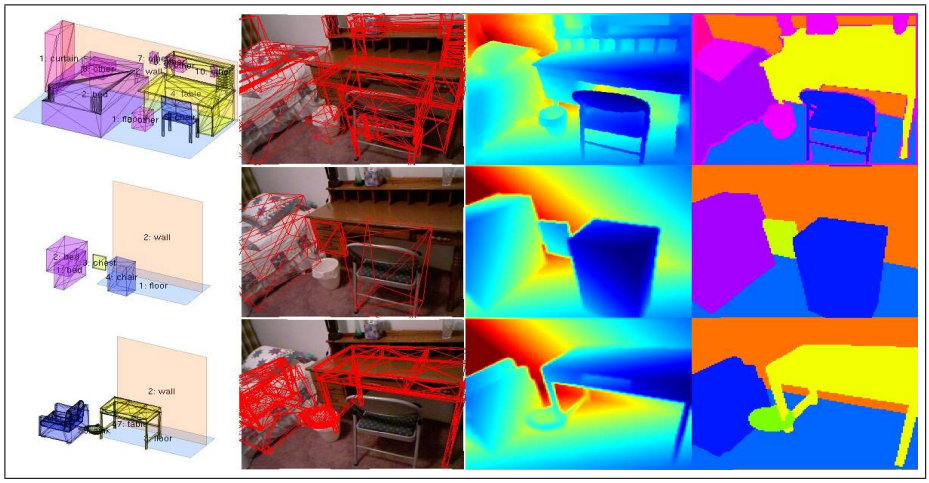


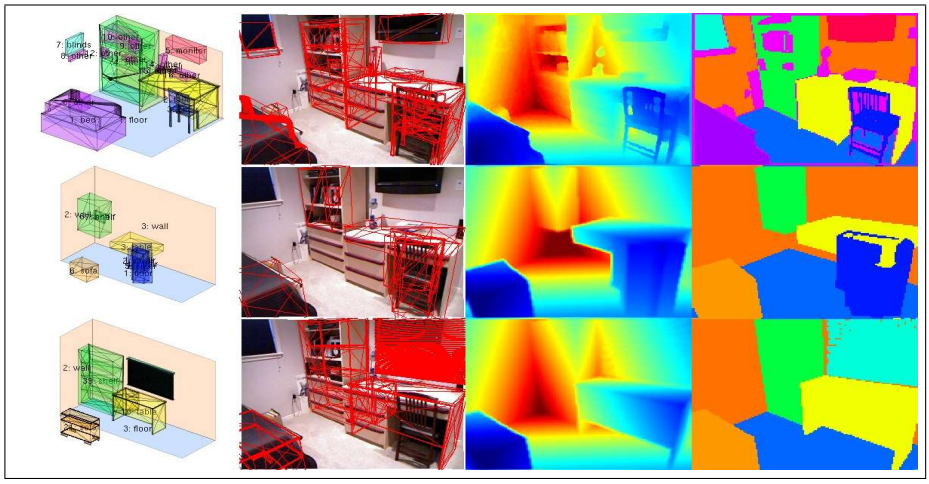


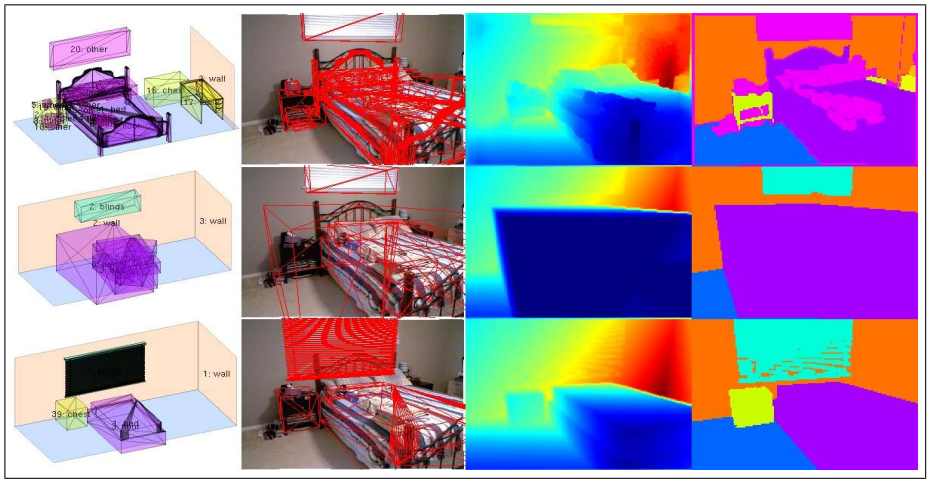


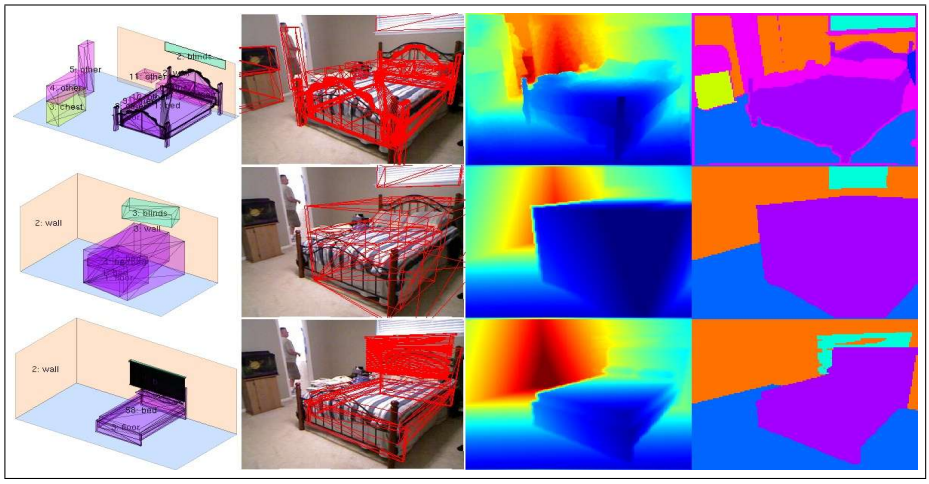


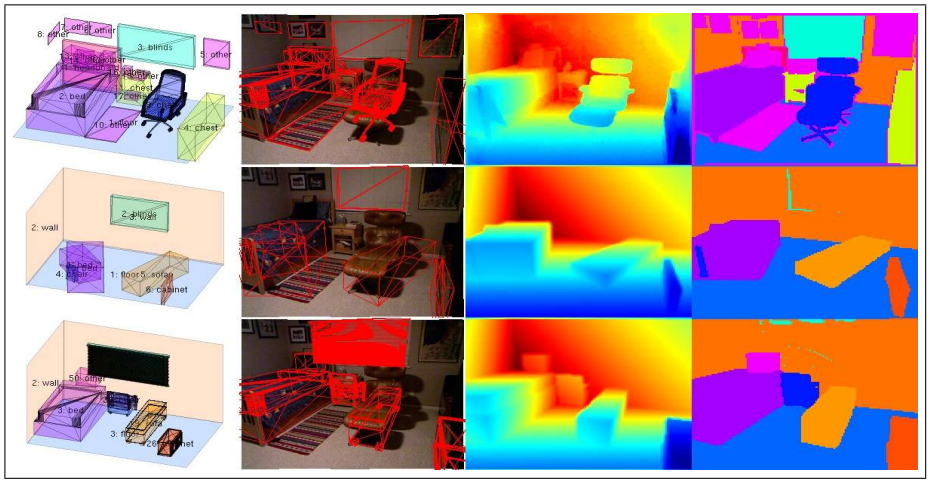

10: curtain
1: floor
2: wall
8: bed
1: floor
45: curtain
31: bed

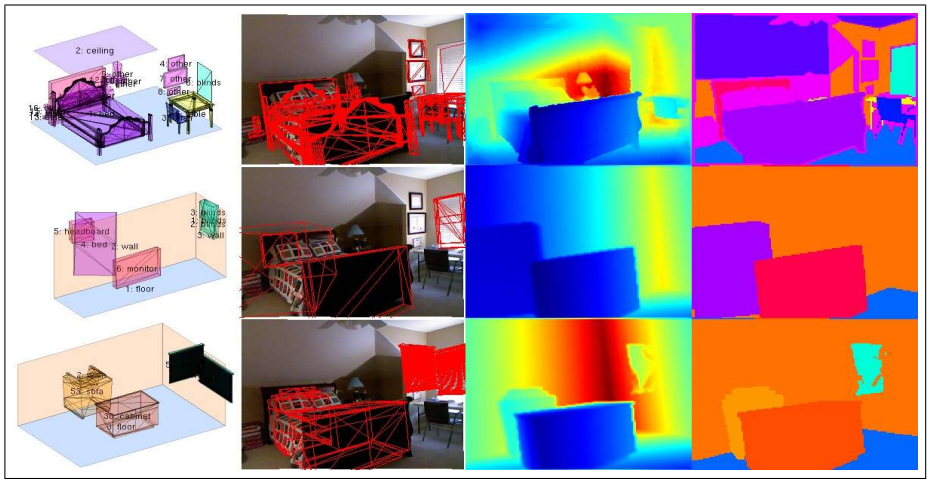

1: curtain
2: wall
2: bed
4: chair
1: floor
2: wall
2: wall
1: floor

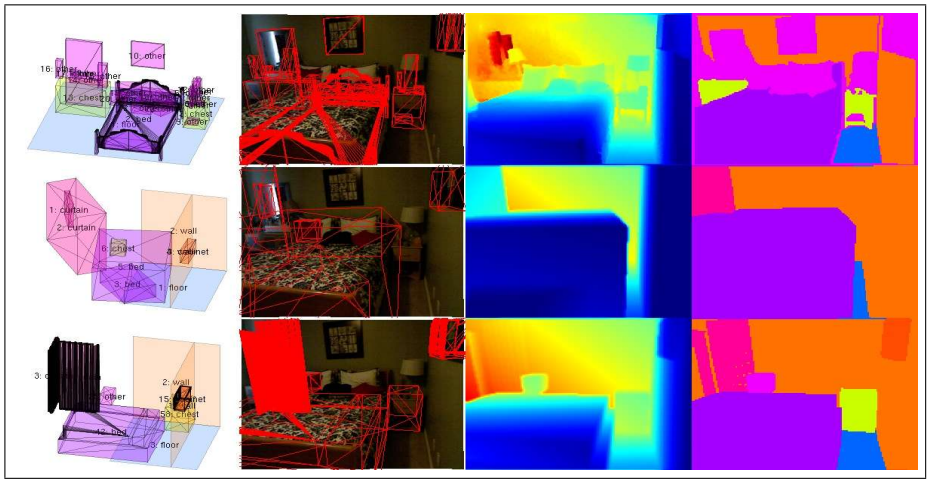

7: blinds
10: other
9: table
5: monitor
1: bed
1: floor
2: wall
3: wall
6: sofa
2: wall
3: floor

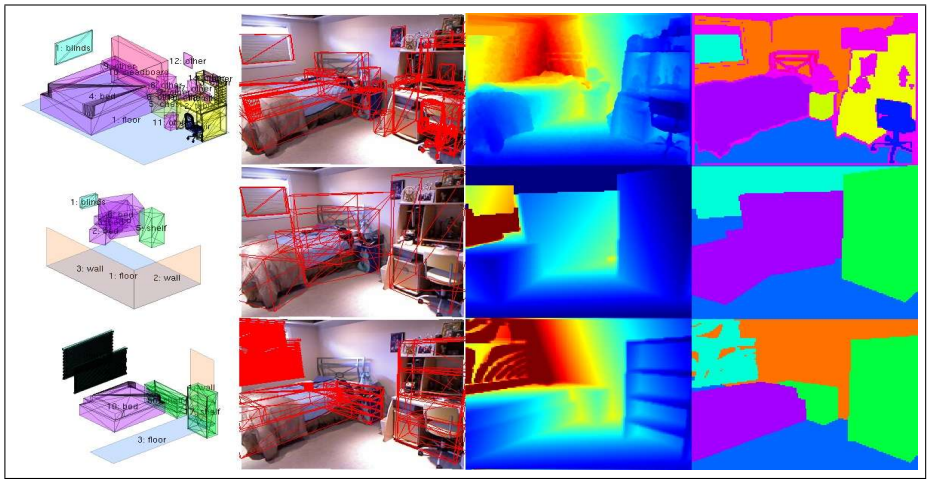

20: other
3: wall
2: blinds
2: wall
3: wall
1: wall
33: chest
5: bed

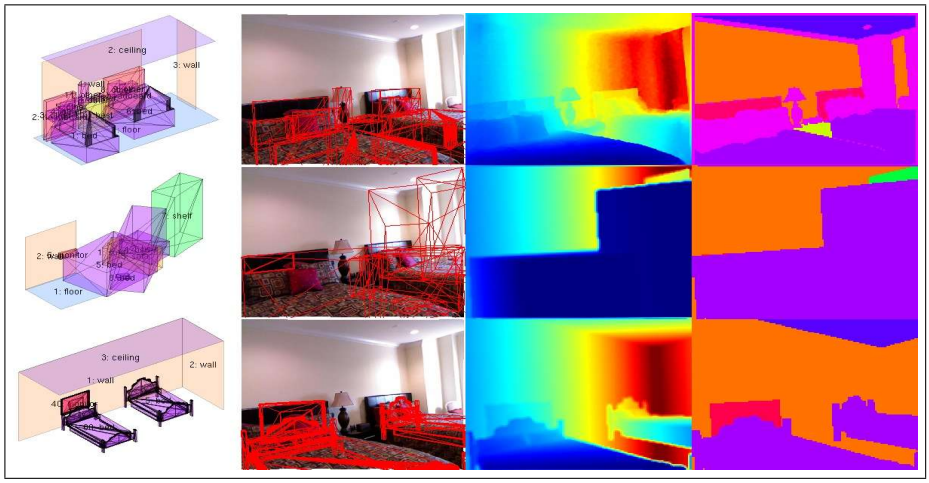


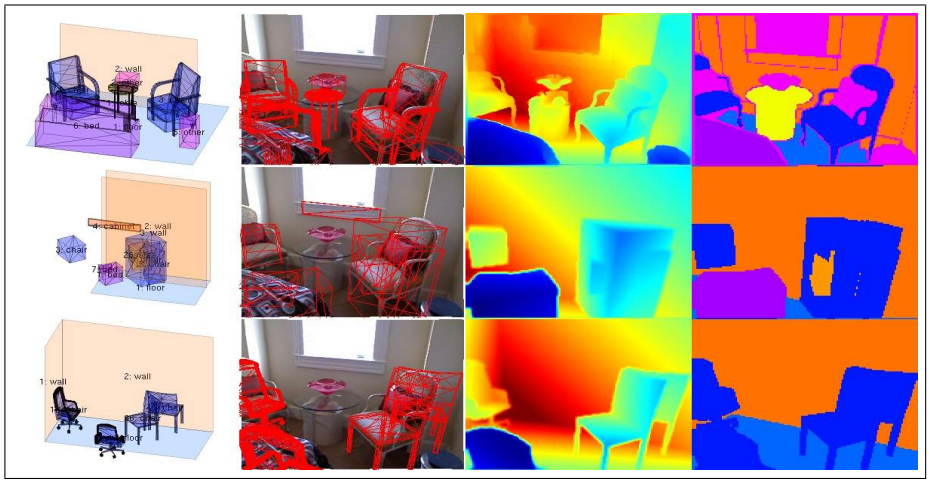


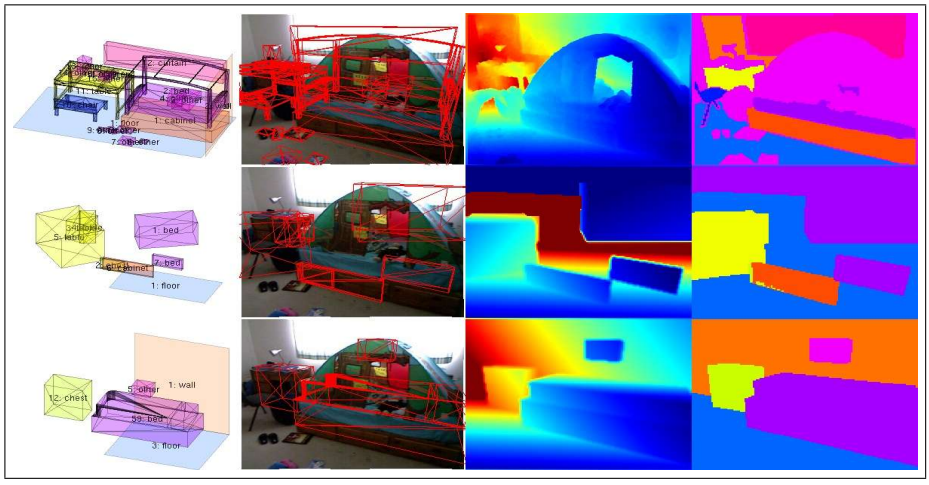

2: ceiling
4: other
7: other
8: other
5: headboard
4: bed
2: wall
6: monitor
1: floor
53: sofa
38: cabinet
1: floor

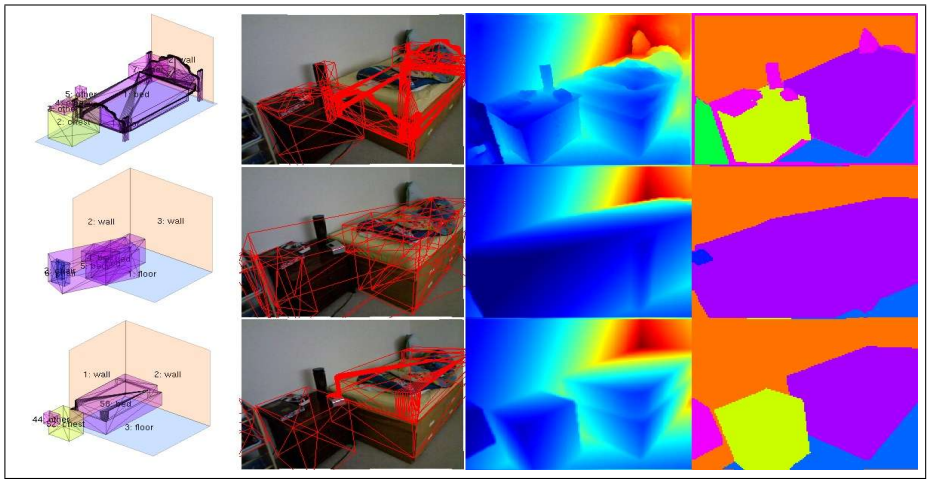

10: other
16: other
15: chest
1: curtain
2: curtain
2: wall
6: chest
8: cabinet
5: bed
3: bed
1: floor
2: wall
15: cabinet
38: chest
12: bed
3: floor

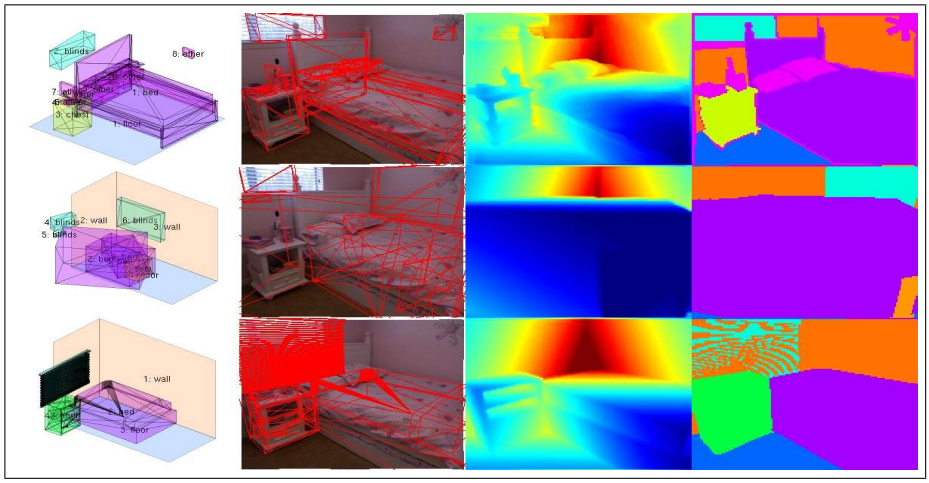

1: blinds
12: other
4: bed
1: floor
11: door
1: blinds
2: bed
5: shelf
3: wall
1: floor
2: wall
3: floor

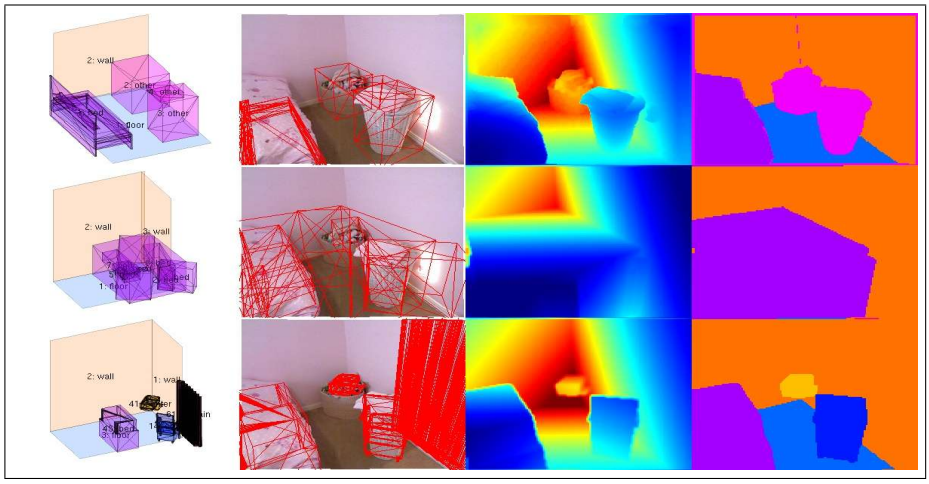

2: ceiling
3: wall
4: wall
6: bed
1: floor
7: shelf
2: monitor
5: bed
1: floor
3: ceiling
2: wall
1: wall

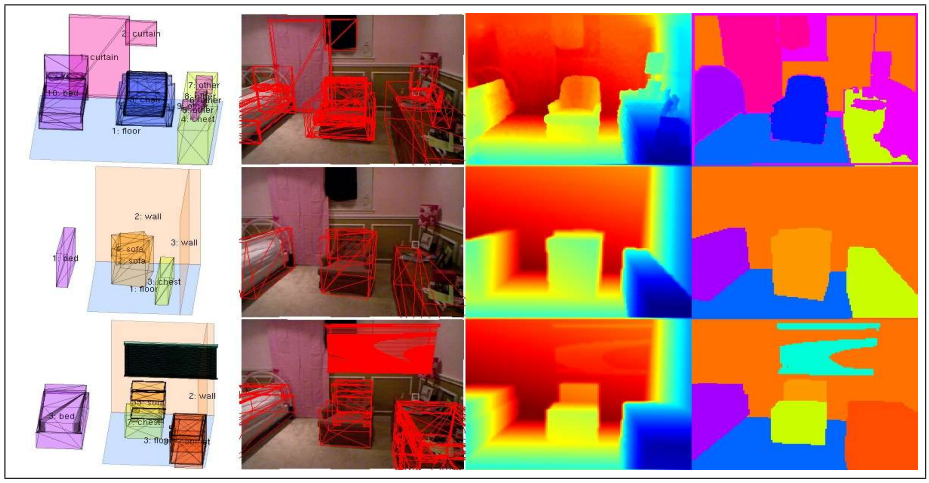


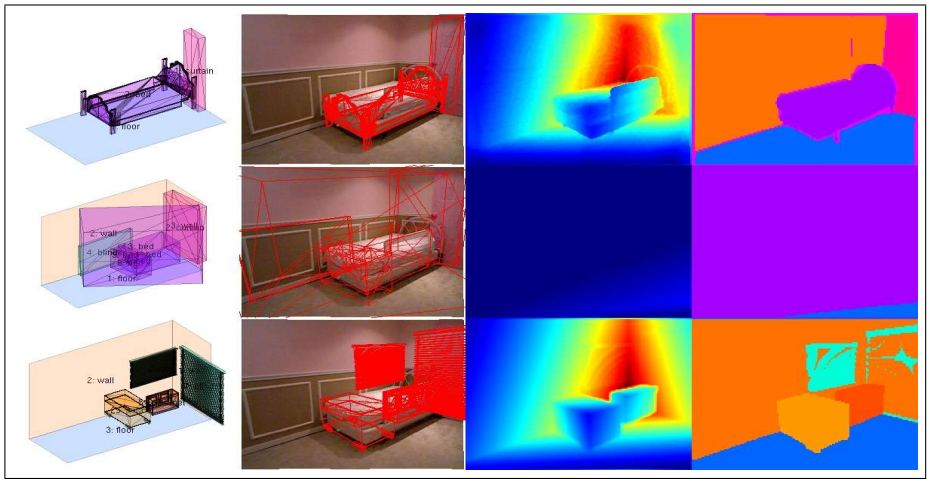







2: wall
2: other
3: other
1: bed
1: floor
2: wall
3: wall
1: floor
2: wall
1: wall
1: floor


2: curtain
1: curtain
10: bed
7: other
8: other
4: chest
1: floor
2: wall
3: wall
1: bed
2: sofa
1: sofa
5: chest
1: floor
2: wall
3: bed
3: floor


1: curtain
2: bed
3: floor
2: wall
2: wall
4: blind
5: bed
6: night
1: floor
2: wall
3: floor

References

1. Aubry, M., Maturana, D., Efros, A., Russell, B., Sivic, J.: Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In: CVPR (2014)
2. Choi, W., Chao, Y.W., Pantofaru, C., Savarese, S.: Understanding indoor scenes using 3D geometric phrases. In: CVPR (2013)
3. Guo, R., Hoiem, D.: Support surface prediction in indoor scenes. In: ICCV (2013)
4. Hedau, V., Hoiem, D., Forsyth, D.: Thinking inside the box: Using appearance models and context based on room geometry. In: ECCV (2010)
5. Jia, Z., Gallagher, A., Saxena, A., Chen, T.: 3D-based reasoning with blocks, support, and stability. In: CVPR (2013)
6. Jiang, H., Xiao, J.: A linear approach to matching cuboids in RGB-D images. In: CVPR (2013)
7. Kim, B.S., Kohli, P., Savarese, S.: 3D scene understanding by Voxel-CRF. In: ICCV (2013)
8. Komodakis, N., Paragios, N.: Beyond pairwise energies: Efficient optimization for higher-order MRFs. In: CVPR (2009)
9. Lim, J.J., Khosla, A., Torralba, A.: FPM: Fine pose parts-based model with 3D CAD models. In: ECCV (2014)
10. Lin, D., Fidler, S., Urtasun, R.: Holistic scene understanding for 3D object detection with RGB-D cameras. In: ICCV (2013)
11. Pero, L.D., Bowdish, J., Kermgard, B., Hartley, E., Barnard, K.: Understanding bayesian rooms using composite 3D object models. In: CVPR (2013)
12. Rother, C., Kohli, P., Feng, W., Jia, J.: Minimizing sparse higher order energy functions of discrete variables. In: CVPR (2009)
13. Satkin, S., Hebert, M.: 3DNN: viewpoint invariant 3D geometry matching for scene understanding. In: ICCV (2013)
14. Schwing, A.G., Fidler, S., Pollefeys, M., Urtasun, R.: Box in the box: Joint 3D layout and object reasoning from single images. In: ICCV (2013)
15. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGB-D images. In: ECCV (2012)
16. Zhang, Y., Song, S., Tan, P., Xiao, J.: PanoContext: A whole-room 3D context model for panoramic scene understanding. In: ECCV (2014)