Supplementary Material for MagneticPillars: Efficient Point Cloud Registration through Hierarchized Birds-Eye-View Cell Correspondence Refinement

Kai Fischer[1,3] Martin Simon[1] Stefan Milz[2,3] Patrick Mäder[3]
[1]Valeo Schalter und Sensoren GmbH [2]Spleenlab GmbH [3]Ilmenau University of Technology
{kai.fischer, martin.simon}@valeo.com {stefan.milz, patrick.maeder}@tu-ilmenau.de

1. Implementation Details

Figure 1 shows the detailed processing pipeline of our proposed network. Here the pillar feature aggregation and the subsequent down sampling by the encoder are shared for both input clouds to generate a common coarse feature representation. The pillar generation module (*Pillar_Gen*) stacks the per voxel features produced by the feature encoder (*Feat_Enc*) in height, to generate the pillar shaped features for the BEV representation. Two self- and cross-attention layers are deployed to compute coarse cell feature correspondences S_c serving as an initial pre-filtering to reduce the overall computational effort. The shared decoder produces fine cell features per cloud which are filtered corresponding to the coarse pre-selection and finally matched via a single cross-attention layer to generate the fine correspondence scores S_f. The n_k most prominent matches are utilized to calculate the respective pillar centroids, used as input of an pose estimator predicting the final transformation.

1.1. Pillar Feature Aggregation

For our voxel grid generation, we used a grid size of $H = 352$, $W = 352$, $D = 32$ in the course of our experiments while cropping the input clouds at $-50m$ to $50m$ for x and y direction and $-4m$ to $2m$ for z. While a smaller grid size will result in a faster runtime, larger grids will yield finer pillar centroid points guaranteeing more accurate poses. We chose the mentioned size as a decent trade-off between the two criteria and its essential divisibility by 2^{n_l}. Moreover, we selected a maximum of $n_z = 20$ points per voxel, with n_v numbers of voxels per cloud, enabling a fixed structure data processing. Therefore the input dimensions for the Pillar- and Positional Encoder are 80 and 3, respectively, with an output feature dimension of $C = 16$ per voxel.

1.2. Encoder-Decoder Network

As stated in the main paper, we used a layer size of $n_l = 4$ with feature output dimensions of $[256, 512, 1024, 1024]$ for the encoder and $[512, 256, 128, 128]$ for the decoder, respectively. At the down scaled stage we filter $n_c = 30$ coarse cell correspondences which will result in n_cxn_f with $n_f = 256$ fine cell matching candidates per cloud.

1.3. Pose Estimation Backends

Regarding pose estimation using RANSAC, we applied the implementation featured in the Python library Open3D using *registration_ransac_based_on_feature_matching* for the feature-based methods and *registration_ransac_based_on_correspondence* for the correspondence estimation methods with 50000 iterations for both convergence criteria and identical parametrization for all methods. Furthermore, we use a distance threshold of 0.3 and the number of RANSAC correspondences of 4.

For pose estimation via weighted SVD, the matches and their respective confidences predicted by the correspondence-based methods are directly passed to the method as input clouds and weights. For the feature estimation methods, the output feature vectors are first multiplied to generate a similarity matrix to obtain the desired point correspondences. Subsequently, the score matrix is filtered via mutual selection, solely leaving matches with row and column-wise maxima.

1.4. Training

If not stated otherwise, we trained each version of our network for 100 epochs applying ADAM optimization with an initial learning rate of 0.0001 and a balancing factor of $\lambda = 4.0$ for $\mathcal{L} = \mathcal{L}_c + \lambda\mathcal{L}_f$. We used augmentation to the training data applying random shuffle, scale $\in \{0.8, \ldots, 1.2\}$, translation $\in \{0, \ldots, 1.0\}$, rotation $\in \{-180°, \ldots, 180°\}$ and random noise to the points with a factor of 0.01.

2. Metrics

2.1. Relative Translational and Rotational Error

For the estimated and ground truth transformations T_E and T_{GT} the respective translation vectors and rotation ma-

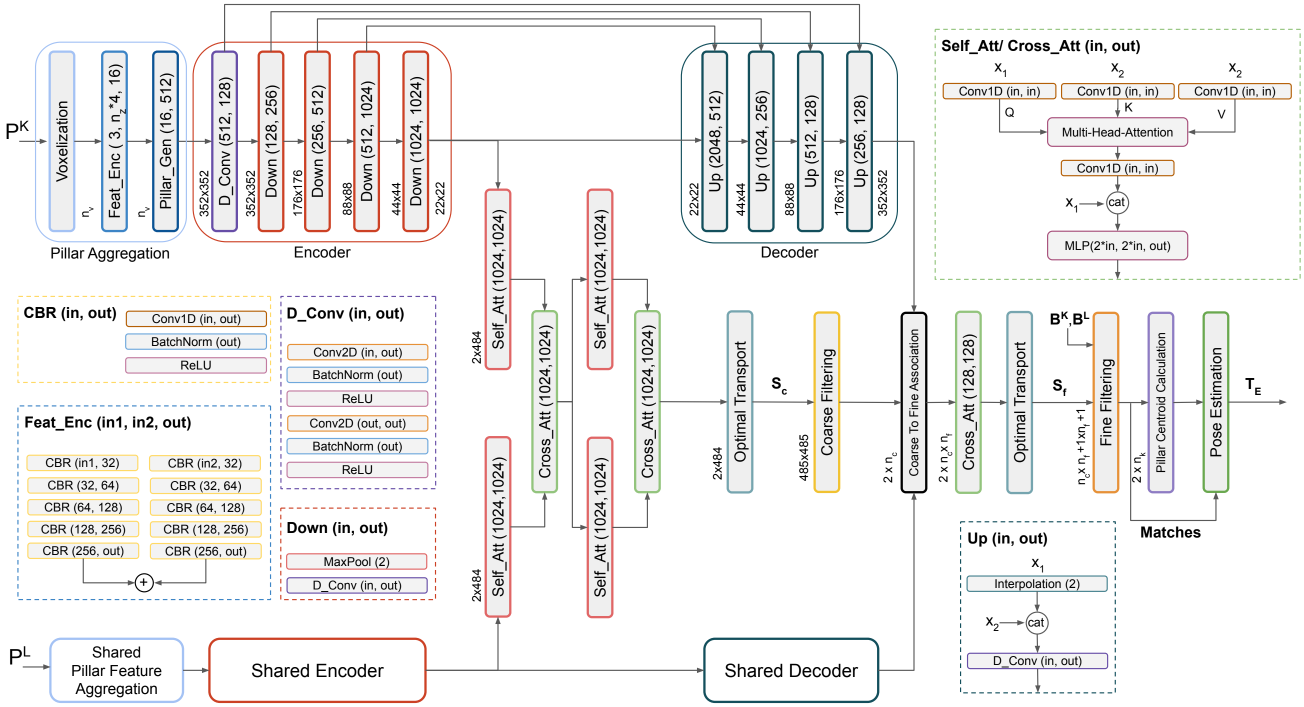

Figure 1. Detailed visualization of our MagneticPillars architecture with a shared 1) pillar feature aggregation and 2) down sampling, 3) coarse cell correspondence pre filtering, 4) fine feature generation by a shared decoder and 5) final fine cell selection based on the coarse initialization with subsequent 6) pose estimation on the most prominent matches.

trices are given as $t_E, t_{GT} \in \mathbb{R}^3$ and $R_E, R_{GT} \in SO(3)$. The RTE and RRE values are determined by:

$$\begin{aligned} RTE &= ||t_E - t_{GT}|| \\ RRE &= arccos\left(\frac{trace(R_E^T R_{GT}) - 1}{2}\right) \end{aligned} \tag{1}$$

2.2. Registration Recall

The Registration Recall (RR) measures the fraction of valid RTE and RRE values below the respective thresholds $0.6m$ and $5°$ to the total amount of frames M in the dataset.

$$RR = \frac{1}{M}\sum_{i}^{M} \mathbb{1}(RTE_i < 0.6m) \wedge \mathbb{1}(RRE_i < 5°) \tag{2}$$

where $\mathbb{1}$ denotes the indicator function.

2.3. Inlier Ratio

This value is determined by applying the ground truth transformation T_{GT} onto the extracted key points and measuring the distance between the putative point matches. Here correspondences with a value lower than a certain threshold $\tau_{ir} = 0.3m$ are considered inliers. Finally, the inlier ratio can be defined as the fraction of valid matches with respect to the total amount of predicted key points n_k according to:

$$IR = \frac{1}{n_k} \cdot \sum_{i=1}^{n_k} \mathbb{1}(||(T_{GT} \cdot \tilde{\pi}_i^L) - \tilde{\pi}_i^K|| < \tau_{ir}) \tag{3}$$

3. Additional Results

3.1. Comparison to Direct Methods

As mentioned in Section 2 of our main paper, direct point cloud registration methods aim to directly estimate the pose based on the raw input scans in an end-to-end manner. Therefore due to not explicitly deploying certain estimation backends and a fixed number of extracted key points or features, which was the main focus of our experiment section, we did not include any comparisons to those kinds of methods. However, we still want to contextualize the performance of MagneticPillars with respect to direct methods and performed experiments on the KITTI test dataset featuring recent state-of-the-art methods Deep Global Registration (DGR) [3], HRegNet [7], and PCAM [2]. We deployed our best-performing model parametrization for performance validation, extracting 5000 key points and applying LGR for pose estimation. The respective results are listed in Table 1.

Table 1. Registration performance of MagneticPillars compared to direct methods on the KITTI test dataset.

Method	n_k	Estimator	RR ↑ (%)	RTE ↓ (cm)	RRE ↓ (°)	Time ↓ (s)
DGR [3]	All	Direct	96.2	21.3	0.348	0.959
HRegNet [7]	All	Direct	95.7	**5.3**	0.992	**0.101**
PCAM [2]	All	Direct	97.1	7.8	0.417	0.137
Ours	5000	LGR	**99.5**	5.4	**0.297**	0.117

Note that we used the KITTI test set with the default data aggregation mentioned in Section 4.1, including the validation thresholds of $0.6m$ and $5°$ for RTE and RRE, respectively.

We are able to outperform all methods concerning RR with a value of 99.5%, where PCAM ranks second with 97.1% and RRE of $0.297°$, where DGR gains second lowest with $0.348°$. We furthermore reach the second lowest RTE ($5.4cm$) and runtime ($0.117s$), solely being surpassed by HRegNet, which, however, features a much lower RR impeding the comparison of RTE and RRE values. All in all, again MagneticPillars is able to reach state-of-the-art results when compared to recent point cloud registration approaches featuring a considerable trade-off between computational effort and pose estimation accuracy.

3.2. RANSAC Results for Varying Number of Key points

Analogous to our experiments in Section 4.2.1, we also conducted an evaluation of the registration performance under a varying number of key points using RANSAC as the pose estimation backend. Here we also included a comparison of the overall runtime of the specific methods since a different number of key points will highly impact the processing time of RANSAC. The corresponding results are listed in Table 2.

Here we are reaching the highest RR score for all keypoint selections, with a higher number generally benefiting the baseline methods, where RANSAC is able to counteract the impact of outlier predictions. In terms of RTE we are able to rank best in 8 out of 9 categories and second best for the remaining one. Geotransformer is generally showing the best accuracy in terms of RRE, with us reaching second lowest for 5 out of 9 selections and lowest for 10 extracted points. Regarding runtime evaluation, we are solely outperformed by FCGF, which however features a much worse estimation accuracy.

Table 2. Registration performance of the considered methods on the KITTI test dataset with varying number of extracted key points using RANSAC for pose estimation.

# Key points n_k	10	25	50	100	250	500	1000	2500	5000
	Registration Recall ↑ (%)								
FCGF [4]	0.4	0.9	6.3	19.8	60.7	82.2	91.2	93.5	93.0
D3Feat [1]	6.7	20.4	52.3	90.5	99.1	**99.5**	**99.5**	**99.5**	**99.5**
Predator [5]	0.7	3.6	16.8	51.2	93.0	98.7	99.3	**99.5**	**99.5**
CoFiNet [9]	66.1	90.5	96.8	98.6	**99.5**	**99.5**	**99.5**	**99.5**	**99.5**
RegTr [6]	63.8	76.9	80.4	82.9	85.2	85.6	85.6	86.1	86.5
GeoTransformer [8]	93.7	98.9	**99.5**	**99.5**	**99.5**	**99.5**	**99.5**	**99.5**	**99.5**
Ours	**96.4**	**99.1**	**99.5**	**99.5**	**99.5**	**99.5**	**99.5**	**99.5**	**99.5**
	Relative Translational Error ↓ (cm)								
FCGF [4]	33.5	39.4	40.7	38.0	31.6	23.6	18.7	15.6	15.0
D3Feat [1]	28.2	23.0	21.4	16.4	10.8	8.6	7.3	6.4	6.9
Predator [5]	47.4	34.9	36.7	31.9	21.7	13.5	9.7	6.7	**6.0**
CoFiNet [9]	23.4	18.9	15.4	12.1	9.9	9.0	8.2	7.8	7.8
RegTr [6]	33.8	29.1	27.1	24.7	23.6	22.9	22.7	22.8	22.8
GeoTransformer [8]	18.2	12.6	10.7	9.5	8.8	8.2	7.8	7.5	7.3
Ours	**14.8**	**10.9**	**9.2**	**8.1**	**7.0**	**6.4**	**6.1**	**6.0**	6.4
	Relative Rotational Error ↓ (°)								
FCGF [4]	1.642	2.962	2.491	2.405	1.832	1.140	0.685	0.456	0.389
D3Feat [1]	1.698	1.736	1.367	0.896	0.523	0.406	0.342	0.306	0.313
Predator [5]	1.640	2.854	2.681	2.198	1.248	0.664	0.435	0.326	**0.275**
CoFiNet [9]	1.453	1.031	0.803	0.605	0.476	0.429	0.351	0.360	0.360
RegTr [6]	1.342	0.855	0.682	0.574	0.496	0.445	0.424	0.408	0.419
GeoTransformer [8]	1.057	**0.599**	**0.461**	**0.387**	**0.333**	**0.299**	**0.289**	**0.286**	0.289
Ours	**1.003**	0.727	0.562	0.476	0.417	0.355	0.345	0.319	0.299
	Total Runtime ↓ (s)								
FCGF [4]	**0.116**	**0.113**	**0.111**	**0.115**	**0.113**	**0.113**	**0.118**	**0.122**	**0.132**
D3Feat [1]	0.225	0.232	0.228	0.228	0.226	0.238	0.236	0.241	0.264
Predator [5]	0.194	0.194	0.195	0.195	0.197	0.196	0.200	0.221	0.273
CoFiNet [9]	0.390	0.390	0.390	0.390	0.391	0.392	0.393	0.398	0.402
RegTr [6]	0.488	0.488	0.489	0.490	0.491	0.493	0.496	0.502	0.503
GeoTransformer [8]	0.264	0.264	0.265	0.266	0.267	0.271	0.275	0.282	0.287
Ours	0.128	0.129	0.130	0.133	0.139	0.151	0.164	0.187	0.194

Overall, RANSAC benefits from a higher number of key points resulting in a more accurate pose prediction, which however will lead to a tremendous increase in computation time. This again shows the general benefit of applying SVD as a registration backend which features a constant low runtime unaffected by varying input points but relies on an accurate correspondence initialization.

3.3. Inlier Ratio Comparison

A convenient indicator for the correspondence prediction capability is the Inlier Ratio (IR) already featured in the qualitative visualizations of our main paper. The IR values for selected methods and varying n_k are listed in Table 3. We are reaching the highest values for all key point selections compared to the other methods by a large margin with a maximum of 76.1% for 25 extracted points and more than double the value compared to the second-ranked approach for 5000 correspondences.

The robustness of our keypoint detection is moreover displayed in Figure 2 where we visualized the extracted points with varying n_k for 4 selected frames of the KITTI test dataset. Due to our fixed grid representation with

Table 3. Inlier Ratio of the considered methods on the KITTI test dataset with varying number of extracted key points for an inlier threshold of $\tau_{ir} = 0.3m$.

# Key points n_k	10	25	50	100	250	500	1000	2500	5000
	Inlier Ratio ↑ (%)								
Predator [5]	0.3	0.6	1.3	2.4	5.2	8.6	13.1	19.7	23.2
CoFiNet [9]	14.9	14.5	14.4	14.5	14.4	14.5	14.4	13.9	13.3
RegTr [6]	36.2	36.0	35.5	35.1	33.7	32.7	30.4	25.0	24.4
GeoTransformer [8]	46.1	45.5	44.0	42.6	40.3	37.8	34.4	28.4	22.5
Ours	**75.7**	**76.1**	**75.5**	**74.2**	**71.1**	**67.6**	**60.9**	**51.0**	**50.7**

the proposed coarse to fine cell filtering, we are able to extract the relevant overlapping information between two input clouds. As shown in the *All Fine* column, the full fine feature candidates based on the coarse cell upsampling already represent an appropriate match candidate pre-filtering, extracting related structures from the two input clouds. Based on the fine score matrix S_f resulting from the final cross-attention module, we are able to extract the n_k most confident match predictions visualized in the subsequent columns for $n_k \in \{10, 250, 1000\}$. In this context, our keypoint selection is able to maintain a shared structural consistency of the input clouds even with increasing sparsity of the matching candidates.

3.4. Qualitative Results

Finally, we want to demonstrate the pose estimation capability of MagneticPillars on a visual level by including additional qualitative results on the KITTI and Nuscenes datasets shown in Figure 3 and 4 respectively. Here we are able to perform an accurate point cloud registration even for the sparser clouds captured within the Nuscenes dataset, resulting in an estimated pose close to the ground truth transformation.

References

[1] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6359–6367, 2020.

[2] Anh-Quan Cao, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Pcam: Product of cross-attention matrices for rigid registration of point clouds. In *IEEE International Conference on Computer Vision (ICCV)*, pages 13229–13238, 2021.

[3] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2514–2523, 2020.

[4] Christopher Choy, Jaesik Park, and Postech Vladlen Koltun. Fully convolutional geometric features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8958–8966, 2019.

[5] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, Konrad Schindler, and Eth Zurich. Predator: Registration of 3d point clouds with low overlap. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4267–4276, 2021.

[6] Zi Jian, Yew Gim, and Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6677–6686, 2022.

[7] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 16014–16023, 2021.

[8] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11143–11152, 2022.

[9] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust point cloud registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23872–23884, 2021.

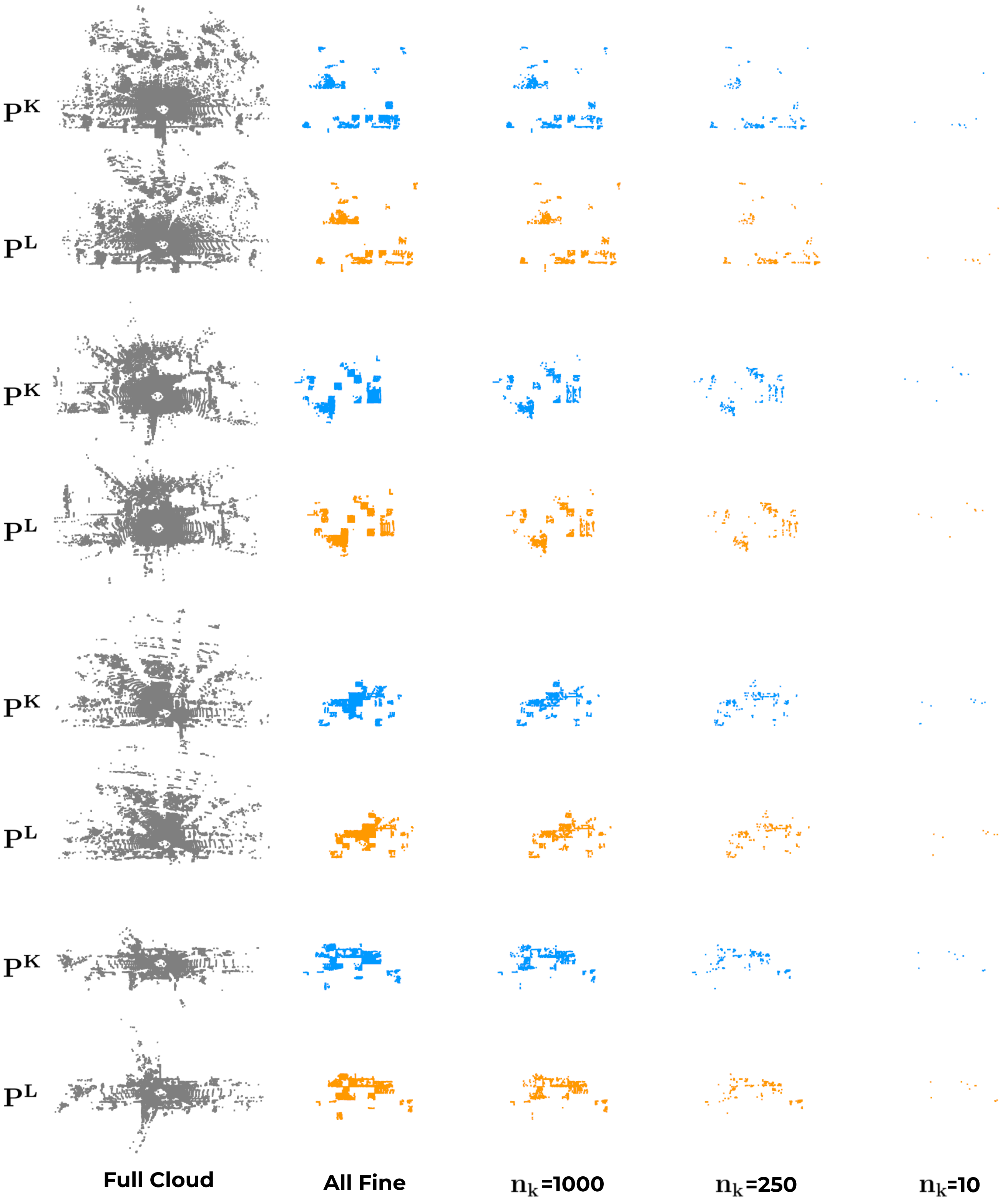

Figure 2. Visualization of the extracted key points for 4 selected frames of the KITTI dataset. Here the first column shows the two input clouds P^K and P^L, the second all fine feature points π^K and π^L extracted from the coarse correspondence cells ($n_c = 30$) and the subsequent entries the respective top 1000, 250 and 10 fine feature matches based on the predicted score matrix S_f. For a varying number of selected key points MagneticPillars is able to constantly extract common structural characteristics in between the two input clouds.

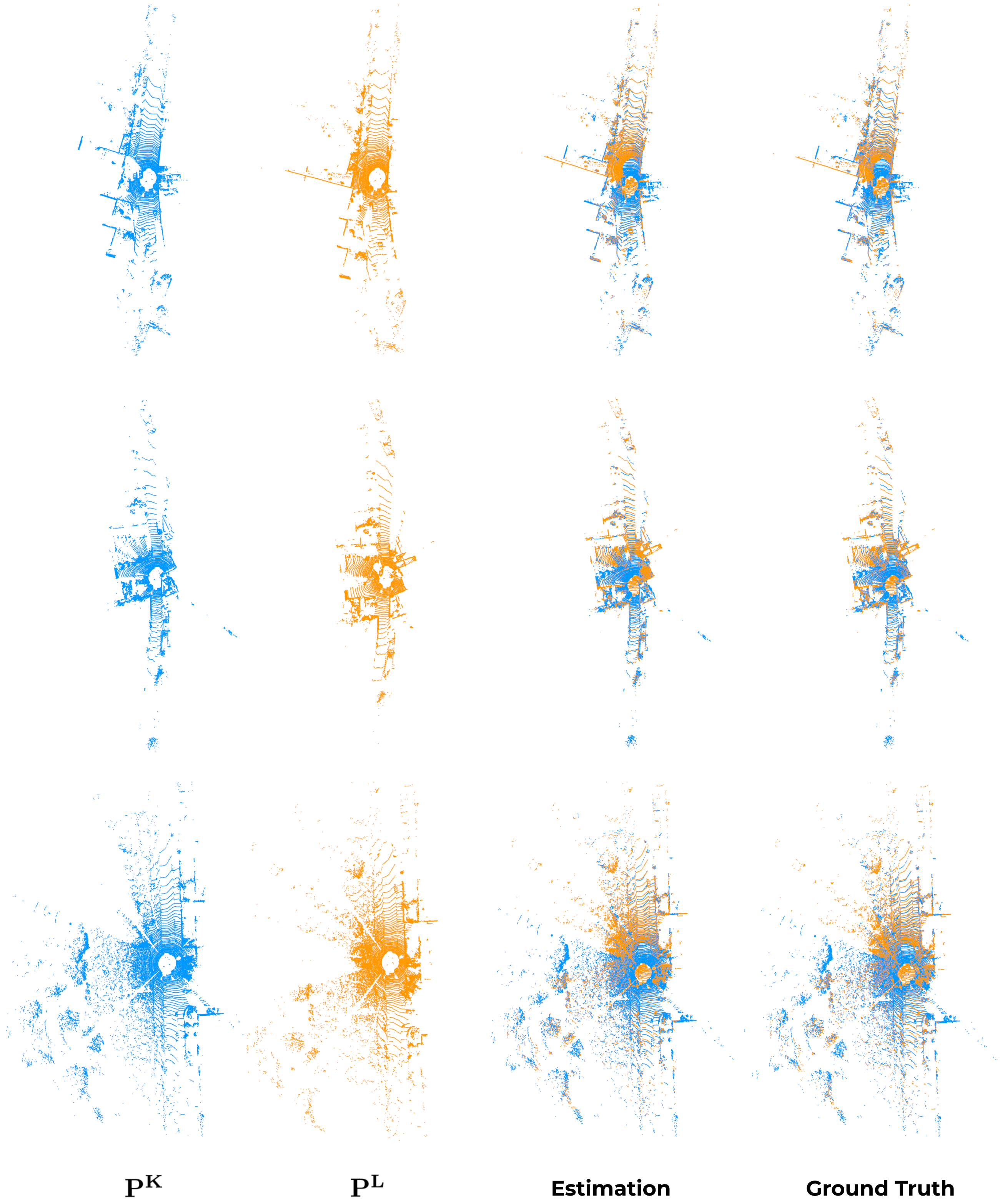

Figure 3. Qualitative registration results on the KITTI dataset for 3 selected frames featuring input clouds P^K and P^L with applied estimated T_E and ground truth transformation T_{GT}.

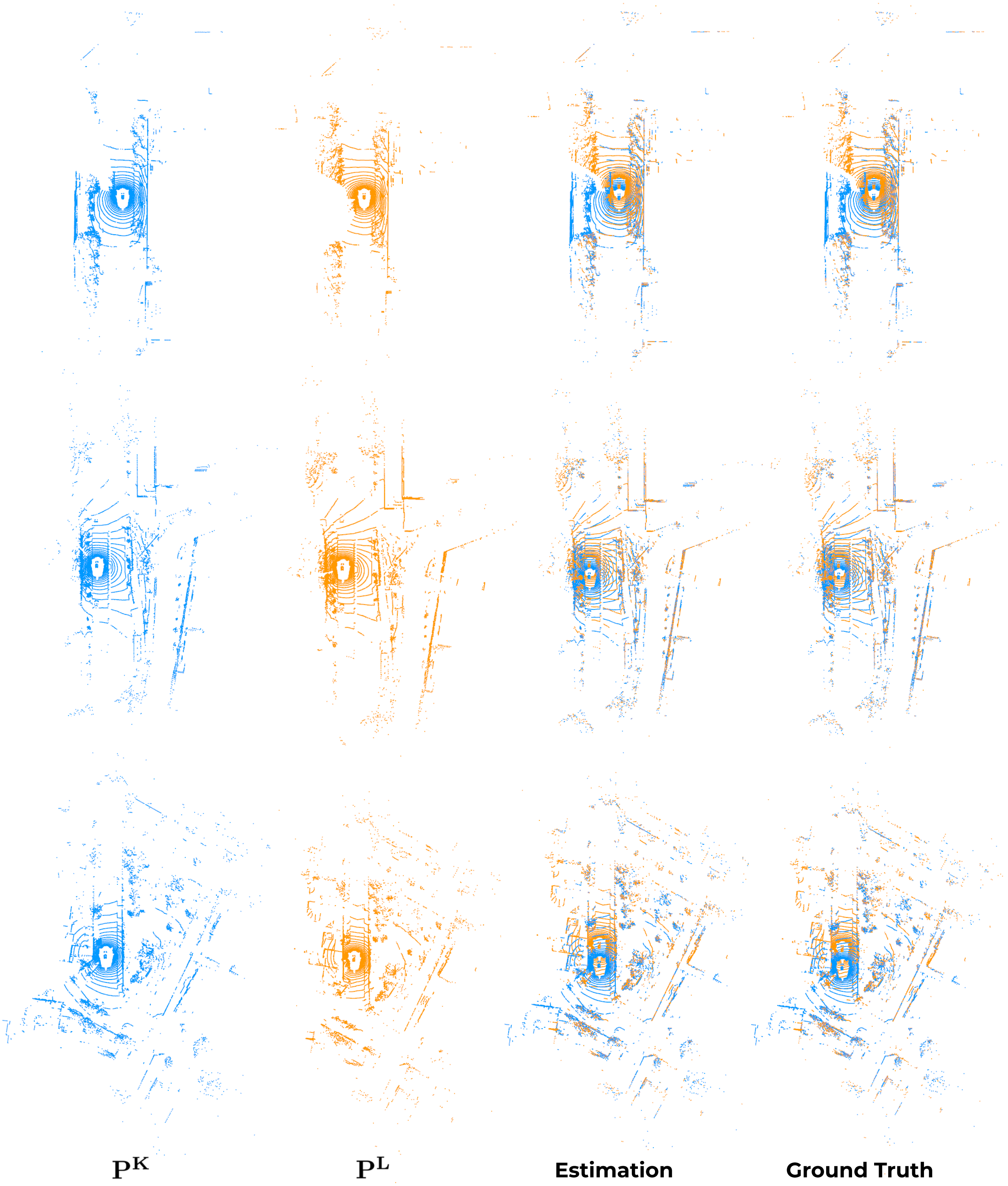

Figure 4. Qualitative registration results on the Nuscenes dataset for 3 selected frames featuring input clouds P^K and P^L with applied estimated T_E and ground truth transformation T_{GT}.