3D Video Object Detection with Learnable Object-Centric Global Optimization

Jiawei He[1,2] Yuntao Chen[3] Naiyan Wang[4] Zhaoxiang Zhang[1,2,3]
[1] CRIPAC, Institute of Automation, Chinese Academy of Sciences (CASIA)
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)
[3] Centre for Artificial Intelligence and Robotics, HKISI_CAS [4] TuSimple
{hejiawei2019, zhaoxiang.zhang}@ia.ac.cn {chenyuntao08, winsty}@gmail.com

Abstract

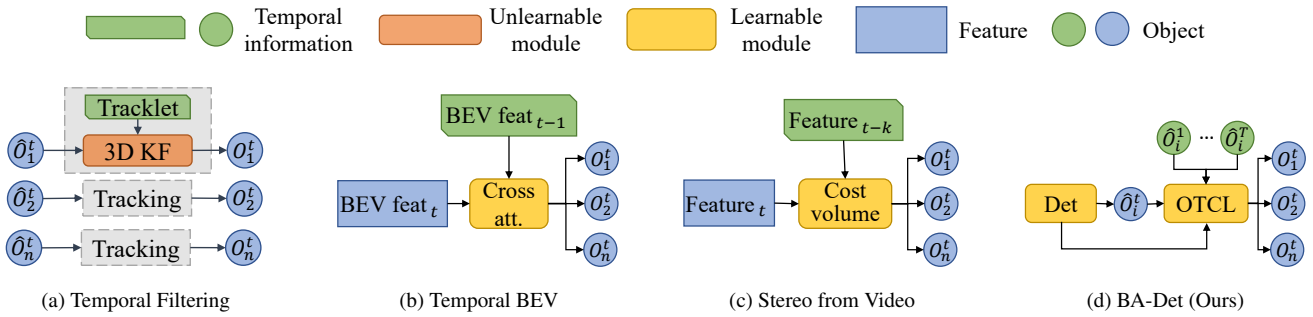
We explore long-term temporal visual correspondence-based optimization for 3D video object detection in this work. Visual correspondence refers to one-to-one mappings for pixels across multiple images. Correspondence-based optimization is the cornerstone for 3D scene reconstruction but is less studied in 3D video object detection, because moving objects violate multi-view geometry constraints and are treated as outliers during scene reconstruction. We address this issue by treating objects as first-class citizens during correspondence-based optimization. In this work, we propose BA-Det, an end-to-end optimizable object detector with object-centric temporal correspondence learning and featuremetric object bundle adjustment. Empirically, we verify the effectiveness and efficiency of BA-Det for multiple baseline 3D detectors under various setups. Our BA-Det achieves SOTA performance on the large-scale Waymo Open Dataset (WOD) with only marginal computation cost. Our code is available at https://github.com/jiaweihe1996/BA-Det.

1. Introduction

3D object detection is an important perception task, especially for indoor robots and autonomous-driving vehicles. Recently, image-only 3D object detection [23, 52] has been proven practical and made great progress. In real-world applications, cameras capture video streams instead of unrelated frames, which suggests abundant temporal information is readily available for 3D object detection. In single-frame methods, despite simply relying on the prediction power of deep learning, finding correspondences play an important role in estimating per-pixel depth and the object pose in the camera frame. Popular correspondences include Perspective-n-Point (PnP) between pre-defined 3D keypoints [22, 52] and their 2D projections in monocular 3D object detection, and Epipolar Geometry [6, 12] in multi-view 3D object detection. However, unlike the single-frame case, temporal visual correspondence has not been explored much in 3D video object detection.

As summarized in Fig. 1, existing methods in 3D video object detection can be divided into three categories while each has its own limitations. Fig. 1a shows methods with object tracking [3], especially using a 3D Kalman Filter to smooth the trajectory of each detected object. This approach is detector-agnostic and thus widely adopted, but it is just an output-level smoothing process without any feature learning. As a result, the potential of video is under-exploited. Fig. 1b illustrates the temporal BEV (Bird's-Eye View) approaches [14, 23, 26] for 3D video object detection. They introduce the multi-frame temporal cross-attention or concatenation for BEV features in an end-to-end fusion manner. As for utilizing temporal information, temporal BEV methods rely solely on feature fusion while ignoring explicit temporal correspondence. Fig. 1c depicts stereo-from-video methods [46, 47]. These methods explicitly construct a pseudo-stereo view using ego-motion and then utilize the correspondence on the epipolar line of two frames for depth estimation. However, the use of explicit correspondence in these methods is restricted to only two frames, thereby limiting its potential to utilize more temporal information. Moreover, another inevitable defect of these methods is that moving objects break the epipolar constraints, which cannot be well handled, so monocular depth estimation has to be reused.

Considering the aforementioned shortcomings, we seek a new method that can *handle both static and moving objects*, and *utilize long-term temporal correspondences*. Firstly, in order to handle both static and moving objects, we draw experience from the object-centric global optimization with reprojection constraints in Simultaneous Localization and Mapping (SLAM) [21, 48]. Instead of directly estimating the depth for each pixel from temporal cues, we utilize them to construct useful temporal constraints to refine the object pose prediction from network prediction. Specifically, we construct a non-linear least-square optimization problem with the temporal correspondence constraint in an



(a) Temporal Filtering (b) Temporal BEV (c) Stereo from Video (d) BA-Det (Ours)

Figure 1. Illustration of how to leverage temporal information in different 3D video object detection paradigms.

object-centric manner to optimize the pose of objects no matter whether they are moving or not. Secondly, for long-term temporal correspondence learning, hand-crafted descriptors like SIFT [27] or ORB [35] are no longer suitable for our end-to-end object detector. Besides, the long-term temporal correspondence needs to be robust to viewpoint changes and severe occlusions, where these traditional sparse descriptors are incompetent. So, we expect to learn a dense temporal correspondence for all available frames.

In this paper, as shown in Fig. 1d, we propose a 3D video object detection paradigm with learnable long-term temporal visual correspondence, called *BA-Det*. Specifically, the detector has two stages. In the first stage, a CenterNet-style monocular 3D object detector is applied for single-frame object detection. After associating the same objects in the video, the second stage detector extracts RoI features for the objects in the tracklet and matches dense local features on the object among multi-frames, called the object-centric temporal correspondence learning (OTCL) module. To make traditional object bundle adjustment (OBA) learnable, we formulate featuremetric OBA. In the training time, with featuremetric OBA loss, the object detection and temporal feature correspondence are learned jointly. During inference, we use the 3D object estimation from the first stage as the initial pose and associate the objects with 3D Kalman Filter. The object-centric bundle adjustment refines the pose and 3D box size of the object in each frame at the tracklet level, taking the initial object pose and temporal feature correspondence from OTCL as the input. Experiment results on the large-scale Waymo Open Dataset (WOD) show that our BA-Det could achieve state-of-the-art performance compared with other single-frame and multi-frame object detectors. We also conduct extensive ablation studies to demonstrate the effectiveness and efficiency of each component in our method.

In summary, our work has the following contributions:

- We present a novel object-centric 3D video object detection approach *BA-Det* by learning object detection and temporal correspondence jointly.
- We design the second-stage object-centric temporal correspondence learning module and the featuremetric object bundle adjustment loss.
- We achieve state-of-the-art performance on the large-scale WOD. The ablation study and comparisons show the effectiveness and efficiency of our BA-Det.

2. Related Work

2.1. 3D Video Object Detection

For 3D video object detection, LiDAR-based methods [4, 8, 49] usually align point clouds from consecutive frames by compensating ego-motion and simply accumulate them to alleviate the sparsity of point clouds. Object-level methods [5, 9, 33, 50], handling the multi-frame point clouds of the tracked object, become a new trend. 3D object detection from the monocular video has not received enough attention from researchers. Kinematic3D [3] is a pioneer work decomposing kinematic information into ego-motion and target object motion. However, they only apply 3D Kalman Filter [17] based motion model for kinematic modeling and only consider the short-term temporal association (4 frames). Recently, BEVFormer [23] proposes an attentional transformer method to model the spatial and temporal relationship in the bird's-eye-view (BEV). A concurrent work, DfM [46], inspired by Multi-view Geometry, considers two frames as stereo and applies the cost volume in stereo to estimate depth. However, how to solve the moving objects is not well handled in this paradigm.

2.2. Geometry in Videos

Many researchers utilize 3D geometry in videos to reconstruct the scene and estimate the camera pose, which is a classic topic of computer vision. Structure from Motion (SfM) [37] and Multi-view Stereo (MVS) [38] are two paradigms to estimate the sparse and dense depth from multi-view images respectively. In robotics, 3D geometry theory is applied for Simultaneous Localization and Mapping (SLAM) [30]. To globally optimize the 3D position of

the feature points and the camera pose at each time, bundle adjustment algorithm [42] is widely applied. However, most of them can only handle static regions in the scene.

In the deep learning era, with the development of object detection, object-level semantic SLAM [21, 31, 48] is rising, aiming to reconstruct the objects instead of the whole scene. These methods can handle dynamic scenes and help the object localization in the video. Besides, feature correspondence learning [36, 39] has received extensive attention in recent years. Deep learning has greatly changed the pipeline of feature matching. Differentiable bundle adjustment, like BANet [41] and NRE [11], makes the whole 3D geometry system end-to-end learnable. Unlike these works, we focus on the representation of the 3D object and integrate feature correspondence learning into 3D object detection. Utilizing the learned temporal feature correspondence, the proposed BA-Det optimizes the object pose of a tracklet in each frame.

3. Preliminary: Bundle Adjustment

Bundle Adjustment [42] is a widely used globally temporal optimization technology in 3D reconstruction, which means optimally adjusting bundles of light rays from a given 3D global position to the camera center among multi-frames. Specifically, we use $\mathbf{P}_i = [x_i, y_i, z_i]^\top$ to denote the i-th 3D point coordinates in the global reference frame. According to the perspective camera model, the image coordinates of the projected 3D point at time t is

$$\Pi(\mathbf{T}_{cg}^t, \mathbf{P}_i, \mathbf{K}) = \frac{1}{z_i^t}\mathbf{K}(\mathbf{R}_{cg}^t \mathbf{P}_i + \mathbf{t}_{cg}^t), \tag{1}$$

where Π is the perspective projection transformation, $\mathbf{T}_{cg}^t = [\mathbf{R}_{cg}^t | \mathbf{t}_{cg}^t]$ is the camera extrinsic matrix at time t. $\mathbf{R}_{cg}^t$ and $\mathbf{t}_{cg}^t$ are the rotation and the translation components of $\mathbf{T}_{cg}^t$, respectively. $\mathbf{K}$ is the camera intrinsic matrix, and z_i^t is the depth of the i-th 3D point in the camera frame at time t.

Bundle adjustment is a nonlinear least-square problem to minimize the reprojection error as:

$$\{\bar{\mathbf{T}}_{cg}^t\}_{t=1}^T, \{\bar{\mathbf{P}}_i\}_{i=1}^m = \underset{\{\mathbf{T}_{cg}^t\}_{t=1}^T, \{\mathbf{P}_i\}_{i=1}^m}{\arg\min} \frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{T} ||\mathbf{p}_i^t - \Pi(\mathbf{T}_{cg}^t, \mathbf{P}_i, \mathbf{K})||^2, \tag{2}$$

where $\mathbf{p}_i^t$ is the observed image coordinates of 3D point $\mathbf{P}_i$ on frame t. Bundle adjustment can be solved by Gauss-Newton or Levenberg–Marquardt algorithm effectively [1, 20].

4. BA-Det: Object-centric Global Optimizable Detector

In this section, we introduce the framework of our BA-Det (Fig. 2), a learnable object-centric global optimization network. The pipeline consists of three parts: (1) First-stage single frame 3D object detection; (2) Second-stage object-centric temporal correspondence learning (OTCL) module; (3) Featuremetric object bundle adjustment loss for temporal feature correspondence learning.

4.1. Single-frame 3D Object Detection

Given a video clip with consecutive frames $\mathcal{V} = \{I_1, I_2, \cdots, I_T\}$, 3D video object detection is to predict the class and the 3D bounding box of each object in each frame. Let $\mathcal{O}_k^t$ be the k-th object in frame t. For the 3D bounding box $\mathbf{B}_k^t$, we estimate the size of the bounding box $\mathbf{s}_t^k = [w, h, l]^\top$ and the object pose ${}^k\mathbf{T}_{co}^t$ in the camera frame, including translation ${}^k\mathbf{t}_{co}^t = [x_c, y_c, z_c]^\top$ and rotation ${}^k\mathbf{r}_{co}^t = [r_x, r_y, r_z]^\top$. In most 3D object detection datasets, with the flat ground assumption, only yaw rotation r_y is considered.

We basically adopt MonoFlex [52] as our first-stage 3D object detector, which is a simple and widely-used baseline method. Different from the standard MonoFlex, we make some modifications for simplicity and adaptation. (1) Instead of ensemble the depth from keypoints and regression, we only used the regressed depth directly. (2) The edge fusion module in MonoFlex is removed for simplicity and better performance. The output of the first-stage object detector should be kept for the second stage. The predicted 2D bounding box $\mathbf{b}_k^t$ for each object is used for the object-centric feature extraction in the second stage. The 3D estimations should be the initial pose estimation and be associated between frames. We follow ImmortalTracker [44] to associate the 3D box prediction outputs with a 3D Kalman Filter frame by frame. For convenience and clarity, we use the same index k to denote the objects belonging to the same tracklet in the video from now on.

4.2. Object-Centric Temporal Correspondence Learning

Based on the predictions from the first-stage detector, we propose an object-centric temporal correspondence learning (*OTCL*) module, which plays an indispensable role in the learnable optimization. Specifically, the OTCL module is designed to learn the correspondence of the dense features for the same object among all available frames. Given a video $\{I_1, I_2, \cdots, I_T\}$ and image features $\{\mathbf{F}^1, \mathbf{F}^2, \cdots, \mathbf{F}^T\}$ from the backbone in the first stage, we extract the RoI features ${}^k\mathbf{F}^t \in \mathbb{R}^{H\times W\times C}$ of the object $\mathcal{O}_k^t$ by the RoIAlign operation [13],

$${}^k\mathbf{F}^t = \texttt{RoIAlign}(\mathbf{F}^t, \mathbf{b}_k^t). \tag{3}$$

We apply L layers of cross- and self-attention operations before calculating the correspondence map to aggregate and enhance the spatial and temporal information for RoI features. Note that the object tracklet is available with the

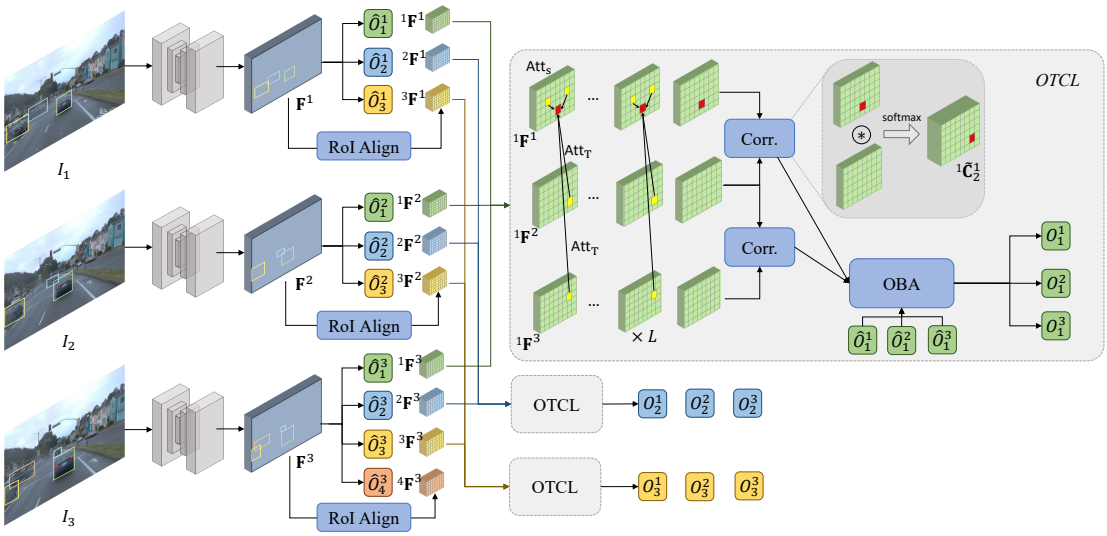


Figure 2. A overview of the proposed BA-Det framework. The left part of the framework is the first-stage object detector to predict the 3D object and its 2D bounding box. The second stage is called *OTCL* module. In the OTCL module, we extract the RoI features ${}^k\mathbf{F}^t$ by RoIAlign, aggregate the RoI features and learn object-centric temporal correspondence using featuremetric object bundle adjustment loss.

aforementioned tracker, so the cross-attention is applied between the objects in different frames for the same tracklet. For each layer of attention operations between two adjacent frames t and t':

$$\begin{cases} {}^k\widetilde{\mathbf{F}}^t = \texttt{Att}_\texttt{S}(Q,K,V) = \texttt{Att}_\texttt{S}({}^k\hat{\mathbf{F}}^t, {}^k\hat{\mathbf{F}}^t, {}^k\hat{\mathbf{F}}^t), \\ {}^k\widetilde{\mathbf{F}}^{t'} = \texttt{Att}_\texttt{S}(Q,K,V) = \texttt{Att}_\texttt{S}({}^k\hat{\mathbf{F}}^{t'}, {}^k\hat{\mathbf{F}}^{t'}, {}^k\hat{\mathbf{F}}^{t'}), \\ {}^k\hat{\mathbf{F}}^{t'} = \texttt{Att}_\texttt{T}(Q,K,V) = \texttt{Att}_\texttt{T}({}^k\widetilde{\mathbf{F}}^{t'}, {}^k\widetilde{\mathbf{F}}^{t}, {}^k\widetilde{\mathbf{F}}^{t}), \end{cases} \tag{4}$$

where ${}^k\hat{\mathbf{F}}^t \in \mathbb{R}^{HW\times C}$ is the flattened RoI feature, $\texttt{Att}_\texttt{S}$ is the spatial self-attention, $\texttt{Att}_\texttt{T}$ is the temporal cross-attention.

We then define the spatial correspondence map between two flattened RoI features after the attention operations. In frame pair (t,t'), we use ${}^k\mathbf{f}_i$ to denote i-th local feature in ${}^k\hat{\mathbf{F}}^{(L)}$ ($i \in \{1,2,\cdots,HW\}$). The correspondence map ${}^k\mathbf{C}_t^{t'} \in \mathbb{R}^{HW\times HW}$ in two frames is defined as the inner product of two features in two frames:

$${}^k\mathbf{C}_t^{t'}[i,i'] = {}^k\mathbf{f}_i^t * {}^k\mathbf{f}_{i'}^{t'}. \tag{5}$$

To normalize the correspondence map, we perform softmax over all spatial locations i',

$${}^k\widetilde{\mathbf{C}}_t^{t'}[i,i'] = \texttt{softmax}({}^k\mathbf{C}_t^{t'}[i,i']). \tag{6}$$

4.3. Featuremetric Object Bundle Adjustment Loss

In this subsection, we present that how to adapt and integrate the Object-centric Bundle Adjustment (OBA) into our learnable BA-Det framework, based on the obtained correspondence map. Generally speaking, we formulate the featuremetric OBA loss to supervise the temporal feature correspondence learning. Note that here we only derive the tracklet-level OBA loss for the same object, and for the final supervision we will sum all the tracklet-level loss in the video.

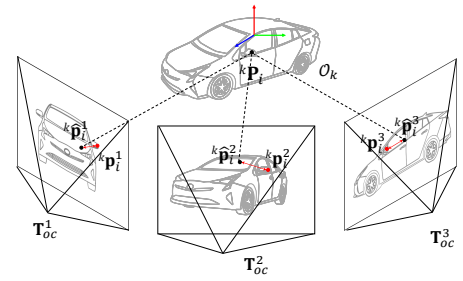
First, we revisit the object-centric bundle adjustment, as shown in Fig. 3a. As proposed in Object SLAM [21, 48], OBA assumes that the object can only have rigid motion relative to the camera. For the object $\mathcal{O}_k$, we denote the 3D points as $\mathcal{P}_k = \{{}^k\mathbf{P}_i\}_{i=1}^m$ in the object frame, 2D points as $\{{}^k\mathbf{p}_i^t\}_{i=1}^m$, 2D features at position ${}^k\mathbf{p}_i^t$ as $\{\mathbf{f}[{}^k\mathbf{p}_i^t]\}_{i=1}^m$, and the camera pose in the object reference frame as $\mathcal{T}_k = \{{}^k\mathbf{T}_{co}^t\}_{t=1}^T$, OBA can be casted as:

$$\bar{\mathcal{T}}_k, \bar{\mathcal{P}}_k = \underset{\mathcal{T}_k,\mathcal{P}_k}{\arg\min} \frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{T} ||{}^k\mathbf{p}_i^t - \Pi({}^k\mathbf{T}_{co}^t, {}^k\mathbf{P}_i, \mathbf{K})||_2^2. \tag{7}$$

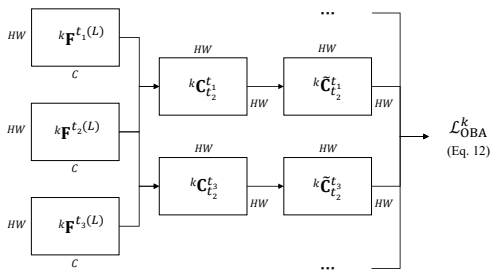
To make the OBA layer end-to-end learnable, we formulate featuremetric [25] OBA:

$$\bar{\mathcal{T}}_k, \bar{\mathcal{P}}_k = \underset{\mathcal{T}_k,\mathcal{P}_k}{\arg\min} \frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{T}\sum_{t'=1}^{T} ||\mathbf{f}[{}^k\mathbf{p}_i^t] - \mathbf{f}[\Pi({}^k\mathbf{T}_{co}^{t'}, {}^k\mathbf{P}_i, \mathbf{K})]||_2^2, \tag{8}$$

where $\mathbf{f}[\mathbf{p}]$ denotes the feature vector in pixel coordinates $\mathbf{p}$. Representing the 3D point ${}^k\mathbf{P}_i$ in Eq. 8 with 2D points in each frame, the featuremetric reprojection error of frame



(a) Object-centric Bundle Adjustment (OBA).



(b) The computation of the featuremetric OBA loss.

Figure 3. Illustration of featuremetric object bundle adjustment.

t could be derived as

$$ {}^{k}e_i^t = \sum_{t'=1}^{T} \mathbf{f}[{}^{k}\mathbf{p}_i^t] - \mathbf{f}[{}^{k}\mathbf{p}_i^{t'}] \tag{9} $$

$$ = \sum_{t'=1}^{T} \mathbf{f}[{}^{k}\mathbf{p}_i^t] - \mathbf{f}[\Pi({}^{k}\mathbf{T}_{co}^{t'}, \Pi^{-1}({}^{k}\mathbf{T}_{co}^{t}, {}^{k}\mathbf{p}_i^t, \mathbf{K}, z_i^t), \mathbf{K})], \tag{10} $$

where $\Pi^{-1}(\cdot)$ is the inverse projection function to lift the 2D point on the image to 3D in the object frame. z_i^t is the ground-truth depth of ${}^{k}\mathbf{p}_i^t$ (from LiDAR point clouds only for training). In the training time, we learn the feature correspondence, given the ground-truth pose of the object $\mathcal{O}_k$, denoted as ${}^{k}\mathbf{T}_{co}^{t}$ and ${}^{k}\mathbf{T}_{co}^{t'}$ in frame t and frame t', respectively. Considering the featuremetric reprojection loss in all frames and all points, the overall loss term for object k can be formulated as

$$ \mathcal{L}_{\text{rep}}^{k} = \sum_{i=1}^{m}\sum_{t=1}^{T} ||{}^{k}e_i^t||_2^2 = \sum_{i=1}^{m}\sum_{t=1}^{T}\sum_{t'=1}^{T} ||{}^{k}\mathbf{f}_i^t - {}^{k}\mathbf{f}_i^{t'}||_2^2 \tag{11} $$

Finally, we replace the $L2$ norm in Eq. 11 with the cosine distance to measure the featuremetric reprojection error. Thus we bring the normalized correspondence map $\widetilde{\mathbf{C}}$ in Sec. 4.2 into the loss term. With log-likelihood formulation, we formulate the featuremetric OBA loss to supervise the object-centric temporal correspondence learning:

$$ \mathcal{L}_{\text{OBA}}^{k} = -\sum_{i=1}^{m}\sum_{t=1}^{T}\sum_{t'=1}^{T} \log({}^{k}\widetilde{\mathbf{C}}_t^{t'}[{}^{k}\bar{\mathbf{p}}_i^t, {}^{k}\bar{\mathbf{p}}_i^{t'}]). \tag{12} $$

where $({}^{k}\bar{\mathbf{p}}_i^t, {}^{k}\bar{\mathbf{p}}_i^{t'})$ are the ground-truth corresponding pair of the i-th local feature. The illustration of the loss computation is in Fig. 3b.

4.4. Inference

After introducing the training loss design, we present the inference process of BA-Det as follows.

First-stage 3D object detection and association. The first-stage detector makes the prediction of classification scores and 2D / 3D bounding boxes. The 3D bounding boxes are associated across the frames by Immortal-Tracker [44]. The following process is on the tracklet level.

Dense feature matching. To optimize the object pose, we need to obtain the feature correspondence in each frame for the same object. As mentioned in Sec. 4.2, the OTCL module is trained to generate a dense correspondence map in all frames. During inference, we match all $H \times W$ dense local features in RoI between adjacent two frames and between the first frame and last frame of the time window $[t, t+\tau]$. We use the RANSAC algorithm [10] to filter the feature correspondence outliers.

Feature tracking. To form a long-term keypoint tracklet from the obtained correspondence, we leverage a graph-based algorithm. First, the matched feature pairs are constructed into a graph $\mathcal{G}$. The features are on the vertices. If the features are matched, an edge is connected in the graph. Then we track the feature for the object in all available frames. We use the association method mainly following [7]. The graph partitioning method is applied to $\mathcal{G}$ to make each connected subgraph have at most one vertex per frame. The graph cut is based on the similarity of the matched features.

Object-centric bundle adjustment. In the inference stage, given the initial pose estimation and the temporal feature correspondence, we solve the object-centric bundle adjustment by Levenberg–Marquardt algorithm, and the object pose in each frame and the 3D position of the keypoints can be globally optimized between frames.

Post-processing. We also apply some common post-processing in video object detection techniques like tracklet rescoring [18] and bounding box temporal interpolation.

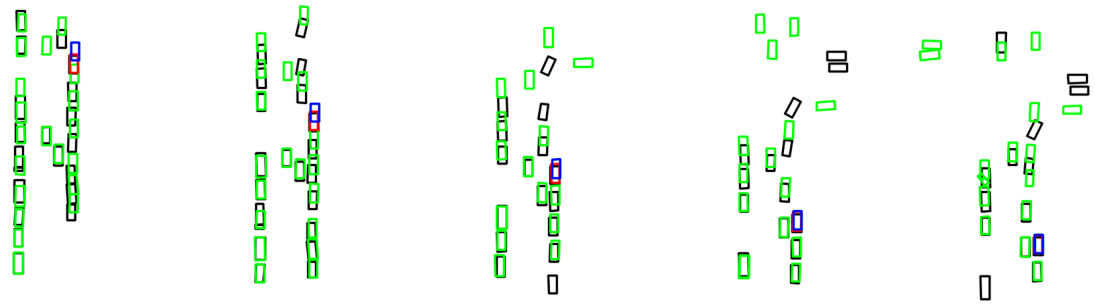
5. Experiments

5.1. Datasets and metrics

We conduct our experiments on the large autonomous driving dataset, Waymo Open Dataset (WOD) [40]. The WOD has different versions with different annotations and metrics. To keep the fairness of the comparisons, we report the results both on WOD v1.2 and WOD v1.3.1. The annotations on v1.2 are based on LiDAR and the official metrics are mAP IoU@0.7 and mAP IoU@0.5. Recently, v1.3.1 is released to support multi-camera 3D object detec-

	LEVEL_1				LEVEL_2			
	3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}	3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}
M3D-RPN [2]	0.35	0.34	3.79	3.63	0.33	0.33	3.61	3.46
PatchNet [29]	0.39	0.37	2.92	2.74	0.38	0.36	2.42	2.28
PCT [43]	0.89	0.88	4.20	4.15	0.66	0.66	4.03	3.99
MonoJSG [24]	0.97	0.95	5.65	5.47	0.91	0.89	5.34	5.17
GUPNet [28]	2.28	2.27	10.02	9.94	2.14	2.12	9.39	9.31
DEVIANT [19]	2.69	2.67	10.98	10.89	2.52	2.50	10.29	10.20
CaDDN [34]	5.03	4.99	17.54	17.31	4.49	4.45	16.51	16.28
DID-M3D [32]	-	-	20.66	20.47	-	-	19.37	19.19
BEVFormer [23]†	-	7.70	-	30.80	-	6.90	-	27.70
DCD [22]	12.57	12.50	33.44	33.24	11.78	11.72	31.43	31.25
MonoFlex [52] (Baseline)	11.70	11.64	32.26	32.06	10.96	10.90	30.31	30.12
BA-Det(Ours)†	**16.60**	**16.45**	**40.93**	**40.51**	**15.57**	**15.44**	**38.53**	**38.12**

Table 1. The results on WODv1.2 [40] *val* set. AP_{70} denotes AP with IoU threshold at 0.7. AP_{50} denotes AP IoU@0.5.† denotes the method utilizing temporal information.



(a) Frame 8. (b) Frame 22. (c) Frame 36. (d) Frame 50. (e) Frame 57.

Figure 4. Qualitative results from the BEV in different frames. We use blue and red boxes to denote initial predictions and optimized predictions of the object we highlight. The green and black boxes denote the other box predictions and the ground truth boxes. The ego vehicle lies at the bottom of each figure.

Method	LET-APL	LET-AP	LET-APH	3D AP_{70}	3D AP_{50}
MV-FCOS3D++ [45]†	58.11	74.68	73.50	14.66	36.02
BA-Det$_{FCOS3D}$(Ours)†	**58.47**	**74.85**	**73.66**	**15.02**	**36.89**

Table 2. The multi-camera results on WODv1.3.1 [16] *val* set. Besides the official LET-IoU-based metrics, we also report the metrics with standard 3D IoU. All metrics are reported for the LEVEL_2 difficulty.†: use temporal information.

tion, and the annotations are camera-synced boxes. On the v1.3.1 dataset, a series of new LET-IoU-based metrics [16] are introduced to slightly tolerate the localization error from the worse sensor, camera, than LiDAR. Early work mainly reports the results on the v1.2 dataset, and we only compare our methods with the ones from WOD Challenge 2022 using the v1.3.1 dataset. Because we mainly focus on rigid objects, we report the results of the VEHICLE class.

LET-3D-AP and LET-3D-APL are the new metrics, relying on the Longitudinal Error Tolerant IoU (LET-IoU). LET-IoU is the 3D IoU calculated between the target ground truth box and the prediction box aligned with ground truth along the depth that has minimum depth error. LET-3D-AP and LET-3D-APL are calculated from the average precision and the longitudinal affinity weighted average precision of the PR curve. For more details, please refer to [16].

5.2. Implementation Details

The first stage network architecture of BA-Det is the same as MonoFlex, with DLA-34 [51] backbone, the output feature map is with the stride of 8. In the second stage, the shape of the RoI feature is 60×80. The spatial and temporal attention module is stacked with 4 layers. The implementation is based on the PyTorch framework. We train our model on 8 NVIDIA RTX 3090 GPUs for 14 epochs. Adam optimizer is applied with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is 5×10^{-4} and weight decay is 10^{-5}. The learning rate scheduler is one-cycle. We use the Levenberg-Marquardt algorithm, implemented by DeepLM [15], to solve object-centric bundle adjustment. The maximum it-

	Method	3D AP_{70}			3D APH_{70}			3D AP_{50}			3D APH_{50}		
		0-30	30-50	50-∞	0-30	30-50	50-∞	0-30	30-50	50-∞	0-30	30-50	50-∞
L1	DCD [22]	32.47	5.94	1.24	32.30	5.91	1.23	62.70	26.35	10.16	62.35	26.21	10.09
	MonoFlex [52]	30.64	5.29	1.05	30.48	5.27	1.04	61.13	25.85	9.03	60.75	25.71	8.95
	BA-Det (Ours)†	**37.74**	**11.04**	**3.86**	**37.46**	**10.95**	**3.79**	**71.07**	**37.15**	**14.89**	**70.46**	**36.79**	**14.61**
L2	DCD [22]	32.30	5.76	1.08	32.19	5.73	1.08	62.48	25.60	8.92	62.13	25.46	8.86
	MonoFlex [52]	30.54	5.14	0.91	30.37	5.11	0.91	60.91	25.11	7.92	60.54	24.97	7.85
	BA-Det (Ours)†	**37.61**	**10.72**	**3.37**	**37.33**	**10.63**	**3.31**	**70.83**	**36.14**	**13.62**	**70.23**	**35.79**	**13.37**

Table 3. The object depth range conditioned result on WODv1.2 [40] *val* set. L1 and L2 denote LEVEL_1 and LEVEL_2 difficulty, respectively. †: use temporal information.

	LEVEL_1				LEVEL_2			
	3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}	3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}
MonoFlex (baseline)	11.70	11.64	32.26	32.06	10.96	10.90	30.31	30.12
Our first-stage prediction	13.57	13.48	34.70	34.43	12.72	12.64	32.56	32.32
+3D Tracking [44]	14.01	13.93	35.19	34.92	13.13	13.05	33.03	32.78
+ Learnable global optimization	15.85	15.75	38.06	37.76	14.87	14.77	35.72	35.44
+ Tracklet rescoring	16.43	16.30	40.07	39.70	15.41	15.29	37.66	37.31
+ Bbox interpolation	16.60	16.45	40.93	40.51	15.57	15.44	38.53	38.12

Table 4. Ablation study of each component in BA-Det.

eration of the LM algorithm is 200. For the object that appears less than 10 frames or the average keypoint number is less than 5, we do not optimize it.

5.3. Comparisons with State-of-the-art Methods

We compare our BA-Det with other state-of-the-art methods under two different settings. WODv1.2 is for the front view camera and WODv1.3.1 has the official evaluator for all 5 cameras. As shown in Table 1, using the FRONT camera, we outperform the SOTA method DCD [22] for about 4AP and 4APH ($\sim$30% improvement) under the 0.7 IoU threshold. Compared with the only temporal method BEVFormer [23], we have double points of 3D AP_{70} and 3D APH_{70}. To validate the effectiveness, we also report the multi-camera results on the newly released WODv1.3.1, as shown in Table 2. No published work reports the results on WODv1.3.1. So, we only compare with the open-source MV-FCOS3D++ [45], the second-place winner of WOD 2022 challenge. We design the variant of BA-Det, called BA-$\text{Det}_{\text{FCOS3D}}$, to adapt to the multi-camera setting. BA-$\text{Det}_{\text{FCOS3D}}$ is also a two-stage object detector. The first stage is the same as MV-FCOS3D++, but with the output of 2D bounding boxes. The second stage is OTCL module supervised with featuremetric object bundle adjustment loss. Although there are overlaps between 5 cameras, to simplify the framework, we ignore the object BA optimization across cameras and only conduct temporal optimization. BA-$\text{Det}_{\text{FCOS3D}}$ outperforms MV-FCOS3D++ under main metrics and traditional 3D IoU-based metrics.

5.4. Qualitative Results

In Fig. 4, we show the object-level qualitative results of the first-stage and second-stage predictions in different frames. For a tracklet, we can refine the bounding box predictions with the help of better measurements in other frames, even if there is a long time interval between them.

5.5. Distance Conditioned Results

We report the results with the different depth ranges in Table 3. The results indicate that the single frame methods, like DCD and MonoFlex, are seriously affected by object depth. When the object is farther away from the ego vehicle, the detection performance drops sharply. Compared with these methods, BA-Det, has the gain almost from the object far away from the ego-vehicle. The 3D AP_{70} and 3D APH_{70} are $3\times$ compared with the baseline when the object is located in $[50\text{m}, \infty)$, $2\times$ in $[30\text{m}, 50\text{m})$ and $1.2\times$ in $[0\text{m}, 30\text{m})$. This is because we utilize the long-term temporal information for each object. In a tracklet, the predictions near the ego-vehicle can help to refine the object far away.

5.6. Ablation study

We ablate each component of BA-Det. The results are shown in Table 4. The first stage detector is slightly better than the MonoFlex baseline mainly because we remove the edge fusion module, which is harmful to the truncated objects in WOD. 3D KF associates the objects and smooths the object's trajectory. This part of improvement can be regarded as similar to Kinematic3D [3]. The core of BA-Det

	LEVEL_1				LEVEL_2			
	3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}	3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}
MonoFlex (baseline)	11.70	11.64	32.26	32.06	10.96	10.90	30.31	30.12
Initial prediction	13.57	13.48	34.70	34.43	12.72	12.64	32.56	32.32
Static BA	14.73	14.62	37.89	37.56	13.82	13.72	35.65	35.34
Ours	16.60	16.45	40.93	40.51	15.57	15.44	38.53	38.12

Table 5. Comparison between object-centric BA-Det and the traditional scene-level bundle adjustment (Static BA). Initial prediction denotes the predictions in the first stage.

	$\bar{L}_t$	LEVEL_1				LEVEL_2			
		3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}	3D AP_{70}	3D APH_{70}	3D AP_{50}	3D APH_{50}
MonoFlex (baseline)	-	11.70	11.64	32.26	32.06	10.96	10.90	30.31	30.12
BA-Det+ ORB feature [35]	2.6	14.05	13.96	35.21	34.95	13.17	13.08	33.05	32.81
BA-Det+ Our feature	10	16.60	16.45	40.93	40.51	15.57	15.44	38.53	38.12

Table 6. Ablation study about different feature corresponding methods. $\bar{L}_t$ denotes the average keypoint tracklet length for each object.

is the learnable global optimization module, which obtains the largest gain in all modules. The tracklet rescoring and temporal interpolation modules are also useful.

5.7. Further Discussions

BA vs. Object BA. We conduct experiments to discuss whether the object-centric manner is important in temporal optimization. We modify our pipeline and optimize the whole scene in the *global* frame instead of optimizing the object pose in the object frame, called Static BA in Table 5. Static BA ignores dynamic objects and treats them the same as static objects. The inability to handle dynamic objects causes decreases by about 2 AP compared with BA-Det.

Temporal feature correspondence. As shown in Table 6, we ablate the features used for object-centric bundle adjustment. Compared with traditional ORB feature [35], widely used in SLAM, our feature learning module predicts denser and better correspondence. We find the average object tracklet length is 19.6 frames, and the average feature tracklet in our method is about 10 frames, which means we can keep a long feature dependency and better utilize long-range temporal information. However, the $\bar{L}_t$ of the ORB feature is only 2.6 frames. The results show the short keypoint tracklet can not refine the long-term object pose well.

Inference latency of each step in BA-Det. The inference latency of each step in BA-Det is shown in Table 7. The most time-consuming part is the first-stage object detector, more than 130ms per image, which is the same as the MonoFlex baseline. Our BA-Det only takes an additional 50ms latency per image, compared with the single-frame detector MonoFlex. Besides, although the dense feature correspondence is calculated, thanks to the shared backbone with the first stage detector and parallel processing for the objects, the feature correspondence module is not very time-consuming.

Total latency	181.5ms
First-stage detector	132.6ms
Object tracking	6.6ms
Feature correspondence	23.0ms
Object bundle adjustment	19.3ms

Table 7. Inference latency of each step in BA-Det per image.

6. Limitations and Future Work

In the current version of this paper, we only focus on the objects, such as cars, trucks, and trailers. The performance of non-rigid objects such as pedestrians has not been investigated. However, with mesh-based and skeleton-based 3D human models, we believe that a unified keypoint temporal alignment module can be designed in the future. So, we will explore the extension of BA-Det for non-rigid objects.

7. Conclusion

In this paper, we propose a 3D video object detection paradigm with long-term temporal visual correspondence, called BA-Det. BA-Det is a two-stage object detector that can jointly learn object detection and temporal feature correspondence with proposed Featuremetric OBA loss. Object-centric bundle adjustment optimizes the first-stage object estimation globally in each frame. BA-Det achieves state-of-the-art performance on WOD.

Acknowledgements

This work was supported in part by the Major Project for New Generation of AI (No.2018AAA0100400), the National Natural Science Foundation of China (No. 61836014, No. U21B2042, No. 62072457, No. 62006231) and the InnoHK program. The authors thank Lue Fan and Yuqi Wang for their valuable suggestions.

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver. https://github.com/ceres-solver/ceres-solver, 2022. 3

[2] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 2019. 6

[3] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, 2020. 1, 2, 7

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2

[5] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *ECCV*, 2022. 2

[6] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In *CVPR*, 2020. 1

[7] Mihai Dusmanu, Johannes L Schönberger, and Marc Pollefeys. Multi-view optimization of local feature geometry. In *ECCV*, 2020. 5

[8] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *CVPR*, 2022. 2

[9] Lue Fan, Yuxue Yang, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Super sparse 3d object detection. *arXiv preprint arXiv:2301.02562*, 2023. 2

[10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[11] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Neural reprojection error: Merging feature learning and camera pose estimation. In *CVPR*, 2021. 3

[12] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *ICCV*, 2021. 1

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[14] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 1

[15] Jingwei Huang, Shan Huang, and Mingwei Sun. Deeplm: Large-scale nonlinear least squares on deep learning frameworks using stochastic domain decomposition. In *CVPR*, 2021. 6

[16] Wei-Chih Hung, Henrik Kretzschmar, Vincent Casser, Jyh-Jing Hwang, and Dragomir Anguelov. Let-3d-ap: Longitudinal error tolerant 3d average precision for camera-only 3d detection. *arXiv preprint arXiv:2206.07705*, 2022. 6

[17] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 2

[18] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2017. 5

[19] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. Deviant: Depth equivariant network for monocular 3d object detection. In *ECCV*, 2022. 6

[20] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *ICRA*, 2011. 3

[21] Peiliang Li, Tong Qin, et al. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In *ECCV*, 2018. 1, 3, 4

[22] Yingyan Li, Yuntao Chen, Jiawei He, and Zhaoxiang Zhang. Densely constrained depth estimator for monocular 3d object detection. In *ECCV*, 2022. 1, 6, 7

[23] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 1, 2, 6, 7

[24] Qing Lian, Peiliang Li, and Xiaozhi Chen. Monojsg: Joint semantic and geometric cost volume for monocular 3d object detection. In *CVPR*, 2022. 6

[25] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *ICCV*, 2021. 4

[26] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 1

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[28] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *ICCV*, 2021. 6

[29] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *ECCV*, 2020. 6

[30] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2

[31] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, 2018. 3

[32] Liang Peng, Xiaopei Wu, Zheng Yang, Haifeng Liu, and Deng Cai. Did-m3d: Decoupling instance depth for monocular 3d object detection. In *ECCV*, 2022. 6

[33] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *CVPR*, 2021. 2

[34] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 6

[35] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 2, 8

[36] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 3

[37] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[38] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2

[39] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 3

[40] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 5, 6, 7

[41] Chengzhou Tang and Ping Tan. BA-net: Dense bundle adjustment networks. In *ICLR*, 2019. 3

[42] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *ICCV Workshops*, 1999. 3

[43] Li Wang, Li Zhang, Yi Zhu, Zhi Zhang, Tong He, Mu Li, and Xiangyang Xue. Progressive coordinate transforms for monocular 3d object detection. *NeurIPS*, 2021. 6

[44] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal tracker: Tracklet never dies. *arXiv preprint arXiv:2111.13672*, 2021. 3, 5, 7

[45] Tai Wang, Qing Lian, Chenming Zhu, Xinge Zhu, and Wenwei Zhang. MV-FCOS3D++: Multi-View camera-only 4d object detection with pretrained monocular backbones. *arXiv preprint arXiv:2207.12716*, 2022. 6, 7

[46] Tai Wang, Jiangmiao Pang, and Dahua Lin. Monocular 3d object detection with depth from motion. In *ECCV*, 2022. 1, 2

[47] Zengran Wang, Chen Min, Zheng Ge, Yinhao Li, Zeming Li, Hongyu Yang, and Di Huang. Sts: Surround-view temporal stereo for multi-view 3d detection. *arXiv preprint arXiv:2208.10145*, 2022. 1

[48] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019. 1, 3, 4

[49] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 2

[50] Yurong You, Katie Z Luo, Xiangyu Chen, Junan Chen, Wei-Lun Chao, Wen Sun, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Hindsight is 20/20: Leveraging past traversals to aid 3d perception. In *ICLR*, 2022. 2

[51] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018. 6

[52] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, 2021. 1, 3, 6, 7