Efficient video annotation with visual interpolation and frame selection guidance

Alina Kuznetsova Aakrati Talati Yiwen Luo Keith Simmons Vittorio Ferrari

Google Research

{akuznetsa,aakrati,yiwenluo,eskiman,vittoferrari}@google.com

Abstract

We introduce a unified framework for generic video annotation with bounding boxes. Video annotation is a long-standing problem, as it is a tedious and time-consuming process. We tackle two important challenges of video annotation: (1) automatic temporal interpolation and extrapolation of bounding boxes provided by a human annotator on a subset of all frames, and (2) automatic selection of frames to annotate manually. Our contribution is two-fold: first, we propose a model that has both interpolating and extrapolating capabilities; second, we propose a guiding mechanism that sequentially generates suggestions for what frame to annotate next, based on the annotations made previously. We extensively evaluate our approach on several challenging datasets in simulation and demonstrate a reduction in terms of the number of manual bounding boxes drawn by 60% *over linear interpolation and by* 35% *over an off-the-shelf tracker. Moreover, we also show* 10% *annotation time improvement over a state-of-the-art method for video annotation with bounding boxes [25]. Finally, we run human annotation experiments and provide extensive analysis of the results, showing that our approach reduces actual measured annotation time by* 50% *compared to commonly used linear interpolation.*

1. Introduction

Progress in machine learning techniques depends on the availability of large volumes of high quality annotated data. Recently several large scale image datasets have appeared [19, 35, 9], as well as large-scale tracking benchmarks [13, 5], but they required tremendous annotation resources to create [19, 41]. The reported annotation time for box annotation ranges between 5.2 [25] and 20 [33] seconds per bounding box. Hence, the time to create a dataset of similar size to Got10k [13] requires about 3000 - 8000 hours of work just for the box annotation stage (provided each box is annotated individually). Due to this high cost, none of the existing large-scale video benchmarks provides exhaustive annotations, not even at the video clip level. Going beyond bounding boxes, video instance segmentation datasets are even smaller [45, 28]. Being able to easily develop such datasets would speed up the progress in unconstrained video understanding [8, 13].

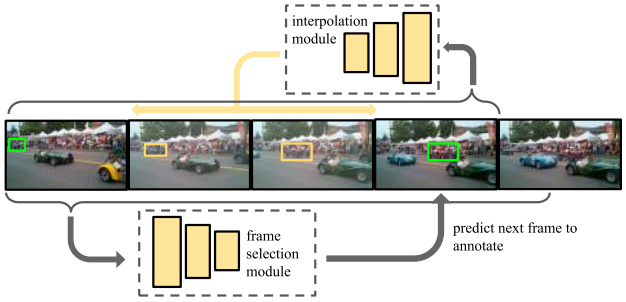

Figure 1: Overview of our video annotation process. A human annotator draws a box on the first frame of the video; then our guiding frame selection mechanism predicts the next frame to annotate and the process iterates. Our method automatically and accurately interpolates bounding boxes for all frames that were not directly annotated by the human. Hence, at the end of the process object annotations are generated for all frames.

In this paper we propose an efficient video annotation. Our framework consists of two interacting modules: (1) a module for interpolation and extrapolation of annotations created by a human annotator (we call it visual interpolation below for simplicity) and (2) a guiding mechanism that selects which frame to annotate.

During the annotation process, a human annotator starts by annotating the object in a single frame. The guiding mechanism produces a prediction for which frame to annotate next and the visual interpolation module propagates the annotation to other frames. Note, that unlike traditional active learning approaches [40, 38] the guiding mechanism produces frame proposals in a sequential manner and per track. See Fig 1 for an overview of the process.

Single-object tracking techniques made big progress in recent years [16]. In particular siamese trackers [1, 22, 43] showed excellent results on tracking benchmarks. Moreover, those models offer real-time performance, making

them suitable for an interactive annotation process. However those techniques are underexplored for annotation purposes. One reason is the lack of a track correction mechanism that would allow to efficiently correct the output of the tracker. Here we propose to alleviate this drawback by extending a siamese tracker to enable corrections and to take advantage of ground-truth annotations in multiple frames, which become available during the annotation process.

Our guiding mechanism is based on the observation that not all frames are equally useful for annotation. For example, a frame where an object is heavily occluded is unlikely to allow the visual interpolation module to propagate well to other frames. Hence, we propose to rank unannotated frames based on the expected quality of annotations generated by our visual interpolation module if those frames would be selected for annotation. The ranking is based on pairwise comparisons of the candidate unannotated frames. In this fashion, our two proposed modules interact and are part of an integrated system.

In summary, we propose: (1) a visual interpolation module that adapts existing trackers to the annotation scenario; (2) a guiding module that automatically selects frames to send for annotation; (3) an integrated framework where both modules work smoothly together. We highlight that the proposed framework allows a real interactive annotation process, as it does not require offline pre- or post-processing.

We provide extensive experimental ablation studies on the ImageNetVID dataset [34]. We compare our approach to the traditionally used linear interpolation and forward tracking using the same base siamese model. Our approach reduces by 60% the number of manually drawn boxes compared to linear interpolation, and by 35% compared to tracking at a fixed quality (80% of all frames annotated at IoU > 0.7). Next, we perform experiments with real human annotators on the Got10k [13] dataset and show that our framework allows to reduce actual annotation time by 50% compared to annotation time when using linear interpolation. Finally, we show that our framework is efficient for annotation of the challenging multi-object tracking dataset MOT2015 [20]. We show 10% time reduction compared to the state-of-the-art framework [25] at the same level of the annotation quality.

2. Related Work

Video datasets. Creating video datasets with detailed localized annotations is very time-consuming and hence large-scale datasets are rare. Recently several object tracking datasets have been proposed [27, 13, 5, 37]. While offering object diversity, they however do not contain annotations for more than a single object track per video.[1] Currently only the Waymo Open Dataset [36] contains exhaustive annotations for all object tracks in each video. However, that dataset focuses on driving scenes and therefore has limited number of annotated classes. The place for a large scale general purpose video dataset is still vacant and efficient video annotation methods are required to create those.

Video annotation. Early works on video annotation propose to speed up annotation process using geometric interpolation of annotated bounding boxes and polygons [42] across frames. Employing video content to assist bounding boxes for video annotation was investigated in [41], where the authors interpolate annotations by solving a dynamic programming problem after each new bounding box provided by a human annotator. Several published approaches [15, 44] for segmentation propagation are not directly targeting the video annotation use-case and do not allow for online corrections. More recent work [3] proposes a solution for interactive video object segmentation annotation problem: they first obtain bounding boxes of the objects by forward tracking and subsequent curve fitting, and employ SiamMask [43] and scribbles to derive segmentation from box tracks. However, the initial problem of bounding box annotations remains not well studied. [13] mentions using tracking to propagate bounding boxes between manual annotations without any further details.

A separate line of works explores training models with a small set of sparse manually annotated bounding boxes and large set of automatically labeled ones obtained via tracking [26, 18]. Those approaches, however, are model-specific and are not focusing on obtaining a large set of annotated data that could be re-used for training multiple models.

Finally, Pathtrack [25] proposes an approach in between the semi-supervised approaches mentioned above and manual labelling approaches like [42], specifically tackling annotation of crowded videos. Annotators first track the center of each person with a mouse pointer through the video. Those point tracks are used to build full bounding box tracks by integrating automatic detections from a person detector.

One of the advantages of the our method over previous work is that it operates in real-time and does not require any offline pre- or post- processing. Once the infrastructure is set up, live annotation can be run immediately on new videos.

Single-object tracking. Single-object tracking is a long-standing computer vision problem. The first few successful approaches [10, 4, 14] relied on hand-crafted features. Recently, trackers based on deep-learned architectures [1, 23, 43, 47, 46, 11, 2] emerged in this area. Trackers based on Siamese architectures [1, 23, 48, 43] are particularly interesting, as they showed strong results on various benchmarks and are relatively simple. In our work we extend the basic model of [1, 48] to form our visual interpola-

[1] [37] dataset does offer 13 videos out of 185 that contain $2-3$ objects.

tion module.

Active learning and other related works. It was noticed [40, 39] that one of the factors slowing down the annotation process is selecting frames for manual annotation and so some works explored the problem of optimal frames selection (both for video segmentation [39] and bounding box annotation [40]). However, those approaches require expensive pre-processing of all frames or online retraining of the propagation algorithm during the annotation process. Further, the annotators have to spend time on context switching, since frames are not presented chronologically [25]. Instead, our proposed method selects frames chronologically.

Another work related to ours is BubbleNets [7], in the domain of video instance segmentation. The task is to automatically segment an object in every frame of a video, given the ground-truth segmentation in one particular frame. The authors show that the quality produced by a segmentation model heavily depends on which frame is given with ground-truth segmentation (which is used for fine-tuning the model). We extend their results by investigating a more complex setting: bounding box annotation for challenging datasets containing multiple objects per frame, as opposed to focusing on a single main object per frame. To achieve that we introduce an attention mechanism that allows the model to focus on a specific object (Sec 3.2). .

Finally, different from general active learning, we do not focus on training the best quality models, but rather on annotating data in the most efficient way. This data can then be used to train any model (also beyond the particular tracker used to assist during annotation). Our framework also does not assume any online training, which makes it more suitable for the specific scenario of interactive real-time video annotation.

3. Video annotation framework

Our overall framework is presented in Fig 1. It consists of two components: the visual interpolation module and the frame selection guiding module. The annotation process alternates between two steps: the human annotator drawing a bounding box in one frame and the machine carrying out the box interpolation/extrapolation and selecting the next frame to annotate. As we show experimentally, such human-machine collaboration is very beneficial as it reduces the total human annotation time (see Sec 4.2).

3.1. Visual interpolation

Video annotation is a time-consuming and tedious process [41]. Existing approaches use linear interpolation of box geometry [42] or more complicated geometric modeling [6] that nevertheless does not rely on visual signals. On the other end of the spectrum are the approaches relying on visual signal only [41]. However, recent developments in single-object tracking are so far under-explored for the task of video annotation, perhaps because trackers typically assume a single target object appearance as input and do not allow any corrections after the tracking started. To this end we propose a set of interpolation models that are based on contemporary trackers. Our model exploit visual information from multiple annotated frames at the same time, and allow to introduce and propagate corrections during the annotation process.

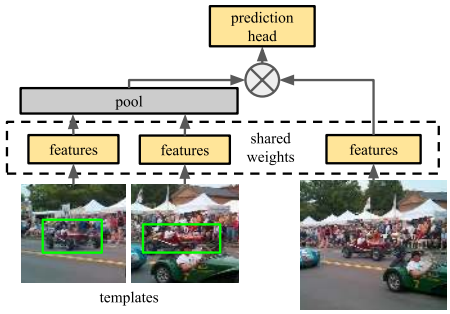

Figure 2: Visual interpolation model: features are first extracted from multiple templates and a joint feature vector is formed by maxpooling. The joint features are then used to derive a final prediction for an unannotated frame by convolving them with features extracted from the search space for that frame.

Many state-of-the-art single object trackers rely on siamese architecture [1, 23, 43, 48], where a single backbone is used to extract the features from the annotated frame and the subsequent video frames to combine those features in various ways to localized the target object. We propose a simple change to siamese architectures to incorporate tracking target appearance in multiple annotated frames. This extends siamese type trackers to interpolation and allows efficient track correction mechanism. In the subsequent sections we explain the proposed modification on the example of two models, SiamFC[1] and DaSiamRPN [48], and in the experimental section we demonstrate that it brings significant performance improvements.

Siamese tracking models. The Siamese tracker model consists of two feature extractor branches with shared weights $\varphi(\cdot)$. One of the branches extracts features from the image patch containing the tracking target z in the initial frame, defined by a manually annotated bounding box (we call this patch *template*). The other branch receives an image patch from the current frame x (we call this patch *search space*). The features extracted from the template $\varphi(z)$ are convolved with the the search space features $\varphi(x)$ to derive the score map (in case of SiamFC) or box prediction and tracker score (in case of DaSiamRPN):

$$A(z, x) = \varphi(z) * \varphi(x), \tag{1}$$

where $*$ denotes convolution.

During tracking, the template is obtained by cropping an image around the initial ground truth bounding box with

equal width and height of $\sqrt{(w+2p)(h+2p)}$, centered around the box center and re-scaled to 127×127 pixels (here w, h are width and height of the initial box and $p = (w+h)/4$). The search space image patch is obtained by cropping a large square patch around the current position of the target. The search space crops are computed at multiple scales for the SiamFC tracker and for a single scale for DaSiamRPN tracker.

Visual interpolation network. Provided ground truth annotations for the same object in multiple frames, we investigate a modification of the base siamese network to incorporate the additional visual information coming from them (Fig 2). Let $\{z_i\}_{i=1}^{K}$ be several templates obtained for the same target in multiple frames (we call them *keyframes*). The model consists of $K+1$ feature extractors with shared weights; the features are combined by max-pooling $g(\cdot)$ as in [29]. Afterwards, max-pooled features are convolved with the search space features as in the base model:

$$A(z_1, \ldots, z_K, x) = g(\varphi(z_1), \ldots, \varphi(z_K)) * \varphi(x). \quad (2)$$

Note, that this architecture is able to take into account arbitrary number of templates both at train and test time, potentially improving performance.

Geometric model. Geometric modelling for annotation propagation has an advantage over visual methods as it is robust against occlusions and bad image quality (such as blur and video decoding artifacts). Hence it is more reliable in the vicinity of the frames that contain annotations.

To benefit from it, we blend the prediction of the visual interpolator model with a geometric interpolation model at each frame. Geometric model prediction is more reliable in a temporal neighborhood of the keyframes and less reliable further away in time. Visual interpolation generally works better for such temporally distant frames, as it follows the object visually. To model this we introduce weight $w(\delta_t, \Delta)$, where δ_t is (absolute) offset in time to the closest keyframe and Δ is a parameter. The higher the weight $w(\delta_t, \Delta)$, the closer the overall process is to geometric interpolation model output:

$$w(\delta_t, \Delta) = \begin{cases} 0, & \delta_t > \Delta \\ \delta_t^2 \Delta^{-2} - 2\delta_t \Delta^{-1} + 1, & \delta_t \leq \Delta \end{cases} \quad (3)$$

As a geometric interpolation model we use linear interpolation between boxes in two frames. The dimensions of a box and its center position are interpolated separately. Outside of the temporal neighborhood $(-\Delta, \Delta)$ of an annotated frame geometric interpolation has no effect.

Training. We train SiamFC visual interpolation model using the train set of ImageNet VID [34] for 10 epochs with batch size 32 and using momentum optimizer [30] with initial learning rate of $1e-3$ and exponential decay. For DaSiamRPN we use ImageNet VID [34], YouTube Bounding Boxes [31] and MSCOCO [24] for traning as proposed in [48] and using the same parameters as for SiamFC visual interpolation training.

Moreover, instead of the original AlexNet backbone we use MobileNetV3 [12] backbone (as it delivers better performance). Since MobileNetV3 is not fully convolutional we extensively use data augmentation in training, as described in [21].

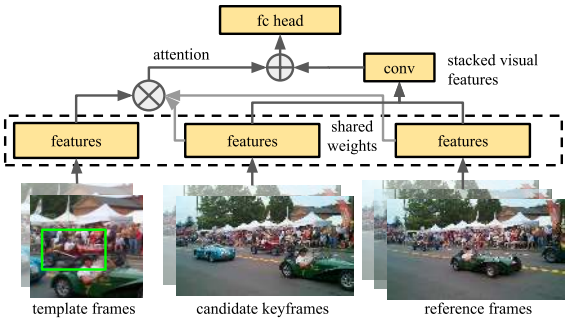

Figure 3: The ranking model architecture. There are three types of input: (1) templates obtained from previously annotated frames (cropped); (2) two candidate keyframes; (3) the video representation as N reference frames randomly subsampled from the video. We build an attention map on the target object by convolving the template features with the full frame features (of either the candidate keyframes or the reference frames). Then we add this attention maps to the visual features extracted from the full frames.

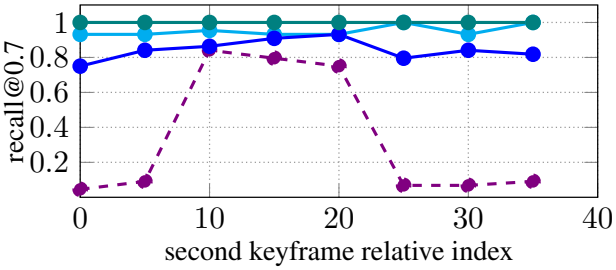

Figure 4: recall@0.7[2] of the visual interpolation vs the second keyframes selected for annotation for 4 objects in *TUD-Stadtmitte* video (the first keyframe is fixed, and marked as frame 0 for simplicity). Notice that for each object a different frame should be annotated to maximize annotation quality for its track.

3.2. Frame selection guidance

As mentioned in Sec 1 and confirmed by experiments in Sec 4.2, one of the major slow-downs for the annotation process is suboptimal selection of the frames to be manually annotated (*keyframes*). In Fig 4 we show that the quality of the visual interpolation model predictions clearly depends on the subset of keyframes manually annotated. To analyze

[2] recall@0.7 is computed as a fraction of generated bounding boxes that have intersection-over-union(IoU) with the groundtruth higher than 0.7

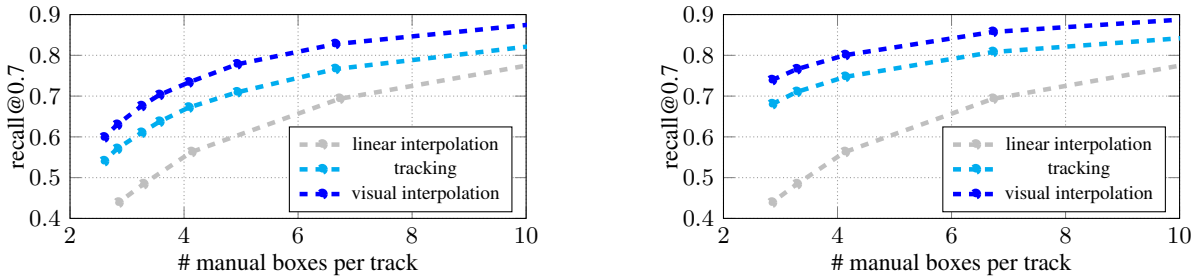

Figure 5: Performance of the linear interpolation, tracking and visual interpolation models ($K = 2$) at recall@0.7 for SiameseFC and DaSiamRPN models. Interpolation models have a clear advantage over the base tracker model.

this, we select a video clip containing 4 different objects and investigate the quality of annotation for each object depending on the selected second keyframe (the first keyframe is the same for all objects). For each object the optimal second keyframe is different and it has large impact on the annotation quality (depending on the object, quality increases by up to $+70\%$ when selecting the optimal keyframe, compared to the worst keyframe).

We propose here to optimize the annotation process by introducing an automatic frame selection mechanism. Given already existing annotations of an object in some previous frames, we want to select the *next keyframe* that would maximize the quality of the annotations produced by our visual interpolation module in the unannotated portion of the video. In this way we avoid the need to jump back and forth across the timeline, which can confuse the annotator and requires expensive context switching [25, 40].

In [7] the authors proposed an architecture to select a single best frame to propagate a segmentation mask to the whole video sequence. However, their approach operates on the full frames and therefore lacks an important element — conditioning on a specific target object. We extend their approach by introducing an attention mechanism to condition the model predictions on the object to be annotated.

Method overview. Our method works as follows. First, we sample candidate keyframes uniformly in an interval of 100 frames after all previously annotated frames. Then, we rank these candidate keyframes by expected annotation quality. At the core of our approach we train a ranking model that operates on pairs of candidate keyframes. It predicts a score indicating which of the two candidates is better, conditioned on the appearance of a specific target object, as captured by bounding boxes in previously annotated frames. The ranking model also takes into account the unannotated video content. The final score for each candidate keyframe is calculated as the sum over all pairwise scores. The single top-scoring candidate is selected as the next keyframe. The annotator then manually draws the object bounding box on this keyframe, and the process iterates.

Ranking model architecture. Fig 3 illustrates the architecture of our model. It takes three kinds of input: (1) a pair of candidate keyframes; (2) a set of N *reference frames* randomly sampled from the unannotated part of the video, enabling to condition on the content of the video; and (3) $K-1$ frames cropped around the bounding box from previously annotated frames (*templates*), enabling to condition on previous annotations for this object.

We use a a fully convolutional feature extractor to extract features from the full candidate and reference frames ($\{f^j\}_{j=1}^{N+2}$) and the templates ($\{z^j\}_{j=1}^{K-1}$). We implement conditioning on templates by computing attention maps a_j. These are computed by cross-correlation ($*$) between template features and the respective video frame features ($g(\cdot)$ denotes max-pooling):

$$a_j = g(\varphi(z_1), \cdots, \varphi(z_{K-1})) * \varphi(f^j)$$

The attention maps help to ensure that the module is focusing on the relevant parts of the image (i.e. on the target object, whose appearance is captured by the template features). The final prediction for a pair of candidate keyframes is a single score computed by several fully convolutional layers ($F'(\cdot)$) operating on top of the extracted features and attention maps (the scores are normalized to $[-1, 1]$):

$$c = F'([a_1 + \varphi'(f_1), \ldots, a_{N+2} + \varphi'(f_{N+2})])$$

Quality score for a candidate keyframe. We run the ranking model for all pairs of candidate keyframes. The overall score of a candidate keyframe is computed as the sum of all positive comparison scores (i.e. for pairs where this candidate keyframe was better than the frame it was compared against). The candidate keyframes are then sorted by their overall scores and the highest-scoring one is selected as the next keyframe to be annotated.

Although the proposed approach is related to [7], it goes well beyond. Thanks to the newly introduced conditioning on the target object we are able to handle the more complex (and realistic) scenario where the prediction must be done not simply at the frame level but for a specific object (see Fig. 4). In Sec. 4.1 we show that conditioning is crucial for the performance of the ranking model.

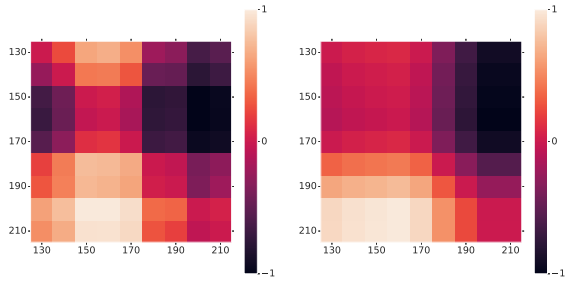

Figure 6: Frame comparison matrix — vertical and horizontal axis represent frame offsets from the already annotated frames; values at cell i, j represent relative annotation quality after annotating frame i or frame j: left — model prediction; right — groundtruth.

Training. The ranking model is trained in a supervised manner. To obtain training labels, we: (1) randomly sample previously annotated frames (templates) and pairs of candidate keyframes; (2) run the visual interpolation model for each candidate keyframe in a pair, and then evaluate its predictions over a 100 frame interval against ground-truth bounding boxes. The difference between the visual interpolation predictions quality (recall@0.7) of the two candidates is used as binary label for training the ranking model.

To reduce noise in the training data, we only consider tracks of objects larger than 5% of the frame area. Moreover, for a given template we sample multiple pairs of candidate keyframes such that there is a significant difference in the quality of the visual interpolation predictions they lead to (empirically set to > 0.3]).

The model is trained with binary cross entropy loss. We employ a feature extractor similar to AlexNet [17], described in [1]. The ranking model is trained for 10 epochs using momentum optimizer [30] with $1e-3$ initial leaning rate and batch size 12. In general we observed better training stability with larger batch size, which confirms findings by [32] that larger batch sizes improve training on noisy labels.

4. Experimental results

First, we evaluate the performance of our framework on the ImageNet VID validation set [34] (Sec. 4.1). Second, we evaluate the proposed framework by running annotation process with human annotators on Got10k validation set [13] (Sec. 4.2) and analysing results of human annotator experiments vs. simulation predictions. Finally, we compare the proposed method with state-of-the-art approaches [25, 42, 41, 40] on MOT2015 dataset [20] and demonstrate generalization across datasets (Sec. 4.3).

4.1. Performance of the framework components

ImageNet VID [34] is a middle-scale video object tracking dataset with dense trajectory annotations. The training set contains 3862 videos and objects of 30 classes. On average, each video contains 2.35 object tracks (with maximum of 47) and the average object size is 16% of the image area. We evaluate on the validation set, which contains 555 videos.

Model name	all	no small obj
no attention	0.51	0.51
no vis. features	0.56	0.61
full model	**0.63**	**0.68**

Table 1: Ranking model accuracy: model with *no attention* uses visual features only; *no vis. features* model only uses attention maps; *full model* is the full model as in Section 3.2; the *no small obj* column reports accuracy for objects with area $> 15\%$ of the image.

Results for visual interpolation. We show that our proposed extension of the tracker models (Sec. 3.1 is applicable to several contemporary deep tracker architectures and consistently increases model performance compared to tracking). We train all configurations of the model with $K = 2$.

We compare visual interpolation to linear interpolation and a forward tracking model as widely used baselines. As a metric, we plot the recall@0.7 curve as a function of the average number of manual boxes annotated per object track. For this comparison we uniformly sample keyframes at different sampling intervals. Fig 5 shows that visual interpolation works clearly better than linear interpolation and tracking. We choose DaSiamRPN visual interpolation as the model with better performance for further experiments.

Results for frame selection guidance. First, to motivate the choice of model architecture, we compare the performance of three variations: the architecture without attention, the architecture without visual features, and the full model. We compare them in terms of binary classification accuracy. More precisely, we randomly sample pairs of test frames from the validation set, such that (1) the difference in performance between two frames within a pair is significant, and (2) the number of pairs where the first frame performs better than second is balanced (i.e. a random classifier produces accuracy 0.5).

The results are presented in Table 1. Our full model clearly wins against both baseline models. Further, the model using no attention does not do better than random chance. The larger gap for the test sample that does not contain small objects is probably explained by the fact that the smaller is an object, the more noisy are the labels on the validation set.

Fig. 6 shows the pairwise comparison matrix predicted by the model and the ground truth matrix that evaluates which frames are better to manually annotate so that the visual interpolation model would work better. Interestingly,

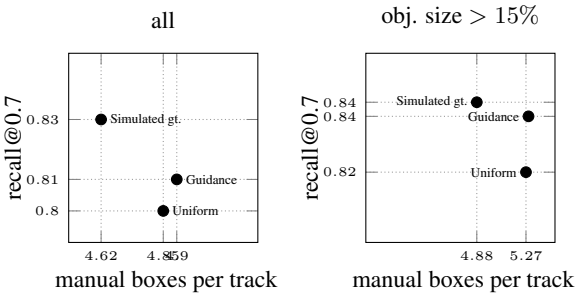

Figure 7: Frame selection guidance vs uniform sampling (in terms of recall@0.7); *simulated gt.*: ground truth is used for frame selection guidance; *guidance*: visual interpolation with keyframes predicted by our guidance module; *uniform*: visual interpolation with uniform keyframe selection.

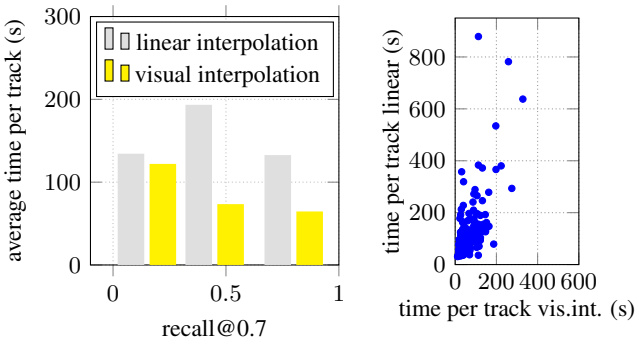

Figure 8: Left: linear interpolation vs visual interpolation: annotation time at three quality levels (lower is better). Right: annotation time comparison between the two approaches, where each blue dot is a different video.

the model confidence in the frame comparison correlates with the performance difference in the ground-truth, although the model is trained for classification.

We further show the improvement from using the frame selection guidance module in the full experiment (Fig 7). We compare running the visual interpolation module using uniformly spaced keyframes, versus with frame selection guidance. We also show guidance based on ground-truth signal for comparison (albeit it does not imply globally optimal keyframe selection per track).

As can be seen, our frame selection module outperforms the uniformly sampling frames and delivers bigger improvement for the subset that does not contain small objects. Overall, we point out that the problem of predicting model performance is a very challenging task, hence even 2% improvement is significant and can result in hours of annotation time spared.

4.2. Experiments with human annotators

Simulations do not provide full insights into the actual benefits and drawbacks of the proposed approach when used in practice. Hence we set up a video annotation ex-

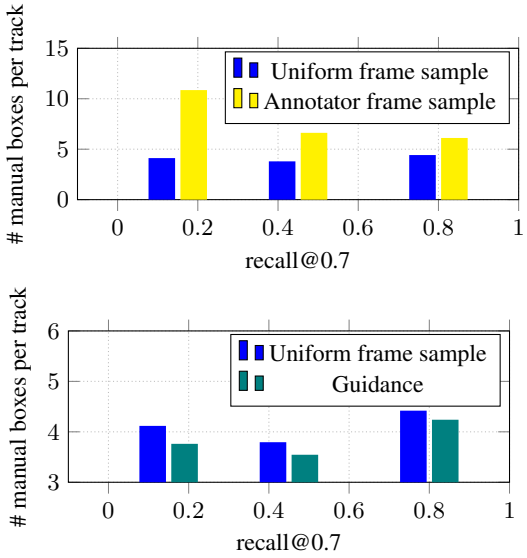

Figure 9: Top: number of manually drawn boxes per object track vs recall@0.7 for simulated annotations at uniform frame sampling (every 40th frame) vs annotations by human annotators. Bottom: simulated annotation at uniform frame sampling (every 40th frame) vs annotation with frame selection guidance.

periment with human annotators. We use the validation set of the Got10k [13] dataset and compare the results obtained by annotators with the simulation results. Got10k is a highly diverse dataset containing in total 563 classes, hence we are able to demonstrate the generalization properties of our model. Got10k validation set contains 180 videos, with a single annotated object in each video. We perform human studies with 10 human annotators. Each annotator is asked to annotate the same set of videos with two annotation methods. The target object is defined by a bounding box annotation in the first frame of each video.

The annotators are given a quality target of 70% overlap with (hidden ideal) groundtruth box in each frame and recommended time per question of 2 minutes.

Fig 8 presents the results of the linear vs visual interpolation comparison. With visual interpolation the annotators are able to achieve significant speedup at all quality level considered. Moreover, overall across all annotators and videos in the dataset, visual interpolation reduced annotation cost by about 50%: it took total of 6.96 hours to annotate the dataset with linear interpolation and only 3.45 hours with visual interpolation. The average quality of annotations in terms of recall@0.7 is 0.73 for linear interpolation and 0.75 for visual interpolation. The annotations were not given any specific guidelines as to how to select which frames should be annotated manually. For both visual and linear interpolation they relied on their understanding of which frames should be annotated.

Next, we investigate how well the annotators select which frame to annotate. In Fig 9 we compare the selec-

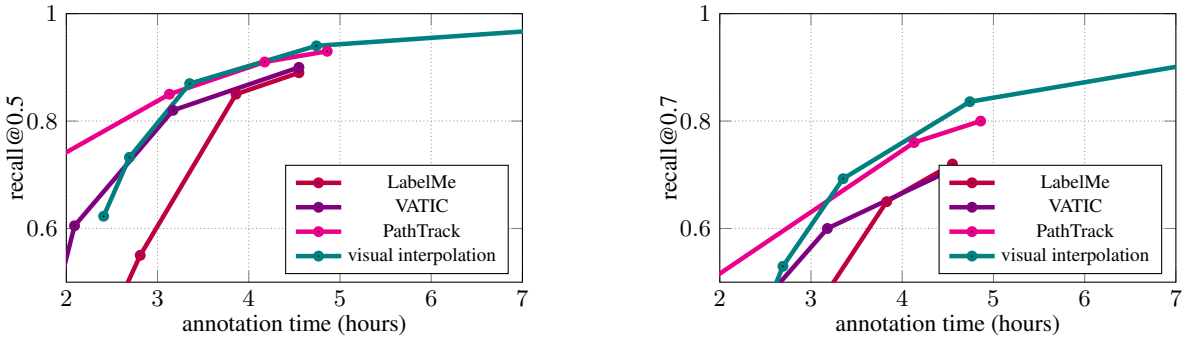

Figure 10: Annotation time required to annotate MOT2015 dataset with a given quality in terms of recall@0.5 and recall@0.7.

tion made by human annotators vs uniform frame sampling. We can clearly see that humans lack the ability to select frames optimally: even uniform frame selection with constant sampling interval (40 in this experiment) leads to a faster annotation process, or better annotation quality at the same speed.

Finally, we also evaluate efficiency gains from applying our frame selection guidance mechanism (Sec. 3.2). In Fig 9-right we show that frame selection model allows to surpass the performance of uniform sampling and improves over the baseline where humans select the frames to annotate themselves. These show that frame selection model delivers on average 6.5% reduction in the number of manual boxes needed, at no loss in quality. Those results demonstrate the importance of the good models for frame selection for the annotation process, as for large-scale annotation even small improvement can bring significant cost savings.

4.3. Comparison to other annotation tools

In this section, we compare our full method to other annotation tools [25, 41, 42] on the MOT2015 [20] dataset. The training set contains 11 video sequences with an average of 45 tracks per video. The dataset contains only annotations for the class "person" but some videos contain 100+ annotated tracks, creating challenging setting for single-object tracking algorithms.

We compare to the results reported in [25] (for PathTrack, as well as for VATIC [41] and LabelMe [42]), as they performed a comprehensive evaluation of their approach and compare to several other state-of-the-art annotation tools. To perform the comparison, we estimate the actual annotation time based on the time measurements provided in [25] and the number of boxes drawn manually in our protocol. According to [25], the average time to draw a box is $t_{box} = 5.2$s and the total annotation time is calculated as:

$$t_{track} = \lambda t_{watch} + t_{box} \cdot N_{box} \quad (4)$$

where t_{watch} is the time for watching through a track, t_{track} is the annotation time per track and N_{box} is the number of boxes the annotator has drawn.

The results are presented in Fig. 10 on two metrics: recall@0.5 and recall@0.7 versus annotation time. Fig 10 shows that, when collecting many boxes of high quality our method outperforms all provided baselines. For example, at 80% of the data annotated with quality of 0.7 IoU or higher, we achieve a 10% reduction of the annotation time compared to the strongest baseline (PathTrack). The more the required annotation quality increases, the bigger is the advantage of our method in terms of annotation time. We want to underline that PathTrack [25] is designed as a method for fast but imprecise annotation, while our method is designed for obtaining more accurate annotations and hence each method serves a different purpose. Further, our method is generic (not specific to the 'person' class) and does not require post-processing of the data (PathTrack needs to align automatically detected boxes with annotated object tracks). For example, compared to VATIC [41] and LabelMe [42], we achieve 33% speedup for the fixed quality of 70% of the boxes annotated with quality of 0.7 IoU or higher.

5. Conclusions

We presented and evaluated a unified framework for interactive video bounding box annotation. We introduced a visual interpolation algorithm which is based on contemporary trackers but allows for track correction. Moreover, we presented a frame selection guidance module and experimentally showed its importance within the annotation process.

We evaluated (in simulations) that using a visual signal allows to annotate 60% less boxes than the traditionally used linear interpolation while keeping the same quality. In experiments with human annotators we have shown that annotation time can be reduced by more than 50% using the proposed framework. Further, we also showed that proposed approach saves 10% of annotation time compared to the state-of-the-art method Pathtrack (and more compared to LabelMe [42] and VATIC [41]) on challenging multi-object tracking dataset MOT2015 [20].

References

[1] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. *arXiv preprint arXiv:1606.09549*, 2016.

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. *CoRR*, abs/1904.07220, 2019.

[3] Bowen Chen, Huan Ling, Xiaohui Zeng, Gao Jun, Ziyue Xu, and Sanja Fidler. Scribblebox: Interactive annotation framework for video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2020.

[4] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002.

[5] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] Pedro Gil-Jiménez, Hilario Gómez-Moreno, Roberto López-Sastre, and Saturnino Maldonado-Bascón. Geometric bounding box interpolation: an alternative for efficient video annotation. *EURASIP Journal on Image and Video Processing*, 2016.

[7] Brent A. Griffin and Jason J. Corso. Bubblenets: Learning to select the guidance frame in video object segmentation by deep sorting frames. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] Chunhui Gu, Chen Sun, David Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. pages 6047–6056, 06 2018.

[9] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Joao Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 04 2014.

[11] Junfei Zhuang Yuan Dong Hongliang Bai hiqun He, Yingruo Fan. Correlation filters with weighted convolution responses. *IEEE International Conference on Computer Vision*, 2017.

[12] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[13] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *CoRR*, 2018.

[14] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 29:5–28, 1998.

[15] Suyog Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. 2014.

[16] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, Abdelrahman Eldesokey, Jani Kapyla, and Gustavo Fernandez. The seventh visual object tracking vot2019 challenge results, 2019.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[18] Alina Kuznetsova, Sung Ju Hwang, Bodo Rosenhahn, and Leonid Sigal. Expanding object detector's horizon: Incremental learning framework for object detection in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[19] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *CoRR*, 2018.

[20] Laura Leal-Taixé, Anton Milan, Ian D. Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *CoRR*.

[21] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *arXiv preprint arXiv:1812.11703*, 2018.

[22] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[25] Santiago Manen, Michael Gygli, Dengxin Dai, and Luc Van Gool. Pathtrack: Fast trajectory annotation with path supervision. In *International Conference on Computer Vision (ICCV)*, 2017.

[26] Ishan Misra, Abhinav Shrivastava, and Martial Hebert. Watch and learn: Semi-supervised learning of object detectors from videos. *CoRR*, 2015.

[27] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-

scale dataset and benchmark for object tracking in the wild. *ECCV*, 2018.

[28] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[30] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, Jan. 1999.

[31] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. *CoRR*, abs/1702.00824, 2017.

[32] David Rolnick, Andreas Veit, Serge J. Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *CoRR*, 2017.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, Dec. 2015.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[35] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[36] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2019.

[37] Jack Valmadre, Luca Bertinetto, João F. Henriques, Ran Tao, Andrea Vedaldi, Arnold W. M. Smeulders, Philip H. S. Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. *ECCV*, 2018.

[38] Sudheendra Vijayanarasimhan and Kristen Grauman. Active frame selection for label propagation in videos. pages 496–509, 10 2012.

[39] Sudheendra Vijayanarasimhan and Kristen Grauman. Active frame selection for label propagation in videos. In *ECCV 2012*, 2012.

[40] Carl Vondrick and Deva Ramanan. Video annotation and tracking with active learning. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 28–36. Curran Associates, Inc., 2011.

[41] Carl Vondrick, Deva Ramanan, and Donald Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV*, ECCV'10, pages 610–623, Berlin, Heidelberg, 2010. Springer-Verlag.

[42] Ketaro Wada. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme, 2016.

[43] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. *arXiv preprint arXiv:1812.05050*, 2018.

[44] Wenguan Wang and Shenjian Bing. Super-trajectory for video segmentation. *ICCV*, 2017.

[45] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *CoRR*, 2018.

[46] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking. *CoRR*, 2018.

[47] Yunhua Zhang, Dong Wang, Lijun Wang, Jinqing Qi, and Huchuan Lu. Learning regression and verification networks for long-term visual tracking. *CoRR*, 2018.

[48] Zheng Zhu, Qiang Wang, Li Bo, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *European Conference on Computer Vision*, 2018.