Efficient Object Annotation for Surveillance and Automotive Applications

Sirnam Swetha[1], Anand Mishra[1], Guruprasad M. Hegde[2] and C.V. Jawahar [1]

[1]International Institute of Information Technology, Hyderabad
[2]Bosch Research and Technology Centre, India

Abstract

Accurately annotated large video data is critical for the development of reliable surveillance and automotive related vision solutions. In this work, we propose an efficient and yet accurate annotation scheme for objects in videos (pedestrians in this case) with minimal supervision. We annotate objects with tight bounding boxes. We propagate the annotations across the frames with a self training based approach. An energy minimization scheme for the segmentation is the central component of our method. Unlike the popular grab cut like segmentation schemes, we demand minimal user intervention. Since our annotation is built on an accurate segmentation, our bounding boxes are tight. We validate the performance of our approach on multiple publicly available datasets.

1. Introduction

In the recent years researchers have developed and demonstrated robust computer vision methods based on supervised machine learning techniques. They show impressive results on tasks such as large scale object detection, object recognition and image classification [4, 10, 14, 18, 19], reinstating the hope that they are nearing mainstream adaptation. Most of these methods find applications in fields such as image search, web based face recognition engines [20] and other applications [16]. However these methods are far from mature when it comes to deploying in mission critical applications such as in surveillance, robotics, and autonomous driving. Accuracy and reliability of the vision algorithms need further improvement.

Since the state-of-the-art computer vision schemes depend on supervised learning techniques, such a deficiency in performance can be attributed to the following reasons. Firstly, it is difficult to annotate and generate large sets of training data covering a wide gamut of foreseeable working conditions which is necessary for supervised learning algorithms to generalize. The second hurdle is to generate a large amount of validation and test datasets that can help to find the model complexity and benchmark the performance. For example, one needs to test a solution for several hundreds of hours to reliably estimate the performance in an autonomous driving to be practical. Finally, the complex models that can be trained with the computational resources and evaluated on a wide range of hardware. There has been many success stories of deep learning in recent years which is known to be data intensive. A common underlying component is the generation of large scale reference data, also known as Ground Truth (GT). We are interested in this. In this work we restrict the scope of GT to a bounding box enclosing the spatial extent of articulate objects (humans) within a video frame. They are the most common and vulnerable subjects in the context of surveillance and autonomous driving scenarios. There have been many attempts in generating annotated data in computer vision in the past (eg. ImageNet). However, large industrial scale annotation efforts are not often reported in literature.

(a) (b) (c) (d)

Figure 1: (a), (c) Relaxed initialization generated with key frame user annotation, (b), (d) Accurate annotations generated by our approach.

In most cases GT is generated via manual annotators [11, 12, 17] who mark the object of interest in a set of images. More recently researchers have started to use video sequences to generate GT as they provide much richer representations of the object [2, 13]. As pointed out in [21], manual annotation especially in videos involves a huge cognition load, and is subject to inefficiency and inaccuracies. This is more evident while annotating humans as the limbs

might move in a nonlinear manner and is difficult to capture the resulting variation in shape and extent of object within neighboring frames. Recently researchers have engineered various approaches to address the above aspects. Some of them are based on crowd-sourcing [2] and others use computer vision and machine learning techniques to develop semi-automatic annotation methods [1, 3, 5, 9, 13, 22]

In this work we focus on one of the aspects of video annotation namely the annotation propagation. This step plays a vital role in reducing the cognition load by incorporating human inputs - in the form of annotations - at certain frames and automatically transferring them to the neighboring frames. This eliminates the need to manually annotate every frame in the video. Our annotation propagation method involves segmenting an object and propagating the segmentation mask across the frames. We illustrate some example results of our method in Figure 1. In section 4 we show that the accuracy of these results is superior to those that rely on interpolation.

2. Related work

There have been many attempts in annotating images and videos in the past. Researchers from IBM developed a video annotation tool namely VideoAnnEX [3]. The objective of the tool is to generate GT, so as to facilitate the process of video/information retrieval based on certain user query (e.g., provide me all the frames/information related to a particular public figure). This uses a supervised learning method to label scenes in video but lacks object level annotation and propagation.

ViPER [1, 9] is an open-source tool developed by Language and Media Processing lab (LAMP) at University of Maryland to generate GT in video sequences and also an evaluation framework to evaluate performance of algorithms for tracking, recognition and detection. To speed up annotation process ViPER provides mechanism to interpolate bounding box position of objects in between key frames. In addition to this it is also possible to propagate object specific attributes between key frames by using the copy functionality or dragging through video frames. Although ViPER has annotation propagation method, in general it is not effective and scalable while generating GT of articulated objects as the accuracy of propagation is directly proportional to granularity of key frame interval and nature of object motion which in our case is nonlinear (e.g. limbs and legs). In other words, it would be difficult to fully capture the limbs of a person moving across the camera's field of view.

LabelMe video (LMV) [22] is another open source web accessible video annotation system that allows to annotate object category, shape, motion, and interactions between them. It is an extension of the popular LabelMe image annotation tool. Here the GT is generated by automatically propagating manual annotation across neighboring frames with the help of offline recorded camera motion parameters. These parameters are used to estimate homography between adjacent frames which is then used to accurately propagate the annotation. Although the method works reasonably well, in most cases it might not be practical as it would involve expensive additional hardware to record precise camera motion parameters.

The authors of Innovative Web based collaborative platform for video annotation [13] provide ways to share annotations through a collaborative web based platform. In addition the tool derives annotations based on annotations from multiple users. Also the tool provides GrabCut based features to extract boundaries of an object. The major limiting factor of this tool is that it doesn't involve propagation of object boundaries or GT across frames.

Another web based video annotation tool known as VATIC [2] is based on crowd-sourcing and developed at University of California at Irvine with collaborations from Massachusetts Institute of Technology. The tool is in experimental stages and was hosted on amazon's mechanical turk to investigate the potential of crowd sourcing platforms for the task of video annotation. The tool incorporates active learning based methods to propagate annotation across frames. However, the authors do not place much importance on the accuracy of an object's spatial extent and hence might not readily fit our purpose to generate GT for mission critical applications and articulated objects.

An interesting video annotation tool named as iVAT [5] incorporates incremental learning based approach to build online detector that aids linear interpolation and template matching based annotation propagation across video frames. The role of the detector is mainly to resolve occlusions and handle any non-uniform/non-linear motion of an object. However, the tool doesn't determine the spatial extent of an object in each frame and hence may significantly affect the accuracy of GT in our case.

As a complementary component to the existing tools, we propose a method where annotations on key frames are propagated automatically using motion cues. We adopt GrabCut [8] based segmentation method for videos to obtain highly accurate annotations for objects in large scale videos efficiently. GrabCut is a successful interactive segmentation scheme. We adapt it for our problem so that it demands very minimal user intervention (say on selected key frames). The advantages of our method are two fold, firstly we only need user interactions on key frames, and hence it reduces the human effort drastically. Secondly, since our method tracks the object using segmentation, we obtain more accurate bounding boxes around objects.

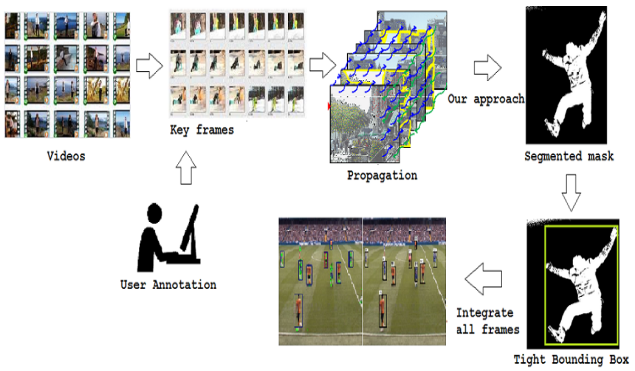

Figure 2: Proposed framework: given a set of videos, user annotates selected key frames. We propagate the annotations for the entire sequence and use it as initialization for our approach and get a tight bounding box.

3. Our Approach

Given a large collection of videos our goal is to design a method which can automatically propagate the annotations for the objects in those videos with very minimal user interaction. We propose a method for accurate propagation of annotations of objects in videos using segmentation. Our proposed scheme is illustrated in Figure 2. For this our method is inspired by the success of GrabCut [8] and thereafter works such as objCut [15] for natural image segmentation. Nevertheless, we propose several useful modification to the original GrabCut [8] to suit our problem. GrabCut has shown state-of-the-art image segmentation results on some of the standard benchmarks. But it has some limitations in real time applications for videos. For example: (i) initialization is critical in GrabCut and often performed by user. On large collection of videos asking for user interaction in all frames (or most frames) is not a feasible solution. (ii) computing min cut on every frame is a costly operation. We resolve the above limitations with a semi-automatic approach and also reducing the computations.

In this section we first formulate the problem as energy minimization problem, briefly describe GrabCut method and discuss about our proposed semi-automatic initialization scheme.

3.1. Energy Minimization Framework

We formulate the problem of segmenting objects in video frames in an energy minimization framework. Segmentation of an image can be expressed as a vector of binary random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where each random variable X_i takes a label $x_i \in \{0, 1\}$ based on whether it is object or background.

We represent pixels of frames as nodes in a graph where all the neighbouring nodes are connected by edges. We associate a unary and pairwise cost of labeling these nodes and define a cost (or energy) function as the sum of these cost for all the nodes as follows:

$$\psi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) = \sum_i \psi_i(x_i, \boldsymbol{\theta}, z_i) + \sum_{(i,j)\in\mathbf{N}} \psi_{ij}(x_i, x_j, z_i, z_j), \tag{1}$$

where, $\mathbf{N}$ denotes the neighborhood system defined in the MRF, and ψ_i and ψ_{ij} correspond to unary and pairwise costs respectively.

A typical unary cost can be expressed as:

$$\psi_i(x_i, \boldsymbol{\theta}, z_i) = -\log p(x_i|z_i), \tag{2}$$

where z_i is the rgb color vector, $p(x_i|z_i)$ is the likelihood of pixel i taking label x_i. This likelihood is computed from learnt foreground and background GMM model. Reader is encouraged to refer [8] for more details. The pairwise cost is the given by [6]:

$$\psi_{ij}(x_i, x_j, z_i, z_j) = \lambda \frac{[x_i \neq x_j]}{ed(i,j)} \exp\left(\beta(z_i - z_j)^2\right), \tag{3}$$

where the parameter λ controls the degree of smoothness, $ed(i, j)$ is the Euclidean distance between neighboring pixels i and j. The constant β allows edge-preserving smoothing, and is computed as follows: $\beta = 1/2\mathbb{E}[(z_i - z_j)^2]$, where $\mathbb{E}[u]$ is expected value of u.

The problem of segmentation is now to find the global minima of the cost function in 1, i.e.,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \psi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}). \tag{4}$$

The global minima of this cost function can be efficiently computed by graph cut [7]. We use iterative graph cut based approach for computing this.

3.2. Semi-automatic Initialization

Given a set of video frames $\{f_1, f_2, \cdots, f_m\}$ we need to run iterative graph cuts in all the m frames. This is computationally not smart way. Moreover, foreground and background region needs to be initialized in each frame. In original GrabCut foreground and background are initialized by performing user interaction. However, we wish to avoid such user interactions to make our annotation process efficient and minimal human intensive. In other words we need two modifications from the original GrabCut method: (i) we wish to avoid performing iterative graph cuts in every frame, and (ii) we wish to minimize user interaction to great extent. To achieve this we propose following three automatic initialization schemes.

M1: Bi-linear interpolation. Given the annotation of key frames we extend the annotations to all the in-between frames by simply interpolating these annotations. These interpolated annotations are used to initialize the Gaussian mixture models of foreground and background in these frames.

M2: Relaxed interpolation. In the relaxed interpolation we extend the width and height of the interpolated bounding boxes by a small offset (w, h). Relaxed bounding boxes contain most or all the object inside them and achieve higher pixel level recall.

M3: Using motion cues and dynamic GMM update. Each pixel is initialized with mixture of Gaussian mixture models (GMM). Using optical flow, we estimate the new position of each pixel. The new position can have different pixel value. Hence, we update the GMM for each frame dynamically. Further, we estimate the number of Gaussian's to fit the model as the appearance changes across frames. This adapts to the incremental changes in the appearance of the object e.g., changes in illumination, object appearance. Assumption is moving pixels are considered as pixels belonging to the object or foreground. Again, there might be multiple moving objects, to get only the object of interest, we use the relaxed bounding box (M2) to eliminate the other pixels. Once we have an approximate estimate of the foreground pixels, the above estimated foreground mask is used as initialization seeds for GrabCut.

In brief, we propose the following simple but effective modifications for efficient segmentation and propagation to accurately annotate objects from the videos:

- We use relaxed interpolation to get the relaxed bounding box for the object of interest. It is calculated initially for the entire sequence (using the user annotations on key frames).
- Compute foreground and background GMM model at the key frames using the user annotations and these models are propagated to in-between frames.
- The above pre-computed models are used to segment object in nearby frames. As the object will be very similar in the nearby frames.
- The object neighborhood is sufficient to segment the object rather than taking the entire image. This reduces the computations drastically without effecting the results.

Summarizing our approach, our method initiates with a very minimal user interaction (say drawing loose bounding box around objects on key-frames). We obtain annotations in the key-frames by segmenting them using iterative graph cuts. We propagate these annotations to the in-between frames using motion cues and dynamically update the GMMs for those frames. With these propagated annotations, segmentation and bounding boxes around all the objects in all the frames are obtained.

(a) (b) (c) (d) (e) (f)

Figure 3: Sample dataset images. Challenges: (a), (b) - change in object size, moving camera and moving object, (c), (d), (e), (f) - complex surroundings, occlusion and object pose variation.

4. Experiments and Results

4.1. Datasets

We have performed experiments on a variety of publicly available datasets for object tracking. Each dataset has its own challenges, for example movement through cluttered areas, objects overlapping in the visual field, lighting changes, moving background, slow-moving objects, and objects being introduced or removed from the scene (refer Figure 3). For analysis, we selected a set of objects from these datasets with the above challenges. Sample images from the datasets we use are shown in Figure 3.

4.2. Performance Measures

There exists abundant performance measures in the field of object annotation in videos. We chose to use average area overlap and recall, as these performance measures the accurateness of annotations (as we aim for tight bounding box). We briefly introduce these measures here.

Average area overlap: It is the intersection area divided by union of ground truth (GT) and the bounding box (BB) generated by an annotation approach. The mean of this measure is calculated by dividing with total number of frames in the database.

$$AreaOverlap = \frac{1}{N}\sum_{k=1}^{n}\frac{Area(B_k^1 \cap B_k^2)}{Area(B_k^1 \cup B_k^2)} \qquad (5)$$

Recall: Recall is computed as a fraction of true positive and true positive plus false negative, which are defined as

follows.

$$TP = \sum_{k=1}^{n} \frac{Area(B_k^1 \cap B_k^2)}{Area(B_k^1 \cup B_k^2)} \quad (6)$$

$$FN = \sum_{k=1}^{n} \frac{Area(B_k^1)}{Area(B_k^1 \cup B_k^2)} - TP \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

where B_k^1, B_k^2 are bounding boxes of GT and annotation approach for k^{th} frame, N is the total number of frames, TP, FN are true positive and false negative respectively.

4.3. Results

We have applied the proposed approach on the datasets by varying the number of key frames, which accounts for user interactions. We have experimented the same in multiple settings by varying the key frame interval. The segmentation based method is not very successful in frames, where the user is occluded or improperly initialized (user initializes object where it is not completely visible). Also, we detect the frames where segmentation is unsuccessful and replace them with the interpolation result, which is not effected by occlusions. Hence, the proposed work clearly outperforms simple interpolation based technique.

Table 1 shows the area overlap and recall for objects with varying key frame intervals. Each object has a different context, TUD-Stadtmitte 1 object is for static camera, ETH-Jelmoli 1 object is for the case where both object and camera motion are in same direction and ETH-Banhof 3 object moves in opposite direction to the camera motion. From this table, we observe that (i) our proposed scheme M3 which uses motion cues performs better than M1 and M2, (ii) as expected with lower key frame interval we achieve higher recall and higher overlap (i.e., more accurate annotations).

Further, it should be noted that for ETH-Banhof 3, there are drastic changes in object size, due to movement in opposite direction. Hence, the errors are higher compared to TUD-Stadtmitte 1 and ETH-Jelmoli 1. Figure 5 shows the variation in area overlap for the methods discussed in Section 3.2 namely M1 and M3 with the change in frame interval for two different objects. To compare area overlap we use M1 and M3, as the annotations generated by M2 are relaxed bounding boxes, due to which the area overlap is less for M2. We also observe that for the object ETH-Jelmoli1, the area overlap for M3 is very high compared to M1 as our approach yields very accurate segmentations.

The proposed method achieves high area overlap and high recall using very few user annotations (refer Figure 4). We can see that our approach yields accurate segmentation which leads to accurate annotations. This is very helpful for large scale video annotations as we generate accurate annotations with drastic reduction in the human annotation efforts. Consider an example where an user have a video of $10,000$ frames to be annotated. For a key frame interval of 10, user has to annotate only 1000 frames in our approach. This reduces the human efforts by 90 % without compromising on the accurateness of the annotations.

Object	Area Overlap			Recall		
	M1	M2	M3	M1	M2	M3
TS1 (5)	**0.873**	0.843	0.869	0.923	0.922	**0.931**
TS1 (15)	0.851	0.831	**0.867**	0.901	0.894	**0.927**
TS1 (50)	0.837	0.771	**0.855**	0.883	0.866	**0.906**
TS1 (100)	0.828	0.761	**0.847**	0.871	0.857	**0.896**
J1 (5)	0.884	0.894	**0.938**	0.937	0.96	**0.965**
J1 (15)	0.846	0.856	**0.906**	0.907	0.937	**0.963**
J1 (50)	0.812	0.836	**0.903**	0.883	0.924	**0.963**
J1 (100)	0.793	0.825	**0.898**	0.868	0.916	**0.964**
B3 (5)	0.831	0.684	**0.84**	0.831	0.836	**0.847**
B3 (15)	0.76	0.701	**0.77**	0.901	0.813	**0.907**
B3 (50)	0.65	0.655	**0.67**	0.829	0.791	**0.834**
B3 (100)	0.61	0.634	**0.65**	0.801	0.784	**0.903**

Table 1: M1: Bi-linear interpolation, M2: Relaxed interpolation, M3: Our approach which uses motion cues to propagate annotations (Section 3.2). The value in the parenthesis indicates the key frame interval. TS: TUD-Stadmitte, J: ETH-Jelmoli, B: ETH-Banhof

5. Conclusion

We have presented a framework for semi-automatic object annotation to generate accurate Ground Truth (GT) data in large scale from videos. Especially, our approach is suitable for generating GT for mission critical applications like surveillance and autonomous driving. Our method of object annotation is based on segmentation and its propagation which results in accurate bounding boxes around the objects. The proposed framework outperforms interpolation based approaches and almost mimics human annotation ability with only minimal user interaction (predominantly at key frames) which makes it scalable to generate large sets of GT. We have verified our claims by conducting comprehensive experiments on multiple challenging video datasets. Our approach can prove useful in generating ground truth and annotations for large scale surveillance and automotive related videos with substantial reduction in human efforts.

References

[1] http://viper-toolkit.sourceforge.net/.

[2] http://web.mit.edu/vondrick/vatic/.

[3] http://www.research.ibm.com/videoannex/index.html.

Figure 4: Results of our approach on object TUD-Stadtmitte 1. Row 1 and Row 2 show the initialization sequence and result of our method respectively. We observe that the object is segmented accurately using our method which results in an accurate annotation.

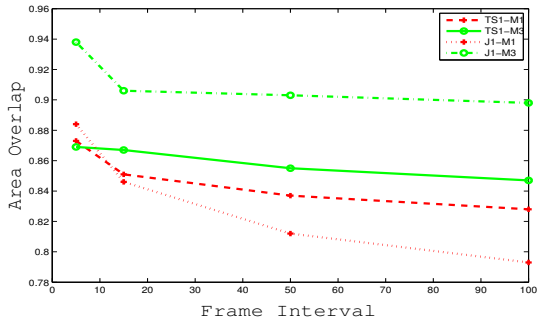

Figure 5: Frame interval vs Area overlap for the objects TS1: TUD-Stadtmitte 1 and J1:ETH-Jelmoli1.

[4] G. C. A. Frome, A. Abdulkader, M. Zennaro, A. B. B. Wu, and L. V. H. Adam, H. Neven. Large-scale privacy protection in street-level imagery. In *ICCV*, 2009.

[5] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini. An interactive tool for manual, semi-automatic and automatic video annotation. *CVIU*, 2015.

[6] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *ICCV*, 2001.

[7] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *PAMI*, 2004.

[8] V. K. Carsten Rother and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[9] D. S. Doermann and D. Mihalcik. Tools and techniques for video performance evaluation. In *ICPR*, 2000.

[10] C. T. F. Nasse and G. A. Fink. Face detection using gpu-based convolutional neural networks. In *CAIP*, 2009.

[11] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVPR Workshops*, 2007.

[12] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[13] I. Kavasidis, S. Palazzo, R. D. Salvo, D. Giordano, and C. Spampinato. An innovative web-based collaborative platform for video annotation. *Multimedia Tools Appl.*, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.

[15] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJCUT: efficient segmentation using top-down and bottom-up cues. *PAMI*, 2010.

[16] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *ISCAS*, 2010.

[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[19] J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012.

[20] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[21] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 2013.

[22] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *ICCV*, 2009.