A TOOL FOR FAST GROUND TRUTH GENERATION FOR OBJECT DETECTION AND TRACKING FROM VIDEO

Francesco Comaschi *Sander Stuijk* *Twan Basten* *Henk Corporaal*

Eindhoven University of Technology, Den Dolech 2, 5600 MB Eindhoven, the Netherlands
{f.comaschi, s.stuijk, a.a.basten, h.corporaal}@tue.nl

ABSTRACT

Object detection and tracking is one of the most important components in computer vision applications. To carefully evaluate the performance of detection and tracking algorithms, it is important to develop benchmark data sets. One of the most tedious and error-prone aspects when developing benchmarks, is the generation of the ground truth. This paper presents FAST-GT (FAst Semi-automatic Tool for Ground Truth generation), a new generic framework for the semi-automatic generation of ground truths. FAST-GT reduces the need for manual intervention thus speeding-up the ground-truthing process.

***Index Terms*—** Object detection, face detection, interactive systems, image databases

1. INTRODUCTION

In the context of detection and tracking systems, establishing common data sets of annotated videos is necessary for two reasons: i) enabling an accurate quantitative comparison between different systems; ii) finding the optimal parameter settings for a given system in different scenarios. The process of annotating datasets with the ideal results which an algorithm is expected to output is known as ground-truthing.

Even if many data sets are available [1–4], in many situations it is necessary to have a ground truth for ad-hoc sequences recorded under specific conditions at varying complexity levels. This is because numerous factors affect the performance of detection and tracking algorithms, such as illumination variation, occlusion and background clutter. However, annotating large volumes of video data is time-consuming and error-prone due to drops in user attention. Hence, we believe in the importance of automating the ground-truthing process.

Various tools for annotating data sets have been introduced [5–7]. One of the tools for ground truth annotation commonly used in literature is ViPER-GT (Video Performance Evaluation Resource Ground Truth), a video-truthing tool designed to allow frame-by-frame markup of videos [5]. ViPER offers two simple mechanisms to speedup the manual annotation task, i.e., propagation and interpolation. Propagation though can be helpful only every few frames, where a target is static in the sequence. On the other hand, for moving objects interpolation can be used only on a very limited sequence of frames if we want it to be accurate. As a consequence, in order to build a reliable ground truth, the user still has to manually annotate large parts of the video sequence frame by frame.

In [6], the authors introduce a modified version of ViPER-GT which automatically tracks the objects in the scene and allows the user to supervise the tracking process online. However, the generation of new objects is entirely left to the human operator and the proposed tool is limited to the single use-case of people tracking. In this paper we propose a framework which can be applied to different detection and tracking benchmarking problems.

In [7] the ground truth generation is achieved employing simple object detection and tracking algorithms. However, the proposed tool does not provide any confidence assessment mechanism of the detected and tracked objects, therefore the user still has to manually check for every frame whether to accept or refuse the suggested associations.

In our *FAst Semiautomatic Tool for Ground Truth generation* (FAST-GT), we integrate a detection module together with a tracking and a scoring module in order to reduce the need for manual intervention. FAST-GT has been made freely available to allow the detection and tracking community to rapidly generate benchmarks[1].

The main contributions of this paper are:

- A generic framework for the semiautomatic generation of ground truths for object detection and tracking benchmarking (Sec. 2). FAST-GT reduces the need for manual intervention thus speeding-up the ground-truthing process. It can be interfaced with ViPER for later visualization and validation.
- An instance of the proposed framework, in the form of a semiautomatic ground truth generator for face detection and tracking benchmarking (Sec. 3).

In Sec. 4, experimental results show that the use of FAST-GT in combination with ViPER allows to speedup the ground-truthing process for a challenging video sequence by reducing the need for manual intervention. Sec. 5 concludes the paper.

2. SEMI-AUTOMATIC GROUND-TRUTHING FRAMEWORK

Our aim is to build a framework that semi-automatically generates ground truths for object detection and tracking algorithm benchmarking. To reach our aim, we propose the structure shown in Fig. 1. To automate the annotation process, a

[1] http://www.es.ele.tue.nl/video/

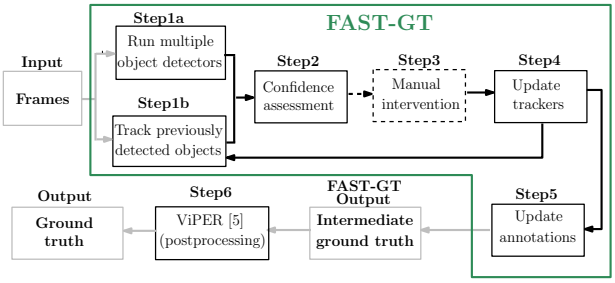

Fig. 1. Semi-automatic ground-truthing framework.

reliable detection and tracking mechanism has to be inserted in the tool. Therefore, in step 1 of FAST-GT, the current frame of the input video sequence is scanned in parallel by multiple detectors while all the objects already present in the previous frame are tracked. Each of the detectors produces as output a set of detected system targets, while the tracker produces a set of tracked objects. The use of multiple detectors allows to compare their output and to implement a confidence assessment mechanism which makes the final annotation output more reliable. Any object detector can be called in step 1a, as long as it provides a scoring mechanism, which is used by our tool to assess the confidence of the detected system targets. Similar to the object detection in step 1a, different tracking algorithms can be used in step 1b. In step 2, a confidence assessment mechanism allows to assign a proper confidence level to each of the objects detected and tracked in the previous step. In particular, objects can be assigned to one of the following output sets: (1) O_h: objects with a high confidence level are directly added to the ground truth without manual intervention; (2) O_l: outputs with low confidence level are directly discarded by the system as false positives; (3) O_m: outputs with an intermediate confidence level are presented to the human annotator for human-in-the-loop feedback. Since in critical situations human operators generally make better decisions than automated approaches [8], whenever the detection and tracking system produces outputs with an intermediate confidence level ($O_m \neq \emptyset$), in step 3 of FAST-GT we invoke the user supervision to manually correct possible errors produced by the automated annotation process. A GUI allows the user to rapidly validate and correct the system output. In step 4, FAST-GT checks for trackers to be updated. This step is required because trackers drift over time, and the detection outputs and the manual corrections can be used to reinitialize the trackers. In step 5, the ground truth is updated with the results of the previous steps. If the end of the sequence has been reached, the result is output to a ViPER-compatible XML file, otherwise the whole procedure starts again for the next frame. In step 6, the intermediate ground truth produced by FAST-GT is provided as input to ViPER, allowing the user to visualize and manually validate the tool output and to create the final ground truth.

One of the main features we aimed at when building FAST-GT was flexibility. The proposed tool has a modular structure where the different steps in the system have been developed as APIs. In this way, researchers can use different functions to implement the steps of FAST-GT.

3. GROUND TRUTH GENERATION FOR FACE DETECTION AND TRACKING FROM VIDEO

This section describes a specific instance of FAST-GT, i.e., a semi-automatic ground truth generator for face detection and tracking benchmarking. Given a video, the tool detects and tracks the faces in it and creates the corresponding ground truth in a ViPER-compatible XML format. For the annotation, we have followed the most widely adopted procedure [9–11], i.e., our tool annotates face regions with rectangular bounding boxes aligned with the image axes. For some of the faces it is very difficult to determine whether they are visible or not, for instance because they are partially occluded or their viewed angle is larger than 90 degrees. In order to establish clear guidelines, we have considered a face as visible, and therefore to be included in the ground truth, when: i) either side of the bounding box is at least 30 pixels; ii) the three main facial features (left-eye, right-eye, mouth) are visible. Integers are used as face IDs. Following [11], if a face disappears and returns later, a new ID is assigned. In step 1a of our FAST-GT instance, three frontal face Haar-detectors are run in parallel. This choice is motivated by the proved robustness of sliding-window based detection algorithms [12–15]. Different features can be used to detect objects, such as histograms of oriented gradients (HOG) [13] or covariance region descriptors [16]. Haar-features prove to be very effective when it comes to faces [12, 17]. In step 1b, the faces already present in the scene are tracked by the system. An approach to tracking which has become particularly popular recently is tracking-by-detection [18], which treats the tracking problem as a detection task applied over time. A recent benchmark on the latest advances of tracking-by-detection [10] reported the Struck algorithm [19] to be one of the most accurate. Therefore, we have selected the Struck algorithm for our system.

In step 2, we first compute the union as well as the intersection of all the detector outputs. We also compute the union of all the pairwise intersection sets of the detection outputs and the set given by the intersection between the tracker output and the union of all the detector outputs. In order to compute the intersection or the union of different sets, a matching criterion is needed to determine whether two different bounding boxes from different sets are equivalent (i.e., cover the same object). In our case, we consider two bounding boxes to be equivalent if the overlapping area of the rectangles, measured as a fraction of their union area, was greater than 0.5.

In Fig. 2 we provide a graphical representation of the confidence assessment criteria used in our FAST-GT instance. We first consider detected objects (Fig. 2 (a)). Green bounding boxes are the output of the detectors while purple boxes are the output of the tracker. We take the intersection between a detected object and a tracked object (case 1) as a high-confidence indicator, and since detection is most often more accurate than tracking, whenever a match occurs, the detector's output is assigned to O_h and used to reinitialize the corresponding tracker in step 4. Then, we consider objects for which there is agreement by all the detectors, but that have not been tracked (case 2), to be new objects in the scene, and we

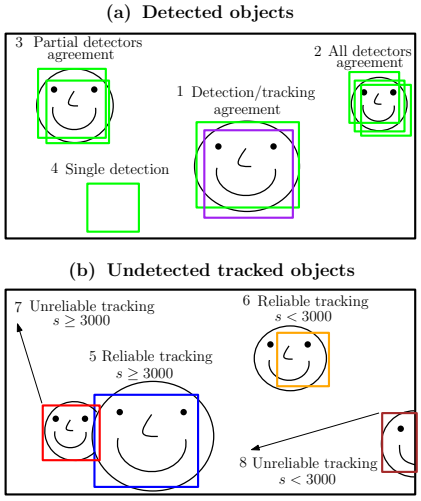

Fig. 2. Confidence assessment of the detectors and tracker output.

assign them also to O_h. The objects that have not been tracked but detected by at least two detectors (case 3) are assigned to O_m for manual feedback. Elements that have not been tracked but are detected by only one of the detectors (case 4) are considered false positives and assigned to O_l. After the matching of the detected and the tracked targets, we take the set of undetected tracked objects (Fig. 2, (b)), and split it into two disjoint subsets: reliable tracked objects (blue and orange boxes) and unreliable tracked objects (red and brown boxes). In our case, we implemented a simple function to determine if any of the unmatched tracked objects are either occluded by bigger bounding boxes, or close to the boundary of the scene. If one of these conditions is met, then the tracked object is considered unreliable, otherwise it is considered reliable. In this way we can more easily prevent the system from erroneously tracking objects which are either going out of scope or becoming completely occluded. We note here that better occlusion-detection mechanisms can be implemented and integrated in FAST-GT in order to generate the ground truth for video sequences characterized by different kinds of occlusions. For all the unmatched tracked objects the corresponding detection score s is considered, provided by step 1a. The detection score for a candidate window can be defined in different ways according to the detection algorithm implemented. In our implementation, we compute the detection score of the most accurate Haar-feature detector [20] as computed in [21]. If the object is a reliable tracking and the detection score is above an empirically determined threshold (case 5), we consider the tracked object to be a valid face missed by the detection module, and we therefore assign it to O_h. If the tracked object is reliable but the detection score is below the threshold (case 6), it most probably means that the tracker is drifting from the target, and we present the frame to the user for manual reinitialization. If the tracked object is unreliable but the detection score is high (case 7), it most often implies that the object is partially occluded or out of scope, and also in this case we ask for manual intervention for a more accurate evaluation. Finally, if the tracked object is unreliable and the score is low (case 8), the object is most often completely occluded or out of scope, and therefore is assigned to O_l (i.e., the object is considered to be invisible and therefore no user intervention is requested). In step 3, if $O_m \neq \emptyset$, the frame is reported to the user for manual intervention. A GUI has been developed to facilitate the manual feedback. Objects (boxes) are presented in different colors according to the confidence level, and the user can correct errors by: i) removing boxes; ii) drawing new boxes; iii) moving/resizing existing boxes; iv) correcting IDs. In step 4, trackers are possibly reinitialized by new detections or manual interventions, or removed if belonging to O_l. In step 5 the ground truth is updated with the annotations from the current frame. The intermediate ground truth produced by FAST-GT is postprocessed through ViPER in step 6 for manual validation.

4. EXPERIMENTAL RESULTS

In order to prove the effectiveness of FAST-GT even under challenging real-world scenarios, we applied it to a video sequence from the Chokepoint dataset [4] where many people are moving in the scene. No ground truth was previously available for the selected video sequence. We further applied it to three simpler video sequences from the selected dataset and compared the output of our tool with the ground truths produced by the dataset authors. Video sequence P2E_S5_C1.1 from [4] contains 807 frames of 800x600 pixels acquired at 30 fps, with a total of 2316 targets. To test FAST-GT, we generated the ground truth for the selected video sequence in three different operational modes: (1) manually annotating the entire video sequence through VIPER-GT; (2) running FAST-GT without the human-in-the-loop feedback mechanism (step 3 from Sec. 2) and then using ViPER-GT as a validation post-processing step; (3) exploiting the full functionality of FAST-GT, including step 3, and then using ViPER-GT as a validation post-processing step.

After running FAST-GT, we use ViPER-GT to visualize the results and correct possible errors. We distinguish between four types of errors and the respective editing operations required: (i) False Negative (FN) / box creation: a truth target is missed by FAST-GT; a new box needs to be created; (ii) False Positive (FP) / box removal: FAST-GT produces a bounding box which is not associated with any real truth target; the corresponding box needs to be removed; (iii) Deviation / box correction: the box generated by FAST-GT deviates from the real target position; the corresponding box needs to be moved/resized; (iv) Fragmentation / ID correction: a truth target visible for a set of consecutive frames in the sequence is associated to different IDs; the IDs need to be corrected.

When running FAST-GT with the inclusion of the manual intervention step, the user's intervention is automatically invoked for the frames of the sequence containing targets with an intermediate confidence level; therefore we have to distinguish between errors corrected online (step 3), and errors corrected through ViPER during postprocessing (step 6).

In Tab. 1 we compare the performance when generating the ground truth for P2E_S5_C1.1 in the three operational

Table 1. Comparison between FAST-GT with and without manual intervention (step 3) and ViPER.

Mode	#Boxes created	#Boxes removed	#Boxes corrected	#IDs corrected	#Total editing ops
Manual (ViPER)	2316	0	0	0	2316
FAST-GT w/o step 3 corrections in step 6	241	52	30	8	331
FAST-GT w/ step 3 corrections in step 3	3	38	9	0	50
FAST-GT w/ step 3 corrections in step 6	44	35	43	0	122

modes listed earlier in this section. We can see that the fully automated tool (line 2) makes few mistakes in terms of FPs (boxes removed) and deviation errors (boxes corrected). The lack of human feedback leads to a high number of FNs though (boxes created). This is because whenever the tool is uncertain about any output, it discards this output. Also, whenever the tool loses track of a target and then detects it in a later frame, a new ID is assigned, causing fragmentation errors (IDs corrected). From lines 3 and 4 of Tab. 1, we can see that the on-line manual validation allows to drastically reduce the number of errors in terms of FNs (boxes created) and that no fragmentation errors occurred (IDs corrected). This is because every time the tool is losing track of a target, the tool detects the error and present the current frame to the user to prevent the propagation of the error to the following frames. In conclusion, running FAST-GT without step 3 required 331 manual editing operations, while the inclusion of step 3 reduced the required operations to 172. We can conclude that the manual intervention step is useful. Manually annotating the entire video sequence through ViPER required the drawing of all the bounding boxes one by one, thus leading to 2316 editing operations. From Tab. 1, we can see that by using FAST-GT in combination with ViPER we could reach an almost 7x reduction in the manual editing operations in the fully automated mode and more than a 13x reduction when including the manual intervention step.

To give an indication in terms of time, annotating the video manually with ViPER takes almost 4 hours; with FAST-GT without step 3 it takes almost 1 hour and with step 3 about half a hour. These numbers are just an illustrative example, since they depend on the user experience and on the specific video sequence. They indicate though that substantial time savings can be obtained with FAST-GT.

Figure 3 shows a screen capture of the GUI presented to the user for manual intervention. The bounding boxes have different colors according to the set they belong to. The green bounding boxes belong to O_h, and in particular they are outputs produced by successful detections (cases 1 and 2 from Fig. 2, (a)). The blue bounding box still belongs to O_h, but it is the result of a successful tracking with no detection (case 5 from Fig. 2, (b)). The respective detection score is reported in yellow. The orange bounding box corresponds to a reliable tracked object with low score (case 6 from Fig. 2, (b)), assigned to O_m. This triggers step 3 of the tool, presenting the window to the user who can correct the corresponding error

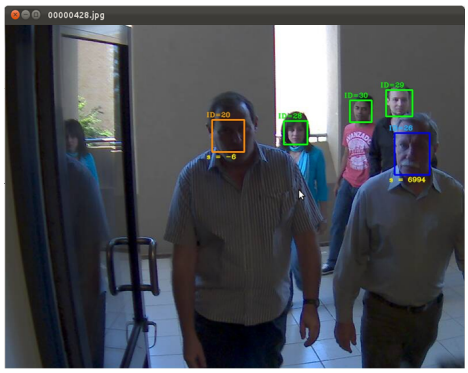

Fig. 3. Screen capture of the on-line manual intervention.

before the tool moves to the next frame.

To prove the effectiveness of the proposed tool even further, we generated the ground truths for three simpler video sequences from [4] (P2E_S1_C2.2, P2E_S2_C2.2, P2E_S3_C2.2,) showing only one person at a time moving in the scene. We compared the output produced by FAST-GT without postprocessing (step 6) but with step 3 and the ground truths created by the dataset authors[2]. FAST-GT made 177 errors (152 FPs, 25 FNs, 0 deviation errors and 0 fragmentation errors) for a total of 1009 faces. These errors can be easily corrected through postprocessing with ViPER. Many of the FPs are the actual faces when they enter the scene. They are then typically very small and as such not indicated as faces by the dataset authors. Detectors do detect those small faces though, so the dataset authors may have made the ground truth too conservative. In total, 44 online corrections were needed while processing the sequences.

5. CONCLUSIONS

We have introduced FAST-GT, a new tool for the semi-automatic generation of ground truths for object detection and tracking benchmarking. FAST-GT has been developed as a generic framework that allows different implementations of the building blocks. A specific instance of FAST-GT was implemented and successfully applied to generate the ground truth for a video sequence containing multiple faces moving in the scene. Empirical results show the ability of FAST-GT to notably decrease the annotation time. FAST-GT has been made freely available on the website http://www.es.ele.tue.nl/video/ to make it easier for the detection and tracking community to rapidly generate benchmarks.

Acknowledgment: This work was supported in part by the COMMIT program under the SenSafety project.

[2]We note here that the ground truths provided by the dataset authors report the position of the eyes rather than the position and size of the faces. Therefore we consider the ground truths to match whenever the eyes are contained in the bounding box generated by our tool.

6. REFERENCES

[1] R.B. Fisher, "The PETS04 surveillance ground-truth data sets," in *PETS*, 2004.

[2] R.T. Collins, X. Zhou, and S.K. Teh, "An open source tracking testbed and evaluation website," in *PETS*, 2005.

[3] S. Ayache and G. Quénot, "Video corpus annotation using active learning," in *ECIR*, 2008, pp. 187–198.

[4] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B.C. Lovell, "Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition," in *CVPR Workshops*, 2011.

[5] "ViPER: The video performance evaluation resource," 2013, http://viper-toolkit.sourceforge.net/.

[6] M.A. Serrano, J. García, M.A. Patricio, and J.M. Molina, "Interactive video annotation tool," in *DCAI*, 2010, pp. 325–332.

[7] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato, "A semi-automatic tool for detection and tracking ground truth generation in videos," in *VIGTA*, 2012, pp. 6:1–6:5.

[8] U. Vural and Y.S. Akgul, "Operator attention based video surveillance," in *ICCV Workshops*, 2011, pp. 1955–1962.

[9] I. Leichter and E. Krupka, "Monotonicity and error type differentiability in performance measures for target detection and tracking in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2553–2560, 2013.

[10] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[11] M. Fischer, M. Bauml, H.K. Ekenel, and R. Stiefelhagen, "Benchmarking face tracking," Tech. Rep., Karlsruhe Institute of Technology, 2013.

[12] P.A. Viola and M.J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.

[14] R.G.J. Wijnhoven and P.H.N. de With, "Fast training of object detection using stochastic gradient descent," in *ICPR*, 2010, pp. 424–427.

[15] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal, "RASW:a run-time adaptive sliding window to improve viola-jones object detection," in *ICDSC*, 2013.

[16] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *ECCV (2)*, 2006, pp. 589–600.

[17] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *BMVC*, 2006, pp. 47–56.

[18] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, 2004.

[19] S. Hare, A. Saffari, and P.H.S. Torr, "Struck: Structured output tracking with kernels," in *ICCV*, 2011, pp. 263–270.

[20] M.C. Santana, O. Déniz-Suárez, D. Hernández-Sosa, and J. Lorenzo, "A comparison of face and facial feature detectors based on the viola-jones general object detection framework," *Mach. Vis. Appl.*, vol. 22, no. 3, pp. 481–494, 2011.

[21] V. Jain and E.G. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Tech. Rep., UMass Amherst, 2010, http://vis-www.cs.umass.edu/fddb/faq.html.