MATLAB EXPO 2016

Designing Perception Systems for Autonomous Driving

Avinash Nehemiah
November 3, 2016

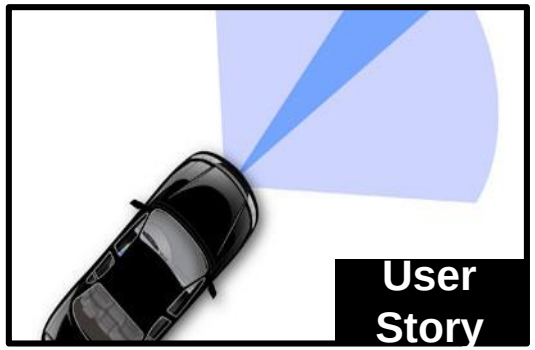
How Autonomous Driving Impact our Lives

1. Safety
   - Reduce traffic accidents 90%
   - Save 30,000 lives/year (on-par with modern vaccines)
2. Redefine logistics – substantial fuel savings
   - Reduce cost
   - Environment friendly
3. Save us from our daily commutes

Engineers in autonomous driving <u>choose MATLAB and Simulink.</u>

MATLAB and Simulink in Perception Systems for Autonomous Driving

Sensor Design



Delphi

Radar Sensor Alignment Algorithm for Automotive Active Safety System

Signal Processing

Continental

Traffic **Sign Recognition** for Driver Assistance Systems

Sensor Fusion

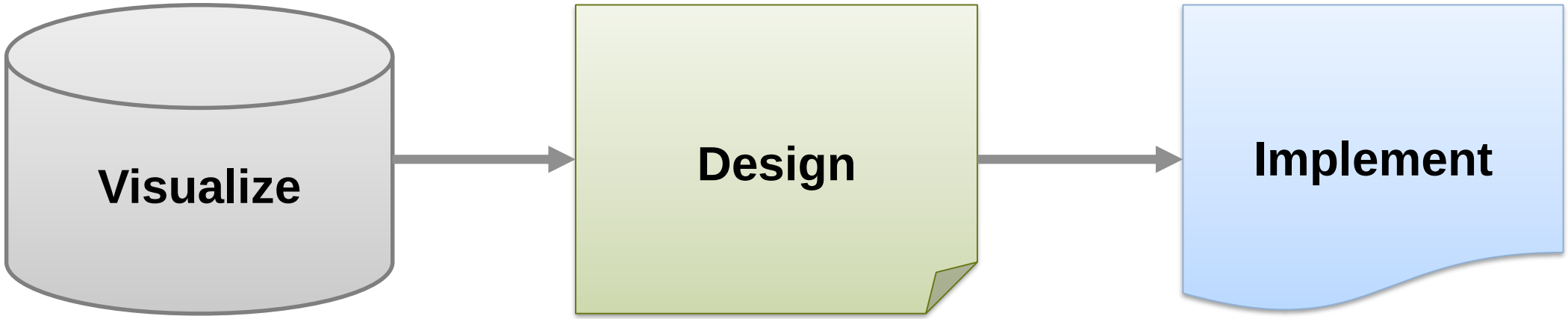
Scania

Model-Based Approach to Resource-Efficient **Object Fusion** for an Autonomous Braking System

Decision Logic

Magna

MATLAB Helps Engineers…



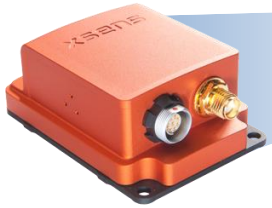
Test Vehicle Equipped with Various Sensors

Velodyne LiDAR HDL-32E
- Horizontal FoV: 360°
- Vertical FoV: +10..-30°
- Range: 80..100m
- 100 Mbps Ethernet

Mobileye 560
- FoV: 38°x150m
- CAN interface

Point Grey Blackfly
- Stand “ice cube” vision camera
- 800x600, 27FPS
- GigE interface

XSENS MTI-G-700
- Stable and sensitive
- MEMS-based AHRS
- USB interface



Delphi ESR
- 76GHz electronically scanning radar
- Dual FoVs, 90°x60m, 20°x174m
- CAN interface

POLYSYNC

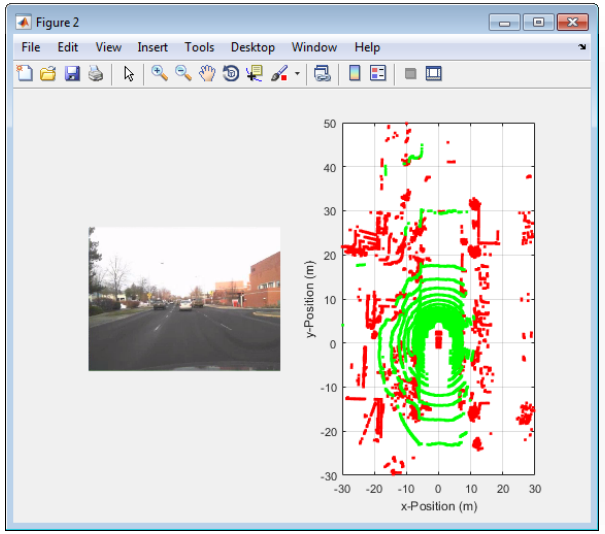
Two Examples for Today

Sensor Design

Signal Processing

Sensor Fusion

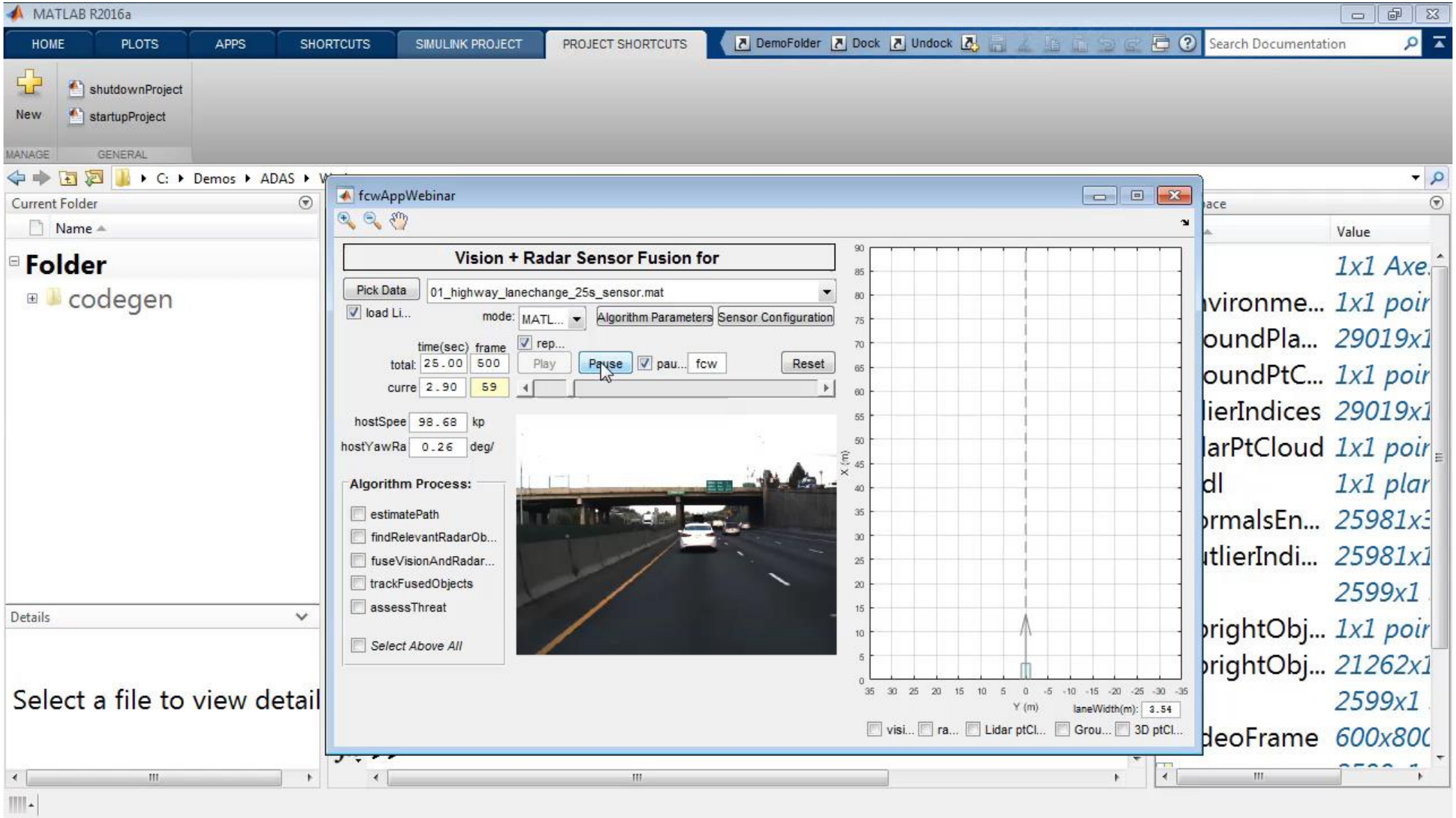
Decision Logic



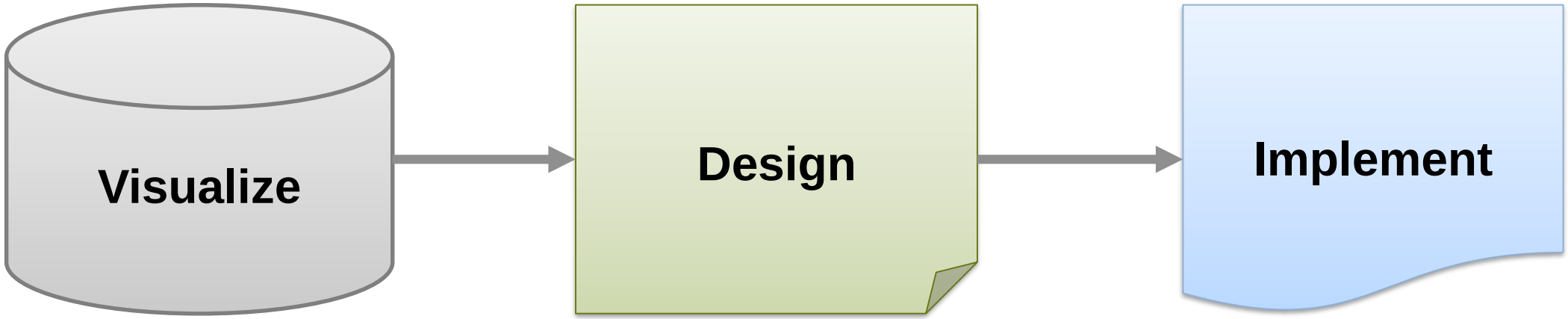
LiDAR Processing

Sensor Fusion
(Camera + Radar)

Visualize Data and Gain Insight

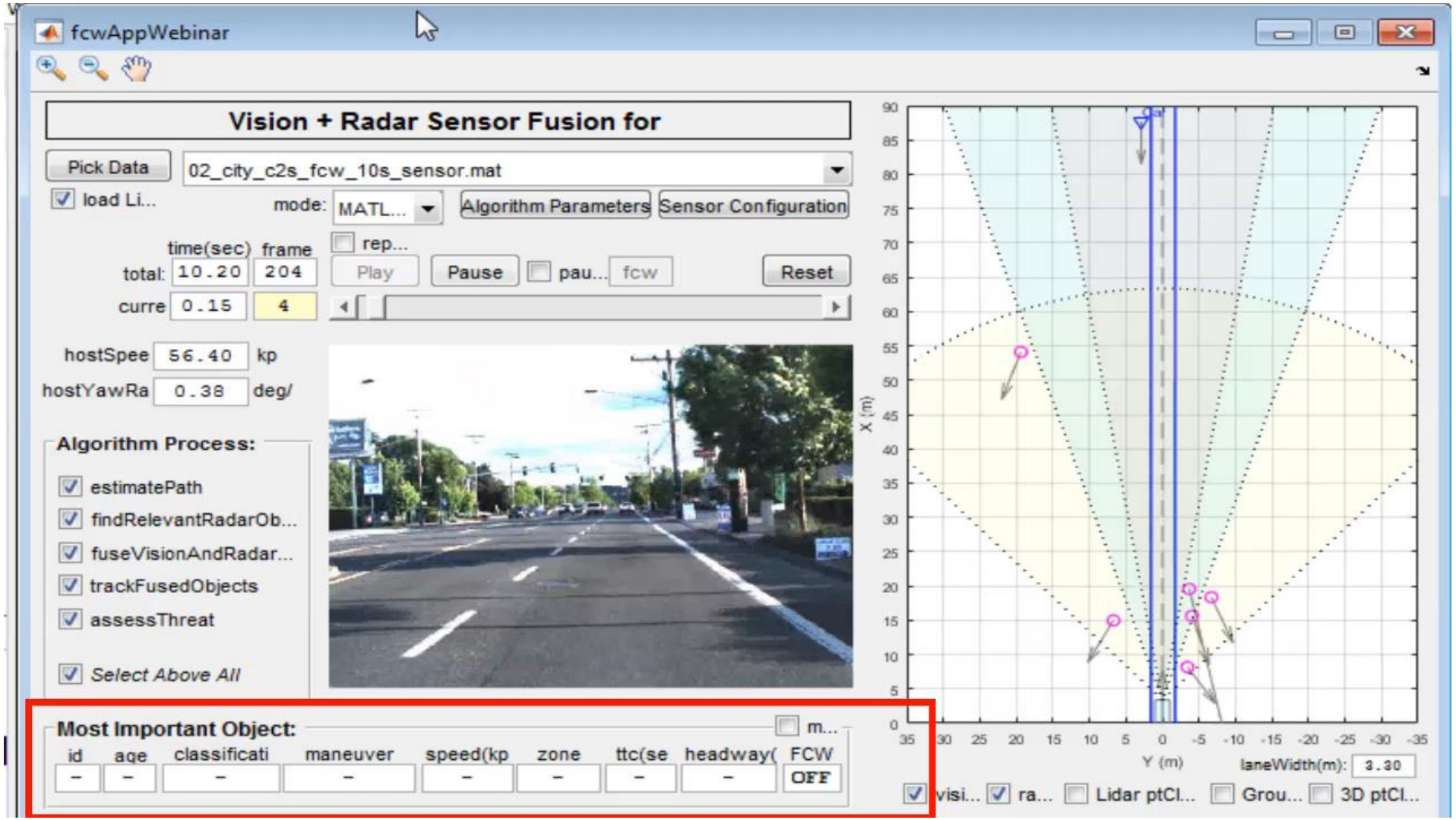


MATLAB Helped Us…

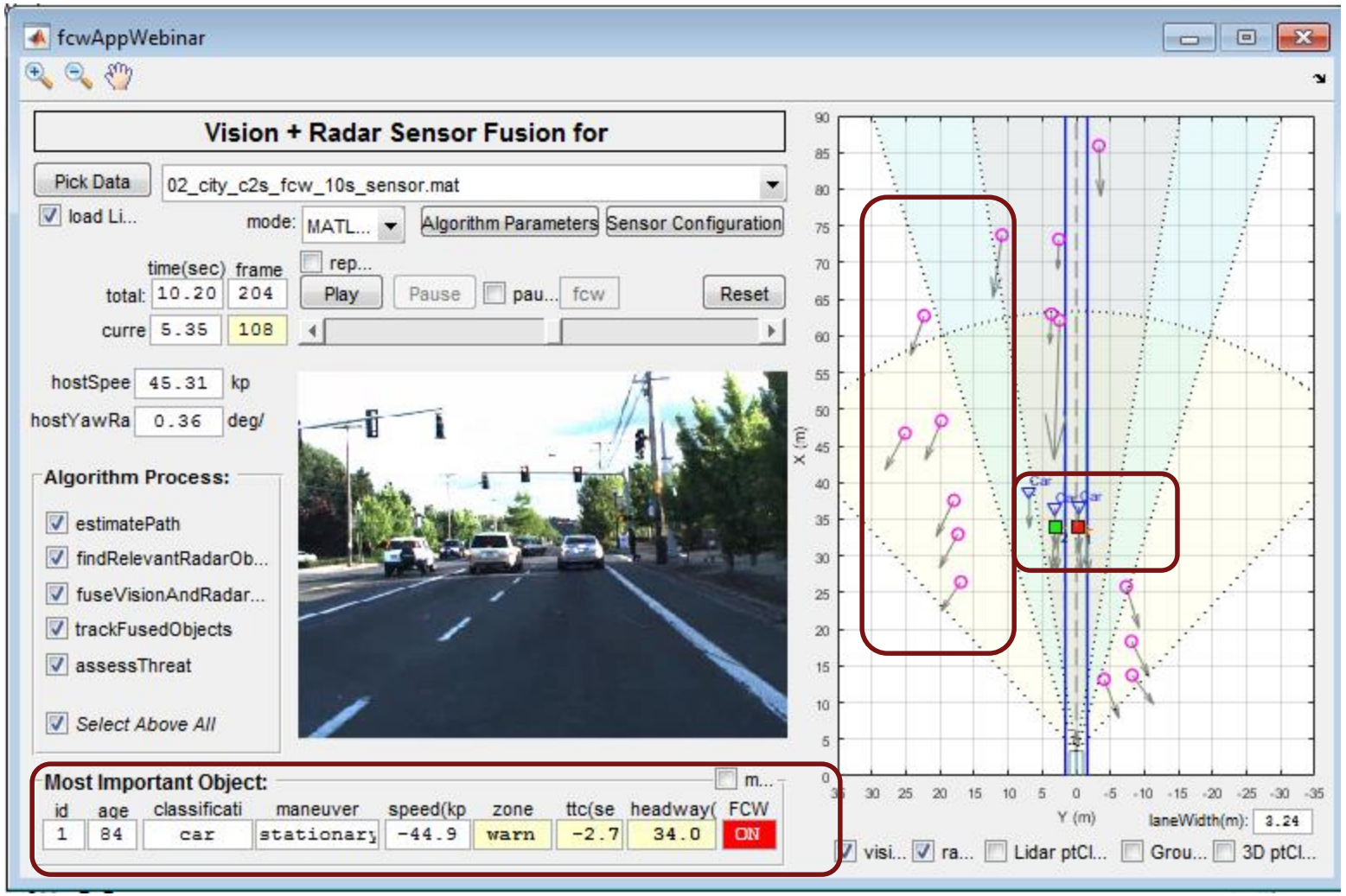


- Visualize sensors
  - Cameras
  - LiDAR
  - Radar
- Create custom apps and visualizations

Design Algorithms with MATLAB



Design Algorithms with MATLAB

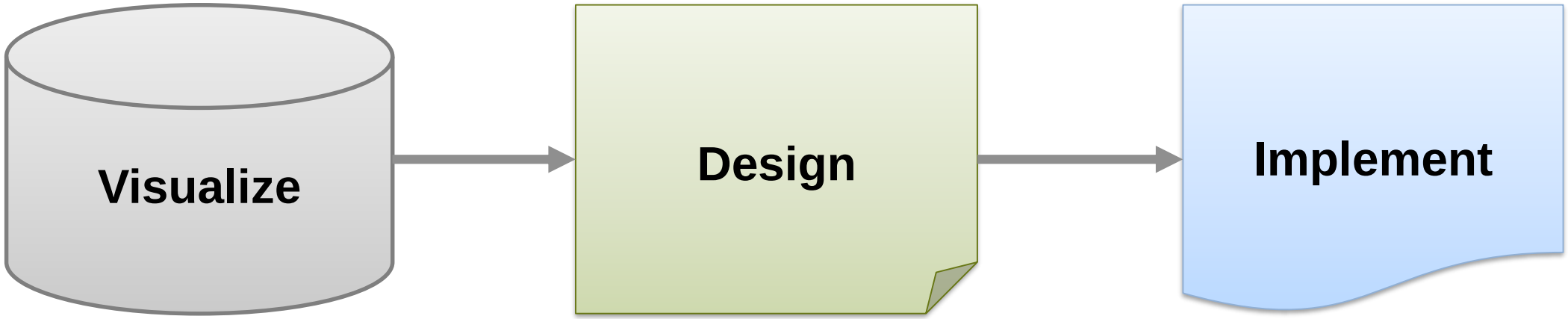


1. Filter false detections from radar

2. Sensor fusion and tracking using Kalman filters

3. Decision logic

MATLAB Helped Us…



- Algorithms for multiple domains:
  - Computer vision
  - Radar
  - Sensor fusion
- Toolboxes just work

Implement Algorithms in C Code

Code Generation Report

MATLAB code | Call stack | C code

Filter

Functions

- assessThreat
- assessThreat > assessCollisionRisk
- cartesianToPolar
- checkInPath
- estimatePath
- filterYawRate
- findMostImportantObject
- findRelevantRadarObjects
- findRelevantRadarObjects > calculateGroundSpeed
- forwardCollisionWarning
- fuseVisionAndRadarObjects
- fuseVisionAndRadarObjects > associateSensors
- fuseVisionAndRadarObjects > fusionCost
- objectZone
- trackFusedObjects
- trackFusedObjects > calculateCost
- trackFusedObjects > createNewTracks
- trackFusedObjects > deleteLostTracks
- trackFusedObjects > detectionToTrackAssignment
- trackFusedObjects > getTrackStruct
- trackFusedObjects > predictNewStateOfTracks
- trackFusedObjects > updateAssignedTracks
- trackFusedObjects > updateUnassignedTracks
- trackVisionObjects
- trackVisionObjects > calculateCost
- trackVisionObjects > createNewTracks
- trackVisionObjects > deleteLostTracks

Function: forwardCollisionWarning Calls: Select a function call:

```
function [mostImportantObject, ...
          egoPath, ...
          radarObjects, ...
          visionObjects, ...
          fusedObjects, ...
          assessedThreats] = forwardCollisionWarning( ...
                                lane, ...
                                inertialMeasurementUnit, ...
                                visionSensor, ...
                                radarSensor, ...
                                params, ...
                                reset)
% forwardCollisionWarning: sensor fusion algorithm for Forward Collision Warning
%
%   Copyright 2016 MathWorks, Inc.
%
% --- Inputs:
%       lane: lane marking data from ME 560
%           >> datainterface.lane.default
%
%       inertialMeasurementUnit: IMU data
%           >> datainterface.inertialMeasurementUnit.default
%
```

Summary | All Messages (0) | Variables | Target Build Log

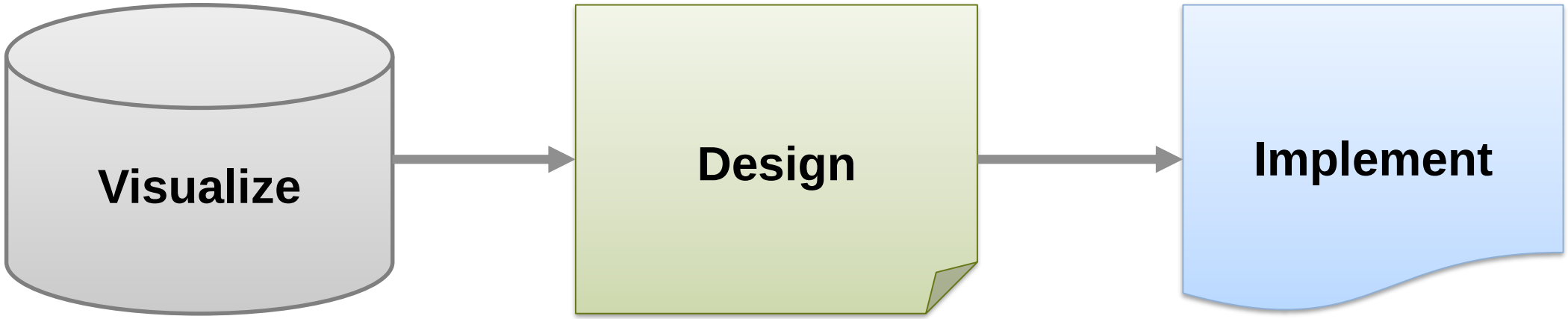
C source code generated on:	31-Oct-2016 12:20:10
Coding target:	Static Library
Number of errors:	0
Number of warnings:	0
Number of notices:	0

Tell Us What You Think

We value your feedback. Please take a few minutes to answer this short questionnaire regarding the Code Generation Report.

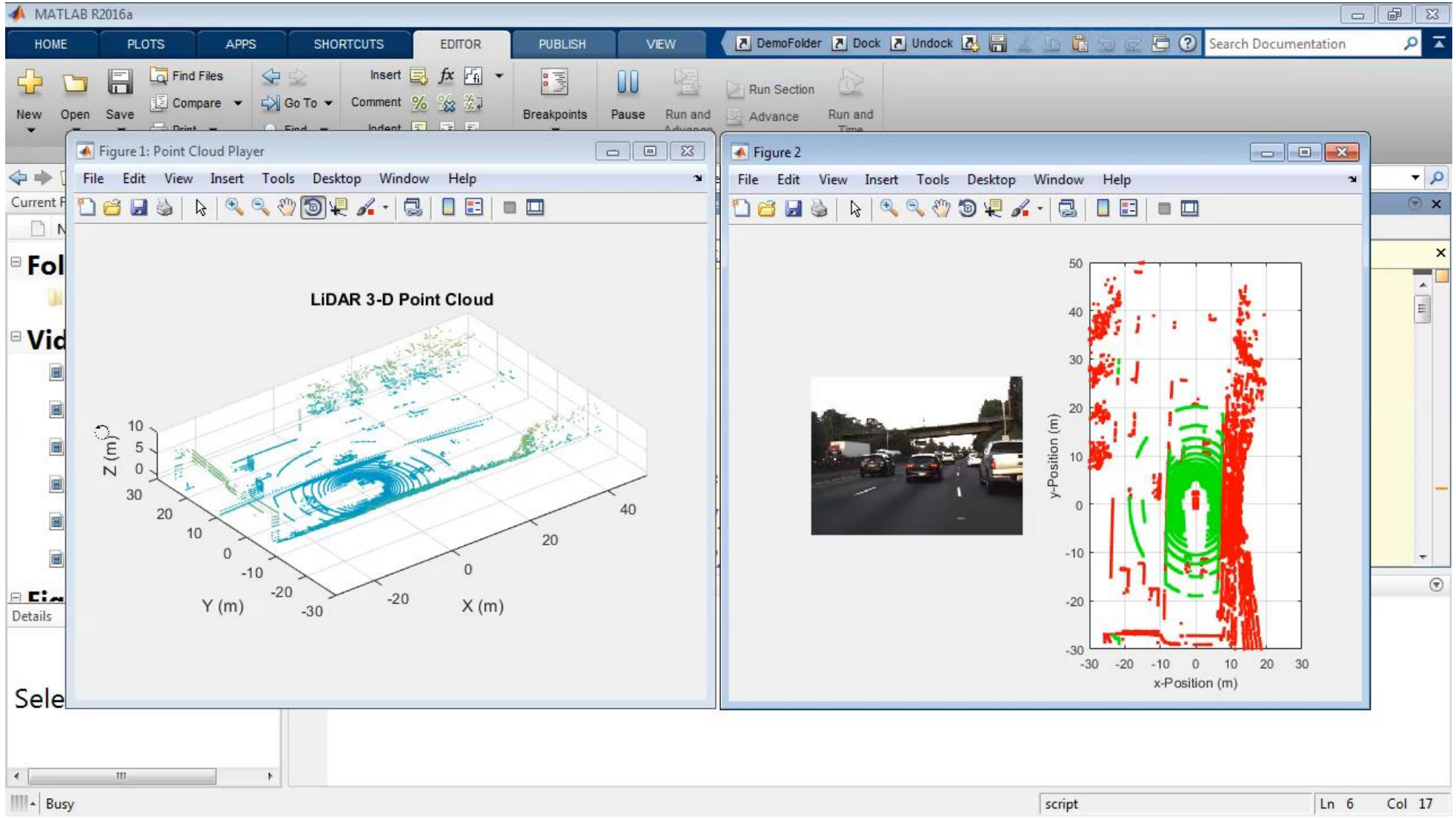
>>Provide Feedback

MATLAB Helped Us…



- Faster iteration by generating C code

Design LiDAR Processing for Autonomous Driving



Design LiDAR Processing for Autonomous Driving

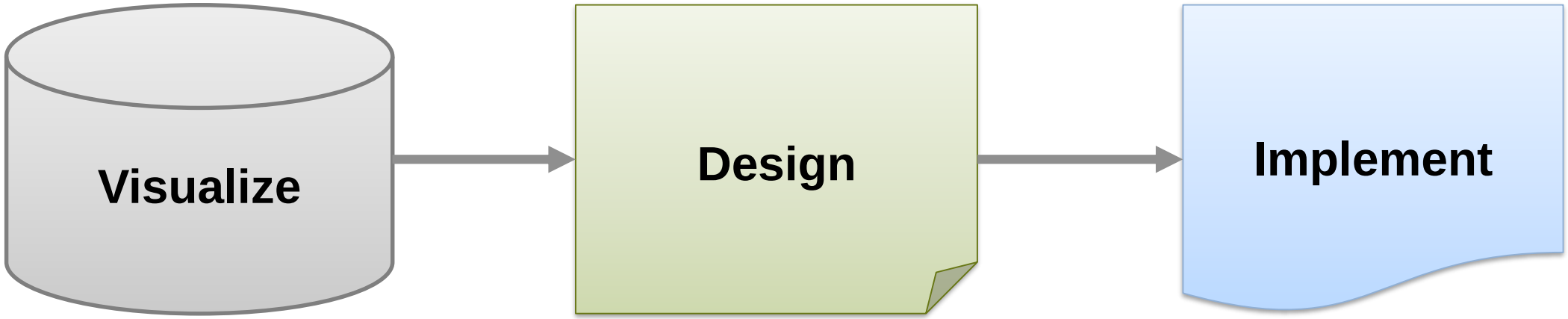
MATLAB R2016a

Editor - C:\Users\anehemia\OneDrive - MathWorks\Shared with Everyone\LiDARProcessing\LiDAR\v1.2\LiDARProcessingWorkflow.m

```
clear;
close all;
load singlePtCloud.mat;
%% Display Image
figure;
subplot(1,2,1);imshow(videoFrame);
title('Input Video Frame');
subplot(1,2,2);
pcshow(lidarPtCloud);
title('Lidar Point Cloud');


%% Find ground plane
[mdl,inlierIndices,outlierIndices] = ...
```

MATLAB Helps Engineers…



- Visualize most sensors
  - Cameras
  - LiDAR
  - Radar
- Create custom apps and visualizations

- Algorithms for multiple domains:
  - Computer vision
  - Deep learning
  - Radar
- Toolboxes just work

- Faster iteration by generating C code

See the demos in person at our booth