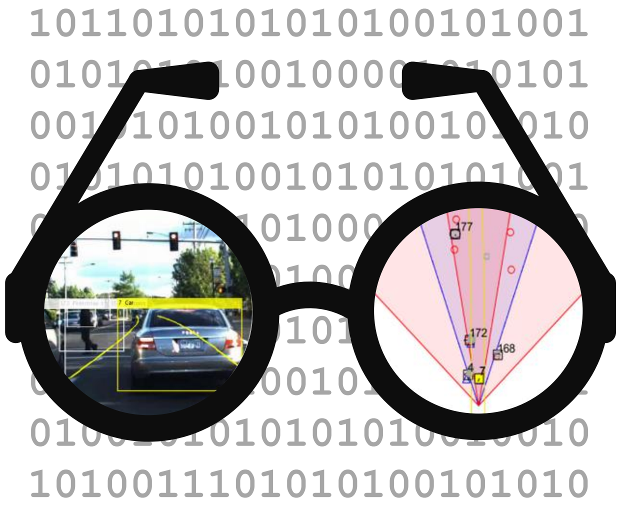
MathWorks®

MATLAB EXPO 2017

自动驾驶：设计和验证感知系统

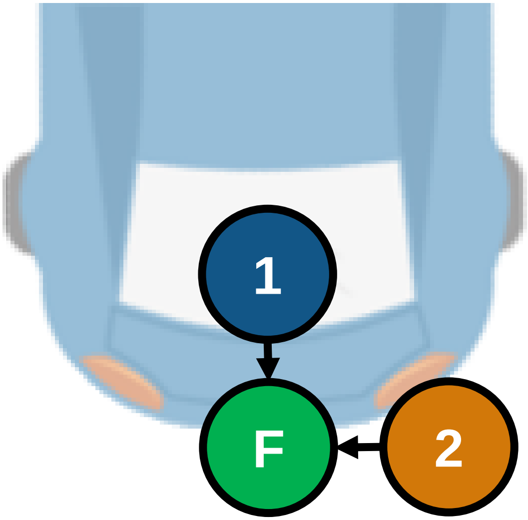
陈小挺　高级应用工程师，MathWorks 中国

自动驾驶工程师经常遇到的问题：



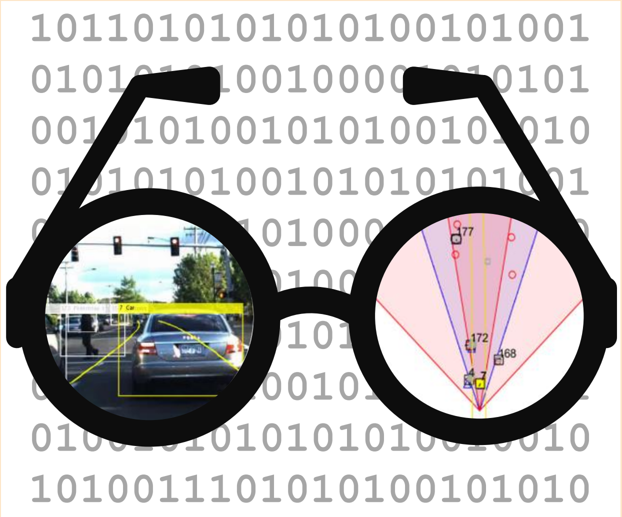
我怎样可视化
车辆的数据?

我怎样检测图
像中的目标?

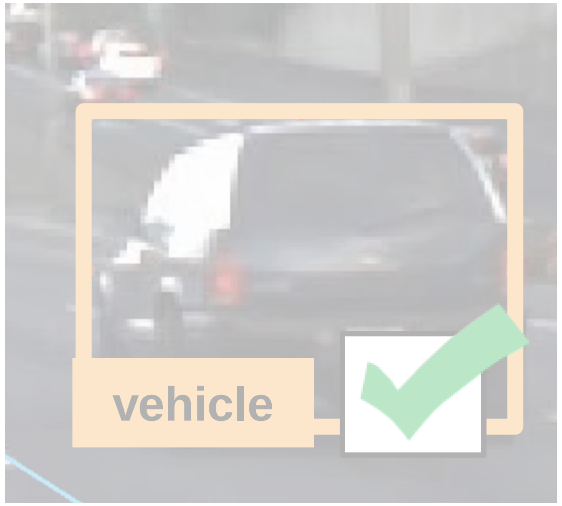


我怎样融合
多个检测结果?

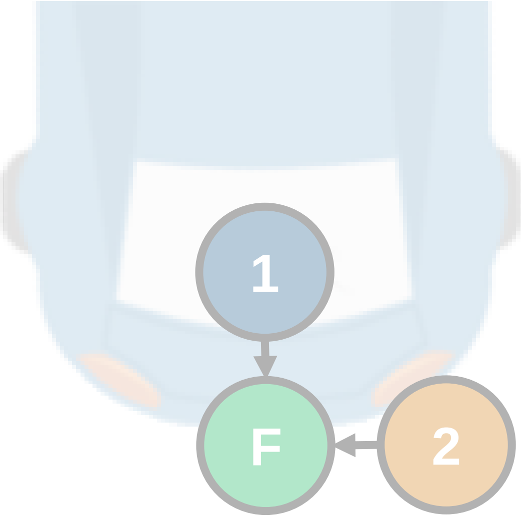
自动驾驶工程师经常遇到的问题：



我怎样可视化车辆的数据?



我怎样检测图像中的目标?



我怎样融合多个检测结果?

自动驾驶中常使用的传感器

摄像头

基于雷达的目标检测

基于视觉的目标检测

激光雷达

车道检测

惯性测量单元

自动驾驶中使用的传感器数据的例子

摄像头 (640 x 480 x 3)

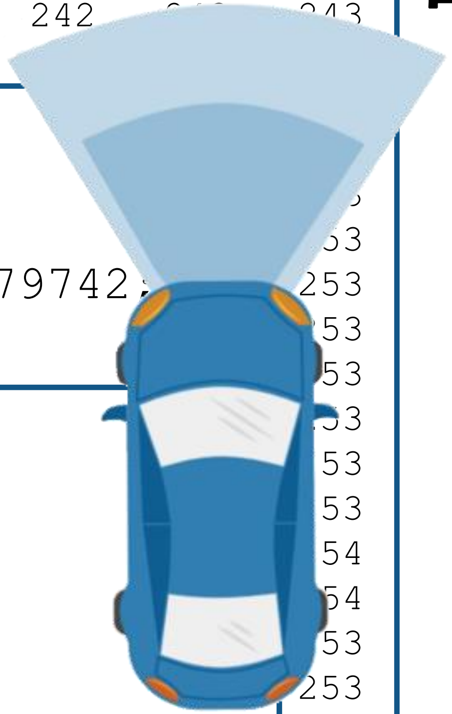
```
239    239    237    238    241    241    241    242
252    252    251    252    252    253    253
```

视觉检测

```
SensorID        = 1;
Timestamp       = 1461634696379742
NumDetections   = 6;
Dete
  Tr
  Cl
  Po
  Ve
  Si
Detec
  Tr
  Cl
  Po
  Ve
  Si
```

车道检测

```
Left
  IsValid:         1
  Confidence:      3
  BoundaryType:    3
  Offset:          1.68
  HeadingAngle:    0.002
  Curvature:       0.000
Right
  IsValid:         1
  Confidence:      3
```



雷达检测

```
SensorID      = 2;
Timestamp     = 1461634696407521;
NumDetections = 23;
Detection
  TrackID
  TrackSt
  Positio
  Velocit
  Amplitu
Detection
  TrackID
  TrackSt
```

激光雷达 (47197 x 3)

```
-12.2911   1.4790   -0.5
-14.8852   1.7755   -0.6
-18.8020   2.2231   -0.7
-25.7033   3.0119   -0.9
 -0.0632   0.0815    1.2
 -0.0978   0.0855    1.2
 -0.2814   0.1064    1.2
                      1.2
                      1.2
                      1.2
                      1.2
                     -0.6
                     -0.7
```

惯性测量单元

```
Timestamp:   1461634696379742
Velocity:    9.2795
YawRate:     0.0040
```

传感器数据可视化

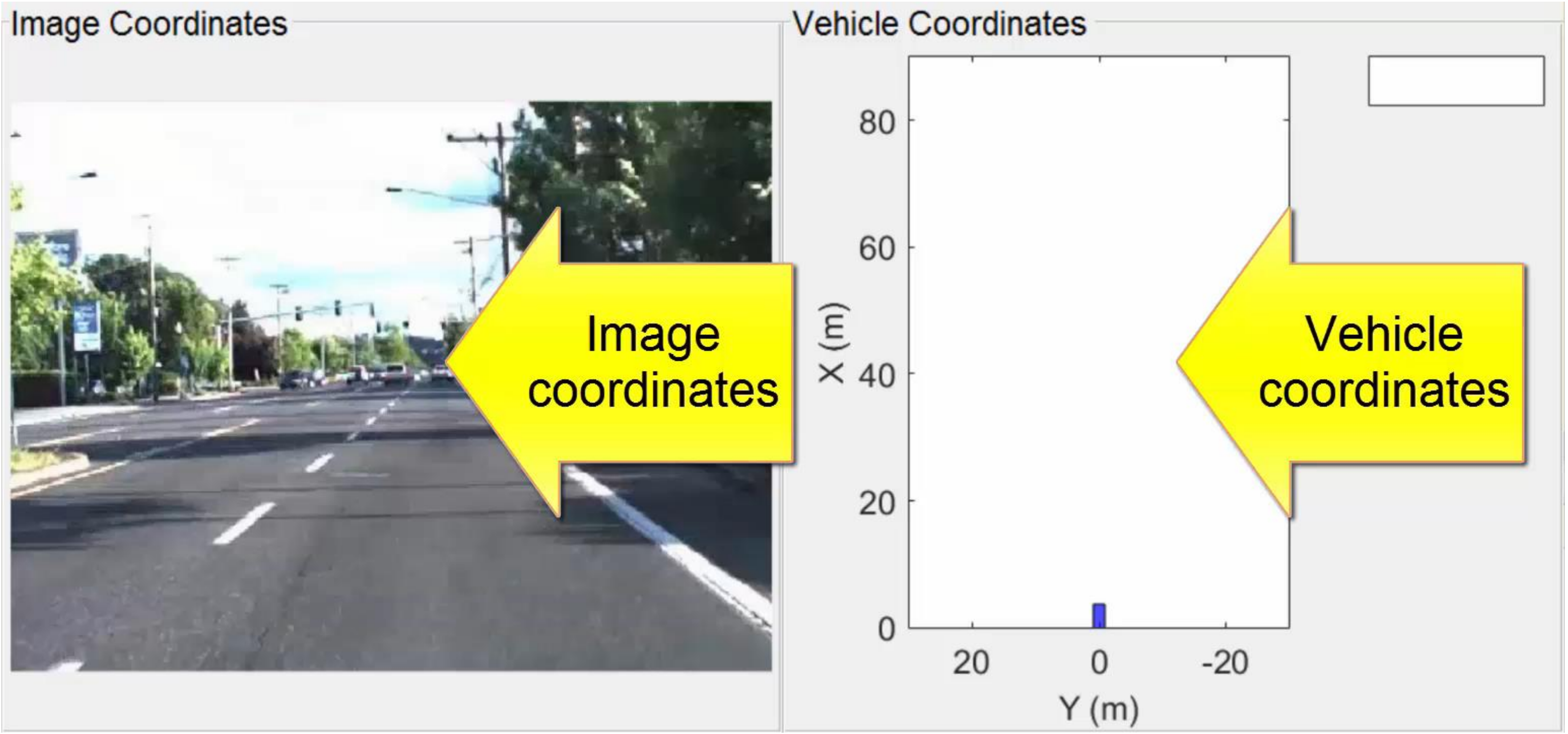
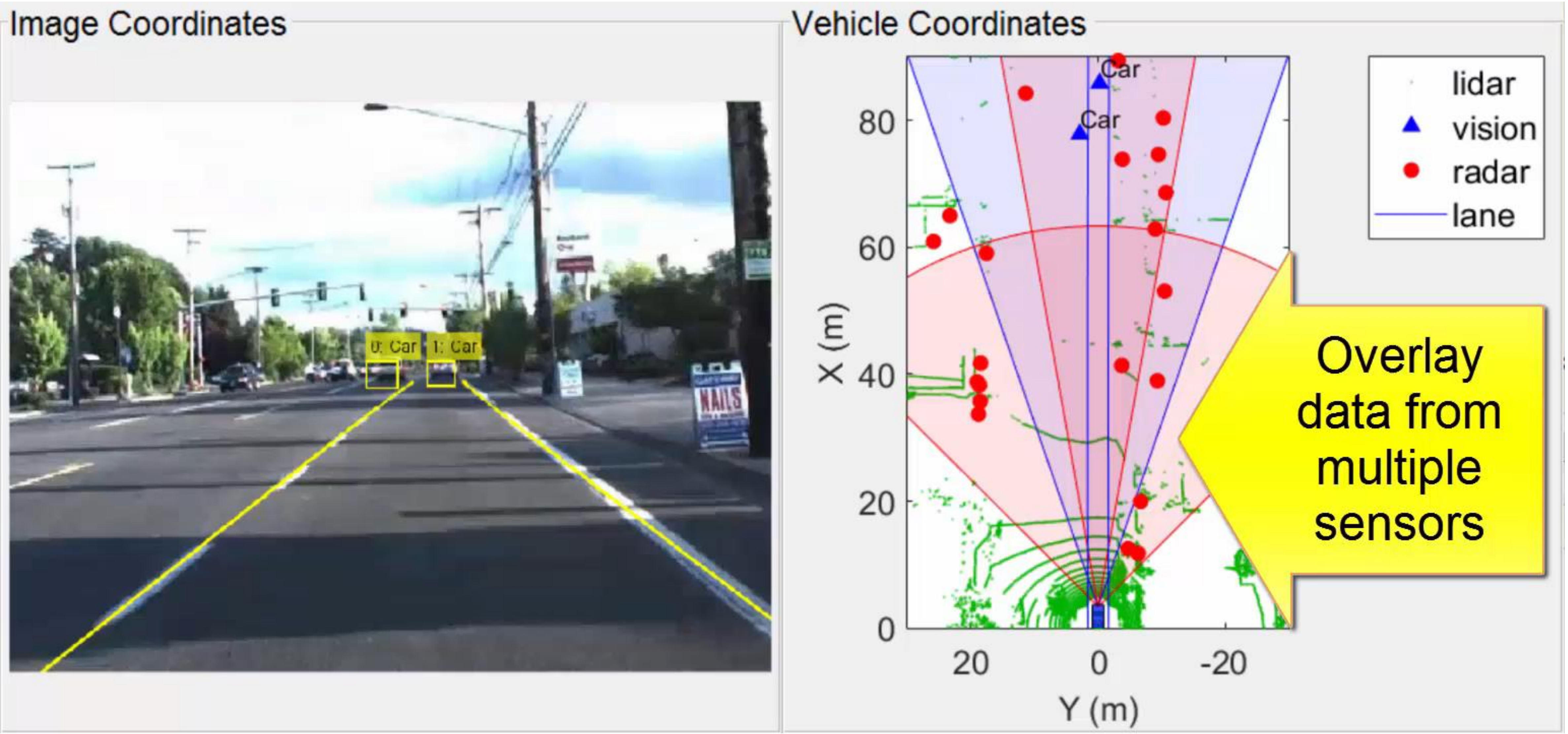


Image Coordinates
Vehicle Coordinates
Image coordinates
Vehicle coordinates
80
60
40
20
0
X (m)
20
0
-20
Y (m)

传感器检测目标差异可视化



探索已记录的车辆数据

- 导入 **摄像头数据** 和相应的 **单摄像头参数**

```
>> video = VideoReader('01_city_c2s_fcw_10s.mp4')
>> load('FCWDemoMonoCameraSensor.mat', 'sensor')
```

- 导入 **传感器检测数据** 和相应的 **参数**

```
>> load('01_city_c2s_fcw_10s_sensor.mat', 'vision','lane','radar')
>> load('SensorConfigurationData.mat', 'sensorParams')
```

- 导入 **激光雷达点云数据**

```
>> load('01_city_c2s_fcw_10s_Lidar.mat', 'LidarPointCloud')
```

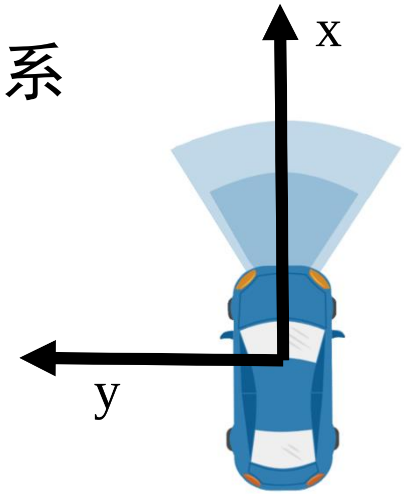
在图像坐标系中可视化

```
%% Specify time to inspect
currentTime = 6.55;
video.CurrentTime = currentTime;

%% Extract video frame
frame = video.readFrame;

%% Plot image coordinates
ax1 = axes(...
    'Position',[0.02 0 0.55 1]);
im = imshow(frame,...
    'Parent',ax1);
```

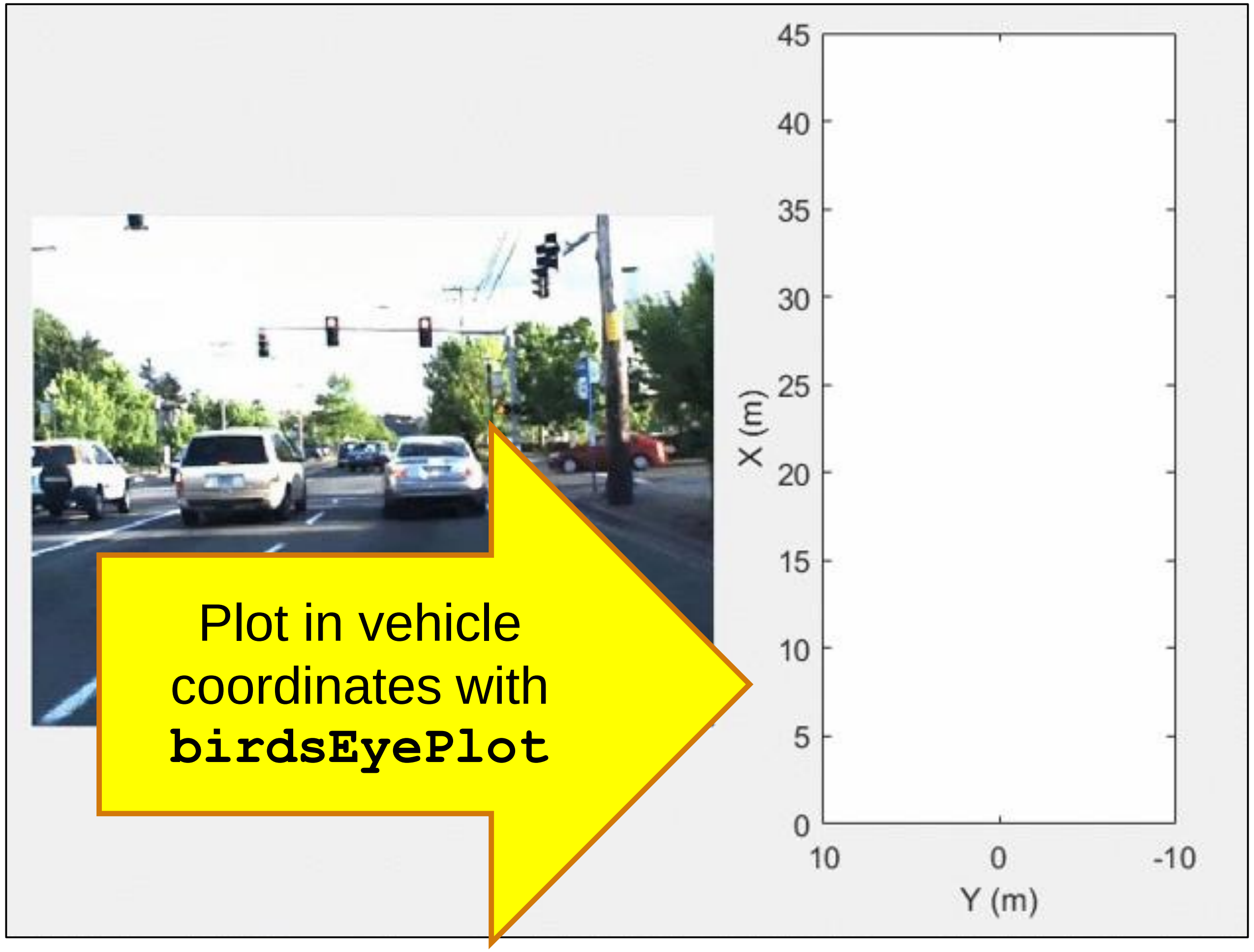
Plot in image coordinates using "classic" video and image functions like **imshow**

在车辆坐标系中可视化

- ISO 8855 车辆坐标系
  - 前向为正x
  - 左向为正y

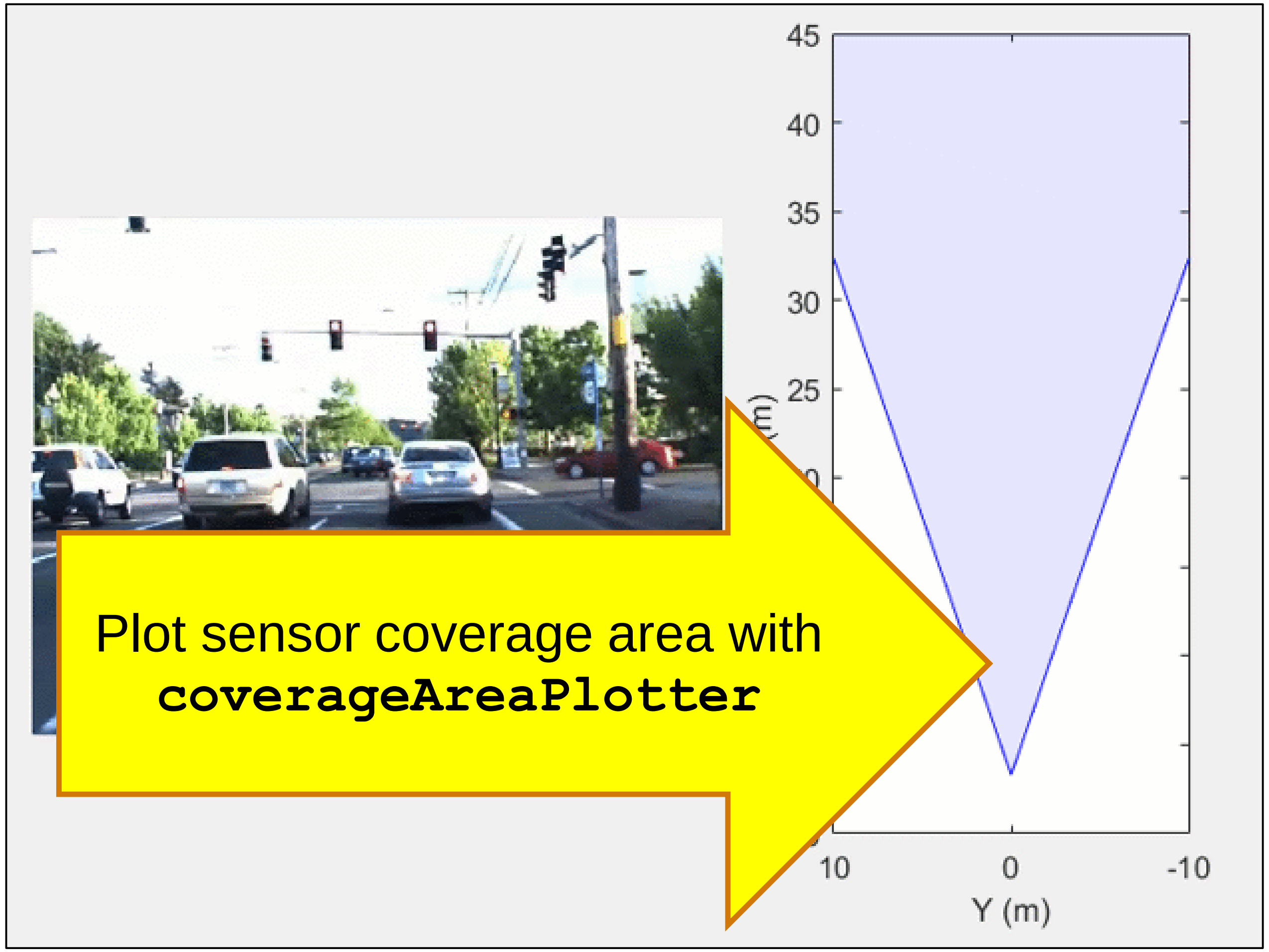



```
%% Plot in vehicle coordinates
ax2 = axes(...
    'Position',[0.6 0.12 0.4 0.85]);
bep = birdsEyePlot(...
    'Parent',ax2,...
    'Xlimits',[0 45],...
    'Ylimits',[-10 10]);
legend('off');
```



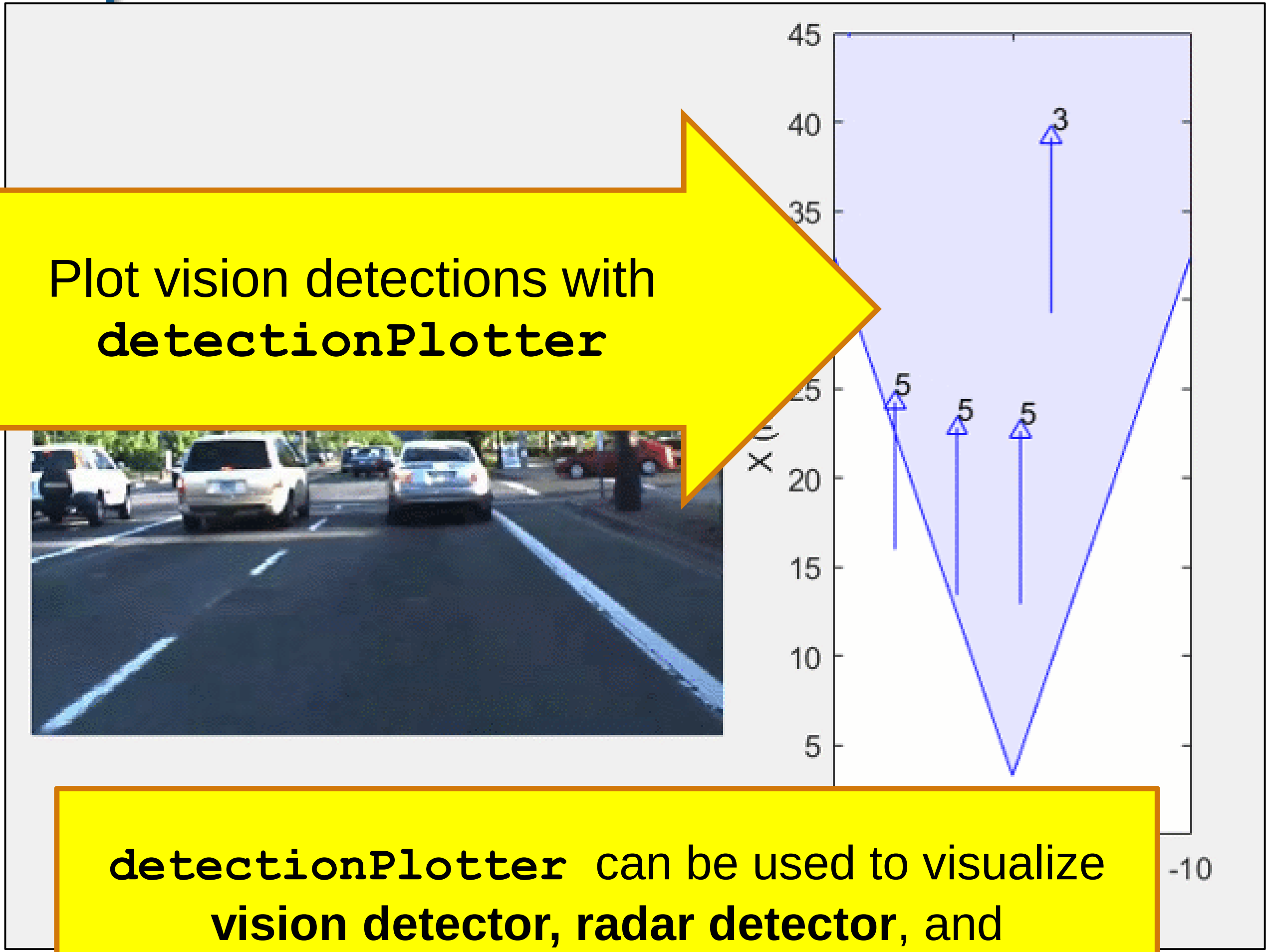
覆盖区域可视化(车辆坐标系)

```
%% Create coverage area plotter
covPlot = coverageAreaPlotter(bep,...
    'FaceColor','blue',...
    'EdgeColor','blue');

%% Update coverage area plotter
plotCoverageArea(covPlot,...
    [sensorParams(1).X ...   % Position x
     sensorParams(1).Y],... % Position y
    sensorParams(1).Range,...
    sensorParams(1).YawAngle,...
    sensorParams(1).FoV(1)) % Field of view
```



检测目标可视化 (车辆坐标轴)

```
%% Create detection plotter
detPlot = detectionPlotter(bep, ...
    'MarkerEdgeColor','blue',...
    'Marker','^');

%% Update detection plotter
n = round(currentTime/0.05);
numDets = vision(n).numObjects;
pos = zeros(numDets,3);
vel = zeros(numDets,3);
labels = repmat({''},numDets,1);
for k = 1:numDets
    pos(k,:) = vision(n).object(k).position;
    vel(k,:) = vision(n).object(k).velocity;
    labels{k} = num2str(...
      vision(n).object(k).classification);
end
plotDetection(detPlot,pos,vel,labels);
```

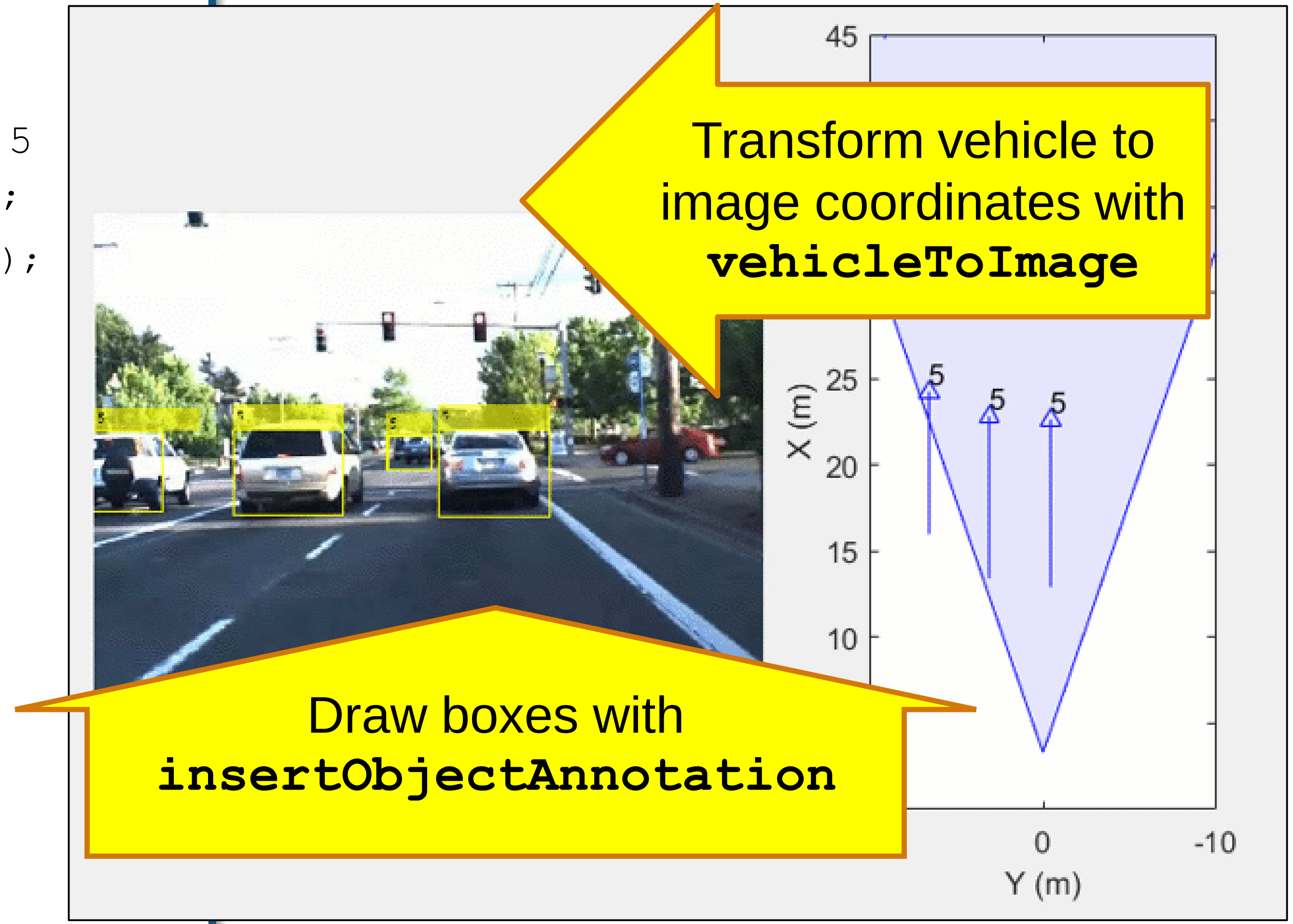


Plot vision detections with **`detectionPlotter`**

`detectionPlotter` can be used to visualize **vision detector, radar detector**, and **lidar point cloud**

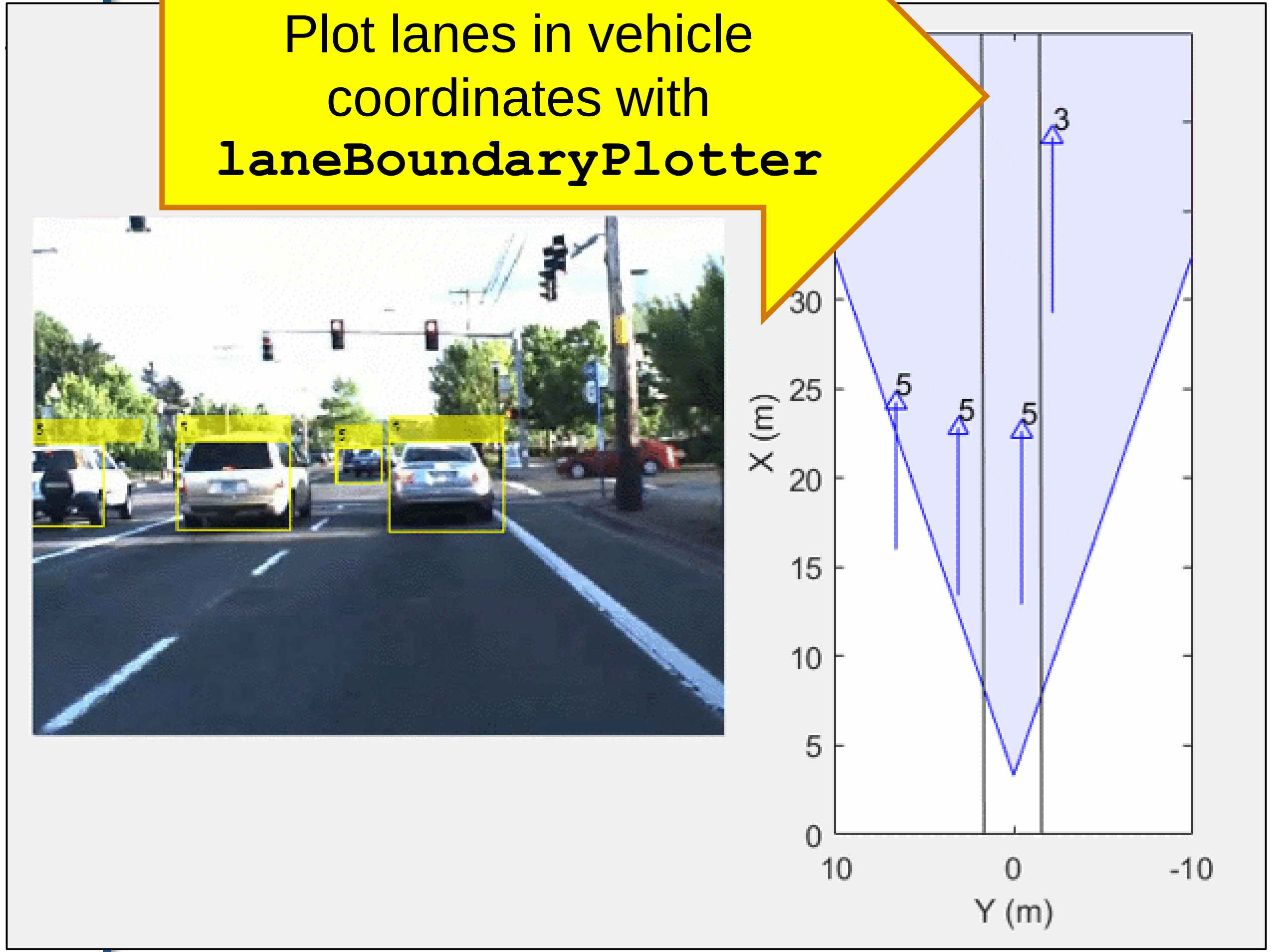
检测目标可视化 (图像坐标系)

```
%% Bounding box positions in image coordinates
imBoxes = zeros(numDets,4);
for k = 1:numDets
  if vision(n).object(k).classification == 5
    vehPosLR = vision(n).object(k).position(1:2)';
    imPosLR = vehicleToImage(sensor, vehPosLR);
    boxHeight = 1.4 * 1333 / vehPosLR(1);
    boxWidth  = 1.8 * 1333 / vehPosLR(1);
    imBoxes(k,:)=[imPosLR(1) - boxWidth/2, ...
                  imPosLR(2) - boxHeight, ...
                  boxWidth, boxHeight];
  end
end

%% Draw bounding boxes on image frame
frame = insertObjectAnnotation(frame, ...
     'Rectangle', imBoxes, labels,...
     'Color','yellow','LineWidth',2);
im.CData = frame;
```



车道线边界可视化 (车辆坐标)

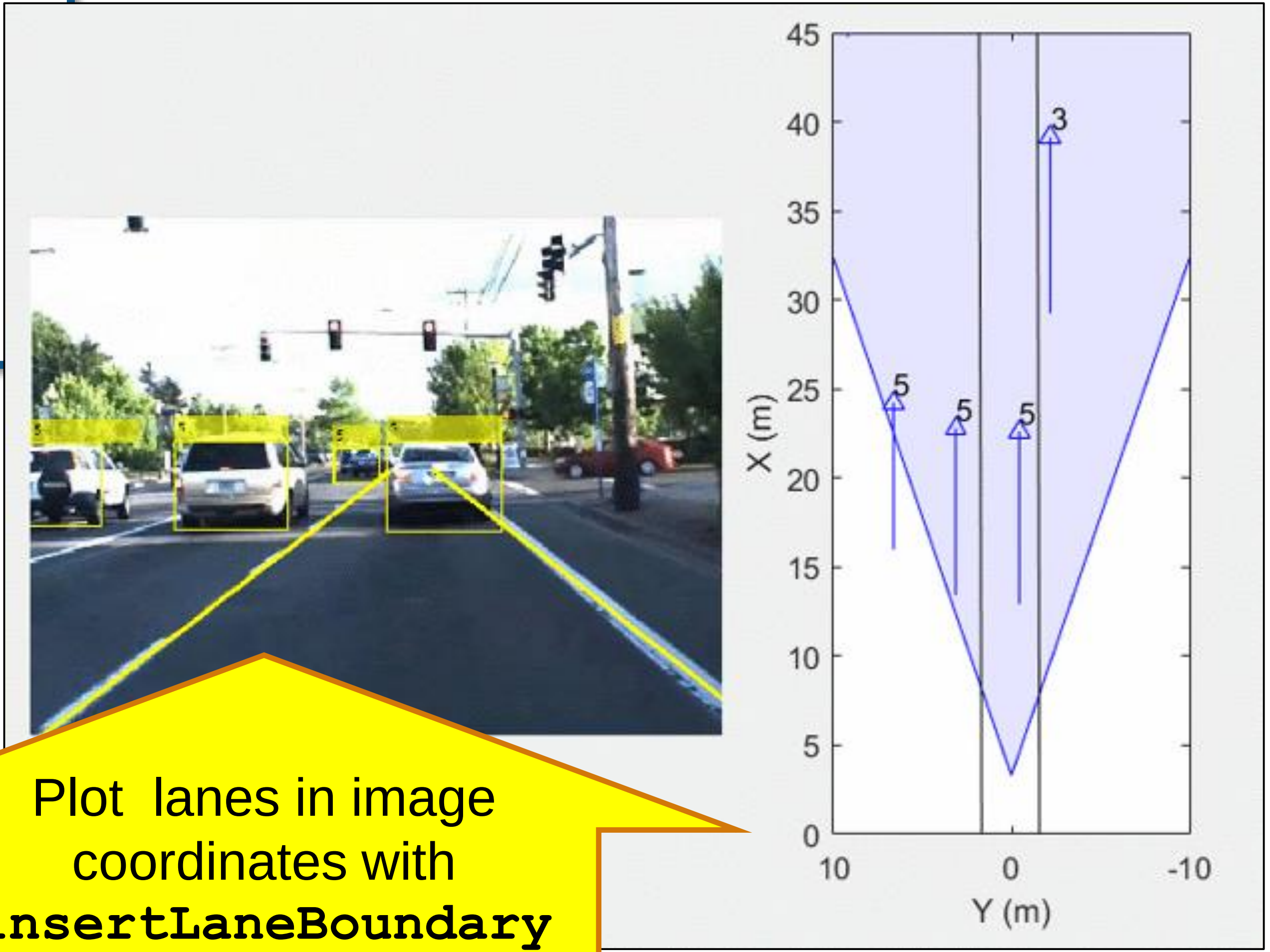
```
%% Create lane detection plotter
lanePlot = laneBoundaryPlotter(bep, ...
    'Color','black');

%% Update lane detection plotter
lb = parabolicLaneBoundary([...
        lane(n).left.curvature,...
        lane(n).left.headingAngle,...
        lane(n).left.offset]);
rb = parabolicLaneBoundary([...
        lane(n).right.curvature,...
        lane(n).right.headingAngle,...
        lane(n).right.offset]);
plotLaneBoundary(lanePlot, [lb rb])
```



Plot lanes in vehicle coordinates with **laneBoundaryPlotter**

车道线边界可视化 (图像坐标系)

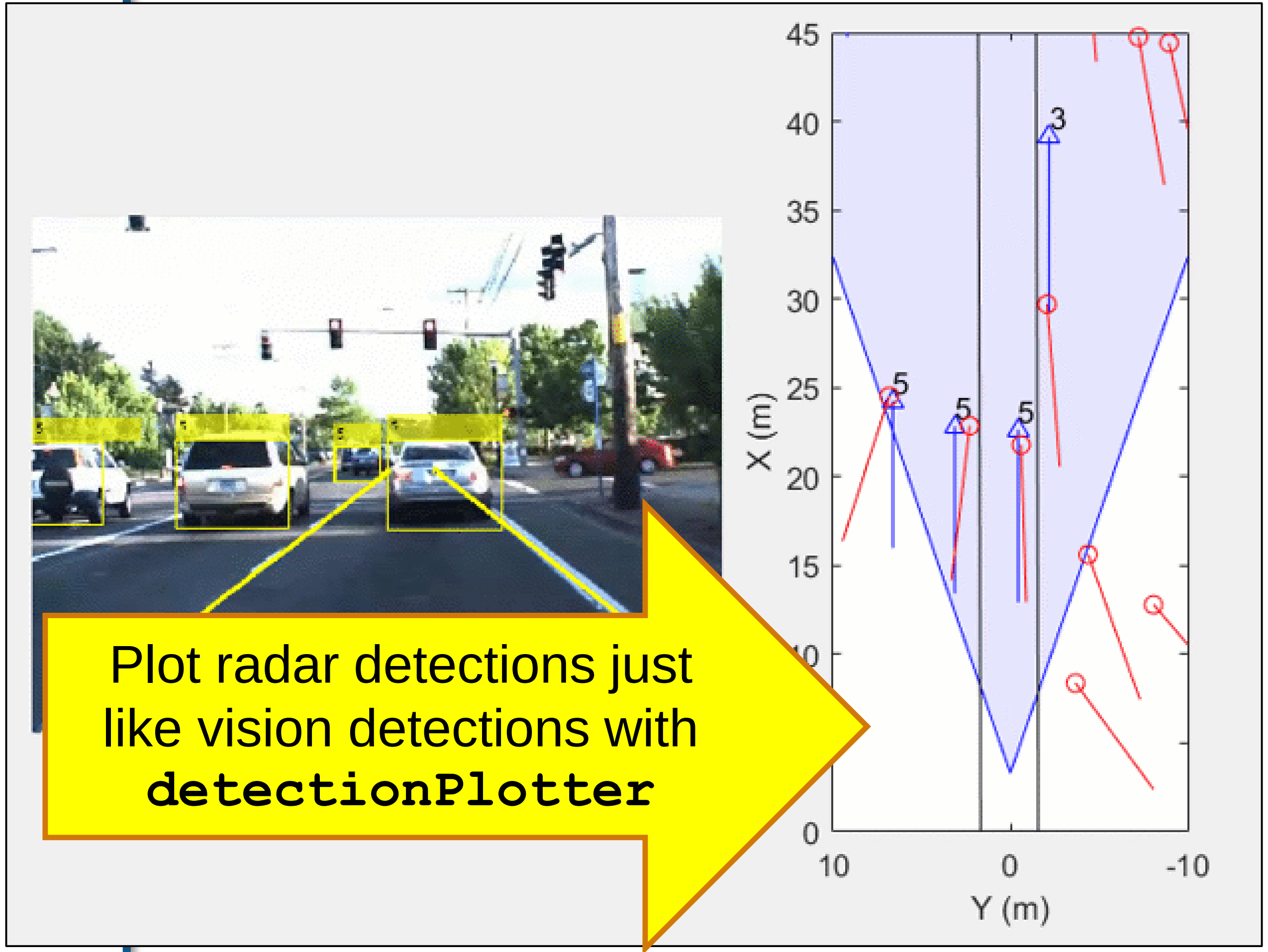
```
%% Draw in image coordinates
frame = insertLaneBoundary(frame, ...
     [lb rb], sensor, (1:100),...
     'LineWidth',5);
im.CData = frame;
```



Plot lanes in image coordinates with **`insertLaneBoundary`**

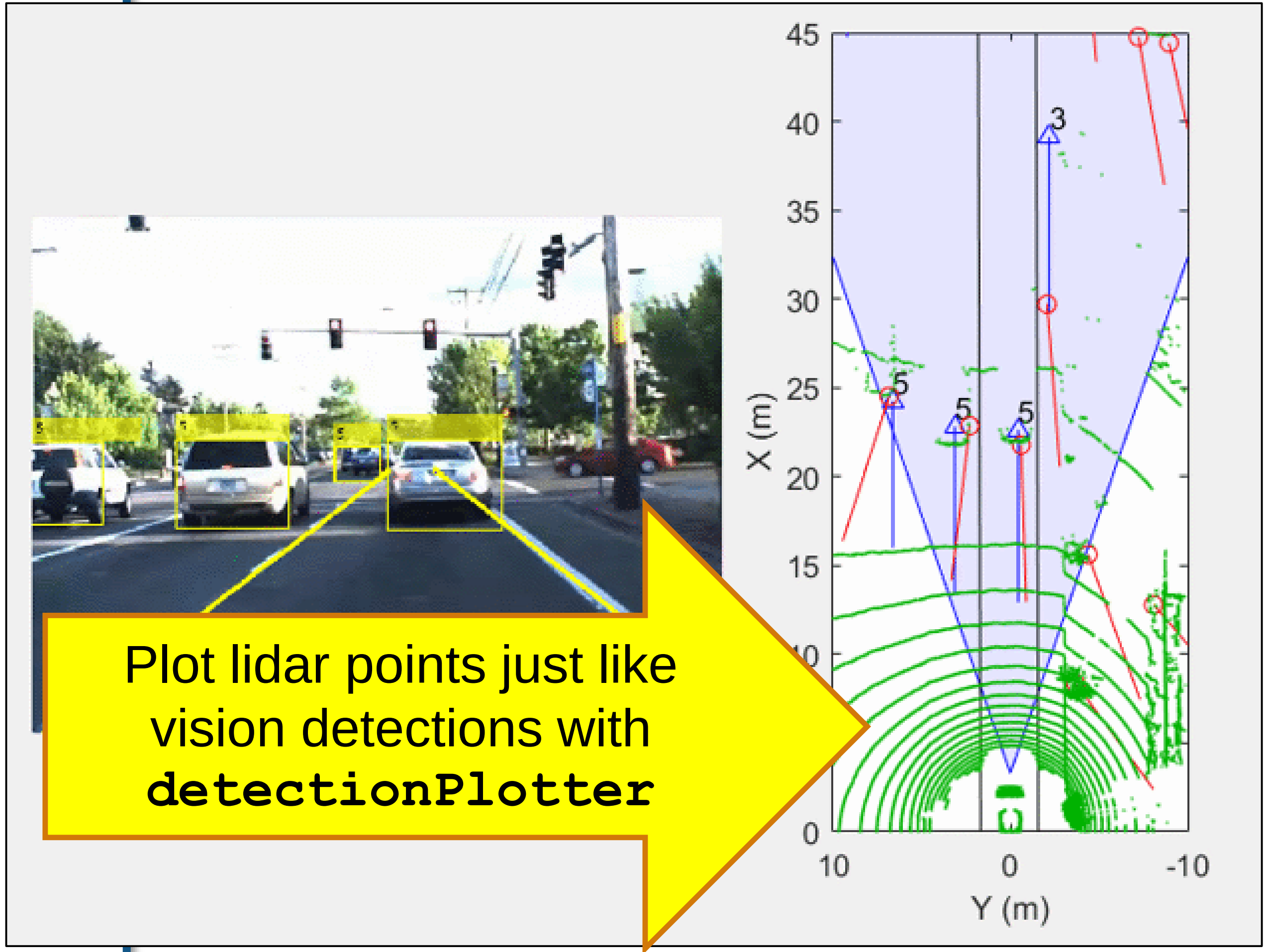
雷达检测结果可视化 (车辆坐标系)

```
%% Create radar detection plotter
radarPlot = detectionPlotter(bep, ...
    'MarkerEdgeColor','red',...
    'Marker','o');

%% Update radar detection plotter
numDets = radar(n).numObjects;
pos = zeros(numDets,3);
vel = zeros(numDets,3);
for k = 1:numDets
    pos(k,:) = radar(n).object(k).position;
    vel(k,:) = radar(n).object(k).velocity;
end
plotDetection(radarPlot,pos,vel);
```



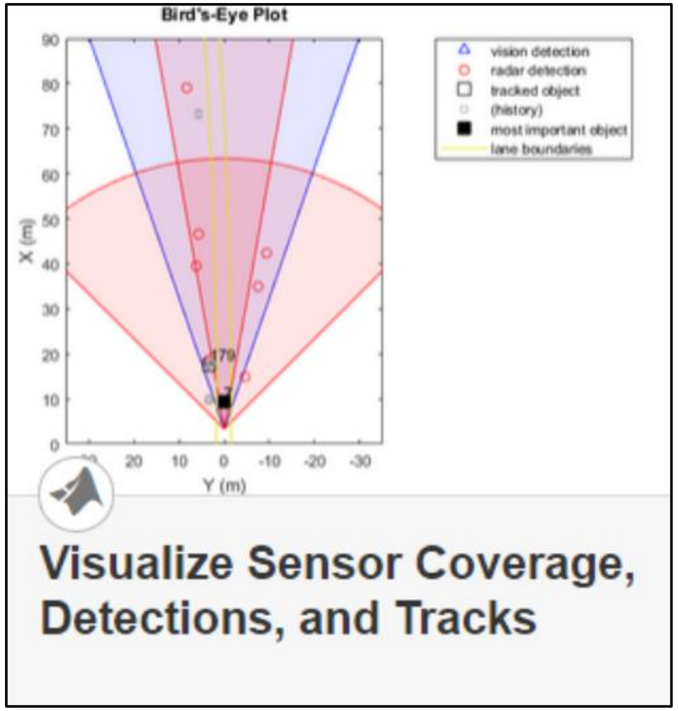
激光雷达点云可视化 (车辆坐标系)

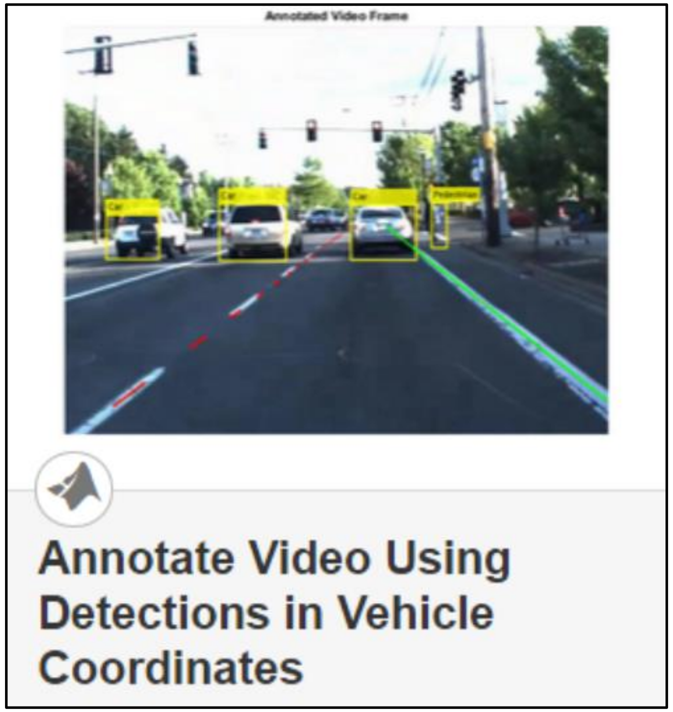
```
%% Create lidar detection plotter
lidarPlot = detectionPlotter(bep, ...
    'Marker','.',...
    'MarkerSize',1.5,...
    'MarkerEdgeColor',[0 0.7 0]); % Green

%% Update lidar detection plotter
n = round(video.CurrentTime/0.1);
pos = ...
  LidarPointCloud(n).ptCloud.Location(:,1:2);
plotDetection(lidarPlot,pos);
```

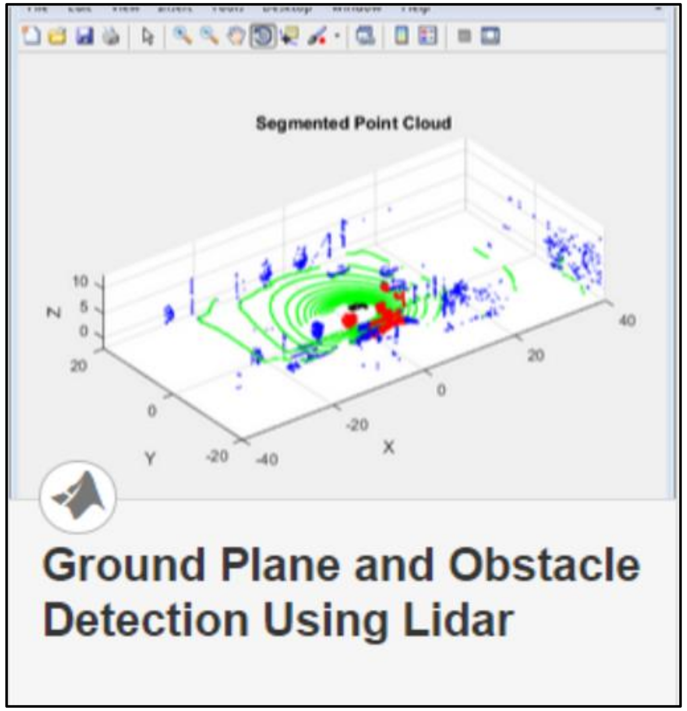


了解更多车辆数据可视化

查看Automated Driving System Toolbox中的例子





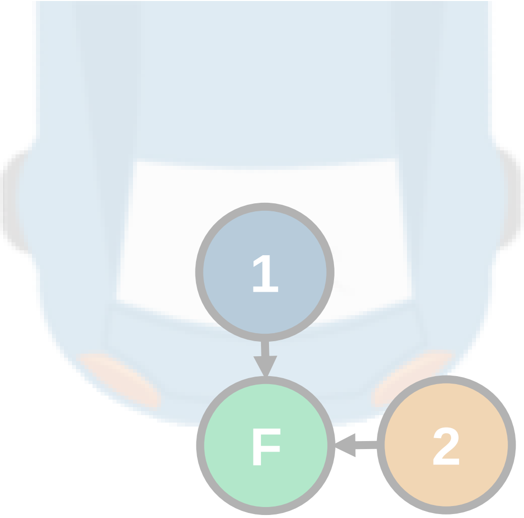


- 在车辆坐标系中呈现检测目标
  - Vision & radar detector
  - Lane detectors
  - Detector coverage areas
- 车辆坐标系和图像坐标系转换
- 绘制点云数据

自动驾驶工程师经常遇到的问题：

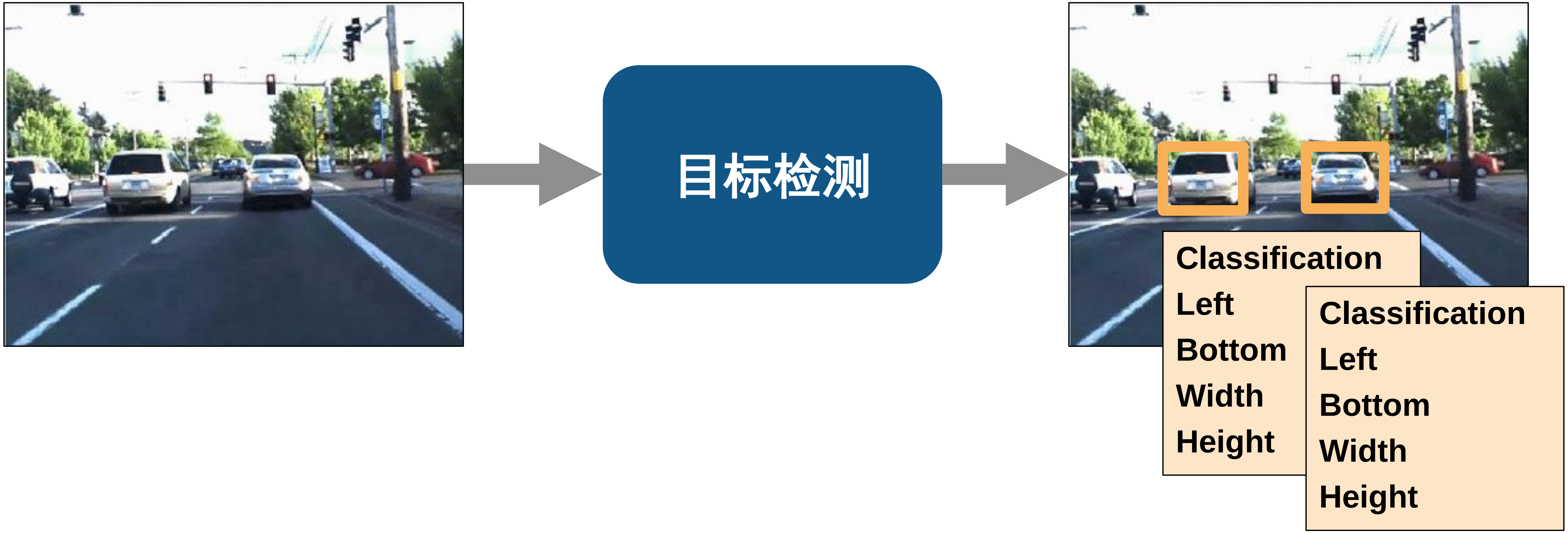
我怎样可视化
车辆的数据?

我怎样检测图
像中的目标?

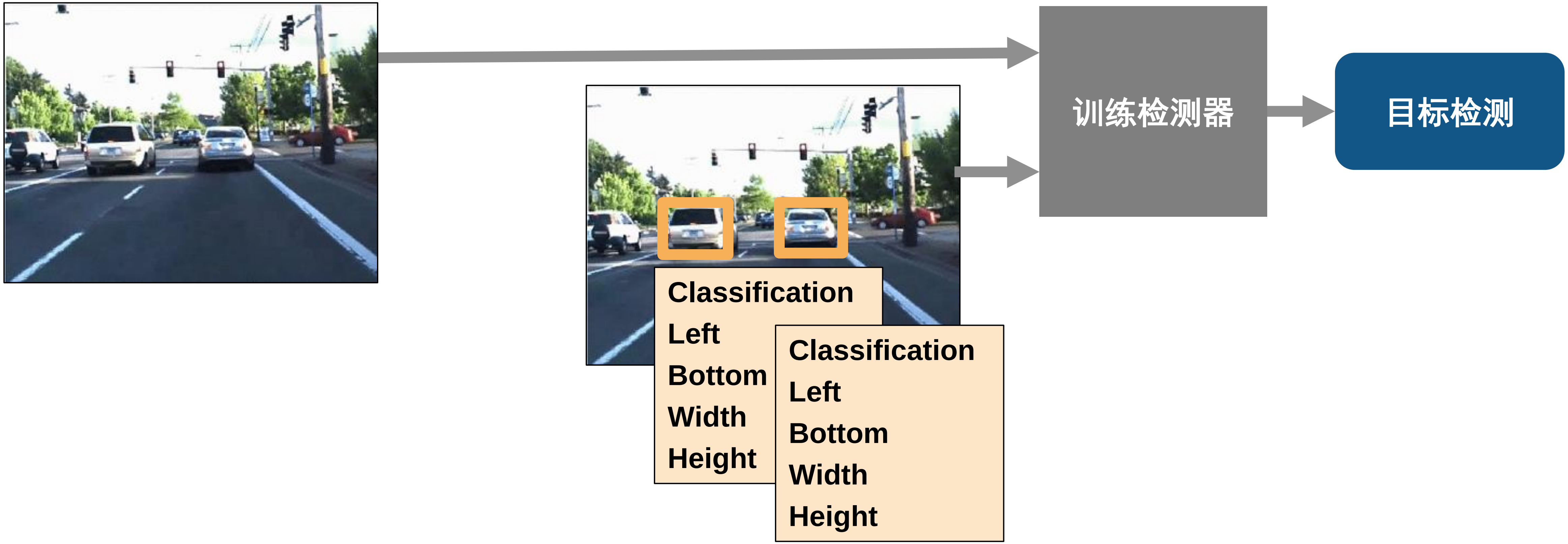


我怎样融合
多个检测结果?

我怎样检测图像中的目标?



基于真实值训练目标检测算法

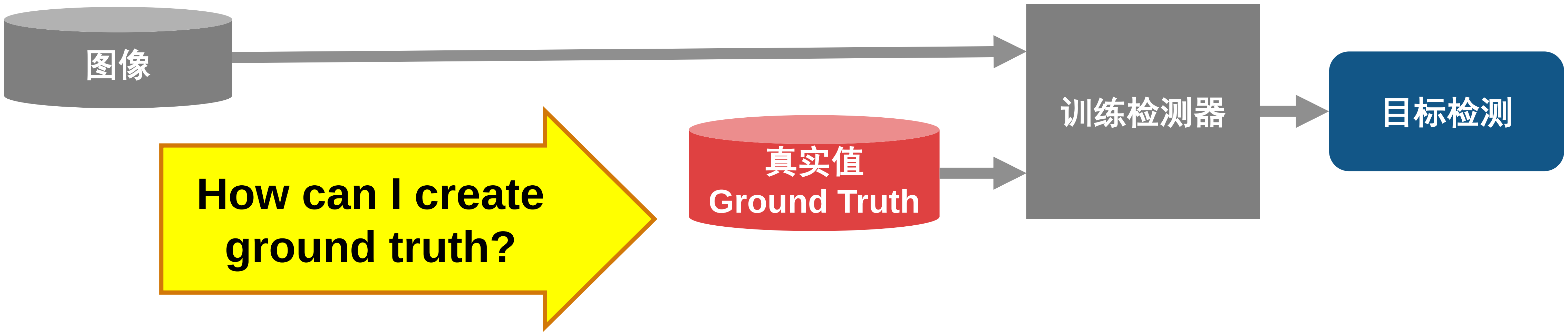


基于真实值训练目标检测算法

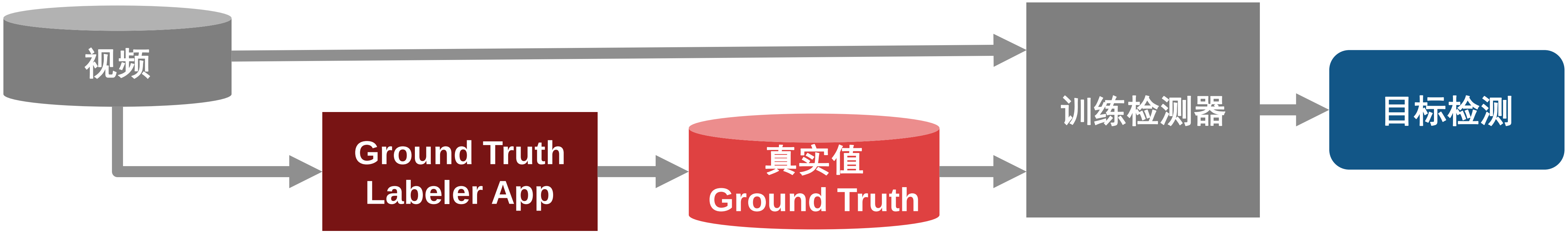
借助Computer Vision System Toolbox设计目标检测器

机器学习	Aggregate Channel Feature	`trainACFObjectDetector`
	Cascade	`trainCascadeObjectDetector`
深度学习	R-CNN (Regions with Convolutional Neural Networks)	`trainRCNNObjectDetector`
	Fast R-CNN	`trainFastRCNNObjectDetector`
	Faster R-CNN	`trainFasterRCNNObjectDetector`

为训练检测器提供真实值

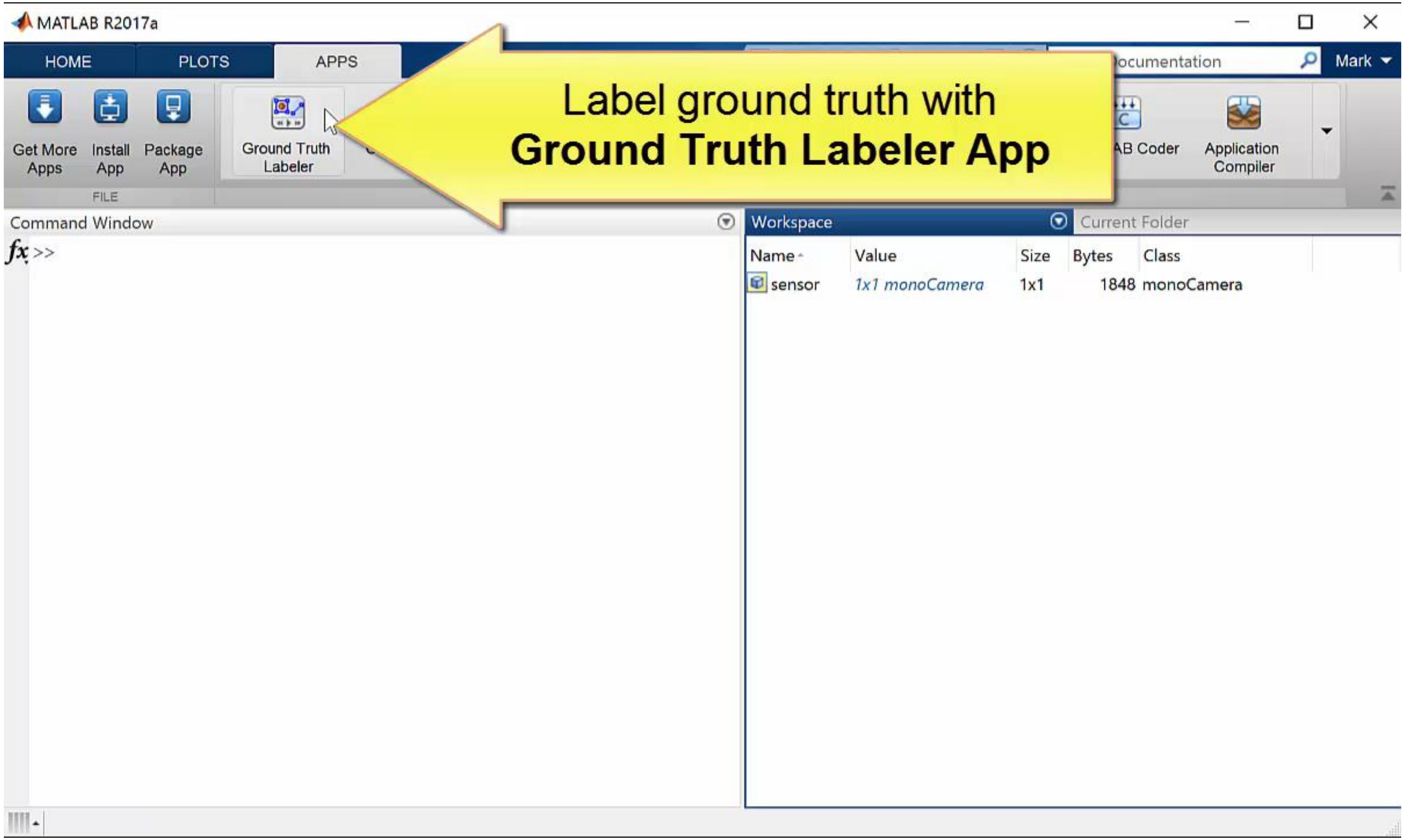


为训练检测器提供真实值



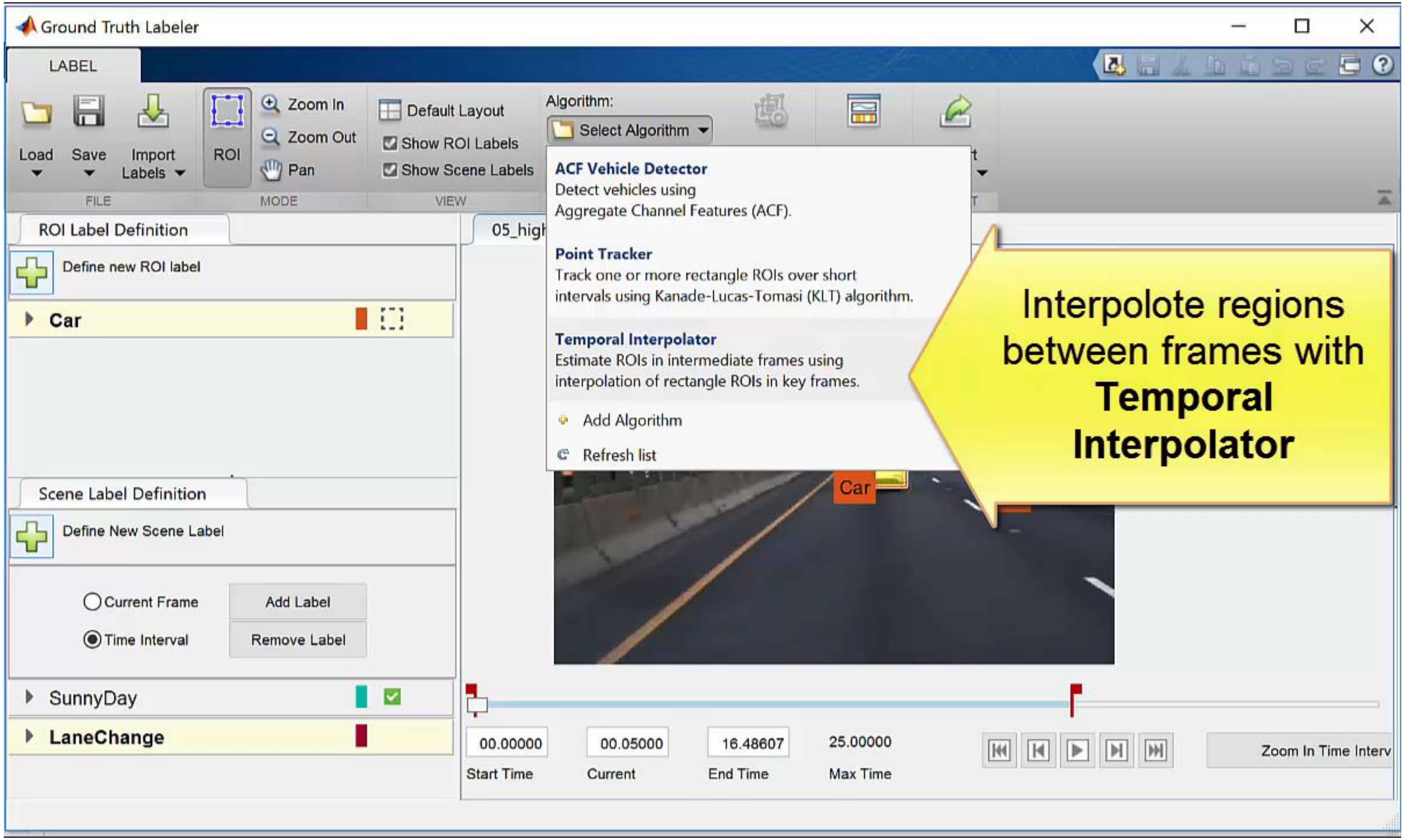
手工标注目标的真实值

with Ground Truth Labeling App



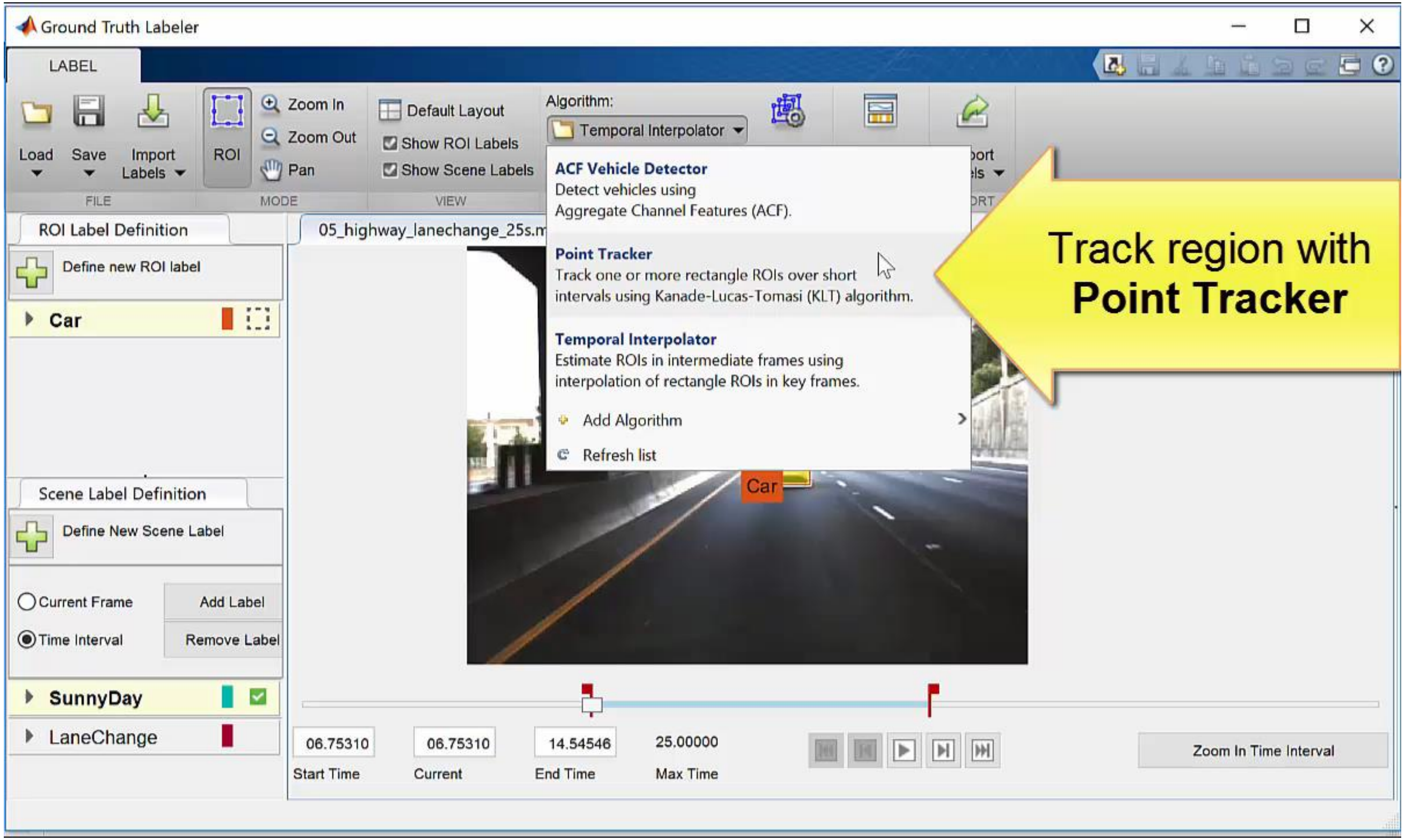
在手工标注的帧之间自动标注

with temporal interpolator



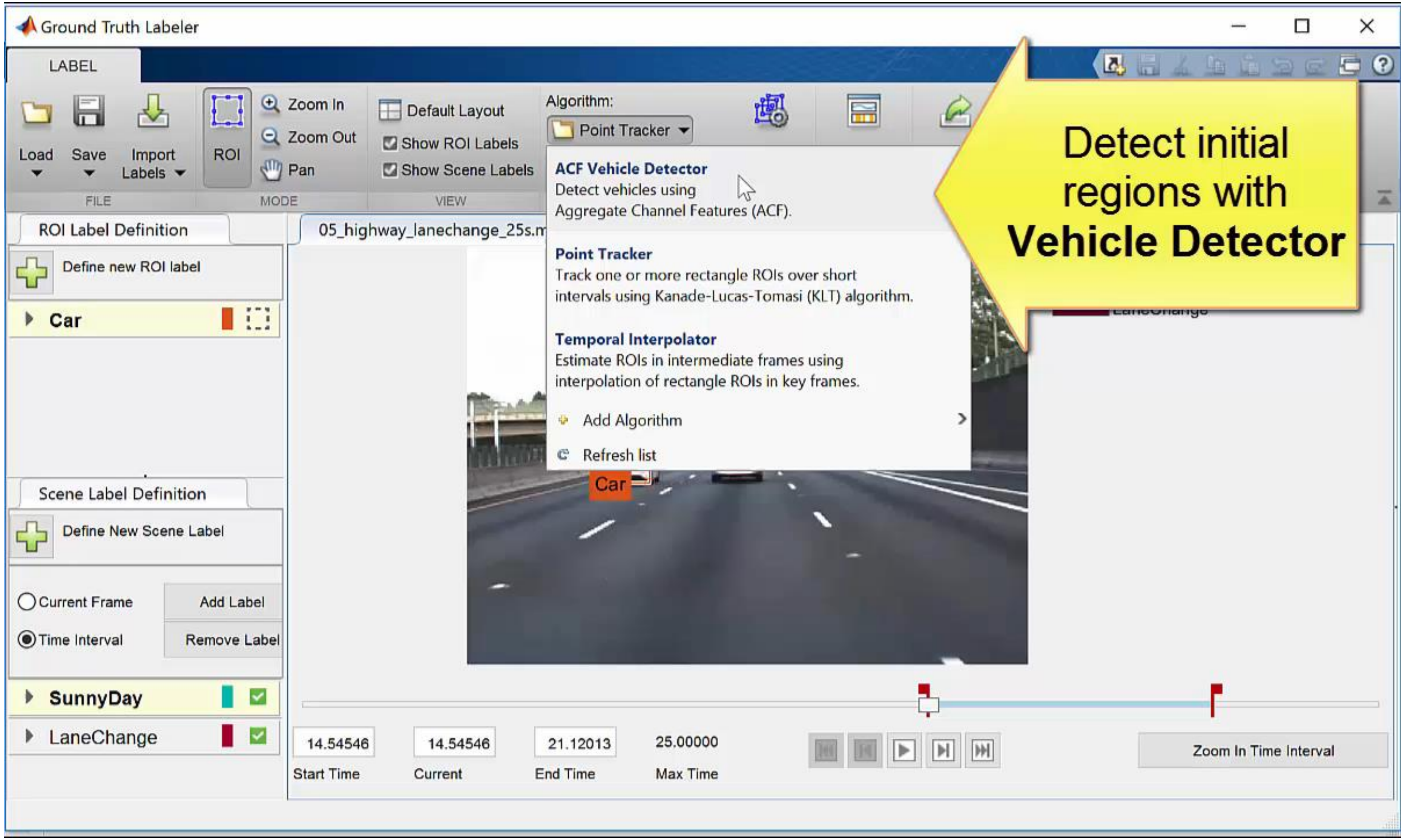
在手工标注帧的基础上自动标注

with point tracker

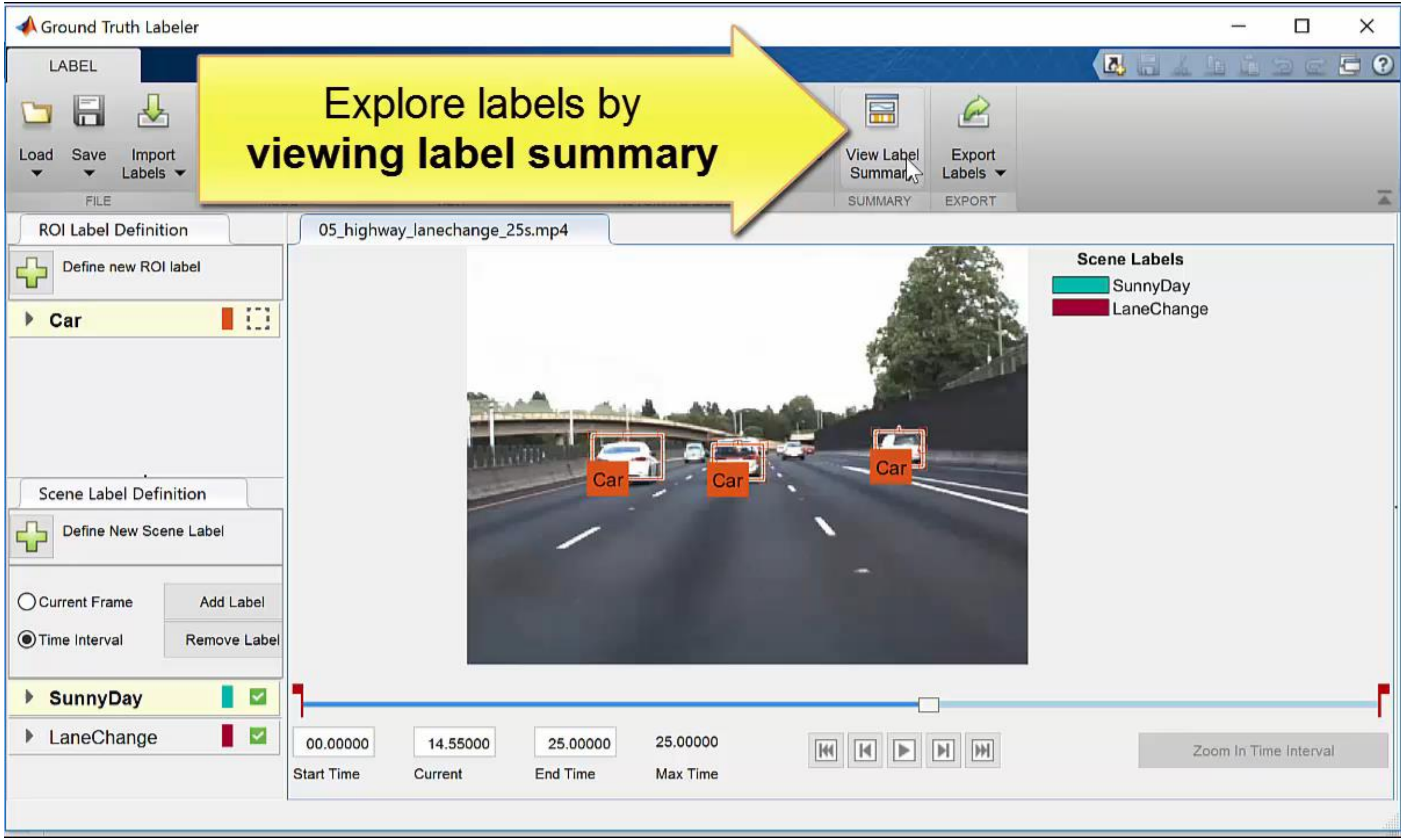


自动化检测车辆目标的真实值

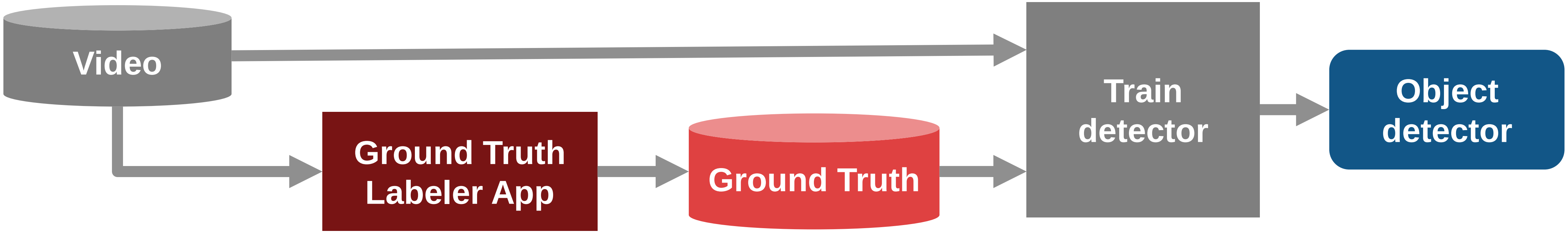
with ACF ground truth detector



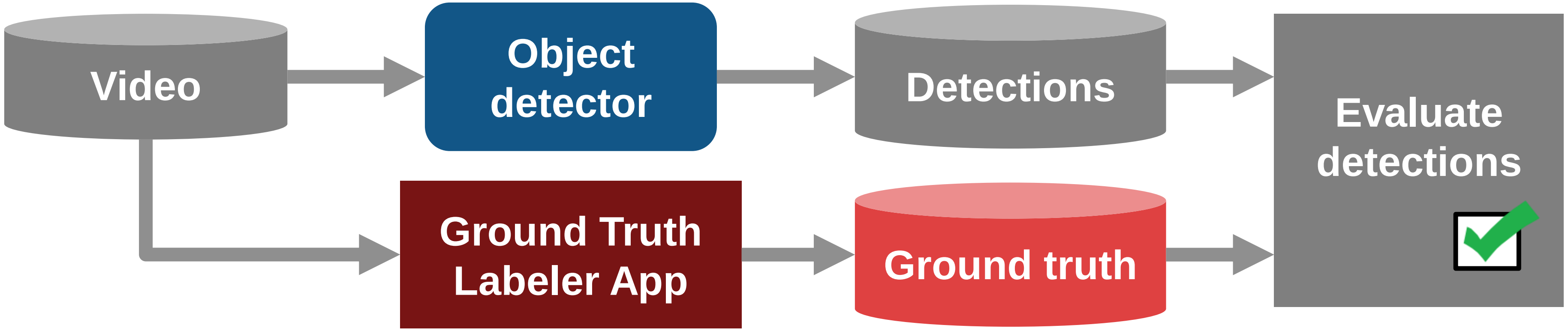
以MATLAB时间表方式导出标注区域



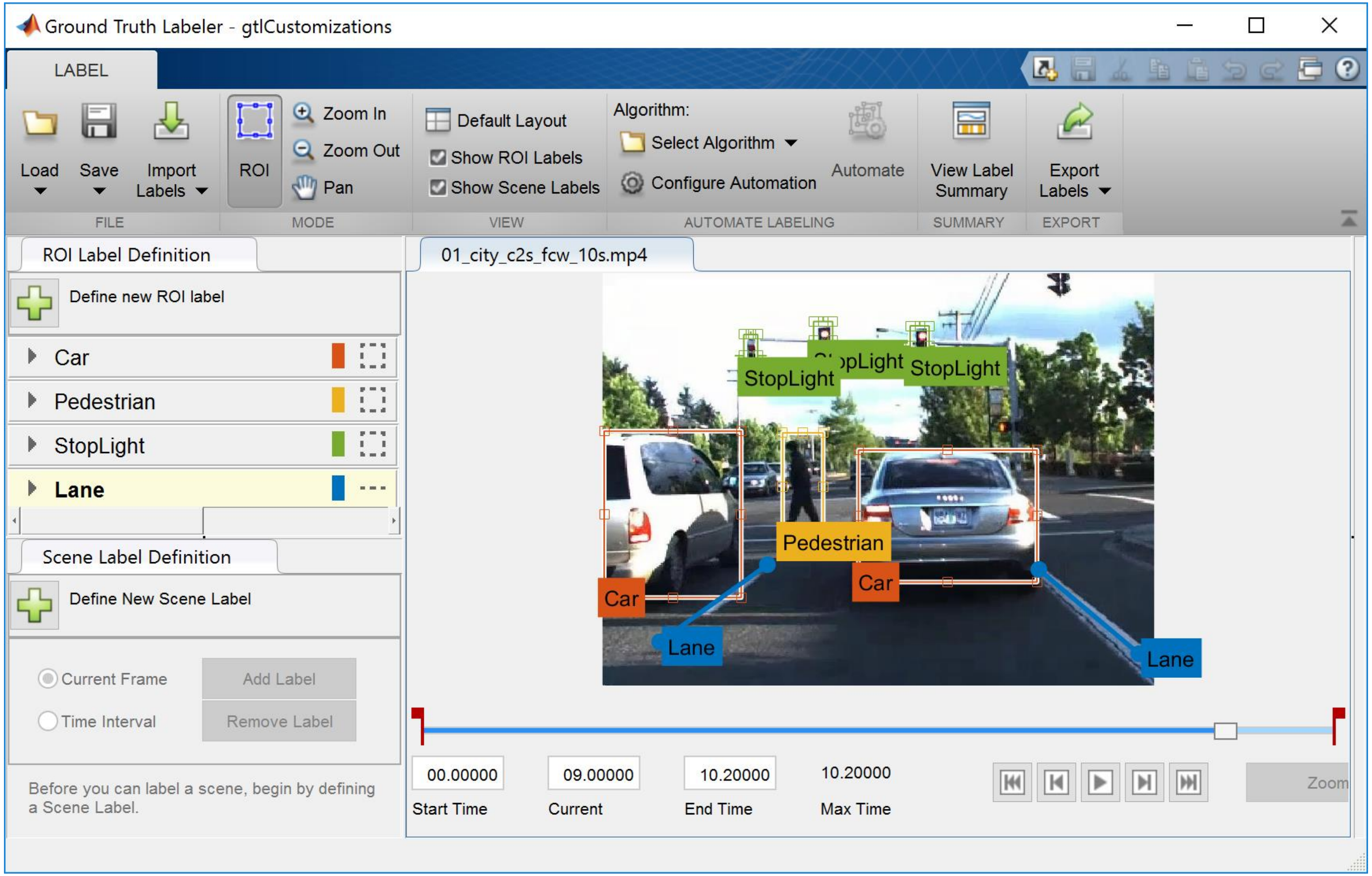
Ground truth labeling to train detectors



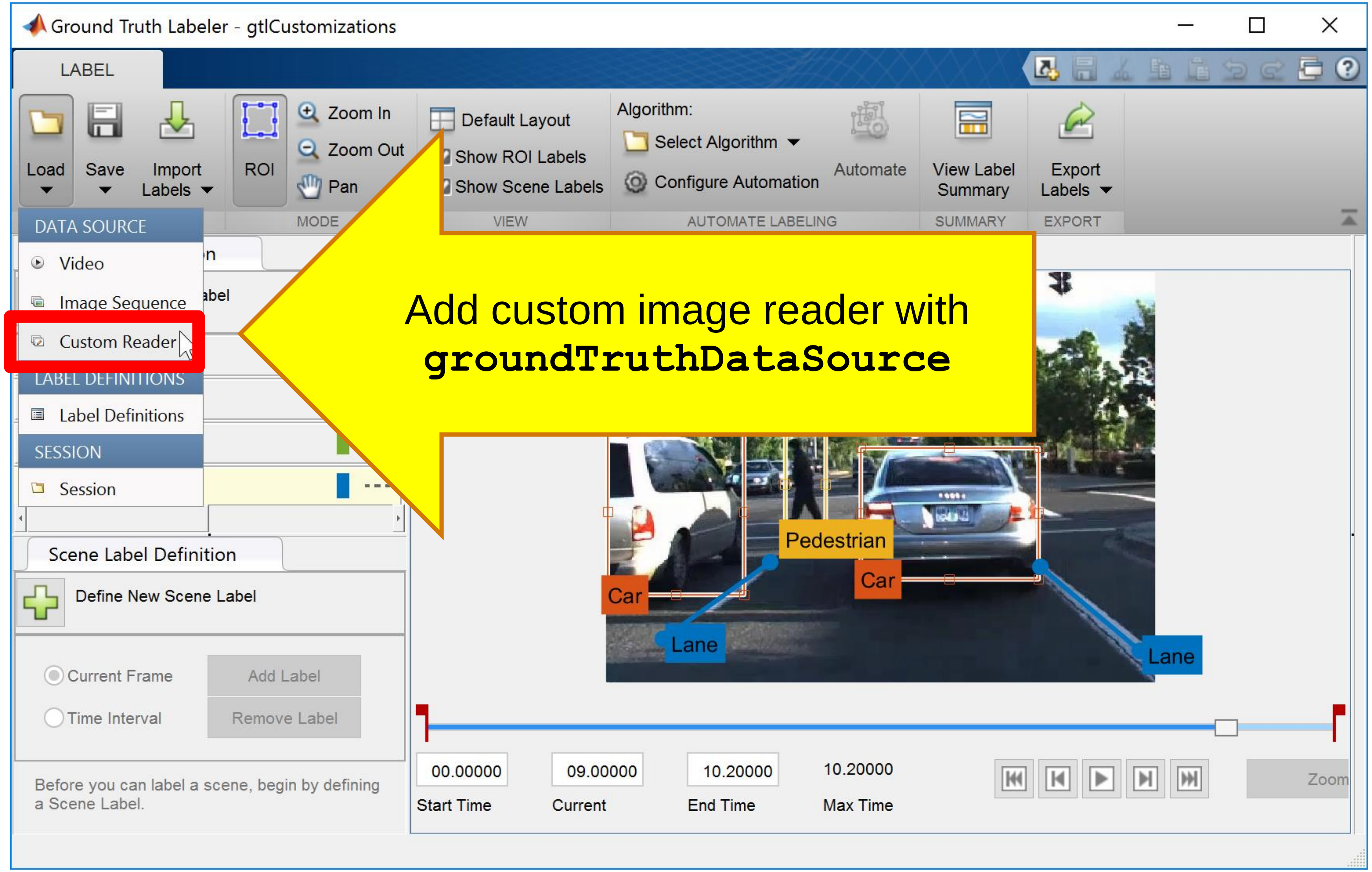
Ground truth labeling to evaluate detectors



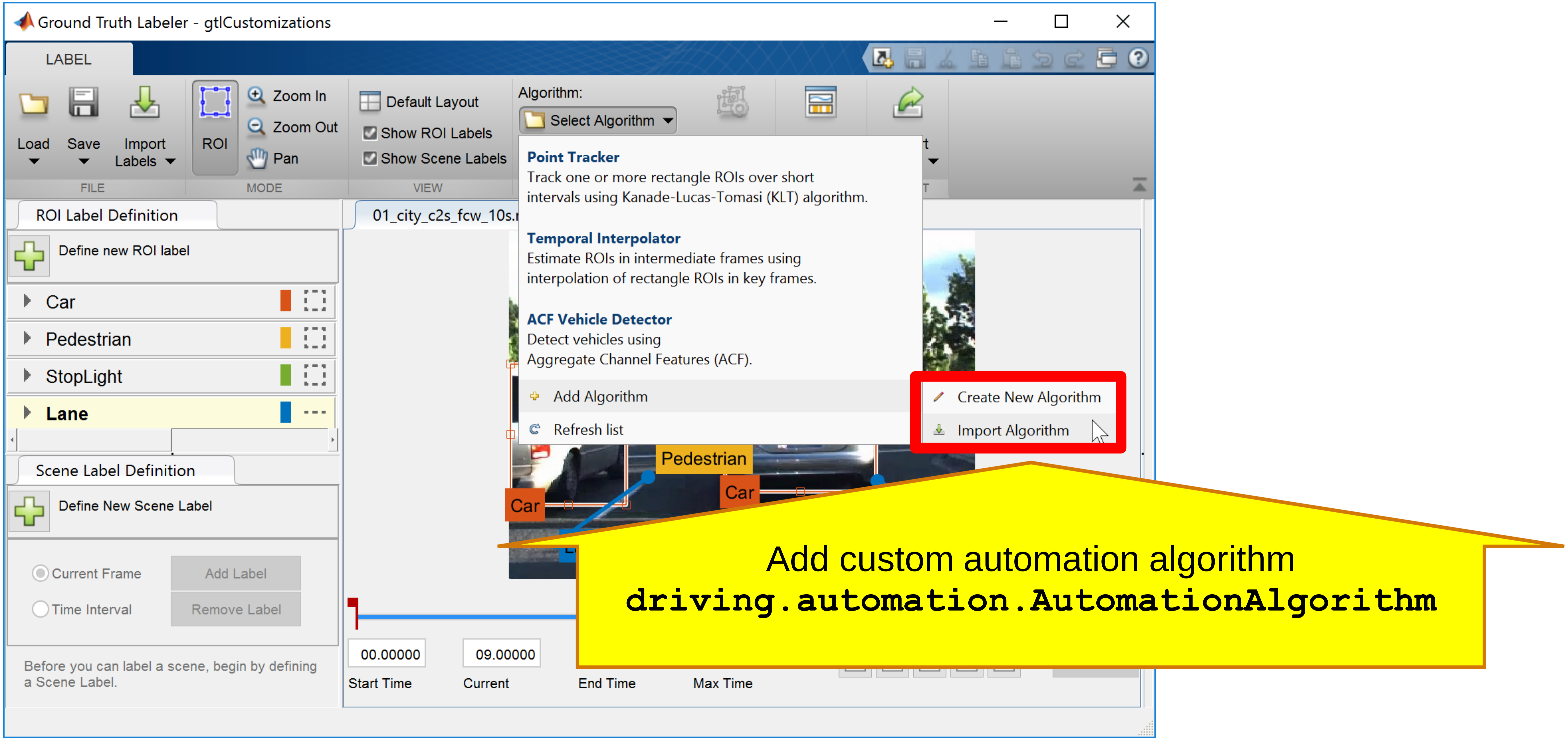
定制化真实值标注应用程序



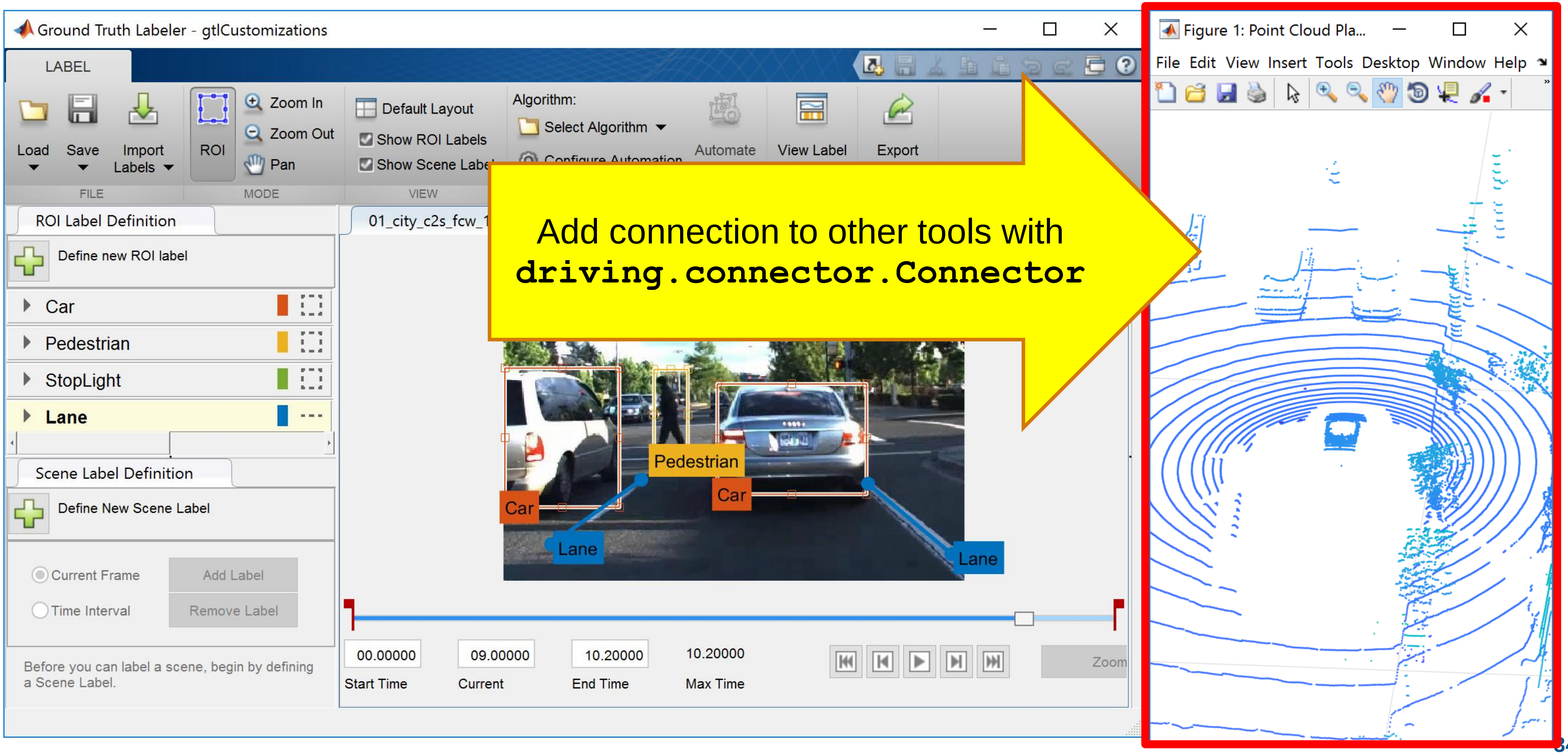
定制化真实值标注应用程序



定制化真实值标注应用程序

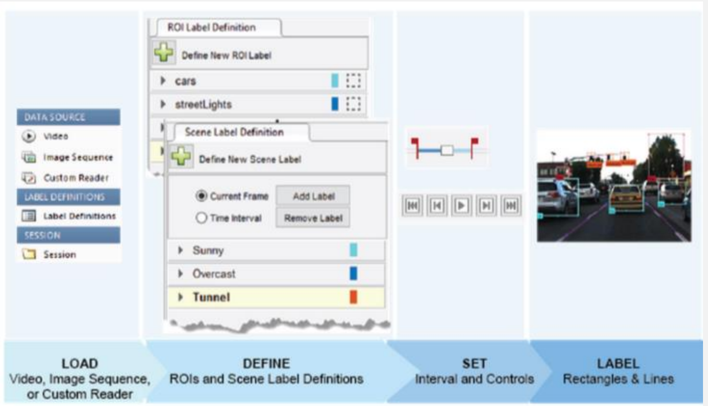


定制化真实值标注应用程序

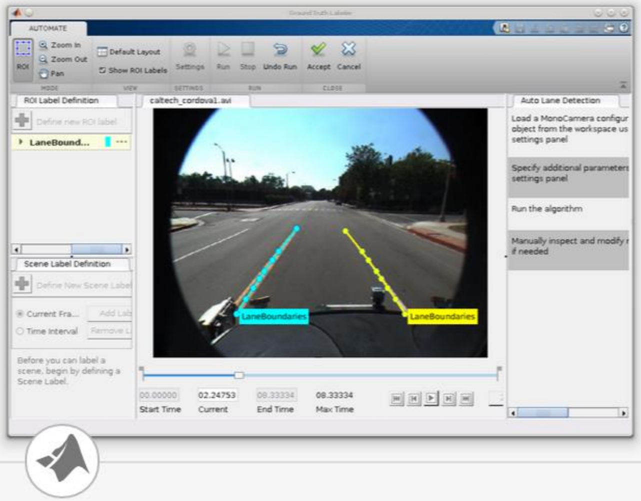


了解更多在图像中检测目标

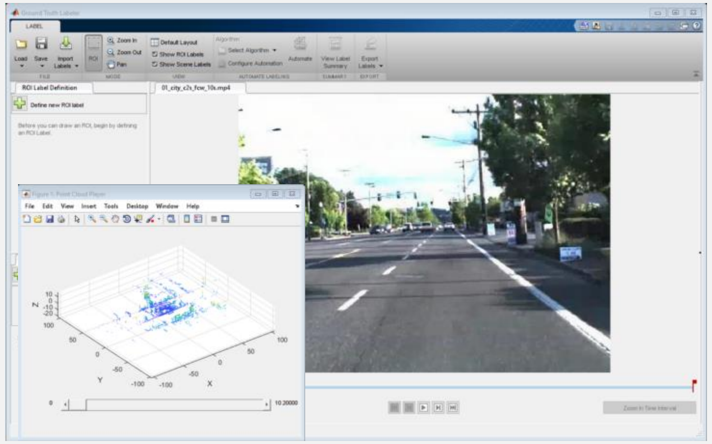
查看Automated Driving System Toolbox中的例子



Define Ground Truth Data for Video or Image Sequences



Automate Ground Truth Labeling of Lane Boundaries



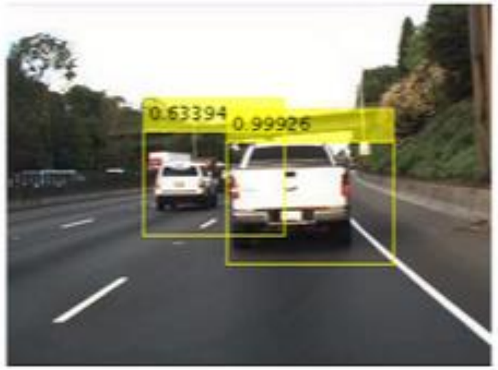
driving.connector.Connector class

Connect Lidar Display to Ground Truth Labeler

- 用Ground Truth Labeler App **标注检测结果**
- 为车道线检测 **加入自动化算法**
- 为Ground Truth Labeler App **连接扩展功能**

了解更多在图像中检测目标

查看Automated Driving System Toolbox中的例子



Train a Deep Learning Vehicle Detector

- **训练目标检测器**
  应用深度学习和机器学习技术

Track Pedestrians from a Moving Car

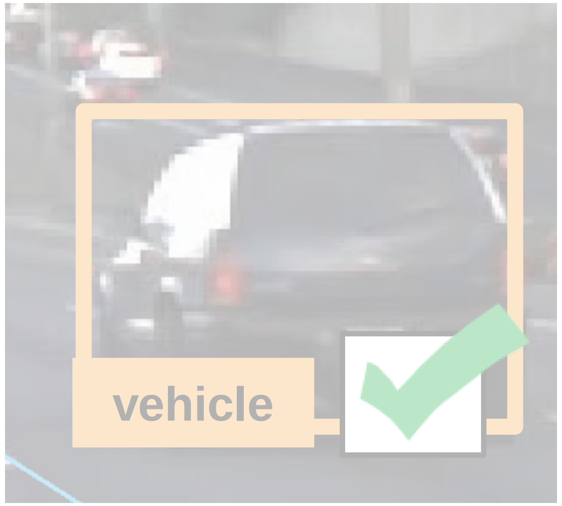
- **探索预先训练好的行人检测器**

Visual Perception Using Monocular Camera

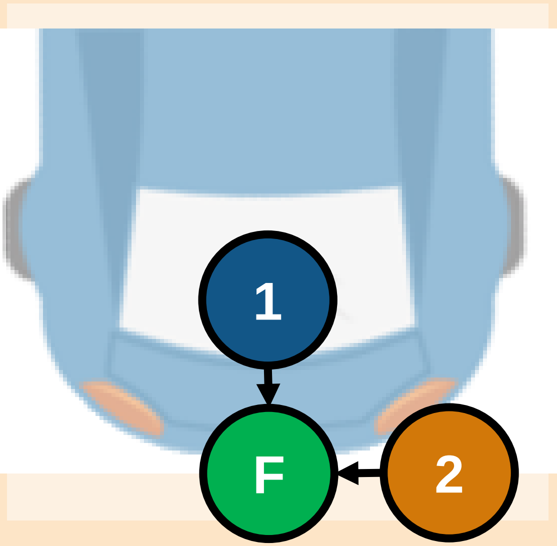
- **考察车道检测器**
  根据摄像头传感器模型转换坐标系

自动驾驶工程师经常遇到的问题：

我怎样可视化
车辆的数据?

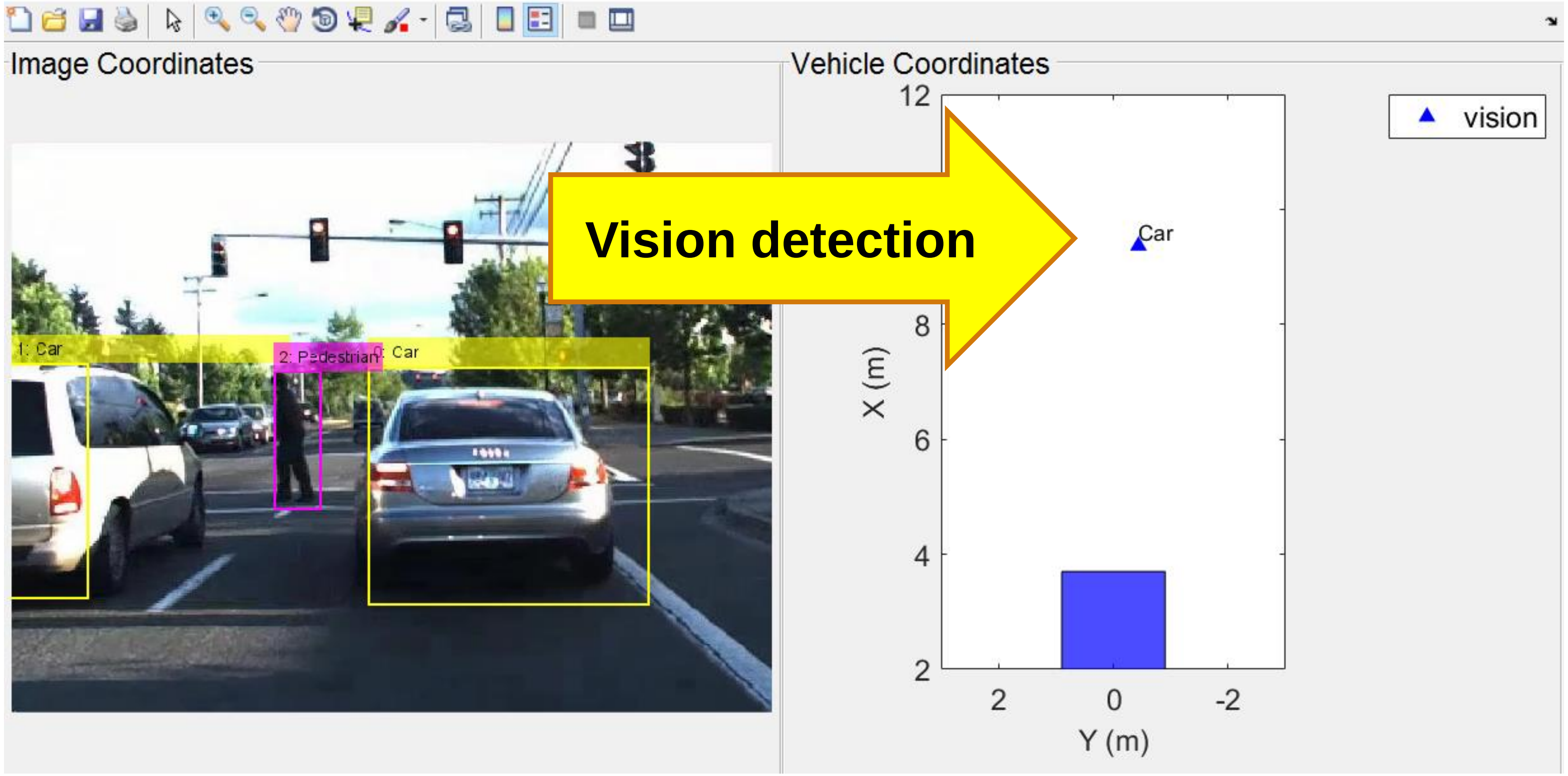


我怎样检测图
像中的目标?

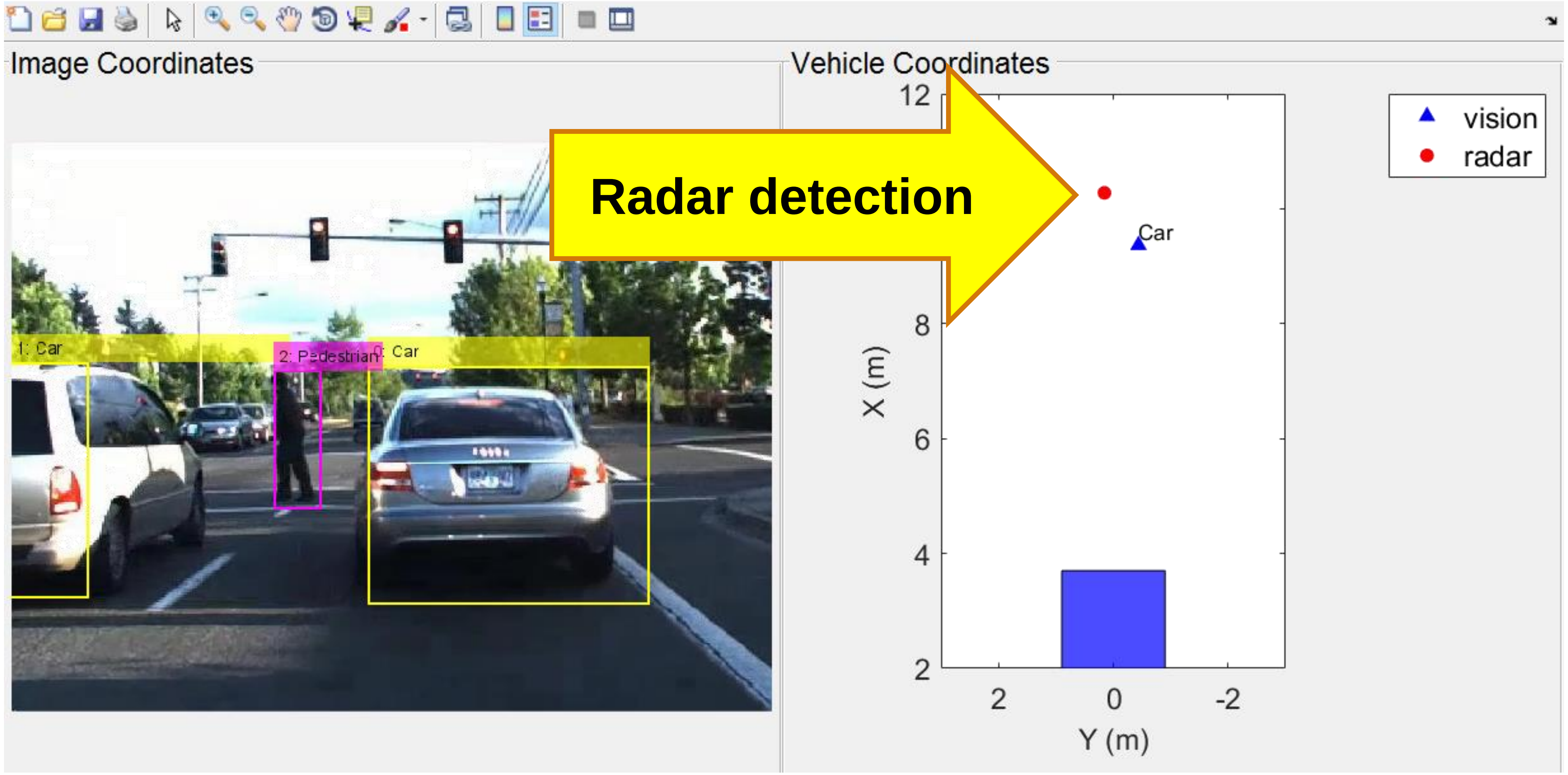


我怎样融合
多个检测结果?

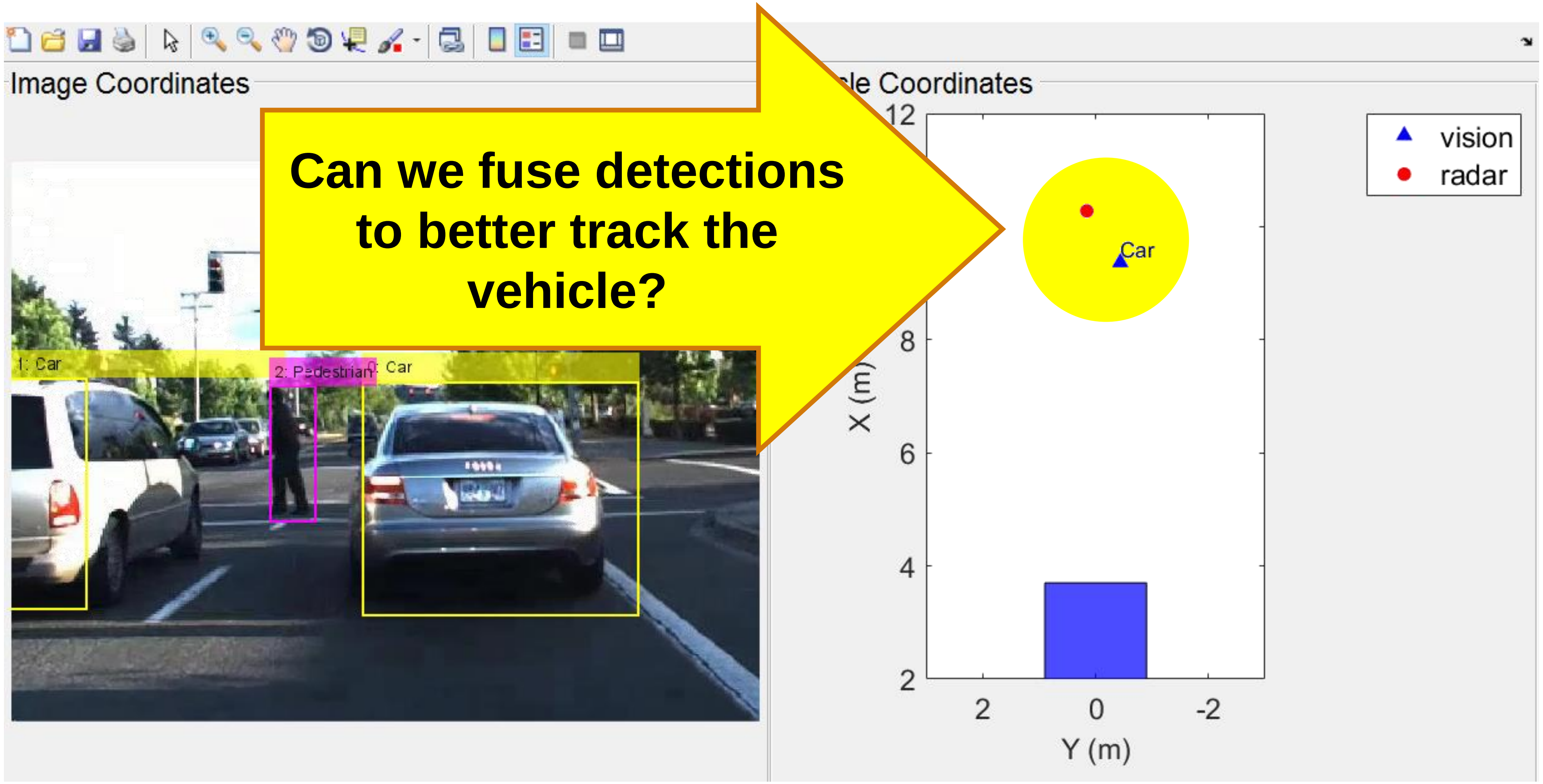
雷达和视觉检测车辆的例子



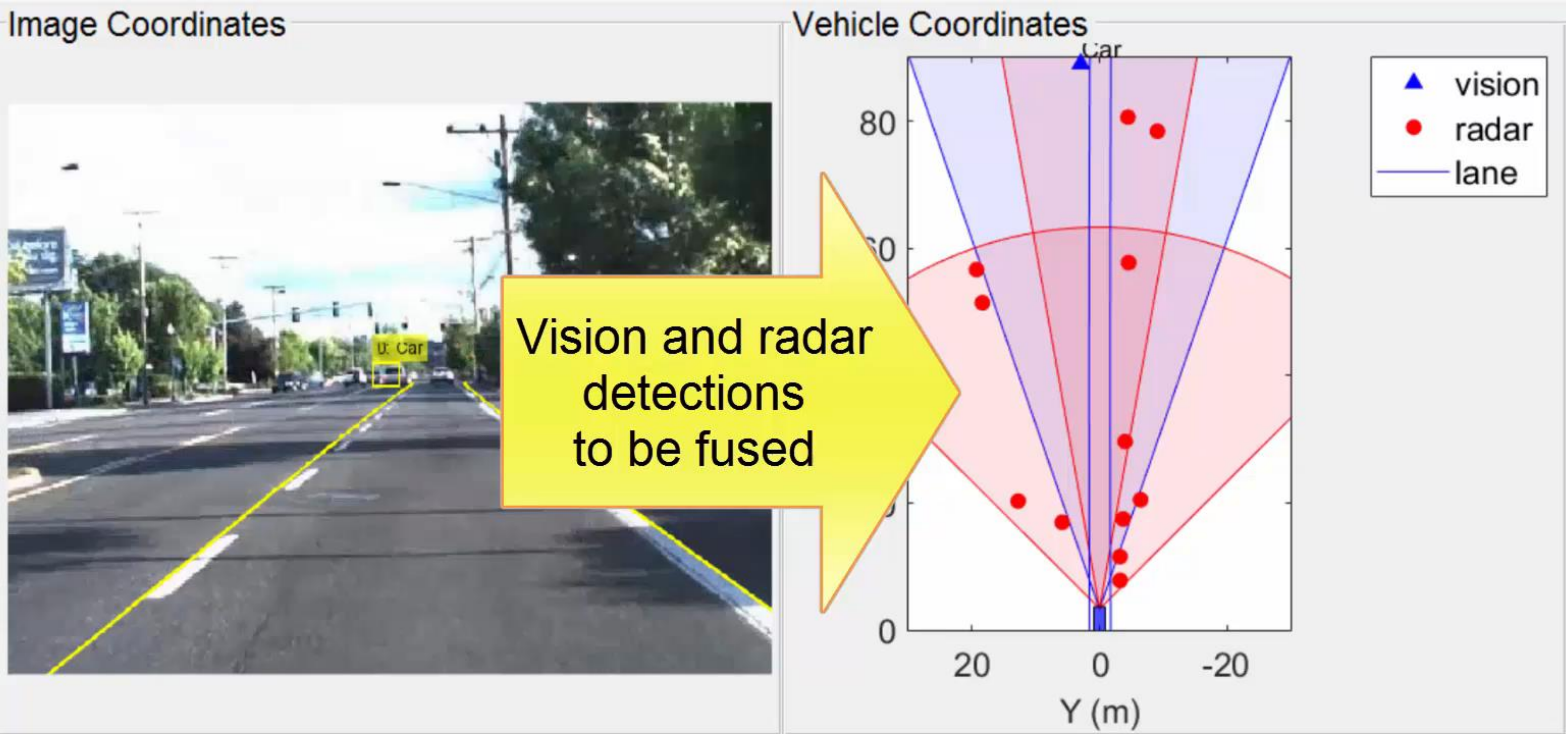
雷达和视觉检测车辆的例子



雷达和视觉检测车辆的例子

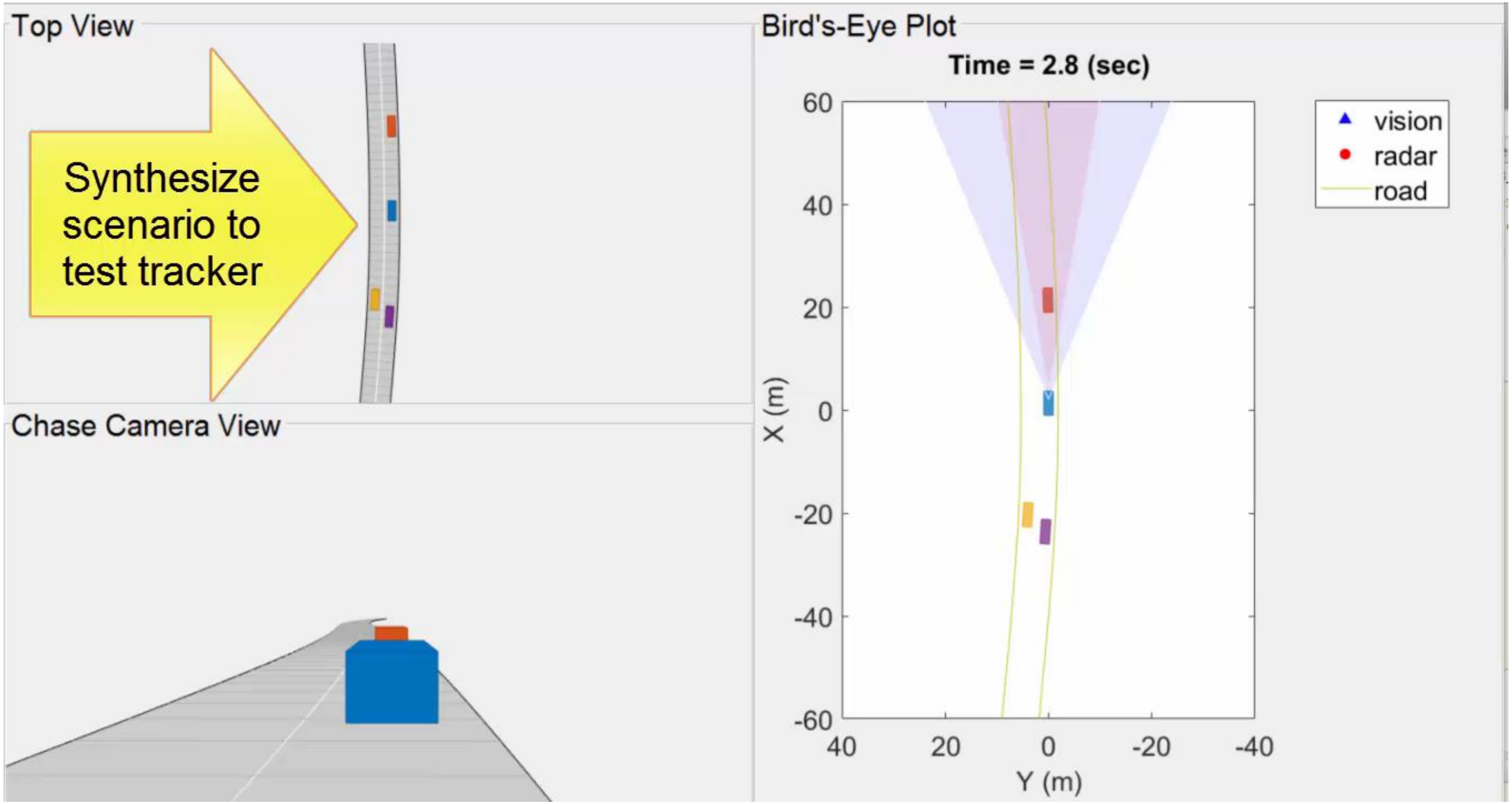


用多目标跟踪器融合检测目标

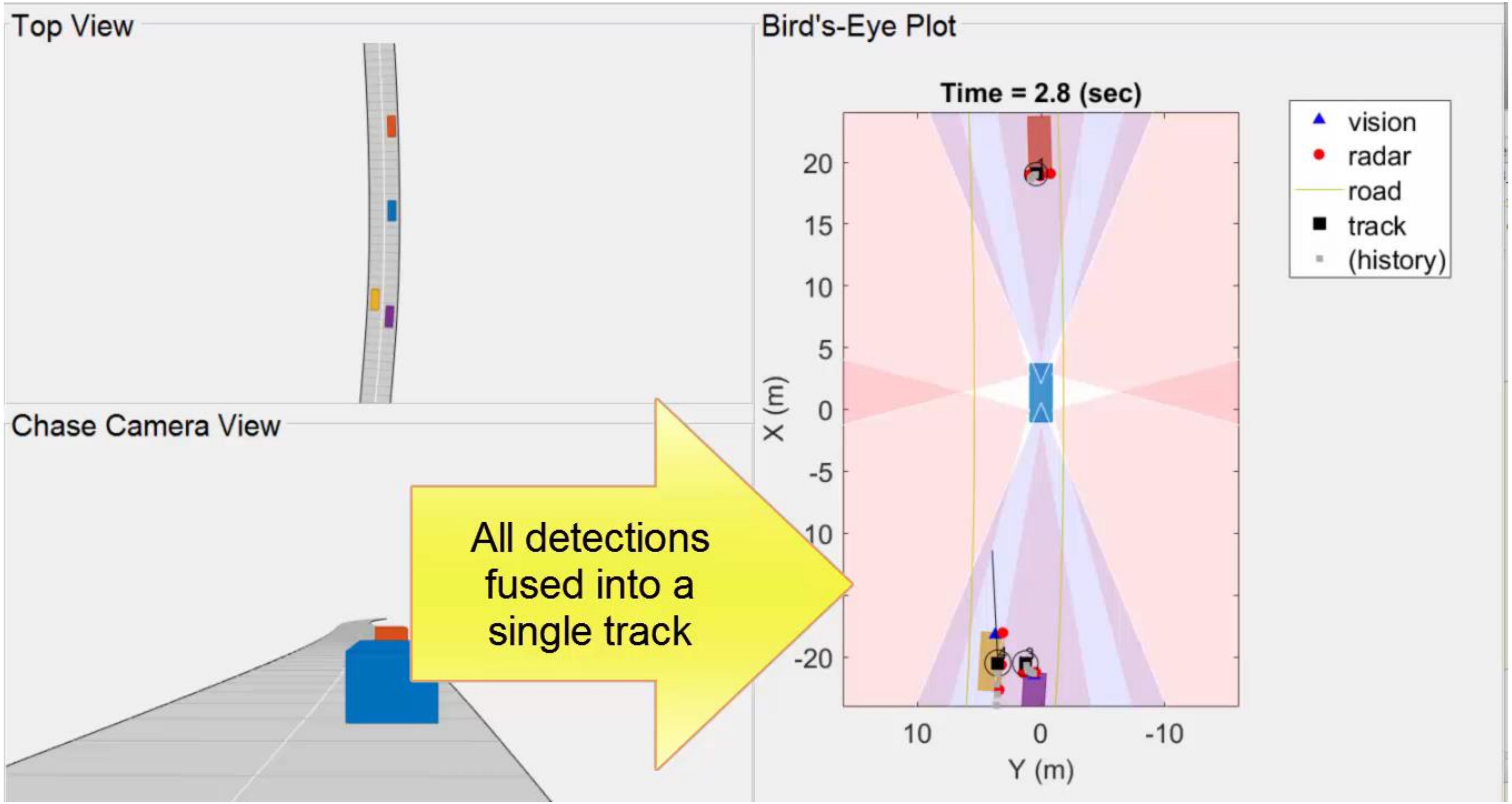


将跟踪器集成到更上层的算法中

生成交通场景测试跟踪器



用合成的数据测试跟踪器



跟踪多目标检测

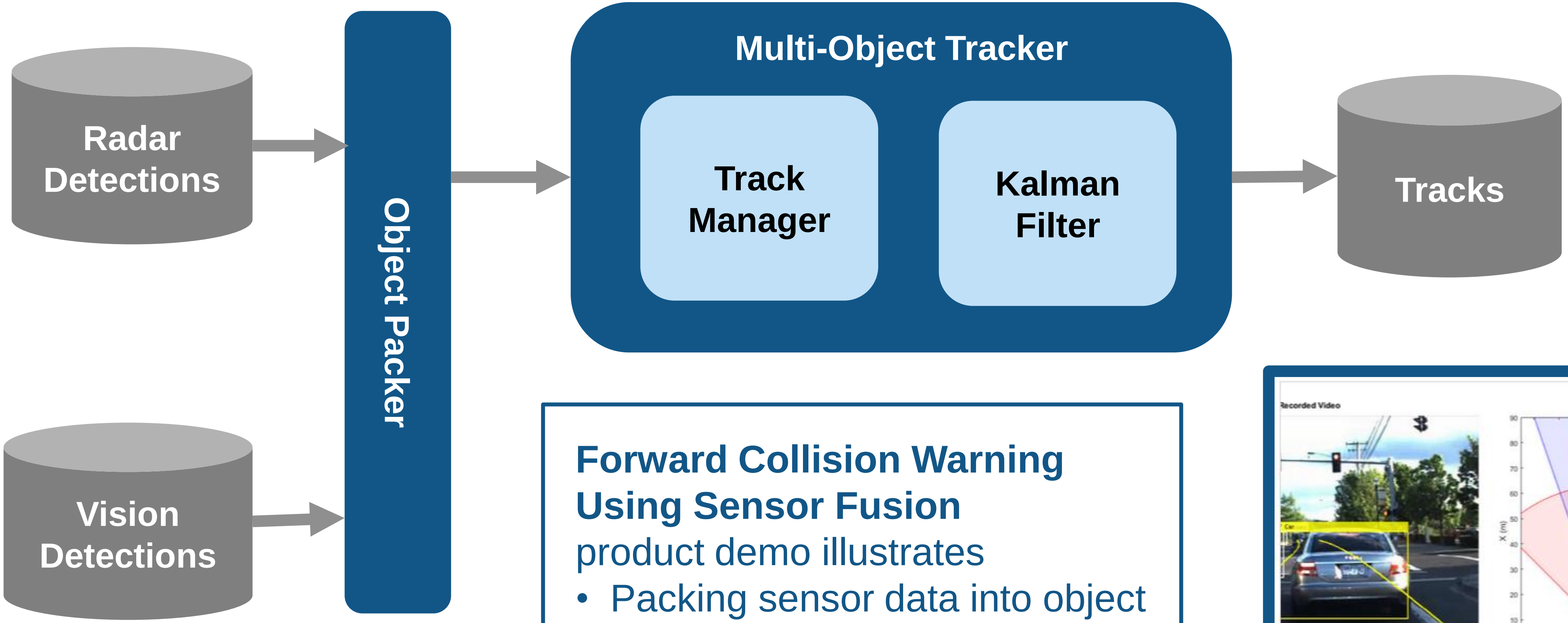
Time
Measurement
Measurement Noise

- 将检测目标赋给跟踪器
- 创建新的跟踪器
- 更新已有的跟踪器
- 删除无效的跟踪器

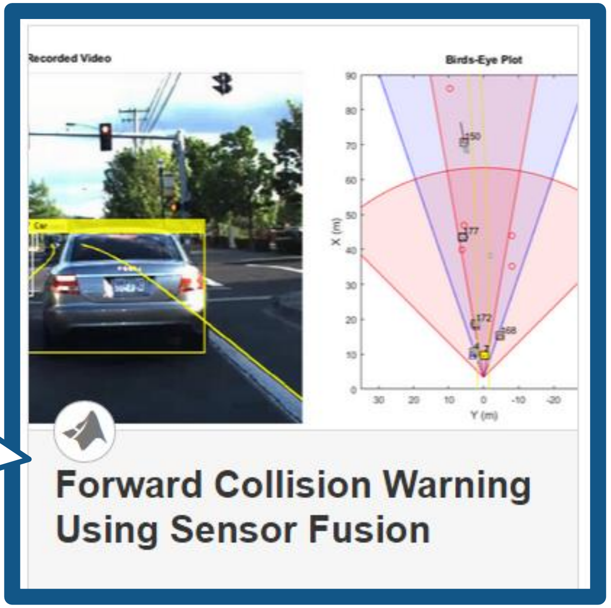
- 预测和更新跟踪器状态
- 支持linear, extended, and unscented Kalman filters

Time
State
State Covariance
Track ID
Age
Is Confirmed
Is Coasted

通过例子了解更多传感器融合



Forward Collision Warning Using Sensor Fusion
product demo illustrates
- Packing sensor data into object detections
- Initializing Kalman filter
- Configuring multi-object tracker



将算法自动生成C代码

with MATLAB Coder

trackingForFCW_kernel.m

```matlab
function [confirmedTracks, egoLane, numTracks, mostImportantObject] = ...
    trackingForFCW_kernel(visionObjects, radarObjects, inertialMeasurementUnit, ...
    laneReports, egoLane, time, positionSelector, velocitySelector)
```

Generate C code with
codegen

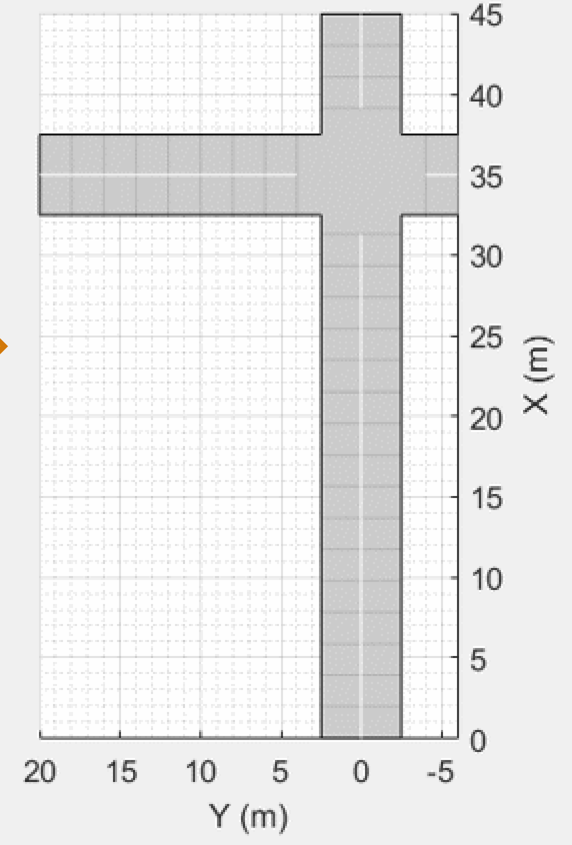
File: trackingForFCW_kernel.c

```c
 */
void trackingForFCW_kernel(const struct0_T *visionObjects, const struct2_T
  *radarObjects, const struct4_T *inertialMeasurementUnit, const struct5_T
  *laneReports, struct7_T *egoLane, double time, const double positionSelector
  [12], const double velocitySelector[12], emxArray_struct8_T *confirmedTracks,
  double *numTracks, struct10_T *mostImportantObject)
```

指定驾驶场景和道路

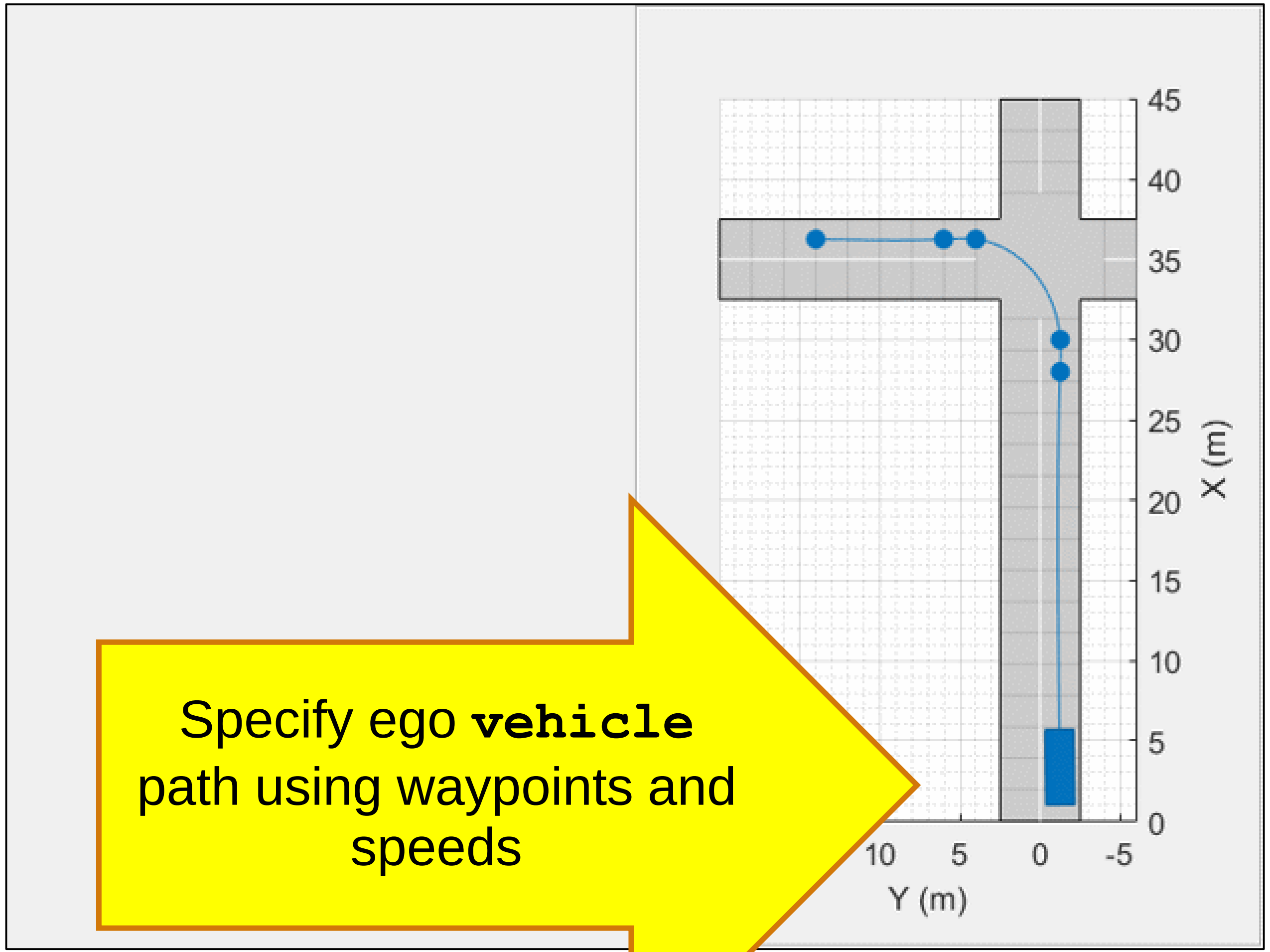
```
%% Create a new scenario
s = drivingScenario('SampleTime', 0.05);

%% Create road
road(s, [ 0  0; ...  % Centers [x,y] (m)
         45  0],...
         5);          % Width (m)
road(s, [35  20; ...
         35 -10],...
         5);

%% Plot scenario
p1 = uipanel('Position',[0.5 0 0.5 1]);
a1 = axes('Parent',p1);
plot(s,'Parent',a1,...
    'Centerline','on','Waypoints','on')
a1.XLim = [0 45];
a1.YLim = [-6 20];
```

Specify **road** centers and width as part of a **drivingScenario**



增加车辆（本车）

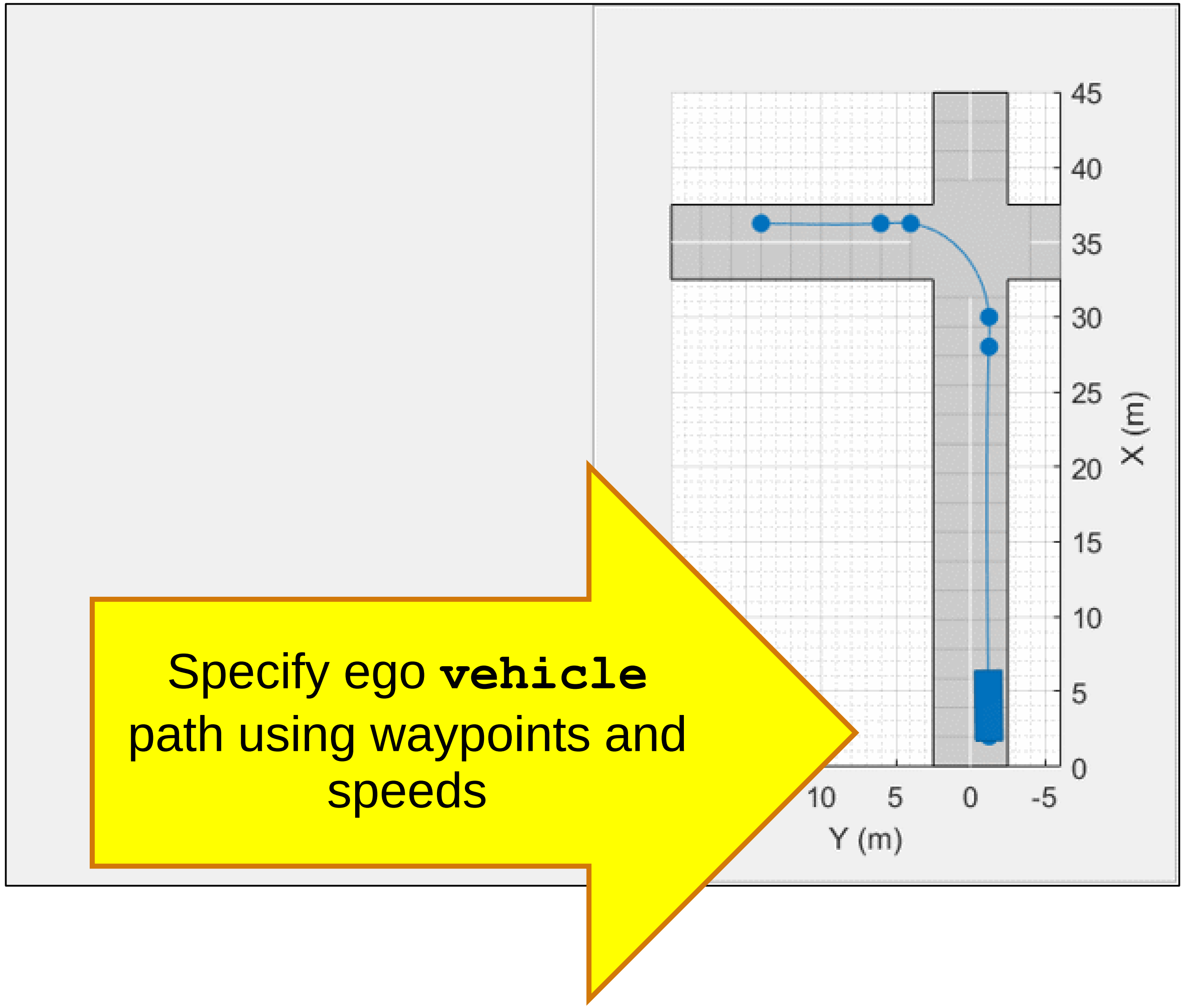
```
%% Add ego vehicle
egoCar = vehicle(s);
waypoints = [ 2   -1.25;... % [x y]  (m)
             28   -1.25;...
             30   -1.25;...
             36.25    4;...
             36.25    6;...
             36.25   14];
speed = 13.89; % (m/s) = 50 km/hr
path(egoCar, waypoints, speed);
```



Specify ego **vehicle** path using waypoints and speeds

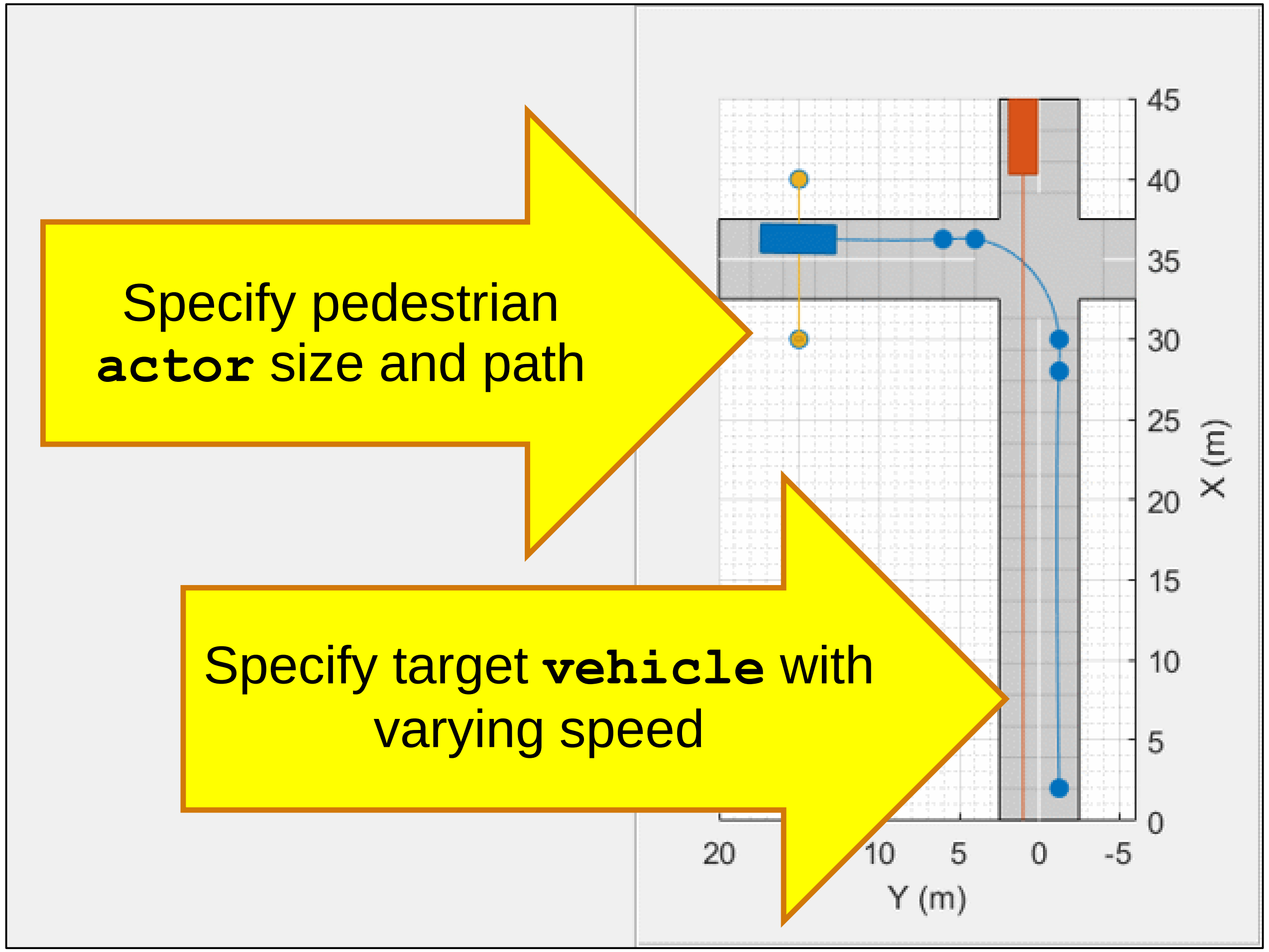
增加车辆（本车）

```
%% Add ego vehicle
egoCar = vehicle(s);
waypoints = [ 2   -1.25;... % [x y]  (m)
             28   -1.25;...
             30   -1.25;...
             36.25    4;...
             36.25    6;...
             36.25   14];
speed = 13.89; % (m/s) = 50 km/hr
path(egoCar, waypoints, speed);

%% Play scenario
while advance(s)
    pause(s.SampleTime);
end
```



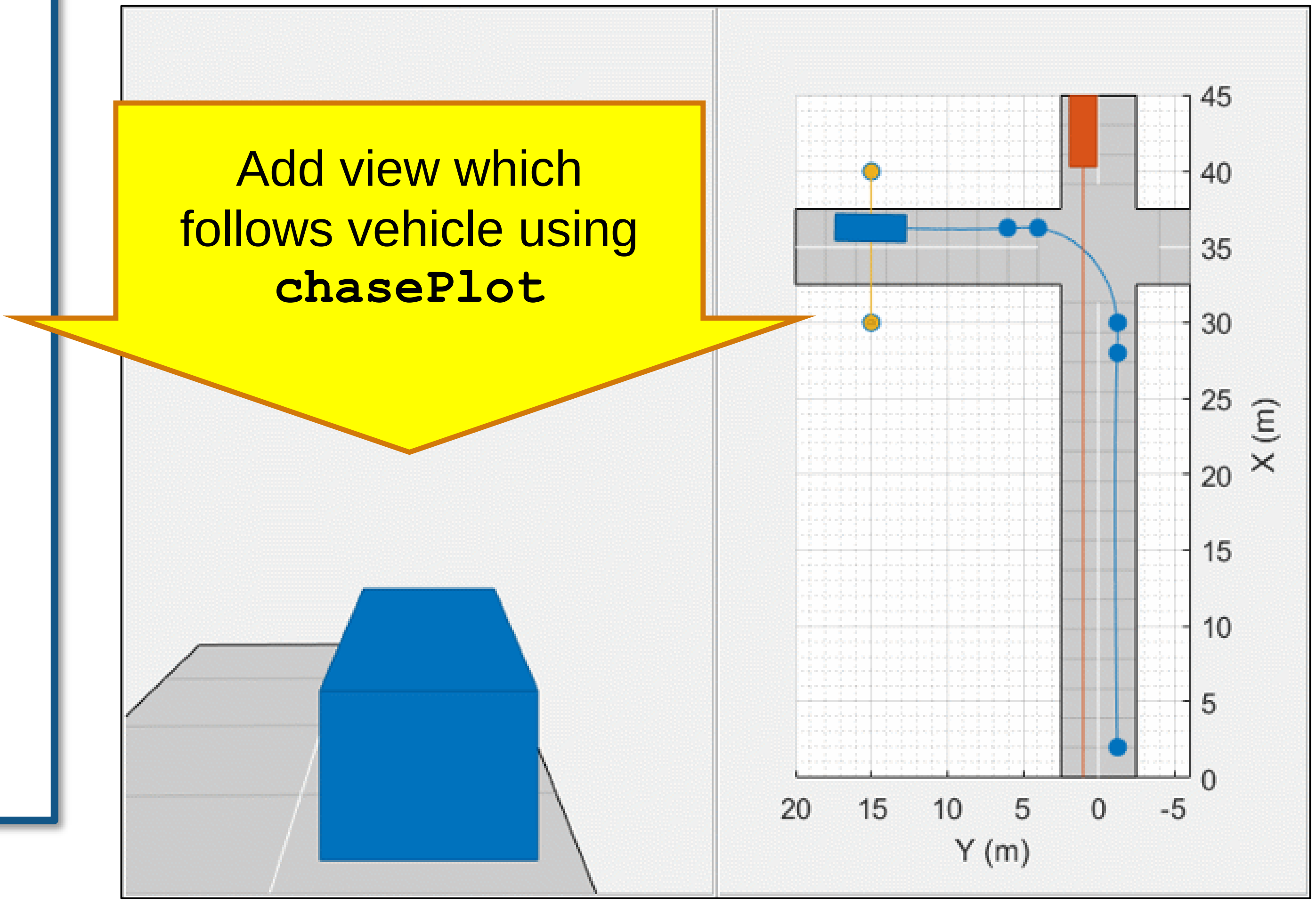
增加目标车辆和行人参与者

```
%% Add Target vehicle
targetVehicle = vehicle(s);
path(targetVehicle,...
      [44 1; -4 1],... % Waypoints (m)
      [5  ; 14]);      % Speeds (m/s)


%% Add child pedestrian actor
child = actor(s,'Length',0.24,...
                'Width',0.45,...
                'Height',1.7,...
                'Position',[40 -5 0],...
                'Yaw',180);
path(child,...
    [30 15; 40 15],... % Waypoints (m)
    1.39); % Speed (m/s) = 5 km/hr
```



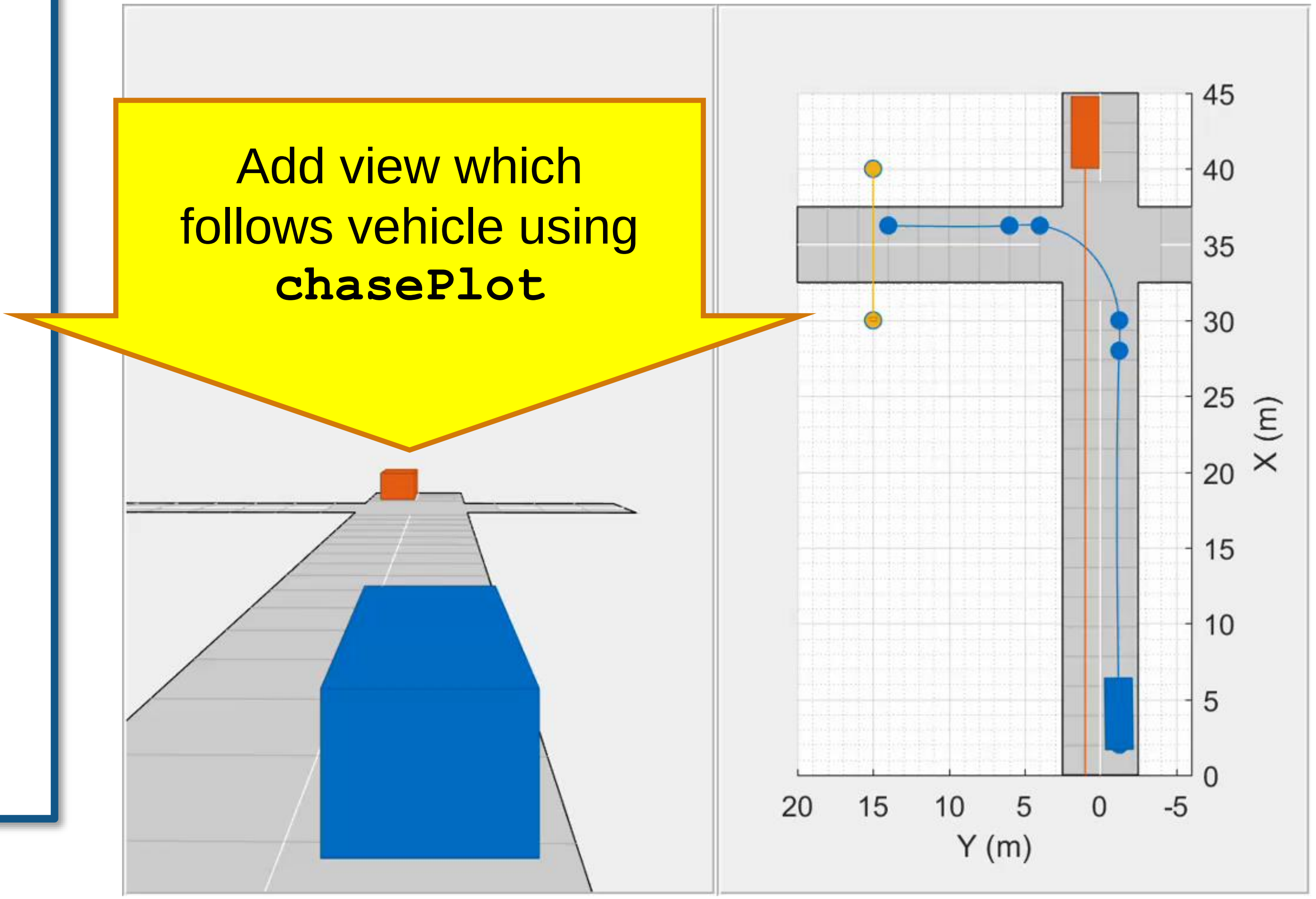
以本车后方的视角观察场景

```
%% Add chase view (left)
p2 = uipanel('Position',[0 0 0.5 1]);
a2 = axes('Parent',p2);
chasePlot(egoCar,...
    'Parent',a2,...
    'Centerline','on',...
    'ViewHeight',3.5,...     % (m)
    'ViewLocation',[-8 0]); % [x y] (m)
```



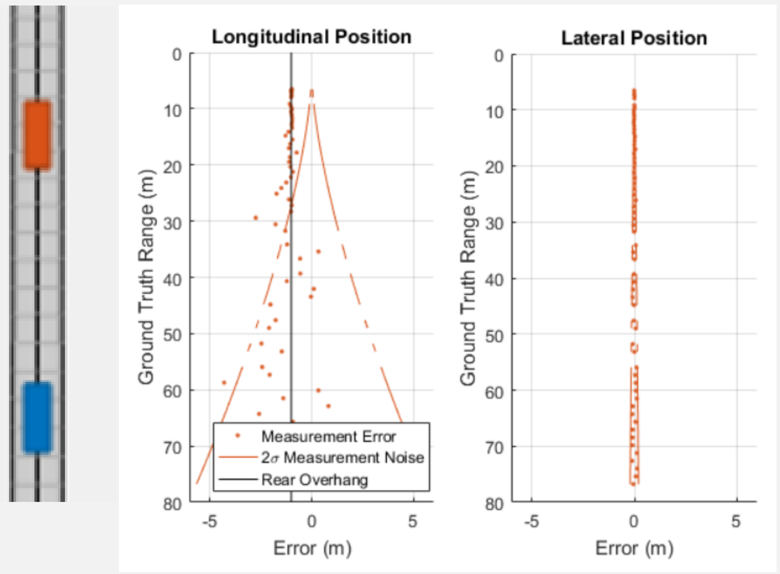
以本体车辆的后方视角观察场景 vehicle

```
%% Add chase view (left)
p2 = uipanel('Position',[0 0 0.5 1]);
a2 = axes('Parent',p2);
chasePlot(egoCar,...
    'Parent',a2,...
    'Centerline','on',...
    'ViewHeight',3.5,...    % (m)
    'ViewLocation',[-8 0]); % [x y] (m)

%% Play scenario
restart(s)
while advance(s)
    pause(s.SampleTime);
end
```



仿真视觉传感器目标检测的效应

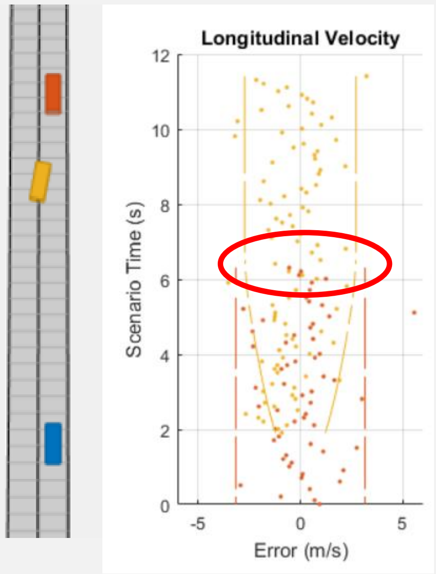
距离效应



距离测量精度随着目标距离增加而降低

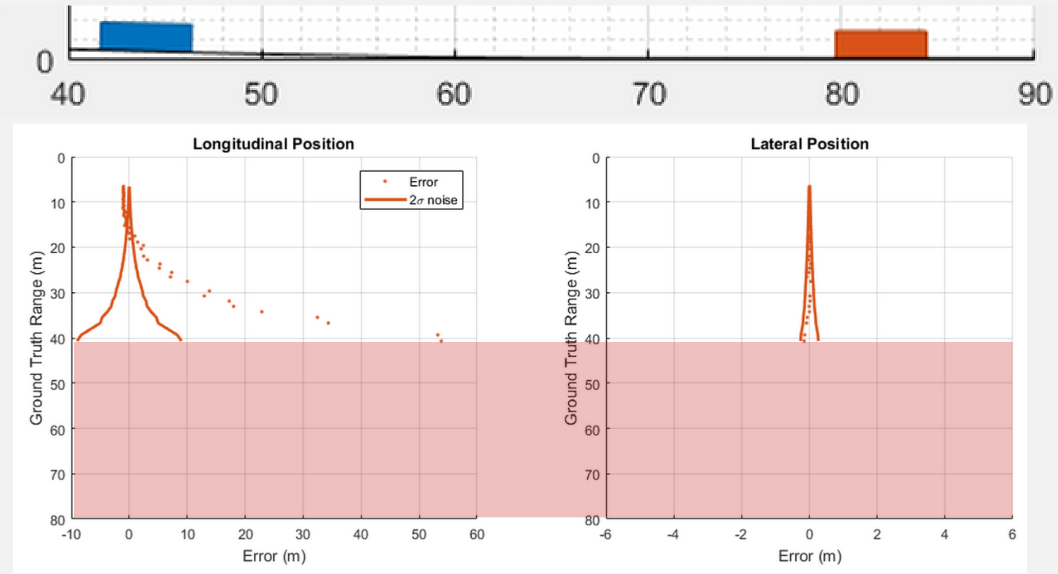
角度测量精度在覆盖范围内保持一致

阻挡效应



部分或完全被阻挡的目标无法被检测到

路面抬升效应



在覆盖区域内的目标可能没有被检测到，因为他们出现在地平线上方

检测到的目标可能也有比较大的距离测量误差

建模视觉传感器

```
%% Create vision detection generator
sensor = visionDetectionGenerator(...
    'SensorLocation', [0.75*egoCar.Wheelbase 0], ...
    'Height', 1.1, ...
    'Pitch', 1, ...
    'Intrinsics', cameraIntrinsics(...
        800,...              % Focal length
        [320 240],...        % Principal point
        [480 640]), ...      % Image size
        'RadialDistortion',[0 0], ...
        'TangientialDistortion',[0 0]), ...
    'UpdateInterval', s.SampleTime, ...
    'BoundingBoxAccuracy', 5, ...
    'MaxRange', 150, ...
    'ActorProfiles', actorProfiles(s));
```

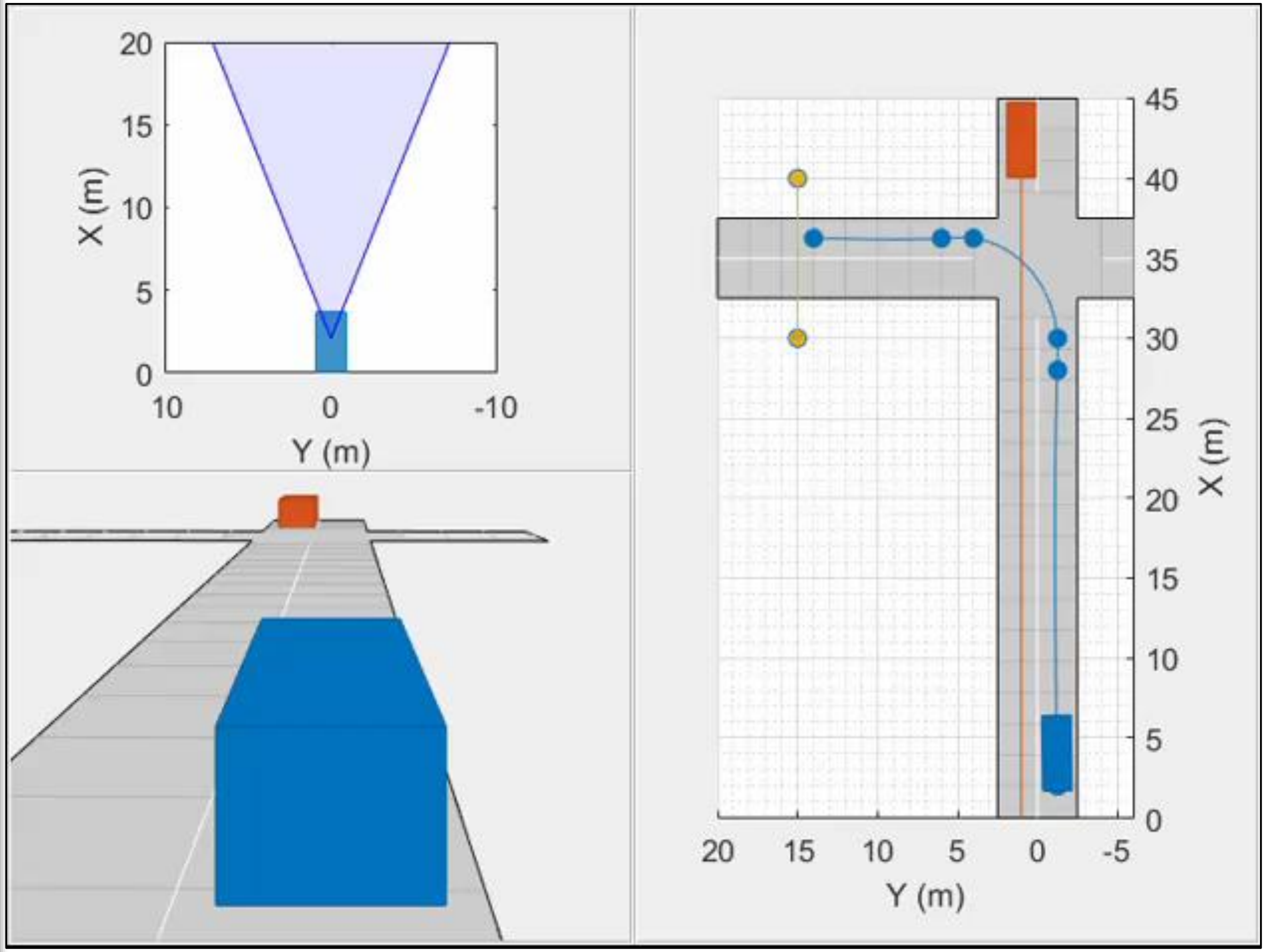
Extrinsic mounting parameters

Coverage area is determined based on **cameraIntrinsics**

Model radar detection sensor using **radarDetectionGenerator**

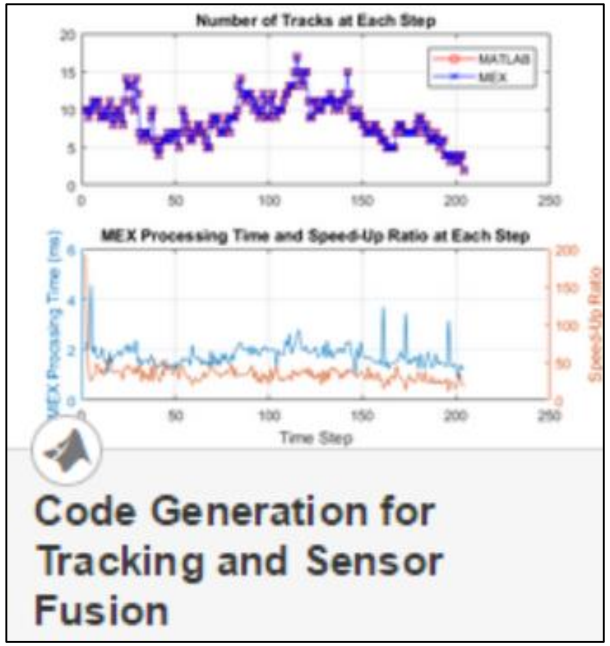
带着传感器模型运行场景

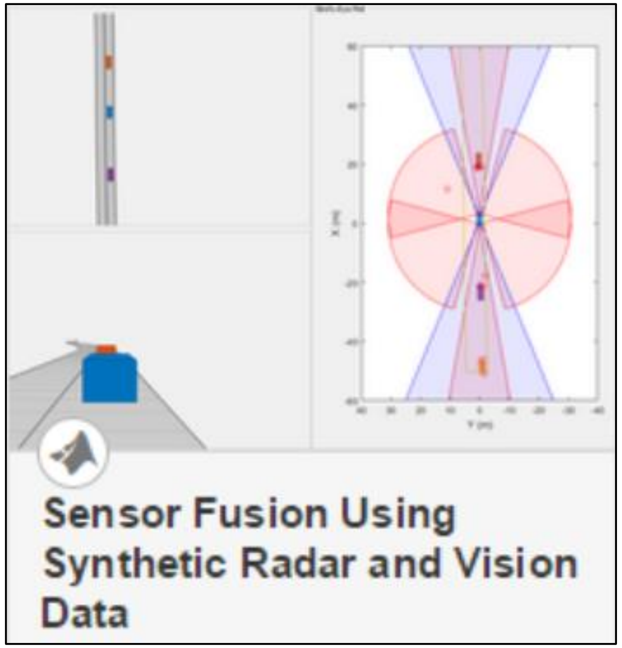
```
restart(s)
while advance(s)
   % Get detections in ego vehicle coordinates
   det = sensor(targetPoses(egoCar),...
                s.SimulationTime);
   % Update plotters
   if isempty(det)
       clearData(detPlot)
   else % Unpack measurements to position/velocity
       pos = cellfun(@(d)d.Measurement(1:2),...
                     det,'UniformOutput',false);
       vel = cellfun(@(d)d.Measurement(4:5),...
                     det,'UniformOutput',false);
       plotDetection(detPlot,...
           cell2mat(pos')', cell2mat(vel')');
   end
   [p, y, l, w, oo, c] = targetOutlines(egoCar);
   plotOutline(truthPlot,p,y,l,w,...
       'OriginOffset', oo, 'Color', c);
end
```



了解更多传感器融合

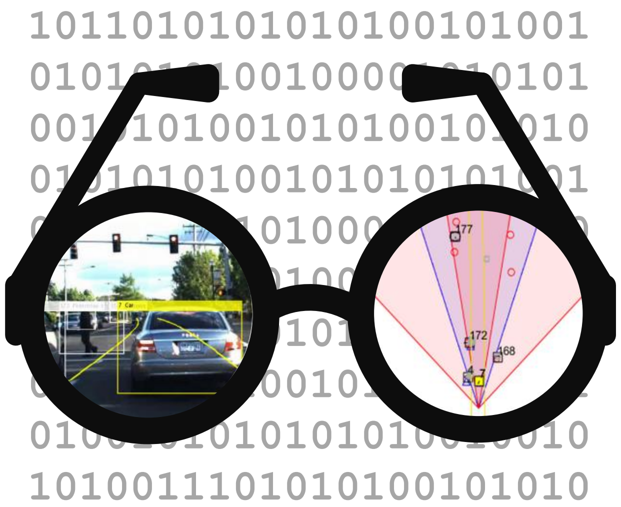
查看Automated Driving System Toolbox中的例子





- **设计**
  基于记录的车辆数据
  设计目标跟踪器
- **生成 C/C++代码**
  将多目标跟踪器
  生成代码
- **合成驾驶场景**
  测试多目标跟踪器

自动驾驶工具箱(Automated Driving System Toolbox)能帮您…

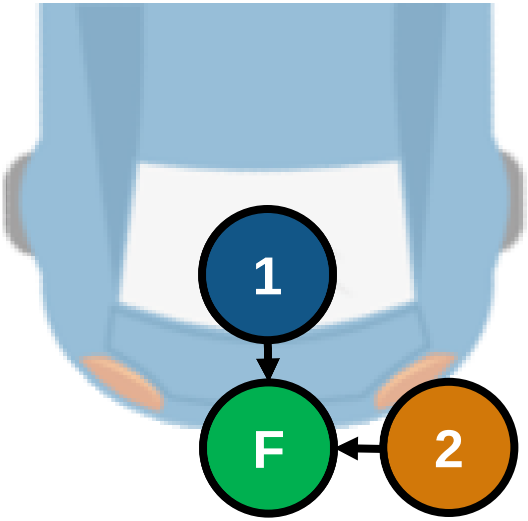


可视化车辆数据

- 绘制传感器检测结果
- 绘制覆盖范围
- 图像坐标系和车辆坐标系转换

在图像中检测目标

- 训练深度学习网络
- 标记真实值
- 连接到其他工具



融合多个检测结果

- 设计多目标跟踪器
- 生成 C/C++
- 合成驾驶场景