```python
1  # import the necessary packages
2  import cv2
3  import numpy as np
4  import sys
5  import json
6  import argparse
7
8  def savePick():
9      global pick
10     data = {}
11     data["pick"] = pick
12     with open('result.json', 'w') as outfile:
13         json.dump(data, outfile)
14
15  def loadPick():
16     global pick
17     with open('result.json') as file:
18         data = json.load(file)
19
20     pick = data["pick"]
21     print(pick)
22
23  def combine():
24     global result, imageC, imageL, imageR, pick
25     (h,w) = imageC.shape[:2]
26
27     cng = cv2.cvtColor(result, cv2.COLOR_BGR2GRAY)
28     th, mask_c = cv2.threshold(cng, 1, 255, cv2.
   THRESH_BINARY)
29     mask_c = mask_c / 255
30
31     # right
32     src_pnts = np.empty([4,2], np.float32)
33     dst_pnts = np.empty([4,2], np.float32)
34     for i in range(4):
35         src_pnts[i][0] = float(pick[0][i][0])
36         src_pnts[i][1] = float(pick[0][i][1])
37         dst_pnts[i][0] = float(pick[1][i][0]+w)
38         dst_pnts[i][1] = float(pick[1][i][1]+h)
39     M = cv2.getPerspectiveTransform(src_pnts,
   dst_pnts)
```

```python
40      rn = cv2.warpPerspective(imageR, M, (w*3,h*3))
41      rng = cv2.cvtColor(rn, cv2.COLOR_BGR2GRAY)
42      th, mask_r = cv2.threshold(rng, 1, 255, cv2.
   THRESH_BINARY)
43      #cv2.imwrite("mask_r.png", mask_r)
44      mask_r = mask_r / 255
45
46      # left image appears upper left corner, but it
   still works in blending.
47      for i in range(4):
48          src_pnts[i][0] = float(pick[2][i][0])
49          src_pnts[i][1] = float(pick[2][i][1])
50          dst_pnts[i][0] = float(pick[3][i][0] + w)
51          dst_pnts[i][1] = float(pick[3][i][1] + h)
52      M = cv2.getPerspectiveTransform(src_pnts,
   dst_pnts)
53
54      ln = cv2.warpPerspective(imageL, M, (w*3,h*3))
55      lng = cv2.cvtColor(ln, cv2.COLOR_BGR2GRAY)
56      th, mask_l = cv2.threshold(lng, 1, 255, cv2.
   THRESH_BINARY)
57      mask_l = mask_l / 255
58      #cv2.imwrite("mask_l.png", mask_l)
59
60      # alpha blending
61      # mask element: number of pictures at that
   coordinate
62      mask = np.array(mask_c + mask_l + mask_r, float
   )
63
64      # alpha blending weight
65      ag = np.full(mask.shape, 0.0, dtype=float)
66      # weight: 1.0 / (num of picture)
67      ag = 1.0 / np.maximum(1,mask) # avoid 0
   division
68
69      # generate result image from 3 images + alpha
   weight
70      result[:,:,0] = result[:,:,0]*ag[:,:] + ln[:,:,
   0]*ag[:,:] + rn[:,:,0]*ag[:,:]
71      result[:,:,1] = result[:,:,1]*ag[:,:] + ln[:,:,
```

```python
71  1]*ag[:,:] + rn[:,:,1]*ag[:,:]
72      result[:,:,2] = result[:,:,2]*ag[:,:] + ln
    [:,:,2]*ag[:,:] + rn[:,:,2]*ag[:,:]
73
74      #cv2.imwrite("result.jpg", result)
75      if dataset == 0:
76          cv2.imwrite("wall-stitched.png", result)
77      elif dataset == 1:
78          cv2.imwrite("door-stitched.png", result)
79      elif dataset == 2:
80          cv2.imwrite("house-stitched.png", result)
81      else:
82          cv2.imwrite("pittsburgh-stitched.png",
    result)
83      cv2.imshow("result", result)
84
85  '''
86  pick 4 points from right image (red point)
87  '''
88  def right_click(event, x, y, flags, param):
89      if event == cv2.EVENT_LBUTTONUP:
90          mousePick(x, y, 0)
91
92  '''
93  pick 4 points from center (correspond to right,
    red point)
94  '''
95  def center_click_r(event, x, y, flags, param):
96      if event == cv2.EVENT_LBUTTONUP:
97          mousePick(x, y, 1)
98
99  '''
100 pick 4 points from left (blue point)
101 '''
102 def left_click(event, x, y, flags, param):
103     if event == cv2.EVENT_LBUTTONUP:
104         # add your code to select 4 points
105         mousePick(x, y, 2)
106         # pass
107
108 '''
```

```python
pick 4 points from center (correspond to left,
blue point)
'''
def center_click_l(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONUP:
        # add your code to select 4 points
        mousePick(x, y, 3)
        # pass

'''
idea: handle mouse pick
idx
0: right
1: center (correspond to right)
2: left
3: center (correspond to left)

you can also create your own function for left +
center selection
'''
def mousePick(x, y, idx):
    global rn, cn, ln, imageR, imageC, imageL,
pick
    if idx == 0:
        src = imageR
        dst = rn
        wn = "right"
        col = (0, 0, 255)  # right side red
    elif idx == 1:
        src = imageC
        dst = cn
        wn = "center"
        col = (0, 0, 255)  # right side red
    # you need to add idx 2, 3 cases
    elif idx == 2:
        src = imageL
        dst = ln
        wn = "left"
        col = (255, 0, 0)  # left side blue
    elif idx == 3:
        src = imageC
```

```python
147             dst = cn
148             wn = "center"
149             col = (255, 0, 0)  # left side blue
150         else:
151             return
152
153     print(idx, x, y)
154     pick[idx].append((x,y))
155     dst = src.copy()
156     # red BGR color in OpenCV, you need to set to
   blue on left side
157     # place circle on the picked point and text
   its serial (0-3)
158
159     if idx > 1:
160         color = (255, 0, 0)
161     else:
162         color = col
163
164     for i in range(len(pick[idx])):
165         dst = cv2.circle(dst, pick[idx][i], 5,
   color, 2)
166         dst = cv2.putText(dst, str(i), (pick[idx][
   i][0]+10, pick[idx][i][1]-10),
167                               cv2.FONT_HERSHEY_SIMPLEX
   ,1, color, 1)
168
169     # please make sure when idx == 3, you need to
   show red color circle in dst
170     # this example erases red circle
171
172     cv2.imshow(wn, dst)
173     # to make sure image is updated
174     cv2.waitKey(1)
175     if len(pick[idx]) >= 4:
176         print('Is it OK? (y/n)')
177         i = input()
178         if i == 'y' or i == 'Y':
179             if idx >= 3:
180                 savePick()
181                 combine()
```

```python
182                     elif idx == 0:
183                         print('center 4 points')
184                         cv2.setMouseCallback("center",
     center_click_r)
185                     elif idx == 1:
186                         # only taking care of right and
     center, you need to replace 2 lines to start
187                         # picking left and center
     correspondence
188                         print('left 4 points')
189                         cv2.setMouseCallback("left",
     left_click)
190                     elif idx == 2:
191                         print('center 4 points')
192                         cv2.setMouseCallback("center",
     center_click_l)
193                         # you need to add pick code
194                 else:
195                     pick[idx] = []
196                     dst = src.copy()
197                     cv2.imshow(wn, dst)
198
199
200 parser = argparse.ArgumentParser(description='
     Combine 3 images')
201 parser.add_argument('-d', '--data', type=int, help
     ='Dataset index', default=1)
202 args = parser.parse_args()
203 dataset = args.data
204
205 if dataset == 0:
206     imageL = cv2.imread("wall-left.png")
207     imageC = cv2.imread("wall-center.png")
208     imageR = cv2.imread("wall-right.png")
209 elif dataset == 1:
210     imageL = cv2.imread("door-left.jpg")
211     imageC = cv2.imread("door-center.jpg")
212     imageR = cv2.imread("door-right.jpg")
213 elif dataset == 2:
214     imageL = cv2.imread("house-left.jpg")
215     imageC = cv2.imread("house-center.jpg")
```

```python
216        imageR = cv2.imread("house-right.jpg")
217 else:
218        imageL = cv2.imread("pittsburgh-left.jpg")
219        imageC = cv2.imread("pittsburgh-center.jpg")
220        imageR = cv2.imread("pittsburgh-right.jpg")
221
222 result = cv2.copyMakeBorder(imageC,imageC.shape[0
    ],imageC.shape[0],imageC.shape[1],imageC.shape[1],
223                                borderType=cv2.
    BORDER_CONSTANT,value=[0,0,0])
224
225 print(imageL.shape,imageC.shape,imageR.shape,
    result.shape)
226
227 cv2.namedWindow("left",cv2.WINDOW_NORMAL)
228 cv2.namedWindow("center",cv2.WINDOW_NORMAL)
229 cv2.namedWindow("right",cv2.WINDOW_NORMAL)
230 cv2.namedWindow("result",cv2.WINDOW_NORMAL)
231
232 ln = imageL.copy()
233 cn = imageC.copy()
234 rn = imageR.copy()
235
236 cv2.imshow("left", ln)
237 cv2.imshow("center", cn)
238 cv2.imshow("right", rn)
239 cv2.imshow("result", result)
240
241 pick = []
242 pick.append([])
243 pick.append([])
244 pick.append([])
245 pick.append([])
246
247 print('use saved points? (y/n)')
248 i = input()
249 if i == 'y' or i == 'Y':
250     loadPick()
251     combine()
252 else:
253     print("right 4 points")
```

```python
254        cv2.setMouseCallback("right", right_click)
255
256 cv2.waitKey()
257
258 # close all open windows
259 cv2.destroyAllWindows()
260
```