PCAM: Product of Cross-Attention Matrices for Rigid Registration of Point Clouds — Supplementary Material —

Anh-Quan Cao[2,*] Gilles Puy[1] Alexandre Boulch[1] Renaud Marlet[1,3]

[1]Valeo.ai, Paris, France [2]Inria, Paris, France[†]
[3]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France

1. Network Architecture

1.1. Multiscale Attention Module $g(\cdot)$

Fig. 1 shows the detailed architecture of the multiscale attention module and of its L different encoders $e^{(\ell)} : \mathbb{R}^{\cdot \times c^{(\ell-1)}} \to \mathbb{R}^{\cdot \times c^{(\ell)}/2}$. We recall that $c^{(\ell)}$ denotes the number of channels at the output of the ℓ^{th} multiscale attention module. The encoder $e^{(\ell)}$ is made of three residual blocks, each yielding features with $c^{(\ell)}/2$ channels. The architecture of the residual block is also presented in Fig. 1 where the first 1D convolution is used only when the number of input and output channels differs. Each residual block consists of two FKAConv layers [2] with $c^{(\ell)}/2$ channels at input and output, a neighborhood of 32 nearest neighbours, a stride of size 1 (no downsampling of the point clouds), and a kernel of size 16. All convolutional layers are followed by instance normalization (IN) [7] with learned affine correction and a ReLU activation [6]. The number of channels $c^{(\ell)}$ used in our experiments in given in Table 1.

ℓ	0	1	2	3	4	5	6
$c^{(l)}$	3	32	32	64	64	128	128

Table 1. **Number of channels $c^{(\ell)}$ at the ouput of the ℓ^{th} block in the multiscale attention module**. $c^{(0)} = 3$ corresponds to the three coordinates x, y, z of the input point cloud.

1.2. Confidence Estimator $h(\cdot, \cdot)$

Fig. 1 shows the detailed architecture of the confidence estimator $h : \mathbb{R}^{\cdot \times 6} \to (0,1)^{\cdot \times 1}$. It consists of nine residual blocks, a FKAConv layer that reduces the number of channels to 1 and a final sigmoid activation. The FKAConv layer has a neighborhood of 32 nearest neighbours, a stride of size 1, and a kernel of size 16. The residual blocks have the same structure as the one used to construct the encoders in the multiscale attention module.

2. Experimental results

2.1. Indoor dataset: 3DMatch

A detailed analysis of the performance of our method for each of the 8 scenes in the test set of 3DMatch [11] is presented in Fig. 2. The results confirm the global scores presented in the main part of the paper. Each of the post-processing steps permits to improve the recall on all the scenes. DGR and PCAM-Sparse achieve nearly similar recall on all scenes with a very noticeable difference on Hotel3 where PCAM-Sparse with hard-thresholding on the weights outperforms DGR with all post-processing steps used.

We present in Fig. 3 and Fig. 4 examples of success and failure, respectively, of PCAM-Sparse with filtering with ϕ on 3DMatch [11]. We notice that the misregistrations are due to wrong mappings between points on planar surfaces in the examples of row 2, 3 in Fig. 4. We also remark in the example at row 5 that our method found matching points between points of similar but different object. Finally, row 1 of Fig. 4 shows a case where the estimated correspondences are correct but the confidence estimator considers them as unreliable which leads to a wrong estimation of the transformation (unless the safeguard is activated). This shows that the performance of the confidence estimator can still be improved.

2.2. Outdoor dataset: KITTI

We present in Fig. 5 and Fig. 6 examples of successful and failed registration results, respectively, with our method on the KITTI odometry dataset [4]. We observe in Fig. 5 that the static objects are well aligned after registrations. Concerning the failed registration results, it seems that this

*Most of the work was done during an internship at valeo.ai in 2020.
[†]Inria, Mines ParisTech, PSL Research University.

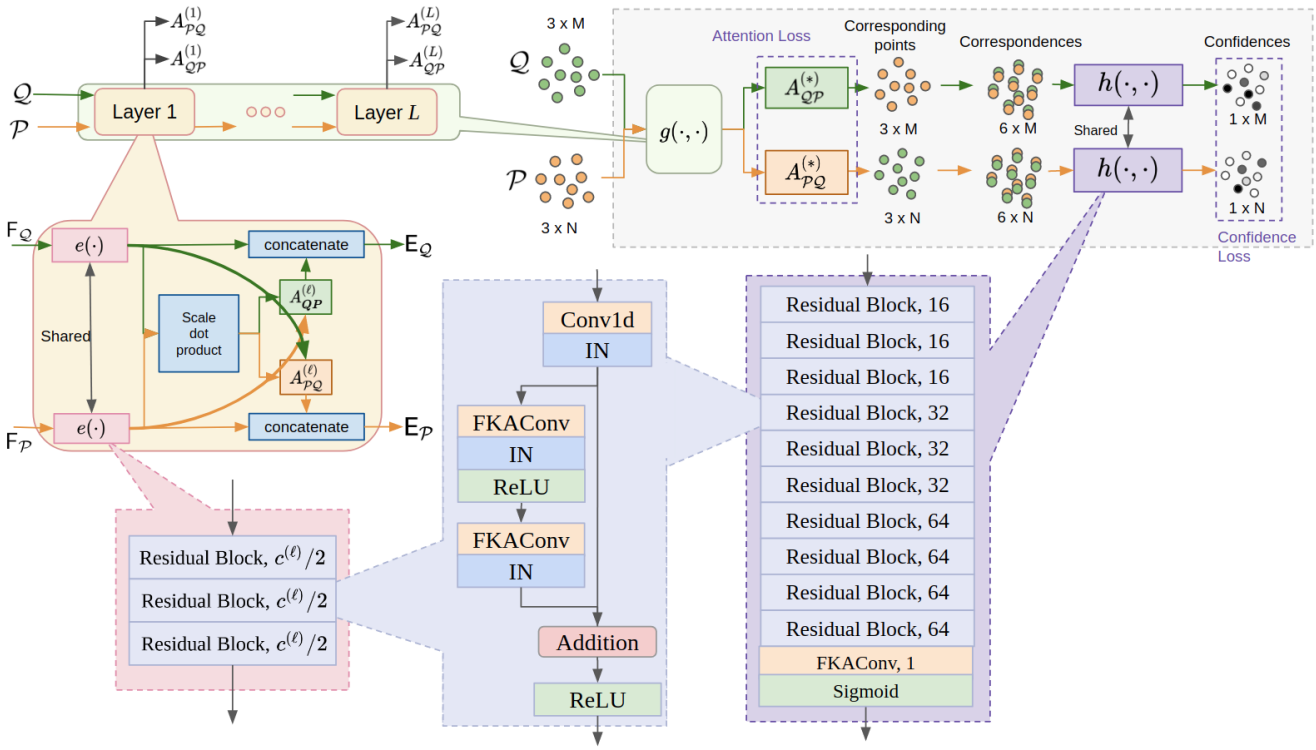

Figure 1. **Detailed network architecture of the multiscale attention module $g(\cdot)$ and the confidence estimator module $h(\cdot,\cdot)$.** Each module is constructed using the same type residual block, in which all internal layers have the same number of channels C indicated in the notation "Residual block, C". Note that the 1D convolution in the residual block is used *only* when the number of input and output channels differ.

is due to mapping between similar but different structure in the scene, at least for the scans with very large registration errors such as on row 1 or 4.

2.3. Synthetic dataset: ModelNet40

We provide in Table 2 the results obtained for ModelNet40 on the split corresponding to unseen objects, unseen categories, and unseen objects with noise [8]. Our method achieves better results than the recent methods that provided scores on these versions of the dataset.

RPM-Net [10] is also evaluated on ModelNet40, but using a slightly different setting than the one used by the methods in Table 2. We also test PCAM on this variant of ModelNet40. RPM-Net uses several passes/iterations to align two point clouds while PCAM, which is not trained on small displacements for refinement, uses only one. For fairness, we evaluate both methods after the first main pass. PCAM outperforms RPM-Net on the 'clean' version of ModelNet40 (Chamfer error of $1.8 \cdot 10^{-5}$ for RPM-Net, $3.4 \cdot 10^{-9}$ for PCAM) and on its 'noisy' version ($7.9 \cdot 10^{-4}$ for RPM-Net, $6.9 \cdot 10^{-4}$ for PCAM).

We present in Fig. 7 examples of registration results with our method for pairs of scans in the unseen objects with Gaussian noise split. The results presented in these figures illustrate the accurate registrations as well as the good quality of the matched pairs of points.

2.4. Sparse vs FKAConv convolutions

DGR uses sparse convolutions, whereas PCAM uses FKAConv point convolutions. To test if DGR suffers from a large disadvantage due to these sparse convolutions, we experiment replacing our point matching network by DGR's pre-trained point matching network. We then retrain our confidence estimator on KITTI using DGR's matched points. The performance of this new system reaches a recall of 94.6%, an $\mathrm{RE}_{\mathrm{all}}$ of 2.9 and $\mathrm{TE}_{\mathrm{all}}$ of 0.46 on KITTI validation set. This is on par with the results obtained with our point matching network, showing using sparse convolutions in the point matching network do not perform much worse than FKAConv.

References

[1] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7156–7165, 2019. 4

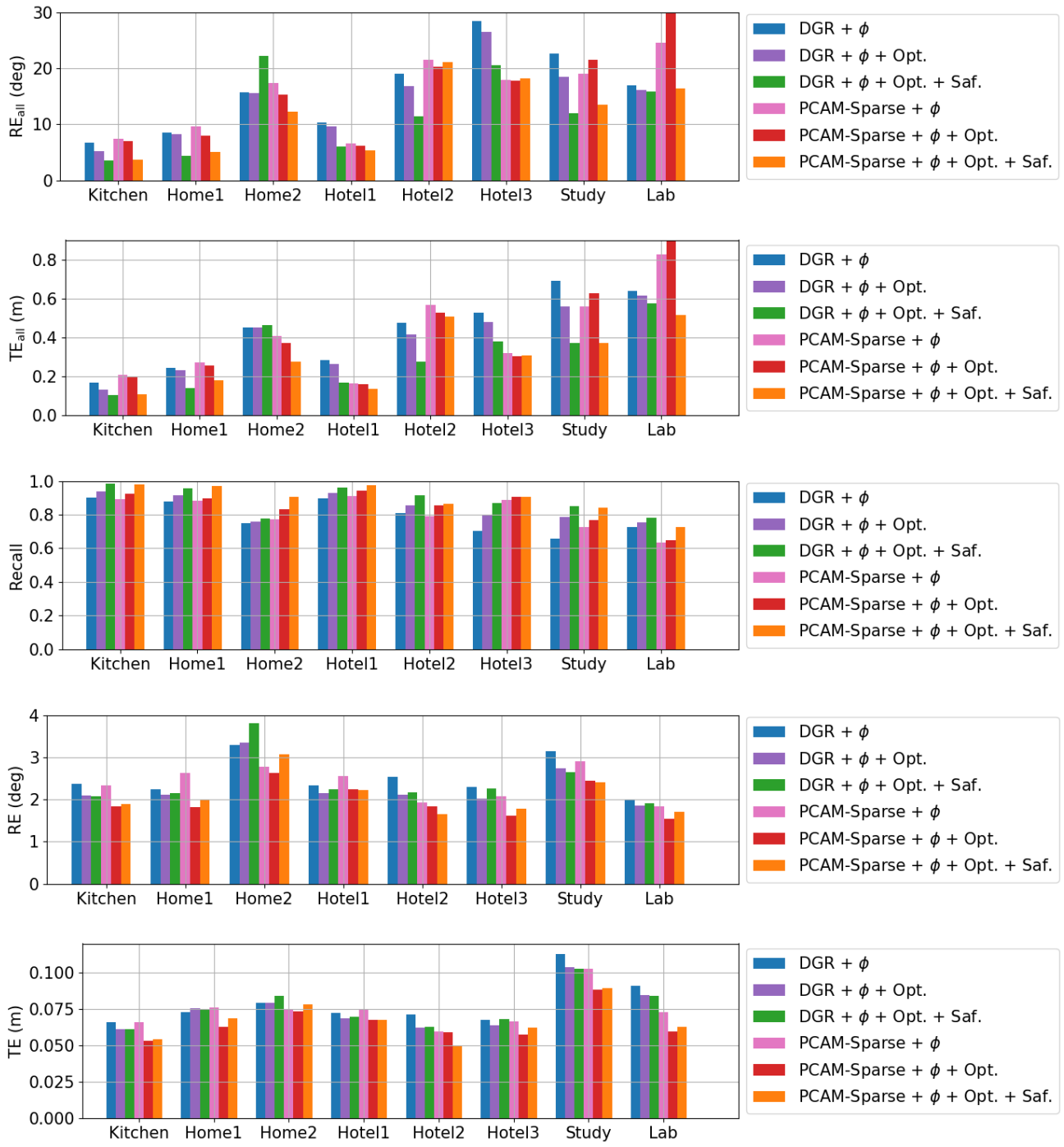

Figure 2. **Analysis of 3DMatch registration results per scene**. Row 1-2: Average TE and RE measured *on all pairs* (lower is better). Row 3: recall rate (higher is better). Row 4-5: TE and RE measured *on successfully registered pairs* (lower is better). Note that the recall of two methods have to be almost identical for their errors RE (resp. TE) to be comparable.

[2] A. Boulch, G. Puy, and R. Marlet. FKAConv: Feature-Kernel Alignment for Point Cloud Convolution. In *Asian Conference on Computer Vision (ACCV)*, 2020. 1

[3] Z. Dang, F. Wang, and M. Salzmann. Learning 3D-3D Correspondences for One-Shot Partial-to-partial Registration. *arXiv:2006.04523*, 2020. 4

	Unseen objects				Unseen categories				Unseen objects + noise			
Method	RMSE (R)	MAE (R)	RMSE ($\boldsymbol{t}$)	MAE ($\boldsymbol{t}$)	RMSE (R)	MAE (R)	RMSE ($\boldsymbol{t}$)	MAE ($\boldsymbol{t}$)	RMSE (R)	MAE (R)	RMSE ($\boldsymbol{t}$)	MAE ($\boldsymbol{t}$)
PointNetLK [1]	16.73	7.55	0.045	0.025	22.94	9.65	0.061	0.033	19.94	9.08	0.057	0.032
DCP-v2 [9]	6.71	4.45	0.027	0.020	9.77	6.95	0.034	0.025	6.88	4.53	0.028	0.021
PRNet [8]	3.20	1.45	0.016	0.010	4.98	2.33	0.021	0.015	4.32	2.05	0.017	0.012
IDAM [5]	2.46	0.56	0.016	0.003	3.04	0.61	0.019	0.004	3.72	1.85	0.023	0.011
OPRNet [3]	0.328	0.052	0.002	0.0003	0.357	0.069	0.002	0.0004	2.06	0.68	0.008	0.003
PCAM-Sparse	**0.023**	0.012	0.0002	0.00009	0.072	0.018	**0.0002**	**0.0001**	0.58	**0.15**	**0.002**	**0.001**
PCAM-Soft	**0.023**	**0.006**	**0.0001**	**0.00004**	**0.056**	**0.013**	0.0006	**0.0001**	**0.47**	**0.15**	**0.002**	**0.001**

Table 2. **Results on ModelNet40 for unseen objects, unseen categories and unseen objects with Gaussian noise.** The scores are reported from [8] for the first 3 methods, and from the associated papers for the others.

[4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 7, 8

[5] J. Li, C. Zhang, Z. Xu, H. Zhou, and C. Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *European Conference on Computer Vision (ECCV)*, 2020. 4

[6] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, page 807–814, Madison, WI, USA, 2010. Omnipress. 1

[7] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2017. 1

[8] Y. Wang and J. Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 8814–8826, 2019. 2, 4, 9

[9] Y. Wang and J. M. Solomon. Deep Closest Point: Learning Representations for Point Cloud Registration. In *International Conference on Computer Vision (ICCV)*, pages 3522–3531, 2019. 4

[10] Z. Yew and G. Lee. Rpm-net: Robust point matching using learned features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11821–11830, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 2

[11] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

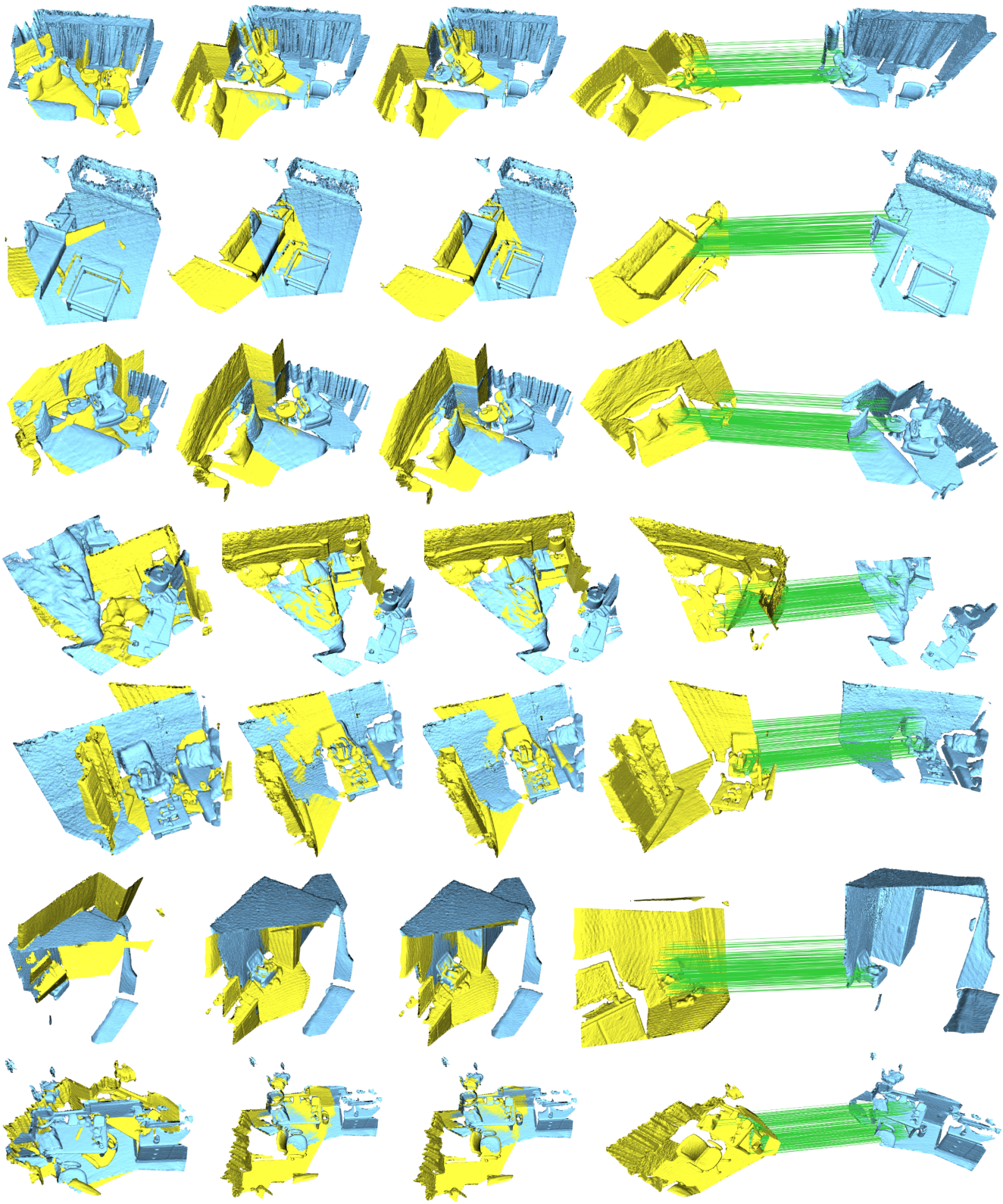

Figure 3. **Examples of successful registrations on 3DMatch with PCAM-Sparse +** ϕ. From left to right: overlaid, non-registered input scans (blue and yellow colors); ground-truth registration; scans registered with our method; and top 256 pairs of matched points with highest confidence.

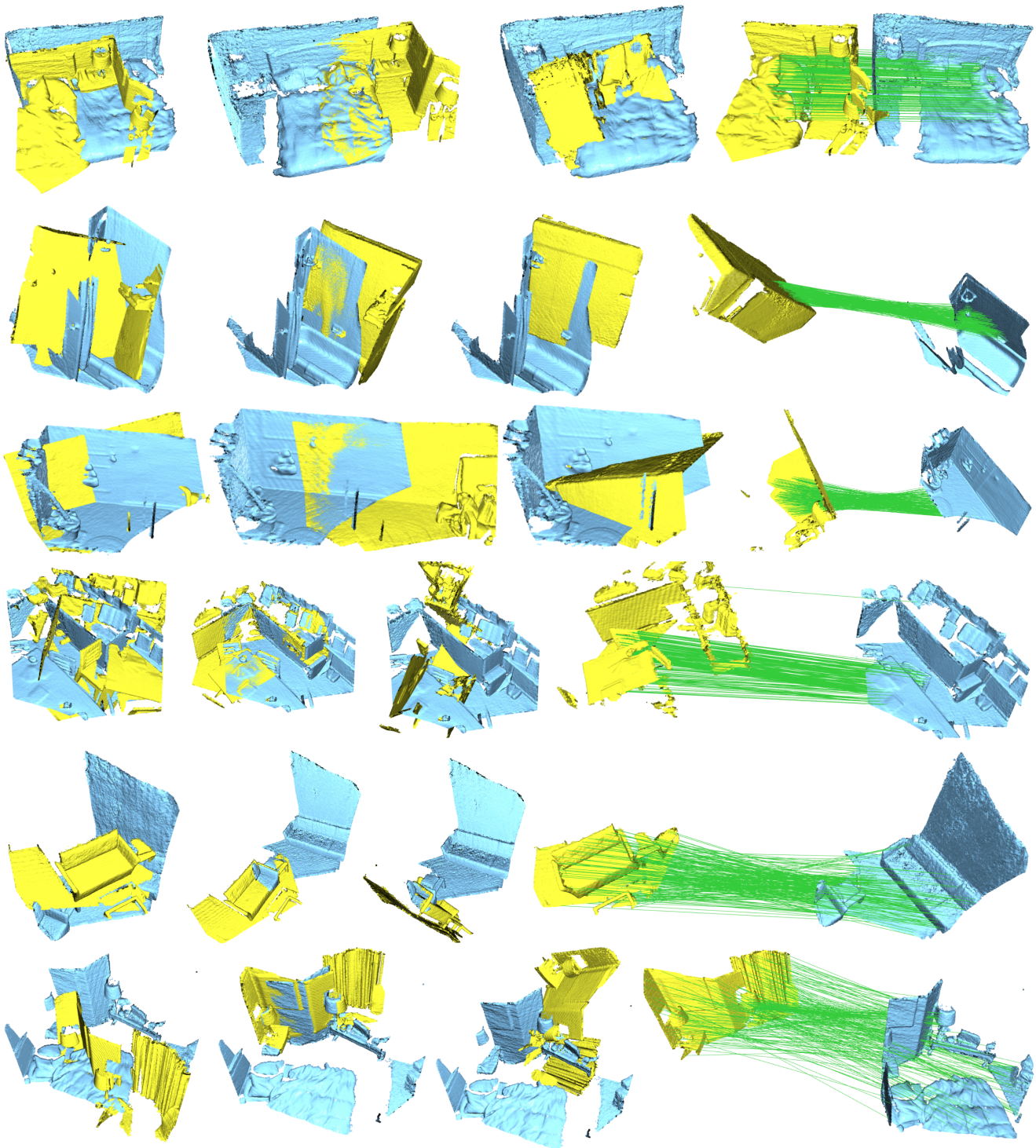

Figure 4. **Examples of failed registrations on 3DMatch with PCAM-Sparse +** ϕ. From left to right: overlaid, non-registered input scans (blue and yellow colors); ground-truth registration; scans registered with our method; and top 256 pairs of matched points with highest confidence.

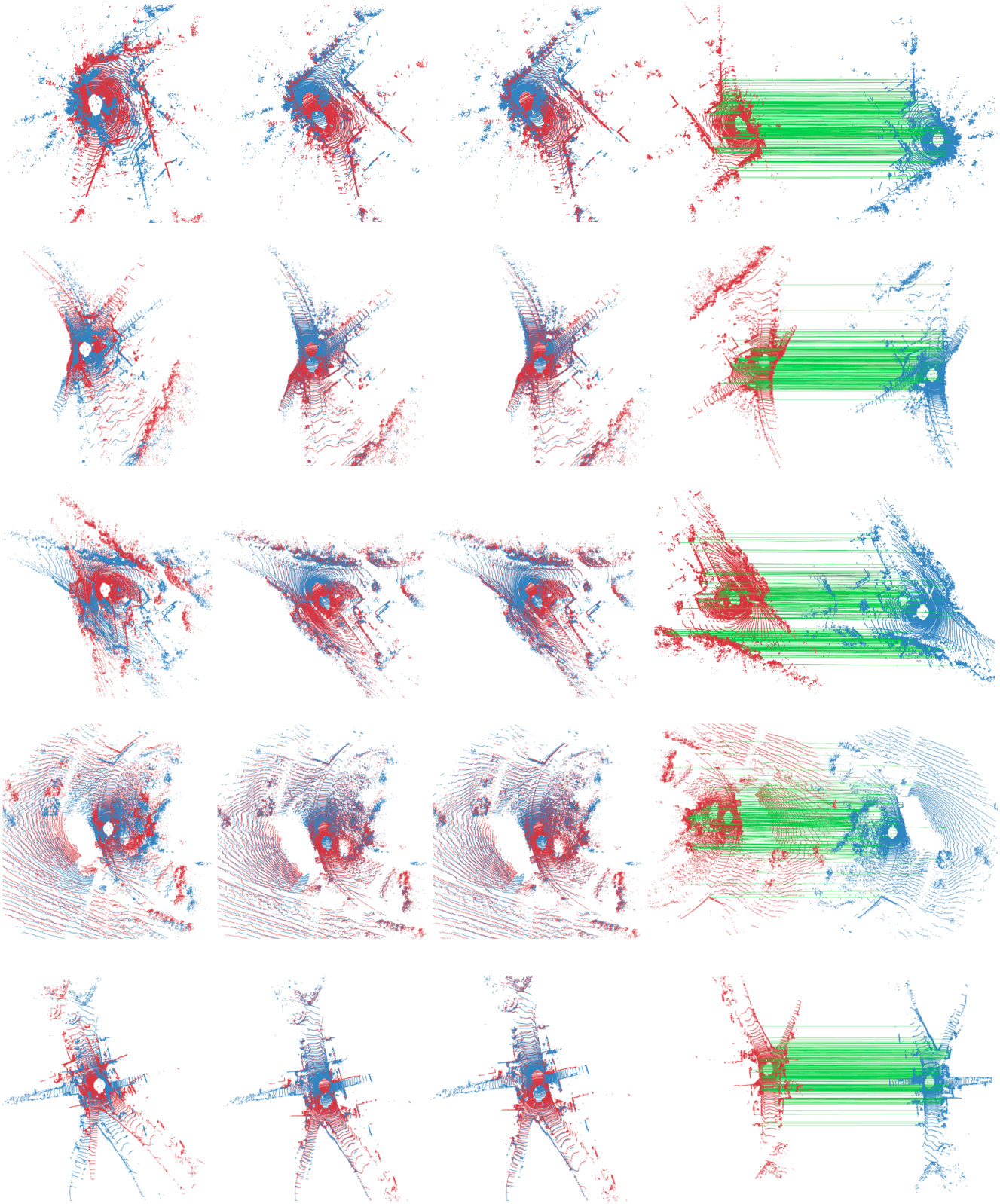

Figure 5. **Examples in bird-eye view of successful registrations on the KITTI odometry dataset [4]**. From left to right: overlaid, non-registered input scans (blue and red colors); ground-truth registration; scans registered with our method; and top 256 pairs of matched points with highest confidence. *Note that we have used the full non-voxelized Lidar scans for a better visualisation. However, all registrations are done using* 4096 *points drawn at random after voxelization of the full point cloud.*

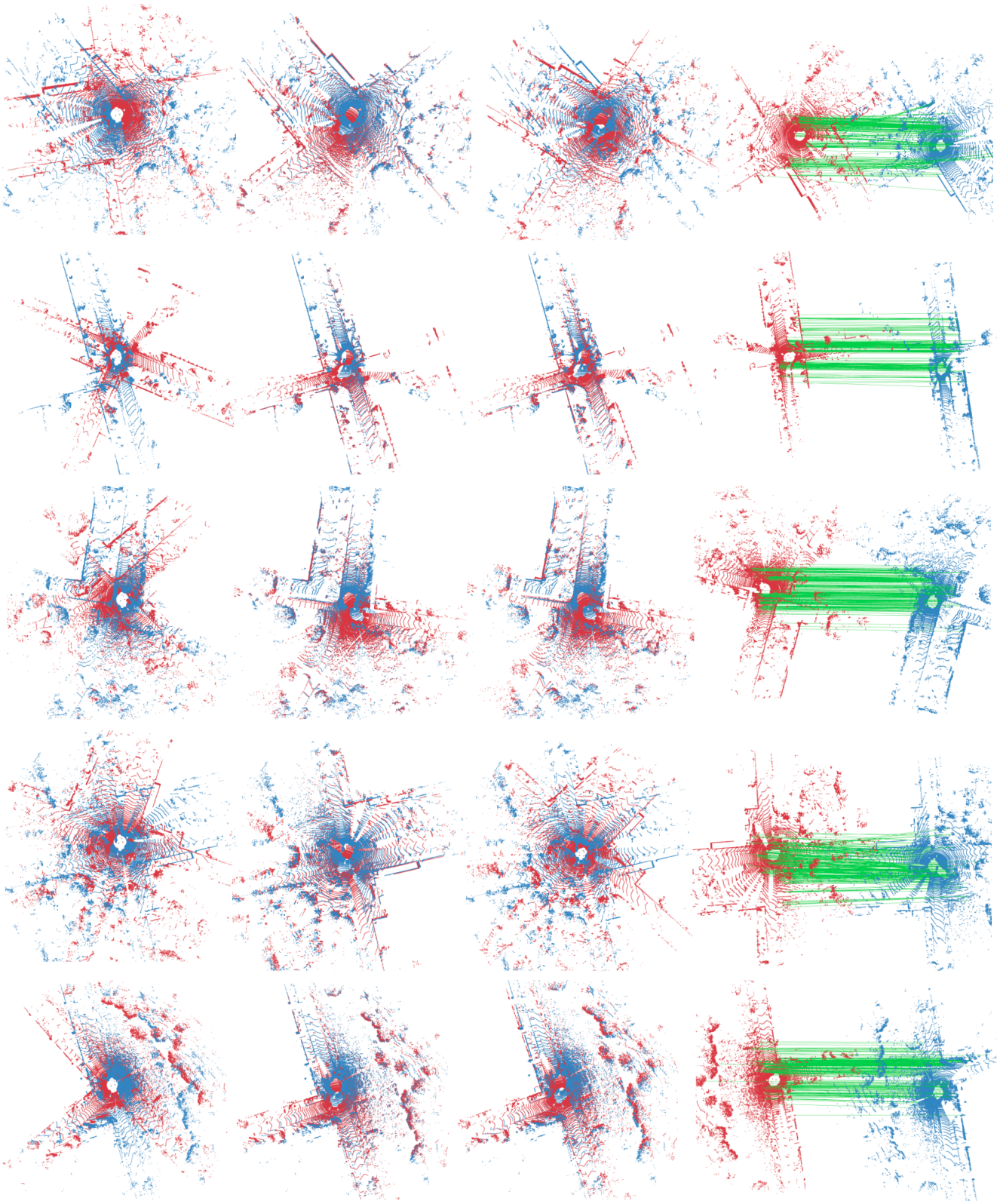

Figure 6. **Examples in bird-eye view of failed registrations on the KITTI odometry dataset [4].** From left to right: overlaid, non-registered input scans (blue and red colors); ground-truth registration; scans registered with our method; and top 256 pairs of matched points with highest confidence. *Note that we have used the full non-voxelized Lidar scans for a better visualisation. However, all registrations are done using* 4096 *points drawn at random after voxelization of the full point cloud.*

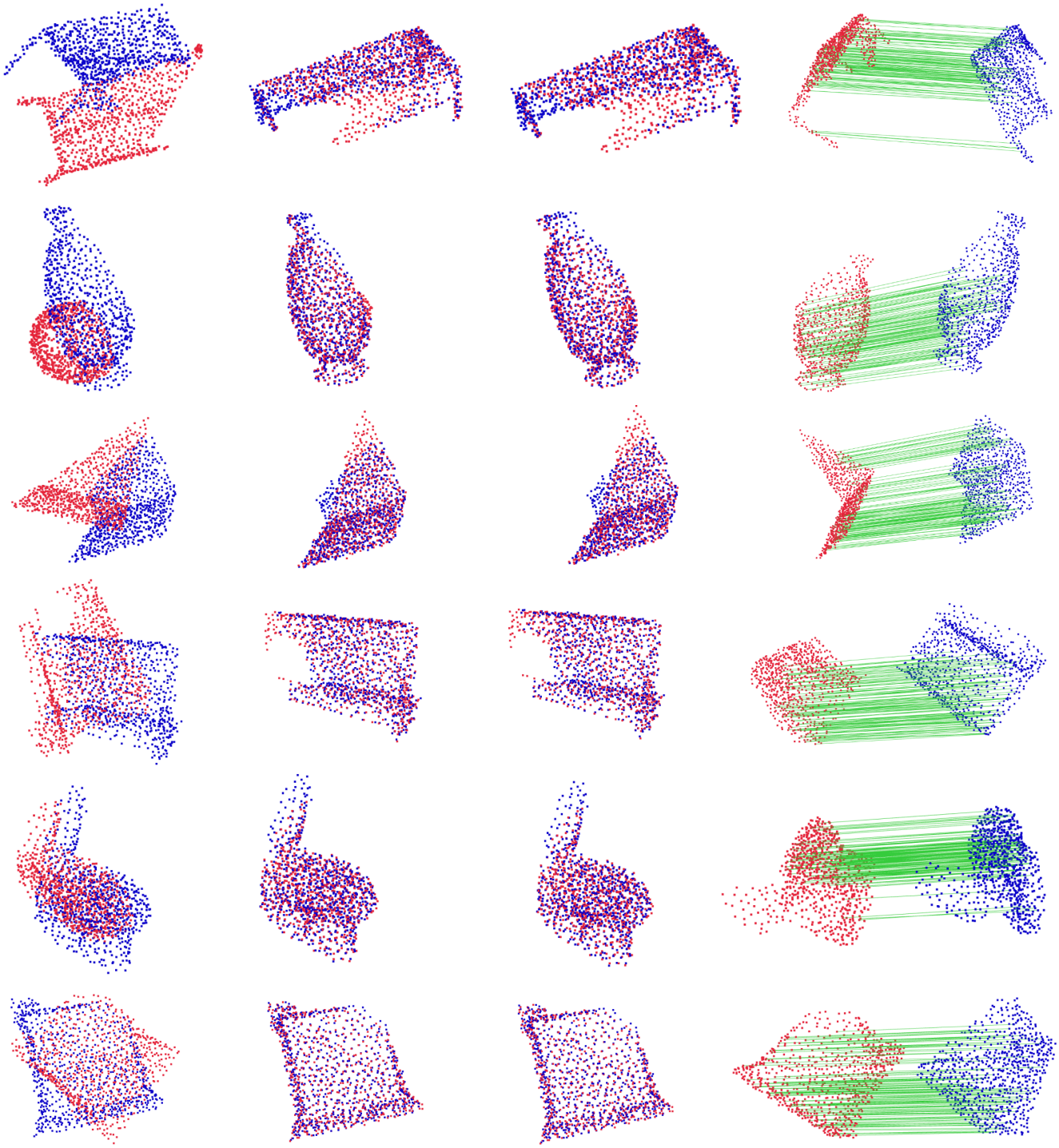

Figure 7. **Examples of registration results on ModelNet40 on the split with unseen objects, with Gaussian noise [8]**. From left to right: overlaid, non-registered input scans (blue and red colors); ground-truth registration; scans registered with our method; and top 128 pairs of matched points with highest confidence.