Computer Vision with MATLAB

Master Class

Bruce Tannenbaum
Agenda

- Introduction
- Feature-based registration
  - Automatic image registration
  - Rotation correction with SURF
  - Stereo image rectification
- Video processing with System objects
  - Tracking cars with optical flow
- Classification
  - Texture classification
  - Face detection
- Summary
Examples of Computer Vision with MATLAB
Computer Vision

Using images and video to detect, classify, and track objects or events in order to “understand” a real-world scene

**Image Processing**
- Remove noise
- Adjust contrast
- Measure
  ...

**Computer Vision**
- Detect
- Identify
- Classify
- Recognize
- Track
  ...

**Interpretation**
- Pedestrian
- Bicyclist
- Truck
- Car
- Traffic violation
- Accident
  ...
Typical Computer Vision Challenges

- Variable lighting conditions
- Unknown scene depth or perspective
- Background clutter
- Partially hidden objects (occlusion)
- Differences in scale, location, and orientation
Technical Computing with MATLAB

Access
- Files
- Software
  - Code and Applications
- Hardware

Explore and Discover
- Data Analysis and Modeling
- Algorithm Development
  - For k=1:max
  - x = fft(data)
  - y = 20*log10
- Application Development

Share
- Reporting and Documentation
  - PDF
  - .doc
  - .html
- Outputs for Design
- Deployment
  - MATLAB
  - Excel
  - .NET
  - C/C++
  - Java
  - .dll
Key Products for Computer Vision

- Computer Vision System Toolbox - NEW
- Image Processing Toolbox
- MATLAB
- Statistics Toolbox

- Additionally…
  - Image Acquisition Toolbox
  - MATLAB Coder
  - Parallel Computing Toolbox
Computer Vision System Toolbox

Design and simulate computer vision and video processing systems

- Feature detection
- Feature extraction and matching
- Feature-based registration
- Motion estimation and tracking
- Stereo vision
- Video processing
- Video file I/O, display, and graphics
Demo: Feature-Based Registration

- **Workflow**
  - Feature detection
  - Feature extraction
  - Feature matching
  - Geometric transformation estimation with RANSAC
Demo: Rotation Correction with SURF

- **Workflow**
  - Feature detection
  - Feature extraction
  - Feature matching

![Matching points (inliers only)](image1)

*Image courtesy of Massachusetts Institute of Technology*
Demo: Stereo Image Rectification
Recovering Scene Depth with Stereo Cameras
Epipolar Geometry
Fundamental Matrix

\[ X_L^T F X_R = 0 \]
Video Processing

- Video file I/O and display
- Video pre-processing
- Motion estimation and analysis
Motion Estimation and Analysis

- Techniques
  - Block matching
  - Optical flow
  - Template matching
  - Background estimation using Gaussian mixture models

- Applications
  - Object tracking
  - Interpolation
  - Compression
Demo: Using Optical Flow to Track Cars

- Video file I/O and display
- Video preprocessing
- Motion estimation
- Segmentation and analysis
Useful System Objects for Video File I/O, Display, and Graphics

- **File I/O**
  - VideoFileReader
  - VideoFileWriter

- **Display**
  - VideoPlayer
  - DeployableVideoPlayer

- **Graphics**
  - AlphaBlender
  - MarkerInserter
  - ShapeInserter
  - TextInserter
Useful System Objects for Video Preprocessing and Statistics

- **Preprocessing**
  - ChromaResampler
  - Deinterlacer
  - DemosaicInterpolator

- **Statistics (running across video frames)**
  - Histogram
  - Maximum
  - Mean
  - Median
  - Minimum
  - StandardDeviation
  - Variance
## Different Interfaces, Different Benefits in Computer Vision System Toolbox

<table>
<thead>
<tr>
<th>Audience</th>
<th>Functions</th>
<th>System Objects</th>
<th>Simulink Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm developers</td>
<td>• Application-specific algorithms and tools</td>
<td>• Algorithms that maintain state</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Efficient video stream processing</td>
<td></td>
</tr>
<tr>
<td>System designers</td>
<td></td>
<td>• Fixed-point modeling</td>
<td>• Multidomain modeling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• C-code generation</td>
<td>• Real-time system design</td>
</tr>
<tr>
<td>Implementers</td>
<td></td>
<td></td>
<td>• Target-specific embedded hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• HIL, PIL</td>
</tr>
</tbody>
</table>
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
Challenge: Accurate Classification is Hard

How can a computer tell that these are all chairs?
Demo: Texture Classification

- Identify features appropriate for classification
- Extract features for training and test data
- Train classifier with features
- Test classifier and analyze results

- Using KTH-TIPS database


Demo: Face Detection
Statistics Toolbox

Perform statistical analysis, modeling, and algorithm development

- Clustering
  - Principle components analysis
  - K-means
  - Gaussian mixture models

- Classification
  - Naïve Bayes
  - K-nearest neighbor search
  - Boosted decision trees
    - AdaBoost, GentleBoost, LogitBoost,…
Key Products for Computer Vision

- Computer Vision System Toolbox - **NEW**
- Image Processing Toolbox
- MATLAB
- Statistics Toolbox

- Additionally…
  - Image Acquisition Toolbox
  - MATLAB Coder
  - Parallel Computing Toolbox
Why Use MATLAB for Computer Vision?

- Comprehensive environment
  - Analysis, algorithm development, visualization, etc.

- Broad library of algorithms
  - Computer vision
  - Image processing
  - Classification and clustering

- Documentation, examples, and technical support

- Increased productivity over C/C++ programming
For More Information

- mathworks.com/products/computer-vision
- Relevant demos:
  - Barcode Recognition
  - Image Rectification
  - Traffic Warning Sign Recognition
  - People Tracking
  - Video Mosaicking
- Documentation
- Contact your sales representative
Questions and Answers