EW309 Computer Vision

Estimating Scale

Goal

The units of a digital image are natively in pixels. To use information from a camera in the real world, we need to estimate a relationship between pixels and some linear unit (e.g. inches or millimeters).

Observation

What happens in your video when you move the target closer to the camera? Similarly, what happens when you move the target farther from the camera? Even though the target is the same size, it appears larger when closer; and smaller when farther away. This means the relationship between pixels and linear units is dependent on the distance between the target and camera!

For an "ideal" (pinhole) camera, this relationship can be represented mathematically given a distance between the camera and the point it is looking at, defined as s:

$$x_{pixels} = a\frac{x_{linearUnits}}{s} + b$$

$$y_{pixels} = c\frac{y_{linearUnits}}{s} + d$$

In these equations, x_{pixels} and y_{pixels} represent the pixel coordinate relative to the upper left of the picture (Fig. 1) and $x_{linearUnits}$ and $y_{linearUnits}$ represents the x and y coordinates in the real world. Note that using a common unit like inches for $x_{linearUnits}$, $y_{linearUnits}$, and s is advisable.

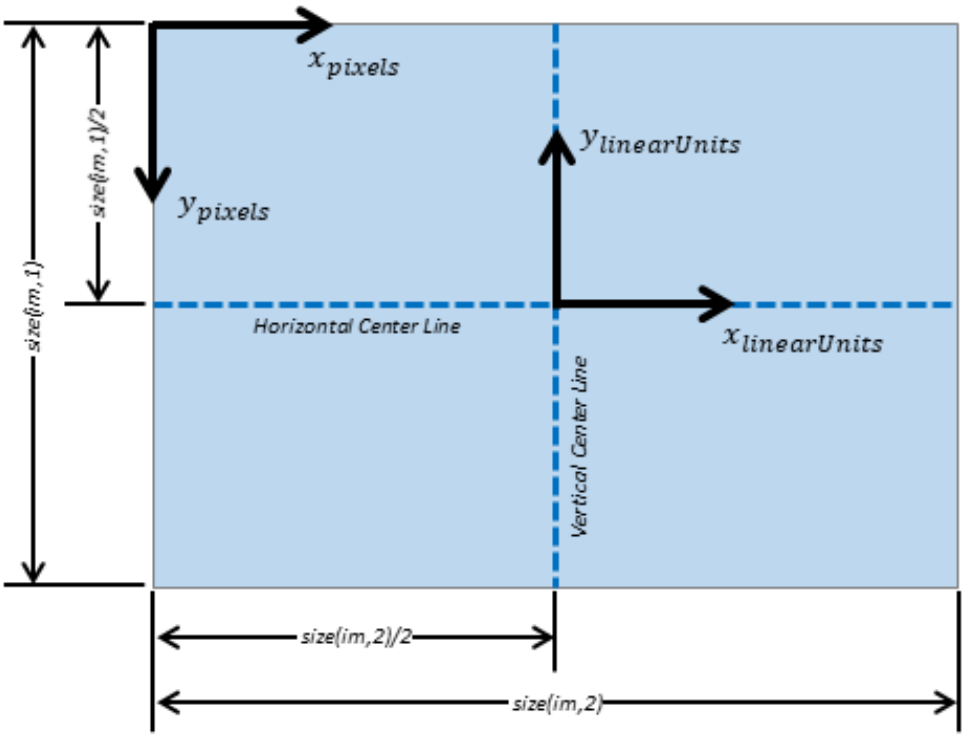

Figure 1: Recommended coordinate frame assignments highlighting the native image frame and image dimensions in MATLAB syntax.

Referencing Data

For our application, we will be using camera mounted to the rail of our weapon as a targeting mechanism. This means we expect our targets to be near the center of the image prior to firing. With this in mind, using the center of the image as a reference for measuring linear units seems to make sense (Fig 1).

Collecting Data

Create a uniform grid on the board (a crosshair with large tick marks labelled with a ruler or tape measure will work), center your camera on the crosshair, and take images at a series of known distances. Be sure to fixture your camera (or hold it very still) when taking the images; and write down the distance associated with each image.

Processing Data

Using the tools from PART 1 – PART 3 and the MATLAB Examples, measure the number of pixels to your tick-marks relative to the center of the image and create a table of values for the x and y directions for each distance. Using this table, investigate the scaling in x and y for each of your distances.

Packaging Your Results

Once you have defined the relationship between pixels and linear units, package your equations in a MATLAB function for later use. The inputs to this function should be s, x_{pixels}, and y_{pixels}; and the outputs to this function should be $x_{linearUnits}$ and $y_{linearUnits}$. Be sure to use a descriptive function name, and document your function! As an example, *if all of your linear units are in inches*, you may want to define your function as follows:

```
function [x_inches, y_inches] = pixelsToInches(s_inches, x_pixels, y_pixels)
```

Documenting Your Function

Take the time to write help documentation for your function. This will be a long semester, and it is easy to forget how to use even code that you have written. The commonly used format for MATLAB help documentation is as follows:

```
function [out1, out2, …] = funcName(in1, in2, …)
%FUNCNAME A brief description of what the function does
%    [out1, …] = FUNCNAME(in1, …) an in depth description of how the
%    function works given the specified input/output combination.
%    Any additional notes/comments/details about the aforementioned function
%    call.
%
%    [out1, …] = FUNCNAME(in1, in2, …) an in depth description of how the
%    function works given the specified input/output combination.
%    …
%
%    See also RELATEDFUNC1, FELATEDFUNC2, …
%
%    Contributor(s) names & date initially created
```