CMT: Context-Matching-Guided Transformer for 3D Tracking in Point Clouds (Supplementary Material)

Zhiyang Guo[1], Yunyao Mao[1], Wengang Zhou[1,2⋆],
Min Wang[2], and Houqiang Li[1,2⋆]

[1] CAS Key Laboratory of Technology in GIPAS, EEIS Department,
University of Science and Technology of China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{guozhiyang, myy2016}@mail.ustc.edu.cn, zhwg@ustc.edu.cn,
wangmin@iai.ustc.edu.cn, lihq@ustc.edu.cn

Abstract. This document supplements our paper *CMT: Context-Matching-Guided Transformer for 3D Tracking in Point Clouds* by providing the results of some additional experiments, as well as the details of some alternative context matching strategies.

1 Complexity Evaluation

The inference speed measured by FPS and the total number of trainable model parameters are listed in Table 1. All the results are produced with one NVIDIA GTX 1080Ti GPU.

Table 1. Comparison on the computational complexity.

	P2B [6]	BAT [11]	CMT (Ours)
FPS	51	45	32
Params (M)	1.2	1.5	2.5

2 Additional Experiments

We explore some variants of the target-specific transformer, the context matching, and the orientation encoder (OE) in the proposed CMT tracker. All the experiments are conducted on the truck class of nuScenes[2].

⋆ Corresponding authors: Wengang Zhou and Houqiang Li.

2.1 Vector versus Scalar Attention

In [9], different transformations to construct the attention weights are explored, including the commonly-used scalar dot-product and the vector subtraction we adopt in CMT. It is claimed in [10] that vector attention is more suitable for point clouds, since it supports adaptive modulation of individual feature channels, rather than a shared scalar weight. To validate the effectiveness of the vector transformer for matching-guided feature fusion in our model, we compare these two kinds of attention mechanisms. Specifically, for scalar dot-product transformer, the attentional target-specific feature $\widehat{F}_a$ is expressed as:

$$A = \mathrm{Softmax}(Q^T K)\,, \tag{1}$$

$$\widehat{F}_a = \mathrm{MLP}(\mathrm{Norm}(A^T(V + PE) + F_s))\,, \tag{2}$$

where Q, K, and V are the Query, Key and Value embeddings, respectively; F_s is the input search feature; PE denotes the trainable relative positional encoding, which is consistent with that in vector transformer.

For comparison, we also provide the performance of our CMT pipeline without the transformer, which simply uses MLPs to fuse features. As shown in Table 2, although being a little bit slower, the vector subtraction attention exhibits significant advantage over the scalar dot-product, thanks to the improved fitting ability brought by larger-scale learnable parameters.

Table 2. Comparison between different types of attention mechanism in transformer for target-aware feature fusion. Bold denotes the best performance.

Attention Type for Transformer	Success	Precision	FPS
without Transformer	47.8	48.3	**42**
Scalar Dot-Product	49.1	50.3	36
Vector Subtraction (CMT)	**52.5**	**52.5**	32

2.2 Transformer versus OE for Spatial Awareness

The target-specific transformer in the proposed CMT tracker focuses on template-search (target-specific) feature fusion using cross-attention. As for further spatial-aware fusion among different points, both the self-attention operation (transformer encoder) and the orientation encoder (OE) introduced in CMT are competent. To this end, we replace the OE with another transformer encoder for spatial awareness, which also adopts vector attention. Furthermore, following the general encoder-decoder structure of a transformer, we try an alternative order of feature fusion, which exchanges the positions of target-specific and spatial-aware feature fusion modules. The spatial-aware fusion before the target-specific transformer is performed on template and search feature, respectively, with a shared network (OE or self-attention).

As listed in Table 3, OE shows notable superiority over the self-attention operation, which implies that neighbors from all eight octants are more helpful than homogeneous globally nearby points from only a few directions for spatial awareness. Meanwhile, the computational overhead of OE is much lower, resulting in a real-time level inference speed. Moreover, exchanging the order of template-search fusion and spatial-aware feature fusion leads to significant performance degradation. A possible explanation is that, the target-specific information is mixed up by spatial-aware fusion, which confuses the following transformer.

Table 3. Comparison between OE and transformer for spatial-aware feature fusion. The two method names between the dash are for template-search fusion and spatial-aware fusion, respectively. An alternative feature fusion order is presented for both two choices. S-Attn denotes the self-attention operation, and X-Attn denotes the cross-attention operation. Bold denotes the best performance.

Feature Fusion Method	Success	Precision	FPS
S-Attn - X-Attn	49.7	50.2	21
X-Attn - S-Attn	51.5	51.1	24
OE - X-Attn	50.6	50.9	31
X-Attn - OE (CMT)	**52.5**	**52.5**	**32**

Table 4. Comparison between different contextual descriptors. Bold denotes the best performance.

Elements of Contextual Descriptor	Success	Precision
3D Coordinates	49.6	50.5
Seed Feature	50.1	50.6
Polar Vector (CMT)	**52.5**	**52.5**

2.3 Effectiveness of Local Contextual Descriptor

In the proposed CMT tracker, we introduce the spatial contextual encoder to produce some horizontally rotation-invariant local contextual descriptors directly from the point clouds. The motivation behind such descriptors mainly lies in the following aspects: 1) 3D SOT is mainly targeted at driving scenarios, where keeping track of a vehicle with possible horizontal rotations is a practical problem. 2) In most 3D trackers, the feature extractor is jointly optimized with subsequent modules from scratch. Thus it is hard to learn a rotation-invariant feature for modern trackers with a long pipeline supervised only by the final prediction of boxes. Data augmentation (randomly rotate some samples) used by other 3D tasks is not directly applicable to SOT due to the consistency of a tracking sequence. 3) The idea of context matching also calls for such rotation-invariant descriptors.

To show the necessity of the proposed local contextual descriptor, we replace the polar vectors in the descriptor with 3D coordinates or the seed feature of points. The results listed in Table 4 show that context matching with some rotation-variant descriptors leads to significant performance degradation.

2.4 Multi-Level OE-Conv

In the orientation encoder (OE) of our CMT tracker, multiple orientation-encoding convolution (OE-Conv) blocks are simply stacked for a more extensive spatial awareness. However, in PointSIFT [4], the input feature is processed through a series of OE-Conv blocks, where multi-level features are obtained and integrated as illustrated in Fig. 1.

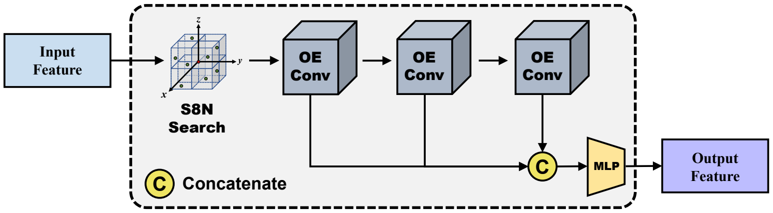

Fig. 1. Illustration of the multi-level version of OE. Stacked 8-neighborhood (S8N) search is adopted at the first place to select neighbors for each point.

We compare the performance of these two designs in Table 5. It can be observed that, in our work, no significant performance improvement is achieved by integrating multi-level features, which implies that the highest-level feature possesses sufficient spatial information for the target proposal.

Table 5. Comparison between different structures of OE convolution. Bold denotes the best performance.

OE-Conv Structure	Success	Precision	FPS
Multi-Level	52.1	52.4	31
Highest-level (CMT)	**52.5**	**52.5**	**32**

2.5 Context Matching Strategies

We compare three other context matching strategies apart from the shifted-window matching adopted in our CMT tracker: linear assignment (LA) with

Hungarian algorithm [5], the inexact proximal point method for optimal transport (IPOT) [8] and the shifted-minimum matching. The details of all these strategies are further interpreted in Section 3. We compare their performance with the same number of neighboring points ($k_1 = k_c = 16$).

As shown in Table 6, LA and IPOT show similar performance, which makes sense since optimal transport here can be viewed as a continuous and approximate version of linear assignment. Note that the inference speed of LA is unacceptably slow, and IPOT is almost three times faster. With additional prior knowledge of partially sequential injective correspondences, the simple shifted-minimum matching shows notable improvement. Shifted-window matching used in the proposed CMT tracker exhibits the best performance and runs with 32 FPS when inferring.

Table 6. Comparison between different context matching strategies. Bold denotes the best performance.

Context Matching Strategy	Success	Precision	FPS
LA	51.4	50.9	11.3
IPOT	51.3	51.0	30.2
Shifted-Minimum	52.0	51.7	**34.5**
Shifted-Window (CMT)	**52.5**	**52.5**	32

2.6 Robustness to Sparsity

One of the major challenges faced by point-cloud-based trackers is that the object points collected by LiDAR sensors are mostly sparse and incomplete. Therefore, the robustness against sparsity is an indispensable property for 3D trackers. To this end, we evaluate the average success rate on the sequences in KITTI-Car split by the number of points in the first frame's car. Fig. 2 shows that CMT holds a significant advance over BAT [11], especially for sparse point clouds.

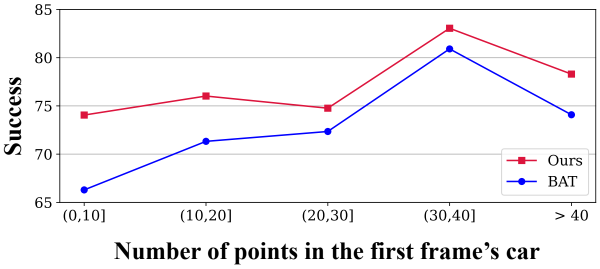

Fig. 2. Performance comparison on sequences with different sparsity.

3 Details of Alternative Context Matching Strategies

In the context matching stage of our CMT tracker, a novel shifted-window matching strategy is designed, which exhibits better performance than other alternative strategies. In this section, we first revisit the context matching problem, then three feasible matching strategies mentioned in our experiments are introduced in detail.

3.1 Problem Restatement

Given a template point and a search point, $2 \times k_c$ neighboring polar vectors are produced by the spatial contextual encoder. By calculating the distance between each template-search neighbor (polar vector) pair, we obtain $d_{mn}, \forall m, n \in [1, k_c]$. Then the distance matrix $\mathbf{D}$ is constructed as $\mathbf{D} = (d_{mn})_{k_c \times k_c}$. The matching strategy is aimed at mapping the matrix $\mathbf{D}$ to a scalar $\hat{d}$ that describes the distance between two distributions of neighbors.

3.2 Linear Assignment

The desired matching strategy can be viewed as a linear assignment problem (LAP) intending to minimize the global cost given a square cost matrix that refers to $\mathbf{D}$ in our problem. Once the problem is solved, we can obtain a fully interpretable neighbor-wise bipartite matching for two points, while the minimum global cost plays the role of $\hat{d}$. However, existing algorithms to solve LAP, *e.g.* the Hungarian method [5], are proved to have an unacceptable time overhead in our work, since a large number of individual LAPs have to be solved during the training and inference.

3.3 Optimal Transport and the IPOT Algorithm

We revisit the matching problem from a continuous perspective and model it as an optimal transport (OT) problem, which has approximate solutions with a much lower computational overhead.

OT provides a way to infer the correspondence between two distributions. Intuitively, the minimum cost (OT distance) from one distribution to another can be regarded as $\hat{d}$. Unfortunately, the exact OT minimization is in general computational intractable [1,7]. To overcome such intractability, the Inexact Proximal point method for Optimal Transport (IPOT) algorithm [8] is adopted to compute the OT matrix $\mathbf{T}$ as well as the OT distance. Specifically, IPOT iteratively solves the following optimization problem using the proximal point method [3]:

$$\mathbf{T}^{(t+1)} = \underset{\mathbf{T} \in \Pi(\mu, \nu)}{\arg\min} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle + \beta \cdot \mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)}) \right\} , \tag{3}$$

where $\Pi(\mu, \nu)$ denotes the set of all joint distributions, $\mathbf{C}$ is the cost matrix, and $1/\beta$ is understood as the generalized stepsize. The proximity metric term $\mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)})$ penalizes solutions that are too distant from the latest approximation. We follow [3] to employ the generalized KL Bregman divergence

Algorithm 1 IPOT algorithm

```
Input: cost matrix $\mathbf{C}_{m \times n}$, generalized stepsize $1/\beta$
$\boldsymbol{\sigma} \leftarrow \frac{1}{m}\mathbf{1_m}$
$\mathbf{T}^{(1)} \leftarrow \mathbf{1_n}\mathbf{1_m}^T$
$\mathbf{A}_{ij} \leftarrow \exp(-\mathbf{C}_{ij}/\beta)$
for $t = 1, 2, 3, \dots$ do
    $\mathbf{Q} \leftarrow \mathbf{A} \odot \mathbf{T}^{(t)}$        // $\odot$ denotes Hadamard product
    for $k = 1, \dots K$ do        // $K = 1$ in practice
        $\boldsymbol{\delta} \leftarrow \frac{1}{n\mathbf{Q}\boldsymbol{\sigma}}$, $\boldsymbol{\sigma} \leftarrow \frac{1}{m\mathbf{Q}^T\boldsymbol{\delta}}$
    end for
    $\mathbf{T}^{(t+1)} \leftarrow \mathrm{diag}(\boldsymbol{\delta})\,\mathbf{Q}\,\mathrm{diag}(\boldsymbol{\sigma})$
end for
Return T
```

$\mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)}) = \sum_{i,j} \mathbf{T}_{ij} \log\left(\mathbf{T}_{ij}/\mathbf{T}_{ij}^{(t)}\right) - \sum_{i,j} \mathbf{T}_{ij} + \sum_{i,j} \mathbf{T}_{ij}^{(t)}$ as the proximity metric. This renders a tractable iterative scheme towards the exact OT solution. The implementation details for IPOT is described in Algorithm 1.

3.4 Shifted-Minimum

In the assignment or transport solutions, all neighbors are included in the matching to achieve a global minimum cost, despite the fact that some of them are meaningless due to target moving or noise. Therefore, in order to model the matching problem with additional prior knowledge, a reasonable assumption can be established that partially sequential injective correspondences exist in similar spatial contexts, with noise points inserted between them.

An intuitive solution to partially sequential matching is aligning two neighbor sequences and shifting one of them until the position with the smallest average distance is found. As illustrated in Fig. 3, this strategy assumes that noise points are always at the ends of a sequence, which is a simplified version of our shifted-window matching. The advantage of shifted-minimum matching strategy is the high inference speed.

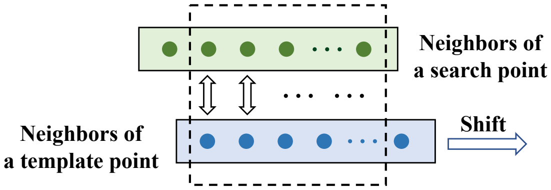

Fig. 3. Illustration of the shifted-minimum matching startegy.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the International Conference on Machine Learning. pp. 214–223. PMLR (2017)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
3. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: UNITER: Universal image-text representation learning. In: Proceedings of the European Conference on Computer Vision. pp. 104–120. Springer (2020)
4. Jiang, M., Wu, Y., Zhao, T., Zhao, Z., Lu, C.: PointSIFT: A SIFT-like network module for 3d point cloud semantic segmentation. arXiv preprint arXiv:1807.00652 (2018)
5. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
6. Qi, H., Feng, C., Cao, Z., Zhao, F., Xiao, Y.: P2B: Point-to-box network for 3d object tracking in point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6329–6338 (2020)
7. Salimans, T., Zhang, H., Radford, A., Metaxas, D.: Improving GANs using optimal transport. arXiv preprint arXiv:1803.05573 (2018)
8. Xie, Y., Wang, X., Wang, R., Zha, H.: A fast proximal point method for computing exact Wasserstein distance. In: Proceedings of the Uncertainty in Artificial Intelligence Conference. vol. 115, pp. 433–453. PMLR (2020)
9. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10076–10085 (2020)
10. Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V.: Point transformer. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 16259–16268 (2021)
11. Zheng, C., Yan, X., Gao, J., Zhao, W., Zhang, W., Li, Z., Cui, S.: Box-aware feature enhancement for single object tracking on point clouds. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 13199–13208 (2021)