AcurusTrack. Data association component for precise multi-tracking.

AIHunters
oss@aihunters.com

Figure 1

1 INTRO

Since neural networks obtained their popularity, virtually any developer rushed to Deep Learning techniques to solve the tasks set in front of public safety software. However, we see it this way: neural networks can be incomparable in terms of detection and identification, but they can't be as effective in analysis. It is way too hard to create dependencies inside the 'black boxes' of neural networks. Thus, to our mind, in tracking it's more effective to use measurable and predictable approaches (such as probabilistic data techniques) to analyze the sequence of frames and the objects on them with their corresponding speed and position.

In this article, we do not claim to invent anything new. We gather some already existent scientific approaches and insights and create a practical realization of those approaches.

2 ABOUT THE TASK OF TRACKING

Tracking is one of the most common and important challenges in the public surveillance area. This is determined by the need to know where the objects are in each moment of the video we're analyzing. The classical detection algorithms aren't solving this issue. When we apply the object detection algorithm to the first frame of the analyzed video segment, the object's presence and position on the frame are determined, or the area of interest is just highlighted. The task of tracking, in turn, is to find the position of the selected object in subsequent frames. The main difficulties that tracking algorithms may face are occlusions (the tracker can easily switch to another object during occlusion, or lose the object that needs to be tracked), zooming, object rotation, changing lighting, motion blur, etc. And even though the classical algorithms are aimed at solving this problem, quite often they fail to do so. This happens due to the difficulties in adaptation and configuration or those algorithms. Neural networks-based algorithms, in their turn, cope with the task of tracking better, but are really hard to be interpreted and controlled. That is why we are offering an option of the highly-interpretable tracker based on the movements' physical realism analysis.

3 CLASSIC APPROACHES

Most of the available tracking algorithms consider the problem of tracking a single object in a video captured by a static camera. In

the case of object tracking in a video shot with a moving camera, additional difficulties arise. There are many tracking algorithms such as: trackers based on correlation filters [8] (used in popular dlib library), detection-based trackers [5], graph-based methods [12], [10], neural network-based [3], one of the best we've tried for moving cameras, occlusions, fast movement so far: [7]. For more detailed comparison of correlation approaches and trackers based on recurrent neural networks you can see the article [9]

Nevertheless, all the trackers mentioned above (except for neural networks-based ones) have virtually no way of coping with scaling, occlusions, objects rotation. And even if they do, they do not always show satisfactory results. Neural networks-based algorithms mostly solve the task of single object tracking, leaving objects interactions and possible options aside.

4 PROBLEM STATEMENT

Let's consider the issue of tracking N objects in a video captured by a moving camera Figure 1. Let's name the sequence of the frames that we consider $F_1, ..., F_N$. Each of these frames have the detections $D_1, ..., D_k, k <= N$. The task is to assign the index $i = 1, ..., N$, to each detection - namely, a certain id.

5 AUXILIARY VISUALIZATION

Before describing the algorithm that we use, let's consider the introduced visualization, which is designed to facilitate the algorithm's testing, debugging, and its performance analysis. We want to inject time dependency into consideration.

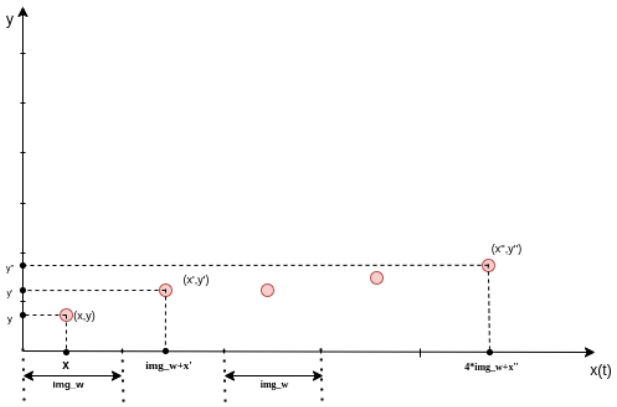

Figure 2

To do this, we draw the coordinates of detections as follows - on the OY axis we put off the y coordinates of objects, and on the OX axis - the x coordinates, but taking into account the number of the frame that we are considering. Thus, the object with coordinates (x, y), viewed on the i-th frame, will be displayed in the figure in the point $(i * img_w + x, y)$, where img_w is the image width.

To improve the quality of any tracking algorithms on a video shot by a moving camera, we recommend using our component for transition to a fixed coordinate system - EvenVizion [2]. After transitioning to fixed coordinates, we are faced with the classical multitracking task. We consider the direction of data association-based methods to be especially attractive for us [6]. Firstly, we

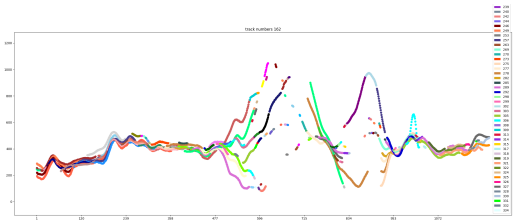

Figure 3: Example of visualizations obtained on arbitrary video (frame numbers are indicated on the OX axis)

move from individual detections to short tracklets, preprocessing the existing metadata on the basis of some similarity metric (for example, when working with faces, it can be iou - intersection over union - detections on k and k-1 frames, in the case of working with the body) it can be the sum of the differences in the coordinates of the corresponding parts of the body. Thus, we get preliminary initialization, a view with which we will continue to work.

We got inspired by the article describing the Markov chain Monte Carlo data association. For fairly simple cases we release a slightly shorter version. But if you are interested in a full version, feel free to contact us at oss@aihunters.com for more information.

The main advantages of this approach: we embed knowledge about the quality of auxiliary DL components through configurable parameters; We work with objects of arbitrary nature. We do not use identification, therefore we are not tied to persons, for example. We can work with objects that are very similar in appearance (with the presence of occlusions, classical tracking algorithms work poorly); we use a likelihood analysis based on physical interpretation of movement (with Kalman filter [4], [11]); the ability to parallelize video processing; the ability to use CPU after receiving metadata.

Thus, to summarize the sequence of steps described above: Firstly, we get the detections on existing video; Secondly, we transfer coordinates to a fixed coordinate system using the EvenVizion component; Then we perform the preprocessing to extract objects' metadata; Finally, we launch the algorithm.

6 POSSIBLE IMPROVEMENTS

We reckon that these improvements may be introduced in the upcoming versions of this software:

- Processing of the cases of long disappearance from the frame. This project does not take into account the long-term link. Here, identification is needed, but pointwise only.
- Complex types of movements processing with the Kalman filter replacement.

7 OUTRO

We can see how the demand for computer vision-based public safety solutions is experiencing an upsurge. The main tasks for such software are detection and tracking, while tracking appears to be harder to maintain due to difficult filming conditions. The AcurusTrack component is intended to beat this challenge and make objects multi-tracking as accurate as possible without relying solely on neural networks. We achieve it by using our custom data association model, probabilistic AI and a physical model of

realistic object movements (inherent to an object in the real world). Furthermore, our component is open-source, and you can easily find it on GitHub [1].

REFERENCES

[1] AIHunters. [n.d.]. *AcurusTrack*. https://github.com/AIHunters/AcurusTrack

[2] AIHunters. [n.d.]. *EvenVizion*. https://github.com/AIHunters/EvenVizion

[3] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. 2017. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*. 4836–4845.

[4] Jeremy Cohen. [n.d.]. . https://towardsdatascience.com/computer-vision-for-tracking-8220759eee85

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2017. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision*. 3038–3046.

[6] Kai Arras Wolfram Burgard Giorgio Grisetti, Cyrill Stachniss. [n.d.]. . http://ais.informatik.uni-freiburg.de/teaching/ws09/robotics2/pdfs/rob2-11-dataassociation.pdf

[7] Daniel Gordon, Ali Farhadi, and Dieter Fox. 2018. Re3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects. *IEEE Robotics and Automation Letters* 3, 2 (2018), 788–795.

[8] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. 2014. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence* 37, 3 (2014), 583–596.

[9] Oxagile. [n.d.]. *Oxagile*. https://www.oxagile.com/article/tracking-live-video-objects-with-a-moving-camera/

[10] Dhaval Salvi, Jarrell Waggoner, Andrew Temlyakov, and Song Wang. 2013. A graph-based algorithm for multi-target tracking with occlusion. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 489–496.

[11] Hyochoong Bang Youngjoo Kim. [n.d.]. . https://www.intechopen.com/books/introduction-and-implementations-of-the-kalman-filter/introduction-to-kalman-filter-and-its-applications

[12] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. 2012. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European Conference on Computer Vision*. Springer, 343–356.