Multi-angle Human Detection & Tracking System
–Term Project Report for Image & Video Processing

Feng Wang[1] Weixi Zhang[2] Zhuoya Shi[3]

***Abstract*— This project proposes an improved version of an existing approach for people detection and tracking among multi-view videos. The main contribution of this project is the improvement for cross-view matching by double trigger.**

I. Overview

The ultimate goal of this project is to propose a framework for pedestrian detection and tracking among videos from multi-angle. The main tasks are: 1) detecting pedestrian in a single-view video by faster-RCNN; 2) pedestrian tracking in single-view video; 3) cross-view matching among multi-view videos.

II. Project Accomplishment

A. Pedestrian Detection in Single-view Video

To detect people from video frames, fast-Region based Convolution Network method is implemented because of its accuracy for object detection. Faster-RCNN is able to detect 20 pre-defined categories of objects. Here we enable the "person" option for pedestrian detection.

Since a good object detection algorithm is not the focus of this project, we used a Tenforflow implementation of fast-RCNN detection framework developed by Chen&Gupta directly. The code supports VGG-16, resNet V1 and Mobilenet V1 models. We adopted VGG16 Model, which is trained on VOC 2007 trainval, and tested on the dataset PETS 2009.

In order to show the detection result directly in the video, a bounding box was added to the output of faster-RCNN. For each frame in the video, faster-RCNN can detect objects with a confidence score, RCNN features with 4096 entries, object ID and frame ID. We set the threshold for confidence score as 0.8, which means the algorithm must be more than 80% sure that the detected object is a person. The detection result for each object in the frame will be saved as a single .mat file with the fore-mentioned output information.

B. Pedestrian Tracking in Single-view Video

To store the historical information of each detected object, a virtual object buffer (VOB) is introduced.The information stored in the VOB are: coordinates and size of bounding boxes, RCNN features with 4096 elements, object ID, frame ID, confidence scores and motion vectors. If an object is detected as a person, a VOB will be generated for it. In the following frames, once an object is recognized as the same person, it's information will be added to the VOB.

For each object detected in the new frame, its feature will be compared with the feature of objects detected in the former frame. There are two criteria for VOB comparison: 1) The intersection-over-union ratio between the newly detected object and the object detected in the former frame. If the ratio is larger than the preset threshold, we will consider the object detected in the new frame and the objected in the former frame is the same person. The threshold for IOU ratio is 0.3. 2) The Euclidean distance between the newly detected object's feature and the feature of the object detected in the former frame is calculated. To determine the newly detected object is the same person with the one detected in the former frame, two conditions must be satisfied: on one hand, the Euclidean distance between the feature of two detected objects is the smallest; on the other hand, the object in the former frame has not been assigned with other object in the new frame.

However, there might still be cases where the newly detected object does not satisfy either of the above criteria. When this happens, if the new object is within the threshold of frame distance (i.e. 5 frames) , a factorized graph matching algorithm is utilized to associate the objects.

To implement the factorized graph matching algorithm, essential elements are defined as follows: 1) Each detected object is defined as a node in the frame, and the RCNN features are considered as node attributes. 2) The relationship between two adjacent objects is defined as an edge in the frame, and the spatial information is used as edge attributes. For the spatial information, we used the coordinates of midpoints on the bottom edges. 3) The moving vector of each object is taken into consideration to predict the potential position of each node.

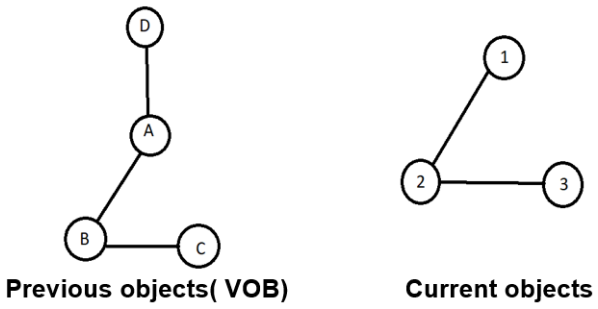

Fig. 1. Illustration of Factorized Graph Matching

[1] Feng Wang is a Master student of Electrical Engineering, New York University, Brooklyn, NY 11201, USA fw778@nyu.edu

[2]Weixi Zhang is a Master student of Electrical Engineering, New York University, Brooklyn, NY 11201, USA wz1219@nyu.edu

[3]Zhuoya Shi is a PhD student of Civil Engineering, New York University, Brooklyn, NY 11201, USA zs1110@nyu.edu

The factorized graph matching algorithm will return the best matching result between the two frames.

	1	2	3
A	1	0	0
B	0	1	0
C	0	0	1
D	0	0	0

Fig. 2. Matching result generated by Factorized Graph Matching Algorithm

In another case, if the frame distance is larger than the threshold, which means the newly detected object is 5 frames away from, it will be treated as a new person, and a new label will be assign to it.

Further steps to deal with the newly detected objects that does not satisfy the predefined criteria will be illustrated in the next section.

C. Cross-view Matching

An important step for multi-view pedestrian tracking is the cross-view matching for objects successfully detected and been tracking in each single-view video. There are three major steps for cross-view matching.

First, homography projection is conducted to project reference view frame (i.e. view 5 or view 7) to ground view(i.e. view 1), that is the two frames are adjusted so that they are from the same view angle. Then we can impose spatial constraints for the next step.

Second, graph matching is done by implementing factorized graph matching algorithm. To verify whether the association is correct or not, the ”cost”(sum of node and edge distance with associated node and edge from the other view) will be calculated and compared with the minimum ”cost” of other previously-associated objects. If it is smaller, we would take the matching result as correct. Otherwise, the new object will be assigned a new label. This graph matching is triggered every time a new person is detected in the reference view video.

Based on the result from the previous matching, labels assigned to the pedestrians in different view of videos are unified. We assign the label of the object with longer tracklet in one view to the associated object with shorter tracklet in the other view.

D. Subtasks Assignment

Weixi Zhang is responsible for establishing the environment for faster-RCNN. Zhuoya Shi and Weixi Zhang both worked on reproducing the original project. Feng Wang is responsible for debugging and improvement of cross-view matching.

III. Existing Problems and Solutions

The major problems we met with this project is the miss-matching after occlusion problems. Pedestrian will be assigned a wrong label after being occluded by other person or objects (e.g. lamp-pole) or disappearing from the view for several frames (number of frames larger than the threshold of frame distance). This problem happens in both pedestrian tracking in single view video and cross-view matching process. There are three situations that can happen this problem. For each situation, a solution is illustrated here.

A. Case 1

If a person i is occluded by person j in a view, person j will occupy the label of person i after the occlusion, and person i will be assigned a new label. Fig.3 shows this situation below.

Fig. 3. Existing Problem: Case 1. frame T-1: The man with an orange bounding box(person i) is walking-by the man with a red bounding box(person j); Frame T: Person i is occluded with person j, and they both occupy a red bounding box; Frame T+1: Person i walks away with the red bounding box and person j is assigned a gray bounding box

To solve this problem, we adjust the frame distance threshold and the IOU threshold of each view, try to treat person i and person j as new VOB objects. After this, in multi-view matching, double trigger will be used to match those who are the same person. Fig.4 shows the refined result.

Fig. 4. Solution: Case 1. frame T-1: The man with an orange bounding box(person i) is walking-by the man with a red bounding box(person j); Frame T: Person i is occluded with person j, and they both occupy a red bounding box; Frame T+1: Person i walks away with the black bounding box and person j stays with the red bounding box

B. Case 2

If a person i’ in one view(e.g. reference view) is already matched with the person i in the other view (e.g. ground view), it still can happen that person j’ in the reference view will be matched to the person i in the ground view. Fig.5 can show this situation clearly.

To solve this problem, we can add a parameter to temporal VOB showing if this object is ”occupied”, meaning if it is already matched with an object in the other view. Fig.6 shows the correct result.

C. Case 3

In the cross-view matching is that if a person i in the ground view (e.g. View 1) is occluded or disappears for longer than the frame distance threshold, and reappears in

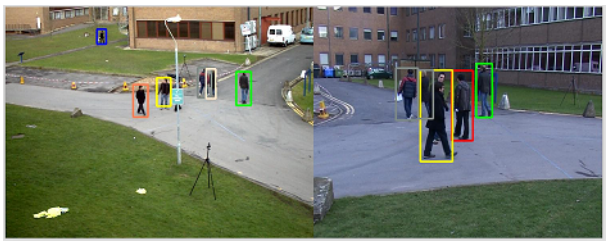

Fig. 5. Existing Problem: Case 2. In the right frame (reference view), two person are matched with the same person in the left frame (ground view) with gray bounding box.

Fig. 6. Solution: Case 2. In the right frame (reference view), two person are matched with the same person in the left frame (ground view) with gray bounding box.

this view, he will be recognized as a new person j and a new label will be assigned to him. However, he is in the reference view(e.g. View 7) showing as i' during this whole time. A simple illustration of this situation is shown in the Fig.7.

To solve this problem, we proposed a double-trigger solution. And it can solve this issue very well. The double trigger solution will be illustrated in detail in the following section. Fig.8 shows the correct result for this situation.

IV. Double Trigger

The main contribution of this project is the double-trigger solution for cross-view matching.

The fore-mentioned problems are mainly caused by the insufficient timing to trigger multi-view match. Fig 9 shows how double-trigger works for cross-view matching.

The idea is that we want to trigger the multi-view matching whenever a new object appears no matter which view he is in. Notice that we have two views: the ground view (i.e.view 1) and the reference view (i.e.view 7). In view 1, there is a person A disappeared for several frames (the number of frames is large than the frame distance threshold), so the two part cant be linked by single view match.When he reappears, he is recognized as a new person C. The same situation happens in view 7,and he is recognized as B and D separately. If the program only triggers multi-view matching when a new person appears in view 7. Person B in view 7 can be matched to person A in view 1 at time 2, and person D in view 7 can be matched to person C in view 1 at time 4. However, theres no way we can claim B and C is the same person.

How double-trigger works is that we also do multi-view matching when a new person appears in view 1. It means that we also do multi-view matching at time 1 and time 3. In match 3, we can match person B to person C, thus we

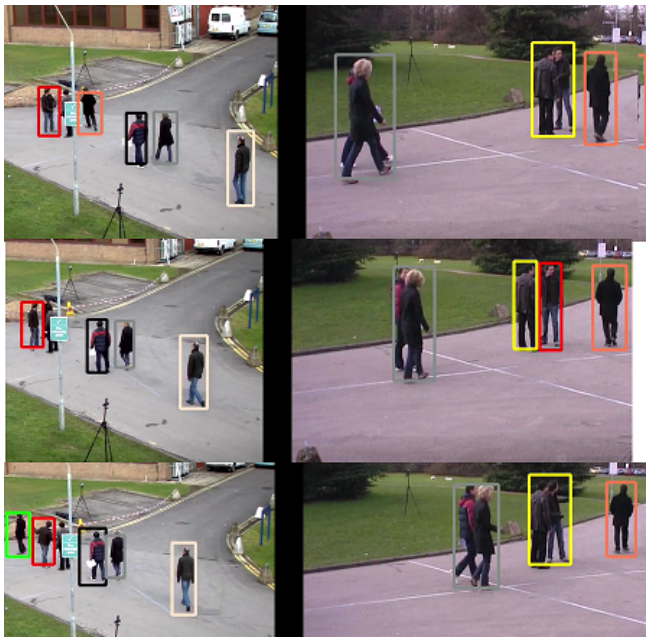

Fig. 7. Existing Problem: Case 3. Top:The man with orange bounding box showing in both views; Mid: The man with orange bounding box is occluded in the left view, but still showing in the right view; Bottom: The man is assigned a new label in the left view, but still showing with orange bounding box in the right view.

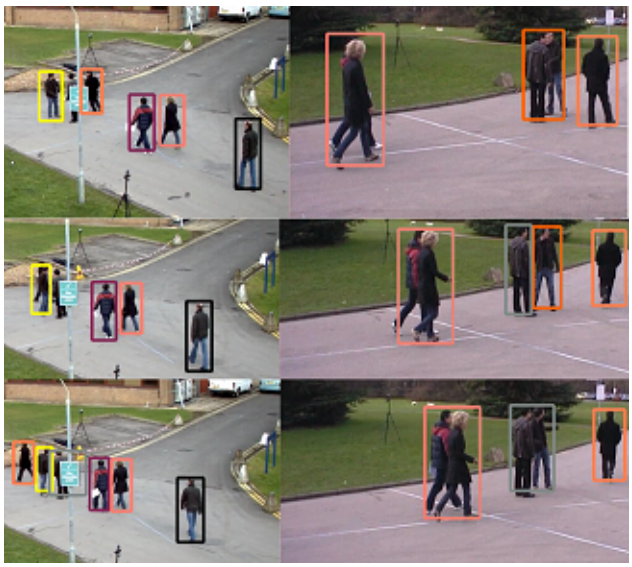

Fig. 8. Solution: Case 3. Top:The man with orange bounding box showing in both views; Mid: The man with orange bounding box is occluded in the left view, but still showing in the right view; Bottom: The man stays in orange bounding box in both views

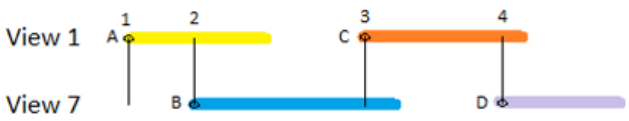

Fig. 9. Double-trigger

can solve this problem. Finally, we can claim that A, B, C, D are in fact a same person.

V. Automatically Generate Homography Matrix

A possible algorithm for automatic homography matrix generation is proposed in this part.

For the multi-view matching, homography matrix is needed to project one view to the other. We cannot generate the homography matrix with Harris feature points and SIFT descriptors. Fig 10 shows the matching result of background images of two views.

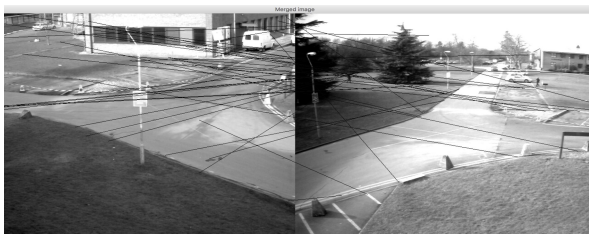

Fig. 10. Harris points and SIFT descriptor fail to generate the homography matrix

The reason is that the difference between the view angles of two views is too large. So the objects near the feature points are showing very different in two videos. Also, we only want the points on the ground for the homogeneous transform, which limits the feature points that can be used. However, with VOB from single view matching, we may generate the homography matrix automatically. We can search among the frames, to find some frames, when theres only one or two people in both scene, and get their position (since at least we will have one pair that matches).

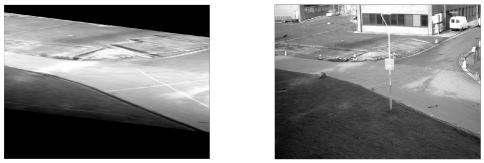

Fig. 11. Warping result of using feet as feature points to match

Note that we cant directly use the lower bound of each person as their feet, matching point to calculate homography matrix (as shown above, different from view 1 on the right, the white cross on the road is not correctly matched). This is because the bound of bounding box of RCNN feature is very unstable. It doesnt correctly describe the feet of the person, and keep on oscillating. However, we can take use of the position of their feet, and search for the Harris points near their feet.

Instead of doing descriptor matching, we limit the number of feature points in this way, and assume the point with largest Harris value is the matching point. After we collect

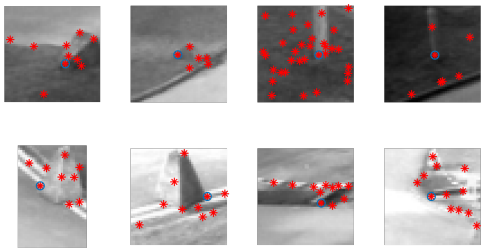

Fig. 12. Local Harris points

enough pairs in this way, we use RANSAC to delete the outliers. Finally, we can get the matrix and warped image.

Fig. 13. Local Harris point matching result

As we can see, comparing to what we got by matching feet, this is very reasonable. Note that during the matching, foreground features wont be selected, since the two views are very different. We need further work to complete it, and find ways to make it robust. There are several points we may need to enhance this algorithm: 1. more videos are needed, since we acquire frames that only 1 or 2 people are in each view, and hopefully they are close to some good background feature points. 2. use match of Harris points (that stress more on surrounding species of textures instead of corner shape (dominant directions of the corner)) on down sampled texture segmentation images, to get stable result.

VI. Remaining Problems

However, there are still some problems that need to be fixed to reach a perfect result.

A. Mis-detection When Occlusion

Sometimes, faster-RCNN fails to match RCNN when people occlude. The man in the middle of the right scene in Fig 14 is not recognized by faster-RCNN, but RCNN get him in Fig 15, the left scene. Since the multi-view matching algorithm is very much not robust, single view misses to match this person in later frames if we use the original algorithm.

B. Representing Multiple People With A Single Large Bounding Box

Sometimes faster-RCNN recognize several occluding people as a single object (in the left scene in Fig 16) and have a

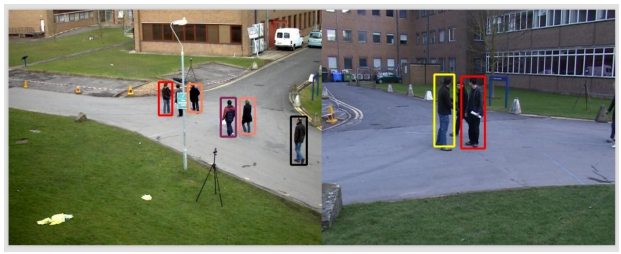

Fig. 14. The man in the middle of the right frame is not detected

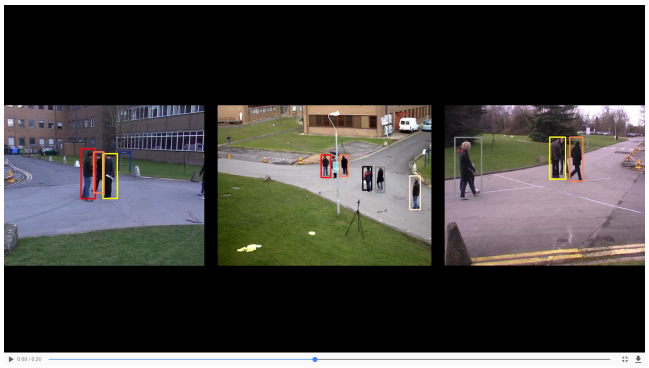

Fig. 15. The man in the middle of the left frame is detected

very high score with mixing feature of these people, which doesnt occur in RCNN, and this program cant handle this problem. We need further improved mask-RCNN algorithm to solve it.

Fig. 16. Large bounding box for multiple people

VII. SUMMARY

Based on the result, our project works well with pedestrian detection and tracking with multi-view videos. Double-trigger successfully improved the pedestrian matching result. Further work needs to be done for automatic generation of homography matrix for cross-matching process. Also, mistakes resulted from detection algorithm need to be addressed to reach a perfect result.

ACKNOWLEDGMENT

We thank Fanyi Duanmu for assistance, and professor Yao Wang for comments that greatly improve the depth of our thoughts to solve this problem.

REFERENCES

[1] Shaoqing Ren, kaiming He, Ross Girshick, jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Advance in Neural Information Processing System(NIPS), 2015.
[2] Nie W, Liu A, Su Y, Luan H, Yang Z, Cao L, Ji R. Single/cross-camera multiple-person tracking by graph matching. Neurocomputing, 2014.
[3] F. Zhou and F. De la Torre, Deformable Graph Matching, IEEE Tranctions on pattern Analysis and Machine Intelligence (PAMI), 38(9):1774-1789, 2016.
[4] K. Bernardin, and S. Rainer, Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP Journal on Image and Video Processing, Issue: 1, pp:1-10,2008.

APPENDIX

Appendix 1. A ”readme” file uploaded via NYU classes. It includes:

1) The link to download python code for Faster-RCNN, and instruction to run the code on NYU HPC
2) The link for result generated with Faster-RCNN
3) The code for factorized graph matching, and instructions to run the code.