Matlab Tutorial

Course web page:

www.cis.udel.edu/~cer/arv
Announcements

• First paper presentations are allocated—two students per paper
• Homework assigned today is due next Thursday
<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Speaker/Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thurs., Sept. 10</td>
<td>“Automatic Mosaic Creation of the Ocean Floor”</td>
<td>Thomen Korah, Bill Ulrich</td>
</tr>
<tr>
<td>Tues., Oct. 15</td>
<td>“Appearance-Based Place Recognition for Topological Localization”</td>
<td>Joe Kirk, Chaitanya Ramineni</td>
</tr>
<tr>
<td></td>
<td>“Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization”</td>
<td>Wei Zhou, Cunjie Zhu</td>
</tr>
<tr>
<td>Thurs., Oct. 17</td>
<td>“Robust Car Tracking Using Kalman Filtering and Bayesian Templates”</td>
<td>Min Li, Chaitra Gowri Murthy</td>
</tr>
<tr>
<td>Tues., Oct. 22</td>
<td>“Precise Image-based Motion Estimation for Autonomous Small Body Exploration”</td>
<td>Vishal Arora, Qi Li</td>
</tr>
<tr>
<td></td>
<td>“Vision-Guided Flight Stability and Control for Micro Air Vehicles”</td>
<td>Ibrahim Halil Saruhan, Xiuman Zhu</td>
</tr>
<tr>
<td>Tues., Sept. 19</td>
<td>“Video Mosaics for Virtual Environments”</td>
<td>Instructor</td>
</tr>
<tr>
<td>Thurs., Sept. 26</td>
<td>“Stochastic Road Shape Estimation”</td>
<td>Instructor</td>
</tr>
</tbody>
</table>
About the First Paper

• Pairs work together on this one!
• Paper write-up and preliminary version of presentation must be turned in to and discussed with the instructor one week before scheduled date
• Oral presentation
  – 30 minutes long (divide speaking time)
  – Use visual aids
  – Be ready to lead discussion and field questions
Paper Write-Ups

• 3-4 pages of prose, jointly written

• Questions
  – What is the problem(s) the authors are trying to solve?
  – What technical methods do they use? Explain unfamiliar or novel techniques and cite sources
  – How well do they succeed? What limitations does their approach have? How might it be applied to other robot vision problems?
  – Are there interesting links to other papers and methods we are studying?
  – How could the work be extended or improved?
Looking up Sources

• UD library
  - Online: Subscriptions to many journals with full-text PDF (www.lib.udel.edu; go to “Subject Guides” and “Electronic Journals”)
  - Hard-copy: Some journals, conference proceedings, vision & robotics texts (and of course related math)

• Citeseer
  - Best bet for conference papers
  - Useful for tracking citations (forward and backward)

  citeseer.nj.nec.com/cs
Second paper

- You’re on your own!
  - Pick something related to project
  - Must be substantially different from first paper topic
  - 1-2 page write-up, 15 minute presentation
    (on the paper—you will present your project results at the very end)
- Same order of presentation as first paper
Meet Matlab

- A high-level language for matrix calculations, numerical analysis, & scientific computing
- Mathematica: More concerned with symbolic math, but increasing overlap between the two
- Language features
  - No variable declarations
  - Automatic memory management (but preallocation helps)
  - Variable argument lists control function behavior
  - Vectorized: Can use for loops, but largely unnecessary (and less efficient)
Running Matlab at UD

- Unix: Installed on Strauss & Mahler (Ren & Stimpy, too)
  - X windows: Type "matlab" at a prompt
  - Terminal: "matlab -nodesktop"—but no graphics, of course
- PC: Should be able to get CDs and install on network-connected machine; contact Anita Schwartz (anita@udel.edu)
- More information on Matlab at UD: First two links on "Matlab links" page
How to Get Help

• In Matlab
  - Type “help” to get a listing of topics
  - “help <topic>” gets help for that topic. The information is at the level of a Unix man page

• On the web
  - “Matlab links” on course web page has pointers
  - Especially MathWorks help desk:

www.mathworks.com/access/helpdesk/help/helpdesk.shtml

  - There’s always Google…but be careful
Entering Variables

• Entering a vector, matrix
  \[ V = [10, 4.5, 1]; \]
  \[ M = [3, 4; -6, 5]; \]

• Without semi-colon, input is echoed (this is bad when you’re loading images!)

• Comma to separate statements on same line

• size: Number of rows, columns
Constructing Matrices

- Basic built-ins:
  - All zeroes, ones: `zeros`, `ones`
  - Identity: `eye`
  - Random: `rand` (uniform), `randn` (unit normal)

- Ranges: `m:n`, `m:i:n` (i is step size)

- Composing big matrices out of small matrix blocks

- `repmat(A, m, n)`: “Tile” a big matrix with \( m \times n \) copies of A
Manipulations & Calculations

- Transpose (‘), inverse (inv)
- Matrix arithmetic: +, -, *, /, ^
- Elementwise arithmetic: .*, ./, .^  
- Functions
  - Vectorized
  - sin, cos, etc.
Deconstructing Matrices

- Indexing individual entries by row, col: \( A(1, 1) \) is upper-left entry
- Ranges: e.g., \( A(1:10, 3), A(:, 1) \)
- Matrix to vector and vice versa by column: \( B = A(:), A(:) = B \)
  - Transpose to use row order
- `find`: Indices of non-zero elements
Matrix Analysis

- Basics (by column)
  - norm
  - max, min
  - sum

- More advanced
  - Linear systems: \( A\backslash b \) solves \( A\times x = b \)
  - QR decomposition: \( qr \)
  - Singular value decomposition: \( svd \)
  - Eigenvalues: \( eig \)
  - Etc.
Control Structures

• Expressions, relations (==, >, |, &, functions, etc.)

• if/while expression statements end
  - Use comma to separate expression from statements if on same line
  - if a == b & isprime(n), M = inv(K);
    else M = K; end

• for variable = expression statements end
  - for i=1:2:100, s = s / 10; end
M-Files

• Any text file ending in “.m”
• Use `path` or `addpath` to tell Matlab where code is (non-persistent?)
• Script: Collection of command line statements
• Function: Take argument(s), return value(s). First line defines:
  – function `y = foo(A)`
  – function `[x, y] = foo2(a, M, N)`
• Comment: Start line with `%"
Plotting

- **2-D vectors**: \( \text{plot}(x, y) \)
  - \( \text{plot}(0:0.01:2*\pi, \sin(0:0.01:2*\pi)) \)

- **3-D**: \( \text{plot3}(x, y, z) \) (space curve)

- **Surfaces**
  - \text{meshgrid} makes surface from axes, \text{mesh} plots it
    - \([X,Y] = \text{meshgrid}(-2:.2:2, -2:.2:2); \)
    - \( Z = X .* \exp(-X.^2 - Y.^2); \)
    - \text{mesh}(Z)

- **surf**: Solid version of \text{mesh}

- **Saving figures, plots**: \text{print} –depsc2 \text{filename}
Miscellaneous

- Diary: Recording a session
  - `diary filename`
  - `diary off`
- `tic, toc` bracketing code to measure execution time
Image Processing Toolbox

• Loading, displaying images:
  \( I = \text{imread}('im1.jpg'), \text{imshow}(I) \)

• Saving images: \( \text{imwrite}(I, 'newim.jpg') \)

• Image representation
  – Grayscale: Matrix of \text{uint8}
  – Color: Stack of 3 matrices for R, G, and B

• Conversion: \( I_2 = \text{double}(I_1) \)

• More in the next lecture
First Homework

• Implement DLT homography estimation in Matlab
• Rectify three test images
• Just map the points, but feel free to use bilinear interpolation for nicer looking output
• Use `ginput` for mouse clicks in Matlab to record correspondences
Unrectified Images