INFOMCV Assignment 3
Martino Fabiani, Taher Jarjanazi (Group 20)

Summary

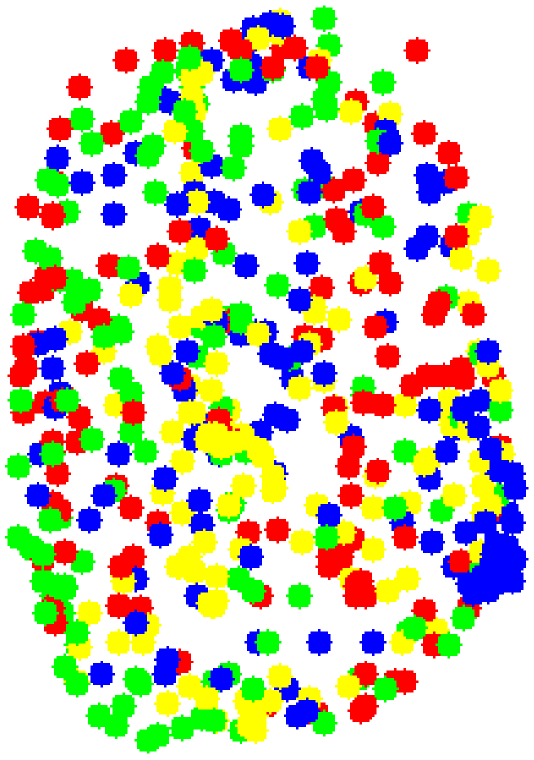
In the code development the first steps were setting the voxel space and the lookup table. Regarding the first, as in the previous assignment, setting the voxel space was performed but this time concerning the handling of infinite values. Firstly, it is checked whether the x and y coordinates of the voxel projection are finite, and if so, they are assigned to projection_x and projection_y. Otherwise, very large values (1e6) are assigned, limiting the value to a preset value. Finally, in the subsequent condition, it is then checked whether the projected voxel lies within the bounds of the foreground mask, setting the voxel to zero if it falls outside of it. Also, the voxel space had to be expanded to allow for more room for the persons' voxels to be visible throughout the video where every person is in a different position. For that we set the dimensions of the voxel space as (180,85,170). Moving on to the second, the creation of the lookup table relied on 2D projections in our space points, particularly focusing on those belonging to the foreground mask for all 4 cameras (the background subtraction code development and consequent foreground masks coincide with the implementation of the same function used in assignment 2). Preparing this table and the subsequent parameter division for each camera is a fundamental step of data preprocessing for creating color models.
Before developing the color models, however, a clustering process was necessary. This involved iteratively dividing the voxels into 4 clusters (subgroups) based on their spatial position, achieving an accurate division of the voxels into 4 groups (labels) of voxels belonging to the 4 people in the video (to achieve this only the x and z coordinates were considered in clustering since the y coordinate representing height is not necessary in this case to identify a subject). Once this was done, cleaning of ghost voxels and outliers was performed (the explanation of this process will be provided in the choice tasks section).
After the clustering, a division of the voxels was made by filtering them, specifically selecting voxels belonging to the torso area of the people bodies. This was done because it was established that the greatest color differences between subjects are most evident in T-shirts, allowing our model greater accuracy. Then, through these voxels and using the lookup table, corresponding colors were selected from the 4 training images (1 per camera), images where a clear division between subjects can be observed. Once the colors for each person were obtained, a Gaussian mixture model (GMM) was implemented and trained on these colors, particularly trained to recognize 3 different color clusters among the fronts, concluding the offline phase with a model for each subject for each camera.
The last part of the process, the online phase, follows a similar procedure at this point. In this section, the explanation of classification with a single camera is explained, leaving the implementation of the other cameras to the choice tasks section. For each frame of the video, clustering and voxel division are performed in the same way as the offline phase. For each voxel belonging to a person, the corresponding 2D pixel color value in the color frame is obtained and the predict2() function is used to extract the logarithmic probability of the model for the provided sample. This value represents how much the test value "resembles" the model, allowing us to determine how much a voxel may or may not belong to a subject in the video. This process is carried out for all voxels belonging to a subject, and the values are stored in an 'overall_logsum' parameter that considers the sum of these values (as it is logarithmic probability, summing is efficient as it coincides with the product of probabilities, thus expressing the probability that an entire subject "resembles" the corresponding model). After doing this for all 4 models, the final result corresponds to 4 "similarity" values, and we simply select the highest one to determine the final label. This process is then carried out with the other camera models. Clearly, this process relying only on one camera is very sensitive to problems of voxel occlusion, making it very fragile in recognizing labels if

one person is walking prospectively behind another. To overcome this problem, considering all 4 cameras is necessary.

Link to video

We make 2 videos. One using a single camera (camera 1) and another using all 4 cameras.
[4_cam_gmm.mp4]
[1_cam_gmm.mp4]

Trajectory image



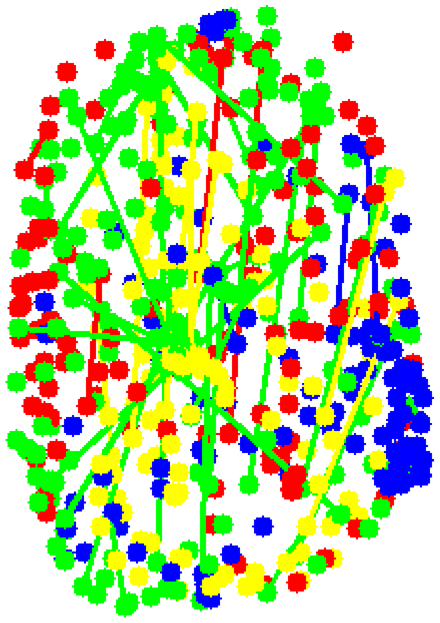
Choice tasks

(Indicate which ones you did, and how you did them; Approx. half a page.)

- CHOICE 1) 4 CAMERAS:
  In the development of a matching system based on 4 cameras rather than 1, the difference in code concerns the online phase. Following the preprocessing of voxels and color extraction, the models from each camera are utilized through 'predic2()' to obtain logarithmic probabilities with the same process used for a single camera. However, the 'sum_logprob' is this time stored in a

list and the calculation is repeated for all 4 camera models, resulting in a probability matrix. The Hungarian assignment method, a bipartite assignment method that minimizes costs to achieve an optimal assignment, is then applied to this matrix to obtain matched labels for the utilized camera compared to the offline phase model. This process is then repeated for all 4 cameras, each reaching an assignment. Finally, the 'assign_final_labels' function is used to define the last label assignment based on the frequency of occurrence of an assignment, assigning the first list that appears twice.

- CHOICE 4) SMOOTH TRAJECTORIES: We have tried to implement a way of achieving smoother trajectories by connecting the points with the same color with each other by a line of the same color using cv2.line(). This approach however might be too naïve when color switching in the final voxels of the people is frequent.



-

- CHOICE 5) DETECTION AND FILTERING OF GHOST VOXELS AND OUTLIERS

In the removal of outliers and ghost voxels, the procedure was as follows. Following clustering performed on the entire list of voxels belonging to the lookup table, the same pixels, the clustered voxels without considering the height component y, are used to compute the distance from the center of the obtained cluster. Through this distance, it is possible to verify for each voxel how far it is from the recognized center within a person and thus filter out all voxels considered too distant according to an experimentally chosen threshold. However, having modified the result of the previously performed clustering, it is necessary to repeat it again to reassign the new labels and the new centers for the filtered voxels.