# Bird's-Eye View from Monocular Cameras

Group 3

Mery Tom
tom.mery@epfl.ch
sciper: 297217

Charfeddine Ramy
ramy.charfeddine@epfl.ch
sciper: 295758

***Abstract*—The objective is to develop a deep learning model that can transform monocular camera images of the surrounding into a bird's eye view map. The output of the model will be bird's eye view map that can be used to train the object detection, tracking and predictions algorithms of the Tesla Autopilot system.**

## I. Introduction

Scene perception is an essential task for autonomous vehicles and robots in order to ensure their motion in an undefined environment. In order to achieve this challenge, and despite the tremendous progress of LiDAR-based methods, camera-based approaches have attracted attention in recent years. Indeed, beyond the low cost for deployment, monocular cameras have the ability to capture abundant textual information, detect long-range distance objects, and identify vision-based road elements (e.g. traffic lights, stoplines), compared to LiDAR-based counterparts. Thus, instead of depicting the 3D environment entirely, the scene is represented in the bird's-eye view (BEV) coordinate frame. This representation has the advantage of providing a more succinct and effective way to understand the surrounding environment of the main vehicle. The geometry and layout of the static world, as well as the location and dimensions of the dynamic agents are captured on a single top-down view, which is more convenient for the downstream motion planning and prediction tasks. To generate the BEV representation from monocular cameras, several approaches exist. The most basic one is to simply consider the problem as a perspective mapping using intrinsic/extrinsic camera parameters and then perform shearing and enlarging to generate BEV images [1]. However this old-fashion technique has many disadvantages and actual methods use deep learning such as Lift-Splat-Shoot that uses convolutional encoder-decoder architecture [2]. More recent literature [3], [4] proposes a transformer-based encoder-decoder architecture to translate the image features from different monocular cameras into the BEV frame. It has the advantage of taking into account the spatial context information in the individual images and the relationship between images in different views. State-of-the-art architectures go even further by incorporating temporal informations using the previous frames [5], [6], [7]. Intuitively, temporal information could help to detect occluded agents or can be used to estimate physical properties such as velocity.

## II. Problem statement

The problem is to develop a deep learning model that can take as input images captured by cameras installed all around a vehicle and produce a bird's-eye view map of the surrounding area. The aim is to improve the vehicle's situational awareness and aiding in autonomous navigation and agent's trajectory prediction. Mathematically this can be written as:

- Input: $\left\{\mathbf{I}_k \in \mathbb{R}^{3\times H\times W} \mid k = 1, \ldots, N_c\right\}$ a set of images from $N_c$ surrounding cameras where $H$ and $W$ represent the height and width of the images [3].
- Output: $B \in \mathbb{R}^{\tilde{H}\times\tilde{W}\times C}$ features with $\tilde{H}$ and $\tilde{W}$ the spatial shape of the BEV plane [6].

## III. Selected Method

BEVFormer models [6] have been shown to achieve state-of-the-art performance on a variety of benchmarks for BEV map generation in autonomous driving applications.

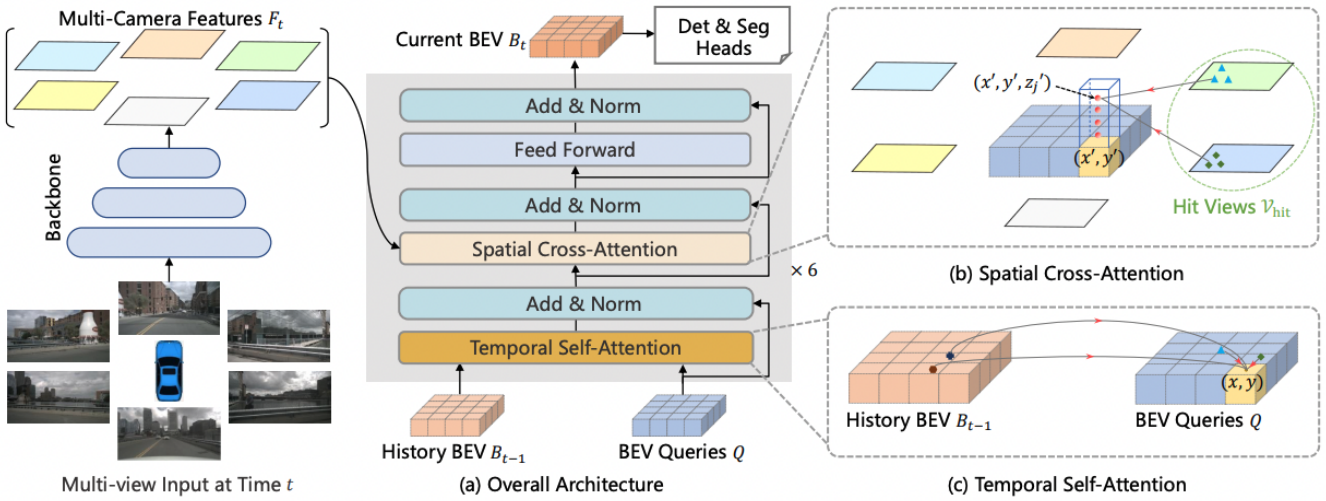


Fig. 1. **Overall architecture of BEVFormer** [6]. (a) The encoder layer of BEVFormer contains grid-shaped BEV queries, temporal self-attention, and spatial cross-attention. (b) In spatial crossattention, each BEV query only interacts with image features in the regions of interest. (c) In temporal self-attention, each BEV query interacts with two features: the BEV queries at the current timestamp and the BEV features at the previous timestamp.

This model consists of a convolutional neural network (CNN) backbone that extracts features from the input images and a transformer-based architecture that converts these features into a top-down representation of the surrounding environment. The overall architecture of BEVFormer is shown in figure 1. The implementation is already available at https://github.com/fundamentalvision/BEVFormer and pre-trained models with different backbones are provided.

The model works such that during inference, at timestamp *t*, the monocular images are fed to the backbone network (e.g. ResNet-101) to obtain the features $F_t = \{F_t^i\}_{i=1}^{N_c}$ of the different camera views. $F_t^i$ is the feature of the i-th view, $N_c$ is the total number of camera views. At the same time, the BEV features $B_{t-1}$ at the prior timestamp $t-1$ are preserved. In each encoder layer, the BEV queries Q are first used to query the temporal information from the prior BEV features $B_{t-1}$ via the temporal self-attention. The BEV queries Q are then employed to inquire about the spatial information from the multi-camera features $F_t$ via the spatial cross-attention. After the feed-forward network, the encoder layer outputs the refined BEV features, which is the input of the next encoder layer. After 6 stacked encoder layers, unified BEV features $B_t$ at current timestamp $t$ are generated. Taking the BEV features $B_t$ as input, the 3D detection head and map segmentation head predict the perception results such as 3D bounding boxes and semantic map can be generated. The genration of these tasks is explained further in the paper [6].

## IV. Dataset and Evaluation Metrics

### A. Dataset

In order to train and evaluate the model, nuScenes, a large-scale autonomous driving dataset with 3D object annotations, will be used. It consists of 1000 scenes captured from four locations in Boston and Singapore, each of 20 seconds in length, covering different conditions. The images are captured from 6 surround-view cameras which provides a 360° view with a slight overlap between the neighboring cameras. As the dataset already provides annotated 3D objects with a category, attributes and 3D bounding box it can be used for training and testing. Besides, vectorized maps of different semantic classes and the pose of the ego-vehicle are provided. Note that in our case, we don't need the 3D representation of the environment, therefore, before using some elements of the dataset as ground truth, some preprocessing needs to be done. Thus, the LiDAR representations with the annotated 3D

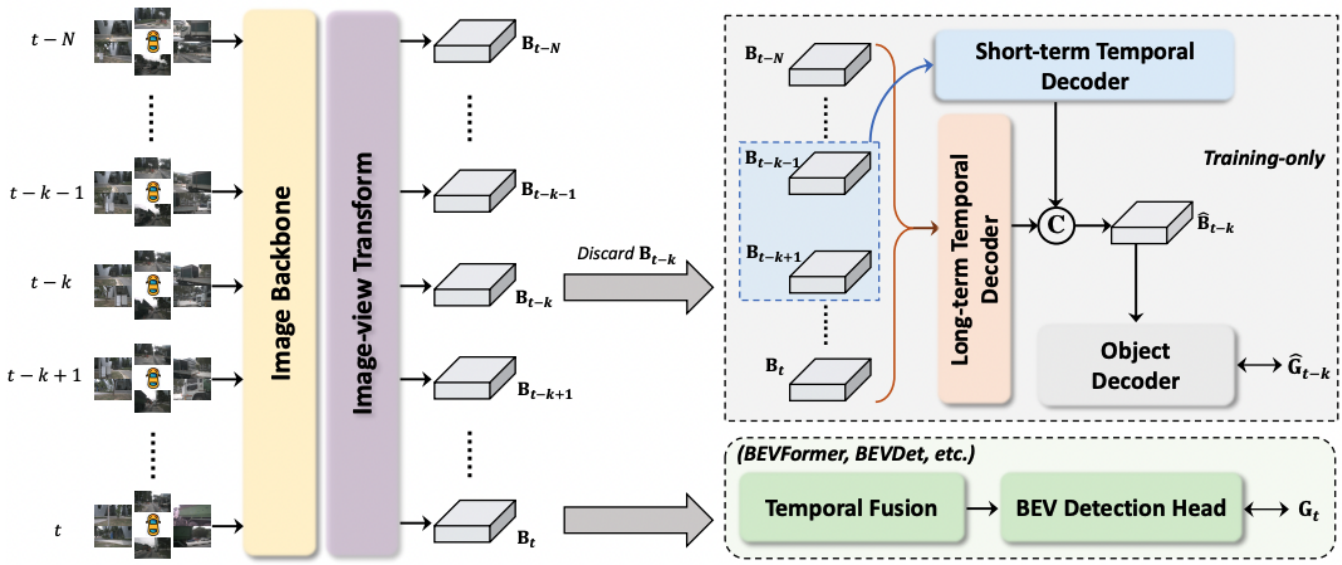


Fig. 2. **Framework of the Historical Object Prediction (HoP)** [8]. The auxiliary branches are discarded during evaluation.

objects will first be projected into a the BEV space to be compared to the results of our model at the evaluation stage.

### *B. Evaluation metrics*

The model must be able to process input images and accurately identify and classify different objects in the scene, such as cars, pedestrians, buildings, and roads. The model must also be able to estimate the distance and orientation of these objects relative to the vehicle, and use this information to construct a bird's-eye view map. Therefore, the evaluation of the model will be done for two different tasks: 3D object detection and BEV map segmentation as presented in [6]. Practically, the implementation of these two tasks is achieved through a shared encoder that generates the features in the bird's eye map as presented above, before going to two separated networks who would perform the map segmentation and 3D object detection. Assessing the performance of the model for different tasks allow to assess the quality of the encoded BEV features $B \in \mathbb{R}^{\tilde{H} \times \tilde{W} \times C}$. Note that as the evaluation can't be performed on the BEV features map at the transformer's output, the 3D objection detection and map segmentation tasks are required even if it goes beyond the objective of the project.

- The Intersection over Union (IoU) score which specifies the amount of overlap between the predicted and ground truth bounding box will be used to evaluate the performance of the model for the map segmentation task.
- The mean Average Precision (mAP), calculated by taking the mean Average Precision over all classes and/or over all IoU thresholds, will be used to assess the performance on the 3D object detection task.

With these metrics, one will be able to compare the model to state-of-the-art implementations by comparing the scores and the runtime for the different classes (vehicles, lines...).

## V. Proposed Contribution

Very recent paper [8] comes up with a new paradigm, named Historical Object Prediction (HoP) for multi-view 3D detection to leverage temporal information more effectively (fig.2). This method allows state-of-the-art architectures to perform even better by generating a pseudo BEV feature map of timestamp $t-k$ from its adjacent frames and utilize this feature to predict the object set at timestamp $t-k$. HoP is performed only during training and thus, does not introduce extra overheads during inference. HoP is described as a plug-and-play approach and can be easily incorporated into

state-of-the-art BEV detection frameworks including BEVFormer [6].

As the paper is very recent, the implementation is yet not available and therefore the main contribution of the project will be to implement the proposed HoP method including it to the BEVFormer model.

## REFERENCES


[1] L.-B. Luo, I.-S. Koh, K. y. Min, J. Wang, and J. Chong, "Low-cost implementation of bird's-eye view system for camera-on-vehicle," 02 2010, pp. 311 – 312.

[2] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," 2020.

[3] Y. Zhao, Y. Zhang, Z. Gong, and H. Zhu, "Scene representation in bird's-eye view from surrounding cameras with transformers," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 4510–4518.

[4] L. Peng, Z. Chen, Z. Fu, P. Liang, and E. Cheng, "Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs," 2022.

[5] A. Saha, O. M. Maldonado, C. Russell, and R. Bowden, "Translating images into maps," 2022.

[6] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," 2022.

[7] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," 2020.

[8] Z. Zong, D. Jiang, G. Song, Z. Xue, J. Su, H. Li, and Y. Liu, "Temporal enhanced training of multi-view 3d object detector via historical object prediction," 2023.