MATLAB and C/C++
The perfect combination for signal processing

```matlab
% Video Reader
hrf = vision.VideoFileReader('visionDeparture.avi', ...
    'VideoOutputDataType', 'uint8');

% Create a [ColorSpaceConverter] System object to ...'
IColor = vision.ColorSpaceConverter( ...'
    'Conversion', 'RGB to intensity');

% Create Blob Detector
bBlob = vision.BlobAnalysis( ...'
    'MinimumBlobArea', 10, ...'
    'MajorAxisLengthOutputPort', true, ...'
    'MinorAxisLengthOutputPort', true, ...'
    'OrientationOutputPort', true, ...'
    'CentroidsOutputPort', true, ...'
    'BoundingBoxOutputPort', false, ...'
    'LabelMatrixOutputPort', false);

% Create Shape Inserter
hShape = vision.ShapeInserter( ...'
    'Shape', 'Lines', ...'
    'BorderColor', 'Custom', ...'
```
Signal Processing Algorithm Design with C/C++
You can enhance your C and C++ development process using the data analysis, visualization, algorithm development, code generation, deployment, test and verification capabilities of MATLAB.
Signal Processing Algorithm Design with C/C++ and MATLAB

- Specification
- Algorithm Development
  - C/C++
  - MATLAB
- Specialized Libraries
- Algorithm Development
- Testing & Debugging
- Implementation & Deployment
- Deployment
In This Presentation

- Combining MATLAB and C/C++ for
  - Test and verification
  - Algorithm development, research, design
  - Implementation and deployment

- Examples from
  - Wireless communications
  - Image processing and computer vision
  - Audio and digital signal processing
MATLAB and C/C++
Use Case 1: Call MATLAB from C/C++

- MATLAB Engine Interface
  - Interactive Algorithm Development
  - Debugging of C or MATLAB Code
  - Plotting / Visualization
  - Connect to Data Acquisition, Test and Measurement Systems
Bluetooth Example

- Find Engineering Defects in C with MATLAB
- Plot and Visualize Data in MATLAB
Bluetooth Example
Test and Debug Your C Project and Visualize Data/Results

- **Add** header file and libraries to your project
- **Package** data into MATLAB data structure (mxArray)
- **Use** engine routines to:
  - Send data to MATLAB
  - Perform computations in MATLAB
  - Plot data in MATLAB

- For C++ example, see:
  - [www.codeproject.com/samples/matlabeng.asp](http://www.codeproject.com/samples/matlabeng.asp)
MATLAB and C/C++
Use Case 2: Generate C code from MATLAB

- MATLAB Coder
  - Generate C from MATLAB
  - Integrate with your C/C++ source
  - Build and deploy complete applications

- No royalty or deployment restrictions on code generated with MATLAB Coder
Simple C Code Generation Example

MATLAB: 
\[ a = b \times c; \]

C: 
?
Implementation Considerations

- MATLAB
  - Polymorphic datatypes
  - Automatic memory management
  - Built-in matrix & array support

- C
  - Explicit datatype declaration & handling
  - Define and manage:
    Data size, memory, matrices, arrays,…

Kalman Filter Example
MATLAB Code: 7 lines
C Code: 107 lines
Lane Markings Example

- Generate C from MATLAB and integrate with C/C++
Lane Markings Example
Generate C from MATLAB

- **Prepare** your MATLAB algorithm
  - Make implementation choices
  - Use supported language features

- **Test** if your MATLAB code is compliant
  - Validate that MATLAB program generates code
  - Accelerate execution of user-written algorithm

- **Generate** source code or MEX
  - Iterate your MATLAB code to optimize
  - Implement as source, executable or library
### MATLAB Language Support for Code Generation

- 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 sub-functions</td>
</tr>
<tr>
<td>• N-dimensional arrays</td>
<td>• Integer math</td>
<td></td>
<td>• Variable length argument lists</td>
</tr>
<tr>
<td>• Subscripting</td>
<td>• Double/single-precision</td>
<td></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>Supported algorithms</td>
</tr>
<tr>
<td>• Global variables</td>
<td>• Structures</td>
<td></td>
<td>• &gt; 400 MATLAB operators and functions</td>
</tr>
<tr>
<td></td>
<td>• Numeric classes</td>
<td></td>
<td>• &gt; 200 System objects for</td>
</tr>
<tr>
<td></td>
<td>• Variable-sized data</td>
<td></td>
<td>• Signal processing</td>
</tr>
<tr>
<td></td>
<td>• System objects</td>
<td></td>
<td>• Communications</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Computer vision</td>
</tr>
</tbody>
</table>

Supported algorithms:
- > 400 MATLAB operators and functions
- > 200 System objects for
Use Case 3: Reuse MATLAB IP in Your C/C++ Code

- Create specialized algorithm libraries in MATLAB
  - Generate C with MATLAB Coder
  - Or, automatically create shared libraries with MATLAB Compiler or MATLAB Coder
    - DLL, LIB
  - Access MATLAB algorithms from C/C++ with DLLs or LIBs

- No royalty or deployment restrictions
Deploying Applications with MATLAB

1. MATLAB Application
2. MATLAB Compiler
3. MATLAB Compiler Runtime (MCR)
Deploying Applications with MATLAB

- Give MATLAB code to other users

- Share applications with end users who do not need MATLAB
  - Stand-alone executables
  - Shared libraries
  - Software components
MATLAB Compiler and MATLAB Coder

MATLAB Coder

MATLAB Compiler

MATLAB

c = fft(dat)
y = 20*log1
for k=1:max

.lib .dll .exe
## MATLAB Compiler and MATLAB Coder

<table>
<thead>
<tr>
<th></th>
<th>MATLAB Compiler</th>
<th>MATLAB Coder</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output</strong></td>
<td>Executable or software component/library</td>
<td>Portable and readable C source code</td>
</tr>
<tr>
<td><strong>MATLAB language support</strong></td>
<td>Full</td>
<td>Subset</td>
</tr>
<tr>
<td><strong>Additional libraries</strong></td>
<td>MATLAB Compiler Runtime (MCR)</td>
<td>None</td>
</tr>
<tr>
<td><strong>Supported toolboxes</strong></td>
<td>Most toolboxes</td>
<td>Some toolboxes</td>
</tr>
<tr>
<td><strong>License model</strong></td>
<td>Royalty-free</td>
<td>Royalty-free</td>
</tr>
<tr>
<td><strong>Extensions</strong></td>
<td>Builder Products</td>
<td>Embedded Coder</td>
</tr>
</tbody>
</table>
MATLAB and C/C++

Use Case 4: Reuse your C/C++ libraries in MATLAB

- `loadlibrary`
  - Load your library functions
  - Access your IP as custom libraries in MATLAB
  - Combine with standard MATLAB functions, scripts, System objects
MATLAB and C/C++
Use Case 5+: Reuse your C/C++ code in MATLAB

- `mex`
  - Compile and reuse your C/C++ code in MATLAB
  - Combine with standard MATLAB functions, scripts, System objects
MATLAB and C/C++: Summary of Use Cases

- Call MATLAB from C
- Generate C from MATLAB
- Use C Libraries in MATLAB
- Use C Code in MATLAB
- Generate IP libraries or deploy applications from MATLAB
You can enhance your C and C++ development process using the data analysis, visualization, algorithm development, code generation, deployment, test and verification capabilities of MATLAB.

MATLAB and C/C++: The Perfect Combination for Signal Processing
Summary

- Develop algorithms in C/C++ and MATLAB
  - MATLAB enhances your C/C++ environment
  - Wide variety of functions and tools available within MATLAB product family

- Move quickly to implementation
  - MATLAB Coder: ANSI C Code
  - MATLAB Compiler and Builders: Deployable Applications
  - Fixed-Point support

- Integrate with hardware and your C/C++. Distribute freely.
  - No need for a MATLAB license
  - No royalties for generated code
Next Steps

- For more information please contact me: giorgia.zucchelli@mathworks.com

- For an evaluation or trial please contact your account manager

- Thank you for your interest!