ECE 661 Fall 2018
Homework #2

Ye Shi
shi349@purdue.edu

September 5, 2018

Contents

1 Methodology

1.1 Least Square Estimation for Homography

Assuming $x' = Hx$ that x and x' are expressed using HC:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \text{ and } x' = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix}$$

with

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}.$$

We end up with

$$\begin{cases} x'_1 = h_{11}x_1 + h_{12}x_2 + h_{13}x_3 \\ x'_2 = h_{21}x_1 + h_{22}x_2 + h_{23}x_3 \\ x'_3 = h_{31}x_1 + h_{32}x_2 + h_{33}x_3 \end{cases}$$

Since x' is in HC, the 2D physical coordinates (x', y') have

$$\begin{cases} x' = \dfrac{x'_1}{x'_3} = \dfrac{h_{11}x_1 + h_{12}x_2 + h_{13}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \\ y' = \dfrac{x'_2}{x'_3} = \dfrac{h_{21}x_1 + h_{22}x_2 + h_{23}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \end{cases}.$$

Since there is no ideal points in real image, $x' \neq 0$, we have

$$\begin{cases} h_{11}x_1x'_3 + h_{12}x_2x'_3 + h_{13}x_3x'_3 - h_{31}x_1x'_1 - h_{32}x_2x'_1 - h_{33}x_3x'_1 = 0 \\ h_{21}x_1x'_3 + h_{22}x_2x'_3 + h_{23}x_3x'_3 - h_{31}x_1x'_2 - h_{32}x_2x'_2 - h_{33}x_3x'_2 = 0 \end{cases}.$$

While we select the points on real images, $x_3 = 1$ and $x'_3 = 1$. And H in HC, we can assign $h_33 = 1$. We then have

$$\begin{cases} h_{11}x_1 + h_{12}x_2 + h_{13} - h_{31}x_1x'_1 - h_{32}x_2x'_1 = x'_1 \\ h_{21}x_1 + h_{22}x_2 + h_{23} - h_{31}x_1x'_2 - h_{32}x_2x'_2 = x'_2 \end{cases}.$$

We then formulate it into a matrix expression

$$Ax = b, \tag{1}$$

where

$$A^{2i\times 8} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & 1 & 0 & 0 & 0 & -x_1^{(1)}x_1'^{(1)} & -x_2^{(1)}x_1'^{(1)} \\ 0 & 0 & 0 & x_1^{(1)} & x_2^{(1)} & 1 & -x_1^{(1)}x_2'^{(1)} & -x_2^{(1)}x_2'^{(1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_1^{(i)} & x_2^{(i)} & 1 & 0 & 0 & 0 & -x_1^{(i)}x_1'^{(i)} & -x_2^{(i)}x_1'^{(i)} \\ 0 & 0 & 0 & x_1^{(i)} & x_2^{(i)} & 1 & -x_1^{(i)}x_2'^{(i)} & -x_2^{(i)}x_2'^{(i)} \end{bmatrix}$$

$$x = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} \text{ and } b^{2i\times 1} = \begin{bmatrix} x_1'^{(1)} \\ x_2'^{(2)} \\ \vdots \\ \vdots \\ x_1'^{(i)} \\ x_2'^{(i)} \end{bmatrix}, \; i \text{ is the number of corresponding-points pairs.}$$

Then we can use pseudo inverse $A^\dagger$ of A to find the LSE of b,

$$\hat{b} = A^\dagger x. \tag{2}$$

We then reshape $\hat{b}$ to find the homography H.

1.2 Algorithm for Problem 1a and 2a

- Step 1: Pick corresponding P, Q, R, S points coordinates by GIMP in subject image, S and target image, T.
- Step 2: Finding the homography H by Formula 1 in Section 1.1.
- Step 3: Mapping the pixel coordinates from T to S by $x_S = H^{-1}x_T$. (Note, H is the homography for $x_T = Hx_S$) If the coordinates of the pixel in the size of image S, then replace the pixel value of T by the pixel value of S. The method will prevent the transformation unable to cover the pixels.
- Step 4: Output the modified image T.

1.3 Algorithm for Problem 1b and 2b

- Step 1: Pick corresponding P, Q, R, S points coordinates by GIMP in subject image, S and target image 1, T_1.
- Step 2: Finding the homography H_1 by Formula 1 in Section 1.1.
- Step 3: Pick corresponding P, Q, R, S points coordinates by GIMP in subject image, S and target image 2, T_2.
- Step 4: Finding the homography H_2 by Formula 1 in Section 1.1.
- Step 5: $H = H_2H_1$
- Step 6: Create an empty image E with the size as S.
- Step 7: Mapping the pixel coordinates from E to S by $x_S = H^{-1}x_E$. (Note, H is the homography for $x_E = Hx_S$) If the coordinates of the pixel in the size of image S, then replace the pixel value of E by the pixel value of S.
- Step 8: Output the modified image E.

---

2 Results

2.1 Problem 1

2.1.1 Problem 1a

(a) Jackie.jpg projected to 1.jpg (b) Jackie.jpg projected to 2.jpg (c) Jackie.jpg projected to 3.jpg

Figure 1: Results of Problem 1a

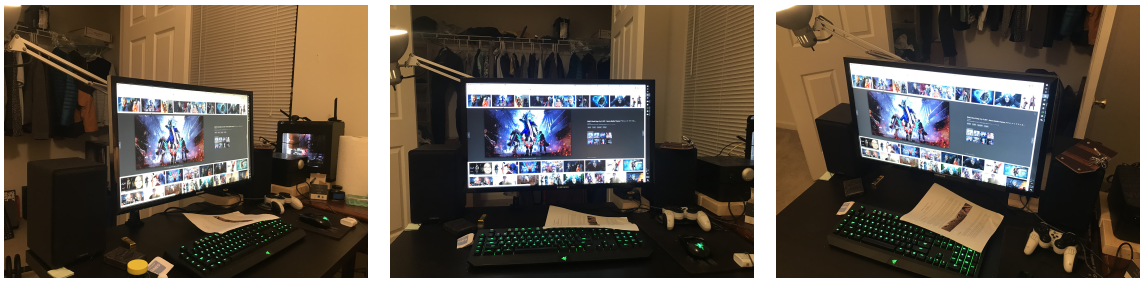
2.1.2 Problem 1b

Fig. 2a is the result of the problem 2b.

(a) 1.jpg transformed to 3.jpg (b) 3.jpg

Figure 2: Compassion between the result and 3.jpg

2.2 Problem 2



(a) Left View (b) Front View (c) Right-Up View

Figure 3: A Computer Display

Fig.3 are photos of a computer display, taken by a Iphone 7 with different viewpoints. There is a poster image in the screen. This poster is going to be replaced by Fig. 4, Chibi Maruko-chan, a famous cartoon character. Momoko Sakura, creator of the much-loved "Chibi Maruko-chan" manga and anime series, died of breast cancer on Aug. 15th, 2018. She was 53.

Figure 4: Chibi Maruko-chan

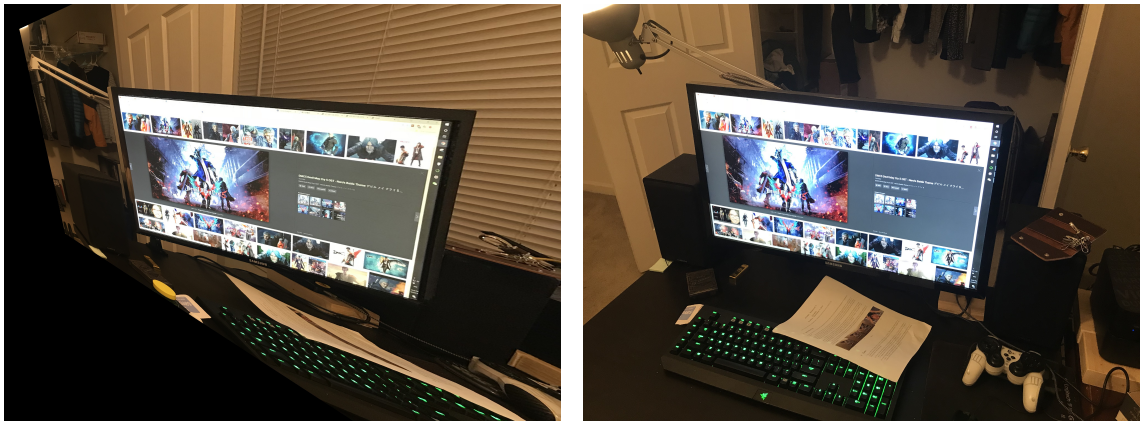
2.2.1 Problem 2a

(a) Left View (b) Front View (c) Right-Up View

Figure 5: Results of Problem 2a

2.2.2 Problem 2b

Fig. 6a is the result of the problem 2b.



(a) Fig. 3a transformed to Fig. 3c (b) Fig. 3c

Figure 6: Compassion between the result and Fig. 3c

3 Source Code

The code is written in Python 3.7 and Windows 10.

```
# -*- coding: utf-8 -*-
"""
ECE661 Homework 2
Ye Shi
shi349@purdue.edu
"""

# Load libs
import numpy as np
import cv2
import math

# Function to map image 2 to image 1 by H
def mapping(im1, im2, H):
    [h2,w2,c2] = im2.shape
    [h1,w1,c1] = im1.shape
    method = 1
    if method == 2:
        # This method may have some points not covered
        for i in range(h2):
            for j in range(w2):
                y = np.array((i,j,1))
                x = np.matmul(H,y.T)
```

```
                    x = np.floor(x/x[2])
                    x = x.astype(int)
                    if x[0] <h1 and x[0]>=0 and x[1] <w1 and x[1] >=0:
                        im1[x[0]][x[1]] = im2[i][j]

        elif method == 1:
            for i in range(h1):
                for j in range(w1):
                    x = np.array((i,j,1))
                    y = np.matmul(np.linalg.inv(H),x.T)
                    y = np.floor(y/y[2])
                    y = y.astype(int)

                    if y[0]<h2 and y[0]>=0 and y[1]<w2 and y[1]>=0:
                        im1[i][j] = im2[y[0]][y[1]]

# Function to project image 1 by H
def project(im1, H):
    [h,w,c] = im1.shape
    imNew = np.zeros((h,w,c))
    for i in range(h):
        for j in range(w):
            x = np.array((i,j,1))
            y = np.matmul(H,x.T)
            y = np.floor(y/y[2])
            y = y.astype(int)

            if y[0]<h and y[0]>=0 and y[1]<w and y[1]>=0:
                imNew[i][j] = im1[y[0]][y[1]]
    return imNew

# Function to find homography from points of image 1 to points of image 2
def findH(pst1,pst2):

    n = np.size(pst1,0)
    # check points number and dimension to be identical
    if n != np.size(pst2,0) or np.size(pst1,1) != np.size(pst2,1):
        print ("Error!␣Sizes␣don't␣match!")
        exit(1)

    n = n+1
    A = np.zeros((2*n,2*n))
    b = np.zeros((2*n,1))
    for i in range(n):
        A[2*i] = [pst2[0][i],pst2[1][i],pst2[2][i],0,0,0,(-pst2[0][i]*pst1[0][i])
            ,(-pst2[1][i]*pst1[0][i])]
        A[2*i+1] = [0,0,0,pst2[0][i],pst2[1][i],pst2[2][i],(-pst2[0][i]*pst1[1][i])
            ,(-pst2[1][i]*pst1[1][i])]
        b[2*i] = pst1[0][i]
        b[2*i+1] = pst1[1][i]

    x = np.matmul(np.linalg.pinv(A),b)
    H = np.zeros((3,3))
    H[0] = x[0:3].T
    H[1] = x[3:6].T
    H[2][0:2] = x[6:8].T
    H[2][2] = 1
    return H

# Problem 1a
# Import Images
im1a = cv2.imread(".\\PicsHw2\\1.jpg")
im1b = cv2.imread(".\\PicsHw2\\2.jpg")
im1c = cv2.imread(".\\PicsHw2\\3.jpg")
im1d = cv2.imread(".\\PicsHw2\\jackie.jpg")

# Grip points from images by order in PQRS with HC
PQRS1a = np.array([[171,1517,1],[721,2954,1],[2239,1485,1],[2051,3003,1]])
PQRS1b = np.array([[338,1326,1],[622,3012,1],[2014,1292,1],[1897,3033,1]])
PQRS1c = np.array([[732,920,1],[388,2797,1],[2094,898,1],[2232,2854,1]])
PQRS1d = np.array([[0,0,1],[0,im1d.shape[1],1],[im1d.shape[0],0,1], [im1d.shape[0],
    im1d.shape[1],1]])

```

```
94  # Find homographies
95  # 1d to 1a
96  Hd2a = findH(PQRS1a.T,PQRS1d.T)
97  # 1d to 1b
98  Hd2b = findH(PQRS1b.T,PQRS1d.T)
99  # 1d to 1a
100 Hd2c = findH(PQRS1c.T,PQRS1d.T)
101
102 # Project Images and Save
103 mapping(im1a, im1d, Hd2a)
104 cv2.imwrite('1a1.jpg',im1a)
105
106 mapping(im1b, im1d, Hd2b)
107 cv2.imwrite('1a2.jpg',im1b)
108
109 mapping(im1c, im1d, Hd2c)
110 cv2.imwrite('1a3.jpg',im1c)
111
112
113 # Problem 1b
114
115 # Import Images again to refresh
116 im1a = cv2.imread(".\\PicsHw2\\1.jpg")
117 im1b = cv2.imread(".\\PicsHw2\\2.jpg")
118 im1c = cv2.imread(".\\PicsHw2\\3.jpg")
119 im1d = cv2.imread(".\\PicsHw2\\jackie.jpg")
120
121 # Find homographies
122 # 1a to 1b
123 Ha2b = findH(PQRS1b.T,PQRS1a.T)
124 # 1b to 1c
125 Hb2c = findH(PQRS1c.T,PQRS1b.T)
126 # 1a to 1b to 1c
127 Ha2b2c = np.matmul(Hb2c,Ha2b)
128
129 # Project 1a by Ha2b2c
130 imNew = np.zeros(im1a.shape)
131 mapping(imNew,im1a, Ha2b2c)
132 cv2.imwrite('1b1.jpg',imNew)
133
134 # Problem 2a
135
136 # Import Images
137 im2a = cv2.imread("im2a.jpg")
138 im2b = cv2.imread("im2b.jpg")
139 im2c = cv2.imread("im2c.jpg")
140 im2d = cv2.imread("im2d.jpg")
141
142 # Grip points from images by order in PQRS with HC
143 PQRS2a = np.array([[1218,1442,1],[1240,2206,1],[1882,1446,1],[1724,2218,1]])
144 PQRS2b = np.array([[1156,1042,1],[1158,1996,1],[1690,1054,1],[1670,2000,1]])
145 PQRS2c = np.array([[952,1010,1],[1084,1914,1],[1438,1066,1],[1586,1916,1]])
146 PQRS2d = np.array([[0,0,1],[0,im2d.shape[1],1],[im2d.shape[0],0,1], [im2d.shape[0],
        im2d.shape[1],1]])
147
148 # Find homographies
149 # 2d to 2a
150 Hd2a2 = findH(PQRS2a.T,PQRS2d.T)
151 # 2d to 2b
152 Hd2b2 = findH(PQRS2b.T,PQRS2d.T)
153 # 2d to 2a
154 Hd2c2 = findH(PQRS2c.T,PQRS2d.T)
155
156 # Project Images and Save
157 mapping(im2a, im2d, Hd2a2)
158 cv2.imwrite('2a1.jpg',im2a)
159
160 mapping(im2b, im2d, Hd2b2)
161 cv2.imwrite('2a2.jpg',im2b)
162
163 mapping(im2c, im2d, Hd2c2)
164 cv2.imwrite('2a3.jpg',im2c)
165
```

```
166
167 # Problem 2b
168
169 # Import Images again to refresh
170 im2a = cv2.imread("im2a.jpg")
171 im2b = cv2.imread("im2b.jpg")
172 im2c = cv2.imread("im2c.jpg")
173 im2d = cv2.imread("im2d.jpg")
174
175 # Find homographies
176 # 2a to 2b
177 Ha2b2 = findH(PQRS2b.T,PQRS2a.T)
178 # 2b to 2c
179 Hb2c2 = findH(PQRS2c.T,PQRS2b.T)
180 # 2a to 2b to 2c
181 Ha2b2c2 = np.matmul(Hb2c2,Ha2b2)
182
183 # Project 2a by Ha2b2c2
184 imNew = np.zeros(im2a.shape)
185 mapping(imNew,im2a, Ha2b2c2)
186 cv2.imwrite('2b1.jpg',imNew)
```