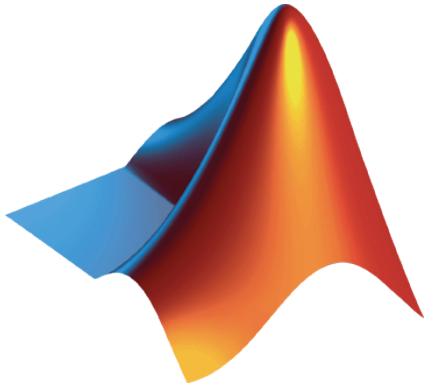
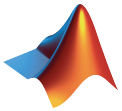
Image and Video Processing with MATLAB



Dr. Roland Michaely

Application Engineer

Agenda



Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary

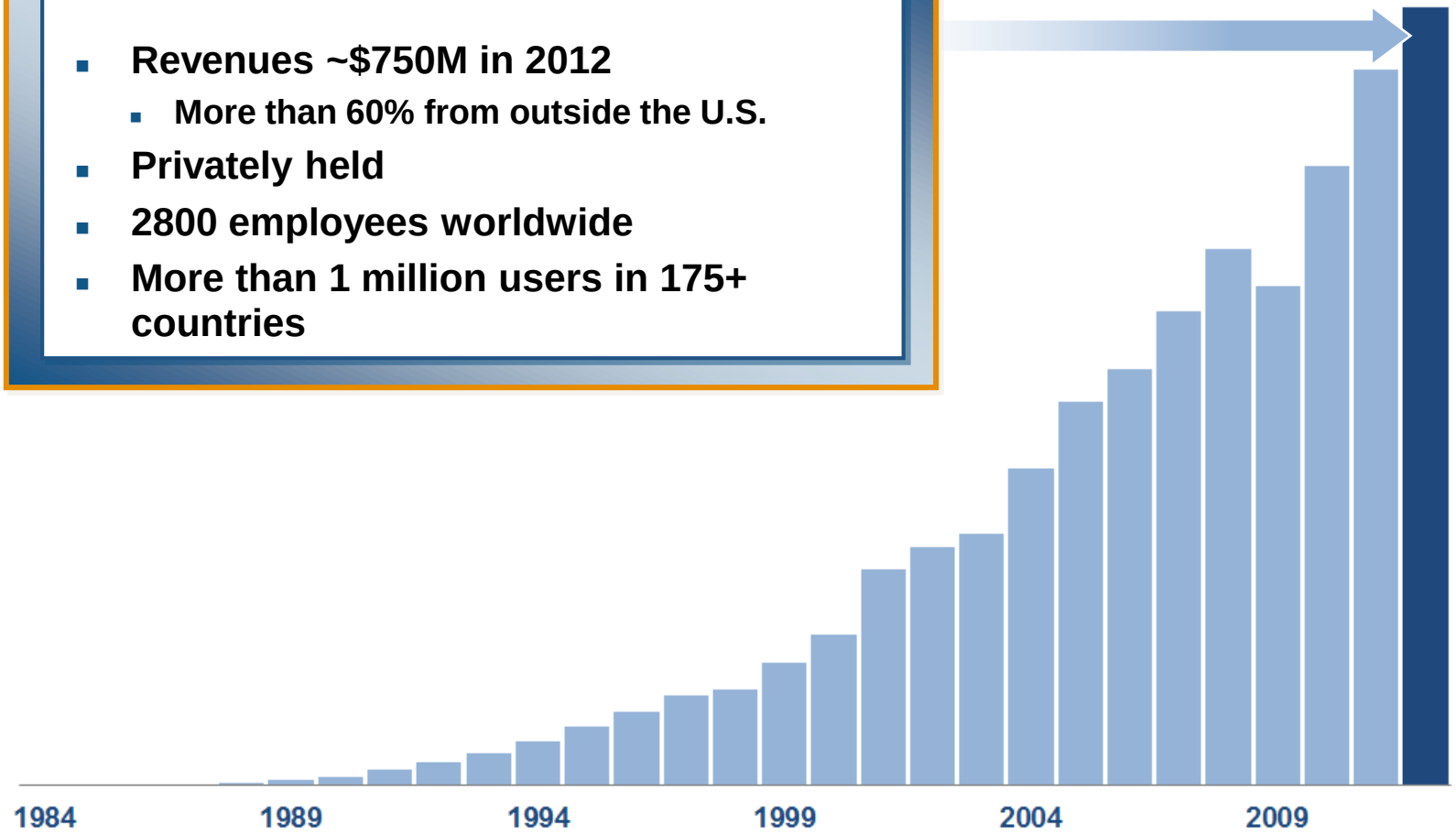
MathWorks at a Glance

Earth's topography on a Miller cylindrical projection, created with MATLAB and Mapping Toolbox.

- **Headquarters:** Natick, Massachusetts U.S.
- **Other U.S. Locations:** California; Michigan; Texas; Washington, D.C.
- **Europe:** France, Germany, Italy, Netherlands, Spain, Sweden, Switzerland, United Kingdom
- **Asia-Pacific:** Australia, China, India, Japan, Korea
- **Worldwide training and consulting**
- **Distributors serving more than 20 countries**

MathWorks Today

- Revenues ~$750M in 2012
  - More than 60% from outside the U.S.
- Privately held
- 2800 employees worldwide
- More than 1 million users in 175+ countries



Key Industries

- Aerospace and defense
- Automotive
- Biotech and pharmaceutical
- Communications
- Education
- Electronics and semiconductors
- Energy production
- Financial services
- Industrial automation and machinery
- Medical devices

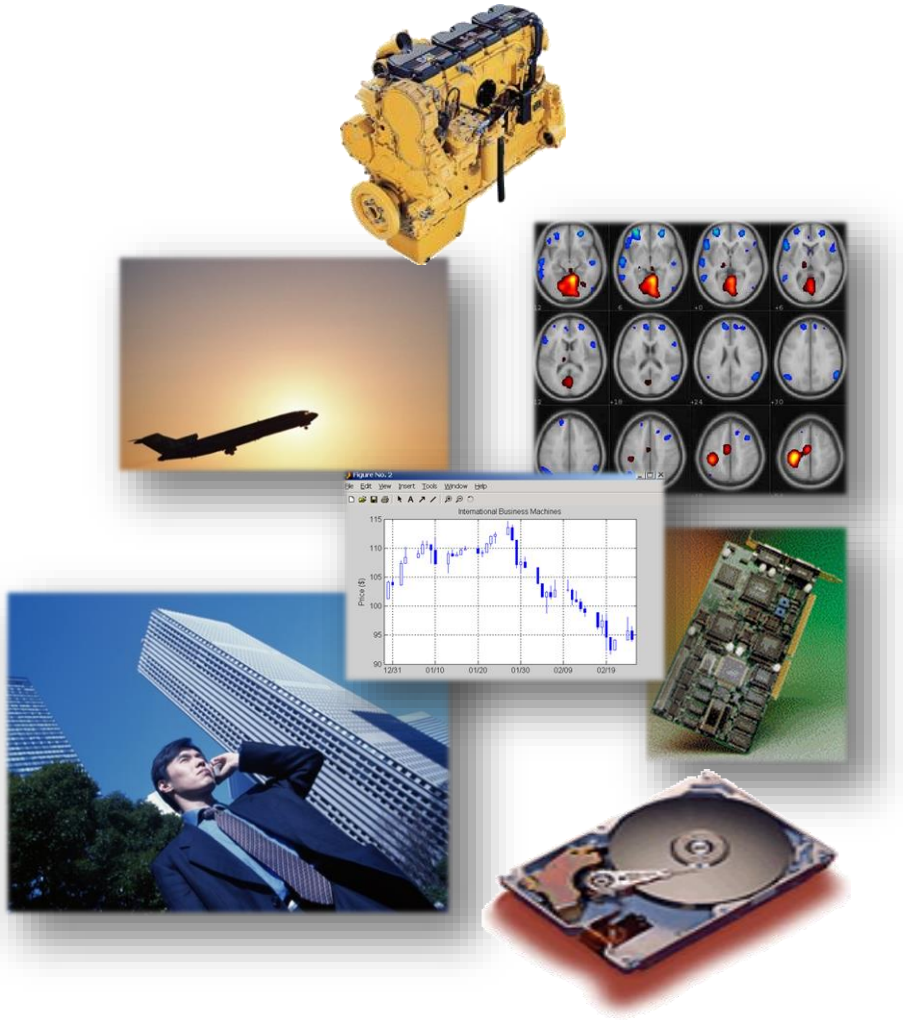
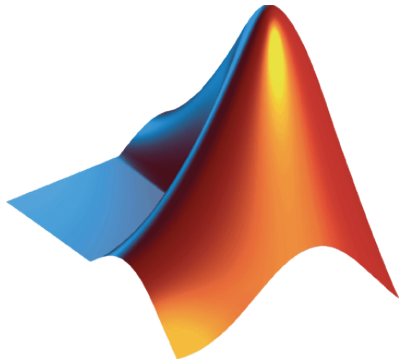


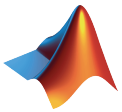
Image and Video Processing with MATLAB

Key Take Aways

- High-level language
- Development environment
- Technical computing platform



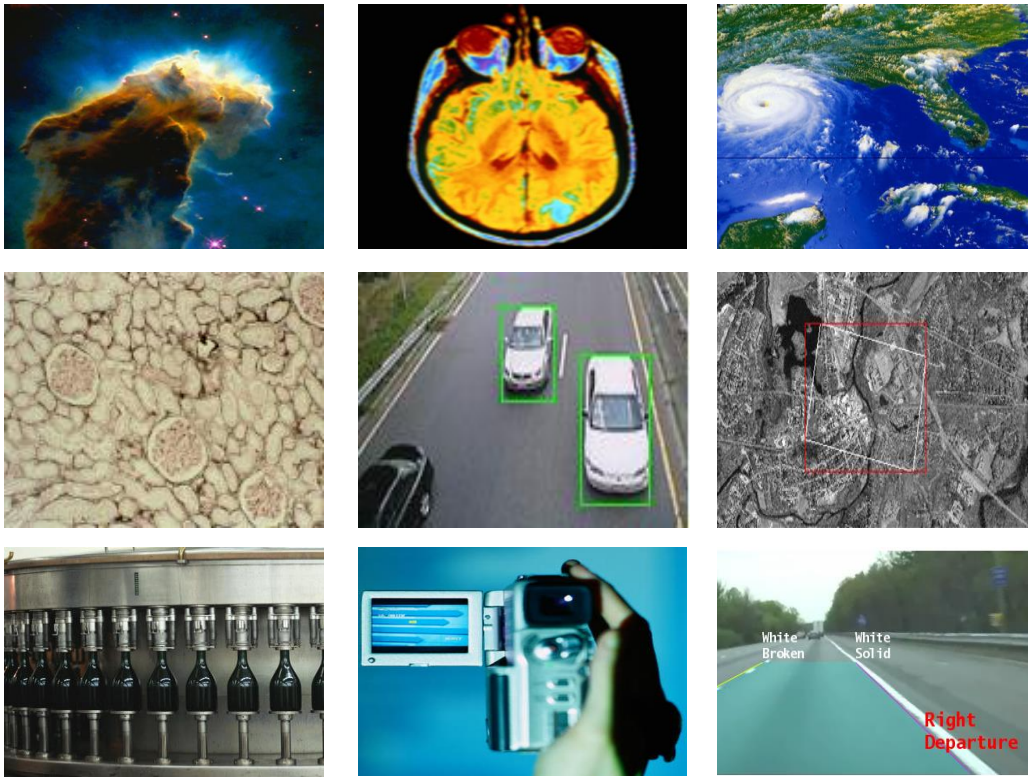
Agenda



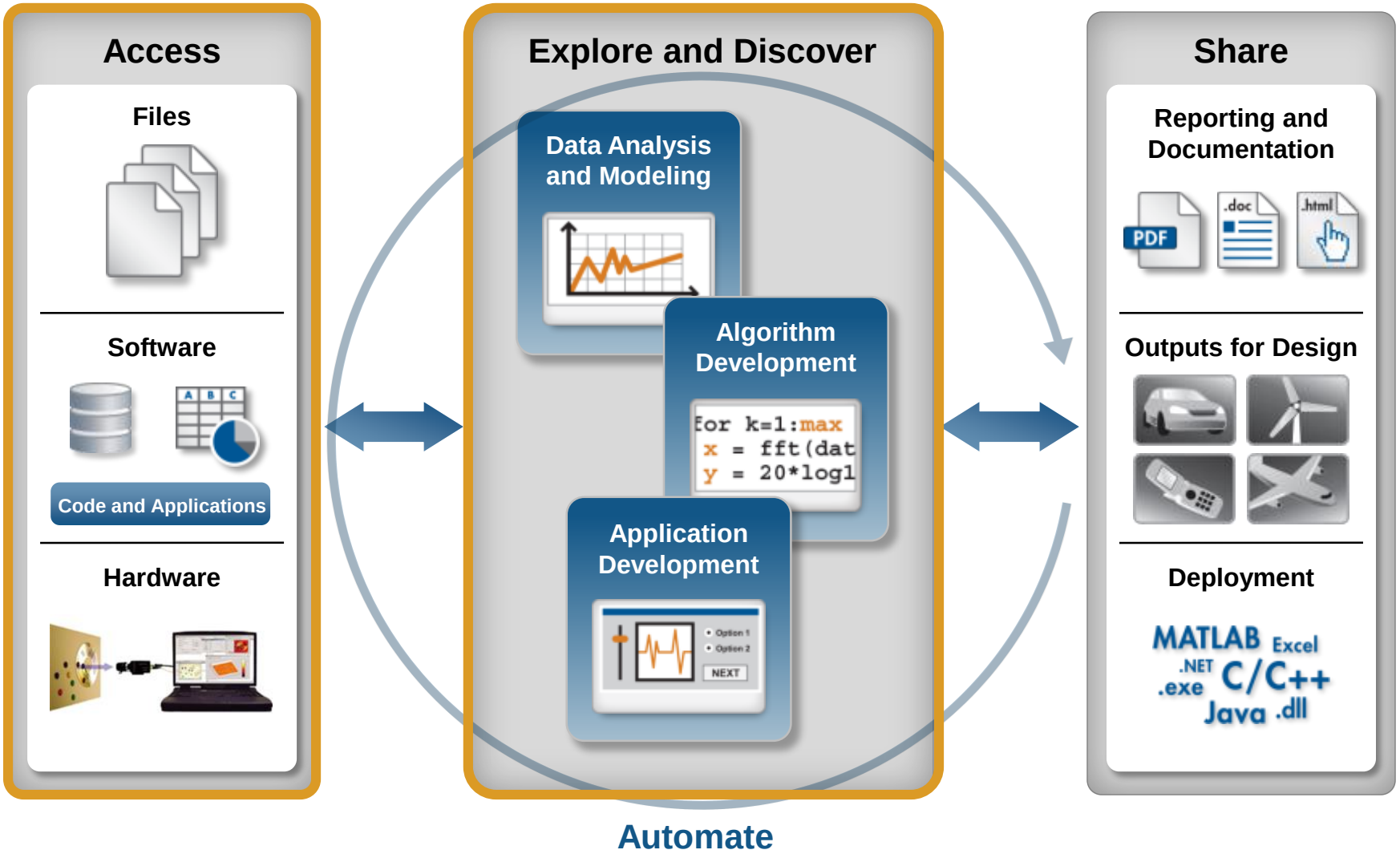
Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary

Applications: Image and Video Processing

- Medical imaging
- Surveillance
- Robotics
- Automotive safety
- Consumer electronics
- Geospatial computing
- Machine vision
- and more…

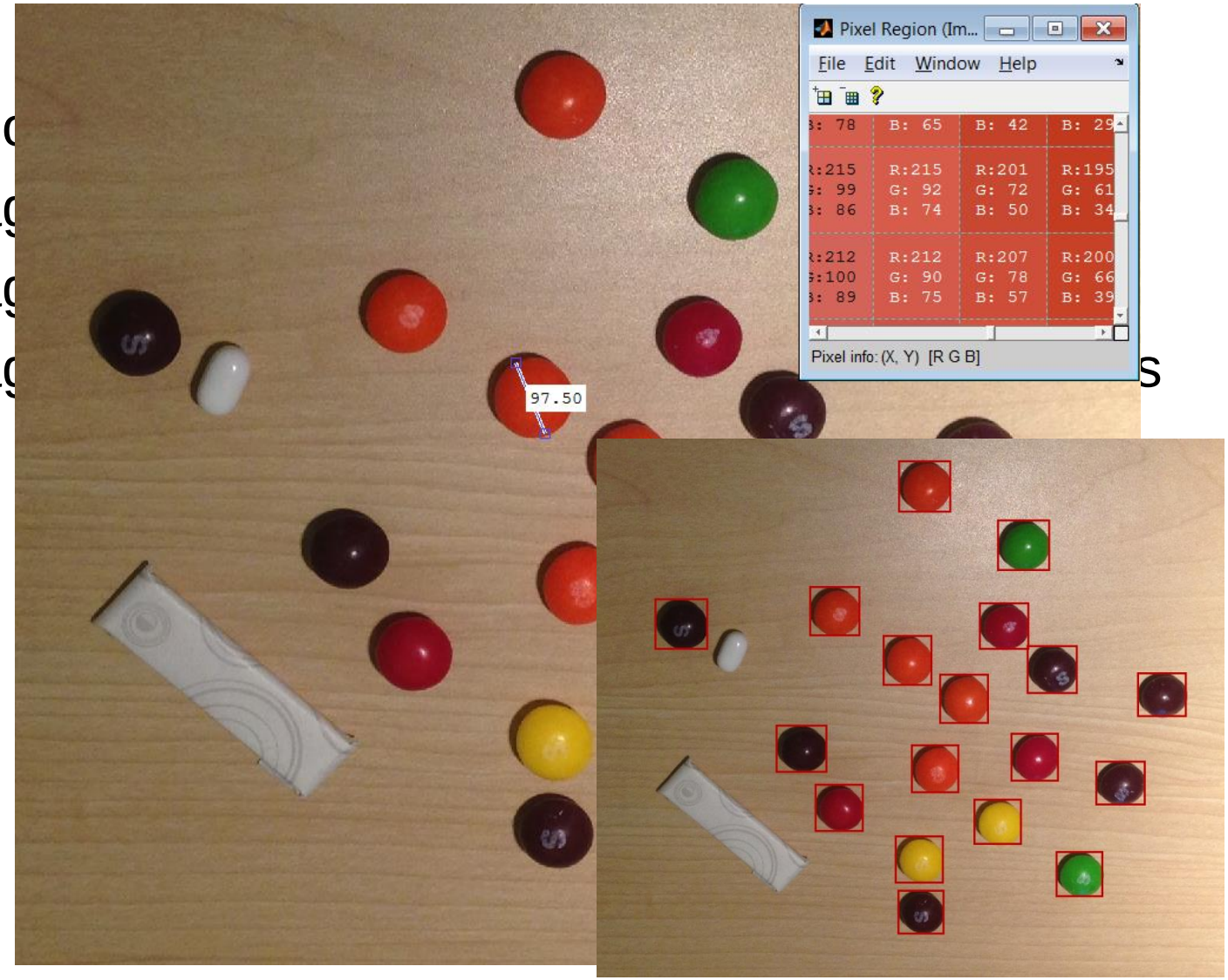


Technical Computing Tasks

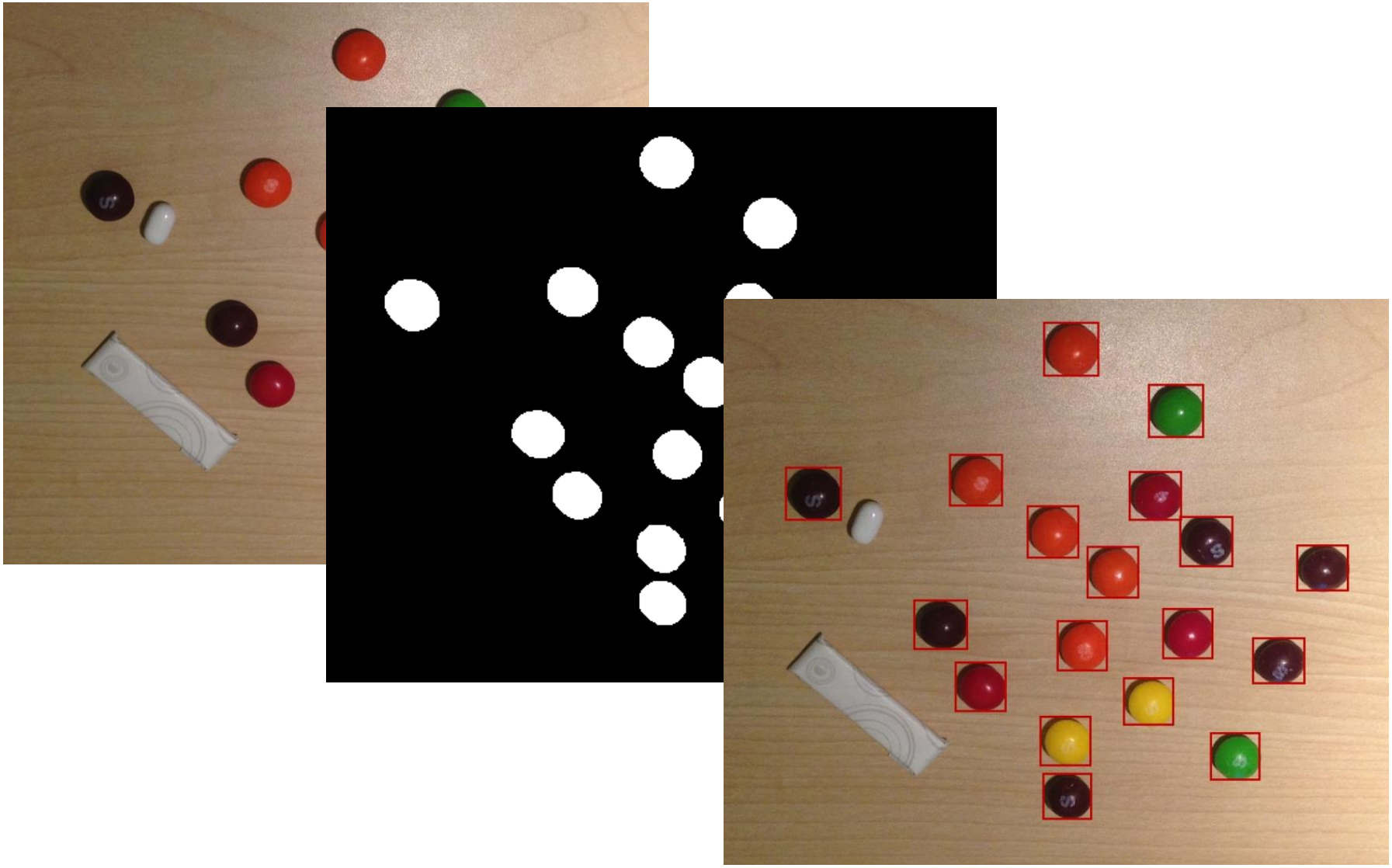


Let's perform some image processing?

- Impo
- Imag
- Imag
- Imag



Demo: Candy Counter



Algorithm Summary

- Import and Explore Images
- Image Enhancement
  - Morphological Operators
- Image Segmentation
  - Thresholding
  - Color Segmentation
- Image Analysis
  - Object properties and statistics

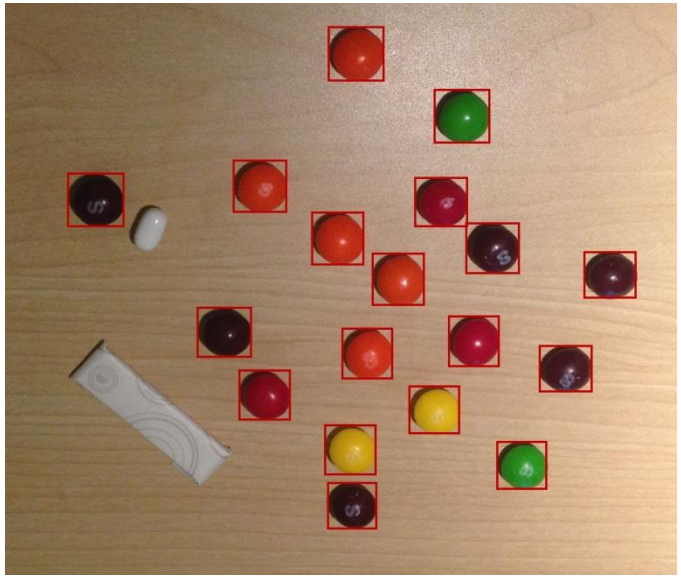
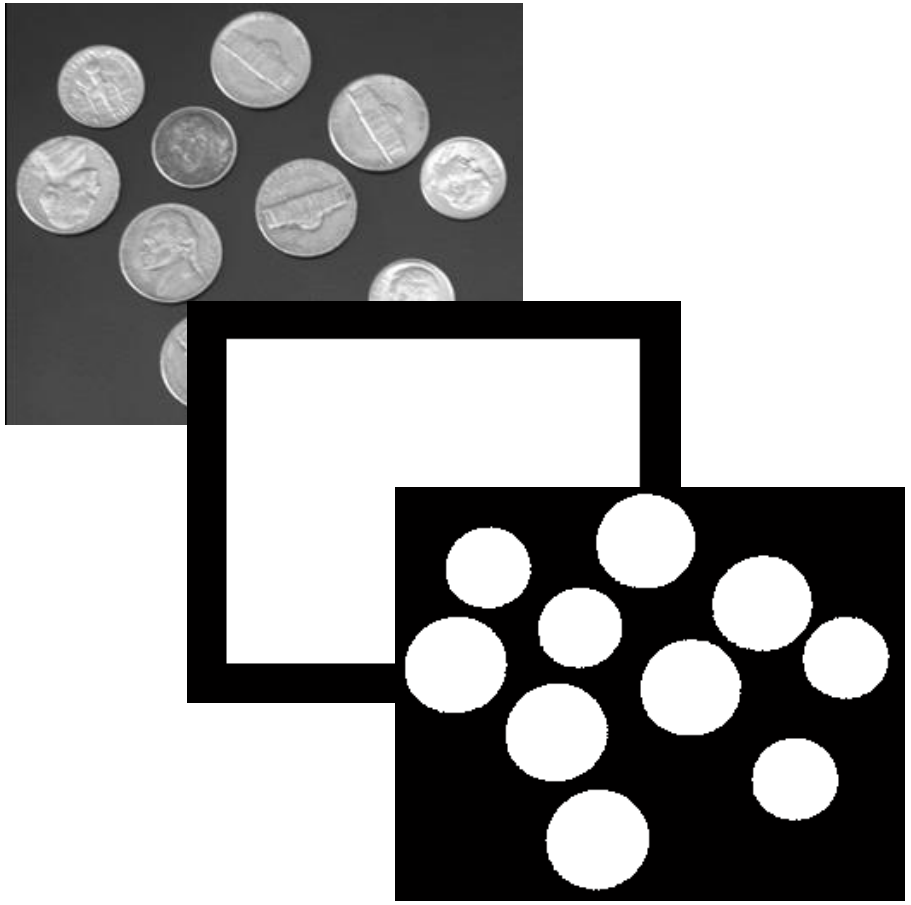


Image Segmentation – Other Methods
Active Contours and Watershed



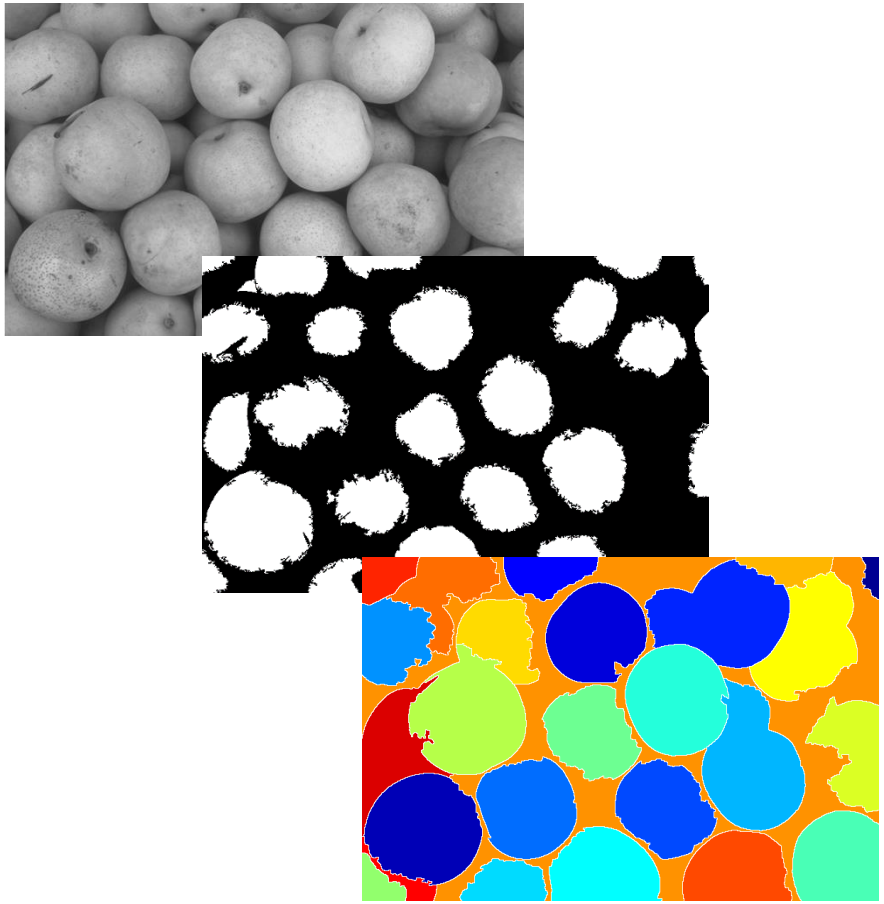


Image Enhancement
Deblurring and Noise Removal

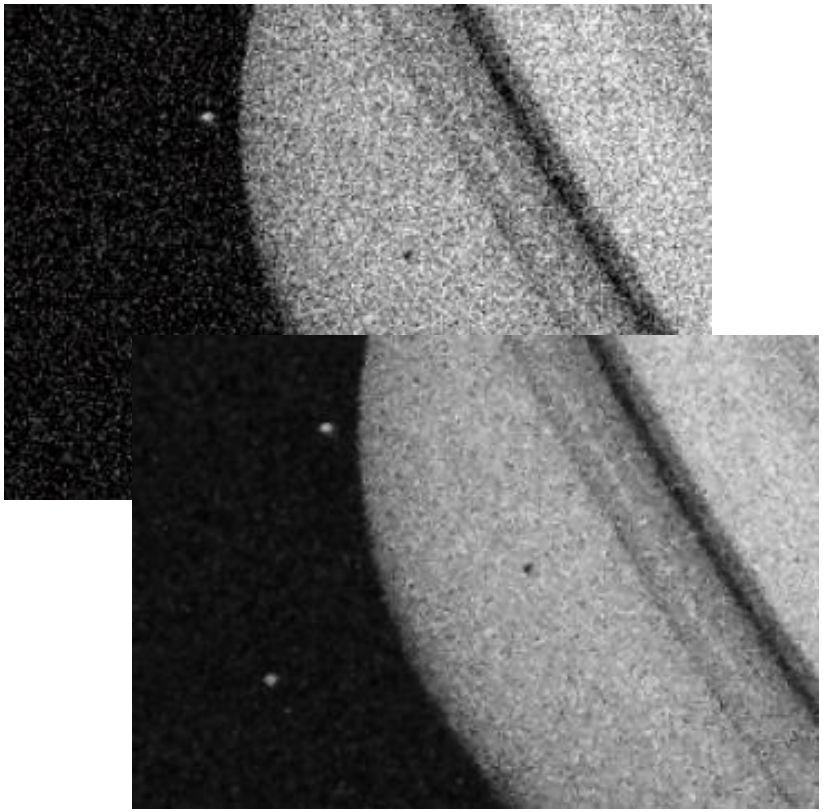


Image Registration
2-D & 3-D Image Alignment

- Align multimodal images

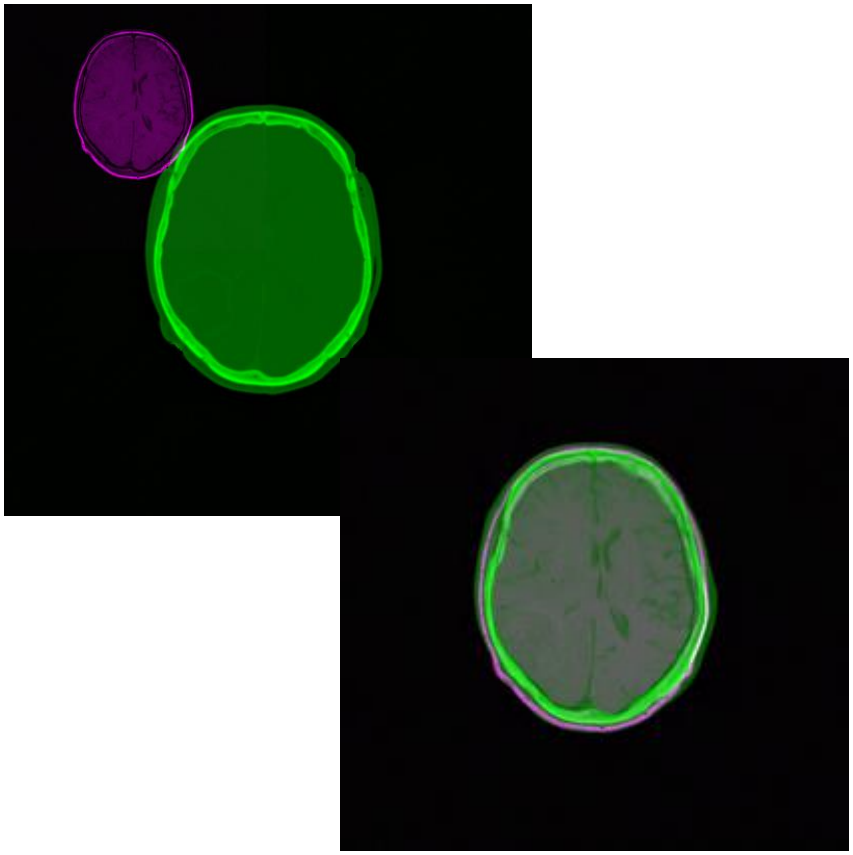
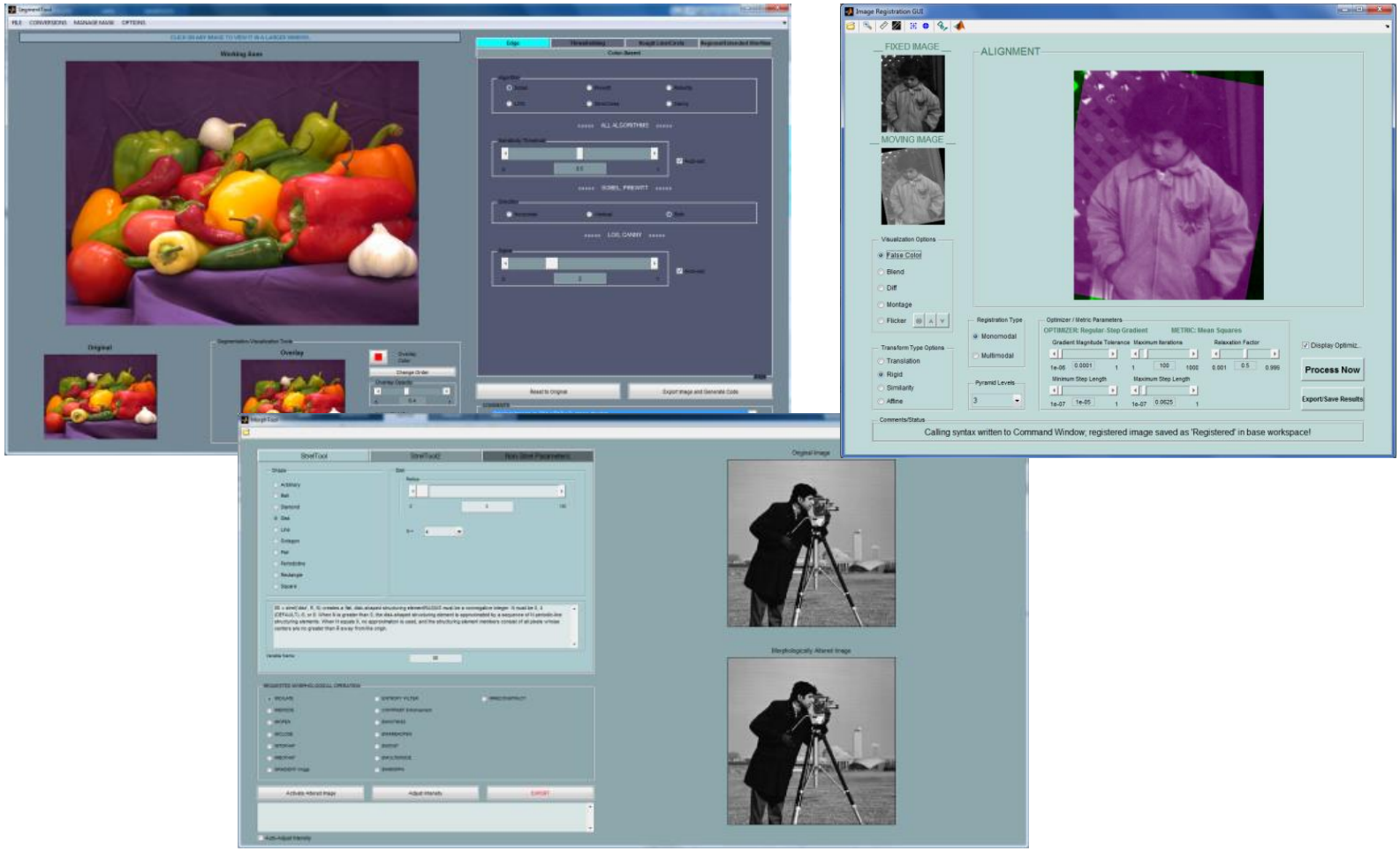


Image Processing with Live Video

- Validate algorithms with live data
- Integrate with low cost webcams
- Webcam support in base MATLAB

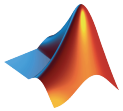
Image Processing Apps:

Segment Tool, MorphTool, Image Registration App



File Exchange: http://www.mathworks.com/matlabcentral/fileexchange/

Agenda



Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary

Features are Critical to Computer Vision

Typical Parts of a Computer Vision Algorithm

1. Image/video acquisition
2. Image/video pre-processing
3. Feature detection
4. Feature extraction
5. Feature matching
6. Using features
   - Stabilization, mosaicking
   - Stereo image rectification
7. Feature classification

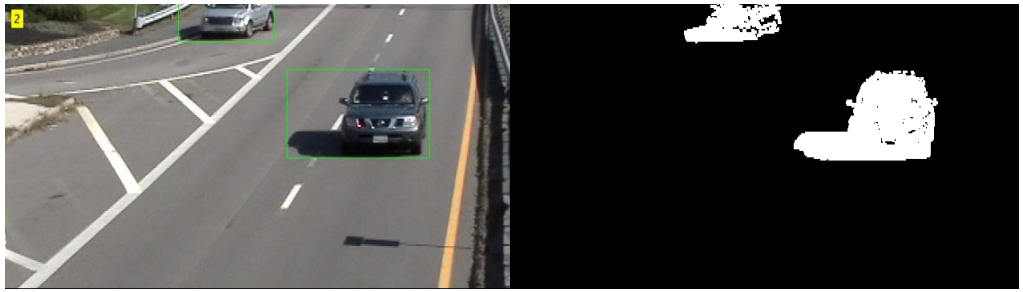
Image Acquisition Toolbox

Image Processing Toolbox

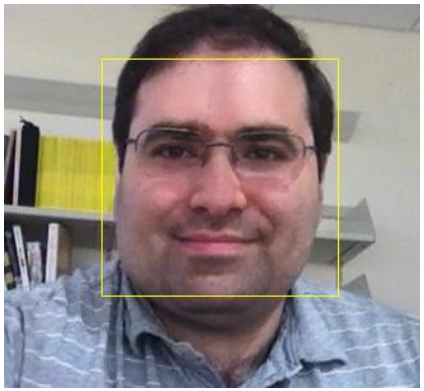
Computer Vision System Toolbox

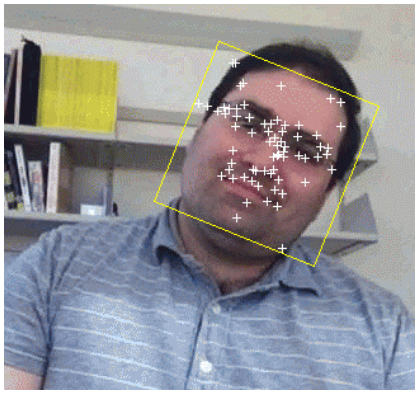
Statistics Toolbox

Face Detection, People Detection, and Background Subtraction



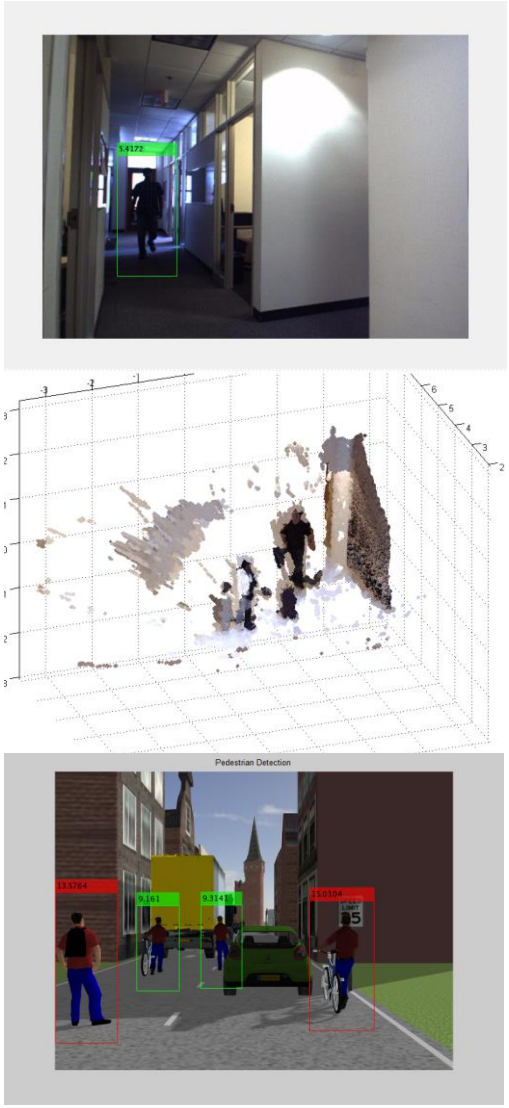
Demo: Face tracking with a webcam





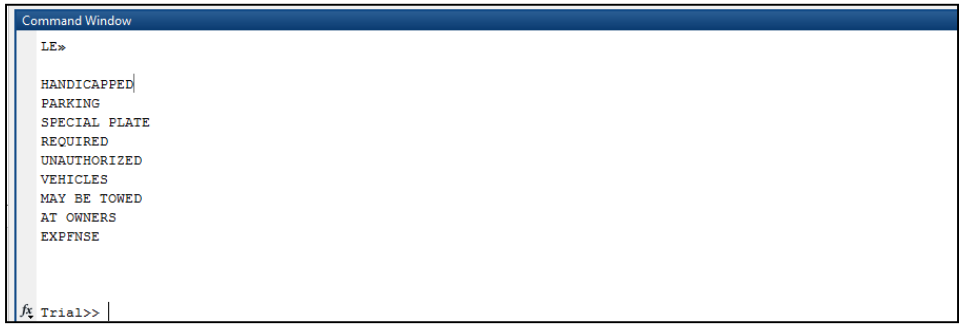
Stereo Vision (14a)

- Stereo calibration
- Semi-global disparity matching
  - Better results than block matching
- 3D scene reconstruction from disparity
- New demos available
  - Estimate real distance to objects in video
  - 3D scene reconstruction
  - Pedestrian detection (14b)



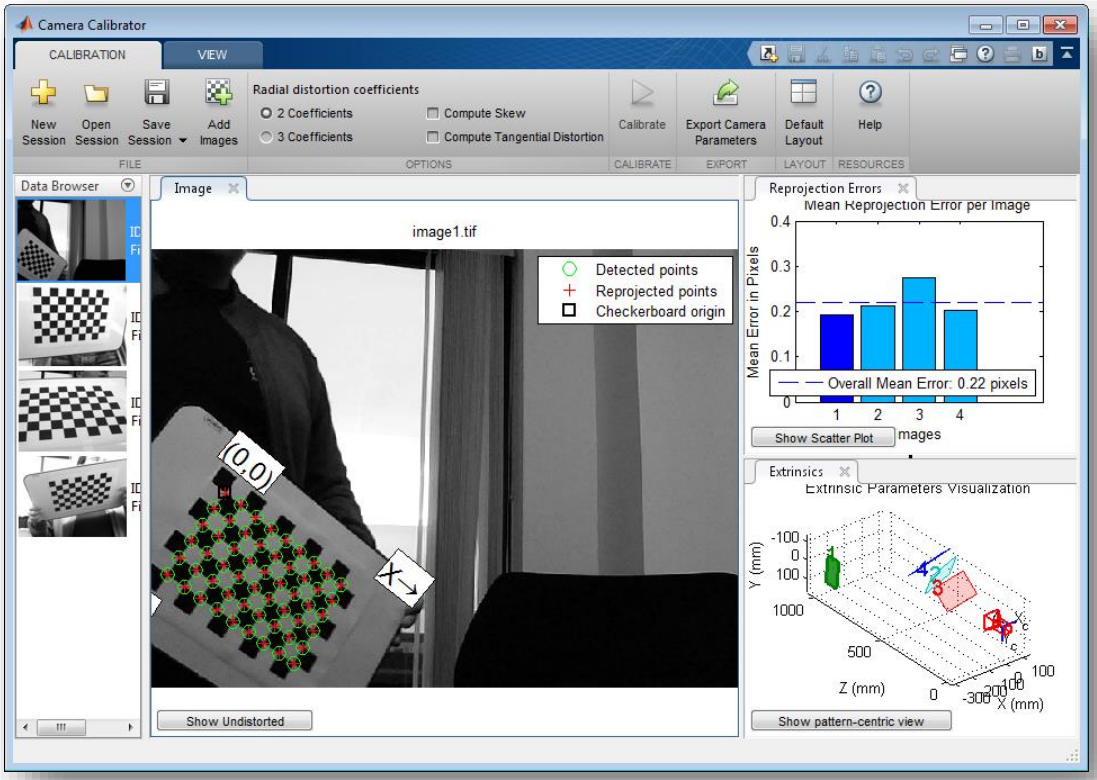
OCR – Optical Character Recognition (14a)

- Highly requested in Frontlines
- Support for English, Japanese
- Users can download additional language support
- Shipping demo with text detection and OCR workflow



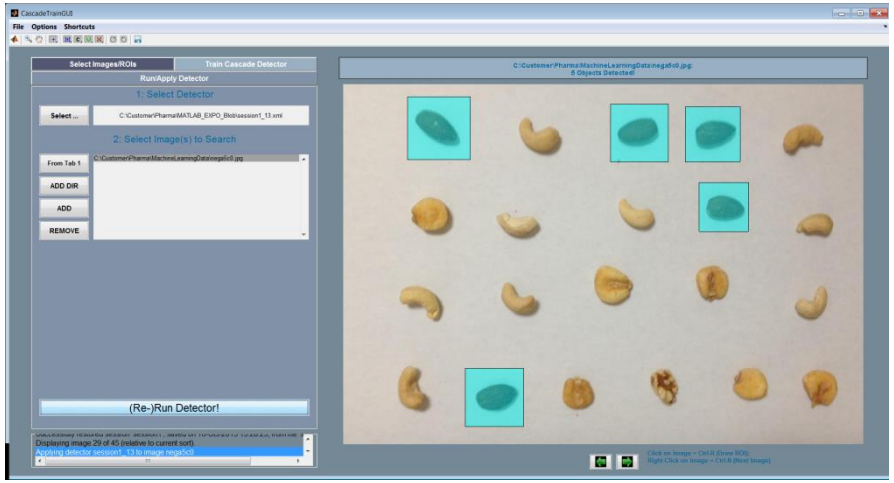
Camera Calibration App: Demo

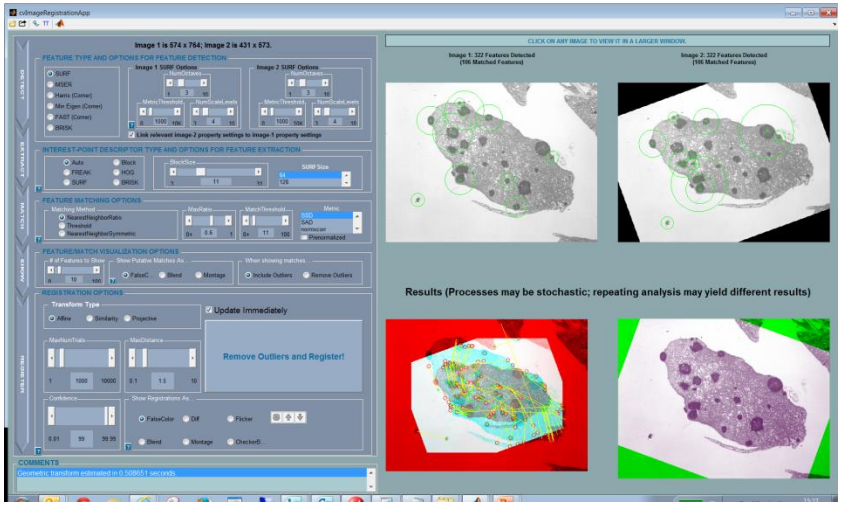
- Simplified workflow for estimating camera intrinsic and extrinsic parameters
- Removes the effects of lens distortion from an image
- Automatically detects checkerboard patterns



Computer Vision Apps on File Exchange

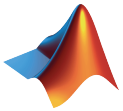
CascadeTrainGUI App, CV ImageRegistration App





File Exchange: http://www.mathworks.com/matlabcentral/fileexchange/

Agenda



Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary

Agenda

Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary

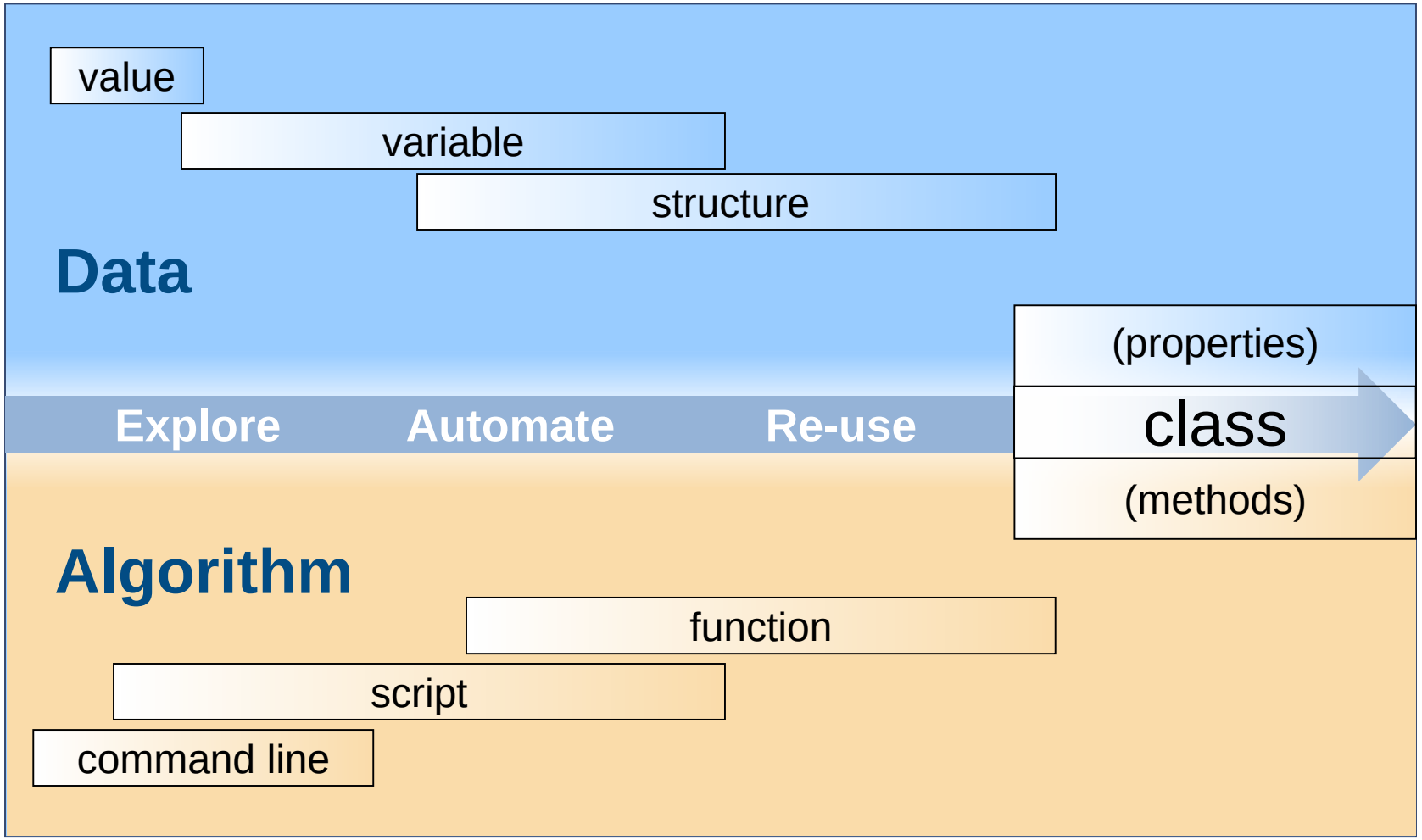
Common Image Processing Challenges

- Reading and writing to various file formats
- Gathering and exploring images to gain insight
- Create, test and debug algorithms

- Refine algorithms, make them robust and re-usable
- Processing large images with limited memory
- Executing algorithms faster

- Sharing results and generating reports
- Creating applications
- Deploy to other environment

Range of Programming Techniques



How are System objects different from MATLAB functions?

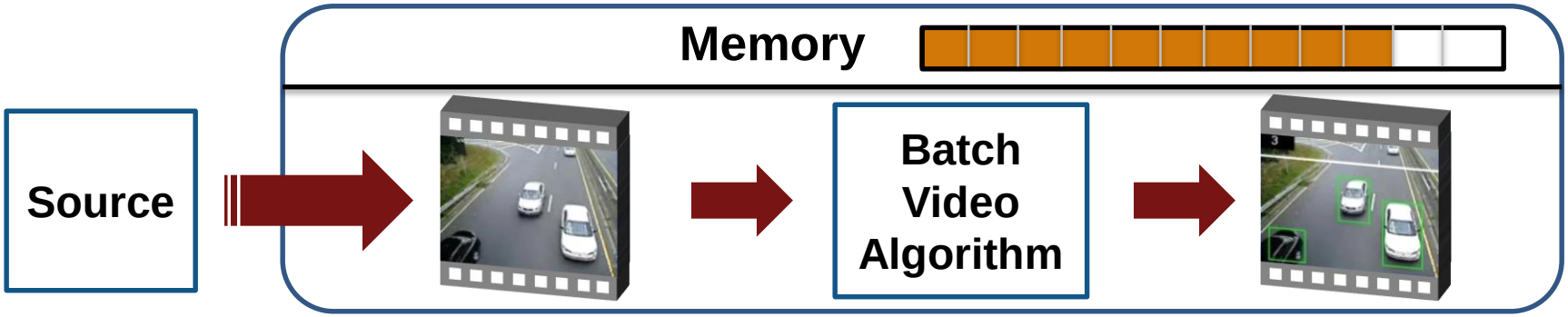
System Objects provide the following unique functionality:

1. Implicit state management, indexing and buffering
2. Support for fixed-point arithmetic+
3. Support for C code generation*#
4. Available in MATLAB and Simulink

+ with Fixed-Point Toolbox or Simulink Fixed Point
*** with MATLAB Coder or Simulink Coder**
 # no code generation with Phased Array System Toolbox as of R2012a

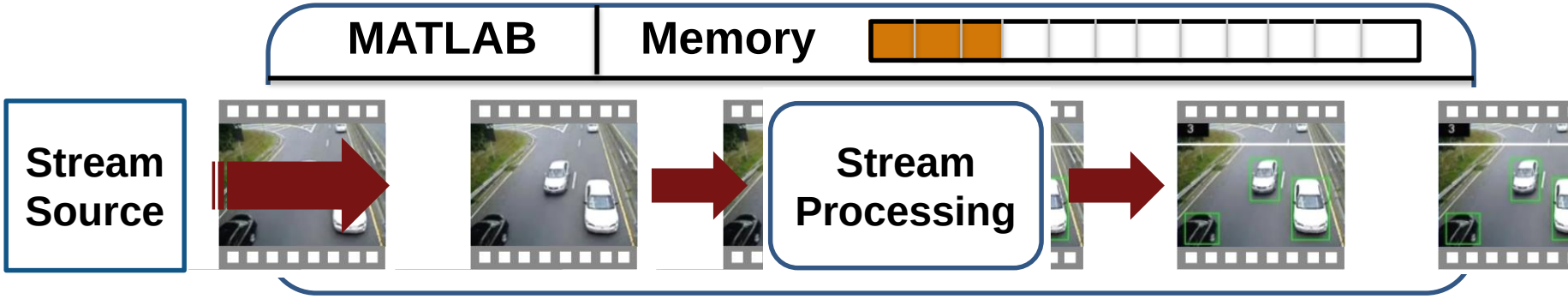
Video Processing in MATLAB… Before

Load the entire video file and process it all at once



… And with System Objects

Load a frame and process it before moving on to the next frame



Video Processing in MATLAB… Before

Video processing is difficult in MATLAB…

```
myVid = mmreader('myvideofile.avi');
numFrames = myVid.NumberOfFrames;
numIter = 10;
opticalFlowIn = zeros([size(currentFrame) 5]);
opticalFlowOutput = zeros([size(currentFrame) numFrames]);

i = 1;
while i <= numFrames
    opticalFlowIn(:,:,2:end) = opticalFlowIn(:,:,1:end-1);
    opticalFlowIn(:,:,1) = read(myVid,i);

    flow = opticalFlow(opticalFlowIn(:,:,1),opticalFlowIn(:,:,5),…
                    'horn-schunck',numIter,'magitude-squared');

    opticalFlowOutput(:,:,i) = flow;

    i = i+1;
end

implay(opticalFlowOutput,30)
```

Explicit state management

Explicit indexing

Needs a buffer

Video Processing in MATLAB… After

Video processing is difficult in MATLAB…

… but system objects make it easier.

```
reader = video.MultimediaFileReader
reader.Filename = 'myvideofile.avi';
viewer = video.DeployableVideoPlayer

optical = video.OpticalFlow
optical.Method = 'horn-schunck';
optical.OutputValue = 'Magitude-squared';
optical.ReferenceFrameDelay = 3;
optical.MaximumIterationCount = 10;
```

Object initialization

```
while ~isDone(reader)
   currentFrame = step(reader);
   OF = step(optical, currentFrame);
   step(viewer, OF);
end
```

Code in **the loop more simple**

Implicit state management, indexing and buffering

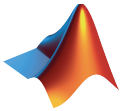
Video player works in the loop

Summary System Objects

- New stream processing capabilities in MATLAB
- Common user interface across all algorithms
- Bridge MATLAB and Simulink
- Code generation in MATLAB & Simulink

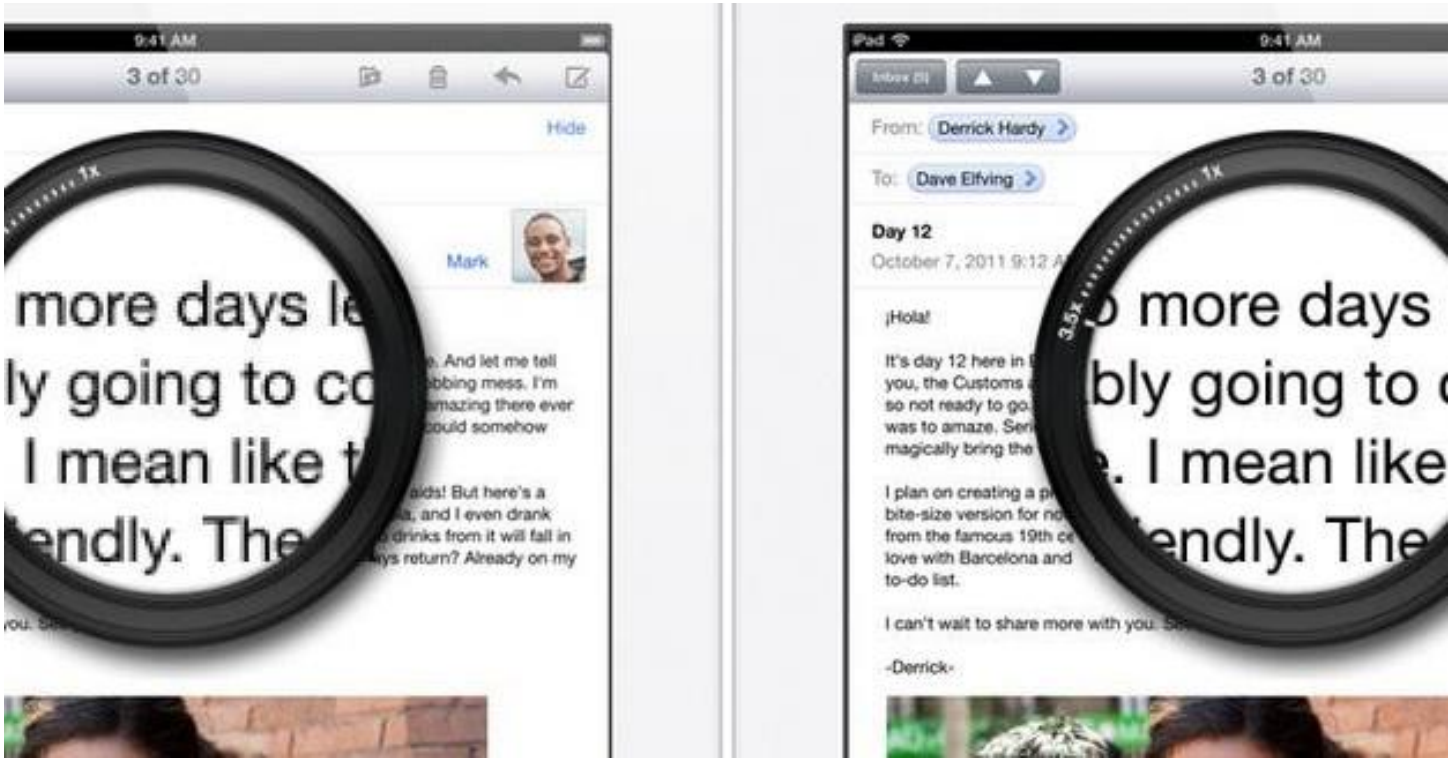
Agenda

Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary



Images & Processing Demands on the Rise

- Moving towards 4k / 8k UHD capture and display.
- Efficiency and performance are becoming crucial.



Solving Larger Problems

- Two primary types of larger problems:
  - Problems that **take too long** for one computer to solve
  - Problems with **data that is too big** for one computer

- Problems can be addressed by leveraging additional hardware
  - Multi-core or multi-processor computer
  - Graphics Processing Unit (GPU)
  - Clusters and clouds

Using More Hardware

- Built-in multithreading
  - Automatically enabled in MATLAB since R2008a
  - Multiple threads in a single MATLAB computation engine
    www.mathworks.com/discovery/multicore-matlab.html

- Parallel computing using explicit techniques
  - Perform MATLAB computations on GPUs
  - Multiple computation engines controlled by a single session
  - High-level constructs to let you parallelize MATLAB applications

GPU vs. CPU Comparison

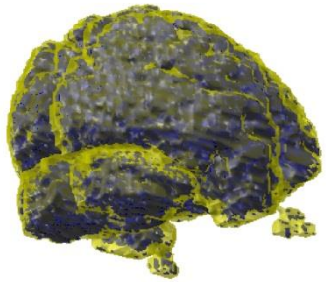
Features	CPU	GPU
# of Threads	10s – 100s	100s – 1000s
Thread Scheduler	Host Operating System (i.e. Windows / Linux)	Dedicated Low Latency Scheduler
Memory	System Memory / Cached (Off Chip)	High Speed / Low Latency (On Chip)
Upgradability	Hard – New motherboard and processor	Easy – New PCIe card
Large Data Capability	Medium	High
Scalability	Clusters and Clouds	Clusters and Clouds

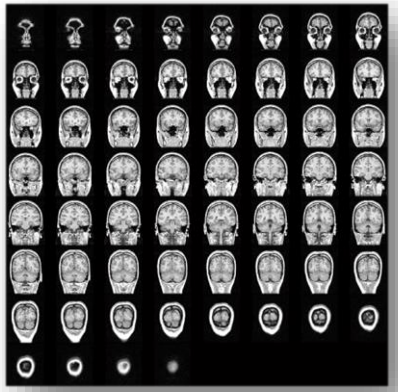
vs.

GPU Acceleration for 43 functions

Use NVIDIA GPUs to accelerate 43 of the most popular Image Processing Toolbox functions

- Accelerate complex algorithms with large data
- Integrate CUDA code for prototyping and testing

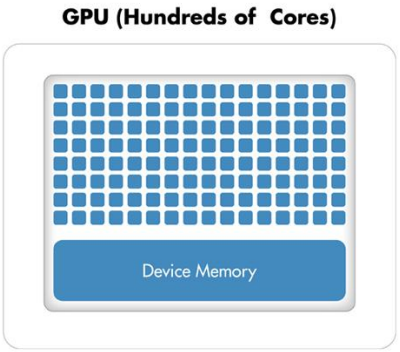




bwmorph	imhist
bwlookup	imnoise
corr2	imopen
edge	imresize
histeq	imrotate
imadjust	imshow
imbothat	imtophat
imclose	imwarp
imdilate	mean2
imerode	medfilt2
imfilter	padarray
imgradient	rgb2gray

Criteria for Good Problems to Run on a GPU

- **Massively parallel:**
  - Calculations can be broken into hundreds or thousands of independent units of work
  - Problem size takes advantage of many GPU cores



- **Computationally intensive:**
  - Computation time significantly exceeds CPU/GPU data transfer time

- **Algorithm consists of supported functions:**
  - Growing list of Toolboxes with built-in support
    - www.mathworks.com/products/parallel-computing/builtin-parallel-support.html
  - Subset of core MATLAB for `gpuArray`, `arrayfun`, `bsxfun`
    - www.mathworks.com/help/distcomp/using-gpuarray.html#bsloua3-1
    - www.mathworks.com/help/distcomp/execute-matlab-code-elementwise-on-a-gpu.html#bsnx7h8-1

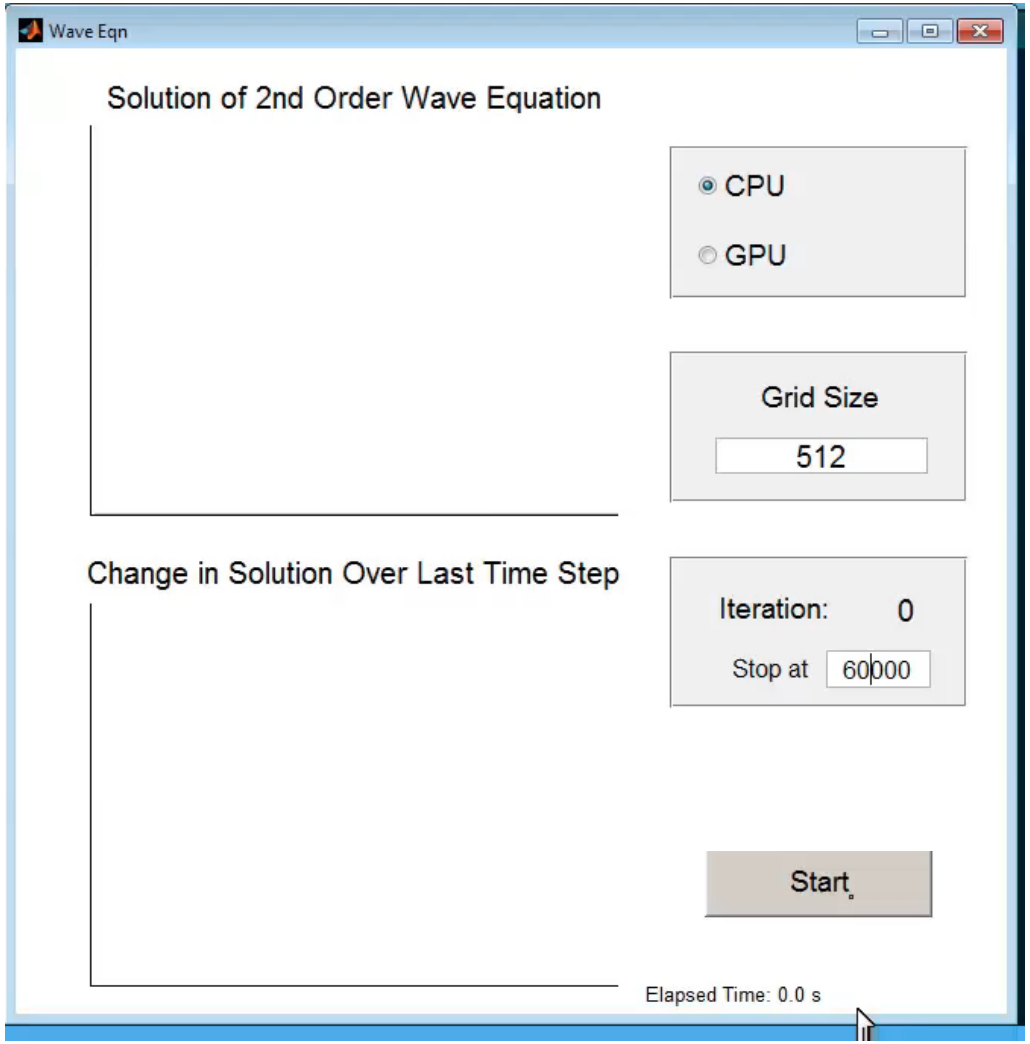
Example: Solving 2D Wave Equation

GPU Computing

- Solve 2nd order wave equation using spectral methods:

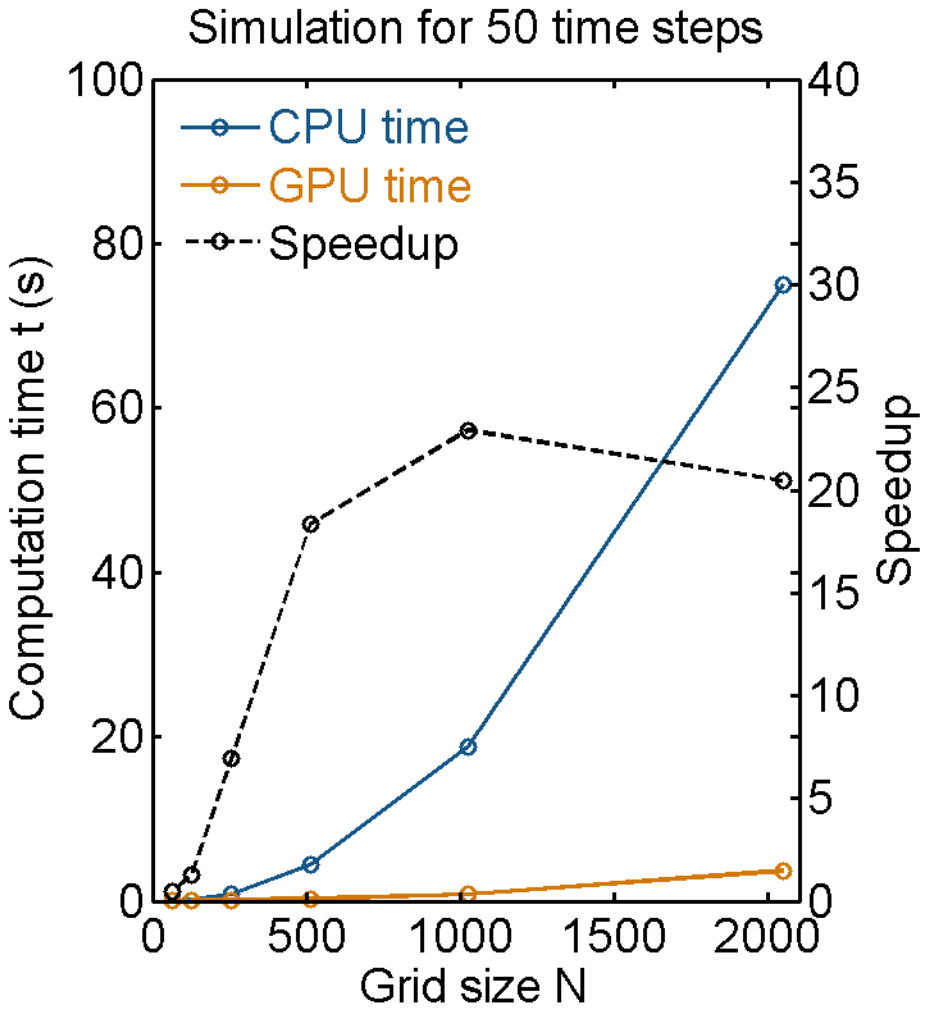
$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

- Run both on CPU and GPU
- Using `gpuArray` and overloaded functions



Benchmark: Solving 2D Wave Equation

GPU Computing

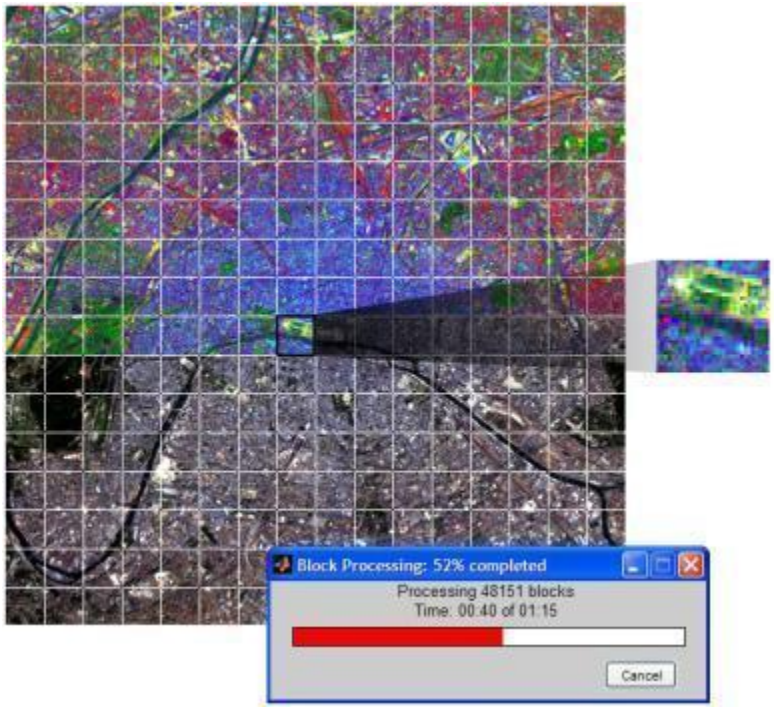


Grid Size	CPU (s)	GPU (s)	Speedup
64 x 64	0.05	0.11	0.4
128 x 128	0.14	0.11	1.3
256 x 256	0.83	0.12	7.2
512 x 512	4.40	0.24	18.
1024 x 1024	18.8	0.82	23.
2048 x 2048	75.0	3.67	20.

CPU: Intel Xeon W3550 (3.07GHz)
GPU: NVIDIA Tesla K20c

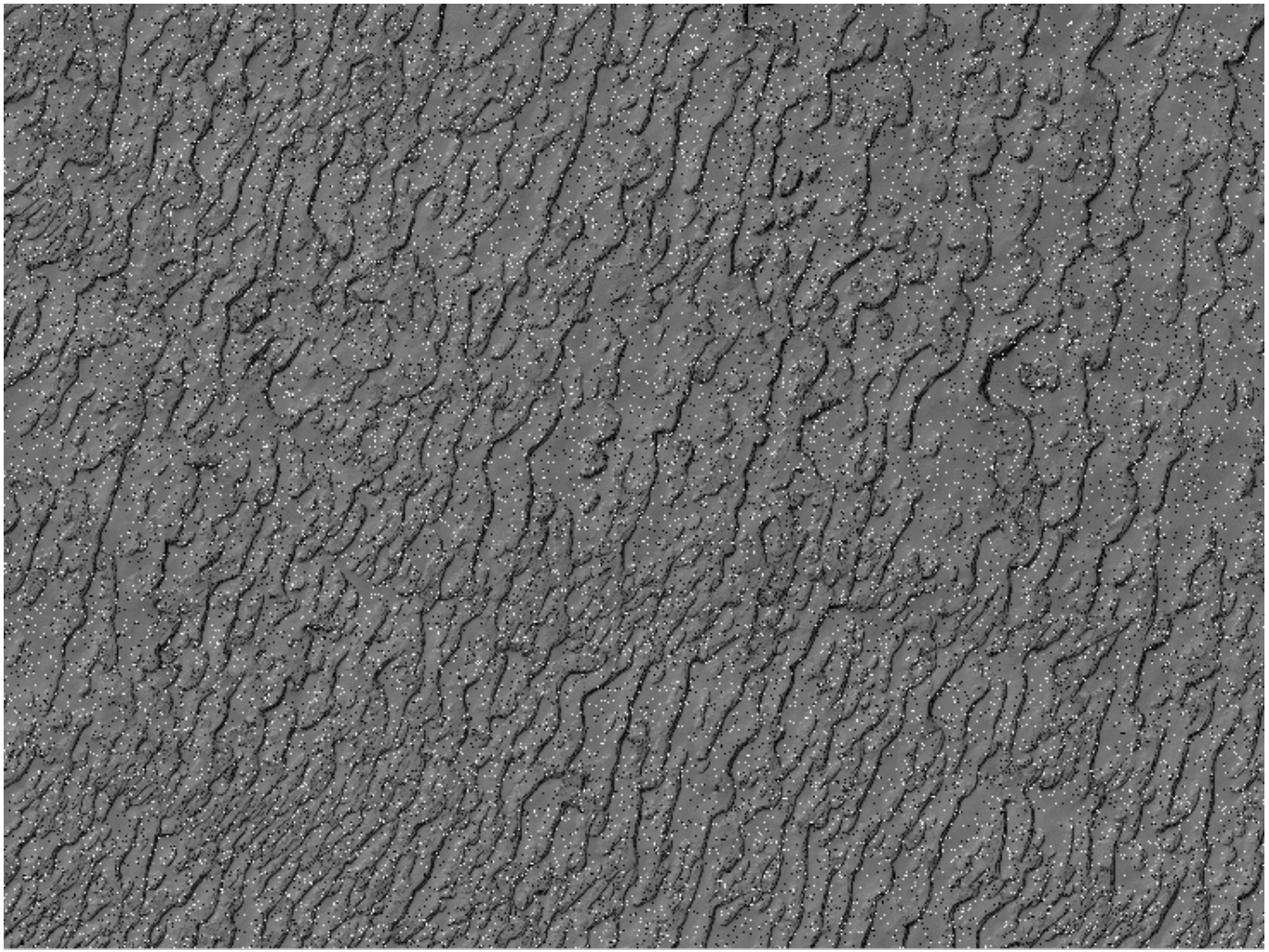
Working with Large Images

- Block processing with `blockproc`
  - Automatically divides an image into blocks for processing
  - Reduces memory usage
  - Processes arbitrarily large images
- Reduced resolution data set
  - `rsetwrite`
  - Avoids memory demands in visualizing large images



Demo: filtering a large Image

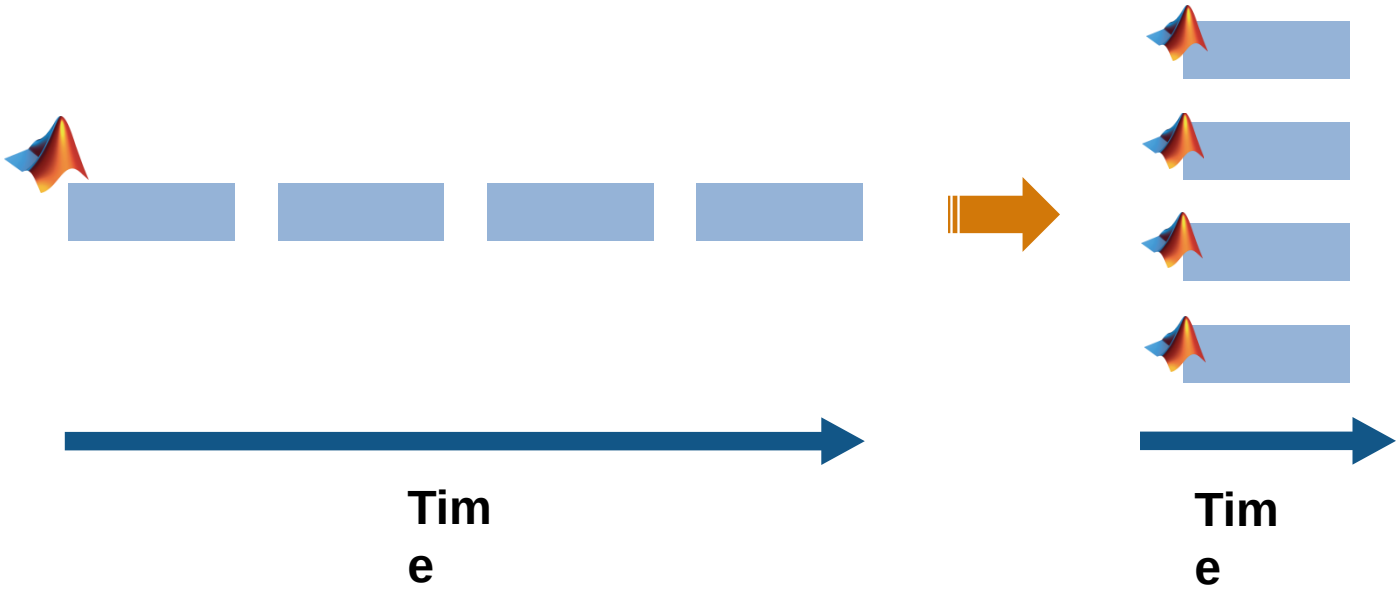
Using blockproc and Built-in Parallel Support



Independent Tasks or Iterations

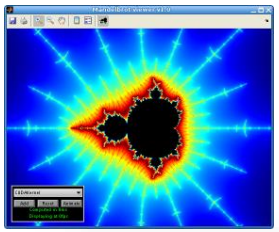
Parallel for-loops

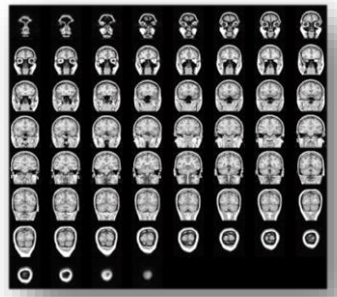
- Ideal problem for parallel computing
- No dependencies or communications between tasks
- Examples: parameter sweeps, Monte Carlo simulations

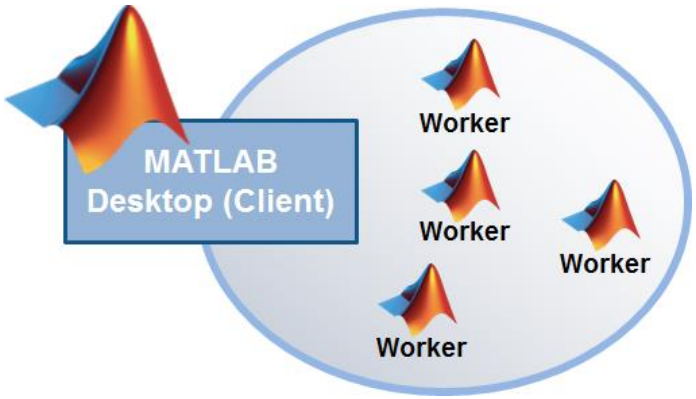


blogs.mathworks.com/loren/2009/10/02/using-parfor-loops-getting-up-and-running/

Demos: GPU Computing, Blockproc, Parallel



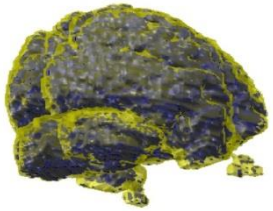


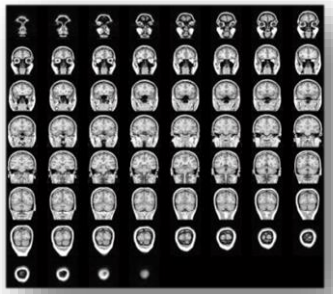


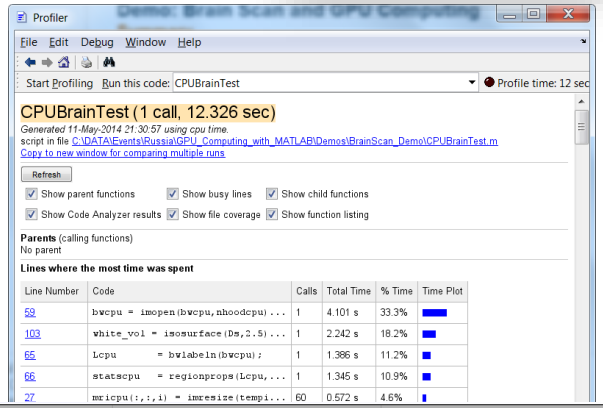
Demo: Brain Scan and GPU Computing

Summary

- Found bottleneck with **`profiler`**
- Run on gpu using **`gpuArray`** and **`gather`**
- Compared speed-up cpu vs gpu using **`timeit`** and **`gputimeit`**







	Results for data-type 'double' (In GFLOPS)			Results for data-type 'single' (In GFLOPS)		
	MTimes	Backslash	FFT	MTimes	Backslash	FFT
Quadro K6000	1489.50	453.38	141.32	3998.82	737.72	295.48
Tesla K20c	1005.00	490.83	110.40	2690.21	772.21	257.51
Tesla C2075	327.83	242.26	69.13	684.97	425.15	144.56
GeForce GTX TITAN	213.35	124.43	90.89	3840.88	735.68	328.85
GeForce GTX 680	139.20	97.53	58.82	1468.69	620.54	214.67
Quadro 2000	38.60	33.01	14.18	232.90	122.57	46.32
NVS 5400M	**24.46**	**18.14**	**9.13**	**143.15**	**87.13**	**42.28**
GeForce GT 640	18.13	14.08	8.51	185.60	95.49	33.62
Host PC	**33.57**	**23.96**	**2.85**	**72.69**	**50.32**	**4.98**
Quadro K600	13.24	10.69	6.17	135.40	0.01	26.57

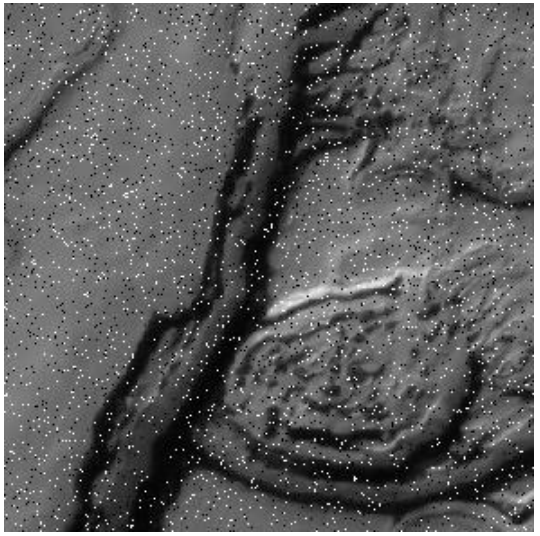
[gpuBenchmark]

Demo: Filtering a large Image

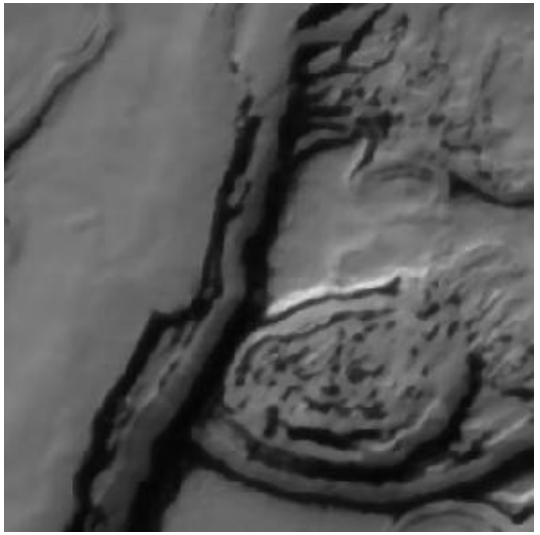
Summary

- Enabled built-in support for Parallel Computing Toolbox in Image Processing Toolbox
- Used pool of MATLAB workers
- Run median filtering in parallel using `blockproc`

Noisy Image



Filtered Image



http://hirise.lpl.arizona.edu/
From - NASA/JPL/University of Arizona

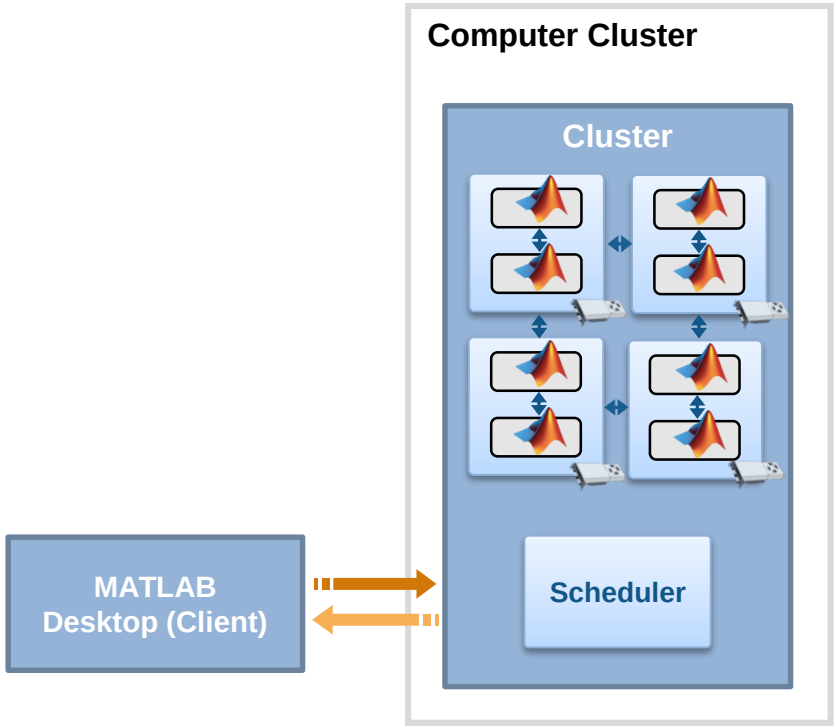
Demo: Candy Counter parallel for loop and batch

Summary

- Ran loops on a pool of MATLAB resources
- Converted `for`-loop into `parfor`-loop
- Run function on worker using `batch`

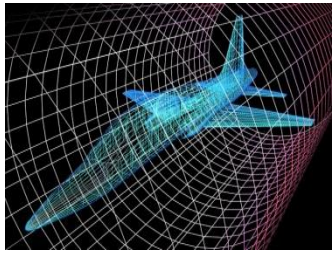
Take Advantage of Cluster Hardware

- Offload computation:
  - Free up desktop
  - Access better computers
- Scale speed-up:
  - Use more cores
  - Go from hours to minutes
- Scale memory:
  - Utilize distributed arrays
  - Solve larger problems without re-coding algorithms



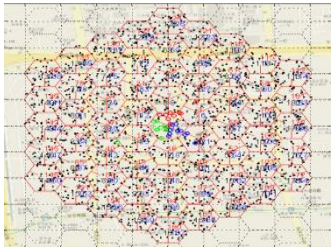
Optimizing JIT Steel Manufacturing Schedule
Cut simulation time from 1 hour to 5 minutes

Heart Transplant Studies
3-4 weeks reduced to 5 days



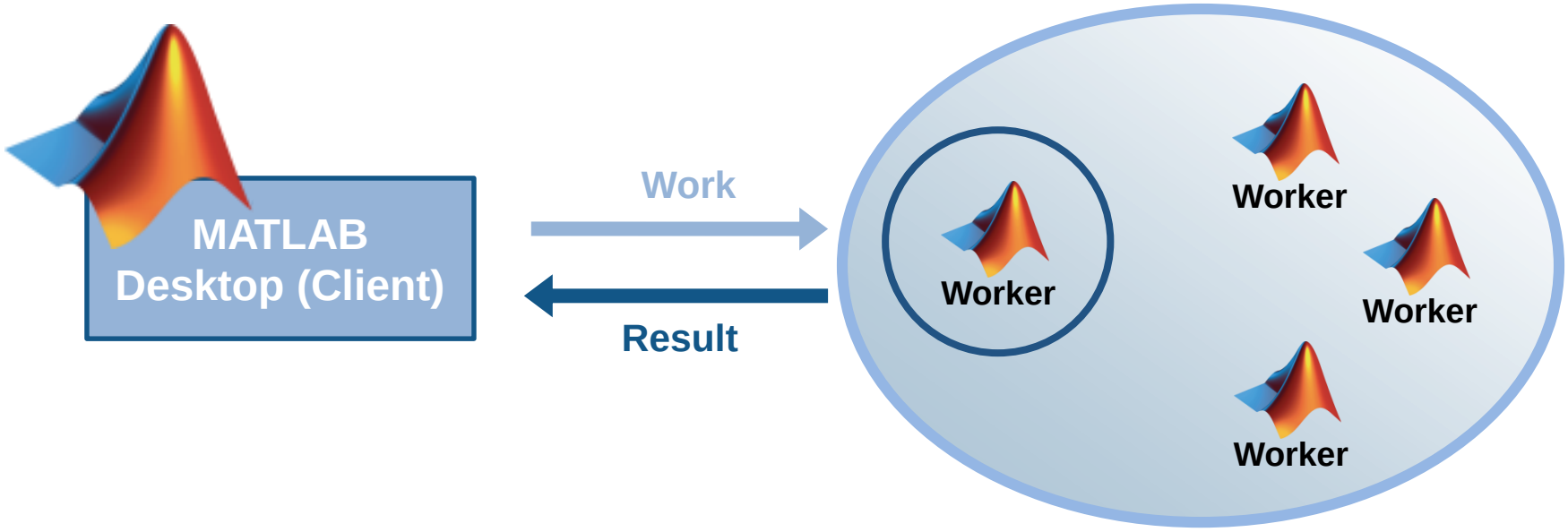
Flight Test Data Analysis
16x Faster

Mobile Communications Technology
Simulation time reduced from weeks to hours, 5x more scenarios



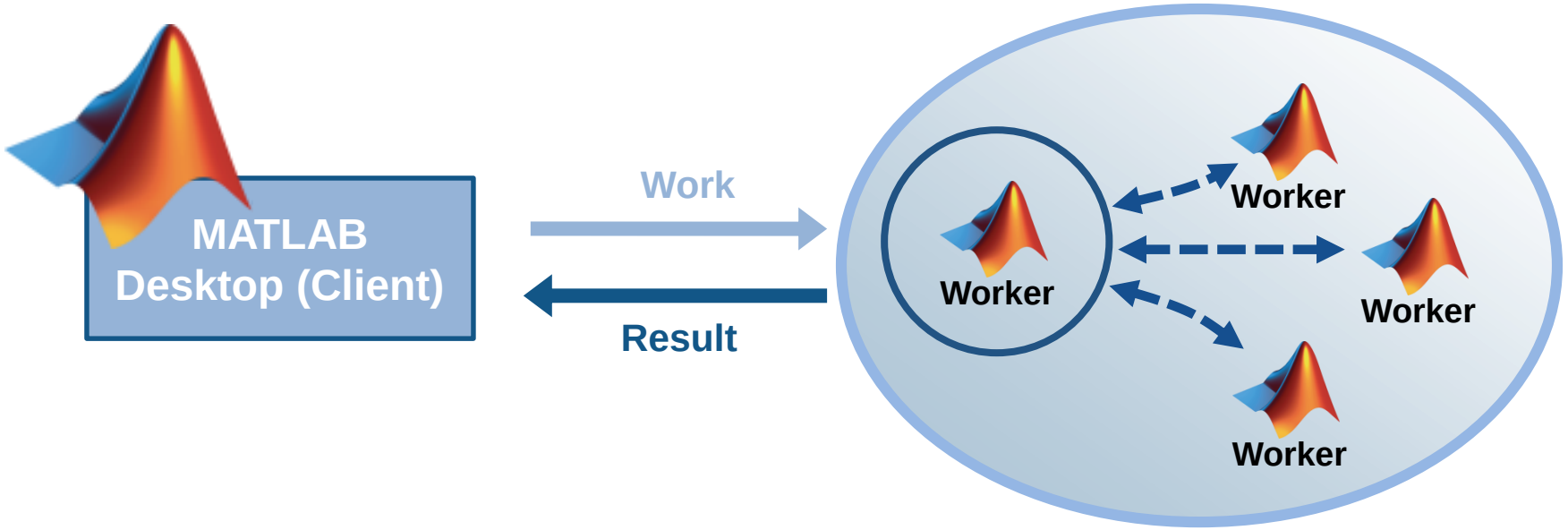
Hedge Fund Portfolio Management
Simulation time reduced from 6 hours to 1.2 hours

Offload Computations with `batch`



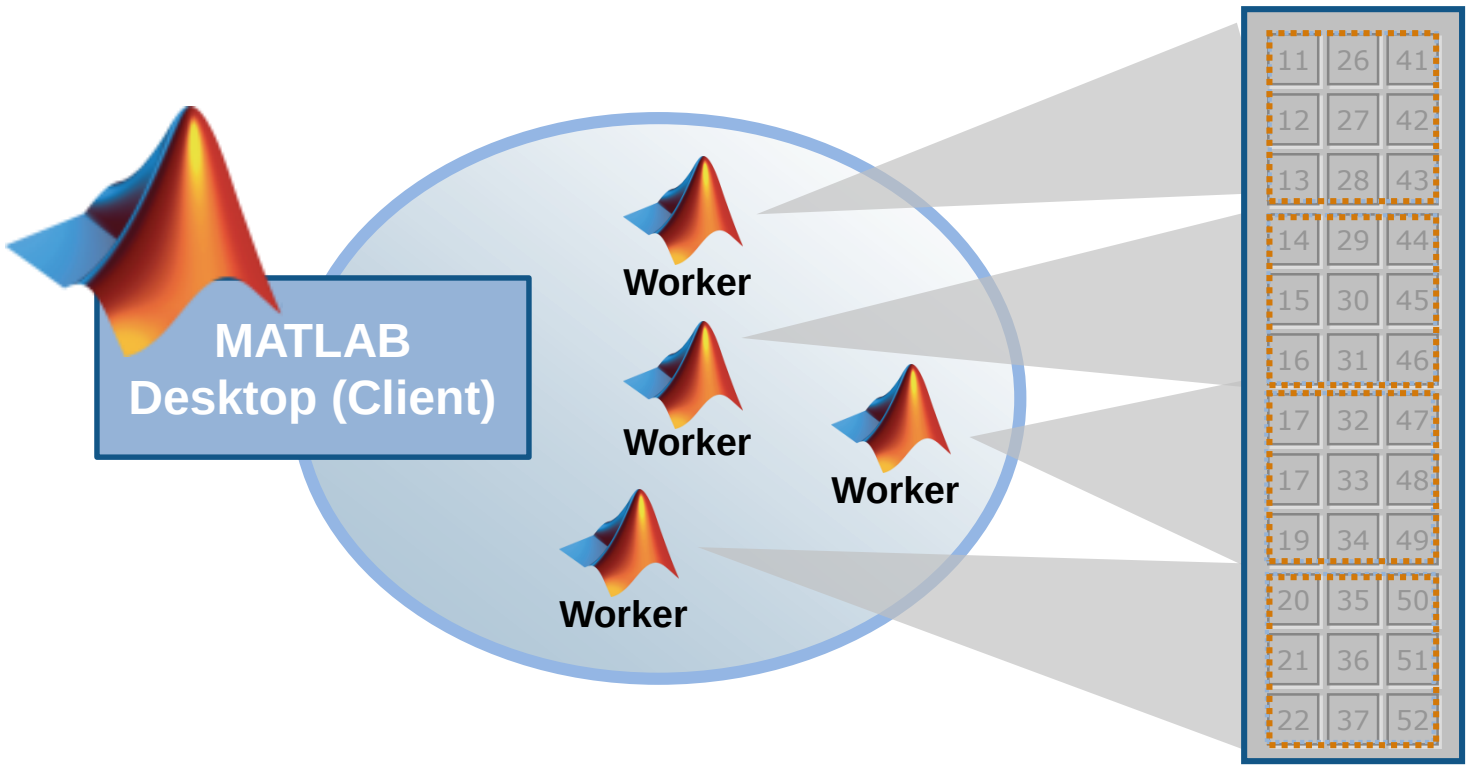
`batch(...)`

Offload and Scale Computations with `batch`




```
batch(…,'Pool',…)
```

Distributing Large Data

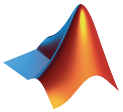


Remotely Manipulate Array from Client

Distributed Array Lives on the Workers

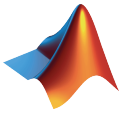
Agenda

Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary



Agenda

Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary



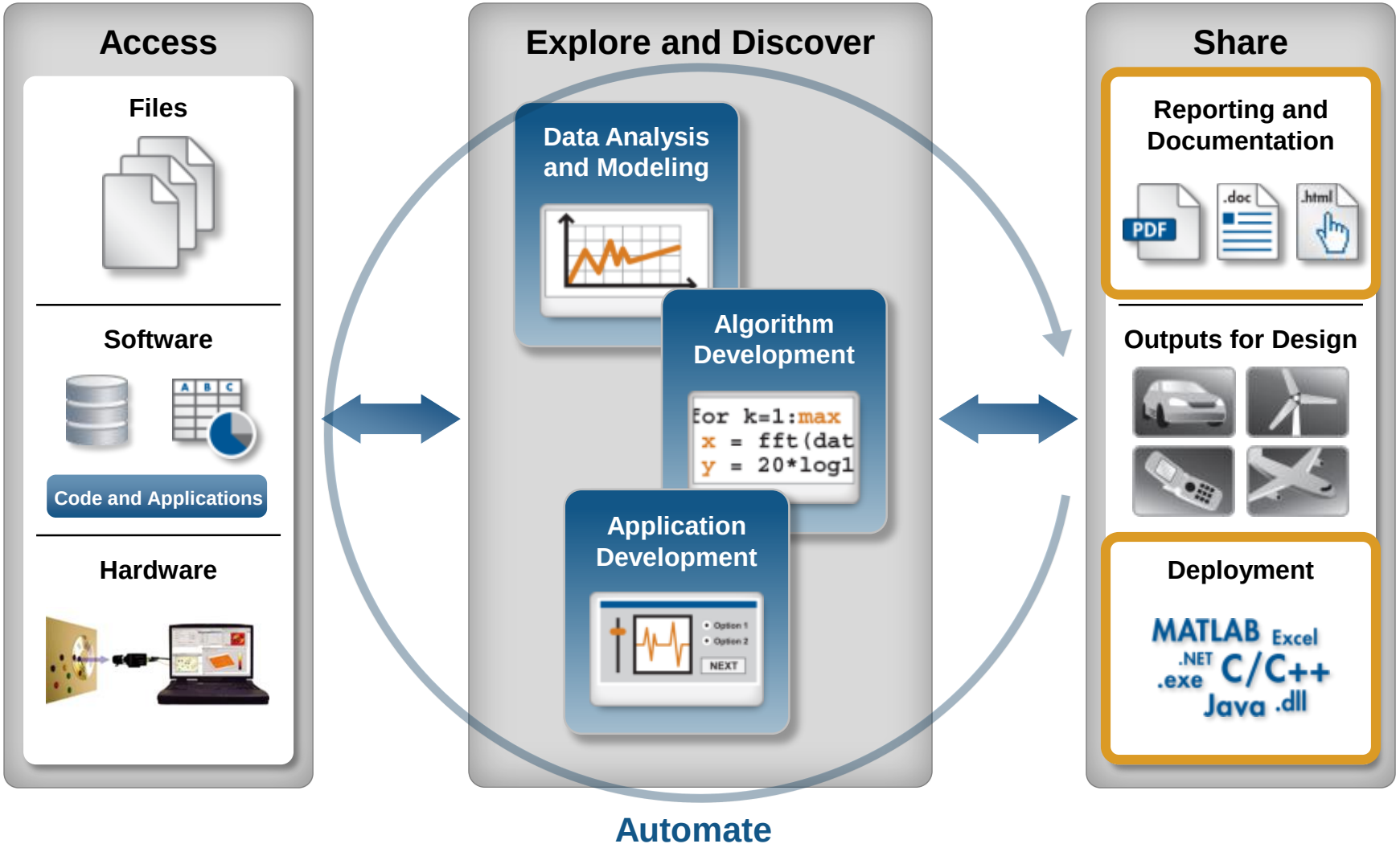
Common Image Processing Challenges

- Reading and writing to various file formats
- Gathering and exploring images to gain insight
- Create, test and debug algorithms

- Refine algorithms, make them robust and re-usable
- Processing large images with limited memory
- Executing algorithms faster

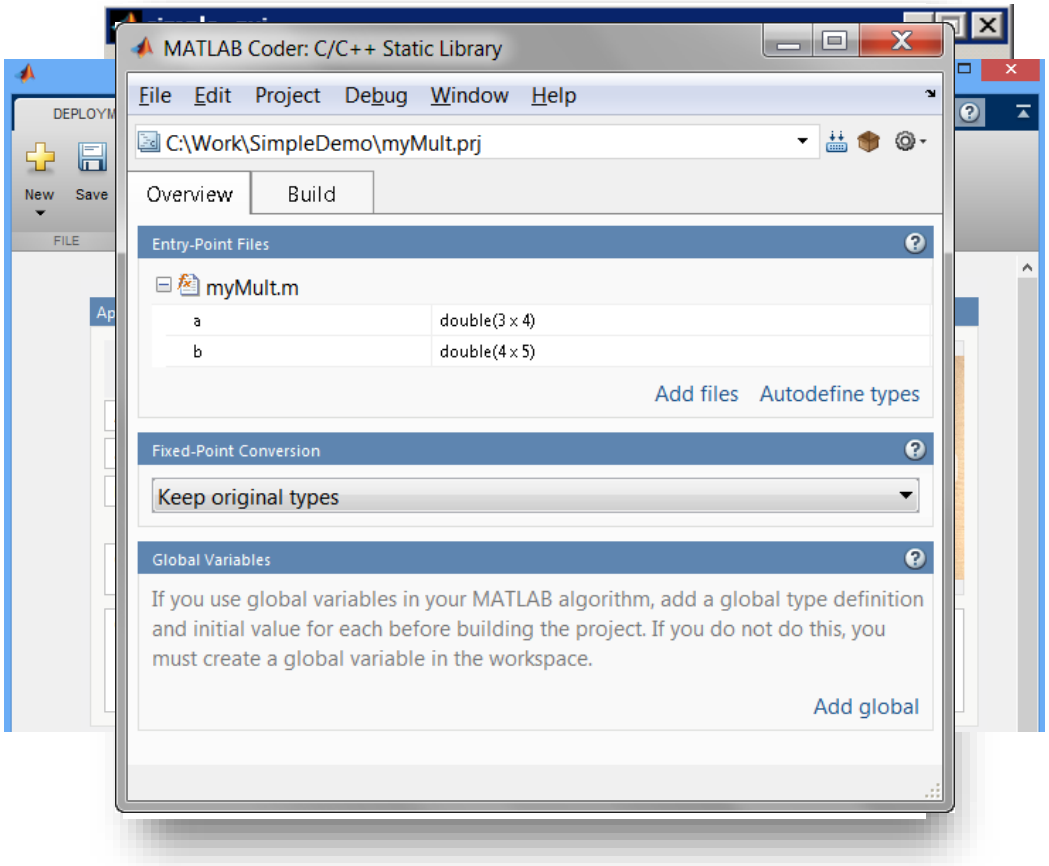
- Sharing results and generating reports
- Creating applications
- Deploy to other environment

Deploying Algorithms and Applications



Next Steps - Sharing Results from MATLAB

- Publish reports
- Package as an App
- Deploy applications
- Deploying C Code



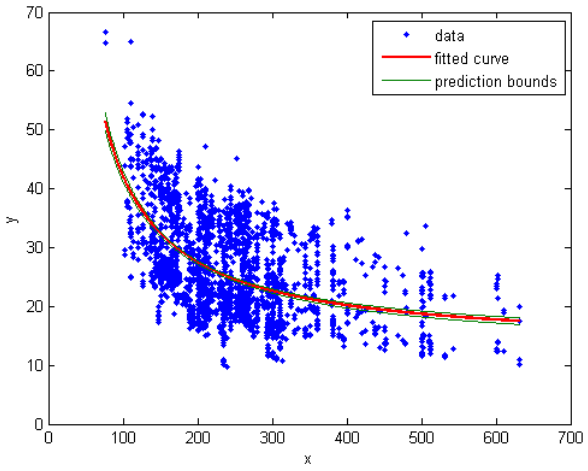
Publish Reports

- Automatically generate reports
  - Publish MATLAB files
  - Detailed comments
  - Code examples
  - Customize reports using MATLAB Report Generator

Plot Data and Model

The result from the Curve Fitting Toolbox has a `plot` method for displaying the result graphically. We can choose to display the prediction bounds for the fit.

```
figure;
hh = plot(cf, 'r', carDataDS.RatedHP, carDataDS.MPG, 'predfunc', 0.95);
set(hh(2)  , 'LineWidth', 2);
set(hh(3:4), 'LineStyle', '-', 'Color', [0 .5 0]);
```

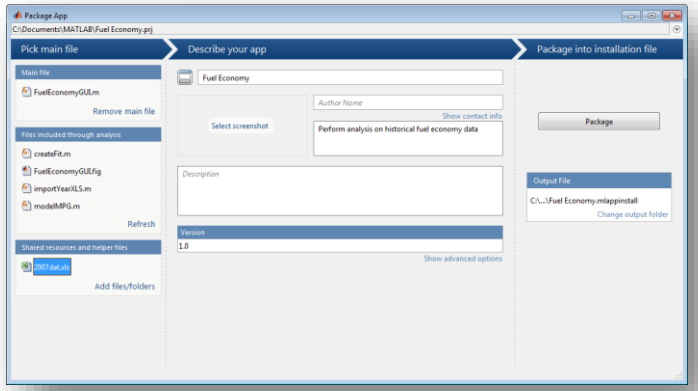


Packaging and Sharing MATLAB Apps

- MATLAB apps
  - Interactive applications to perform technical computing tasks
  - Displayed in apps gallery

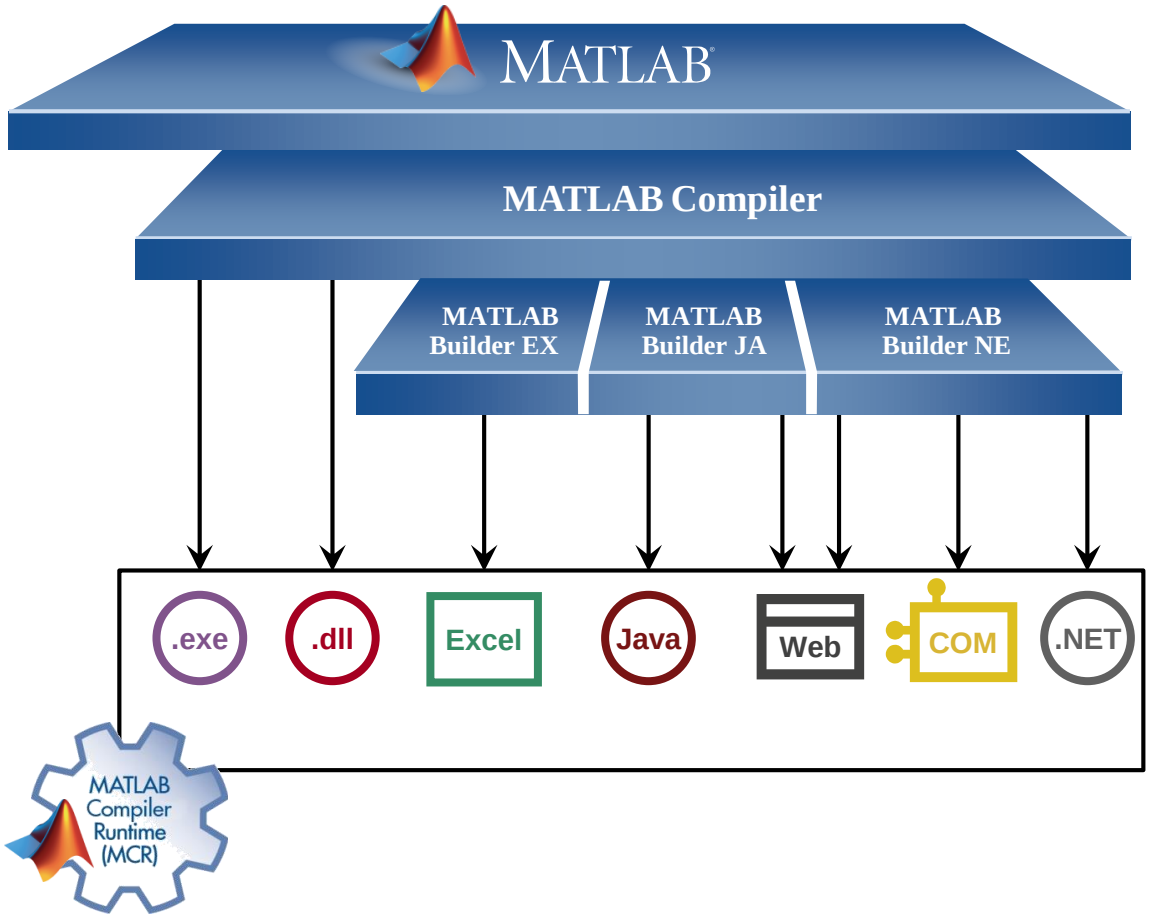
- Included in many MATLAB products

- Package your own app
  - Create single file for distribution and installation into gallery
  - Packaging tool:
    - Automatically includes all necessary files
    - Documents required products



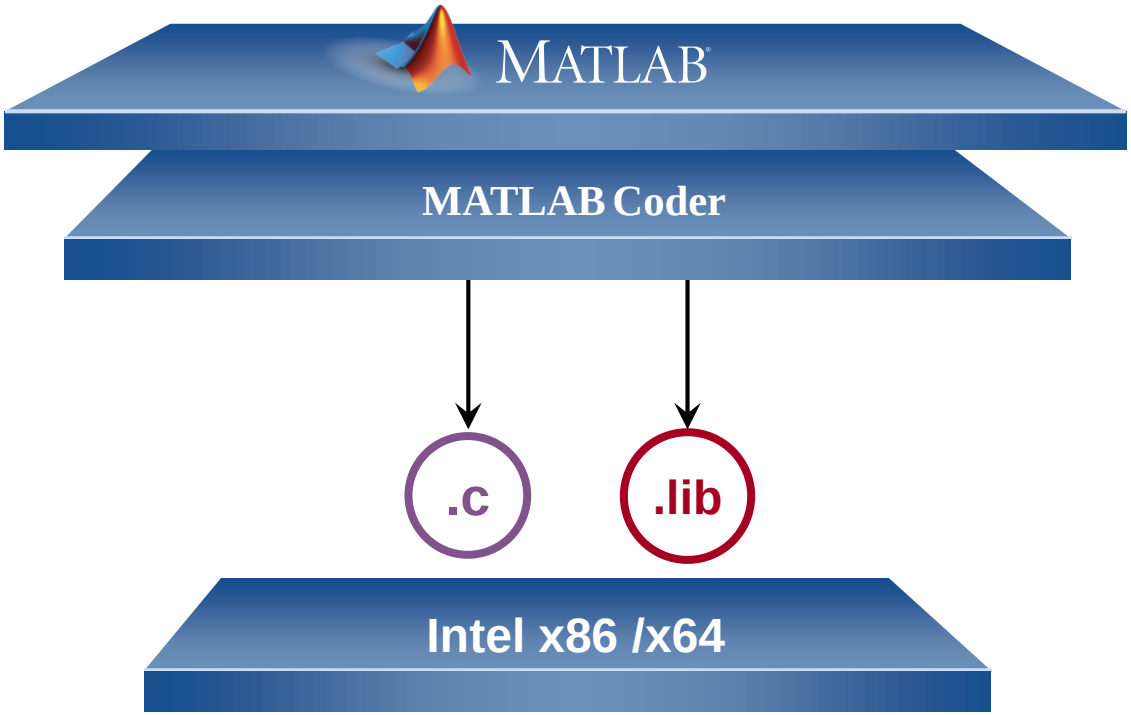
Deploying Applications - MATLAB Compiler

- Share applications
  - Creates desktop or web software components
  - Supports full MATLAB language and most toolboxes
  - Requires MATLAB Compiler Runtime
    - Free run-time library
    - Royalty-free deployment

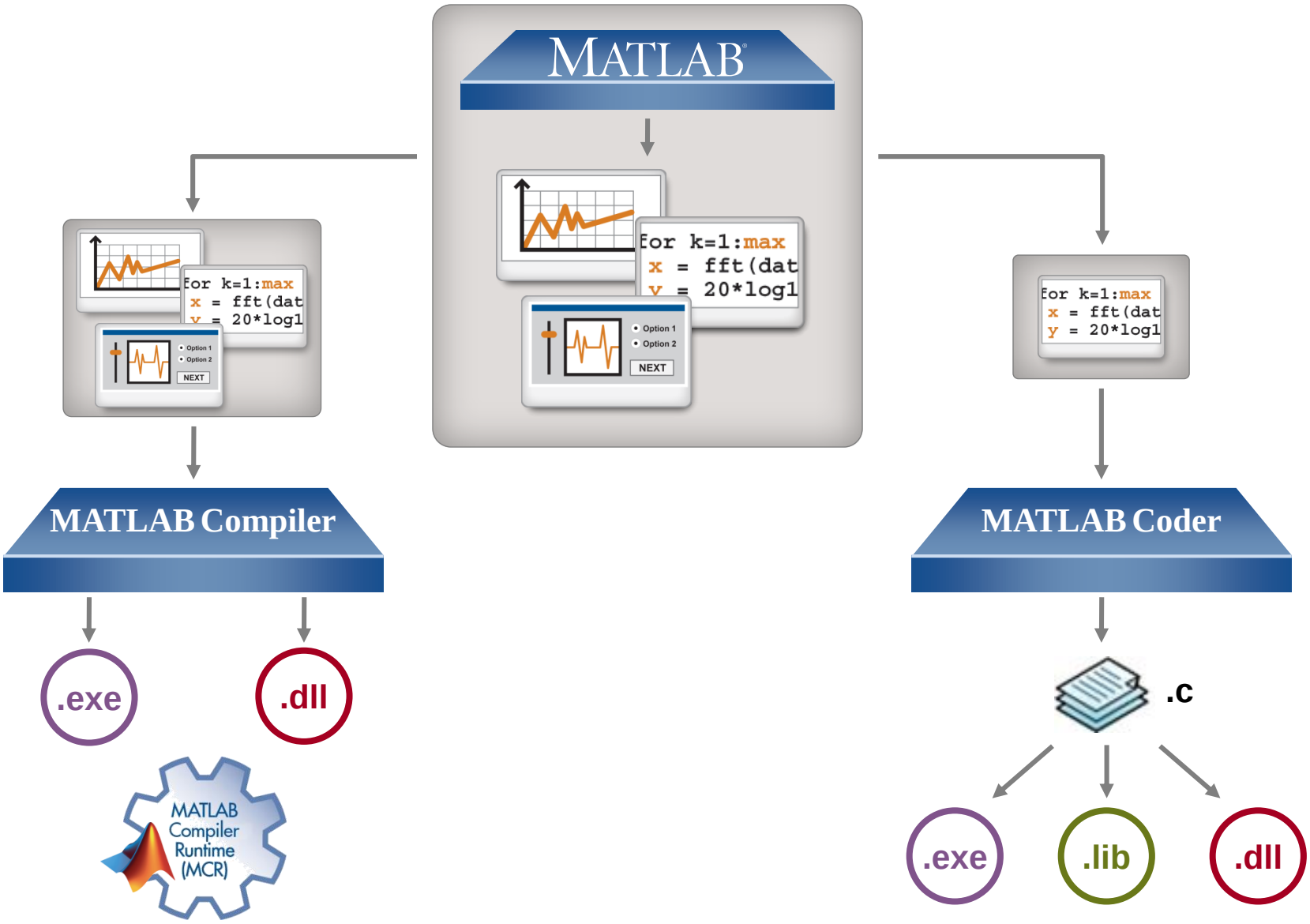


Deploying C Code - MATLAB Coder

- Generate C code
  - High performance C code with shared libraries
  - Integrate with existing software
  - Royalty-free deployment
  - Supports subset of the MATLAB language and some toolboxes



MATLAB Compiler and MATLAB Coder



MATLAB Compiler or MATLAB Coder?

The Middle Ground

MATLAB Compiler

Builders, MPS

Excel

Web

.NET

Java

.exe

Desktop

.dll

.exe

.c

MATLAB Coder

Readable

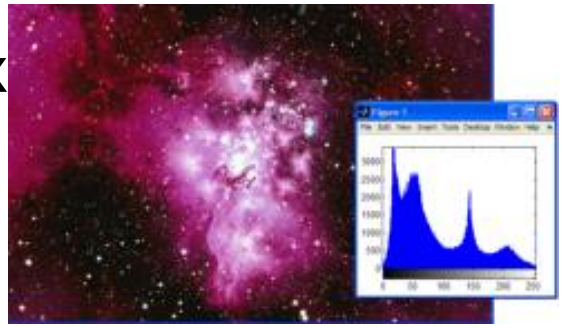
Portable

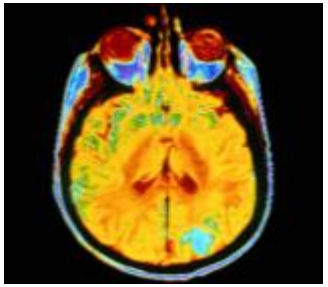
.c

.cpp

Code Generation for 41 IPT Functions

- padarray
- bwmorph
- bwlookup
- conndef
- fspecial
- imcomplement
- imfill
- imhmax
- imhmin
- imreconstruct
- imregionalmax
- iptcheckconn
- label2rgb
- …

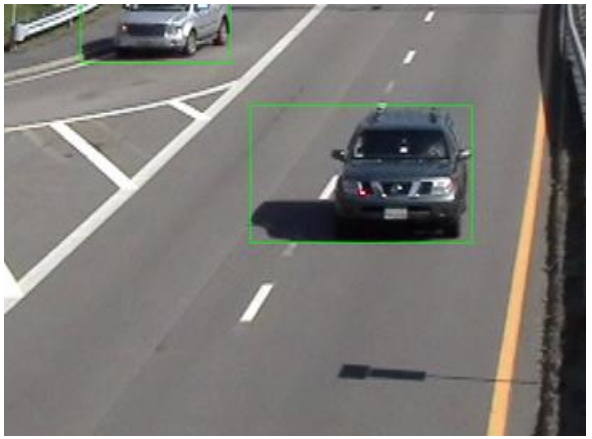


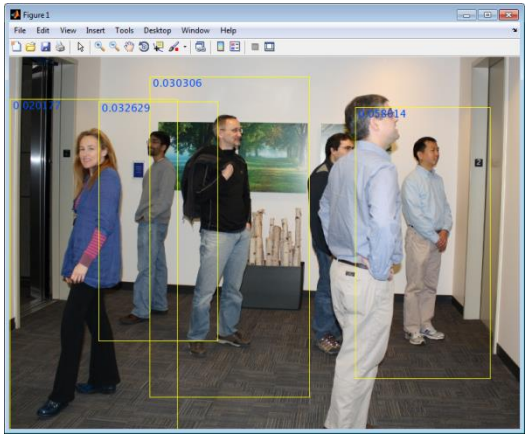


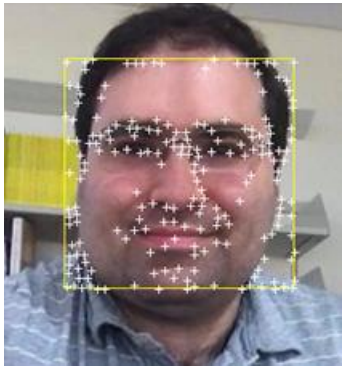
Only for PC platforms – Intel & AMD devices

Code Generation for ALL 86 CVST Functions

- Recently added functions that use OpenCV
- Support now includes
  - vision.ForegroundDetector
  - extractFeatures
  - matchFeatures
  - detectSURFFeatures
  - vision.CascadeObjectDetector







Why Engineers Translate MATLAB to C Today


.exe

Prototype MATLAB algorithms on desktops as standalone executables


.lib
.dll

Integrate MATLAB algorithms with existing C environment using source code and static/dynamic libraries


.c

+


.lib

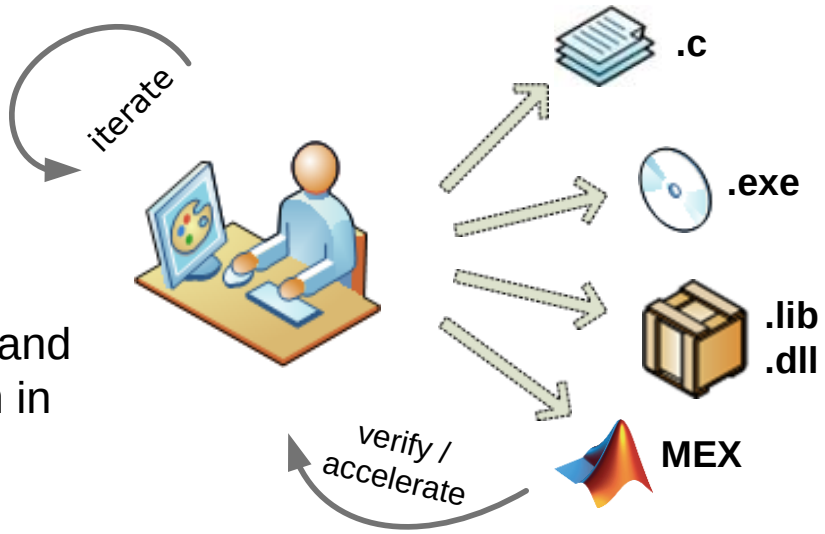
Implement highly optimized C code and Libraries on Intel processors


MEX

Accelerate user-written MATLAB algorithms

Automatic Translation of MATLAB to C

Algorithm Design and Code Generation in MATLAB



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

Choosing the Right Deployment Solution

MATLAB Coder and MATLAB Compiler

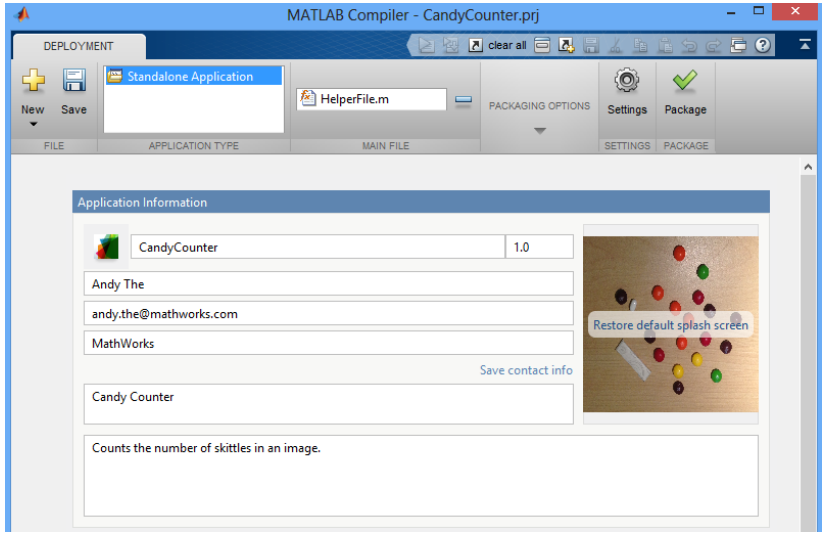
MATLAB Compiler

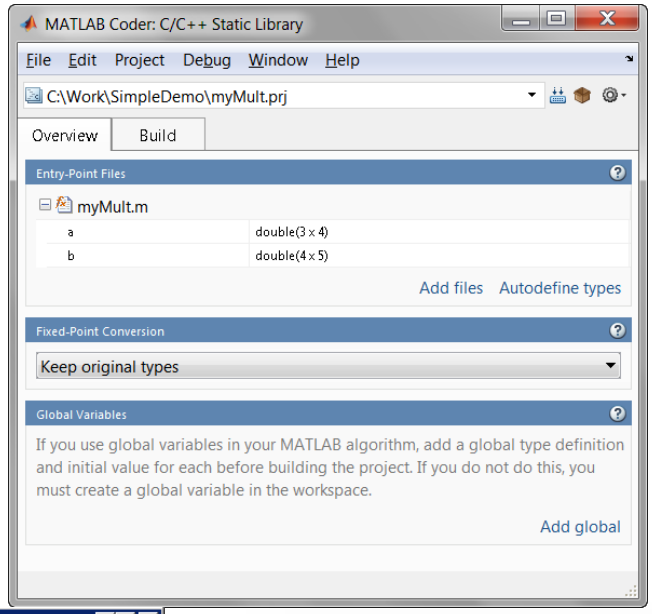
MATLAB Coder

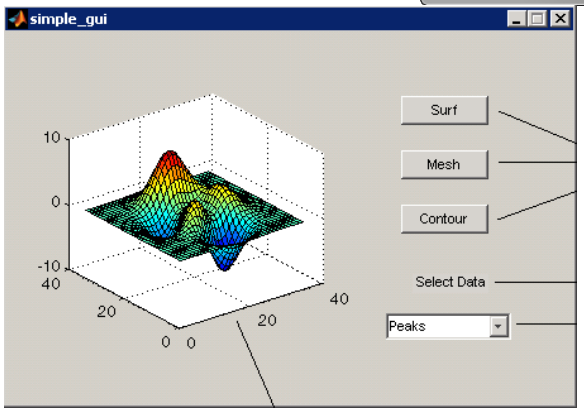
	MATLAB Compiler	MATLAB Coder
Output	Executable or software component/library	Portable and readable C source code
MATLAB support	Full language Most toolboxes Graphics	Subset of language Some toolboxes
Runtime requirement	MATLAB Compiler Runtime	None
License model	Royalty-free	Royalty-free

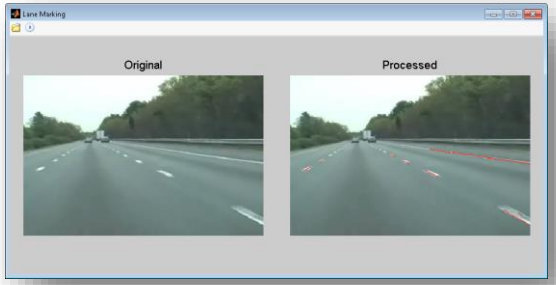
Deployment: Demos

- MATLAB Compiler, MATLAB Coder, and Apps



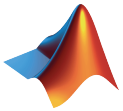






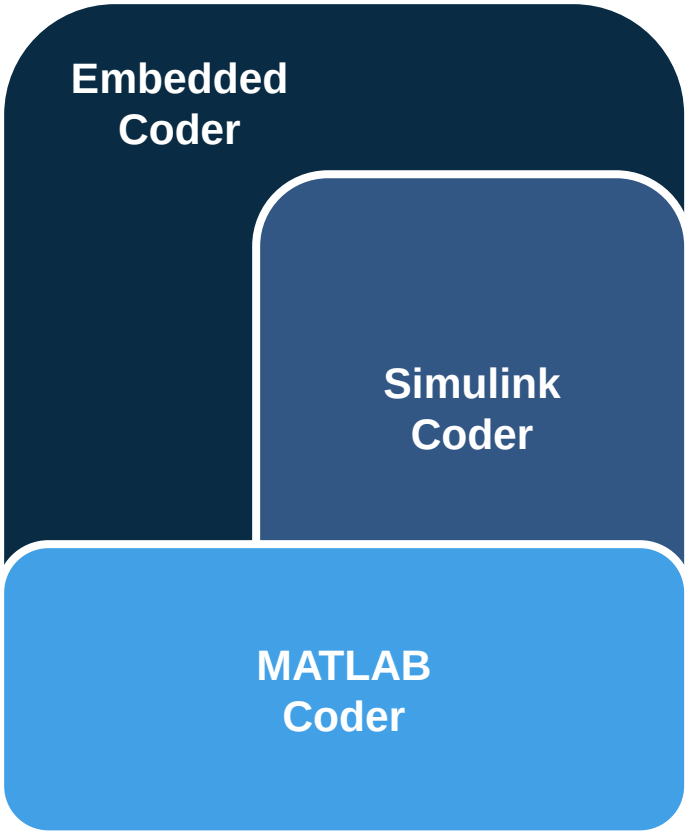
Agenda

Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary



Target External Hardware

Code Generation Technologies



Embedded Coder™

Automatically generate C and C++ optimized for embedded systems comparable to the efficiency of handwritten code

Simulink® Coder™

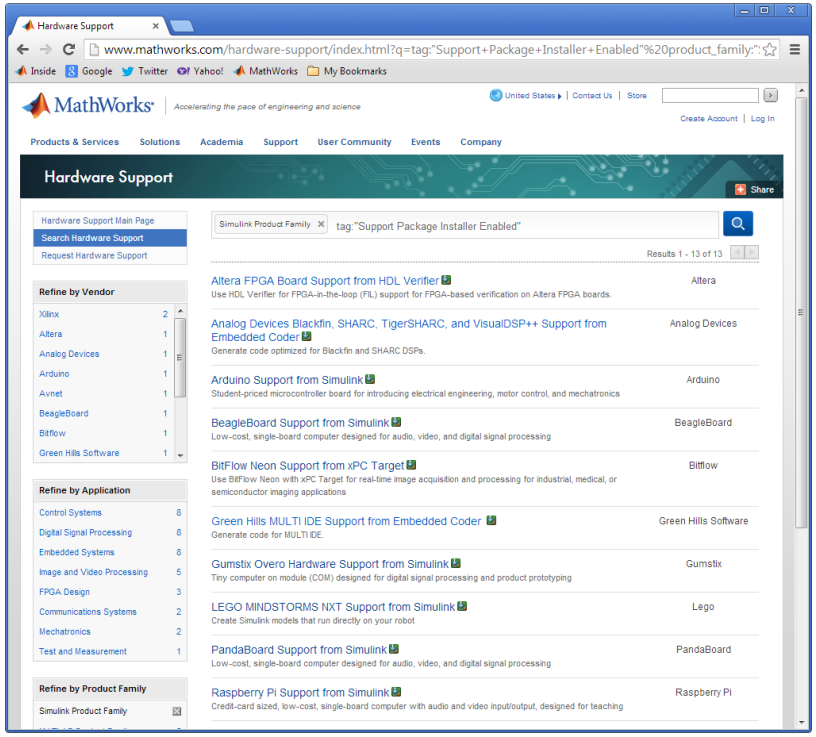
Automatically generate C and C++ from Simulink models and Stateflow charts for Rapid Prototyping and Hardware-in-the-Loop

MATLAB® Coder™

Automatically generate C and C++ from the suitable MATLAB subset

Target External Hardware

Simulink Hardware Support



http://www.mathworks.com/hardware-support/home.html

Arduino

Lego NXT

Raspberry Pi

BeagleBoard

PandaBoard

Gumstix Overo

Selected Simulink Supported Target Hardware:

- Raspberry Pi Model B ($40)
- Arduino® Uno, Mega 2560, Nano, Ethernet Shield ($30-$70)
- LEGO® MINDSTORMS® NXT ($280)
- BeagleBoard-xM ($150)
- PandaBoard ($180)
- Gumstix® Overo hardware ($180)

Available in Student Version!

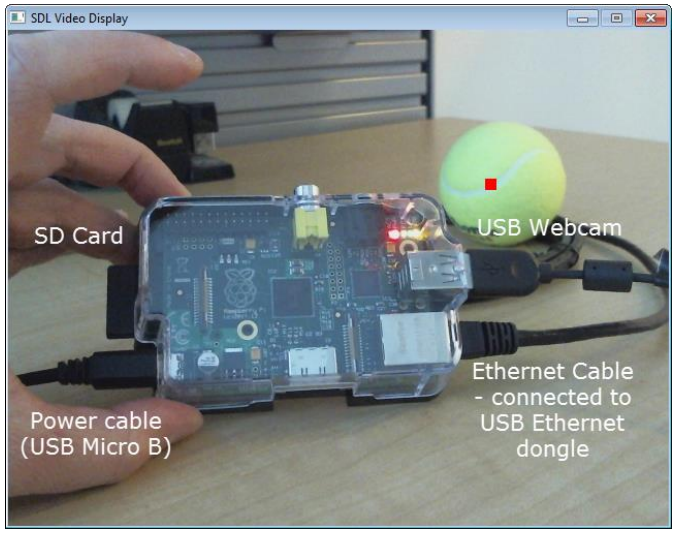
Target External Hardware

Example Raspberry Pi

- Raspberry Pi™ hardware support package
  - Capturing live data from sensors and imaging devices (camera board)
  - Connects to MATLAB over Ethernet (wired or wireless)

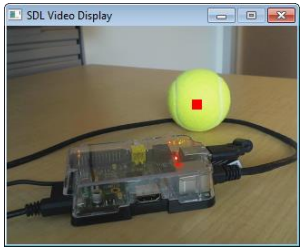
Raspberry Pi Hardware Setup

- Connect Raspberry Pi
  - Plug in USB Ethernet Adapter to PC, and Ethernet cable into adapter and Raspberry Pi
  - Plug USB webcam into Raspberry Pi
  - Insert SD card (which may be a carrier for SD micro) into Raspberry Pi
  - Attach USB power cable – USB A to PC, and USB micro B to power port on Raspberry Pi
    - Wait a minute for Raspberry Pi to boot

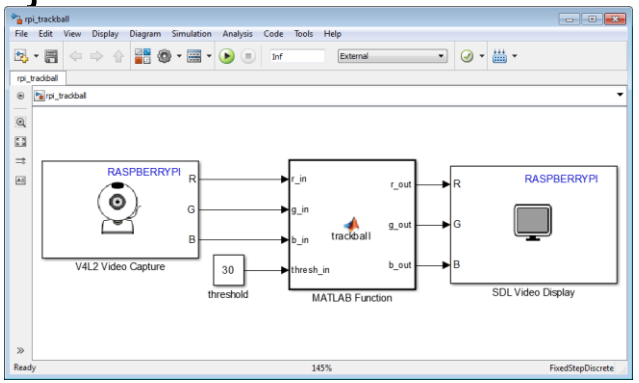


Let's go test it!

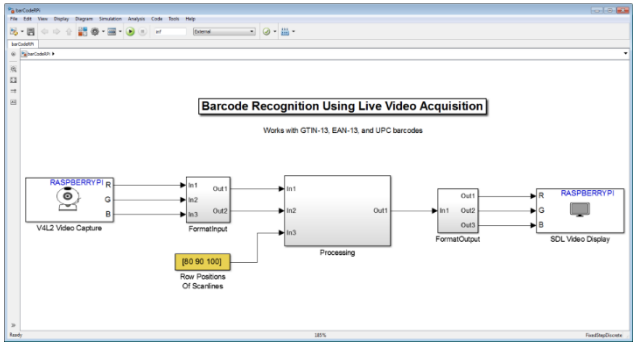
Raspberry Pi Demos



- Object Detection
  - Locate and mark the center of green object



- Barcode scanner
  - Scan barcode and output numerical digits



Agenda

Welcome and Introductions
Image Processing with MATLAB
Computer Vision with MATLAB
Break
Programming Techniques
Speeding-up your Applications
Break
Deploy your Applications
Target External Devices
Summary

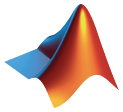
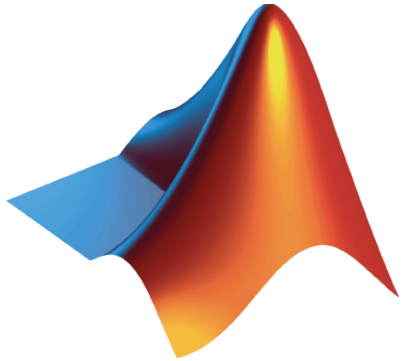


Image and Video Processing with MATLAB

Key Take Aways

- High-level language
  - Native support for vector and matrix operations
  - Built-in math and visualization functions



- Development environment
  - Interactive and easy to get started
  - Ideal for iterative exploration and design

- Technical computing platform
  - Add-on products for a range of application areas
    (e.g., signal processing and communications, image and video processing, control systems, test and measurement)

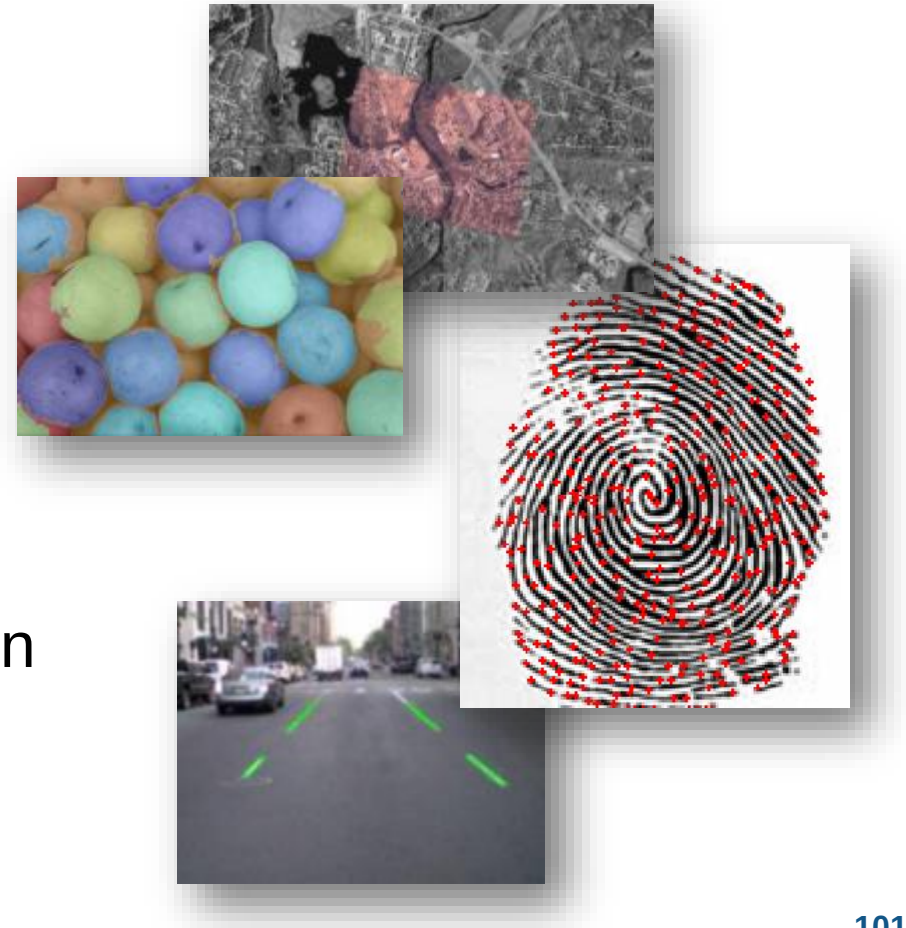
Why use MATLAB for Image Processing?

- Read and write many image file formats
- Visualize and explore images interactively
- Work with live video cameras
- Use a large library of built-in functions
- Quickly build custom image processing algorithms
- Block-process large images to avoid memory issues
- Process images faster with multiple cores and GPUs

Image Processing Toolbox

Perform image processing, analysis, visualization, and algorithm development

- Image analysis
- Image enhancement
- Geometric transformation
- Image registration
- Morphological operations
- ROI-based processing
- Image display and exploration



Why Use MATLAB for Computer Vision?

- Increased productivity over C/C++ programming
  - Faster to prototype
  - More options with the camera
- Comprehensive environment
  - Analysis, algorithm development, visualization, etc.
- Broad library of algorithms
  - Computer vision
  - Image processing
  - Classification and clustering
  - Control system design, optimization, code generation
- Documentation, examples, and technical support
- Integrated workflow from design to implementation

Computer Vision System Toolbox

Design and simulate computer vision and video processing systems

- Feature detection and extraction
- Registration and stereo vision
- Object detection and tracking
- Motion estimation
- Video processing, file I/O, display, and graphic overlays

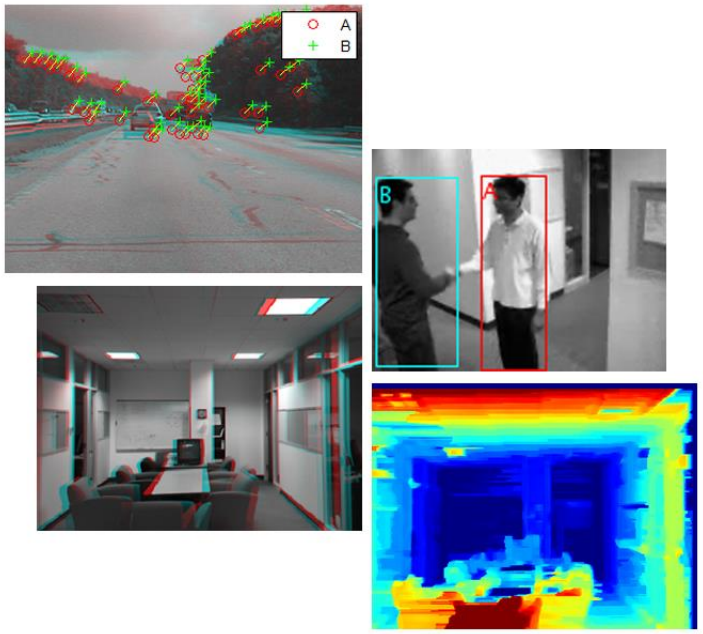


Image Acquisition Toolbox

- Acquire images & video from industry-standard hardware:
  - Camera Link
  - DCAM compatible FireWire (IIDC 1394)
  - GigE Vision
  - GenICam Interface

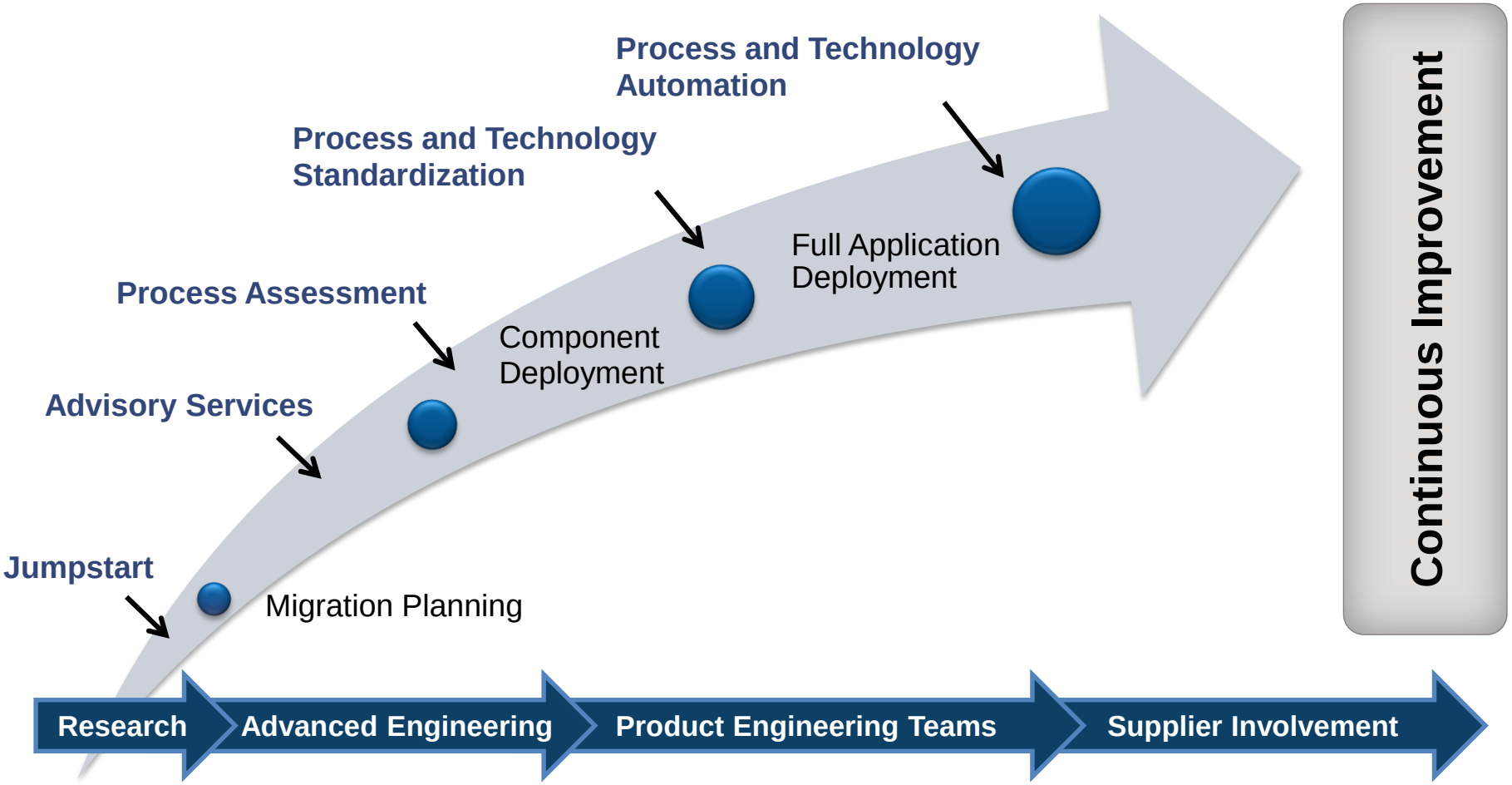
- Manufacturers include:
  - Allied Vision Technologies
  - Basler
  - Baumer
  - FLIR
  - Hamamatsu
  - Point Grey
  - Teledyne DALSA
  - See Hardware Catalog for more details: www.mathworks.com/hardware

Support and Community

Consulting Services

Accelerating return on investment

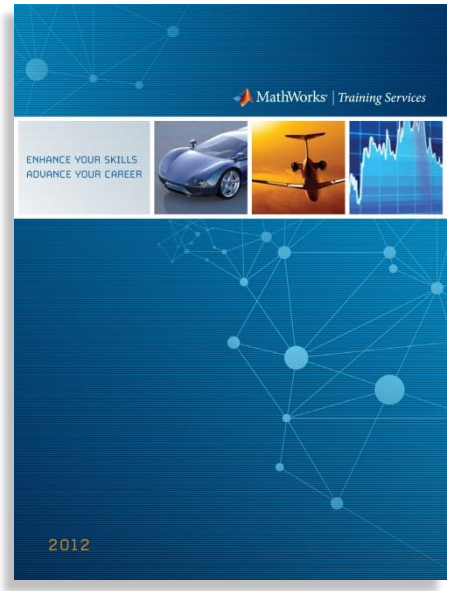
A global team of experts supporting every stage of tool and process integration



Training Services

Exploit the full potential of MathWorks products

- Flexible delivery options:
  - Public training available worldwide
  - Onsite training with standard or customized courses
  - Web-based training with live, interactive instructor-led courses
  - Self-paced interactive online training



- More than 30 course offerings:
  - Introductory and intermediate training on MATLAB, Simulink, Stateflow, code generation, and Polyspace products
  - Specialized courses in control design, signal processing, parallel computing, code generation, communications, financial analysis, and other areas

MATLAB Central

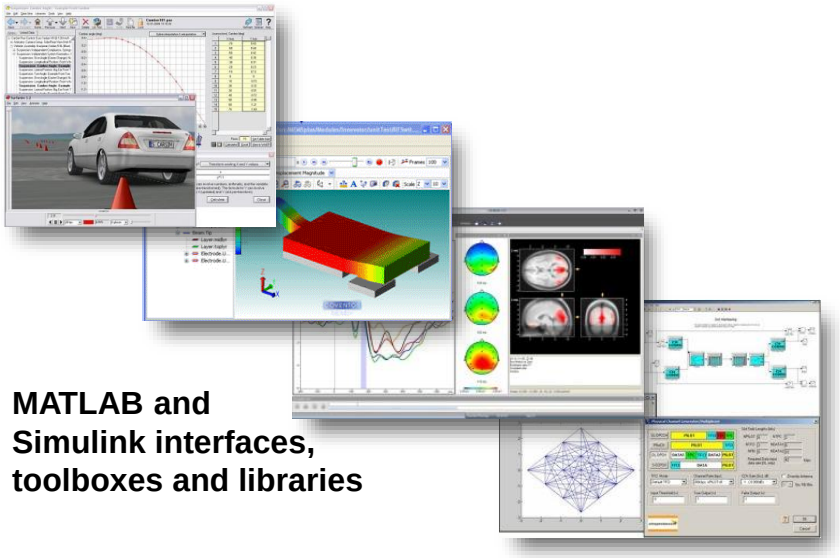
- Community for MATLAB and Simulink users
  - 70k daily visits
- File Exchange
  - Access more than 10k free files including functions, apps, examples, and models
- MATLAB Answers
  - Ask programming questions or search 18k+ community-answered Questions
- Newsgroup
  - Participate in technical discussions
  - 1,400 posts per day
- Blogs
  - Read commentary from engineers who design, build, and support MathWorks products

Based on average 2011 data

Connections Program

More than 400 add-on products and services that complement and extend MathWorks products:

- Third-party toolboxes and libraries for MATLAB and Simulink
- Interfaces to third-party software and hardware products
- Specialized training courses and consulting services
- System integrators that incorporate MathWorks products



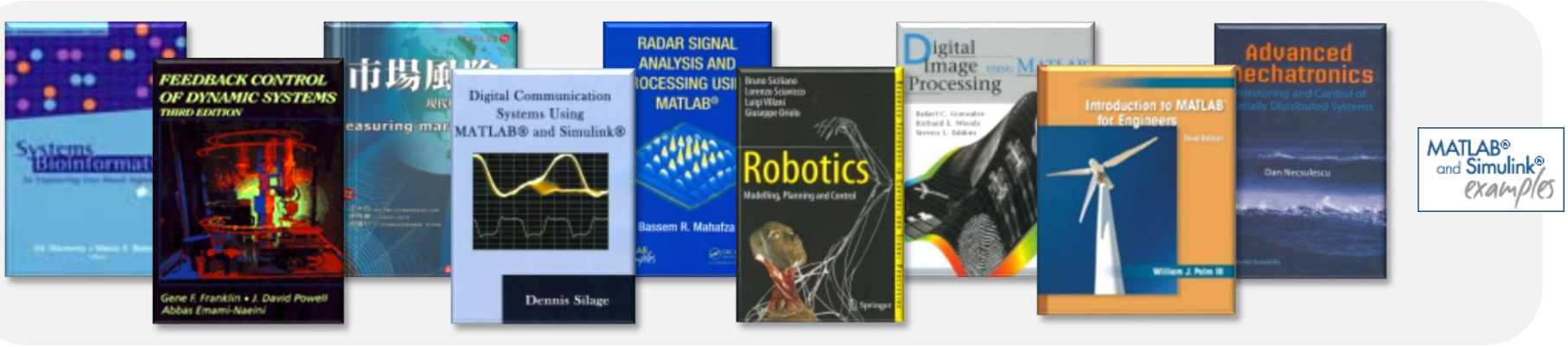
MATLAB and Simulink interfaces, toolboxes and libraries

Embedded processors, real-time systems and instrumentation

Book Program

More than 1400 books for educational and professional use, in 28 languages

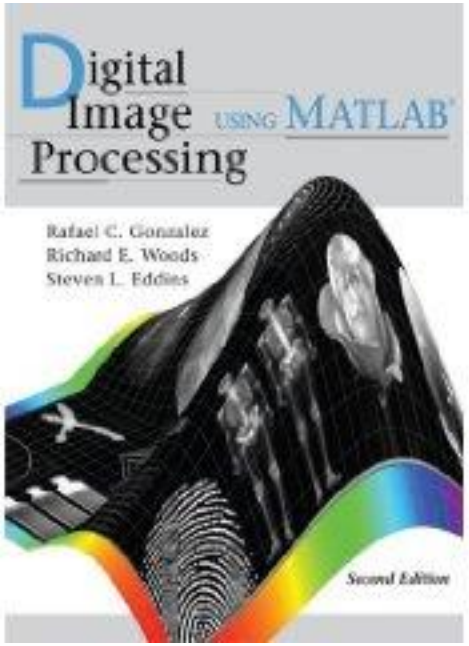
- **Control systems**
- **Digital signal processing**
- **Image & video processing**
- **Biosciences & biomedical**
- **Communications systems**
- **Mechanical engineering**
- Mathematics
- Physics
- Systems modeling
- Chemistry
- Computational finance
- Electronics



From a Leading Textbook Author ...

"I have used a number of commercial image processing packages over the years, and prefer the MathWorks Image Processing Toolbox for several reasons: the wide variety of functions it provides, the user's ability to write additional functions with minimal effort, the quality of the software, and the high level of support."

Rafael C. Gonzalez
Professor Emeritus
University of Tennessee



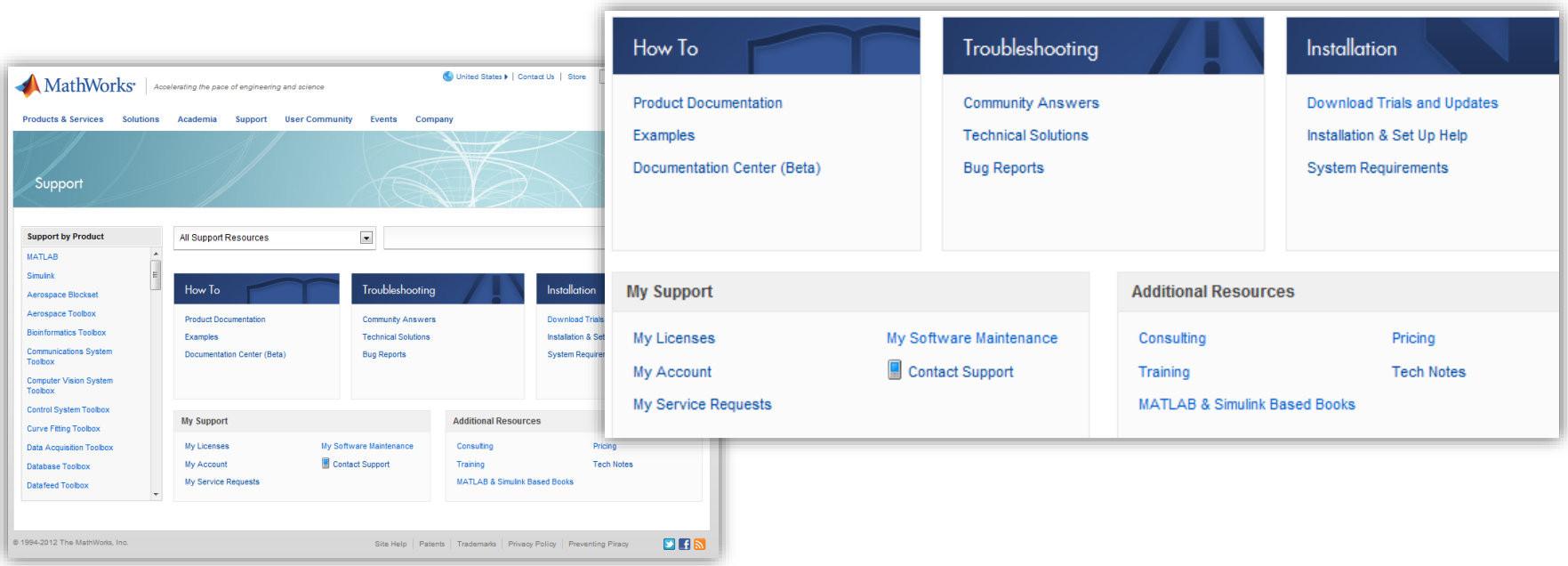
Technical Support

Resources

- Over 100 support engineers
  - All with MS degrees (EE, ME, CS)
  - Local support in North America, Europe, and Asia
- Comprehensive, product-specific Web support resources

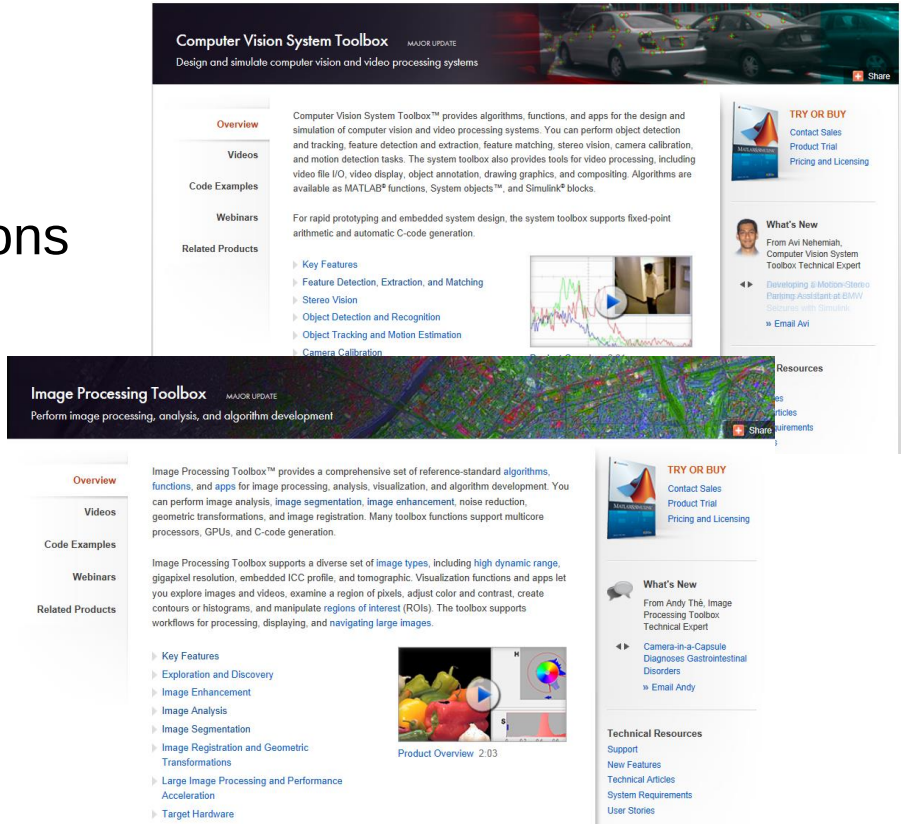
High customer satisfaction

- **95% of calls answered within three minutes**
- **70% of issues resolved within 24 hours**
- **80% of customers surveyed rate satisfaction at 80–100%**



For More Information

- Experiment with product by downloading a trial
- Peruse videos, webinars, user stories, and demos online
- Contact us
  - Talk to a sales representative to get answers to your questions
  - Discuss your projects with MathWorks applications engineers



Thank You for Attending Today's Seminar

Pete Tsinzo

Account Manager

Email: Pete.Tsinzo@mathworks.com

Phone: 310-819-4996

Add relevant info here

Customer Service

Email: service@mathworks.com

Phone: 508.647.7000 option 1

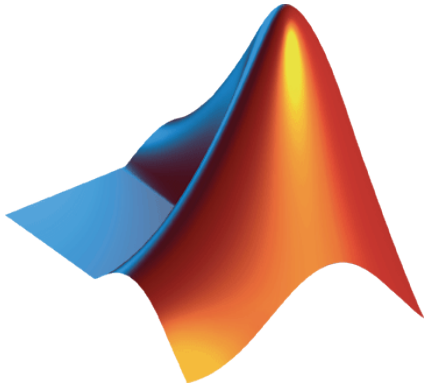
MathWorks

www.mathworks.com

Technical Support

Email: support@mathworks.com

Phone: 508.647.7000 option 2



Dr. Roland Michaely

Application Engineer