Collaborative Contributions for Better Annotations

Priyam Bakliwal[1], Guruprasad M. Hegde[2] and C. V. Jawahar[1]
[1] *International Institute of Information Technology, Hyderabad, India*
[2] *Bosch Research and Technology Centre, Bengaluru, India*

Keywords: Video-Processing, Active-Leaning, Surveillance Video Annotations, Tracking.

Abstract: We propose an active learning based solution for efficient, scalable and accurate annotations of objects in video sequences. Recent computer vision solutions use machine learning. Effectiveness of these solutions relies on the amount of available annotated data which again depends on the generation of huge amount of accurately annotated data. In this paper, we focus on reducing the human annotation efforts with simultaneous increase in tracking accuracy to get precise, tight bounding boxes around an object of interest. We use a novel combination of two different tracking algorithms to track an object in the whole video sequence. We propose a sampling strategy to sample the most informative frame which is given for human annotation. This newly annotated frame is used to update the previous annotations. Thus, by collaborative efforts of both human and the system we obtain accurate annotations with minimal effort. Using the proposed method, user efforts can be reduced to half without compromising on the annotation accuracy. We have quantitatively and qualitatively validated the results on eight different datasets.

1 INTRODUCTION

With increase in use of surveillance cameras and decrease in cost of storage and processing of surveillance videos, there is a huge availability of unlabeled video data. This data can be utilized in many high level computer vision tasks such as motion analysis, event detection and activity understanding. Computer vision models that do video analysis (Zhong and Chang, 2001; Zhong et al., 2004) require accurately annotated data for both training and evaluation. However, annotating massive video sequences is extremely expensive and may not be feasible.

The use of tracking algorithms to generate annotated data lack in terms of detection accuracy and reliability making them unsuitable for critical applications like surveillance systems, transport, sports analysis, medical imaging, etc. Most of the recent algorithms (Gray et al., 2007; Zhou et al., 2009; Thomas et al., 2010) use the appearance model as a prerequisite for the success of a tracking system. It is extremely challenging to design a robust appearance model which can be adaptive to all the working conditions like partial/full occlusion, illumination changes, motion blur, shape changes, etc.. These methods do give us a significant improvement in tracking output but they are still not reliable enough to be used for generation of annotated data.

There have been many attempts in the past to generate annotated data from videos. However, these methods are not often used for large industrial scale annotations because they usually lack annotation consistency and accuracy. In most cases (Kavasidis et al., 2012), human annotators mark the object of interest in a video sequence. As pointed out in (Vondrick et al., 2013), manual annotations, involve a huge cognition load, and is subjected to inefficiency and inaccuracies. Some efforts that use crowd sourcing to increase the number of annotations, mainly for building large corpora (Deng et al., 2009; Oh and et. al., 2011; Russell et al., 2008), suffer from inconsistent annotations as most workers are poor annotators. In video annotation, the marking consistency of an annotator is extremely important as it becomes difficult to capture the marking variation in shape and extent of object within neighboring frames. Thus, crowd sourcing mandates robust quality control protocols.

Due to extremely high cost of human annotation for large video datasets, much of the research efforts have been dedicated towards leveraging the use of unlabeled data. Many algorithms developed recently are using semi-supervised learning (Fergus et al., 2009; Lee et al., 2011), or weakly-labeled data (Thomas et al., 2010), which is faster to annotate. All of these algorithms aim at reducing the number of annotations needed.

The video annotation framework proposed by Vondrick and Ramanan (Vondrick and Ramanan,

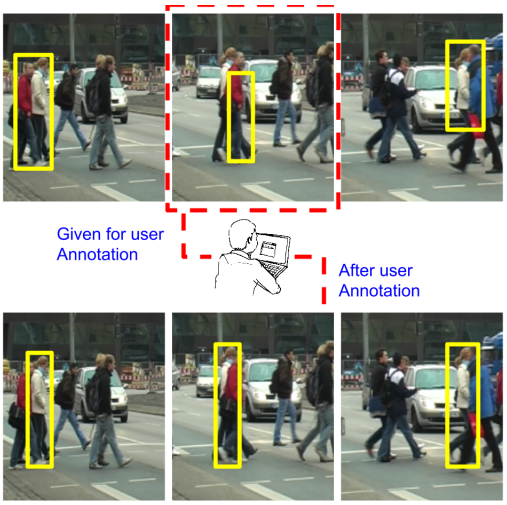


Figure 1: Use of Active Learning for object tracking in video sequences. Top row shows the tracking output on 3 frames of TUD-Crossing(4) sequence. The tracker fails to track the person in white jacket accurately. Our sampling technique selects the most informative frame (shown in red rectangle) for user annotation. The proposed algorithm ensures more accurate tracking with minimal user efforts. Bottom row shows better predictions for entire sequence with only one user annotation.

2011) is based on the video annotations using active learning. In this system annotations are derived by tracking results and active learning is used to intelligently query the human annotator for corrections on the tracks. Angela *et. al.* (Angela et al., 2012), uses an incremental learning approach which continuously updates an object detector and detection thresholds, as an user interactively corrects annotations proposed by the system. In their work, the learning approach is paired with an active learning element which predicts the most difficult images. Their solution is purely based on detection and does not consider the tracking. However, our approach incorporated tracking into detection, making it more robust while ensuring minimal annotation effort.

Works in similar lines includes (Chatterjee and Leuski, 2015; Zha et al., 2012; Höferlin et al., 2012) which utilize active learning for video indexing and annotation. However, they do not incorporate the power of existing efficient tracking algorithms to create a robust and accurate framework for real time object detection in video sequences.

Contributions: In this work, we use tracking algorithms to detect the objects in multiple frames and active learning is used to improve the correctness of the tracks. We propose *(i)* an effective tracking algorithm and *(ii)* an adaptive key-frame strategy that use active learning to intelligently query the annotator to label the objects at only certain frames which are most likely to improve the performance. The proposed active learning strategy can also be used in other computer vision tasks. We propose a framework that can easily incorporate various tracking algorithms, making it more generalized. Multiple tracking algorithms (2 in our case) are combined efficiently to produce a reliable and accurate track for the object.

One of the major contributions of this method is consideration of neighborhood in selection of key frames. Also, we have used 'Query by Committee' strategy for key frame selection. Consideration of temporal neighborhood makes sure that with each user annotation, the tracking is best updated for neighboring frames as well. The advantages of our method includes easy incorporation of tracking algorithms, automatic detection of key frames thereby drastically reducing human efforts and scalable annotation process. This makes our approach suitable for annotations of large video datasets.

We performed experiments on objects of multiple datasets and show that the user efforts for doing annotation can be reduced up to 50% when using the proposed active learning strategy without compromising on tracking accuracy. We also show that with the same amount of user efforts the proposed method achieves an improvement of up to 200% for tracking task. We report experimental results on 8 different datasets consisting of more than 2500 frames and 17 objects. The consistent improvement in all scenarios demonstrate the utility of our approach.

2 TRACKING ALGORITHMS

We employ three tracking algorithms in this work. The most simple uses bi-linear interpolation which does not consider object characteristics and predicts tracks using initialization only. The other two algorithms are the modification of two state of the art tracking methods, Weighted Multiple Instance Learning Tracker (WMILT) (Zhang and Song, 2013) and Discriminative Scale Space Tracker (DSST) (Danelljan et al., 2014). WMILT uses weighted instance probabilities to detect object of same size in other frames. On the other hand, DSST uses discriminative correlation filters based on a scale pyramid representation to track the object. DSST algorithm is scale invariant while the WMILT algorithm works for objects with not much scale change.

2.1 Bi-linear Interpolation

Interpolation is the basic approach to the problem of object tracking. In simple terms, the linear interpo-

lation of two known points given by the coordinates (x_0, y_0) and (x_1, y_1) is the straight line between these points. In the problem of video annotation, the main criteria is how to decide the key frames. In this approach the user is asked to annotate every n^{th} frame and rest of the frames are simply tracked using interpolation. There is a trade-off between tracking accuracy and annotation cost. Smaller value of n leads to better track but higher annotation cost.

2.2 Bidirectional WMILT

Weighted Multiple Instance Learning Tracker (WMILT) (Zhang and Song, 2013) integrates the sample importance into the learning procedure. A bag probability function is used to combine the weighted instance probability. The algorithm weighs the positive instances according to their importance to the bag probability, it assumes that the weight for the instance near the target location is larger than that far from the target location.

The algorithm relies on positive and negative samples. The positive samples and negative samples are separated into two bags. The initialized target is labeled as positive. The contribution of each positive sample is calculated using a monotone decreasing function with respect to the Euclidean distance between the locations of sample and target. In this way the tracker integrates the sample importance into the learning procedure.

Intuitively, all the instances in the negative bag are very far and completely dissimilar to the target. Therefore, the algorithm treats all negative instances to contribute equally to the negative bag. Finally, a bag log-likelihood function is used to find the instance for which the probability is maximized. The algorithm efficiently integrates the sample importance into the learning procedure to detect the similar sized target in rest of the image sequence. We have used the algorithm to detect the object location both in backward as well as forward image sequence resulting in higher tracking accuracy.

2.3 Bidirectional DSST

The DSST algorithm (Danelljan et al., 2014) extends discriminative correlation filters (Bolme et al., 2010) to multi-dimensional features for visual object tracking. We utilize this method to predict the target locations in both the temporal directions of the video sequences, so as to improve the prediction.

The algorithm uses HOG features along with image intensity features. An image is represented as d-dimensional feature map from which a rectangular target patch is extracted. An optimum correlation filter is found by minimizing the cost function. We build a 3-dimensional scale space correlation filter for scale invariant visual object tracking. The filter size is fixed to $M \times N \times S$, where M and N are height and width of the filter and S is the number of scales. A feature pyramid is constructed from a rectangular area around the target and the pyramid is centered at the target's location and scale. A 3-dimensional Gaussian function is then used to get the desired correlation output.

This correlation filter is used to track the target both in previous and next frames of the image sequence. Given a new frame, a rectangular cuboid of size $M \times N \times S$ is extracted from the feature pyramid. Similar to above, the cuboid is centered at the predicted location and scale of the target. We compute the correlation scores and the new target location and scale is obtained by finding the maximum score.

3 ACTIVE LEARNING BASELINES

Generally, annotating massive videos is extremely expensive. There are hundreds of hours of surveillance video footage of cars and pedestrians which will require a lot of human effort to annotate. Currently video annotations are done typically by having paid users on Mechanical Turk labeling a set of key frames followed by linear interpolation (Yuen et al., 2009).

3.1 Interpolation with key frame selection

We extend the interpolation based tracking by adding dynamic key frame selection strategy. As discussed earlier, the interpolation method is highly dependent on key frame selection interval n. The optimum value of n vary from object to object. For example, consider an object moving at a constant pace for few frames and then change speed during later frames. Such cases makes it hard to find a single optimum value of n for even one object.

A slight modification in naive linear interpolation approach can significantly reduce the human efforts. We have designed a tool that initially asks the user to annotate first and last frame of the video sequence. the tool calculates the object track using linear interpolation. It also gives flexibility to the user to decide which frame to annotate next so as to improve the tracking accuracy. This avoids the problem that occurs due to fixing the n for a given object.

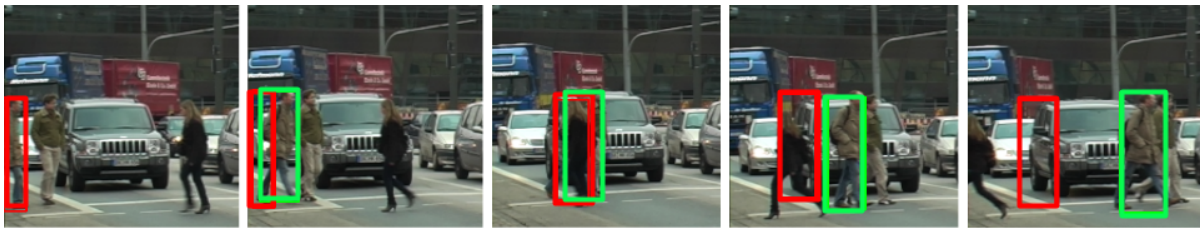


Figure 2: The behavior of proposed method 'Collaborative Neighborhood Tracker' in case of occlusion. First frame is the initialization frame and red bounding boxes are the initial tracking output. The algorithm selects middle frame as the key frame. After user annotation, the track updates (shown in green). Clearly, the tracking algorithm is handling occlusion well and the key frame selection is improving the overall track.

3.2 Uncertainty based Active Learning

One of the simplest and most intuitive key frame selection strategy is uncertainty sampling. In this method, the algorithm queries the frames about which the tracker is least certain. in this approach we use both the tracking algorithms, *viz.*, WMILT and DSST, separately. For WMILT, we use classifier probability as the measure to define uncertainty. Whereas, to calculate uncertainty for DSST, we consider both tracker's translation and scale correlation confidence scores. The frame with minimum tracker's score / confidence, is considered as the next frame for user annotation. The tracker's output is updated after every user annotation.

4 COLLABORATIVE TRACKING

We propose a new collaborative approach to improve tracking accuracy while ensuring minimal user efforts. We use the tracking algorithms described in section 2 and combine them in a novel way to get an enhanced hybrid tracking algorithm. The DSST being scale invariant algorithm is complimentary to WMILT algorithm which detects similar sized objects. Hence, a combination of both gives a higher tracking accuracy.

4.1 Collaborative Tracker

We have collaborated the two trackers (WMILT and DSST) into a new tracker named 'Collaborative Tracker'. We represent the target as a bounding box enclosing its spatial extent within a video frame. The bounding box is represented as a 4-dimensional vector representing top-left and bottom-right corner coordinates of the target. Let the predicted bounding boxes of above two trackers be P_w and P_D. Then the collaborated output is given by:

$$P_C = \varepsilon P_W + (1-\varepsilon)P_D \tag{1}$$

where ε ($0 \leq \varepsilon \leq 1$) is the weight assigned to the individual tracker outputs. The value of ε is fixed at the start of the annotation. If the size of object across the whole track is constant and does not vary much, the value of ε is greater than 0.5 else it is less than 0.5.

We also propose an adaptive frame sampling scheme which uses active learning to intelligently asks the human annotator to annotate the target only in few specific frames that are likely to improve the performance. This approach is based on the fact that for any tracking algorithm, not all the objects/videos can be treated equally.

The performance of any tracking algorithm can vary significantly depending upon the scenario in the video. Some objects are comparatively easier to annotate automatically. For example, the frames in which a person is standing. In such cases, only one frame initialization might give required track. However, the more complex scenarios require a lot more annotation efforts to get the desired track. Thus, the proposed key frame sampling scheme helps to utilize the annotation efforts on more complex objects (or frames) that are visually ambiguous, such as occlusions or sudden change of appearance (see Fig 2).

Suppose at time t, the task is to figure out the frame that the user should annotate next. We utilize the difference in opinion principle to determine the next frame. The center (C) of the predicted bounding box (P) is given by:

$$C = \frac{P_1 + P_3}{2}, \frac{P_2 + P_4}{2} \tag{2}$$

where P_1 and P_2 are coordinates of top left corner of the bounding box and P_3 and P_4 are coordinates of bottom right corner. For a frame i, we determine the difference in center predictions D_i^t for all three algorithms at time $t-1$ using:

$$D_i^t = C_{ic}^{t-1} \oplus C_{id}^{t-1} \oplus C_{iw}^{t-1} \tag{3}$$

where,

$$a \oplus b \oplus c = dist(a,b) + dist(b,c) + dist(c,a) \tag{4}$$

$dist(x,y)$ is the Euclidean distance between x and y. Next, we determine the frame that best helps in improving the tracker output for all other frames. We select the most useful key frame as the frame with largest center difference as per 'Query by Committee' strategy. The key frame f^t at any time t is found using:

$$f^t = \arg\max_i(D_i) \tag{5}$$

These key frame annotations are used to track the object using different tracking algorithms and ultimately each track adds up to the accuracy of the final output. Intuitively, the track of an object at a particular frame is more accurate when the initialization is done in the near by frame. Therefore, for every frame we use the tracking output for the iteration where (initialization) key frame is closest. The tracker output for frame i at time t (P^t_{Ci}) is calculated using Eq1 as:

$$j = \arg\min_{k=1}^{t-1} |f^k - i| \tag{6a}$$

$$P^t_{Ci} = \varepsilon P^j_W + (1-\varepsilon)P^j_D \tag{6b}$$

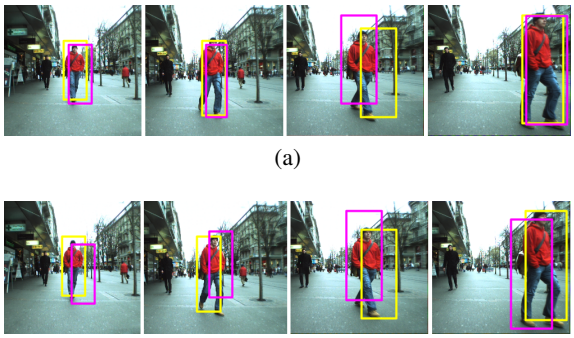
The algorithm finds out most uncertain frame and asks the user for its annotation. The correction of this frame results in overall improvement of the tracking accuracy. Therefore, with every iteration the tracker improves and the algorithm updates the object positions, thereby making the prediction more accurate.

4.2 Collaborative Neighborhood Tracker

In this approach, we consider the uncertainty of temporal neighboring frames to determine the next key frame. The intuition behind this is that every user annotation should improve the object location in the whole video sequence and not just the current frame (See Fig 3). Thus, we consider the temporal neighborhood center difference along with the current frame's center difference to decide the next key frame to be given for user annotation. This makes our sampling scheme robust. In this approach, the key frame selection is done using:

$$f^t = \arg\max_i(\sum_{j=1}^{T} \eta e^{-|j-i|} D_j) \tag{7}$$

where η is a normalization constant and T is the total number of frames in the sequence. All the frames in the sequence are considered in the neighborhood of every key frame, more closer the neighboring frame to the key frame the greater is the impact of its center difference. Similar to Collaborative Tracker, the Collaborative Neighborhood Tracker is expected to become more accurate with each user annotation. In this case, the improvements are expected to be more because the selection of key frame is based on collective uncertainty of temporal neighborhood making the selection process more informative. Finally, Eq 6 is used to give the final track of the object.



(a)

(b)

Figure 3: Importance of consideration of neighborhood frames while key frame selection. The output of two different trackers are shown in separate color bonding boxes. To decide next key frame to annotate there are two ways: (a) Based on tracker disagreement of candidate frame. (b) Based on tracker disagreement of candidate neighborhood. Clearly, second scenario is better to be given to user for annotation. User annotation of its third frame will update the tracking output of neighbors as well resulting in better track and more reduction in error.

5 EXPERIMENTS

We have performed multiple experiments on eight different publicly available datasets and show the effectiveness of the proposed algorithm in various scenarios.

5.1 Datasets and Evaluation Measures

We have used 17 sequences from standard tracking datasets like ETH-Bahnhof, ETH-Jelmoli, ETH-Sunnyday, TUD-Campus, TUD-Crossing, TUD-Stadmitte, David, Couple, etc. to evaluate the performance of the proposed technique. The video sequences pose several challenges such as illumination changes, size and pose changes, motion blurs, partial and full occlusions etc. to tracking algorithms.

We have selected the following criteria to measure the performance and provide comparisons among different tracking algorithms.

Average Error: Average Error is the mean of difference between each side of the bounding box generated by the tracker $\{(x_3,y_3),(x_4,y_4)\}$ and the ground truth $\{(x_1,y_1),(x_2,y_2)\}$.

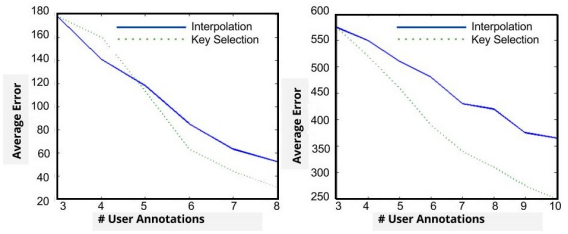


Figure 4: Change in Average Error with the number of user annotations for Liner Interpolation and Key Frame Selection(M1) for TUD-Campus(2) and TUD-Crossing(8).

$AvgError = (|x_1 - x_3| + |x_2 - x_4| + |y_1 - y_3| + |y_2 - y_4|)/4$.

Edge Error: An edge error occurs if the difference between an edge of the bounding box generated by the tracker and the ground truth is more than 5 pixels, i.e., if $|e_i^{Tracker} - e_i^{GT}| \geq 5$ then edge error is 1 else 0 (where, $e_i^{Tracker}, e_i^{GT}$ are edges of tracker and ground truth). The edge error is then summed up for all 4 edges of the bounding box.

Centroid Error: Centroid Error is the Euclidean distance between centroids of bounding boxes of the tracker and the ground truth. Centroid error is calculated as $\sqrt{(\bar{c_x} - c_x)^2 + (\bar{c_y} - c_y)^2}$, where, $(\bar{c_x}, \bar{c_y})$ are ground truth centroid coordinates and (c_x, c_y) are tracker centroid coordinates.

5.2 Comparison of various Active Learning Strategies

The main aim of video annotation framework is to generate the track for different objects with minimal user interaction. In this section we describe several experiments to show the effectiveness of the proposed algorithm. We have referred Key Frame Selection as M1, Uncertainty (WMILT) as M2, Uncertainty (DSST) as M3, Collaborative Tracker (proposed approach) as M4 and Collaborative Neighborhood Tracker (proposed approach) as M5. Also for datasets we have used notations, TC for TUD-Campus, TCr for TUD-Crossing, EJ for ETH-Jelmoli, ES for ETH-Sunnyday and EB for ETH-Banhof. As mentioned earlier the value of ε depends on the variations in the size of object across the whole track. We have used different values of ε for each object in the dataset. For objects such as ETH-Banhof(2,3) where the variation in object size is much we have used lower values (0.20, 0.22), where as, for objects like Couple and David the value of ε is higher (0.75,0.80).

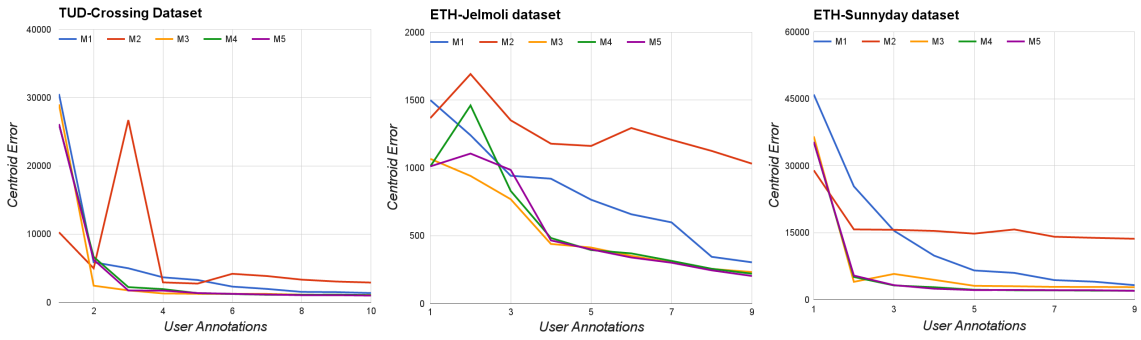
In this experiment we compare the traditional annotation technique with the active based method. Existing video annotation framework (Yuen et al., 2009) typically have users labeling frames at regular intervals followed by linear interpolation. Fig 4 shows the decrease in average error with increase in number of user annotations for Linear Interpolation and Key Frame Selection(M1). Clearly, the decrease in error shows that the active learning based solution is better than traditional annotation technique.

In this experiment, we show that the proposed tracker is suitable for mission critical applications like automotive surveillance. For such applications limb precision is very important. The limb precision of an algorithm can be captured accurately using the edge error. Thus, the aim is to calculate the number of user annotations required by different algorithms to get an 'Edge Error' less than 1 per frame. A better algorithm should achieve this error rate with minimum possible user annotations. Table 1 shows the number of user annotations required to get an edge error less than 1 per frame for different active learning algorithms. We observe that our proposed method 'Collaborative Neighborhood Tracker' (M5) consistently outperforms all other approaches. This is due to the incorporation of temporal neighborhood information into key frame sampling scheme. Notice that, 'Collaborative Tracker' (M4) which lacks neighborhood information performs better than M1, M2 and M3 confirming that our hybrid tracker is more accurate than individual trackers.

Fig 2 shows the behavior of proposed method (M5) in case of occlusion. The algorithm intelligently selects the key frame so as to improve the overall track. Clearly, the tracking algorithm is improving after every user annotation.

Another important measure to decide the effectiveness of any object detection framework is the 'Average Error'. The annotation algorithm having least average error after a certain number of user annotations is the better one. Table 2 shows the average error achieved by different annotation algorithms with same number of user interactions. Clearly, the proposed method (M5) is performing better than other annotation algorithms. We are able to achieve nearly half error with same user efforts then the other approaches.

Centroid Error measures the precision of the center of the bounding box. For every good annotation algorithm the centroid of trackers output should be as close as possible to the center of the ground truth. In this experiment we have measured the centroid error precision for different active learning strategies on three datasets namely TUD-Crossing, ETH-Jelmoli and ETH-Sunnyday. We have aggregated the centroid errors for all the object after each user annotation to check the convergence of these algorithms. Fig 5



(a) (b) (c)

Figure 5: Change in 'Centroid Error' with increasing user annotations for (a) TUD-Crossing, (b) ETH-Jelmoli and (c) ETH-Sunnyday datasets. Clearly, error for our proposed algorithms 'Collaborative Tracker' and 'Collaborative Neighborhood Tracker' is decreasing faster than other annotation algorithms.

Objects	Active Learning Approaches				
	M1	M2	M3	M4	M5
TC(2)	12	15	10	**6**	7
TCr(1)	7	13	5	3	**2**
TCr(5)	10	20	15	13	**12**
TCr(8)	16	22	19	15	**12**
EB(2)	12	18	8	11	**7**
EB(3)	12	20	7	7	**5**
EJ(1)	9	14	3	**2**	**2**
EJ(2)	8	12	4	**2**	**2**
EJ(5)	9	17	5	6	**4**
ES(2)	9	14	6	5	**4**
ES(5)	24	20	18	16	**15**
ES(12)	8	15	9	7	**5**
ES(34)	7	10	8	7	**6**
Couple	41	28	11	9	**7**
David	12	15	11	9	**8**
Total	196	253	139	118	**98**

Table 1: The number of user annotations required to get an edge error less than 1 per frame. The value in the parenthesis indicates the object ID. The best results are reported in bold.

Objects	Active Learning Approaches				
	M1	M2	M3	M4	M5
TC(2)	205	206	175	159	**157**
TCr(1)	176	224	147	148	**105**
TCr(5)	485	815	525	**398**	426
TCr(8)	908	937	816	713	**701**
EB(2)	485	887	427	398	**381**
EB(3)	288	305	209	164	**128**
EJ(1)	85	224	77	70	**56**
EJ(2)	104	314	99	88	**80**
EJ(5)	206	447	266	193	**187**
ES(2)	222	487	286	199	**184**
ES(5)	3277	1889	1756	1487	**1401**
ES(12)	418	725	300	263	**233**
ES(34)	195	400	153	171	**146**
Couple	2099	1204	1644	1140	**934**
David	3962	1742	921	1258	**880**
Total	13122	10811	7807	6852	**6005**

Table 2: Average error(rounded to nearest integer) achieved by different annotation algorithms after 5 user annotations. The value in the parenthesis indicates the object ID. The better algorithm should achieve less error in same number of user annotations.

shows the change in 'Centroid Error' with increasing user annotations for different datasets. Clearly, error for our proposed algorithms 'Collaborative Tracker' and 'Collaborative Neighborhood Tracker' is decreasing faster than other annotation algorithms.

Another major concern while doing large scale video annotation is the scalability of the annotation algorithm. The annotation cost increases significantly for videos with larger duration. For these experiments we have taken objects (multiple datasets) of varied length and calculated the number of user annotations required to get a satisfactory track (Average error less than 5 pixels per frame). From fig 6, the perfor-

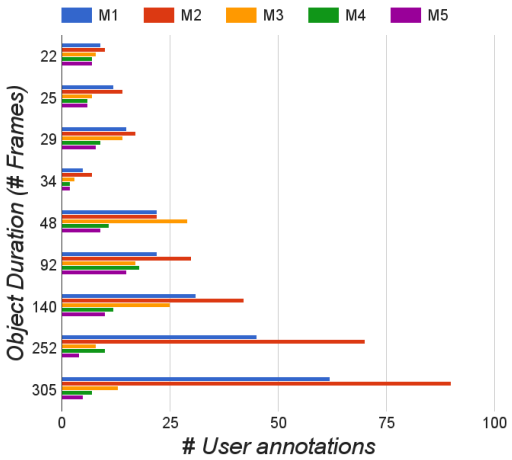


Figure 6: The number of user annotations required by objects of different length to achieve average error less than 5 pixel per frame. Clearly, M4 and M5 (proposed methods) requires significantly less user efforts especially for objects with longer video sequences.

mance difference is high for objects that are present for larger number of frames which shows that the proposed method is effective for both short as well as long video sequences. This shows that the proposed approach is highly scalable.

Thus, from above experiments it is clear that using the proposed approach user efforts required for video annotations can be reduced to 50%. Also, the method is scalable and robust to challenges like occlusion.

6 CONCLUSION

In this paper, we propose an efficient and accurate method to effectively annotate huge video sequences with minimal user efforts. The approach is suitable for generating large annotated datasets for mission critical applications like surveillance and autonomous driving. We effectively utilize the active learning approach to decide the best selection of key frames. This makes our approach scalable to generate huge annotations for large scale surveillance and automotive related videos with substantial reduction in human efforts. We have verified that using the proposed approach, annotation efforts can be reduced to half while maintaining the track quality.

REFERENCES

Angela, Y., Juergen, G., Christian, L., and Luc, Van, G. (2012). Interactive object detection. *CVPR*.

Bolme, a. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. *CVPR*.

Chatterjee, M. and Leuski, A. (2015). CRMActive: An active learning based approach for effective Video annotation and retrieval. *ICMR*.

Danelljan, M., Haumlger, G., Shahbaz Khan, F., and Felsberg, M. (2014). Accurate scale estimation for robust visual tracking. *BMVC*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *CVPR*.

Fergus, R., Weiss, Y., and Torralba, A. (2009). Semi-supervised learning in gigantic image collections. *NIPS*.

Gray, D., Brennan, S., and Tao, H. (2007). Evaluating appearance models for recognition, reacquisition, and tracking. *PETSW*.

Höferlin, B., Netzel, R., Höferlin, M., Weiskopf, D., and Heidemann, G. (2012). Inter-active learning of ad-hoc classifiers for video visual analytics. *VAST*.

Kavasidis, I., Palazzo, S., Di Salvo, R., Giordano, D., and Spampinato, C. (2012). A semi-automatic tool for detection and tracking ground truth generation in videos. *VIGTAW*.

Lee, Jae, Y., and Grauman, K. (2011). Learning the easy things first: Self-paced visual category discovery. *CVPR*.

Oh, S. and et. al. (2011). A large-scale benchmark dataset for event recognition in surveillance video. *CVPR*.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *IJCV*.

Thomas, D., Bogdan, A., and Ferrari, V. (2010). Localizing objects while learning their appearance. *ECCV*.

Vondrick, C., Patterson, D., and Ramanan, D. (2013). Efficiently scaling up crowdsourced video annotation. *IJCV*.

Vondrick, C. and Ramanan, D. (2011). Video annotation and tracking with active learning. *NIPS*.

Yuen, J., Russell, B., Liu, C., and Torralba, A. (2009). Labelme video: Building a video database with human annotations.

Zha, Z. J., Wang, M., Zheng, Y. T., Yang, Y., Hong, R., and Chua, T. S. (2012). Interactive video indexing with statistical active learning. *Transactions on Multimedia*.

Zhang, K. and Song, H. (2013). Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*.

Zhong, D. and Chang, S.-F. (2001). Structure analysis of sports video using domain models. *International Conference on Multimedia and Expo*.

Zhong, H., Shi, J., and Visontai, M. (2004). Detecting unusual activity in video. *CVPR*.

Zhou, H., Yuan, Y., and Shi, C. (2009). Object tracking using sift features and mean shift. *Computer vision and image understanding*.