Homography

Part1

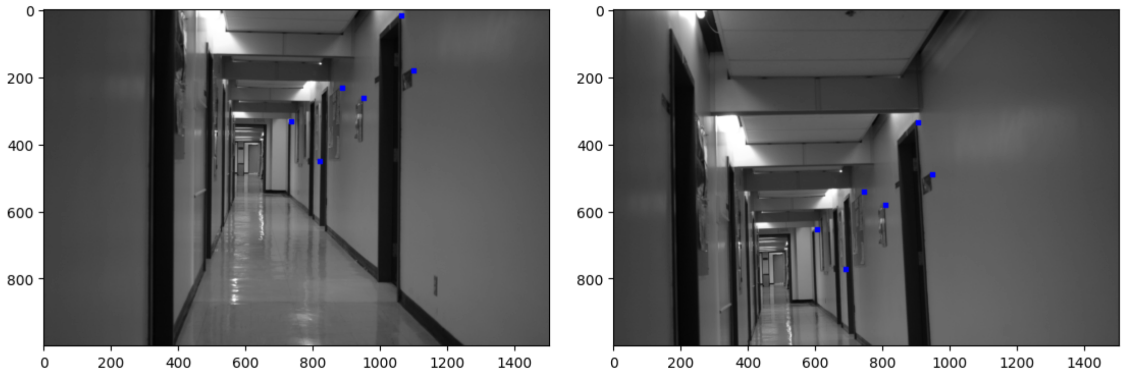


Figure1: Case A, and the left is hallway1, the right is hallway2

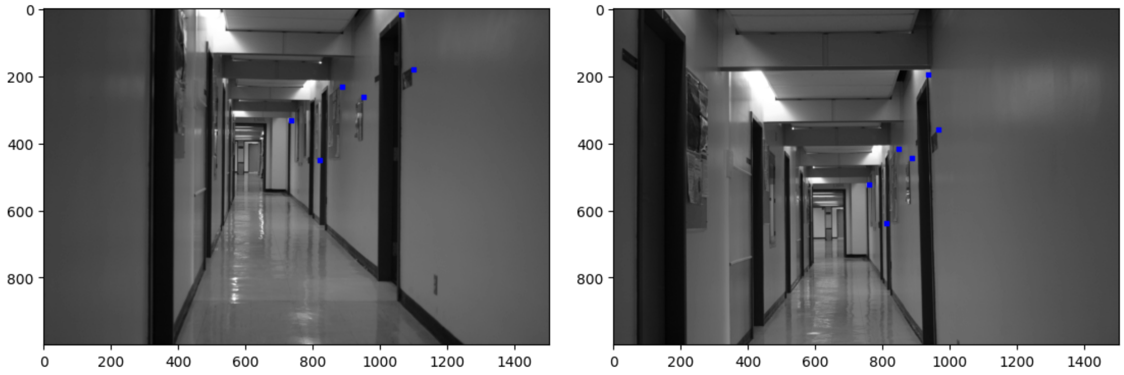


Figure2: Case B, and the left is hallway1, the right is hallway3

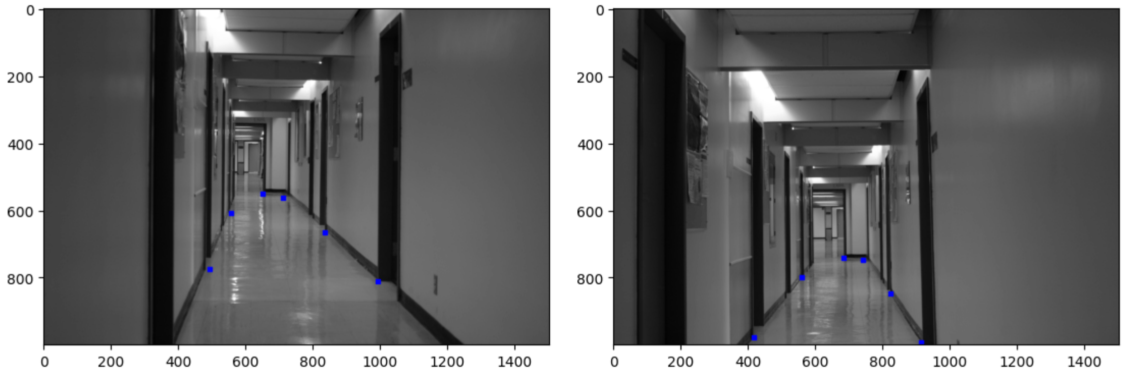
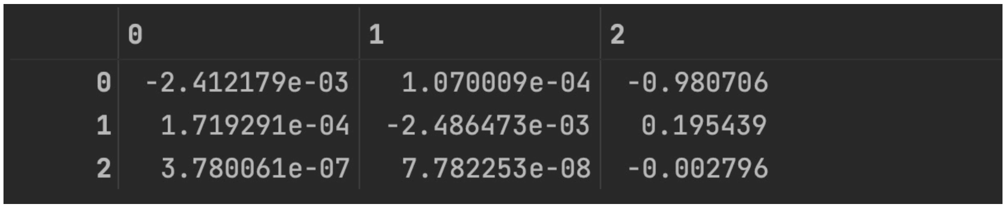


Figure3: Case C, and the left is hallway1, the right is hallway3

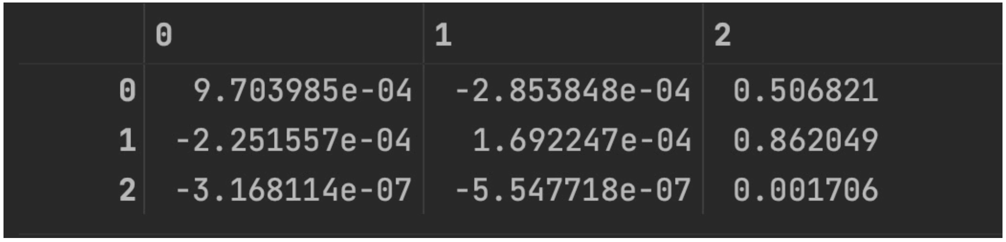
To clarify, my labtop is MacBook, which has poor support of **datacursor**. As a result, I choose the point by hand through checking their pixel coordinates.
I take six pairs of coordinates. Figure 1, 2, 3 show the Case of A, B, and C in turn.

Part2 As my code shown, I use get_homography function to get a fit homography H to the selected points.



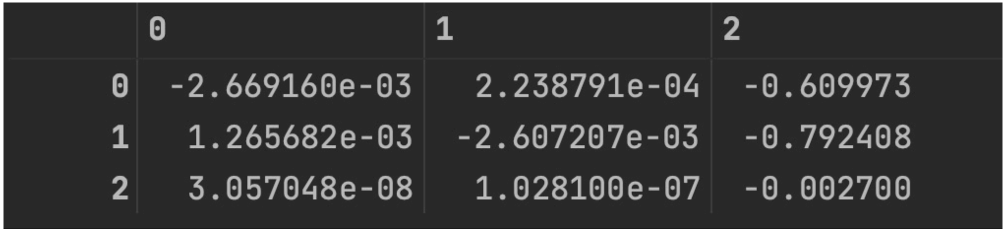
	0	1	2
0	-2.412179e-03	1.070009e-04	-0.980706
1	1.719291e-04	-2.486473e-03	0.195439
2	3.780061e-07	7.782253e-08	-0.002796

Figure4: Homography matrix H1



	0	1	2
0	9.703985e-04	-2.853848e-04	0.506821
1	-2.251557e-04	1.692247e-04	0.862049
2	-3.168114e-07	-5.547718e-07	0.001706

Figure5: Homography matrix H2



	0	1	2
0	-2.669160e-03	2.238791e-04	-0.609973
1	1.265682e-03	-2.607207e-03	-0.792408
2	3.057048e-08	1.028100e-07	-0.002700

Figure6: Homography matrix H3

Figure 4, 5, 6 shows the homography matrix of Case A, B, C correspondingly.
For Case A, the matrix scale the hallway1 a bit larger, then translate down and left.
For Case B, the matrix scale the hallway1 a bit larger, then translate down and left.
For Case C, the matrix scale the hallway1 a bit smaller, then rotate right.

Part3

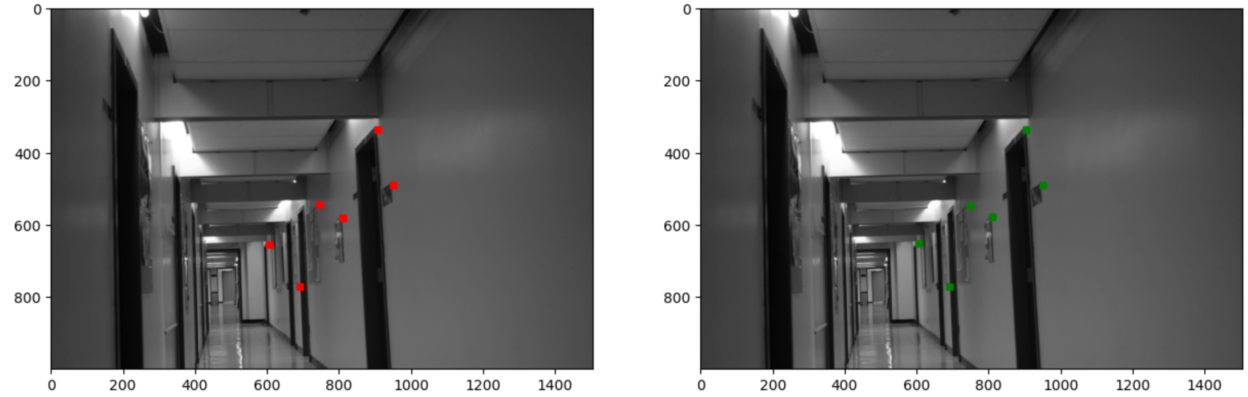


Figure7: Case A

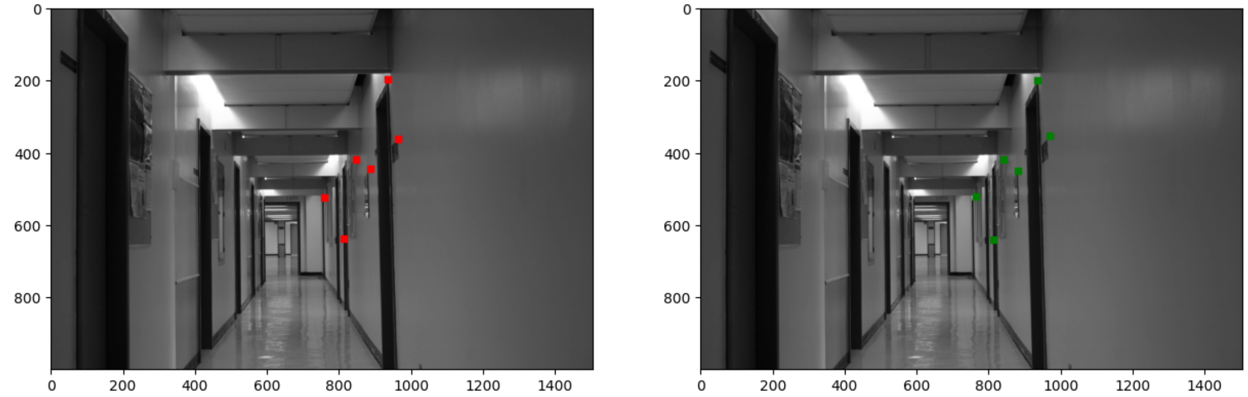


Figure8: Case B

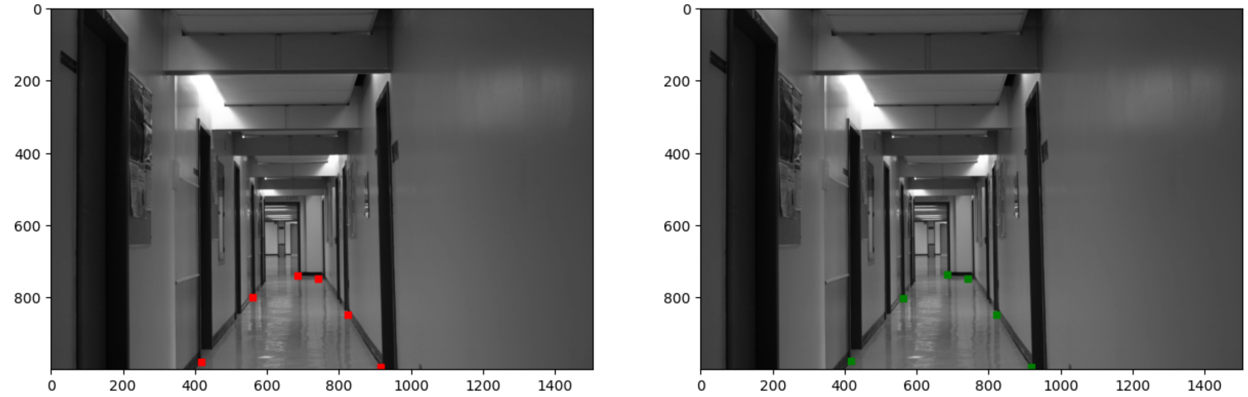


Figure9: Case C

Part4

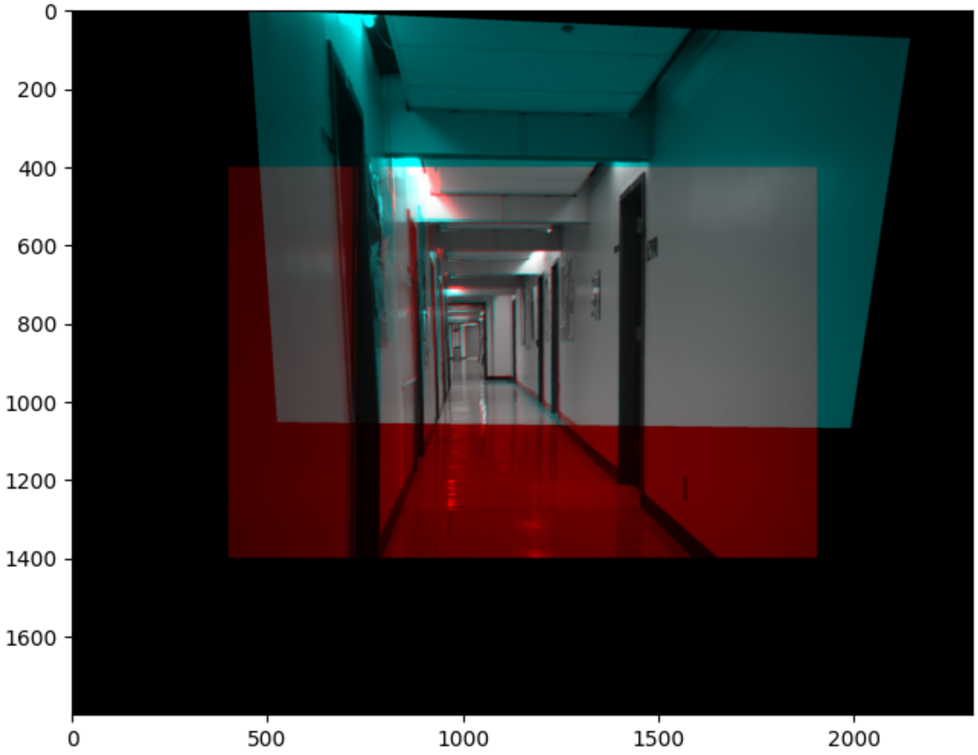


Figure10: Case A

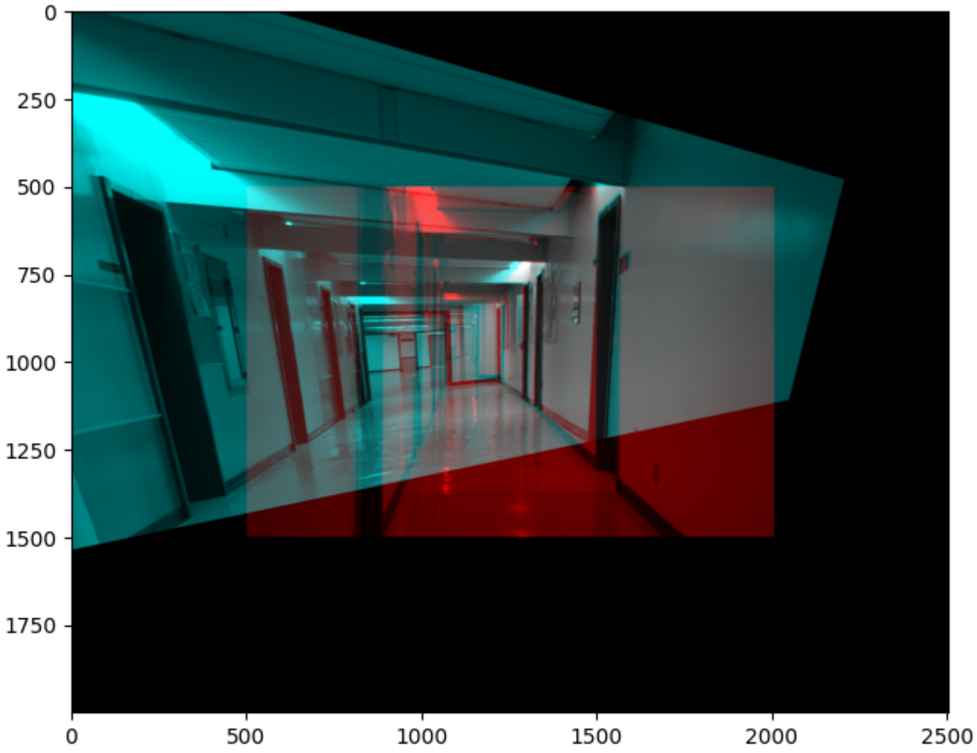


Figure11: Case B

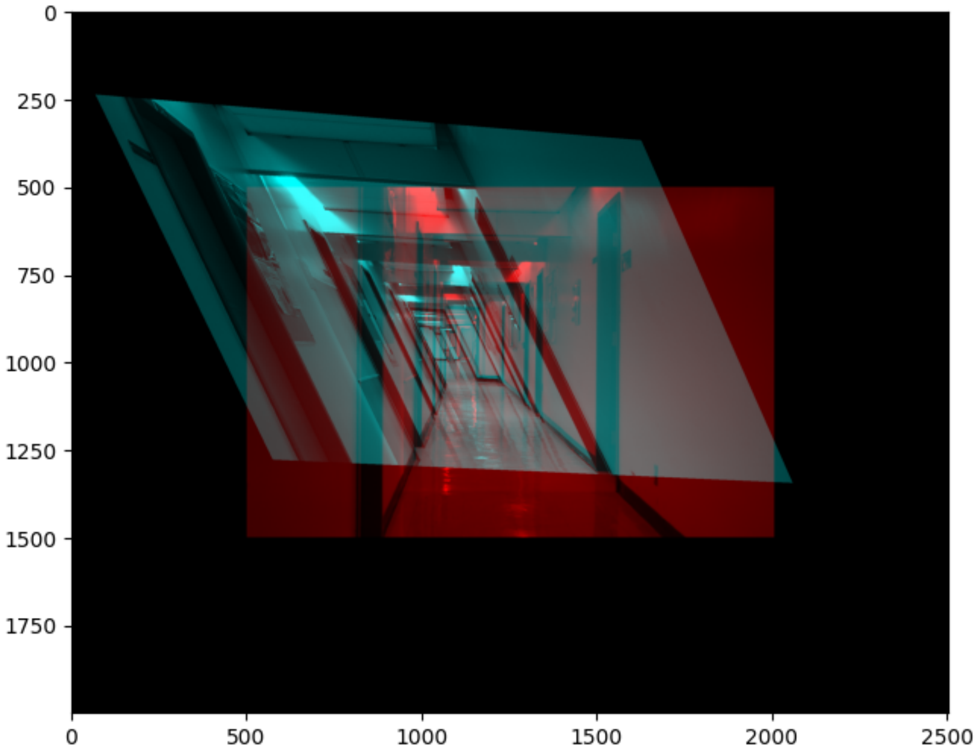


Figure12: Case C