

# How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines

## Supplementary Material

Kunal Chelani<sup>1</sup> Fredrik Kahl<sup>1</sup> Torsten Sattler<sup>1,2</sup>

<sup>1</sup>Chalmers University of Technology <sup>2</sup>Czech Technical University in Prague

This supplementary material provides the following information: Sec. 1 provides additional details about the **peak finding** step of our algorithm (*cf.* Sec. 4.2 in the main paper). Sec. 2 provides additional details for the experiments conducted in the main paper, including more qualitative results (*cf.* Sec. 5 in the main paper).

### 1. Peak Finding Algorithm

In the following, we provide additional details on the peak finding algorithm used in our approach.

The input to the peak finding algorithm is a set of candidates for the 3D point position  $\mathbf{p}_i$ . These candidates are obtained as the closest points on the line  $\mathbf{l}_i$  to the lines contained in the neighborhood  $\mathcal{N}^K(\mathbf{l}_i)$  provided by the previous stage of our algorithm (Neighborhood Estimation, *cf.* Sec. 4.2 in the main paper). Each candidate is parameterized as a scalar value  $\beta_{ij}$  that provides a 3D point position on the line  $\mathbf{l}_i = \mathbf{o}_i + \beta_{ij}\mathbf{v}_i$ . Here,  $\mathbf{o}_i$  is any point on the line and  $\mathbf{v}_i$  is a unit vector describing the direction of the line. The set can thus be written as:

$$\mathcal{E}_i = \{\beta_{ij} | \hat{\mathbf{p}}_{ij} = \mathbf{o}_i + \beta_{ij}\mathbf{v}_i, \forall j \in \mathcal{N}^K(i)\} . \quad (1)$$

We use these candidates to compute the unweighted empirical cumulative distribution function (CDF) of the candidates along the line as

$$F_i(x) = \frac{1}{K} \sum_{j=1}^K I_{\beta_{ij} < x} . \quad (2)$$

As described in the main paper,  $I_{\beta_{ij} < x}$  is an indicator variable taking value 1 if  $\beta_{ij} < x$  and 0 otherwise. This CDF is compared against the CDF  $F_U(x)$  of a uniform distribution of points along the line.<sup>1</sup>

As described in the main paper, the Kuiper's statistic (KS) is used to compare the two CDFs. More precisely, the KS is used to identify regions where both CDFs differ

<sup>1</sup>For practical reasons, we only consider the interval between the minimum and maximum values from  $\mathcal{E}_i$  when computing  $F_U(x)$ .

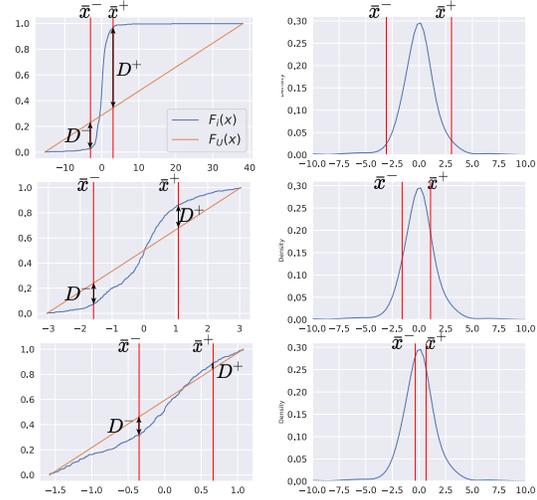


Figure 1. Illustrative example of our peak finding algorithm for a KS threshold of 0.3. The rows from top to bottom indicate three subsequent iterations of our approach, starting from the top. Left column: empirical cumulative distribution function (CDF)  $\mathbf{F}_i(x)$  of the point position candidates and the CDF of a Uniform distribution  $\mathbf{F}_U(x)$ . We show the two points  $\bar{x}^-$  and  $\bar{x}^+$  defining the interval via red lines and also illustrate the meaning of  $D^-$  and  $D^+$ . Right column: the approximate density distribution of the candidates along the line, together with the intervals defined by  $\bar{x}^-$  and  $\bar{x}^+$ . As can be seen, the peak finding approach iteratively narrows down the interval towards the peak of the density distribution.

the most. Intuitively, these regions correspond to intervals along the line where there is a higher density of candidates than can be accounted for by a uniform distribution.

As detailed in the main paper, we compute the two points  $\bar{x}^- = \operatorname{argmax}_x (F_U(x) - F_i(x))$  and  $\bar{x}^+ = \operatorname{argmax}_x (F_i(x) - F_U(x))$  corresponding to the positions along the line where the two distributions differ most. The differences between the distributions at these points are given as  $D^- = (F_U(\bar{x}^-) - F_i(\bar{x}^-))$  and  $D^+ = (F_i(\bar{x}^+) - F_U(\bar{x}^+))$  and the Kuiper's statistic is then de-

Scene	Point Density	Iteration-1 (Coarse Estimation)	Iteration 2 and later (Refinement)
Outdoor Scenes from [1]	more than 5%	500	200
Outdoor Scenes from [1]	5% or less	100	50
Indoor Scenes from [3]	more than 5%	250	100
Indoor Scenes from [3]	5% or less	50	25

Table 1. Information about the number of nearest neighbours used for estimating point positions in different scenes, under varying densities of input line cloud and at different stages of estimation.

fined as  $KS = D^- + D^+$ . As illustrated in Fig. 1, we recursively use this process to find regions of high density: applying the process to the interval defined by  $\bar{x}^-$  and  $\bar{x}^+$  (shown via the red lines in the figure), the points  $\bar{x}^-$  and  $\bar{x}^+$  as well as the Kuiper’s Statistic are re-estimated within this interval. Once the KS falls below a given threshold, set to 0.3 for the example shown in Fig. 1, the recursion is aborted. In practice, we observe that a KS threshold in the range of 0.3 - 0.4 performs well.

The right column of Fig. 1 shows the density distribution along the line for the example considered in the figure. As can be seen, our peak finding approach iteratively shrinks the interval towards the peak of the density distribution.

There are also instances where multiple peaks are obtained in the distribution of estimates. Such cases correspond to, *e.g.*, situations where a line passes through more than one region in the scene that contains many of the original 3D points. To handle such cases, we break at points where  $F_U(x)$  intersects  $F_i(x)$  from below, *i.e.*  $F_U(x) = F_i(x)$  and  $F_i(x_-) - F_U(x_-) > 0$  where  $\epsilon = x - x_- > 0$  is very small. Let  $\{x_1, x_2 \dots x_k\}$  be such points of intersection. Then each of the ranges  $(x_{min}, x_1), (x_1, x_2), \dots (x_k, x_{max})$  corresponds to a particular peak. Among all these detected peaks, the one with the highest KS value is selected.

## 2. Detailed Results

This section provides additional qualitative results (*cf.* Sec. 5 in the main paper). It also contains a more detailed analysis of the impact of the neighborhood size on the quality of the recovered point clouds and images (*cf.* Sec. 4.1 and Fig. 3 in the main paper).

**Impact of the neighbourhood size.** In the main paper, we showed results for the case where an oracle provides the true neighborhood for each 3D point / line based on the original point cloud. Following the example provided in Sec. 4.1, we used the 50 closest neighbors for these experiments. In an ablation study, we thus study the impact of varying the number of true neighbors. We also list the number of nearest neighbouring lines/points used in our recovery algorithm for different scenes in table

Fig. 2 shows cumulative distributions of the errors in recovering point positions when varying the number of true nearest neighbors (NN) for all scenes from the 12 Scenes

dataset [3]. As can be expected from the analysis presented in the main paper, using a smaller neighborhood leads to more accurate point estimates. This can be explained by the relation between the error in estimation of a point’s position and distance to neighbours used for estimation (*cf.* Sec. 4.1 in the main paper).

Fig. 3 shows the point clouds and images recovered using different numbers of nearest neighbors for the *Apt2-Kitchen* scene of the 12 Scenes dataset [3]. As can be seen, the point clouds and images (obtained via the SfM inversion process from [2] applied on our recovered point clouds) are visually similar for all numbers of nearest neighbors. This shows that the recovered point clouds do not need to be extremely accurate in order to be able to obtain good quality images.

**More results.** In contrast to Fig. 3, where we vary the number of true neighbors, we now fix the number of neighbors provided by the oracle to 50. As for Fig. 3 in the main paper, we instead vary the percentage of outliers among the neighbors by randomly replacing a fraction of the true neighbors with randomly selected points / lines. Fig. 4 quantitatively compares the cumulative distribution of errors in different scenarios for all scenes of the 12 scenes dataset. Figures 5 to 9 show additional qualitative results for the recovered point clouds and the images obtained from them. Figures 10 and 11 further show some qualitative results of our method for two of the outdoor scenes considered in our paper. As can be seen, our approach is able to faithfully recover the 3D point clouds and obtain detailed images as long as the neighborhoods do not contain too many unrelated points / lines. This holds both for the neighborhoods provided by the oracle and those estimated by our method. In particular, the results show that it is possible to recover image details via the point clouds estimated by our approach. As in the main paper, we thus conclude that lifting point clouds to line clouds alone does not guarantee that image details cannot be recovered.

## References

- [1] Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In: ICCV (2015) 2, 11
- [2] Pittaluga, F., Koppal, S.J., Kang, S.B., Sinha, S.N.: Revealing Scenes by Inverting Structure From Motion Reconstructions.

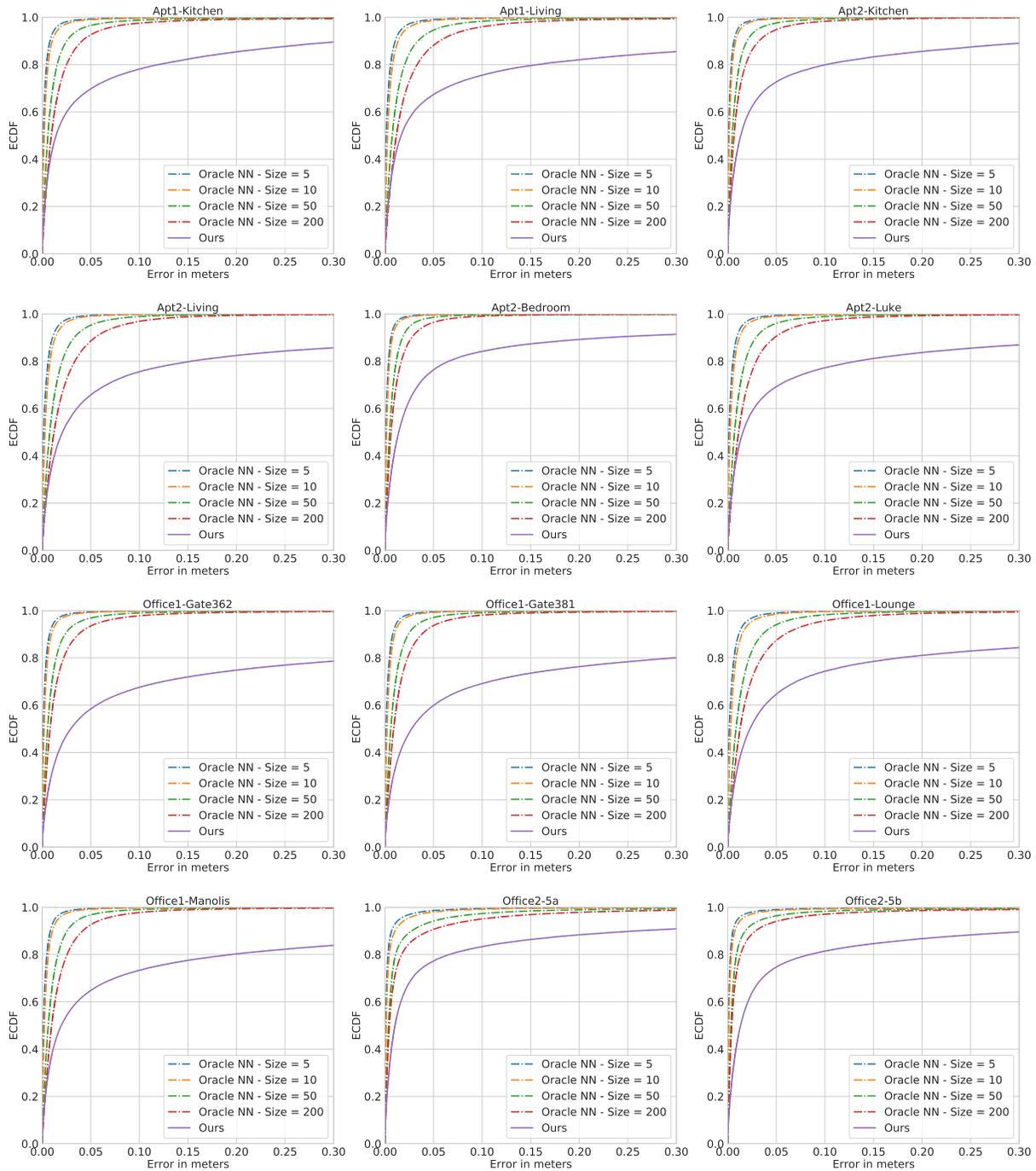


Figure 2. Quantitative results showing the cumulative distribution of errors in the recovering point positions. We show results obtained on all scenes of the 12 Scenes dataset [3], for our approach and when the true neighborhood of each point / line is provided by an oracle. For the latter, we vary the number of neighbors.

In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 4, 6, 7, 8, 9, 10, 11

- [3] Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In: CVPR (2013) 2, 3, 4, 5, 6, 7, 8, 9, 10

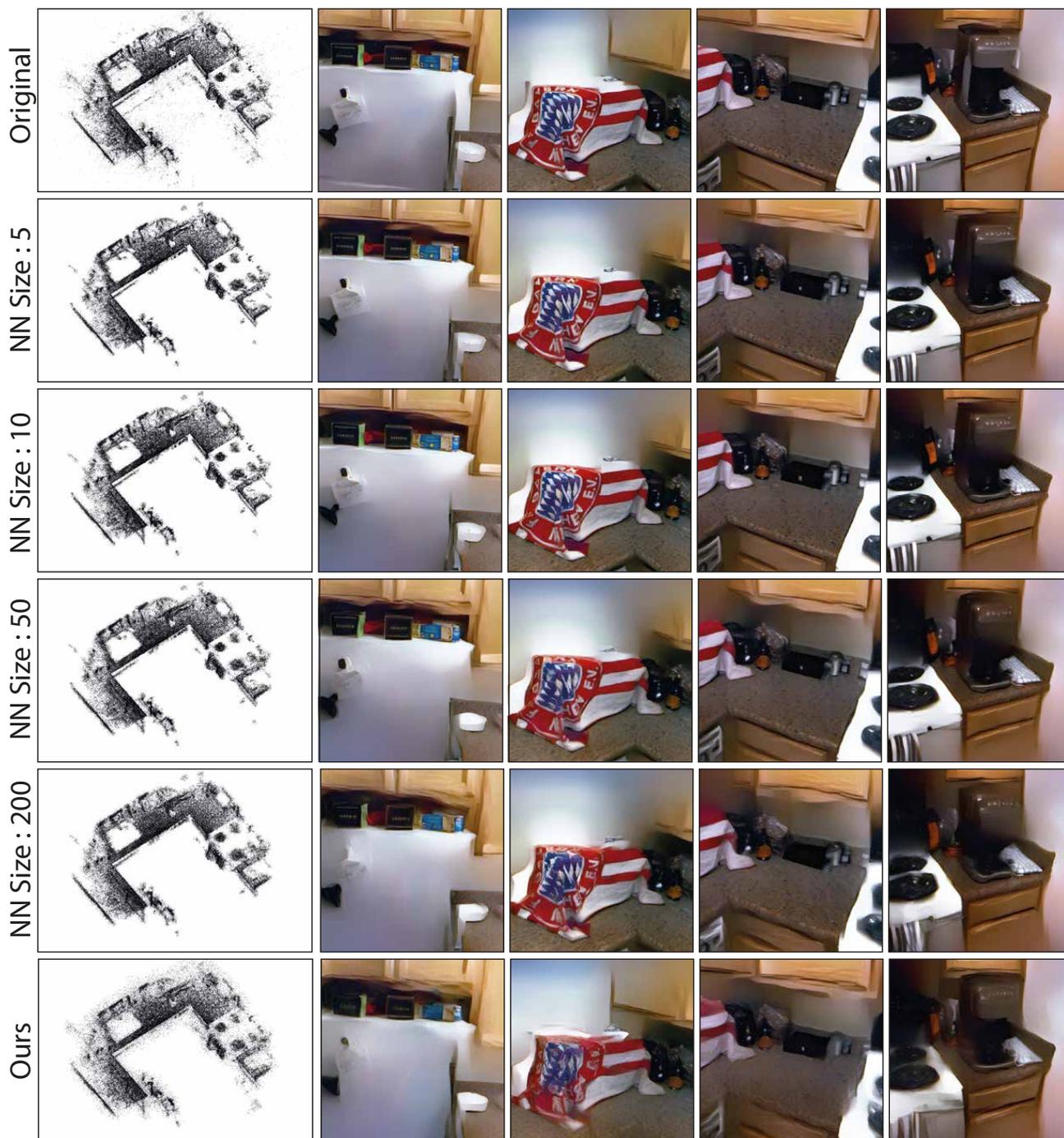


Figure 3. Qualitative results showing the impact of the size of neighbourhoods used for estimating the point position. For rows 2 through 5, the true neighbourhood is provided by an oracle and we vary the number of used neighbors. The results are shown on the "Apt2-Kitchen" scene from the 12 Scenes dataset [3]. In addition, we also show the original point cloud and the point cloud recovered by our full approach (which also estimates the neighborhoods). In each case, we show images obtained from the corresponding point clouds via the SfM inversion process from [2].

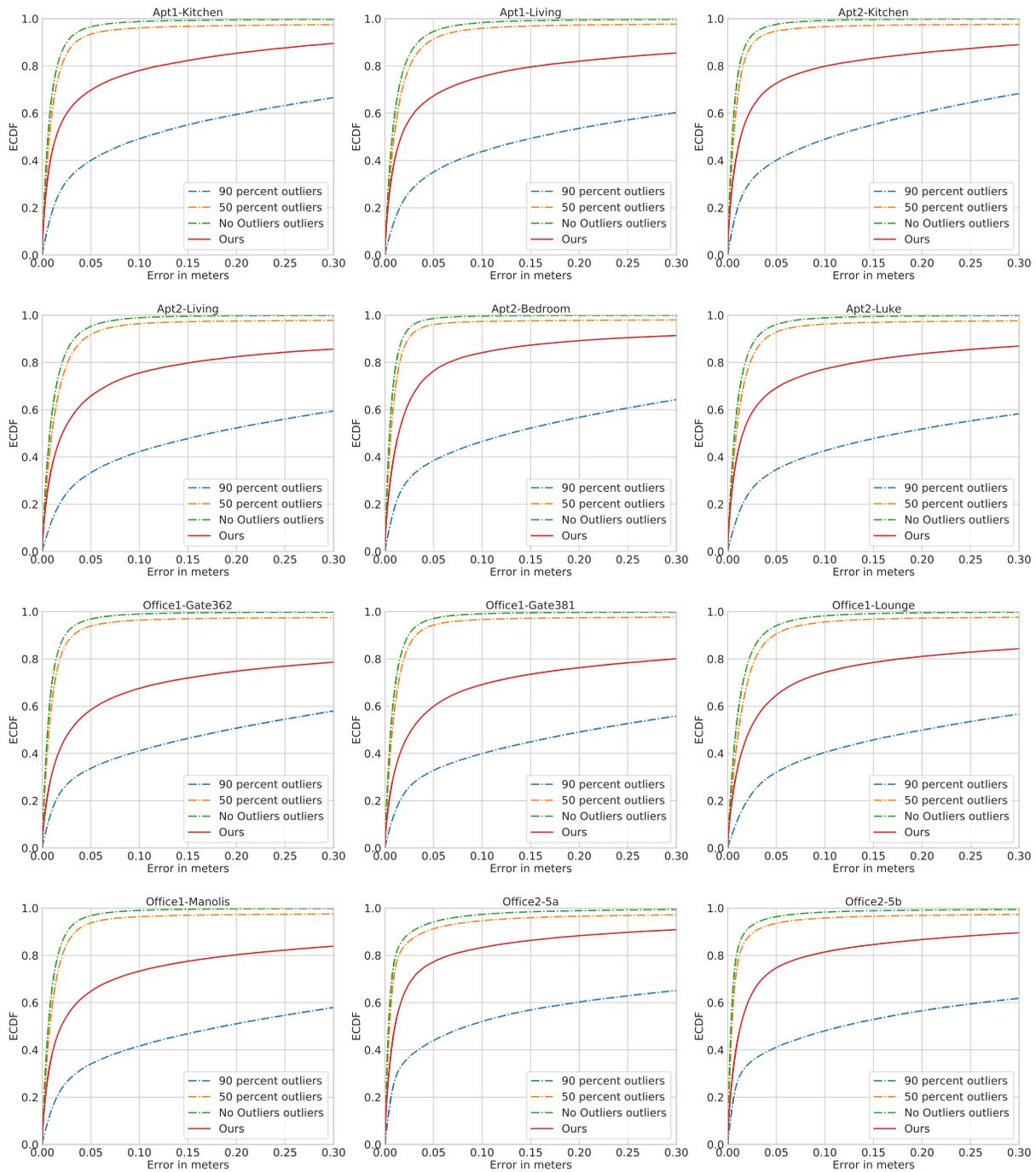


Figure 4. Quantitative results showing the cumulative distribution of errors in the recovering point positions. We show results obtained on all scenes of the 12 Scenes dataset [3], for our approach and when the true neighborhood (of size 50) of each point / line is provided by an oracle. For the latter, we vary the level of contamination by outliers.

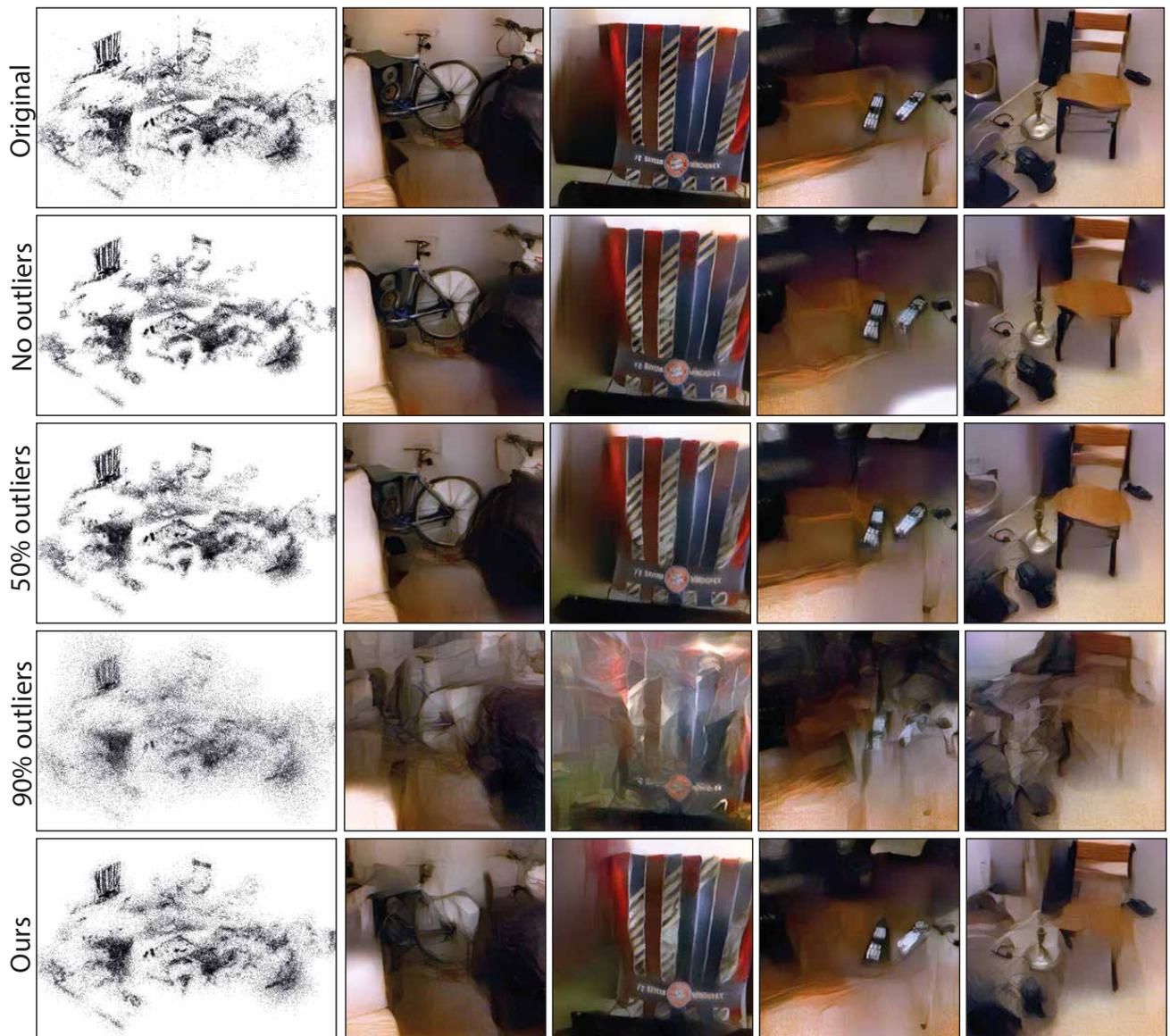


Figure 5. Qualitative results for the 'Apt2-Living' scene from the 12 Scenes dataset [3]. We show point clouds as well as the images recovered from them using the approach from [2]. Besides the original point cloud and the one recovered by our method, we also show results obtained by using an oracle to provide the neighborhood of each point / line. For these neighborhoods, we vary the percentage of outliers contained in them.

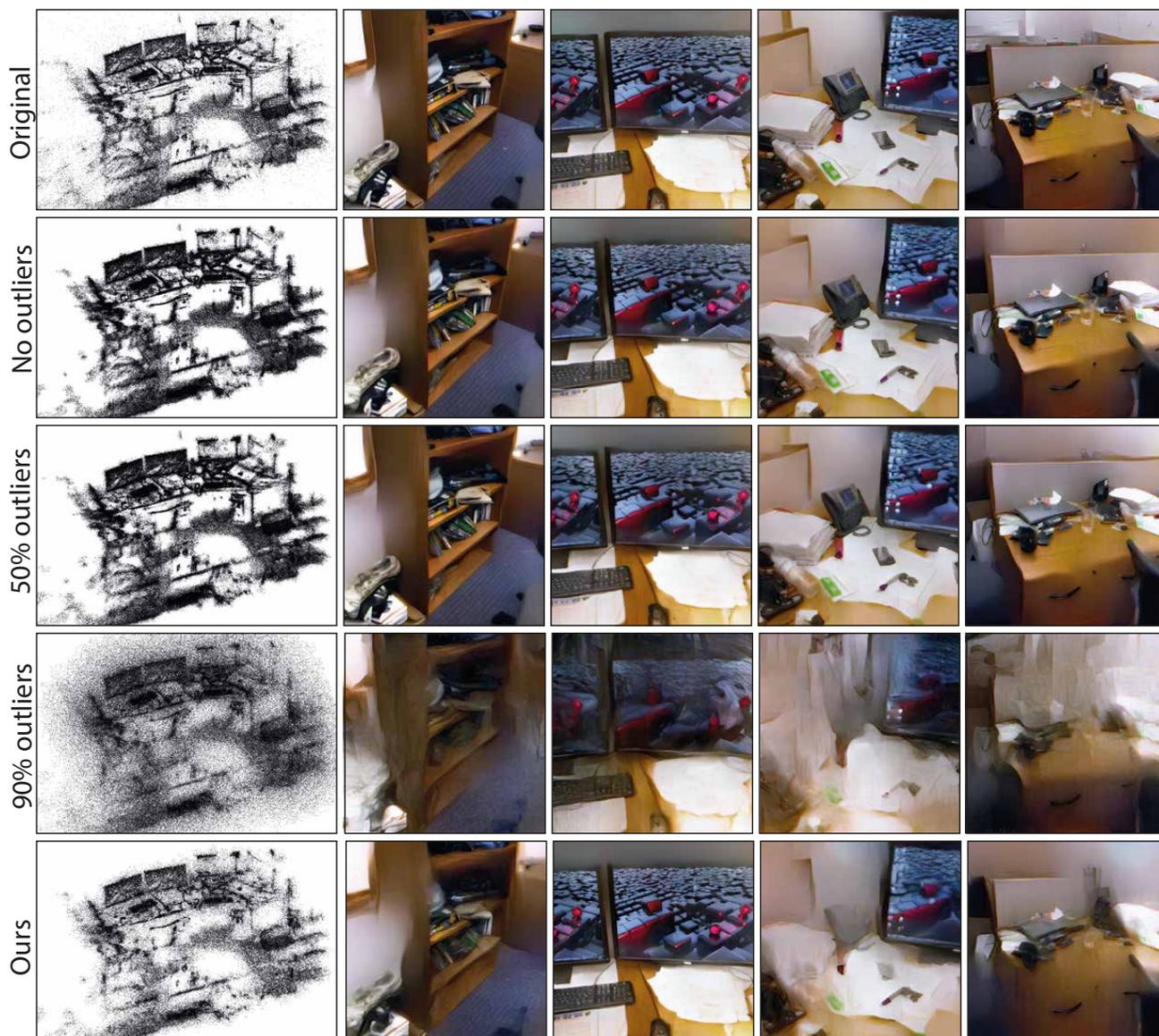


Figure 6. Qualitative results for the 'Office1-Gate362' scene from the 12 Scenes dataset [3]. We show point clouds as well as the images recovered from them using the approach from [2]. Besides the original point cloud and the one recovered by our method, we also show results obtained by using an oracle to provide the neighborhood of each point / line. For these neighborhoods, we vary the percentage of outliers contained in them.

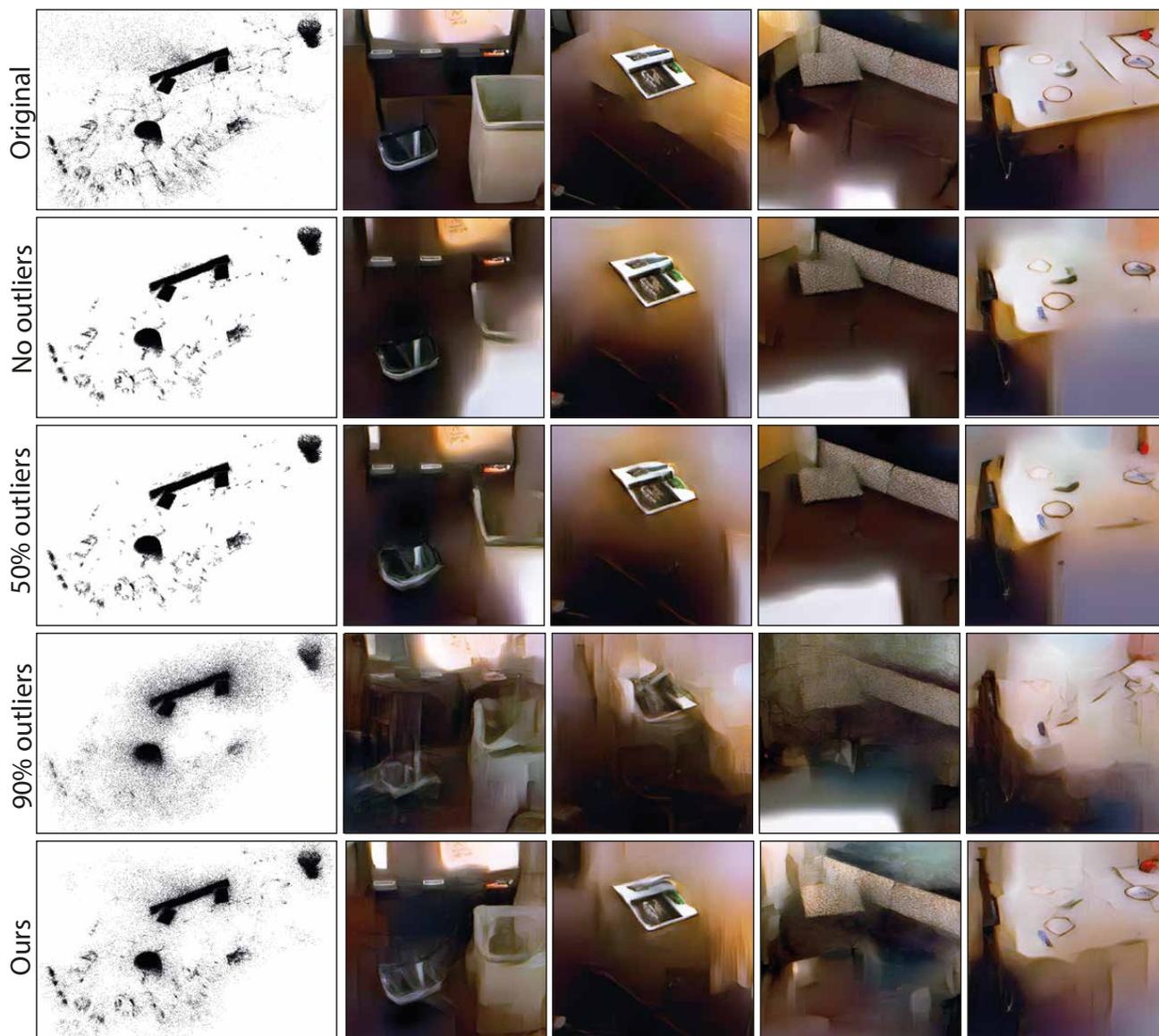


Figure 7. Qualitative results for the 'Office2-5a' scene from the 12 Scenes dataset [3]. We show point clouds as well as the images recovered from them using the approach from [2]. Besides the original point cloud and the one recovered by our method, we also show results obtained by using an oracle to provide the neighborhood of each point / line. For these neighborhoods, we vary the percentage of outliers contained in them.

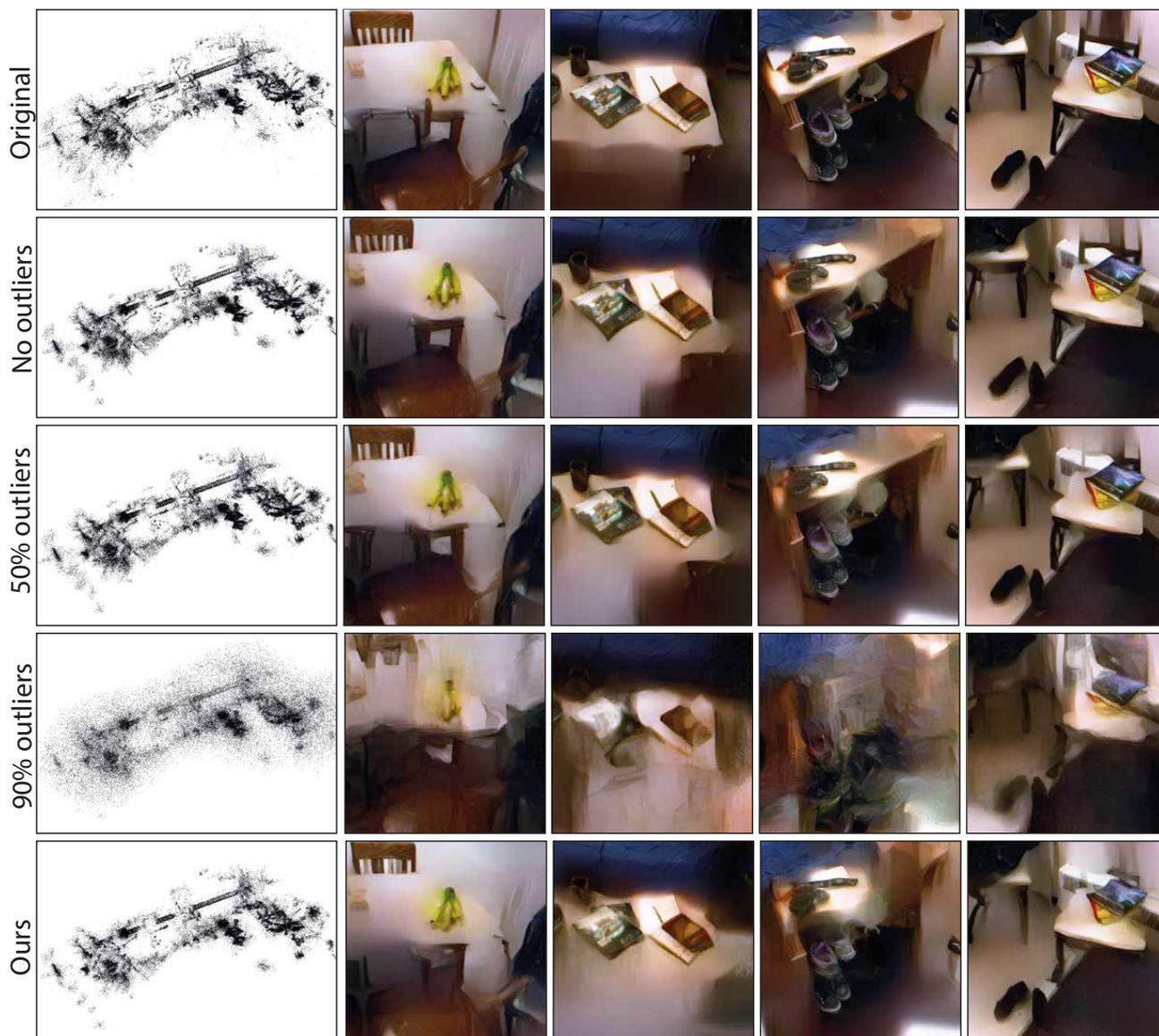


Figure 8. Qualitative results for the 'Apt1-Living' scene from the 12 Scenes dataset [3]. We show point clouds as well as the images recovered from them using the approach from [2]. Besides the original point cloud and the one recovered by our method, we also show results obtained by using an oracle to provide the neighborhood of each point / line. For these neighborhoods, we vary the percentage of outliers contained in them.

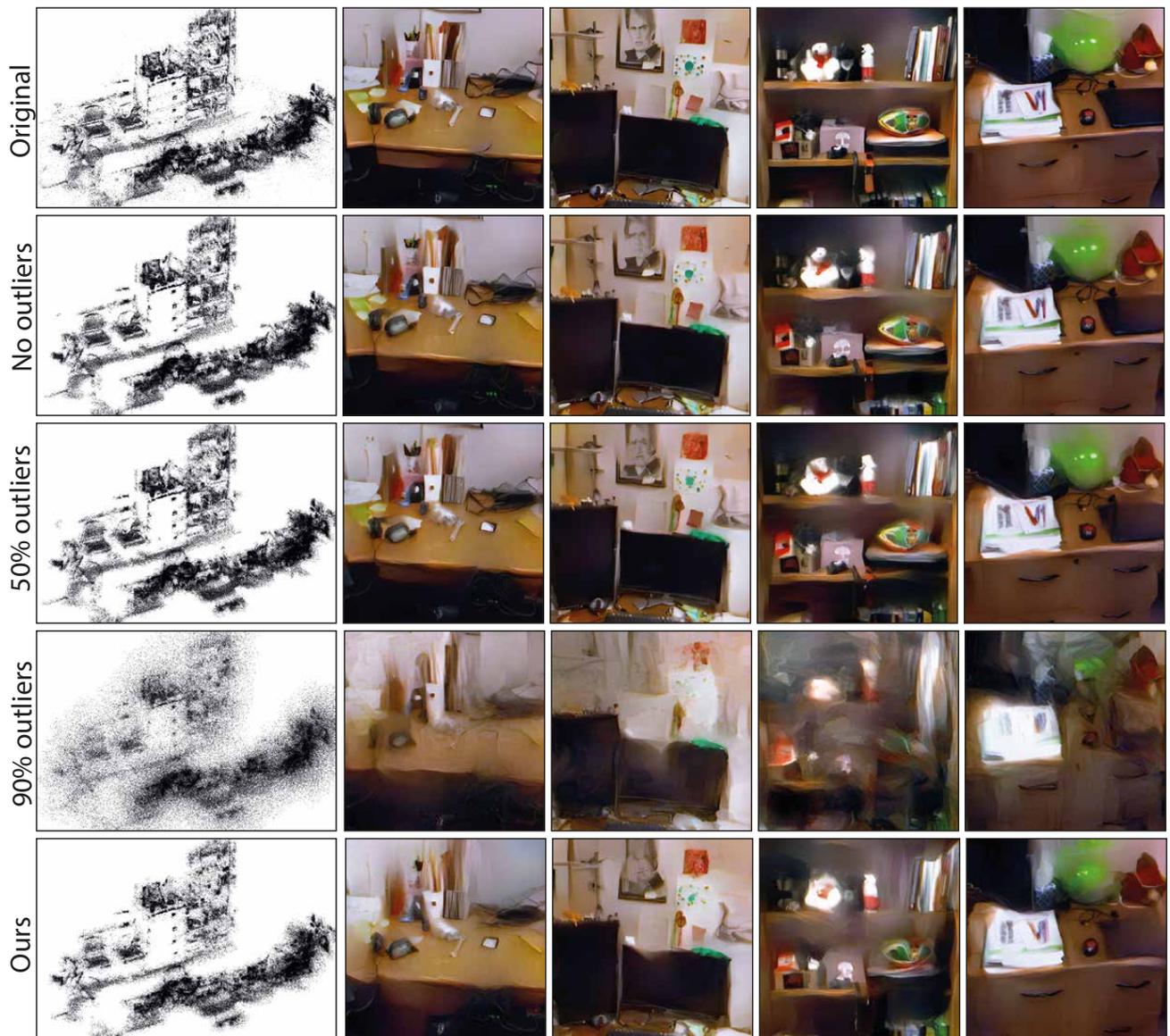


Figure 9. Qualitative results for the 'Office1-Manolis' scene from the 12 Scenes dataset [3]. We show point clouds as well as the images recovered from them using the approach from [2]. Besides the original point cloud and the one recovered by our method, we also show results obtained by using an oracle to provide the neighborhood of each point / line. For these neighborhoods, we vary the percentage of outliers contained in them.

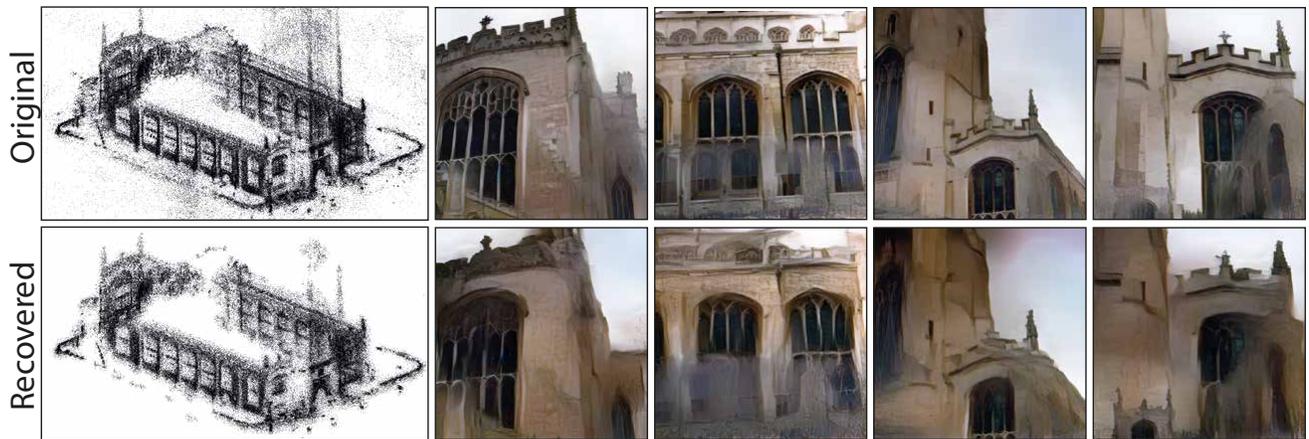


Figure 10. Qualitative results showing the recovered point cloud and the images obtained by applying the inversion method of [2] for the 'St Mary's Church' scene from the Cambridge dataset [1].



Figure 11. Qualitative results showing the recovered point cloud and the images obtained by applying the inversion method of [2] for the 'Shop Facade' scene from the Cambridge dataset [1].