A Forecast-Then-Detect Pipeline in Autonomous Driving

Xinshuo Weng, Kris Kitani
Robotics Institute, Carnegie Mellon University

Presented at the 3rd Workshop on Autonomous Navigation in Unconstrained Environments
June 19, 2021
Perception and prediction are important components in the autonomous driving stack
Standard Perception and Prediction Pipeline

Sensor Data → 3D Object Detection → 3D Multi-Object Tracking → Trajectory Forecasting

Perception → Prediction
Standard Perception and Prediction Pipeline

Sensor Data

3D Object Detection

3D Multi-Object Tracking

Trajectory Forecasting

LiDAR

RGB
Standard Perception and Prediction Pipeline

Sensor Data ➔ 3D Object Detection ➔ 3D Multi-Object Tracking ➔ Trajectory Forecasting

Detection results
Standard Perception and Prediction Pipeline

1. Sensor Data
2. 3D Object Detection
3. 3D Multi-Object Tracking
4. Trajectory Forecasting

Tracking results
Standard Perception and Prediction Pipeline

1. Sensor Data
2. 3D Object Detection
3. 3D Multi-Object Tracking
4. Trajectory Forecasting

Forecasting results
Any possible improvement at the pipeline level?

- Sensor Data
  - 3D Object Detection
    - 3D Multi-Object Tracking
      - Trajectory Forecasting

Lots of progress on improving each individual module

The pipeline stays the same!
Standard Perception and Prediction Pipeline

Sensor Data

3D Object Detection

3D Multi-Object Tracking

Trajectory Forecasting

Requires predefined object categories

No consensus on how to propagate uncertainty

Sequential processes propagate errors (rarely evaluated in papers)

Limited evaluation based on down-stream tasks like planning and control

Is this really the best place to perform prediction?
Standard Perception and Prediction Pipeline

Can we do prediction here?

- Sensor Data
- 3D Object Detection
- 3D Multi-Object Tracking
- Trajectory Forecasting
The answer is yes!

A forecast-then-detect pipeline that inverts the order of forecasting
SPF^2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting

- Detect-then-forecast pipeline:
  - Detection -> MOT -> Trajectory Forecasting

- Forecast-then-detect pipeline
  - Sequential Pointcloud Forecasting -> Detection -> MOT

- Differences
  - Invert the order of forecasting
  - Forecast at the sensor level (not object positions)

Move forecasting upfront
SPF$^2$ Provides Stable Object Trajectory Prediction

- Predicted point clouds preserve object information
- Equivalent to results obtained from standard pipeline, i.e., object trajectories

Weng et al. Inverting the Pose Forecasting Pipeline with SPF2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting. CoRL 2020
Performance Scaling with More Point Cloud Data

- Advantages?
  - Does not require human annotation for forecasting
  - Point cloud data is prevalent nowadays
- The key is this new task -- Sequential Pointcloud Forecasting (SPF)
Performance Scaling with More Point Cloud Data

- CD: Chamfer distance (lower is better)
- EMD: Earth Mover’s Distance (lower is better)
SPF: Sequential Pointcloud Forecasting

- Goal: a sequence of past clouds -> a sequence of future clouds
- Predict the entire scene, including background
- Deal with large-scale points (1.5M) rather than 1k points

Comparison to trajectory forecasting
Comparison to point cloud generation

Weng et al. Inverting the Pose Forecasting Pipeline with SPF2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting. CoRL 2020
Problem Formulation

• Inputs: a sequence of point clouds
  $\mathcal{P} = \{P_{-M+1}, \ldots, P_{-1}, P_0\}$
  $P_t = \{(x, y, z)_j\}_{j=1}^{K_t}$, where $t \in [-M + 1, \ldots, 0]$, $j \in [1, \ldots, K_t]$
  $K_t > 100,000$, varies across frames

• Outputs: a sequence of point clouds
  $\mathcal{F} = \{F_1, F_2, \ldots, F_N\}$
  $F_t = \{(x, y, z)_j\}_{j=1}^{K_t}$, where $t \in [1, 2, \ldots, N]$, $j \in [1, \ldots, K_t]$

• Goal
  • Learn $\mathcal{F} = f(\mathcal{P})$
SPFNet

- How to learn representations from large-scale point cloud sequences?
- Baseline: FC-LSTM autoencoder model
- Four modules
  - Frame-wise point cloud encoder
  - FC-LSTM for temporal modeling
  - Frame-wise point cloud decoder
  - Losses: chamfer distance

Weng et al. Inverting the Pose Forecasting Pipeline with SPF2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting. CoRL 2020
Predicted Point Clouds Preserves Objects

Weng et al. Inverting the Pose Forecasting Pipeline with SPF2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting. CoRL 2020
Detection on Predicted Point Clouds

- Green: detected 3D boxes
- Yellow: GT 3D boxes
- Detections mostly match with GT on predicted point clouds from SPFNet

Weng et al. Inverting the Pose Forecasting Pipeline with SPF2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting. CoRL 2020
Tracking on Predicted Point Clouds

- Colored boxes: GT boxes or tracked boxes from the SPF² pipeline
- GT Objects can be mostly tracked
Tracking on Predicted Point Clouds

- Color represents the object identity
- Predicted point clouds from SPFNet preserves objects
What Are the Limitations?

- Predicted point clouds and objects do not look real, losing details
- What about small objects such as pedestrians?
Learning Fine-Grained Point Cloud Prediction with Pyramid ConvLSTM
Fine-Grained Point Cloud Prediction with Pyramid ConvLSTM

• Issues
  • Missing details in predicted point clouds and objects
  • Cannot preserve small objects such as pedestrians
• Hypothesis: hidden representation is compressed too much
Preserve Details with Finer Representation and Stronger Backbone

- Straightforward solution
  - Increase details in hidden features: 1D feature -> 2D heatmap
  - Replace FC-LSTM with C-LSTM
  - Upgrade encoder-decoder: 4 Conv+BN+LeakyReLU -> DarkNet53

**SPFNet with ConvLSTM**

**DarkNet53 encoder-decoder (RangeNet++)**

Milioto et al. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. IROS 2019
Preserve Details with Finer Representation and Stronger Backbone

- Training does not converge well. What is the problem?
- Feature synchronization issue
Pyramid LSTM for Feature Synchronization

- Spatial Hierarchical LSTM
  - With each hierarchy operates on different feature scales

- Training converges stably
- 3 – 8x improvement on different metrics over SPFNet

Weng et al. Learning Fine-Grained Point Cloud Prediction with Pyramid ConvLSTM. 2021
SPFNet vs. Pyramid LSTM: More Details are Preserved

Red: Points with large error
Cyan: Points with small error

SPFNet

Pyramid LSTM
SPFNet vs. Pyramid LSTM: Pedestrians can be Predicted

Red: Points with large error
Cyan: Points with small error
Take Home Message

• Innovation is not only possible to happen at the modular level but also at the pipeline level

• Forecast-then-detect pipeline naturally fits to the real-world trajectory forecasting setting

• Simple SPFNet baseline can preserve information for large objects but loses details

• Pyramid LSTM increases details on global structure and small objects
AIONDrive Point Cloud Prediction Challenge

- The first challenge on Point Cloud Prediction
- Coming soon at www.aiodrive.org

Weng et al. All-In-One Drive: A Comprehensive Perception Dataset with High-Density Long-Range Point Clouds. 2021
A Forecast-Then-Detect Pipeline in Autonomous Driving

Xinshuo Weng, Kris Kitani
Robotics Institute, Carnegie Mellon University

Presented at the 3rd Workshop on Autonomous Navigation in Unconstrained Environments
June 19, 2021