Augmenting Visual AI Through Radar and Camera Fusion

Sébastien Taylor
V.P. of Research & Development
Au-Zone Technologies

Outline

- Introduction
- Camera Benefits and Limitations
- How Radar Data Looks
- Ground Truth Representations
- Camera and Radar Fusion
- Summary

Camera Benefits and Limitations

Camera Benefits and Limitations

- **Benefits**
  - High-resolution colour and texture information
  - High refresh rates such as 60 FPS and beyond
  - Easy for people to interpret camera data

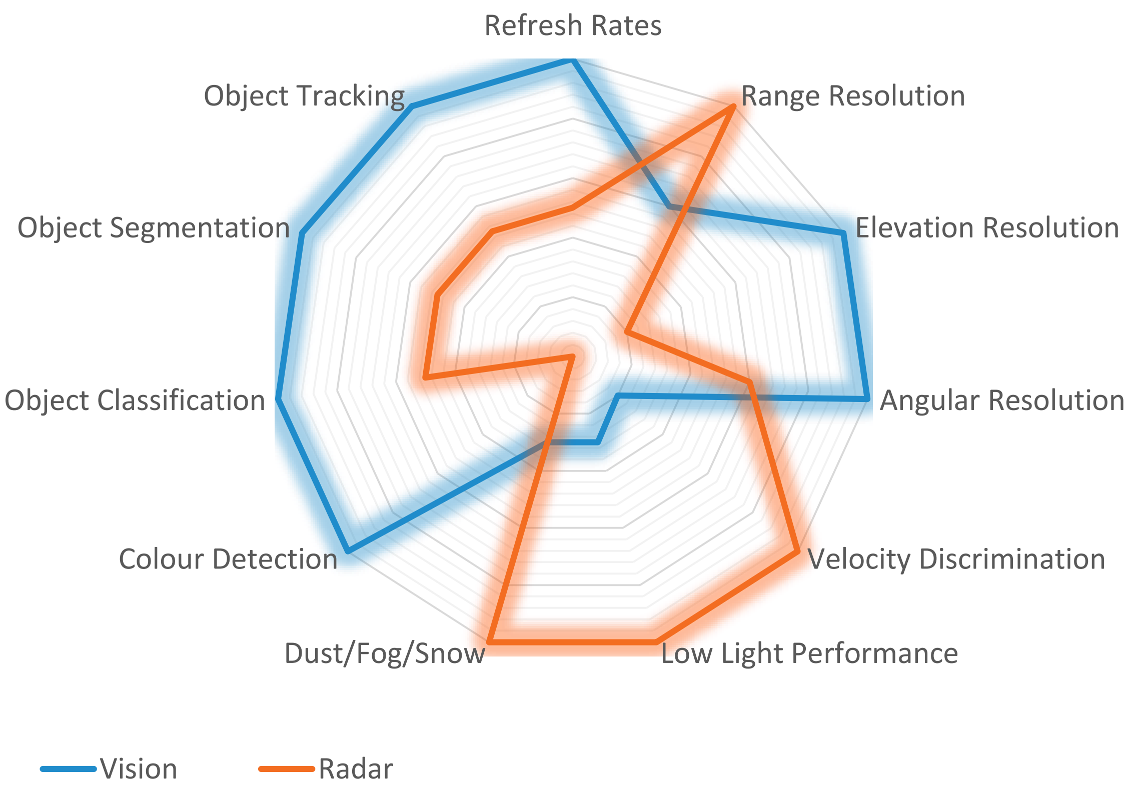
- **Limitations**
  - Requires good lighting conditions
  - Susceptible to adverse weather
  - Susceptible to occlusions such as debris on the lens
  - Some use cases are limited because of privacy

Radar Benefits and Limitations

- **Benefits**
  - Not affected by weather conditions
  - Not affected by lighting conditions
  - Provides accurate range and speed information
  - Can address privacy concerns

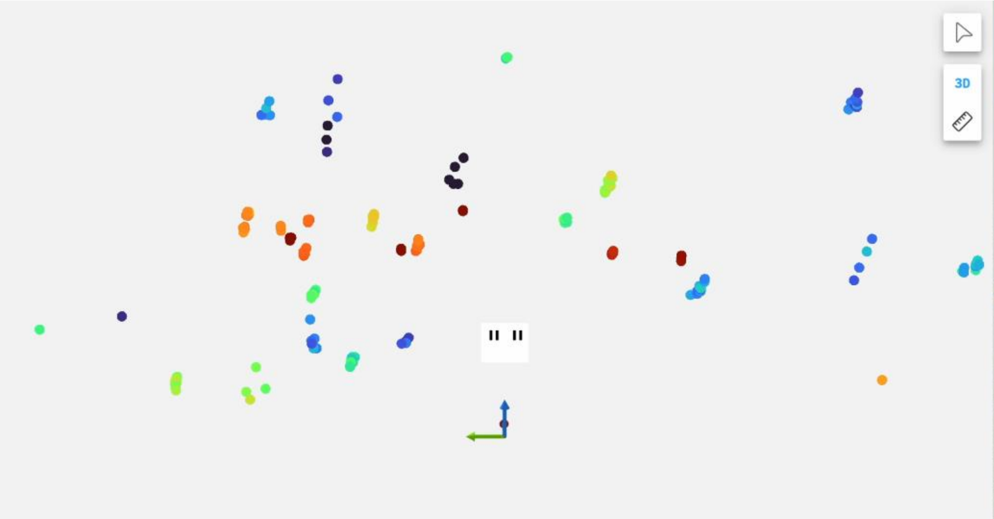
- **Limitations**
  - Low resolution and refresh
  - No colour or texture information
    - Hard to classify objects
  - Difficult to interpret data
  - Emerging technology in AI space, lots still to be discovered and understood!

How Radar Compares to Vision



How Radar Data Looks

How Radar Point Clouds Look

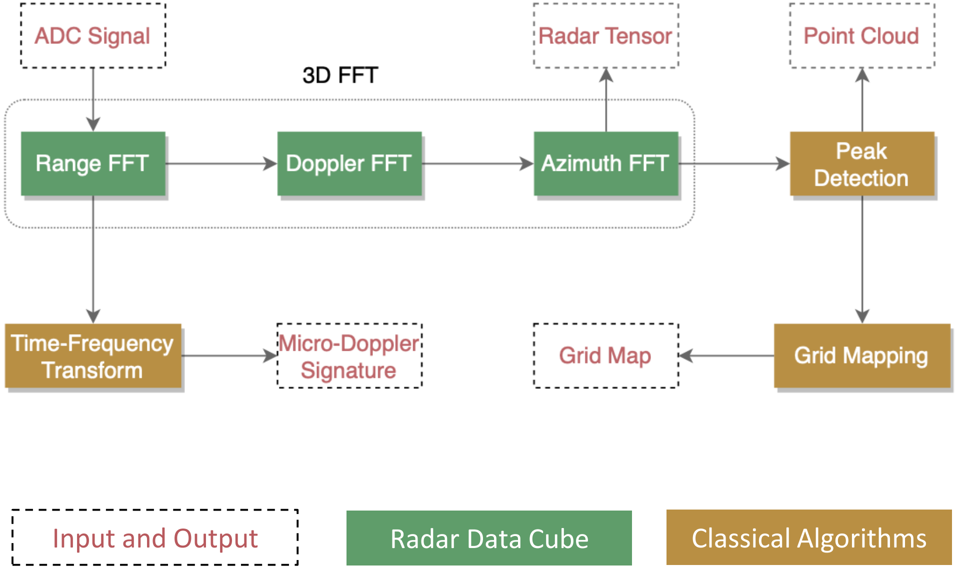
Radar Point Cloud	Camera Reference
	

Radar Point Clouds

- Limited size of input features for the model
  - Power, noise, radar cross-section, speed
  - The speed feature is helpful for clustering but less so for classification
- Compared with LiDAR point clouds, radar point clouds are very sparse
  - As few as a dozen points in some cases
  - Depends on the antenna configuration
- People have shown good results with point clouds, typically when using high-end radar modules
  - We will focus processing the lower-level radar data cube

What Is a Radar Data Cube?

- The radar data cube is the combination of the various FFT stages applied to the radar ADC input
  - Range, Doppler, Azimuth, and Elevation
- The radar cube is further processed using traditional algorithms to generate various outputs
  - Point cloud, micro-doppler, grid map
- The radar data cube is interesting because we want to avoid the information loss from classical processing algorithms

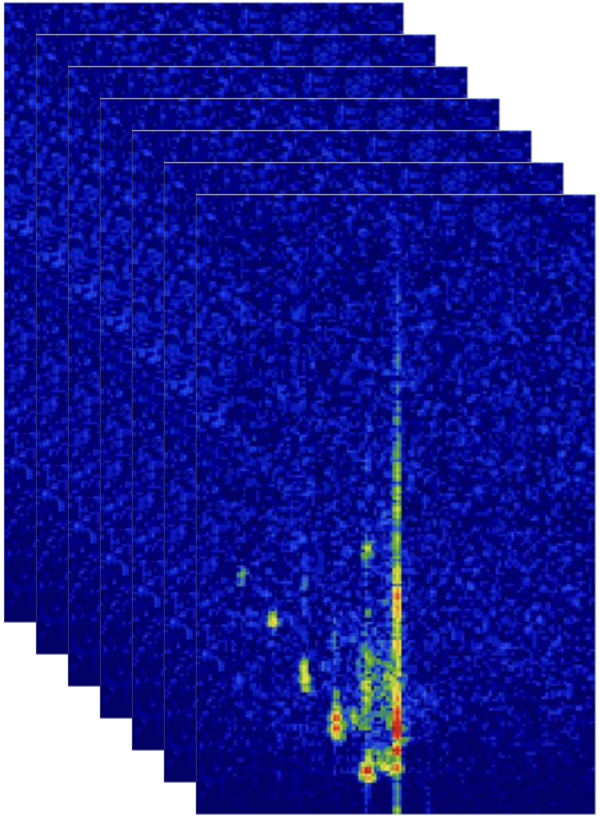


Range Doppler Data Cube

- Harder to understand compared to images

- Cube is dense but data is sparse

- Richer data compared to heavily filtered point cloud data



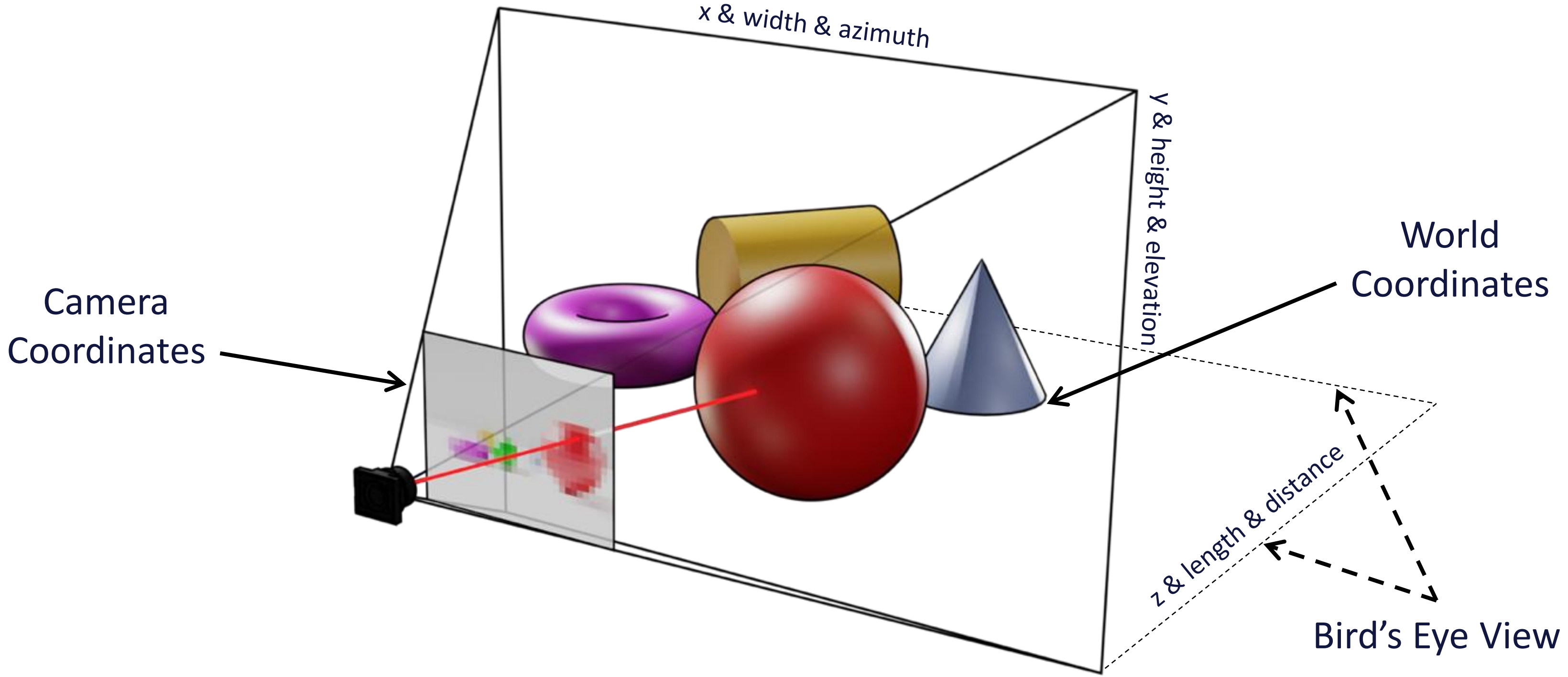
Radar Data Cube Variations

- Data cube availability and format varies between radar modules
  - *Not all modules support data cube output!*

- The data cube format also varies greatly
  - R, RA, RD, RAD, RAED, etc…
  - **R**ange, **A**zimuth, **E**levation, and **D**oppler

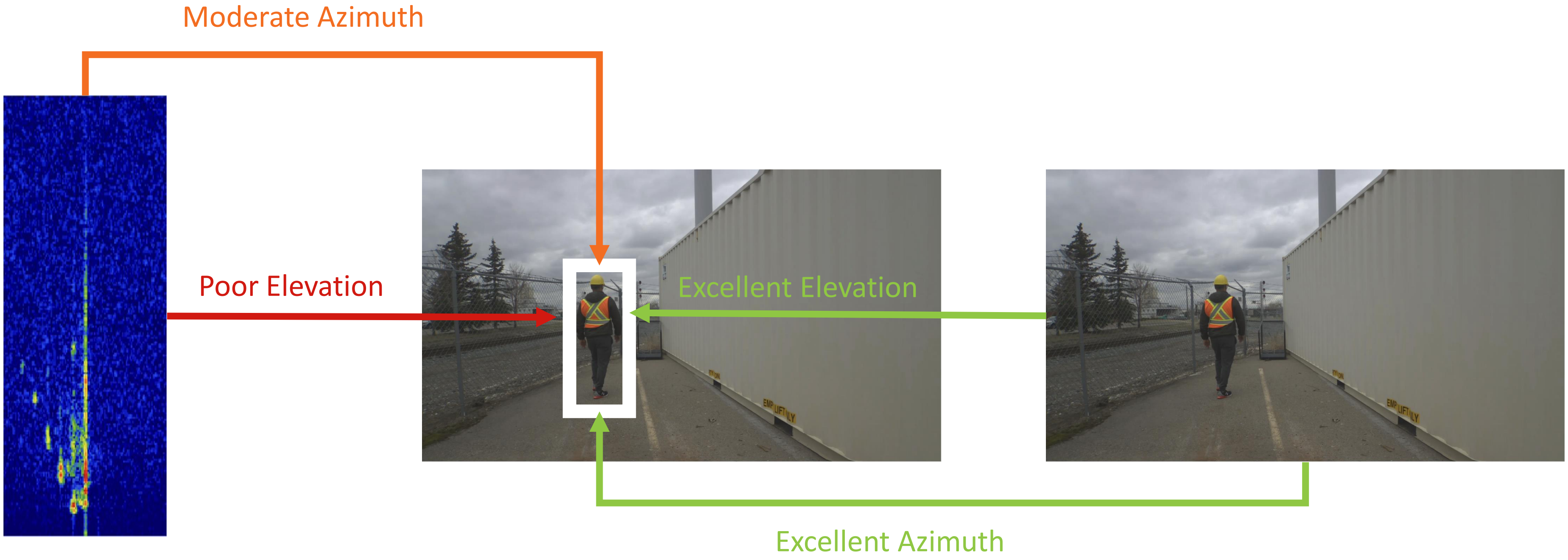
Ground Truth Representations

Ground Truth Representation Terminology

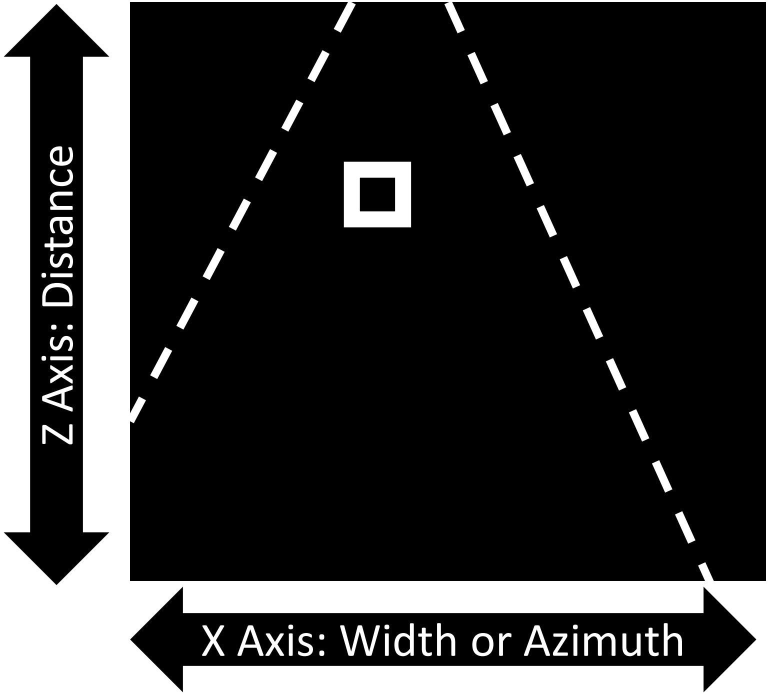


Ground Truth – 2D Boxes in Camera Coordinates

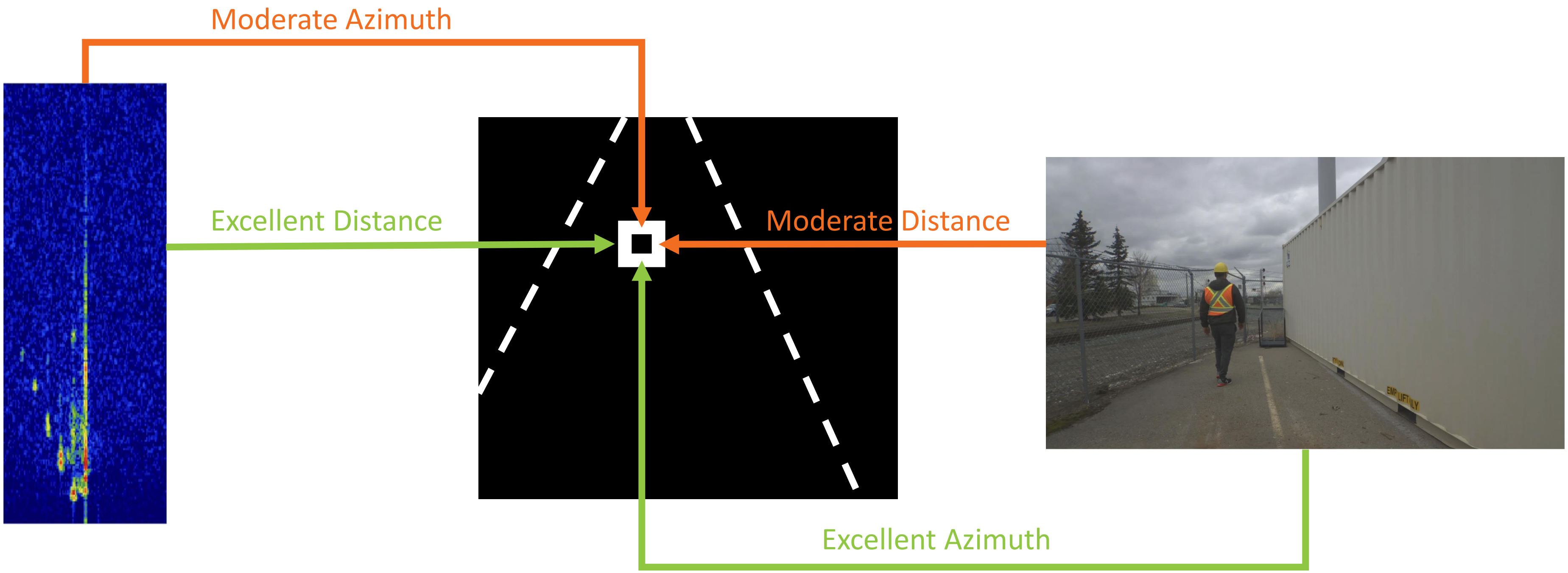
Ground Truth – 2D Boxes, Poor Synergy



Ground Truth – Bird's Eye View of Global Coordinates



Ground Truth – Bird's Eye View, Good Synergy



Camera and Radar Fusion

Flavours of Fusion

- **Object Level Fusion**
  - Independent models for each sensor
  - Fusion by correlation of detected objects
- **Input Level Fusion**
  - A single model with a common backbone
  - Fusion requires adapting one sensor representation to the other
- **Feature Level Fusion**
  - A single model with split backbones
  - One backbone per-sensor to avoid adapting representations
  - Model unified, each backbone is trained to learn directly from each sensor

Object Level Fusion

- Arguably the easiest fusion
- Each model is totally independent
- Fusion through correlation
- Misses out on performance and learning benefits of a unified model

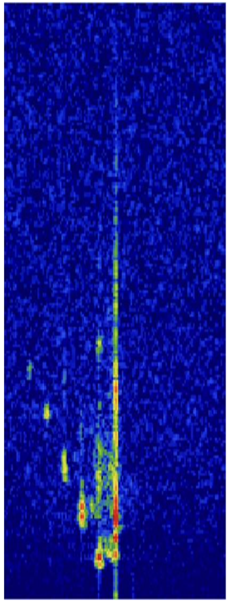
Input Level Fusion

- Adapt input representation of one sensor to match the other
  - After adaptation input tensors can be concatenated together

- Provides the highest level of model reuse as the two sensors are totally integrated

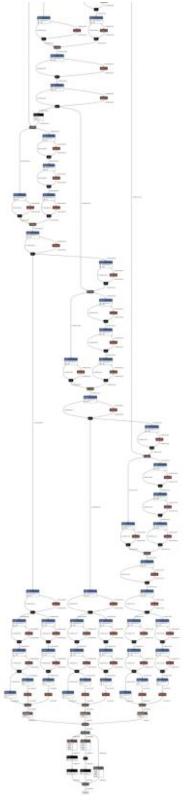
- Sensor representations are very different which means an engineered adaptation will not be optimal

Input Level Fusion – Incompatible Inputs



Seq, Rx, Range, Doppler

Ex: 2, 4, 200, 128



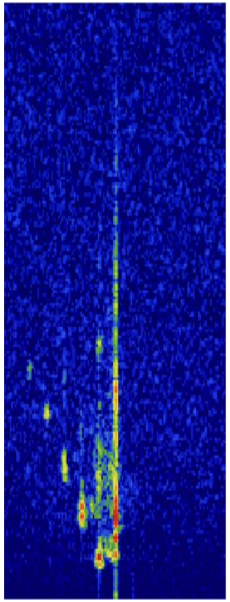
Height, Width, Channels (RGB)

Ex: 540, 960, 3

Feature Level Fusion

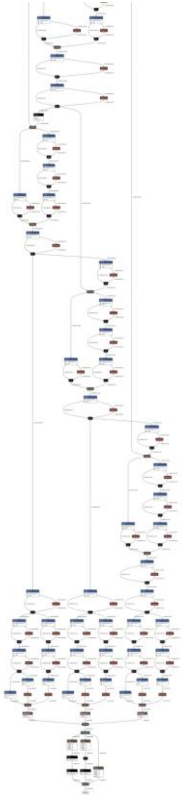
- Unified model but with split backbones
  - Fusion is through concatenation of compatible layers
- Each backbone learns directly from the sensor’s native representation
- Highest level of learning without specialized adaptations
- Backbones can be specialized for each input representation

Feature Level Fusion – Compatible Layers



Anchor, Box, Object, Class

Ex: 4032, 4, 1, 4



Anchor, Box, Object, Class

Ex: 4032, 4, 1, 4

Summary

Summary

- Cameras have limitations in adverse lighting and environmental conditions
- Radar systems can overcome camera limitations but impose their own limitations
- Fusing camera and radar data can provide a best of both worlds solution
- Fusion requires a common ground truth so that the two baseline models are speaking the same language
- Fusion also requires a common internal representation to perform the fusion
- Once ground truth and a common internal layer are provided the actual fusion is straight forward, simple concatenation
- Radar datasets are very sensor specific unlike camera datasets, challenging AI problem!

Resources

- https://github.com/radarsimx/radarsimpy
  - Radar simulation environment for Python
- https://www.radartutorial.eu/index.en.html
  - Excellent resource covering diverse radar topics
- https://github.com/Radar-Camera-Fusion/Awesome-Radar-Camera-Fusion
  - Links to many more camera radar fusion resources

Thank you,

Q & A