Johnny Wang
October 5, 2015

Homographies, Warp Functions & Armadillo

1a). Affine warp result:

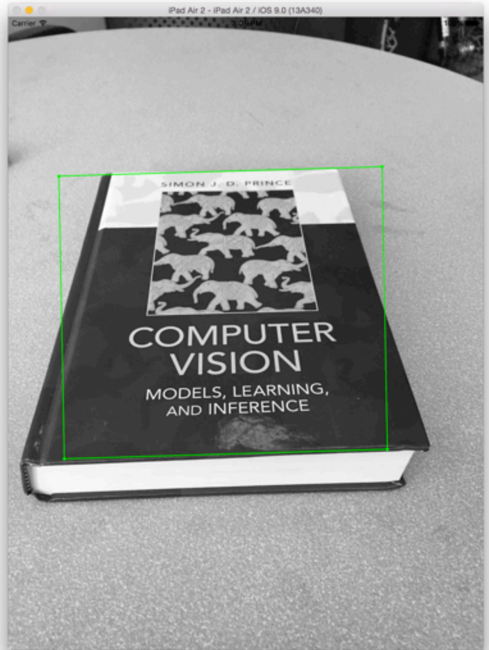


Figure 1: Affine warp result.

1b). Homography result:

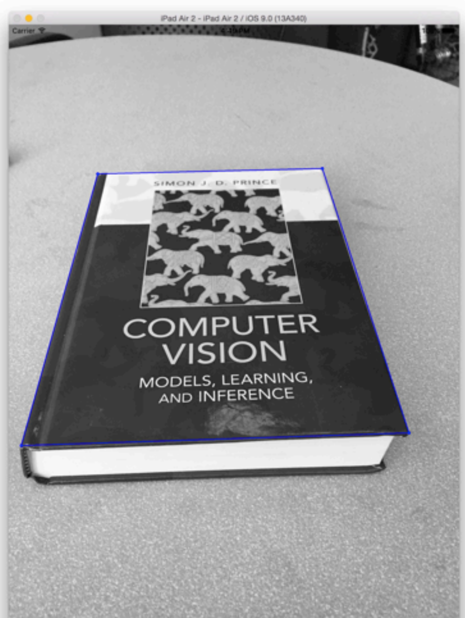


Figure 2: Homography result.

Johnny Wang
October 5, 2015

1c). In Homography, lines that are parallel in the object are not constrained to remain parallel once the projective transformation is applied. This means that it can map any four points in the plane to any four points within the image. To gain this flexibility, we must use 4 points (8 unknowns) and this equation:

$$\begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & y_1 u_1 & y_1 v_1 & y_1 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1 u_1 & -x_1 v_1 & -x_1 \\ & & & & \vdots & & & & \\ 0 & 0 & 0 & -u_I & -v_I & -1 & y_I u_I & y_I v_I & y_I \\ u_I & v_I & 1 & 0 & 0 & 0 & -x_I u_I & -x_I v_I & -x_I \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{12} \\ \phi_{13} \\ \phi_{21} \\ \phi_{22} \\ \phi_{23} \\ \phi_{31} \\ \phi_{32} \\ \phi_{33} \end{bmatrix} = 0$$

Affine warp is a good approximation to homography when the depth does not vary a lot within the planar object. If parallel lines of the object remains somewhat parallel in the image, then affine transformation can represent it. We see from the equation that the z-axis (depth) is essentially "zeroed" out, so we cannot handle changes in z.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} & 0 & \tau_x \\ \phi_{21} & \phi_{22} & 0 & \tau_y \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} & \tau_x \\ \phi_{21} & \phi_{22} & \tau_y \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

2a).

$$\left(\begin{array}{ccc|c} -0.9994 & -0.0340 & 0.0084 & 5.8581 \\ 0.0293 & -0.6782 & 0.7343 & 2.0000 \\ -0.0193 & 0.7341 & 0.6788 & -45.3727 \end{array}\right) = \Omega \mid \tau$$

2b).

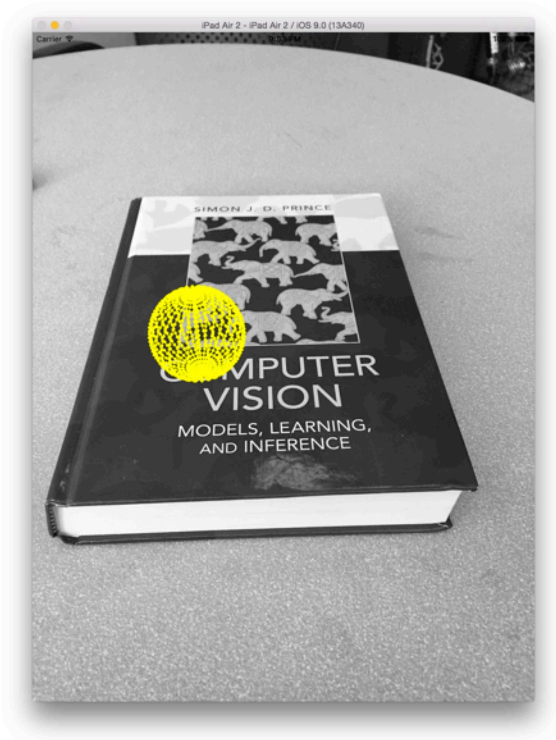
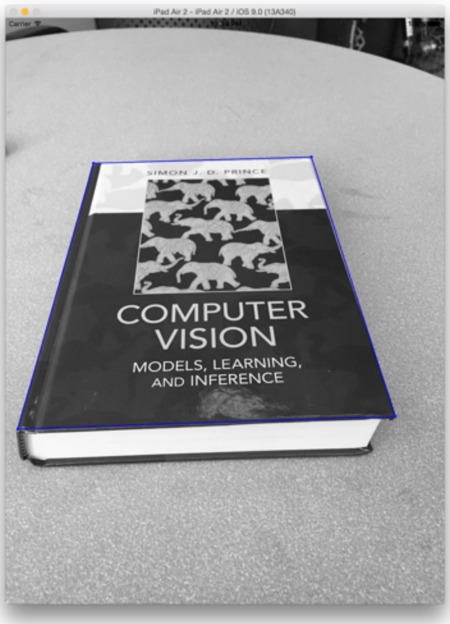


Figure 3: Projected tennis ball result.

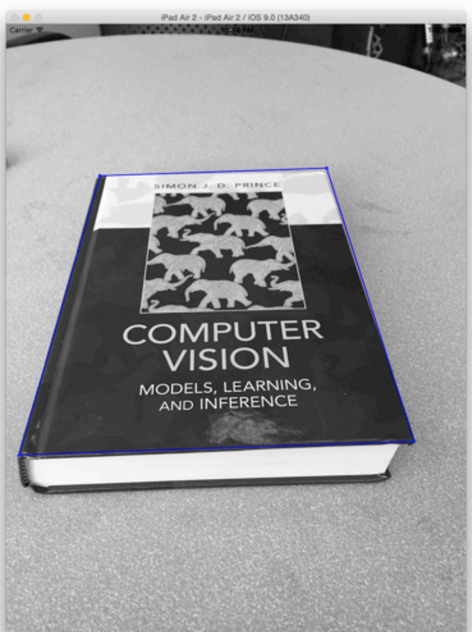
2c). myfit_homography is minimizing the algebraic error.
Geometric error is the error between the image distance of a projected point and the actual measured point. This is usually a nonlinear optimization problem that takes extra computing power to find the optimal solution. Algebraic error is the error between the image point and the reprojected point. The homography we found is a good initial linear estimate.

I used the randn() function to generate a 2x4 matrix (the same size as the 2d X matrix) and added this noise to the X matrix. I did not see any changes with this initial implementation so I scaled the noise matrix by factors of 5, 10, and 20 (see Figures 4-6).



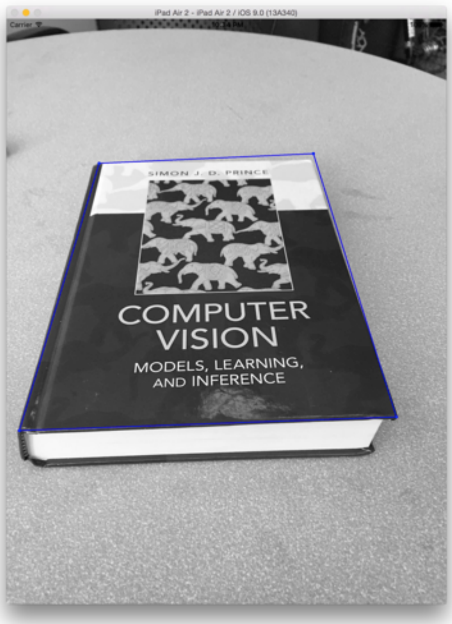

```
X:
   4.8300e+02   1.7040e+03   2.1750e+03   6.7000e+01
   8.1000e+02   7.8100e+02   2.2170e+03   2.2860e+03
scale: 5
noise:
   8.0080    0.8738   -1.5101    0.9449
  -1.2955   -7.4948    0.5963    2.2909
X:
   4.9101e+02   1.7049e+03   2.1735e+03   6.7945e+01
   8.0870e+02   7.7351e+02   2.2176e+03   2.2883e+03
T
   -0.9994   -0.0300    0.0167    5.8581
    0.0327   -0.6771    0.7352    2.0000
   -0.0107    0.7353    0.6777  -45.9998
```

Figure 4: 5x the noise matrix added to X with before and after values (right).




```
X:
   4.8300e+02   1.7040e+03   2.1750e+03   6.7000e+01
   8.1000e+02   7.8100e+02   2.2170e+03   2.2860e+03
scale: 10
noise:
   16.0159    1.7477   -3.0202    1.8898
   -2.5909  -14.9896    1.1926    4.5818
X:
   4.9902e+02   1.7057e+03   2.1720e+03   6.8890e+01
   8.0741e+02   7.6601e+02   2.2182e+03   2.2906e+03
T
   -0.9993   -0.0260    0.0251    5.8581
    0.0361   -0.6759    0.7361    2.0000
   -0.0021    0.7365    0.6764  -52.6865
```

Figure 5: 10x the noise matrix added to X with before and after values (right).




```
X:
   4.8300e+02   1.7040e+03   2.1750e+03   6.7000e+01
   8.1000e+02   7.8100e+02   2.2170e+03   2.2860e+03
scale: 20
noise:
   32.0318    3.4954   -6.0405    3.7797
   -5.1819  -29.9792    2.3853    9.1636
X:
   5.1503e+02   1.7075e+03   2.1690e+03   7.0780e+01
   8.0482e+02   7.5102e+02   2.2194e+03   2.2952e+03
T
   -0.9990   -0.0180    0.0420    5.8581
    0.0432   -0.6736    0.7378    2.0000
    0.0151    0.7388    0.6737  -42.7176
```

Figure 6: 20x the noise matrix added to X with before and after values (right).

As can be seen in Figure 4, 5x the noise already shifts the upper 2 corners of the homography a little bit. With 20x the noise, we can definitely see the shifted upper corners.

This noise mainly affects myfit_extrinsic function by shifting its z translation around—as can be see by the T matrices in Figures 4-6.

3a).

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \overset{3\times 3}{\Lambda} \begin{bmatrix} w_{11} & w_{12} & w_{13} & \tau_x \\ w_{21} & w_{22} & w_{23} & \tau_y \\ w_{31} & w_{32} & w_{33} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

multiply both sides by Λ^{-1}

$$\lambda' \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \tau_x \\ w_{21} & w_{22} & w_{23} & \tau_y \\ w_{31} & w_{32} & w_{33} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

$$x = \frac{w_{11}u + w_{12}v + w_{13}w + \tau_x}{w_{31}u + w_{32}v + w_{33}w + \tau_z} \qquad y = \frac{w_{21}u + w_{22}v + w_{23}w + \tau_y}{w_{31}u + w_{32}v + w_{33}w + \tau_z}$$

$$x(w_{31}u + w_{32}v + w_{33}w + \tau_z) = w_{11}u + w_{12}v + w_{13}w + \tau_x$$

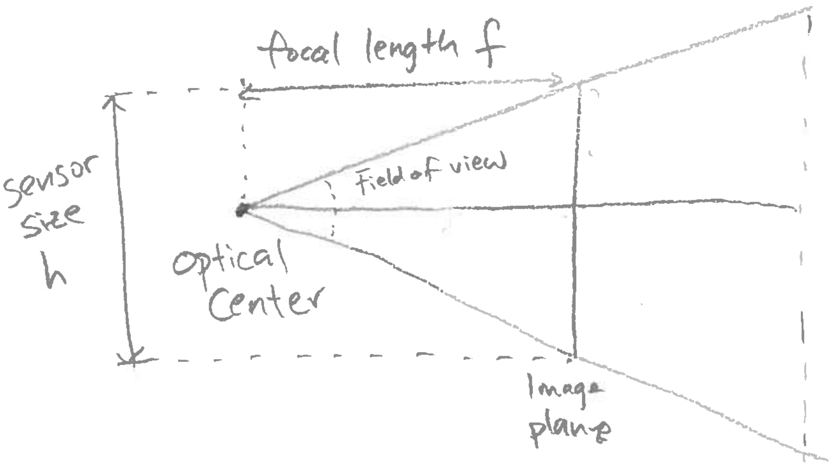
$$y(w_{31}u + w_{32}v + w_{33}w + \tau_z) = w_{21}u + w_{22}v + w_{23}w + \tau_y$$

$$x(w_{31}u + w_{32}v + w_{33}w + \tau_z) - w_{11}u - w_{12}v - w_{13}w - \tau_x = 0$$

$$y(w_{31}u + w_{32}v + w_{33}w + \tau_z) - w_{21}u - w_{22}v - w_{23}w - \tau_y = 0$$

$$\begin{bmatrix} -u_1 & -v_1 & -w_1 & -1 & 0 & 0 & 0 & 0 & x_1u_1 & x_1v_1 & x_1w_1 & x_1 \\ 0 & 0 & 0 & 0 & -u_1 & -v_1 & -w_1 & -1 & y_1u_1 & y_1v_1 & y_1w_1 & y_1 \\ & & & & & \vdots & & & & & & \\ -u_n & -v_n & -w_n & -1 & 0 & 0 & 0 & 0 & x_nu_n & x_nv_n & x_nw_n & x_n \\ 0 & 0 & 0 & 0 & -u_n & -v_n & -w_n & -1 & y_nu_n & y_nv_n & y_nw_n & y_n \end{bmatrix} \begin{bmatrix} w_{11} \\ w_{12} \\ w_{13} \\ \tau_x \\ w_{21} \\ w_{22} \\ w_{23} \\ \tau_y \\ w_{31} \\ w_{32} \\ w_{33} \\ \tau_z \end{bmatrix} = 0$$

3b). Yes the max field of view for the image sensor can be determined.



Assuming focal length $\phi_x = \phi_y$

$$\text{Maximum field of view in degrees} = 2 \cdot \tan^{-1}\left(\frac{h}{2f}\right)$$