MATLAB EXPO 2017
KOREA
4월 27일, 서울
등록 하기 matlabexpo.co.kr
 컴퓨터비전의 최신기술
: Deep Learning, 3D Vision and Embedded Vision

김 종 남
Application Engineer
Three Main Topics

New capabilities for computer vision system design:

- Deep Learning
- 3-D Vision and Image Processing
- Embedded Vision
New MATLAB framework makes deep learning easy and accessible

and

MATLAB can be used by experts for real deep learning (computer vision) problems
What is Deep Learning?

Deep learning is a type of machine learning that performs end-to-end learning by learning tasks directly from images, text, and sound.
Two Approaches for Deep Learning

1. Train a Deep Neural Network from Scratch

2. Fine-tune a pre-trained model (transfer learning)
Example: Classify Vehicles With Transfer Learning

AlexNet Pretrained Model

- 1000 classes
- Trained on millions of images

Transfer learning – use AlexNet as starting point

Vehicle Classifier (5 Class)

- Car
- SUV
- Van
- Truck
- Large Truck

New Data
Transfer Learning to Classify New Objects
MATLAB makes Deep Learning Easy and Accessible

Learn about new MATLAB capabilities to

- Handle and label large sets of images
- Accelerate deep learning with GPU’s
- Visualize and debug deep neural networks
- Access and use models from experts
3D Image Processing
3-D Image Processing

Over **40 functions** support 3-D volumetric image processing

Capabilities Includes:
- Image arithmetic
- Morphology
- Segmentation
- Geometric transforms
- Enhancement

**Volume Viewer App** for exploration
3-D Image Processing
3D Vision – LiDAR Processing
What are Point Clouds?

- Point clouds represent a set of data points in a 3-D coordinate system.
- Typically used to measure physical world surfaces.
- Used for navigation and perception in robotics and Advanced Driver Assistance Systems (ADAS).
Common Sources of Point Cloud Data

- LIDAR
- Laser Scanner
- Depth Camera
- Stereo Camera
3-D Vision: Design LiDAR Processing
3:D Vision: Design LiDAR Processing

clear;
close all;
load singlePtCloud.mat;

```matlab
% 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] = ...
Embedded Vision System Development using Automatic Code Generation
Typical Workflow for Embedded Vision System Development

Algorithm Development
- Is my idea new? What is required?
- Is it robust to all kinds of conditions? (lighting noise, etc.)

Implementation
- Consideration of HW platform
  - FPGA? CPU? DSP? GPU?
- Speed and resource requirement
  - Resolution, Frame-rate constraint
  - Memory constraint

Development of the algorithm and implementation are often done by different groups
MATLAB Coder app with Integrated Editor and Simplified Workflow

New user interface simplifies code generation workflow
Embedded Coder for Optimized Code

Embedded Coder extends MATLAB Coder with:

- Processor-specific code generation
  - Built-in support for select processors
  - Open APIs for use with any processor
- Speed, memory, and code appearance advanced features
MATLAB Language Support for Code Generation
### Supported MATLAB Language Features and Functions

A broad set of language features and functions/system objects supported for code generation.

<table>
<thead>
<tr>
<th>Matrices and Arrays</th>
<th>Data Types</th>
<th>Programming Constructs</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Matrix operations</td>
<td>• Complex numbers</td>
<td>• Arithmetic, relational, and logical operators</td>
<td>• MATLAB functions and subfunctions</td>
</tr>
<tr>
<td>• N-dimensional arrays</td>
<td>• Integer math</td>
<td>• Program control</td>
<td>• Variable-length argument lists</td>
</tr>
<tr>
<td>• Subscripting</td>
<td>• Double/single-precision</td>
<td>(if, for, while, switch)</td>
<td>• Function handles</td>
</tr>
<tr>
<td>• Frames</td>
<td>• Fixed-point arithmetic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Persistent variables</td>
<td>• Characters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Global variables</td>
<td>• Structures</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Cell arrays</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Numeric class</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Variable-sized data</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• MATLAB Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• System objects</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Supported algorithms:
- More than 1300 MATLAB operators, functions, and System objects for:
  - Communications
  - Computer vision
  - Image processing
  - Neural networks
  - Phased Array signal processing
  - Robotics
  - Signal processing
  - Statistics and machine learning
Automatic Translation of MATLAB to C

With MATLAB Coder, design engineers can:

- Maintain one design in MATLAB
- Design faster and get to C quickly
- Test more systematically and frequently
- Spend more time improving algorithms in MATLAB
Vision HDL Toolbox

*Design and prototype video image processing systems*

- Modeling hardware behavior of the algorithms
  - Pixel-based functions and blocks
  - Conversion between frames and pixels
  - Standard and custom frame sizes

- Prototyping algorithms on hardware
  - *(With HDL Coder)* Efficient and readable HDL code
  - *(With HDL Verifier)* FPGA-in-the-loop testing and acceleration
Pixel Based Video Image Algorithms

- **Analysis & Enhancement**
  - Edge Detection, Median Filter

- **Conversions**
  - Chroma Resampling, Color-Space Converter
  - Demosaic Interpolator, Gamma Corrector, Look-up Table

- **Filters**
  - Image Filter, Median Filter

- **Morphological Operations**
  - Dilation, Erosion,
  - Opening, Closing

- **Statistics**
  - Histogram
  - Image Statistics

- **I/O Interfaces**
  - Frame to Pixels, Pixels to Frame, FIL versions

- **Utilities**
  - Pixel Control Bus Creator
  - Pixel Control Bus Selector
Frame To Pixels and Pixels To Frame
A Complete Solution for Embedded Vision

Frame based

Pixel based

- Computer Vision System Toolbox
- Image Processing Toolbox
- Vision HDL Toolbox

- HDL Coder
- MATLAB Coder
- Fixed Point Designer
- HDL Verifier

MATLAB / Simulink
FLIR Accelerates Development of Thermal Imaging FPGA

**Challenge**
Accelerate the implementation of advanced thermal imaging filters and algorithms on FPGA hardware

**Solution**
Use MATLAB to develop, simulate, and evaluate algorithms, and use HDL Coder to implement the best algorithms on FPGAs

**Results**
- Time from concept to field-testable prototype reduced by 60%
- Enhancements completed in hours, not weeks
- Code reuse increased from zero to 30%

“With MATLAB and HDL Coder we are much more responsive to marketplace needs. We now embrace change, because we can take a new idea to a real-time-capable hardware prototype in just a few weeks. There is more joy in engineering, so we’ve increased job satisfaction as well as customer satisfaction.”

Nicholas Hogasten
FLIR Systems