Part 1 : I used same method I utilized in theory question 3. Made matrix A exactly same as Q3, and used svd to obtain last column of V correesponding to smallest singular value of A. And then return that value whose size is 1 as 3*3 matrix.

In compute_h_norm, I normalized coordinates by dividing each of them with 1600 or 1200(x&y coordinate's maximum value). In case of x coordinates, divided with 1600. In case of y coordinates, divided with 1200. And obtained matrix H using the result. After that, I had to perform some multiplication like below to use matrix H in de-normalized scale coordinates. I multiplied 1600/1200 to h_01, 1600 to h_02, and did similar things to h_10 and h_12 & h_20 and h_21.

(2N×9)

$$\begin{bmatrix} \frac{x_1^i}{1600} \\ \frac{y_1^i}{1200} \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \frac{x_2^i}{1600} \\ \frac{y_2^i}{1200} \\ 1 \end{bmatrix} \leftarrow \text{P\_in}$$

$$\frac{x_1^i}{1600} = \frac{h_{00} \times \frac{x_2^i}{1600} + h_{01} \times \frac{y_2^i}{1200} + h_{02}}{h_{20} \times \frac{x_2^i}{1600} + h_{21} \times \frac{y_2^i}{1200} + h_{22}}$$

$$x_1^i = \frac{h_{00} \times x_2^i + \frac{1600}{1200} h_{01} \times y_2^i + 1600 \times h_{02}}{\frac{1}{1600} h_{20} \times x_2^i + \frac{1}{1200} h_{21} \times y_2^i + h_{22}}$$

Part 2 : set_cor_mosaic() – I used corners of pillar, chair, sign, and other structures to make p_in and p_ref.

warp_image(igs_in, igs_ref, H) – I used two kinds of matrixes here. Ip and wp. Each column of them has x coordinate value in first row, y coordinate value in second row, and 1 in third row(homogeneous coordinates). wp is a matrix of homogeneous coordinates in viewpoint of igs_ref. Columns of wp have values of 0 0 1 / 1 0 1 / … / 1599 0 1 / 0 1 1 / …, which are every possible (x,y) combination in 1600*1200 size image. Ip is a matrix of homogeneous coordinates in viewpoint of igs_in. I obtained each ip columns corresponding to each wp columns by matrix multiplying inverse of H to each wp columns, and used that corresponding ip column as a source of pixel extraction. By applying binary interpolation method, I found appropriate pixel value for each igs_ref viewpoint. For igs_warp, I only obtained part of igs_in that can be seen in viewpoint of porto2(i.e. x value 0~1600, y value 0~1200 in viewpoint of igs_ref). For igs_merge, I obtained entire igs_in by introducing negative coordinate values in viewpoint of igs_ref(see x-1600 & y-500 in line 88 & 89).

Part 3 : set_cor_rec() – designated four vertices of iphone and set them in c_ref. And, made c_in by imagining appropriate rectangular that can be seen when I see iphone at the front.

Rectify(igs, p1, p2) : obtained matrix H to transform p2(c_ref)-corresponding coordinate system into p1(c_in)-corresponding coordinate system. And performed same operations I used in part 2 to obtain igs_warp.