Supplementary Material
Line-based 3D Building Abstraction and Polygonal Surface Reconstruction from Images

Jianwei Guo, Yanchao Liu, Xin Song, Haoyu Liu, Xiaopeng Zhang, Zhanglin Cheng

1 Plane hypotheses generation

This supplementary material gives the technical details of obtaining a set of candidate planes $\{f_1, f_2, \cdots, f_k\}$, (where k is unknown) from a given 3D line cloud $\mathcal{L} = \{l_1, l_2, \cdots, l_n\}$.

Geometry representation. Each 3D line segment is represented by using two end points:

$$l = \mathbf{p}_1 : (x_1, y_1, z_1) \rightarrow \mathbf{p}_2 : (x_2, y_2, z_2)\,.$$

For a plane, we use a point in it and a normal vector to represent:

$$f = \{\mathbf{v} : (a, b, c),\, \mathbf{n} : (n_x, n_y, n_z)\}\,.$$

Probability modeling. Assume we already know the k which indicates the number of planes and the probability distribution of those planes should satisfy: $\mathrm{P}(f_1), \mathrm{P}(f_2), \cdots, \mathrm{P}(f_k)$, $\sum_{j=1}^{k} \mathrm{P}(f_j) = 1$. Additionally, we define the conditional probability hypothesis of a 3D line segment $l_i = (\mathbf{p}_{i1}, \mathbf{p}_{i2})$ belonging to a plane f_j as:

$$\mathrm{P}(l_i\,|f_j) = \frac{1}{\sqrt{2\pi}\sigma_j}\exp\{-\frac{[(\mathbf{p}_{i1} - \mathbf{v}_j)\cdot\mathbf{n}_j]^2 + [(\mathbf{p}_{i2} - \mathbf{v}_j)\cdot\mathbf{n}_j]^2}{2\sigma_j^2}\}.$$

Then, we can get the joint probability distribution of the line segment l_i and the plane f_i by using Bayes' theorem:

$$\mathrm{P}(l_i, f_j) = \mathrm{P}(l_i\,|f_j)\mathrm{P}(f_j)\,.$$

The marginal probability is exactly the probability that the line l_i occurs:

$$\mathrm{P}(l_i) = \sum_{j=1}^{k} \mathrm{P}(l_i\,|f_j)\mathrm{P}(f_j)\,.$$

Since we get the probability of each sample line segment and each sample is regarded independent, we will maximize the likelihood function $\mathrm{P}(D|\theta)$ to get parameters of each plane f_j:

$$\max \prod_{i=1}^{n} \mathrm{P}(l_i)\,.$$

Here we use θ to denote all parameters in our likelihood function of the parameterized model.

Parameters estimation. Since the product of probabilities cost too much, we use the log-likelihood function instead:

$$\ln[\mathrm{P}(D|\theta)] = \sum_{i=1}^{n} \ln[\mathrm{P}(l_i|\theta)] = \sum_{i=1}^{n} \ln[\sum_{j=1}^{k} \mathrm{P}(l_i|f_j, \theta_j)\mathrm{P}(f_j)]\,.$$

In order to calculate the likelihood, we use a hidden variable γ_{ij} to indicate which plane f_j the line l_i belongs to:

$$\gamma_{ij} = \begin{cases} 1, & l_i \in f_j \\ 0, & otherwise \end{cases}$$

So the complete likelihood function can be written as:

$$\begin{aligned} \mathrm{P}(l, \gamma|\theta) &= \prod_{i=1}^{n} \mathrm{P}(l_i, \gamma_{i1}, \gamma_{i2}, \ldots, \gamma_{ik}|\theta) \\ &= \prod_{i=1}^{n} \prod_{j=1}^{k} [\mathrm{P}(l_i|f_j, \theta_j)\mathrm{P}(f_j)]^{\gamma_{ij}} \\ &= \prod_{j=1}^{k} \mathrm{P}(f_j)^{c_j} \prod_{i=1}^{n} [\mathrm{P}(l_i|f_j, \theta_j)]^{\gamma_{ij}}\ , \end{aligned}$$

where $c_j = \sum_{i=1}^{n} \gamma_{ij}$. Then the log-likelihood function on data D is defined as below:

$$\begin{aligned} \ln \mathrm{P}(l, \gamma|\theta) = \sum_{j=1}^{k} \Bigg\{ c_j \ln \mathrm{P}(f_j) + \sum_{i=1}^{n} \gamma_{ij} \Bigg[\ln \frac{1}{\sqrt{2\pi}\sigma_j} \\ - \frac{1}{2\sigma_j^2}\left(((\mathbf{p}_{i1} - \mathbf{v}_j)\cdot\mathbf{n}_j)^2 + ((\mathbf{p}_{i2} - \mathbf{v}_j)\cdot\mathbf{n}_j)^2\right) \Bigg] \Bigg\}\,. \end{aligned}$$

EM algorithm for optimization. Because the likelihood function contains hidden variables, we use the EM algorithm to solve the maximum problem.

- *Jianwei Guo and Xiaopeng Zhang are with NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. (Jianwei Guo and Yanchao Liu are joint first authors with equal contribution.)*
- *Yanchao Liu and Haoyu Liu are with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and Institute of Automation, Chinese Academy of Sciences.*
- *Xin Song and Zhanglin Cheng are with the Shenzhen Key Laboratory of Visual Computing and Analytics (VisuCA), Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences, Shenzhen 518055, China. (Zhanglin Cheng is the corresponding author. E-mail: zl.cheng@siat.ac.cn)*

First, *In the expectation (E) stage*, let

$$\begin{aligned} \mathrm{Q}(\theta,\theta^{(m)}) =&\mathcal{E}[\ln \mathrm{P}(l,\gamma|\theta)|l,\theta^{(m)}] \\ =&\mathcal{E}\left\{\sum_{j=1}^{k} \left\{c_j \ln P(f_j) + \sum_{i=1}^{n} \gamma_{ij}\left[\ln\frac{1}{\sqrt{2\pi}\sigma_j}\right.\right.\right. \\ &-\left.\left.\left.\frac{((\mathbf{p}_{i1}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2+((\mathbf{p}_{i2}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2}{2\sigma_j^2}\right]\right\}\right\} \\ =&\sum_{j=1}^{k}\left\{\sum_{i=1}^{n} \mathcal{E}(\gamma_{ij})\ln P(f_j)+\sum_{i=1}^{n}\mathcal{E}(\gamma_{ij})\left[\ln\frac{1}{\sqrt{2\pi}\sigma_j}\right.\right. \\ &-\left.\left.\frac{((\mathbf{p}_{i1}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2+((\mathbf{p}_{i2}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2}{2\sigma_j^2}\right]\right\}, \end{aligned}$$

The expectation $\mathcal{E}(\gamma_{ij}|l_i,\theta)$ here is denoted as $\hat{\gamma}_{ij}$ for convenient which can be computed as:

$$\begin{aligned}\hat{\gamma}_{ij} =&\mathcal{E}(\gamma_{ij}|l_i,\theta)=P(\gamma_{ij}=1|l_i,\theta_j)\\ =&\frac{P(\gamma_{ij}=1,l_i|\theta_j)}{\sum_{j=1}^{k} P(\gamma_{ij}=1,l_i|\theta_j)}\\ =&\frac{P(l_i|\gamma_{ij}=1,\theta_j)P(\gamma_{ij}=1|\theta)}{\sum_{j=1}^{k} P(l_i|\gamma_{ij}=1,\theta_j)P(\gamma_{ij}=1|\theta)}\\ =&\frac{P(f_j)P(l_i|f_j,\theta)}{\sum_{j=1}^{k} P(f_j)P(l_i|f_j,\theta)}\,.\end{aligned}$$

With $\hat{\gamma}_{ij}, c_j=\sum_{i=1}^{n}\hat{\gamma}_{ij}$, we can easily get the Q function:

$$\begin{aligned}\mathrm{Q}(\theta,\theta^{(m)}) = \sum_{j=1}^{k} c_j\ln P(f_j)+\sum_{i=1}^{n}\hat{\gamma}_{ij}&\left[\ln\frac{1}{\sqrt{2\pi}\sigma_j}\right.\\ &-\left.\frac{((\mathbf{p}_{i1}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2+((\mathbf{p}_{i2}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2}{2\sigma_j^2}\right].\end{aligned}$$

Next, *in the maximum (M) stage*, we need maximize the Q function to get parameters $\theta^{(m)}$ in each iteration. The maximum problem could be formulated as:

$$\begin{aligned}&\max_{\theta}\ \mathrm{Q}(\theta,\theta^{(m)})\\ &s.t. \sum_{j=1}^{k}\mathrm{P}(f_j)=1\,.\end{aligned}$$

In general, we use Lagrange multiplier to convert the constrained optimization problem to a Lagrangian function:

$$\mathcal{L}=\mathrm{Q}(\theta,\theta^{(m)})+\lambda\left[1-\sum_{j=1}^{k}\mathrm{P}(f_j)\right]\,.$$

Then, we calculate the partial derivatives of Lagrange function for each parameter. Because our Lagrange function is a convex function, the function reaches maximum when all partial derivatives are 0 values. As for $P(f_j)$:

$$\begin{gathered}\frac{\partial\mathcal{L}}{\partial P(f_j)}=\frac{c_j}{P(f_j)}-\lambda=0\\ \Downarrow\\ c_j=\lambda P(f_j)\\ \Downarrow\\ \sum_{j=1}^{k}c_j=\sum_{j=1}^{k}\lambda P(f_j)\\ \Downarrow\\ \lambda=k\,.\end{gathered}$$

So we can get:

$$P(f_j)=\frac{c_j}{k}$$

(1) For σ_j, let the partial derivative be 0:

$$\frac{\partial\mathcal{L}}{\partial\sigma_j}=\sum_{i=1}^{n}\hat{\gamma}_{ij}\frac{((\mathbf{p}_{i1}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2+((\mathbf{p}_{i2}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2-\sigma_j^2}{\sigma_j^3}=0$$

Then we obtain the value of σ_j:

$$\sigma_j^2=\frac{\sum_{i=1}^{n}\hat{\gamma}_{ij}[((\mathbf{p}_{i1}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2+((\mathbf{p}_{i2}-\mathbf{v}_j)\cdot\mathbf{n}_j)^2]}{\sum_{i=1}^{n}\hat{\gamma}_{ij}}$$

(2) Similarly, let

$$\frac{\partial\mathcal{L}}{\partial\mathbf{v}_j}=\sum_{i=1}^{n}\frac{\hat{\gamma}_{ij}}{\sigma_j^2}[\mathbf{n}_j^T(\mathbf{p}_{i1}-\mathbf{v}_j)\mathbf{n}_j+\mathbf{n}_j^T(\mathbf{p}_{i2}-\mathbf{v}_j)\mathbf{n}_j]=0$$

Then, we can get the parameter of $\mathbf{v}_j$:

$$\begin{gathered}\frac{1}{\sigma_j^2}\sum_{i=1}^{n}\hat{\gamma}_{ij}\left[\mathbf{n}_j\mathbf{n}_j^T(\mathbf{p}_{i1}+\mathbf{p}_{i2}-2\mathbf{v}_j)\right]=0\\ \Downarrow\\ \frac{1}{\sigma_j^2}\mathbf{n}_j\mathbf{n}_j^T\left[\sum_{i=1}^{n}\hat{\gamma}_{ij}(\mathbf{p}_{i1}+\mathbf{p}_{i2})\right]=\frac{2}{\sigma_j^2}\mathbf{n}_j\mathbf{n}_j^T\left(\sum_{i=1}^{n}\hat{\gamma}_{ij}\right)\mathbf{v}_j\\ \Downarrow\\ \mathbf{v}_j=\frac{\sum_{i=1}^{n}\hat{\gamma}_{ij}(\mathbf{p}_{i1}+\mathbf{p}_{i2})}{2\sum_{i=1}^{n}\hat{\gamma}_{ij}}\end{gathered}$$

(3) Finally, setting the partial derivative as 0:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{n}_j}=-\sum_{i=1}^{n}\frac{\hat{\gamma}_{ij}}{\sigma_j^2}&[(\mathbf{p}_{i1}-\mathbf{v}_j)^T\mathbf{n}_j(\mathbf{p}_{i1}-\mathbf{v}_j)+\\ &(\mathbf{p}_{i2}-\mathbf{v}_j)^T\mathbf{n}_j(\mathbf{p}_{i2}-\mathbf{v}_j)]=0\,,\end{aligned}$$

we can get the parameter of $\mathbf{n}_j$:

$$\begin{aligned}-\frac{1}{\sigma_j^2}\left\{\sum_{i=1}^{n}\hat{\gamma}_{ij}\right.&\left[(\mathbf{p}_{i1}-\mathbf{v}_j)(\mathbf{p}_{i1}-\mathbf{v}_j)^T+\right.\\ &\left.\left.(\mathbf{p}_{i2}-\mathbf{v}_j)(\mathbf{p}_{i2}-\mathbf{v}_j)^T\right]\right\}\mathbf{n}_j=0\end{aligned}$$

For convenient in the late discuss, we use matrix $\mathbf{A}$ to denote the matrix in the left of equal sign above. To solve the linear equation $\mathbf{A}\mathbf{n}_j=\mathbf{0}$, considering the equation may have no non-zero solutions, we convert it to an optimization problem:

$$\min_{\mathbf{n}_j}\|A\mathbf{n}_j\|.$$

After adding the constraint of normal $\mathbf{n}_j$, we can get

$$\begin{aligned}&\min_{\mathbf{n}_j}\mathbf{n}_j^TA^TA\mathbf{n}_j\\ &s.t.\quad \mathbf{n}_j^T\mathbf{n}_j=1\end{aligned},$$

This problem is equal to the Rayleigh quotient problem. Since $\mathbf{A}$ is a semi-positive definite symmetric matrix, $\mathbf{n}_j$ must be the eigenvector corresponding to the minimum eigenvalue that is:

$$\mathbf{A}\hat{\mathbf{n}}_j=\lambda_{min}\hat{\mathbf{n}}_j\,.$$

TABLE 1
A quantitative study of the detected line segments on different image resolutions. The original resolution is 640×480. $|L|$ is the number of detected segments, and l_{avg} is average length of all line segments in one image.

Scene	Image No.	0.25x		0.5x		1.0x		2.0x		3.0x		4.0x	
		$\|L\|$	l_{avg}	$\|L\|$	l_{avg}	$\|L\|$	l_{avg}	$\|L\|$	l_{avg}	$\|L\|$	l_{avg}	$\|L\|$	l_{avg}
Indoor	1	124	78.2573	440	46.8173	1165	30.20	1553	22.1111	1728	16.9567	1734	13.6109
	2	133	79.0504	304	52.2456	721	32.89	888	25.2391	1026	18.665	1016	15.46
	3	83	87.6051	245	49.7764	630	29.15	926	19.2731	985	14.3942	932	11.163
	4	85	103.503	150	74.1285	263	48.33	291	41.3061	364	31.67	403	26.7109
	5	140	63.3353	356	42.1044	1047	22.36	1630	15.4868	1914	12.4044	2023	10.4497
	6	133	107.658	264	74.3048	617	44.12	681	34.934	690	29.9118	824	21.5923
	7	106	93.1815	272	61.4677	709	34.64	1036	23.4001	1245	17.1196	1263	14.3316
	8	90	76.6843	288	43.642	1005	20.95	1517	13.6175	1892	9.86138	2055	8.28433
	9	133	76.0159	363	50.3593	1086	25.61	1617	17.0247	1888	13.2578	2051	11.0655
	10	76	82.6764	151	68.1072	361	37.11	443	28.8096	500	21.0465	485	16.2492
Outdoor	1	93	79.4491	294	42.7977	1544	22.37	2104	17.2915	2463	12.1952	2323	9.74857
	2	99	60.5676	273	38.201	806	20.64	1497	13.1333	2294	8.70146	2723	6.56693
	3	50	94.6057	126	54.6287	427	23.37	778	14.3778	1204	9.48372	1427	7.39014
	4	45	78.3989	138	44.9543	972	18.56	1695	12.0164	2023	8.76439	1925	7.40929
	5	55	89.7073	176	49.5047	512	27.87	858	17.4857	1052	12.7129	1010	11.2438
	6	86	68.9187	258	40.204	907	18.49	1737	10.565	2857	6.63371	3327	5.25962
	7	125	81.037	410	50.7672	1243	26.80	1905	18.1855	2373	13.3093	2461	11.379
	8	137	71.7757	428	43.2967	1393	25.41	2481	15.7332	3367	10.4702	3597	8.0886
	9	114	72.5957	260	50.7367	691	26.39	1059	17.36	1368	12.1259	1355	10.5473
	10	81	68.0393	228	40.5233	876	20.92	1208	15.0142	1399	11.4713	1489	9.42491

2 Discussion and More results

As we show in the main paper, the quality of extracted line segment features also depends on the image resolution. In Table 1 in this appendix, we use 20 images of indoor and outdoor scenes from *YorkUrbanDB* dataset [7] to conduct a quantitative study. We set six sub/upsampling levels (the scaling factor is s) to count the number of detected line segments and compute the average length of all line segments under different image resolutions. As we can see, as scaling factor s increases, the number of detected line segments increases but the average length drops. It indicates that the detection results under high image resolution are more precise but generate more broken line segments, thus lacking completeness and continuity. Besides, when s is larger than 2.0, the detection quality would not change too much. On the other hand, we detected a smaller number of line segments with longer length on low resolution images, meaning that it can better capture global structures but lose precision. Especially when the sub-sampling factor is smaller than 0.5, it would introduce large detection errors. Therefore, we use three levels, and set the sampling factor as 0.5, 1.0, 2.0, which works well in our used examples.

Fig. 1 shows the full comparison with all of previous clustering methods and plane detection methods on two real-world data. The visual comparison results demonstrate that our approach achieves the best performance in terms of clustering 3D line segments for plane detection.

References

[1] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996, pp. 226–231.

[3] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[4] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Comput. Graph. Forum*, vol. 26, no. 2, 2007, pp. 214–226.

[5] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra, "Globfit: Consistently fitting primitives by discovering global relations," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–12, 2011.

[6] F. A. Limberger and M. M. Oliveira, "Real-time detection of planar regions in unorganized point clouds," *Pattern Recognition*, vol. 48, no. 6, pp. 2043–2053, 2015.

[7] P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating manhattan frames in urban imagery," in *European Conference on Computer Vision (ECCV)*, 2008, pp. 197–210.

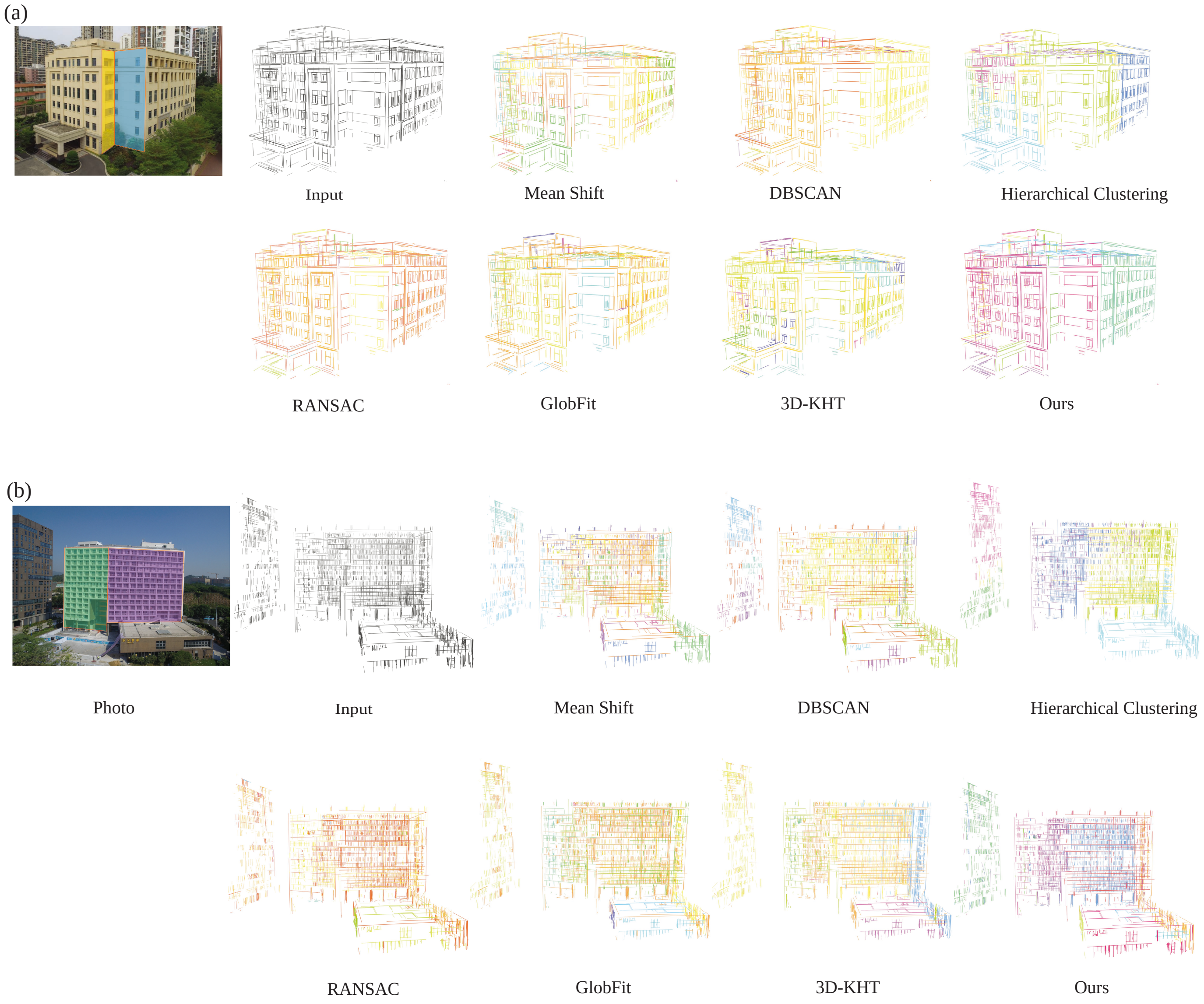

Fig. 1. Visual comparison of different clustering methods using real-world data. For each example, we show the reference photo, input 3D line clouds, the clustering results of mean shift [1], DBSCAN [2], hierarchical clustering [3], RANSAC [4], GlobFit [5], 3D-KHT [6] and ours.