1. Dependencies:
   sys, numpy, matplotlib.pyplot, cv2

2. Explanation of the method:
   In this case, we assume the perception system is going to be used always in 1.5 meter wide corridors with similar shelves and walls and a single human in the corridor. The human is always approximately 2m away from the depth camera, and the robot is assumed to be always approximately in the center of the corridor, oriented parallel to the walls.
   Thus, the problem can be separated into two parts: detect the human and get the depth, then calculate the clearance according to the depth. In this scenario, wo can directly use the depth image instead of converting it into point cloud, because the camera is in the center of the corridor, and it is always oriented parallel to the walls.
   We use two provided images to stitch a “human free” background. Then we use this background to get the human depth information in all these three images. After generating the mask of human, we use sliding window to get rid of the noise in order to calculate the accurate depth of human. Next, we look for the points on the shelf and wall with approximately same depth to get the information of the corridor. In the end, we can get the safer side and clearance by comparing the distance between the shelf and human, and the distance between human and the wall.

3. Next steps:
   In this implementation, we only consider very similar situations. We assume the distance between human and the depth camera is always 2m, and we always have the same background. However, if the background is different, or there is more than one person in the depth image, our current algorithm will fail.
   In this case, we need to generalize a new algorithm, which should be based on point cloud. Starting with transfer depth image into point cloud, we can use off-the-shelf algorithms such as RANSAC to fit planes. That is, we can separate the point cloud to different clusters, for example, “ground”, “shelf”, “wall”, and “human”.
   Then, we can get the depth of human by averaging the depth of “human” point cloud, and search in “shelf” and “wall” clusters to look for the coordinates of the points which have similar depth as the average of “human” does. In this way, we can have the information of where are the shelf and wall in the same plane as human. From this, we can easily calculate the distance between the shelf and the human, and the distance between the human and the wall.
   There are other methods which are capable to detect human from depth images. For examples, “Human Detection Using Depth Information by Kinect” by L. Xia, CC. Chen, and J.K. Aggarwal (http://cvrc.ece.utexas.edu/Publications/HAU3D11_Xia.pdf) introduce a method to detect human in depth images by using head contour model. After successfully detecting the human, we can calculate the distance by following the rest of the steps mentioned above.