DLAV Loomo Race

June 2022
Hugo Casademont, Sushen Jilla Venkatesa, Antoine Delaloye, Jonas Blanc

I. Introduction

We implemented a person tracking pipeline based on video. This pipeline was used in a human-robot race with a Loomo Robot following a person around a track based on the camera feed. A specific hand gesture is all that is needed to be selected as the person of interest to be then detected, tracked and followed by the robot. The main challenge consists of guiding the robot through many people without loosing the person of interest.

II. Models

Multiple goal-specific models are used in our pipeline. In this part we will briefly explain each of them and for what purpose it was used.

A. Mediapipe - Hand detection

The first tool we make use of is ”Mediapipe Hands” [1] which is a high-fidelity hand and finger tracking solution. This solution is useful to detect the hands and their landmarks (features) in the picture. This is done in two steps:

- Firstly, a ”Palm Detection Model” is applied on the whole picture in order to detect the overall hands.
- Secondly, the ”Hand Landmark Model” is applied on each of the detected hand regions in order to extract sets of 21 landmarks (3D positions) per hand.

B. Key point classifier - Hand classification

Once the hand landmarks are obtained, we need to determine the pose of each hand in the picture. For that purpose, a classifier is used, which takes the landmarks as inputs and sorts out, as a result, one of the 8 hand gestures for which the model was trained. This classifier is adapted from [2].

C. Yolox - Person detection

A YOLOX model is used for detecting the people appearing on an image. YOLOX stems from the classic multi-class object detector YOLO but it is anchor-free, has better performance and simpler design. The YOLOX model we used is an implementation in PyTorch [3]. The classification of the gesture of each hand from (II-B) will be combined with the people's bounding box information given by YOLOX. Indeed, the first person who performs the chosen hand gesture with both hands (needs to have them inside his bounding box) becomes the tracked subject.

D. Bytetrack - Person tracking

Bytetrack is a multi-object tracking model. Working with the bounding boxes created by the YOLOX model II-C, bytetrack assigns a unique ID to every box and tracks them in time. It outperforms many current (transformer-based) models and comes with a high frame rate [4]. The algorithm functions the following way:

- First, the next frame's positions are predicted with a *Kalman* filter. They are matched with high confidence bounding boxes using motion similarity.
- Then, a second matching is performed with bounding boxes of lower confidence for the predictions that didn't correspond to a high confidence bounding box

We use Bytetrack for tracking when the person of interest is inside the picture. The model handles partial occlusion but is not conceived for re-identifying a person who left the frame. In order to manage that situation we use the model presented in the next section.

E. OmniScaleNet - Person-reID

ByteTrack covers most of the tracking. However, as explained previously, it is not sufficient to complete the objective with a unique initialization. To ensure this step, we combine Bytetrack with a model called OmniScaleNet [5]. This algorithm uses a combination of multiple scales to extract and compare discriminative features. In addition to the conventional layers of a person-reID model, a unified aggregation gate is introduced to dynamically fuse multi-scale features. The architecture of this model led to state-of-the-art performances.

III. Tracking pipeline

In this section we explain how we combined the five models presented above to implement a robust tracking pipeline.

A. Initialization

The first step of our tracking algorithm is the initialization. YOLOX and Bytetrack are detecting and tracking every person in the frame. Mediapipe and the sign classification model are detecting and classifying all hands gestures. As soon as someone is detected performing a specific gesture with both hands (we are classifying both hands to avoid false positive) the algorithm tags the detected person as the person of interest.

Once the person of interest is selected, the OmniScaleNet is activated during a few sec to give the algorithm a reference for future re-identification. After these few seconds, ByteTrack takes back the tracking work.

B. Run time

During run time, when the person to follow stays visible/in the frame, only the ByteTrack model is used as it offers good performance and high frame rate. In situations where ByteTrack loses the target (out of the frame / total occlusion) it switches for the OmniScaleNet to re-identify the person of interest. As soon as the person of interest is re-identified, ByteTrack takes over for a smooth tracking. Since ByteTrack and OmniScaleNet don't have a direct matching between assigned ID, we used IOU score (Intersection Over Union) on bounding boxes to establish a correspondence between labels. These two models complement each other perfectly and provide robust tracking.

IV. Robot controls

To ensure equity between the groups competing, we were not allowed to modify the controls of the Loomo. We are able to run the detection/tracking on a usual MacBook Pro (and don't actually use the provided V100) thanks to the high frame rate (for reasons explained above). The Loomo sends images via wifi and we reply with the bounding box (x, y, width, height) of the person to follow, and the certainty score (if smaller than 0.5 the robot won't move).
Despite this limited flexibility for controls, we were able to improve the robot tracking based on some observations during training sessions.

A. Depth sensor clamping

The Loomo uses a depth sensor to estimate the distance to the person to track. Thus the person of interest needs to be in the view of the depth sensor for the robot to move. The depth sensor has a narrower view than the RGB camera. Then, if the detected person is in the frame of the camera and in the blind spot of the sensor, we clamp its position to make it fit in the sensor view (on its side thus making the robot turn in the direction of the detected person).

B. Inertia

Moreover we implemented (with limited success) ”inertia” in the robot movements. The goal is to keep the robot turning on himself (scanning around) in the direction where the person of interest went out of the frame. For that purpose we feed the Loomo the last detected bounding box in case of no detection. To avoid unwanted behavior of the robot when an occlusion occurs in the center of the image, we limited the inertia scenarios to the side of the image.

C. Bounding box size

On advice of the TAs, we used 10x10 bounding box which apparently provides a more robust depth estimation.

V. Results

This pipeline offers strong performance for person tracking and identifying. Even though we encounter minor difficulties the day of the race, we were able to achieves a race lap in 60s (whereas the overall best lap time was 49s).

VI. Discussion

We are somewhat disappointed with our performance during the race. Indeed, the robot lost focus in a straight line, which had never happened to us during a training session. In order to obtain as high a frame rate as possible we did not record the robot's feedback and we are therefore unable to determine the exact reason for this problem. We suspect, however, without tangible proof, that the light conditions played a role in this under-performance. All the training sessions took place in an underground environment, whereas the race took place in a building with abundant natural light. Due to the rather simple race track, the slightest hesitation of the robot was eliminatory.
On a side note, we are surprised that the victory was awarded to a team that was only detecting a hat, as we didn't think the rules allowed this kind of tracking. We are nevertheless very happy with the results obtained by our pipeline, indeed our robot had no difficulty to follow us in the turns and never confused the person to follow (using a single initialisation), which were, according to us, the most challenging tasks.

VII. Conclusion

We implemented a computer vision pipeline to track a person under partial and total occlusion. It combines multiple state-of-the-art models. We used it as a driving unit on a Loomo to make it follow a person. We participated in the second edition of the EPFL human-robot race and obtained satisfying results.

References

[1] Google. Mediapipe hands. https://google.github.io/mediapipe/solutions/hands.html. Last accessed on 27.05.2022.

[2] kinivi. Dji tello hand gesture control. https://github.com/kinivi/tello-gesture-control. Last accessed on 27.05.2022.

[3] Megvii-BaseDetection. Yolox. https://github.com/Megvii-BaseDetection/YOLOX. Last accessed on 27.05.2022.

[4] Yifu Zhang. Bytetrack. https://github.com/ifzhang/ByteTrack. Last accessed on 31.05.2022.

[5] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. pages 3701–3711, 2019. doi:http://dx.doi.org/10.1109/ICCV.2019.00380.