

Introduction to Matlab for Computer Vision cs461
By Daniel Mirota dan@cs.jhu.edu

Outline

- Basics
- Help system
- Scripts vs. Functions
- Indexing vs. For loops
- Plotting
- Writing, displaying and opening images
- Debugging

Basics

Directory panel – Is here for your straight forward Directory navigation

Workspace panel – Is where you will find all of the variables in the current workspace or stack location if you are in a program with multiply function calls. This leads me to the Command window. <Create Variable>

Command window – In the command window you can enter anything you write in a program and Matlab will evaluate it for you. Example `x = 1 + 1`. Now you can see the variable `x` in the workspace and the output in the command window. Example `x = 1 + 1;` With a semicolon the outout is suppressed. Similar to most command prompts you can hit up for previous commands.

Command history – Now, in the command history you can see both of these lines.

Help system – beyond basic math Matlab has many functions and I use the help system periodically for commands I'm not familiar with. `fprintf`, for example is something you might want to look up when you are outputting to a file. You can use `help fprintf` which will give you something similar to man pages if you are familiar with UNIX or for the full Documentation, use `doc fprintf`. With the full Documentation you can browser through the help files.

Scripts vs. Functions – In Matlab the programs you write are saved as an `.m` files which can be a scripts or functions To create a basic script click the new document button and it opens the editor and you enter the same you would enter into the command windows. It is just a list of commands you would like saved. Then save the script and you can call it from the command window.

Then a function has inputs and output and an easy way to create `.m` function file is to right click in the Directory panel and select new m-file. Name and then open it. You'll find everything already typed for you to write a function. The `input_args` and `output_args` could be used as is and you could use variable argument lists or you can just add the arguments you'd like to the list of input or output arguments. Then write what processing you would like to do inside. REMEMBER Matlab is case-sensitive.

Indexing vs. For loops – Everyone I assume has some programming experience, be it in C, C++, Java. In any of those languages you need to write for loops to operate on arrays.

Matlab has been designed with Math in mind and a for loop isn't always the fastest way to do operations on arrays or matrices. Matlab has been optimized to use indices instead. REMEMBER not need to declared variables ahead of time.

Plotting – is very straight-forward with the plot command. And you can go through the help for all of the details on customizing plots.

Writing, displaying and opening images

Use demo .m file

imread

imwrite

imshow

USE a lossless file format.

Debugging

First you need to open the editor. This is easily done by double clicking the file. Once open you can set breakpoints with the breakpoint icon on the toolbar. Now we can rerun the script and stop at the breakpoint and step through the code. To clear all breakpoints click the tool next to it. To step through the code, use F10, F11 to step in, F5 to run and Shift F11 to step out.

Use demo .m file image script and functions