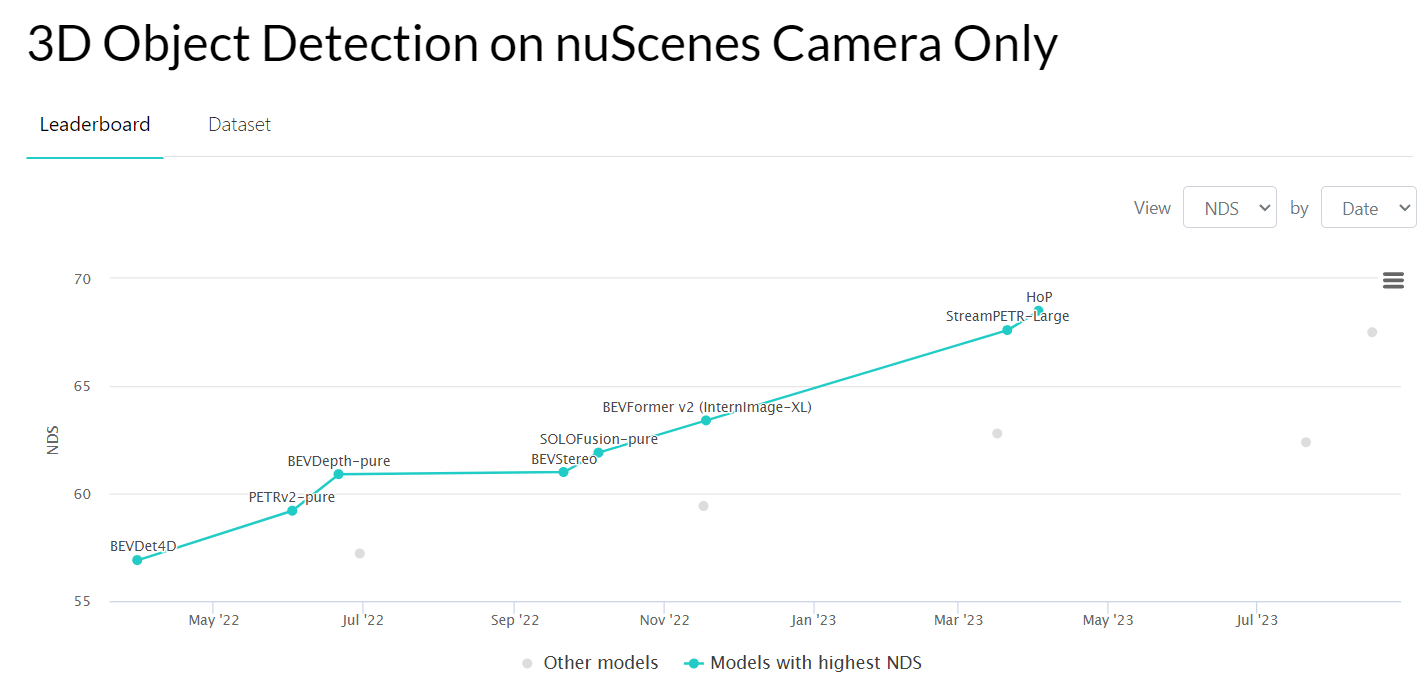
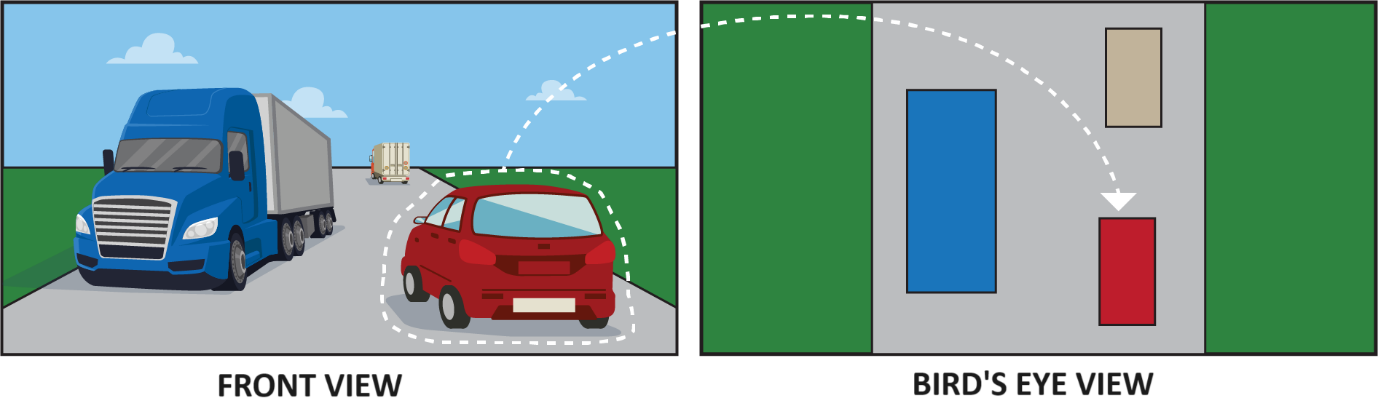
Perception tasks, such as 3D Object Detection and Segmentation are essential for Self-Driving Cars. Post 2021, when Tesla switched to [Hydranets](https://www.thinkautonomous.ai/blog/how-tesla-autopilot-works/), Camera-only solutions are becoming more and more popular. Among Camera-only solutions, Birdseye view (BEV) based approaches have almost been the go-to approach, as seen in the [nuScenes Camera only 3D Object detection leaderboard](https://paperswithcode.com/sota/3d-object-detection-on-nuscenes-camera-only)



What is meant by BEV approach? What is the reason, for it to be adopted by all top performing models? Let’s find out.

## Birdseye View

Bird’s Eye View is nothing, but a top-down view of the environment w.r.t reference frame, mostly the Ego vehicle. It can cover just part of entire FOV, or entire 360degree based on the application.



[Image source](https://multicorewareinc.com/birds-eye-view-a-primer-to-the-paradigm-shift-in-autonomous-robotics/)

## What are the advantages of using BEV?

Birdseye View offers a unified framework to create detailed representations of Ego vehicle’s surroundings with following advantages:

* ***Flexible representation -*** We can use BEV framework to have 2D / 3D / 4D representations of world, in discretized form, with minimal change in architecture.
* ***Modular Fusion -*** We can use the same framework for Multi-Camera Fusion, Lidar-Camera fusion, and even extend it to Spatio-Temporal Fusion.
* ***Efficient* -** We can have a single model to process all sensor data**,** saving on computation power, closer to real-time processing.
* **Rich information –** Boundaries of different objects and infrastructure in top view, is very useful for Path Planning, Mapping applications.

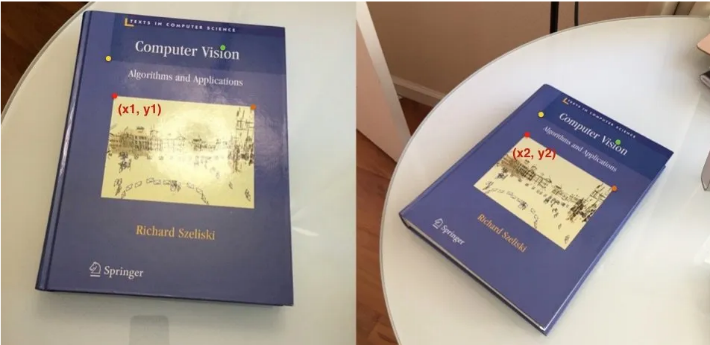
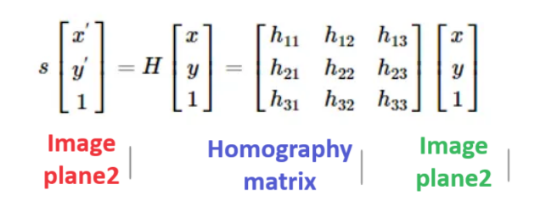
Although BEV has so many advantages, it has its own set of challenges. Cameras are mostly, mounted on ego vehicles, parallel to ground and facing outwards. The images are captured in a Perspective View (as seen in Front View of 1st image), which needs to be transformed to BEV. Let’s explore how it’s done.

## Approaches to Birdseye view

As with any Computer vision task, Birds eye view can be achieved via Traditional image processing as well as Deep Learning. In either of approaches, the input is set of images in the sensor perspective, First, we’ll look at Traditional method.

## Traditional approach

InComputer Vision, homography (H) is a transformation matrix, which when applied to a Plane (in our case, Image), maps to another plane. Using the homography matrix, we can go from one perspective to another, just as shown below.

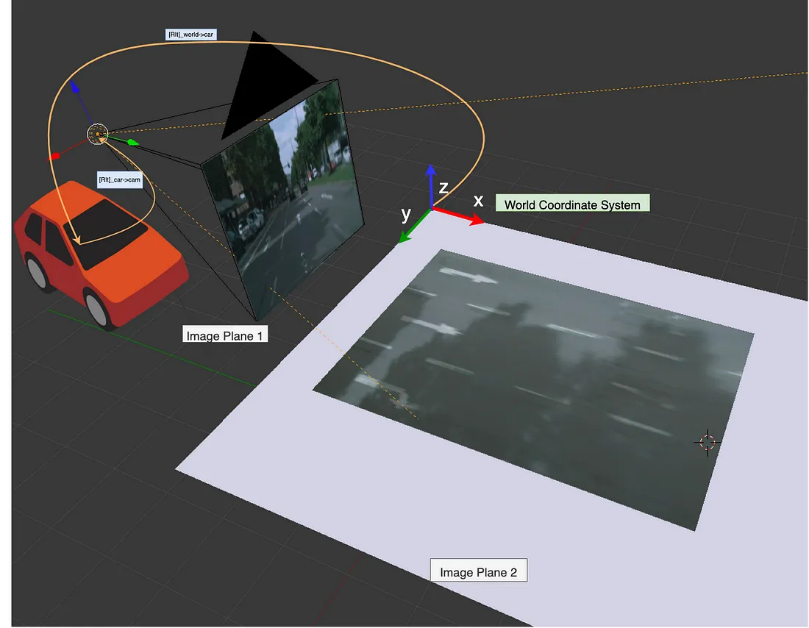


[Image source](https://mattmaulion.medium.com/homography-transform-image-processing-eddbcb8e4ff7#:~:text=Homography%2C%20also%20referred%20to%20as,in%20a%20homogenous%20coordinates%20space.)

In the case of Inverse Perspective Mapping (**IPM**), we want to produce a birds-eye view image of the scene from the front-facing image plane. Assuming world to be flat on a plane, we need to map each pixel on image to a point on 2D plane. To understand better, we define the following coordinate frames:

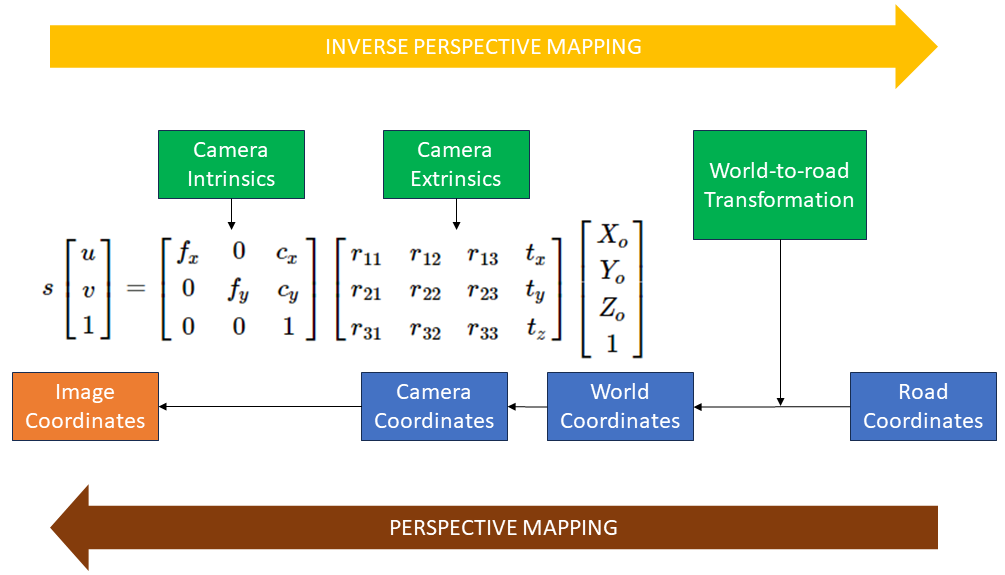
* Image Coordinate frame (in pixels)
* Camera Coordinate frame
* World Coordinate frame

**NOTE:** The world coordinate frame is often discretized (in metres). Hence an approximation on resolution is to be made.



[Image source](https://towardsdatascience.com/a-hands-on-application-of-homography-ipm-18d9e47c152f)

Mathematically, inverse perspective mapping can be represented as follows:



Once we have the Camera intrinsics and extrinsics, IPM procedure can be summarized as follows:

1. Take a point in Road plane (X, Y, Z=0).
2. Apply the projection matrix (comprising of World to road, camera intrinsics and extrinsics) to get image coordinates (in pixels)
3. Find the pixels, closest to image coordinates and project to BEV plane.
4. Repeat steps (1-3) for all points in discretized road plane. Additionally, we need to perform some form of interpolation to prevent holes in the output.

Sounds straight forward. It might, but IPM make some assumptions about the environment, which limits its real-world applications.

### Drawbacks of IPM

1. Flat road surface assumption. Any deviation would produce artefacts / distortions
2. Zero height object assumption. Objects with height would appear distorted / warped in BEV output

Following image highlights the above issues



[Image source](https://towardsdatascience.com/a-hands-on-application-of-homography-ipm-18d9e47c152f)

Even with the above limitations, ***IPM is still commonly useful in specific environments with static objects, mostly for Lane detection algorithms.***

## Deep Learning based approach

The Deep learning approach involves using Neural Networks to solve the aforesaid issues, by learning from Ground truth data. ***Most of DL approaches combine the BEV transformation with either Object detection and /or Segmentation.*** This is different from the traditional approach, where we were trying to get the semantic texture information in BEV.

1. UNetXST architecture
   1. Spatial Transformers
   2. Simulated data usage
   3. BEV segmentation from 4 cameras notebook
   4. <https://www.youtube.com/watch?v=TzXuwt56a0E>
2. SOTA Approaches to BEV
   1. BEVFormer Preview
      1. Deformable DETR module
      2. BEVFormer architecture
      3. Temporal Extension possibilities

## References

**Dummy**

Such Perception tasks, require fusion of data from different sensors–Monocular, Stereo Cameras, Lidars, Radars etc, for redundancy purposes. As highlighted in [previous post](https://www.thinkautonomous.ai/blog/9-types-of-sensor-fusion-algorithms/), there are multiple ways to fuse sensor data, each with its advantages and disadvantages.