Visual Object Tracking
Project Report

theophile.stourbe

January 2025

1 IoU-based tracking

This technique is the most basic one; the goal is to calculate a score based on the area of intersection and union between the object we want to track and all of its candidates.
The algorithm to track multiple objects over multiple frames is the following:

```
for each frame do:
    current <- list of detected objects on current frame
    prev <- list of previously tracked objects

    correlation_matrix <- initialize correlation matrix with shape (c, p) with ones
                          c is length of current, p is length of prev

    for each object in current do:

        for each candidate in prev do:
            jacc_idx <- computed similarity score between object and candidate

            update correlation_matrix with jacc_idx for the right object and candidate

    associated_tracks <- we choose the best associations from the correlation_matrix scores

    if there is more detections in current frame than tracks:
        create new tracks for detections with the lowest IoU scores

    for each track in associated_tracks do:
        update associated object track
```

We use the Hungarian algorithm to solve the correlation matrix and find the best association for each object.

1.1 Implementation details

Missing frames

To improve tracking results, we include multiple previous frames, in case some detected objects overlap with each other or with the background. We only keep the last detection for each track.

Threshold on the Jaccard index

To prevent a new detected object to be associated with a track even if it is not even overlapping, we apply a threshold on the Jaccard index and create a new track for this object.

1.2 Observations

The algorithm works well; an example can be found in the video *out_iou.avi*. The main problems are:

- objects tracks are mixed when the two objects are close and overlap
- without the threshold on the Jaccard index, the tracks keep being reassigned to new objects, even if the object is at the opposite of the image from the other tracked one
- when two people walk by, we lose the track and a new track is created for the object that was obstructed

2 Kalman-Guided IoU Tracking

We now integrate Kalman filters in our algorithm to improve tracking, the goal is to predict the future position of the detected object given all its previous movements.
To integrate that in our algorithm here are the main changes:

- we need to create a list of Kalman filters for each tracked object along the video.
- when we detect a new object: we need to create a new Kalman filter and add it in the list, and initialize the filter by making a prediction and updating it with the current object position.
- before iterating the current detections: we predict the future position of all previously tracked objects from the selected ones, we keep the predictions and errors in a list to update later.
- when we associate an object with a track: we need to update the right filter from the list with the current object position, using the predictions and errors that we stored in the previous step.

2.1 Implementation details

Counter for the Kalman filter

The predictions from the Kalman filter are not accurate in the first steps and need a minimum of updates before being reliable. To prevent that I added a counter in the Kalman filter class and use the prediction values only if this counter is above a certain threshold.

Plotting predictions

In order to visualize better the improvements of the Kalman filter on the tracking results and tune better its parameters, I also plotted the prediction of the Kalman filter for each tracked object.

2.2 Extension to improve result

The addition of the Kalman filter, while being reliable, did not outperform much the original IoU-based tracking. As the frame rate is very small, each prediction is really close to the previous position and the results don't really change for objects tracked on consecutive frames. We would like the Kalman filter to allow us to retrieve tracks from objects that were obstructed during multiple frames, but the prediction is only one step ahead of the last detected point, so it does not always allow us to retrieve the right track.

An extension to correct this problem is to perform a multi-step ahead prediction with the Kalman filter. In order to do so we predict step by step by assuming that each prediction is the right one. We don't update the filter's values during these predictions and keep the original prediction and error to update during the association phase.

2.3 Observations

As said above, the integration of Kalman filters to the IoU-based tracking leads to results very similar to the original technique. The addition of multi-step ahead prediction can help to retrieve the tracks of objects obstructed during multiple frames, as can be seen in the video *out_kalman.avi*.

3 Appearance-Aware IoU-Kalman Object Tracker

In this technique we also rely on a similarity score between the detected object and its candidate, given an image classifier deep learning model. We retrieve the latent spaces of the object and the candidate through the model and compare them using the cosine similarity. We integrate this value with the IoU score in the Jaccard index with a weighted average. The cosine similarity being in the range $[-1, 1]$ we rescale it between $[0, 1]$.

3.1 Implementation details

To improve the results, another thresholding technique was tested:

- add a threshold for the similarity score, if the similarity is below a this threshold we set it to 0 to penalize more in the decision.
- add a threshold for the IoU score, we don't consider the track for this object if:
  the IoU score between the detection and the candidate is below this threshold

  or

  the similarity score is below the similarity threshold and the IoU is not based on a Kalman prediction

- filter predictions based on the confidence score

To help adjust the threshold, we plot the similarity score between the object and the matched track.

3.2 Observations

These thresholds were added following empirical observations during the tests. The similarity score is often high, this is why the similarity threshold was added. The second threshold is to prevent old tracks to be re-attributed to new objects, because one of the problem we can see is that tracks of objects leaving the frame are re-assigned to new objects. Finally, the confidence threshold is added to remove low quality detections and try to get better tracking results even if less objects are tracked.

We can still observe some errors, for example some objects leaving the screen are reassigned to tracks of objects leaving the screen at the same time or overlapping objects, but it happens in more complex scenarios than in the original IoU-based tracking technique.

The results using this technique can be seen in the video *out_reid.avi*.

4 To go further

As suggested in the last practical, we could improve these results using a better model to compute the similarity scores. Other approaches could be tested such as: increasing the IoU threshold for consecutively tracked objects once the Kalman predictions are used, thus restricting the choice of a track if the predictions are accurate, assuming that the objects keep its trajectory and was obstructed or not detected for this frame. Another approach would be to rely on the similarity scores of previous detections of the same tracks to assign this track for new detections, and restrict more the decision with the similarity threshold.