CSC420: Introduction to Image Understanding
Assignment 3 Solutions

Jake Snell
jsnell@cs.toronto.edu

October 25, 2017

Problem 1

The shoe and the bill lie on the same planar surface and so we can use a homography to transform between a point $\mathbf{p}$ on the image plane and a point $\mathbf{P}$ in the world plane. Expressing $\mathbf{P}$ and $\mathbf{p}$ in homogeneous coordinates with homography matrix $\mathbf{H}$, we have the following:

$$\mathbf{P} = \mathbf{H}\mathbf{p} \tag{1}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2}$$

We know that a five-dollar bill is $d_1 = 15.24$ cm long and $d_2 = 6.985$ cm tall[1]. So let's express the bottom-left, bottom-right, top-left, and top-right world coordinates of the bill, respectively, as follows:

$$\mathbf{P}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} d_1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{P}_3 = \begin{bmatrix} 0 \\ d_2 \\ 1 \end{bmatrix}, \mathbf{P}_4 = \begin{bmatrix} d_1 \\ d_2 \\ 1 \end{bmatrix} \tag{3}$$

We can extract the corresponding points in the image plane p_1, p_2, p_3, and p_4 directly from the image. Then we can solve for $\mathbf{H}$ using the technique described in the appendix of [Criminisi, 2002]. We solve $\mathbf{A}\mathbf{h} = 0$ for $\|\mathbf{h}\| = 1$, where $\mathbf{A}$ and $\mathbf{h}$ are the following:

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -y_1X_1 & -X_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1Y_1 & -y_1Y_1 & -Y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -y_4X_4 & -X_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4Y_4 & -y_4Y_4 & -Y_4 \end{bmatrix} \tag{4}$$

$$\mathbf{h} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}^\top \tag{5}$$

[1] http://www.bankofcanada.ca/banknotes/bank-note-series/polymer/5-polymer-note/

This can be solved by setting $\mathbf{h}$ to be the null vector of $\mathbf{A}$. With $\mathbf{H}$ now in hand, we can estimate the world coordinate of any image coordinate by pre-multiplying it by $\mathbf{H}$. In the end, our estimated shoe length is 27.16 cm and width is 7.86 cm.

Problem 2

Part (a)

Greedy and RANSAC matchings can be found in Figures 1 and 2.

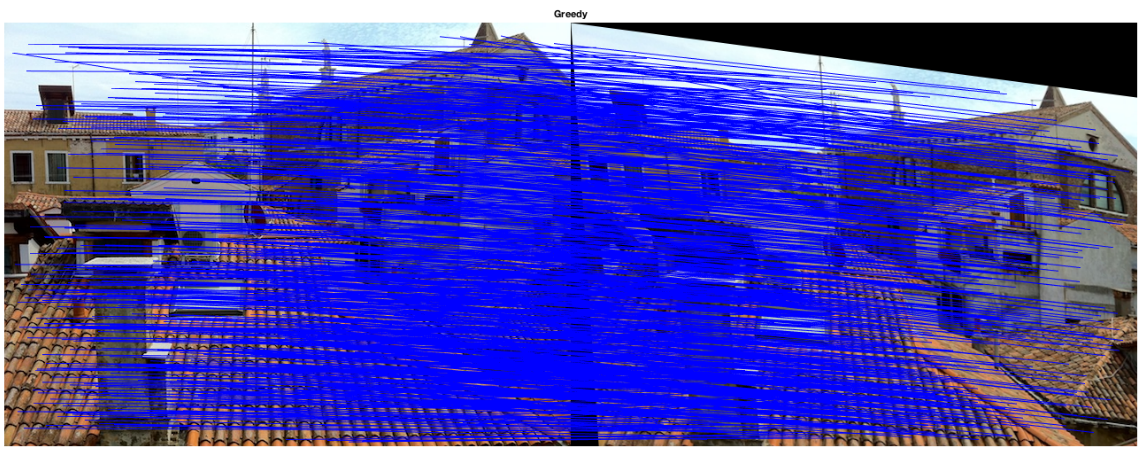

Figure 1: Greedy Matching

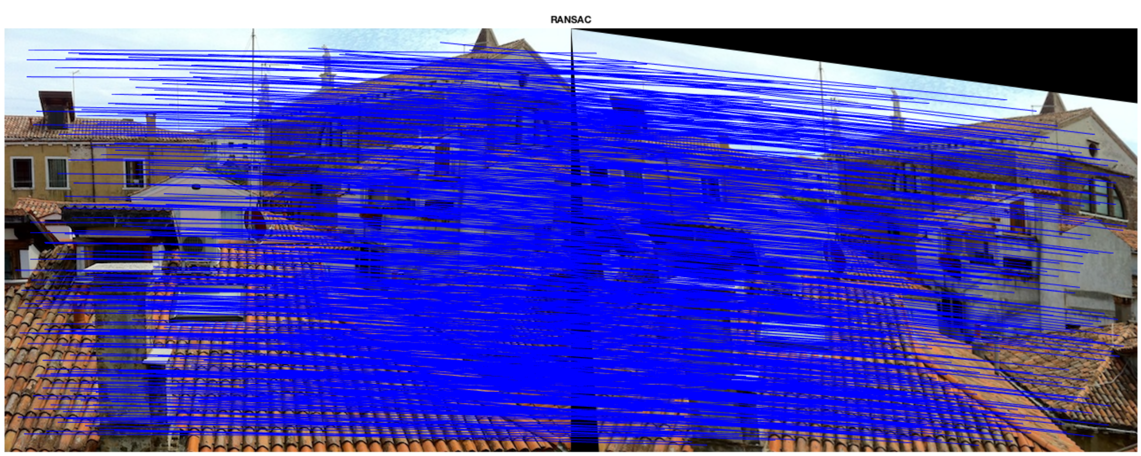

Figure 2: RANSAC Matching

```
prob2a
1021 matches found
```

```
iter 1: 507 inliers (49.66%)
iter 2: 966 inliers (94.61%)
iter 3: 970 inliers (95.00%)
iter 4: 971 inliers (95.10%)
iter 5: 14 inliers (1.37%)
iter 6: 925 inliers (90.60%)
iter 7: 970 inliers (95.00%)
iter 8: 969 inliers (94.91%)
iter 9: 972 inliers (95.20%)
iter 10: 970 inliers (95.00%)
iter 11: 971 inliers (95.10%)
iter 12: 763 inliers (74.73%)
iter 13: 423 inliers (41.43%)
iter 14: 967 inliers (94.71%)
iter 15: 971 inliers (95.10%)
iter 16: 817 inliers (80.02%)
iter 17: 970 inliers (95.00%)
iter 18: 83 inliers (8.13%)
iter 19: 969 inliers (94.91%)
iter 20: 969 inliers (94.91%)
iter 21: 971 inliers (95.10%)
iter 22: 969 inliers (94.91%)
iter 23: 969 inliers (94.91%)
iter 24: 970 inliers (95.00%)
iter 25: 973 inliers (95.30%)
iter 26: 14 inliers (1.37%)
iter 27: 31 inliers (3.04%)
iter 28: 970 inliers (95.00%)
iter 29: 968 inliers (94.81%)
iter 30: 939 inliers (91.97%)
iter 31: 969 inliers (94.91%)
iter 32: 969 inliers (94.91%)
iter 33: 734 inliers (71.89%)
iter 34: 28 inliers (2.74%)
iter 35: 967 inliers (94.71%)
iter 36: 15 inliers (1.47%)
iter 37: 971 inliers (95.10%)
iter 38: 970 inliers (95.00%)
iter 39: 968 inliers (94.81%)
iter 40: 518 inliers (50.73%)
iter 41: 958 inliers (93.83%)
iter 42: 969 inliers (94.91%)
iter 43: 969 inliers (94.91%)
iter 44: 32 inliers (3.13%)
iter 45: 19 inliers (1.86%)
```

```
iter 46: 770 inliers (75.42%)
iter 47: 966 inliers (94.61%)
iter 48: 971 inliers (95.10%)
iter 49: 970 inliers (95.00%)
iter 50: 937 inliers (91.77%)
iter 51: 969 inliers (94.91%)
iter 52: 457 inliers (44.76%)
iter 53: 972 inliers (95.20%)
iter 54: 969 inliers (94.91%)
iter 55: 972 inliers (95.20%)
iter 56: 5 inliers (0.49%)
iter 57: 961 inliers (94.12%)
iter 58: 969 inliers (94.91%)
iter 59: 970 inliers (95.00%)
iter 60: 893 inliers (87.46%)
iter 61: 919 inliers (90.01%)
iter 62: 954 inliers (93.44%)
iter 63: 6 inliers (0.59%)
iter 64: 969 inliers (94.91%)
iter 65: 19 inliers (1.86%)
iter 66: 882 inliers (86.39%)
iter 67: 954 inliers (93.44%)
iter 68: 14 inliers (1.37%)
iter 69: 12 inliers (1.18%)
iter 70: 961 inliers (94.12%)
iter 71: 970 inliers (95.00%)
iter 72: 123 inliers (12.05%)
iter 73: 122 inliers (11.95%)
iter 74: 967 inliers (94.71%)
iter 75: 899 inliers (88.05%)
iter 76: 969 inliers (94.91%)
iter 77: 965 inliers (94.52%)
iter 78: 971 inliers (95.10%)
iter 79: 872 inliers (85.41%)
iter 80: 819 inliers (80.22%)
iter 81: 971 inliers (95.10%)
iter 82: 566 inliers (55.44%)
iter 83: 972 inliers (95.20%)
iter 84: 826 inliers (80.90%)
iter 85: 972 inliers (95.20%)
iter 86: 677 inliers (66.31%)
iter 87: 752 inliers (73.65%)
iter 88: 15 inliers (1.47%)
iter 89: 498 inliers (48.78%)
iter 90: 454 inliers (44.47%)
```

```
iter 91: 970 inliers (95.00%)
iter 92: 131 inliers (12.83%)
iter 93: 86 inliers (8.42%)
iter 94: 970 inliers (95.00%)
iter 95: 972 inliers (95.20%)
iter 96: 879 inliers (86.09%)
iter 97: 969 inliers (94.91%)
iter 98: 19 inliers (1.86%)
iter 99: 18 inliers (1.76%)
iter 100: 966 inliers (94.61%)
best fit: 973 inliers (95.30%)
predicted A:

A =

    1.0000    0.0653    0.3718
    0.1316    1.0000    0.2900

ground truth A:

Agt =

    1.0000    0.0655         0
    0.1317    1.0000         0
```

Part (b)

The reconstructed image can be found in Figure 3. The boundaries are permuted because there are no shared keypoints between the source and target in the boundaries. Thus they will not enter into either our RANSAC procedure or the mean residual SSD evaluation.

```
prob2b
iter 01: ssd = 50645.443561
iter 02: ssd = 168960.765256
iter 03: ssd = 43341.673180
iter 04: ssd = 191435.078331
iter 05: ssd = 156381.868566
iter 06: ssd = 171596.763752
iter 07: ssd = 26790.576800
iter 08: ssd = 116372.616659
iter 09: ssd = 111828.275772
iter 10: ssd = 43902.558478
iter 11: ssd = 113318.556949
iter 12: ssd = 250064.701823
iter 13: ssd = 181237.467202
```

Figure 3: Reconstructed Image

```
iter 14: ssd = 61399.429897
iter 15: ssd = 96040.776773
iter 16: ssd = 42152.337681
iter 17: ssd = 43791.868773
iter 18: ssd = 158163.914009
iter 19: ssd = 271645.684797
iter 20: ssd = 70897.212568
iter 21: ssd = 146241.913398
iter 22: ssd = 79004.554345
iter 23: ssd = 86684.725812
iter 24: ssd = 195017.273926
iter 25: ssd = 52890.660926
iter 26: ssd = 68159.665685
iter 27: ssd = 177483.821008
iter 28: ssd = 59126.620154
iter 29: ssd = 80507.872053
iter 30: ssd = 51923.131455
iter 31: ssd = 360745.576507
```

```
iter 32: ssd = 111929.711680
iter 33: ssd = 90286.817995
iter 34: ssd = 119055.970599
iter 35: ssd = 118881.290814
iter 36: ssd = 120795.592978
iter 37: ssd = 45867.865296
iter 38: ssd = 70592.503920
iter 39: ssd = 351107.418768
iter 40: ssd = 235113.543155
iter 41: ssd = 84461.098650
iter 42: ssd = 294202.613619
iter 43: ssd = 367212.946061
iter 44: ssd = 60409.426579
iter 45: ssd = 92911.238950
iter 46: ssd = 159776.528505
iter 47: ssd = 57292.025953
iter 48: ssd = 79902.848050
iter 49: ssd = 188112.868497
iter 50: ssd = 99298.132651
iter 51: ssd = 325726.601465
iter 52: ssd = 81243.878100
iter 53: ssd = 178591.811927
iter 54: ssd = 157285.047968
iter 55: ssd = 74372.187976
iter 56: ssd = 215840.101759
iter 57: ssd = 64295.003817
iter 58: ssd = 178915.548894
iter 59: ssd = 114304.060832
iter 60: ssd = 77851.588039
iter 61: ssd = 299298.678313
iter 62: ssd = 120366.828135
iter 63: ssd = 66884.861988
iter 64: ssd = 158695.391520
iter 65: ssd = 170090.318625
iter 66: ssd = 75070.281054
iter 67: ssd = 51776.623744
iter 68: ssd = 95955.039140
iter 69: ssd = 218008.568426
iter 70: ssd = 110510.602840
iter 71: ssd = 107405.844454
iter 72: ssd = 104839.380105
iter 73: ssd = 101526.963258
iter 74: ssd = 118979.965711
iter 75: ssd = 133873.581454
iter 76: ssd = 77601.698198
```

```
iter 77: ssd = 70363.699652
iter 78: ssd = 235404.232251
iter 79: ssd = 65157.441586
iter 80: ssd = 304239.262917
iter 81: ssd = 39449.711764
iter 82: ssd = 119015.978452
iter 83: ssd = 145658.224191
iter 84: ssd = 245551.733790
iter 85: ssd = 313905.249065
iter 86: ssd = 59436.521578
iter 87: ssd = 244452.213185
iter 88: ssd = 259929.436616
iter 89: ssd = 213722.579280
iter 90: ssd = 77058.683640
iter 91: ssd = 62245.220947
iter 92: ssd = 300307.690761
iter 93: ssd = 142872.725114
iter 94: ssd = 149278.139161
iter 95: ssd = 79693.919078
iter 96: ssd = 96729.115346
iter 97: ssd = 340534.993240
iter 98: ssd = 80997.809577
iter 99: ssd = 256692.712187
iter 100: ssd = 185127.086008
best ssd = 26790.576800
```

Problem 3

Part (a)

We will need to collect the image coordinates of three points along a single rail. Suppose that we have three world points along a rail, each 1 tie apart:

$$V_1 = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, V_2 = \begin{bmatrix} X + D_x \\ Y + D_y \\ Z + D_z \\ 1 \end{bmatrix}, V_3 = \begin{bmatrix} X + 2D_x \\ Y + 2D_y \\ Z + 2D_z \\ 1 \end{bmatrix} \tag{6}$$

The corresponding image coordinates v_1, v_2, and v_3 are:

$$v_1 = \begin{bmatrix} w_1 x_1 \\ w_1 y_1 \\ w_1 \end{bmatrix} = HV_1, v_2 = \begin{bmatrix} w_2 x_2 \\ w_2 y_2 \\ w_2 \end{bmatrix} = HV_2, v_3 = \begin{bmatrix} w_3 x_3 \\ w_3 y_3 \\ w_3 \end{bmatrix} = HV_3, \tag{7}$$

where $H = K \left[\begin{array}{c|c} R & t \end{array} \right]$. Subtracting $v_1 = HV_1$ from both $v_2 = Hv_2$ and $v_3 = Hv_3$, we have:

$$\begin{bmatrix} w_2x_2 - w_1x_1 \\ w_2y_2 - w_1y_1 \\ w_2 - w_1 \\ w_3x_3 - w_1x_1 \\ w_3y_3 - w_1y_1 \\ w_3 - w_1 \end{bmatrix} = \begin{bmatrix} H & \\ & H \end{bmatrix} \begin{bmatrix} D_x \\ D_y \\ D_z \\ 0 \\ 2D_x \\ 2D_y \\ 2D_z \\ 0 \end{bmatrix}, \tag{8}$$

a linear system of six equations with six unknowns: w_1, w_2, w_3, D_x, D_y, and D_z. After solving for D_x, D_y and D_z, the distance between adjacent railway ties is $\sqrt{D_x^2 + D_y^2 + D_z^2}$.

Part (b)

We can use the relationship from Derek Hoiem's notes on single-view geometry[2].

$$\frac{Y_0}{Y_c} = \frac{v_t - v_b}{v_h - v_b}, \tag{9}$$

where Y_0 is the height of the man, Y_c is the camera height, v_t is the top of the man in the image, v_b is the bottom of the man in the image, and v_h is the image height of the horizon. By measuring the image, we find $v_b = 65$, $v_h = 240$, and $v_t = 359$. Substituting into (9), we can estimate the height of the man to be approximately:

$$Y_0 = Y_c \frac{v_t - v_b}{v_h - v_b} = (95 \text{ cm}) \frac{359 \text{ px} - 65 \text{ px}}{240 \text{ px} - 65 \text{ px}} = 159.6 \text{ cm}. \tag{10}$$

Problem 4

Part (a)

$$K = \begin{bmatrix} 721.5 & 0 & 609.6 \\ 0 & 721.5 & 172.9 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Part (b)

$$Y = -1.7 \text{ m} \tag{12}$$

[2] `https://courses.engr.illinois.edu/cs543/sp2011/materials/3dscene_book_svg.pdf`

Part (c)

Since the point is on the ground, $Y = -1.7$ m From similar triangles, we know:

$$\frac{Z}{1.7 \text{ m}} = \frac{f}{p_y - y} \Rightarrow Z = \left(\frac{721.5}{172.9 - y}\right)(1.7 \text{ m}) \tag{13}$$

Additionally,

$$\frac{X}{Z} = \frac{x - p_x}{f} \Rightarrow X = \left(\frac{x - 609.6}{721.5}\right) Z \tag{14}$$

Thus the 3D location is:

$$(X, Y, Z) = \left(\left(\frac{x - 609.6}{172.9 - y}\right)(1.7 \text{ m}), -1.7 \text{ m}, \left(\frac{721.5}{172.9 - y}\right)(1.7 \text{ m})\right) \tag{15}$$

Extra Credit

Part (a)

Since we are working in the camera coordinate system, we have:

$$\begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} = \begin{bmatrix} fx & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{16}$$

Solving for X and Y, we get:

$$X = \frac{Z}{f_x}(x - p_x) \tag{17}$$

$$Y = \frac{Z}{f_y}(y - p_y) \tag{18}$$

The resulting point cloud can be found in Figure 4.

Part (b)

```
extra_credit
object 1: mu_x = 0.487630, mu_y = -0.232279, mu_z = 1.388624
object 2: mu_x = 0.864993, mu_y = -0.194680, mu_z = 2.745325
object 3: mu_x = 0.076532, mu_y = 0.331478, mu_z = 3.752931
object 4: mu_x = -0.542094, mu_y = -0.424440, mu_z = 1.732951

object 3 is farthest from the camera with distance 3.768319
object 3 is highest above the floor with height 0.331478
```

References

Antonio Criminisi. Single-view metrology: Algorithms and applications. *Pattern Recognition*, pages 224–239, 2002. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/criminisi_dagm2002.pdf.

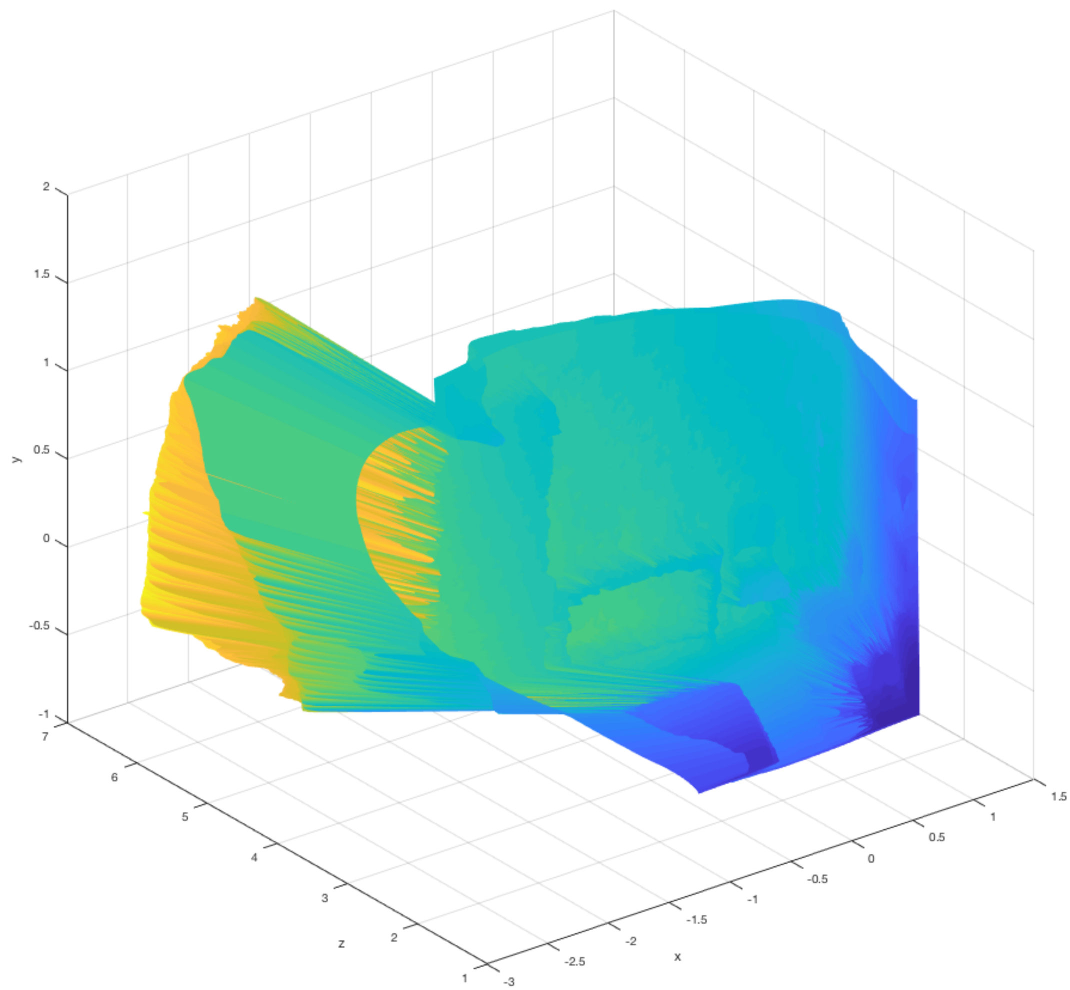

Figure 4: Point cloud of bedroom image.