ViTBAT: Video Tracking and Behavior Annotation Tool

Tewodros A. Biresaw[†‡] Tahir Nawaz[⋆] James Ferryman[⋆] Anthony I. Dell[†‡]

[†]National Great Rivers Research and Education Center, Alton, IL, USA
[‡]Washington University in St. Louis, St. Louis, MO, USA
[⋆]University of Reading, Whiteknights, Reading, UK
t.biresaw@wustl.edu, t.h.nawaz@reading.ac.uk, j.m.ferryman@reading.ac.uk, adell@lc.edu

Abstract

Reliable and repeatable evaluation of low-level (tracking) and high-level (behavior analysis) vision tasks require annotation of ground-truth information in videos. Depending on the scenarios, ground-truth annotation may be required for individual targets and/or groups of targets. Unlike the existing tools that generally allow an explicit annotation for individual targets only, we propose a tool that enables an explicit annotation both for individual targets and groups of targets for the tracking and behavior recognition tasks together with effective visualization features. Whether for individuals or groups, the tool allows labeling of their states and behaviors manually or semi-automatically through a simple and friendly user interface in a time-efficient manner. Based on a subjective assessment, the proposed tool is found to be more effective than the well-known ViPER tool on a series of defined criteria. A dedicated website makes the tool publicly available for the community.

1. Introduction

Ground-truth information is often used to test and evaluate the performance for different computer vision tasks such as optical flow estimation [1], stereo correspondence estimation [23] and video tracking [18], where the term 'ground truth' refers to the ideal performance an algorithm is desired to achieve. In video tracking, as the case for other tasks, the deviation of an estimated result with respect to the corresponding ground-truth information is measured to provide algorithmic performance [17, 18, 24]. Non-ground-truth-based methods also exist [3, 4, 22, 28] that could be useful when ground-truth information is not available. Ground-truth-based evaluation however offers an advantage over non-ground-truth-based evaluation of providing a more reliable, confident and repeatable algorithmic evaluation and comparison with a performance benchmark available *a priori* [19]. Moreover, ground-truth information may also be employed as a training data for machine learning algorithms in order to make predictions for unforeseen data [7].

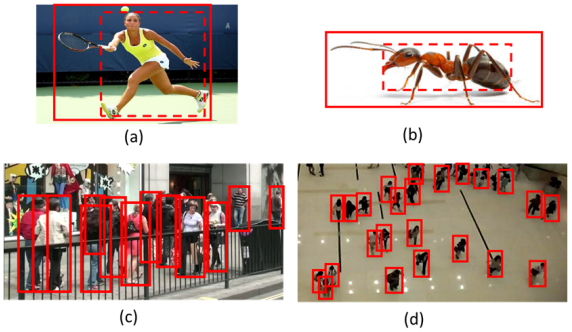


Figure 1. (a,b) Subjectivity in target state annotation: applications involving motion analysis may need annotation of only key body parts, e.g. head and torso, (dotted lines), whereas action recognition may need full body within the bounding box (solid lines). Moreover, the annotation task becomes challenging in a crowded scene (c) or when size of targets is small (d).

Tracking and behavior recognition are widely-researched topics in computer vision [5, 9, 20] with applications in surveillance [15, 20], robotics [11] and biology [6], to name a few. Existing tracking and behavior recognition methods are usually tested on a small number of datasets containing short-term video sequences [16]. Indeed, there is a plethora of datasets that could be obtained nowadays using for example cheap and hand-held cameras as well as from online sources such as YouTube; however, an absence of the associated ground-truth information hinders their use for evaluation and assessment of algorithms. Additionally, it is also important to note that ground-truth generation is often a highly subjective task. Indeed, ground-truth labels for one application domain may not be directly applied to other applications (Fig. 1(a,b)). Moreover, ground-truth generation becomes

978-1-5090-3811-4/16/$31.00 ©2016 IEEE

quite challenging in crowded scenes (Fig. 1(c)) or when target size is small (Fig. 1(d)). Therefore, the need remains for effective annotation tools in order to generate the desired ground-truth information for different datasets, applications and situations.

Considering the importance of ground-truth generation, several annotation tools have been proposed over the years including LabelME [29], VATIC [27], ViPER [8], iVAT [2] and JAABA [14]. These tools are generally more suitable for annotating ground-truth information at individual target level in terms of tracking and/or behavior of targets. Indeed, depending on an application at hand, annotation of tracking and behavior could also be needed for groups of targets [10, 26], which is explicitly not considered in existing tools. Moreover, an annotation tool is desired to be user friendly, minimize human effort and maximize annotation quality [2].

In this paper we propose a tool, named **Vi**deo **T**racking and **B**ehavior **A**nnotation **T**ool (ViTBAT), that allows users to generate ground-truth information for low-level (tracking) and high-level (behavior recognition and analysis) tasks in video sequences. Specifically, ViTBAT offers: (1) a comprehensive annotation of states and behavior labels at individual-target as well as group-target level; (2) representation of annotations (together with their IDs and behavior labels) of multiple targets and multiple groups in a simple-to-access matricial formats; (3) a simple but effective visualization of the annotations of individual targets as well as groups of targets and their associated behaviors over time; and (4) a friendly graphical user interface that minimizes human effort and maximizes annotation quality. The tool is made publicly available for the community at http://vitbat.weebly.com/.

The rest of this paper is organized as follows. The related works are discussed in Sec. 2. The proposed annotation tool is described in Sec. 3 that is followed by its comparison with an existing tool in Sec. 4. Sec. 5 concludes the paper.

2. Related work

Various interactive tools have been proposed for the annotation of tracking and behavior recognition. ViPER is a widely-used tool for target state annotation [8]. The tool also allows behavior annotation of targets in the form of attributes. However, those attributes do not offer enough simplicity and flexibility to annotate time-varying (appearing and disappearing) behaviors. VATIC annotates videos from crowd-sourced market places and provides a simple and easy-to-use interface [27]. VATIC has been extended by third parties to annotate target behaviors in particular the actions of humans. However, the behavior annotation and labeling is not so generic to be used for other applications of interest. LableME:Video, an extension of LableME:Image annotator, is a web-based tool for annotating arbitrary shapes across a video sequence [29]. A tool that allows the user to extract target states and group the targets into categories is presented in the form of iVAT [2]. A semi-automated machine-learning-based behavior annotator is presented as JABBA [14] that takes the already annotated states (trajectories) as input for performing the task. A more detailed review of existing annotators is provided in [2]. These tools aim at reducing human effort and time as well as maintaining the quality of the annotations. The annotation effort is reduced by automatically estimating states between selected key frames using linear interpolation and homography-preserving techniques [27, 29]. To drastically minimize the human effort, iVAT uses automated tracking and other computer vision methods together with interpolation for assisting manual annotation. However, human interventions and verifications are still mandatory for validating the quality of results obtained from automated methods.

Table 1. Summary of video annotation tools and their task. ✓* - allows grouping of targets at an elementary level.

Annotation tool	Individual target			Group target		
	State	State type	Behavior	State	Behavior	Visualization
LabelME:Video	✓	Arbitrary shape	✓	-	-	-
VATIC	✓	Rectangle	-	-	-	-
ViPER	✓	Ellipse, polygon	✓	✓*	-	-
iVAT	✓	Ellipse, polygon	✓	✓*	-	-
JAABA	-	-	✓	-	-	-
ViTBAT	✓	Point, rectangle	✓	✓	✓	✓

Existing tools are mainly aimed at annotating target states with some limitations in the annotation of behaviors [8, 27]. In fact, most of the tools do not enable time-varying behavior annotation and adding or deleting behavior labels [27]. Moreover, existing tools are generally more suitable for annotating state and behavior at an individual target level only, whereas annotation of state and behavior of groups of targets (as needed in various applications [10, 12, 15, 26]) are not explicitly considered. For instance, the tools do not allow an easy group annotation where members can leave and join the group at different times [2, 8]. Existing tools also generally do not provide an effective visualization feature for the annotated behaviors that are usually defined in segments of a sequence [2, 8, 27, 29]. Table 1 provides a summary of existing annotation tools.

3. Annotation tool

The proposed tool, ViTBAT, is an effective solution for annotation and visualization of temporal states and behaviors for both individual and group targets in videos. It aims to minimize the annotation effort while not compromising the annotation quality and provides a user-friendly interface. ViTBAT outputs the annotated state attributes (on image plane) and behavior labels for both individual targets

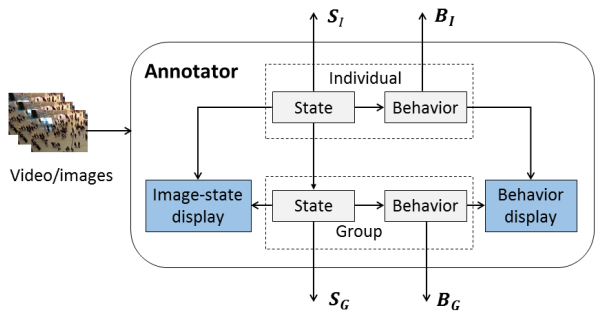


Figure 2. Flow diagram of components and outputs of ViTBAT. The outputs of the ViTBAT are four matrices: $\mathbf{S}_I$: matrix containing states of individual targets; $\mathbf{B}_I$: matrix containing behavior annotations for individual targets; $\mathbf{S}_G$: matrix containing states of groups of targets; $\mathbf{B}_G$: matrix containing behavior annotations for groups of targets.

and groups of targets in matrix form. This output format is chosen as it is extensively used to store output of algorithms and hence could facilitate their evaluation and comparison [17]. Fig. 2 shows the flow diagram of the annotation tool.

3.1. Annotation of states and behaviors of targets

Given a video sequence, V, ViTBAT allows annotating states ($\mathbf{S}$) as well as behaviors ($\mathbf{B}$) of both individual targets and groups at a frame k, where $k = 1, ..., K$ and K is the number of frames in V. Let us define $\mathbf{S}_I$ to be a matrix containing the state annotations for individual targets across V (with the subscript I referring to the case of individual targets):

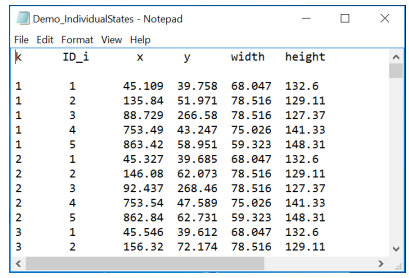
$$\mathbf{S}_I = [\mathbf{S}_{I,k}]_{k=1,...,K}, \tag{1}$$

where $\mathbf{S}_{I,k}$ denotes a sub-matrix in $\mathbf{S}_I$ containing the annotation attributes of states of targets at frame k. For a set of i_k targets present at frame k:

$$\mathbf{S}_{I,k} = [\mathbf{s}_{I,k,i}]_{i=1,...,i_k}, \tag{2}$$

such that $\mathbf{s}_{I,k,i}$ denotes the state annotation attributes for i^{th} target present at frame k, where the state representations can be broadly categorized as follows: point-based and area-based representation [18]. Point-based representation includes target positional information, whereas area-based representation includes also the information about the area occupied by a target on the image plane. ViTBAT offers flexibility in terms of annotating target state using point-based representation (x, y position) as well as area-based representation (x, y, width, height of a rectangular bounding box or an ellipse). For the case of point-based representation,

$$\mathbf{s}_{I,k,i} = [k \;\; ID_i \;\; x_{k,i} \;\; y_{k,i}], \tag{3}$$



Demo_IndividualStates - Notepad

File Edit Format View Help

k	ID_i	x	y	width	height
1	1	45.109	39.758	68.047	132.6
1	2	135.84	51.971	78.516	129.11
1	3	88.729	266.58	78.516	127.37
1	4	753.49	43.247	75.026	141.33
1	5	863.42	58.951	59.323	148.31
2	1	45.327	39.685	68.047	132.6
2	2	146.08	62.073	78.516	129.11
2	3	92.437	268.46	78.516	127.37
2	4	753.54	47.589	75.026	141.33
2	5	862.84	62.731	59.323	148.31
3	1	45.546	39.612	68.047	132.6
3	2	156.32	72.174	78.516	129.11

Figure 3. Snapshot of a sample file containing the matrix for state annotation of individual targets ($\mathbf{S}_I$).

where $(x_{k,i}, y_{k,i})$ and ID_i denote the target position on the image plane and its unique ID, respectively. For the case of area-based representation,

$$\mathbf{s}_{I,k,i} = [k \;\; ID_i \;\; x_{k,i} \;\; y_{k,i} \;\; w_{k,i} \;\; h_{k,i}], \tag{4}$$

where $(x_{k,i}, y_{k,i})$, $w_{k,i}$ and $h_{k,i}$ denote position (the top-left corner of the bounding box), width and height of the target bounding box on the image plane. Fig. 3 shows a snapshot of a sample file containing the values of $\mathbf{S}_I$.

Let us also define $\mathbf{B}_I$ to be a matrix containing the behavior annotations for a set of ID_I individual targets in V:

$$\mathbf{B}_I = [\mathbf{B}_{I,i}]_{i=1,....ID_I}, \tag{5}$$

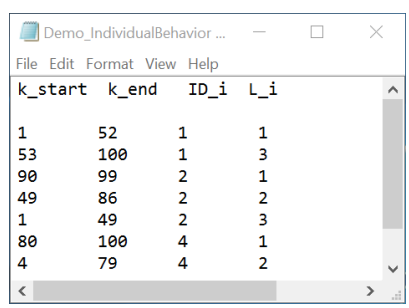
where $\mathbf{B}_{I,i}$ denotes a sub-matrix containing behavior annotations for the ith target across different segments of V:

$$\mathbf{B}_{I,i} = [k_{start,i,b} \;\; k_{end,i,b} \;\; ID_i \;\; L_{i,b}]_{b=1,...,L_I}, \tag{6}$$

where $k_{start,i,b}$ and $k_{end,i,b}$ are starting and ending frame numbers of a segment of V where a target with ID, ID_i, exhibits a behavior with a label $L_{i,b}$ such that L_I denotes the total number of behavior labels exhibited by individual targets in V. Fig. 4 shows a snapshot of a sample file containing the values of $\mathbf{B}_I$.

3.2. Annotation of states and behaviors of groups

Similarly to individual targets annotation, ViTBAT also allows annotating states as well as behaviors of groups of

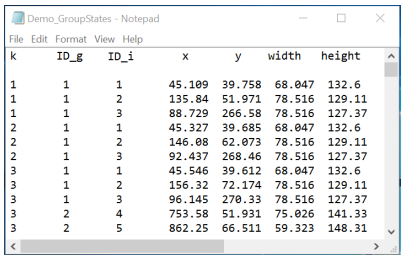


Demo_IndividualBehavior ...

File Edit Format View Help

k_start	k_end	ID_i	L_i
1	52	1	1
53	100	1	3
90	99	2	1
49	86	2	2
1	49	2	3
80	100	4	1
4	79	4	2

Figure 4. Snapshot of a sample file containing the matrix for behavior annotations of individual targets ($\mathbf{B}_I$).

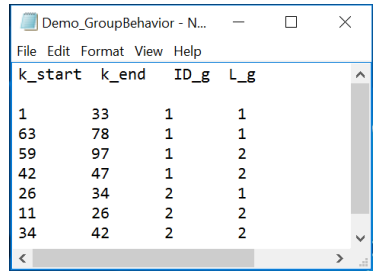


Demo_GroupStates - Notepad

File Edit Format View Help

k	ID_g	ID_i	x	y	width	height
1	1	1	45.109	39.758	68.047	132.6
1	1	2	135.84	51.971	78.516	129.11
1	1	3	88.729	266.58	78.516	127.37
2	1	1	45.327	39.685	68.047	132.6
2	1	2	146.08	62.073	78.516	129.11
2	1	3	92.437	268.46	78.516	127.37
3	1	1	45.546	39.612	68.047	132.6
3	1	2	156.32	72.174	78.516	129.11
3	1	3	96.145	270.33	78.516	127.37
3	2	4	753.58	51.931	75.026	141.33
3	2	5	862.25	66.511	59.323	148.31

Figure 5. Snapshot of a sample file containing the matrix for group state annotations of targets ($\mathbf{S}_G$).



Demo_GroupBehavior - N...

File Edit Format View Help

k_start	k_end	ID_g	L_g
1	33	1	1
63	78	1	1
59	97	1	2
42	47	1	2
26	34	2	1
11	26	2	2
34	42	2	2

Figure 6. Snapshot of a sample file containing the matrix for group behavior annotations of targets ($\mathbf{B}_G$).

targets across V. Let $\mathbf{S}_G$ be a matrix containing the state annotations for groups of targets across V (with the subscript G referring to the case of groups of targets):

$$\mathbf{S}_G = [\mathbf{S}_{G,k}]_{k=1,\ldots,K}, \tag{7}$$

where $\mathbf{S}_{G,k}$ denotes a sub-matrix in $\mathbf{S}_G$ containing the annotation of group states at frame k. For a set of g_k groups present at frame k:

$$\mathbf{S}_{G,k} = [\mathbf{S}_{G,k,g}]_{g=1,\ldots,g_k}, \tag{8}$$

where $\mathbf{S}_{G,k,g}$ denotes a sub-matrix in $\mathbf{S}_{G,k}$ containing the annotation of states for the gth group across V. For a set of i_{gk} individual targets in group g at frame k:

$$\mathbf{S}_{G,k,g} = [\mathbf{s}_{G,k,g,i}]_{i=1,\ldots,i_{gk}}, \tag{9}$$

such that $\mathbf{s}_{G,k,g,i}$ denotes the state annotation attributes for i^{th} target present in group g at frame k, where the point-based state representations is given by:

$$\mathbf{s}_{G,k,g,i} = [k \;\; ID_g \;\; ID_i \;\; x_{k,i} \;\; y_{k,i}], \tag{10}$$

and the area-based state representation can be written as:

$$\mathbf{s}_{G,k,g,i} = [k \;\; ID_g \;\; ID_i \;\; x_{k,i} \;\; y_{k,i} \;\; w_{k,i} \;\; h_{k,i}]. \tag{11}$$

The representation format of $\mathbf{S}_G$ allows to incorporate variable individual target members at different frames in a group. The representation also helps to assign an individual target to different groups at the same frame. Fig. 5 shows a snapshot of a sample file containing the values of $\mathbf{S}_G$.

Analogous to $\mathbf{B}_I$, $\mathbf{B}_G$ denotes a matrix containing the behavior annotations for a set of ID_G groups of targets in V:

$$\mathbf{B}_G = [\mathbf{B}_{G,g}]_{g=1,\ldots.ID_G}, \tag{12}$$

where $\mathbf{B}_{G,g}$ denotes a sub-matrix containing behavior annotations for the gth group across different segments of V:

$$\mathbf{B}_{G,g} = [k_{start,g,b} \;\; k_{end,g,b} \;\; ID_g \;\; L_{g,b}]_{b=1,\ldots,L_G}, \tag{13}$$

where $k_{start,g,b}$ and $k_{end,g,b}$ are starting and ending frame numbers of a segment of V where the group with ID, ID_g, exhibits a behavior with a label $L_{g,b}$ such that L_G denotes the total number of behavior labels exhibited by groups in V. Fig. 6 shows a snapshot of a sample file containing the values of $\mathbf{B}_G$.

3.3. Graphical User Interface (GUI)

The Graphical User Interface (GUI) of ViTBAT is shown in Fig. 7. The GUI is developed in MATLAB using its computer vision toolbox that allows implementation of the desired functions for ViTBAT. Since the annotation is done offline, high speed performance of GUI may not be mandatory for performing the annotation task. The GUI has three key parts: *image/video display* window, *annotated-behavior display* window and a set of *display and command button panels* for performing annotation.

The top-left window of GUI is dedicated for displaying annotated target/group states in image/video. The individual states are indicated by drawing either a bounding box or a point together with an associated ID (Fig. 8). For an effective visualization and to better distinguish one object from another, we utilize different colors that are maximally perceptually distinct [13]. The group annotations are indicated by straight lines interconnecting group members (Fig. 9). Like state annotations for individual targets, perceptually-distinct coloring is used for group annotations too. Just below the image/video display window are a bar and buttons that enable navigating across frames in the sequence.

The bottom-left window of GUI is dedicated for displaying annotated behaviors (Fig. 7). The window displays either the individual or group behavior annotation. In the window, the y-axis shows the list of defined behavior and the x-axis shows segments of frames exhibiting different behaviors in the form of colored horizontal lines (Fig. 10). The start and end points of a horizontal line correspond to the start frame and end frame of annotated behavior for a target/group, respectively. The color of each horizontal line matches the respective annotated state color of both target/group (in the image/video display window). The navigation bar in the image/video display window also allows a

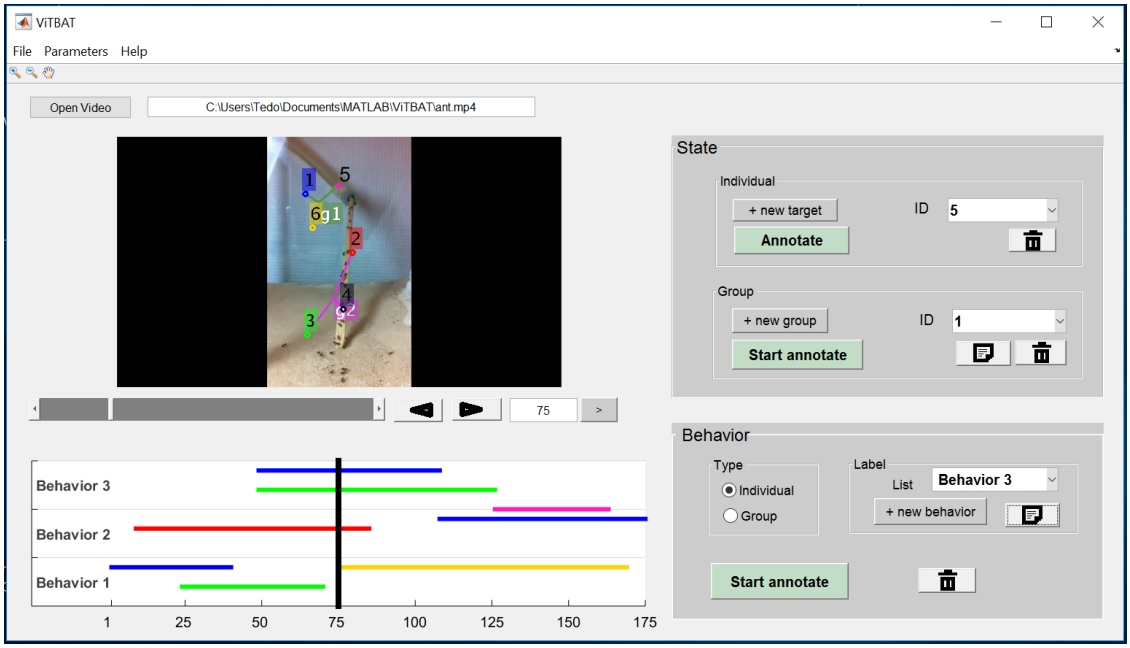


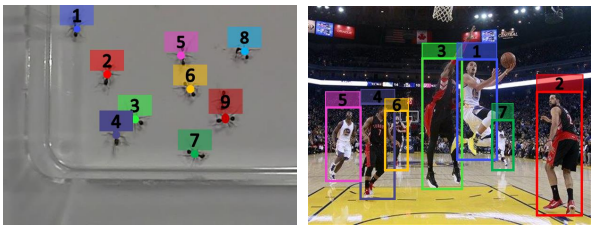
Figure 7. Graphical User Interface (GUI) of ViTBAT.

user to scroll the behavior annotations across the sequence.

The right side of the GUI is divided into two panels: one for state annotations for individual targets and groups, and the other one for behavior annotations for individual targets and groups. Each panel contains command buttons to generate annotations, ID displays for annotated states and a list of defined behaviors both for individual and group targets.

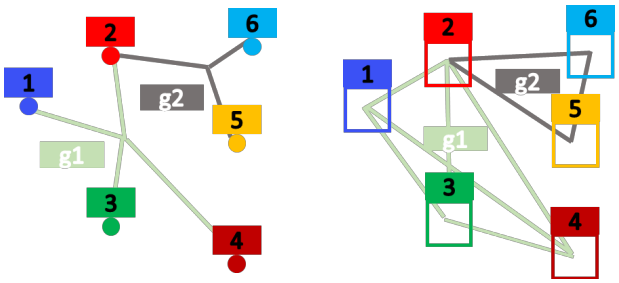
4. Comparison between ViTBAT and ViPER

We performed a comparison between ViTBAT and the widely-used ViPER tool based on subjective judgments on a series of criteria (C1 to C7) (as listed in Table 2) that are designed to reflect desired characteristics of user friendliness, annotation quality and annotation effort in a tool [2]. For a preliminary comparison we asked a set of five human subjects to participate in the assessment. All subjects exhibited a substantial knowledge of object tracking and behavior analysis in videos, and a good working understanding of ViPER. Each subject was provided with a uniform set of written instructions and asked to perform the assessment in a simple evaluation form by rating the two tools against each criterion (C1 to C7) by assigning a score between 1 to 5: '5' corresponds to the best score and '1' corresponds to the worst score. In order to perform the assessment the subjects are required to have a working understanding of ViTBAT. To this end we also provided the subjects with ViTBAT software and a sample video sequence together with a detailed video tutorial (made available on the ViTBAT website: https://vitbat.weebly.com) and a one-page step-by-step user guide on how to use the tool. The assessment was not time restricted.



(a) (b)

Figure 8. Visualization of targets' state annotation using (a) point-based representation in a sample image containing ants and (b) area-based representation in a sample image showing persons in a basketball court.



(a) (b)

Figure 9. Visualization of groups' state annotation either (a) by connecting every group member to the mean of group members state or (b) by connecting every group member state to the rest of group members.

Figure 10. Example of the annotated behaviors of targets on UT-interaction dataset [21]: left: image snapshot; right: annotated behavior display.

Fig. 11 shows the average scores assigned by all subjects to each tool on C1 to C7. On all criteria (C1 to C7) ViTBAT obtains a higher score than ViPER. The overall mean score of ViTBAT on all criteria is hence better than ViPER. We also checked the statistical significance of the scores obtained by the two tools on C1 to C7. We used the two-sample t-test as there are two groups of data each containing a set of seven scores (shown in Fig. 11). Statistical significance is achieved at the standard 5% significance level.

The ease of understanding how to use ViTBAT is due to its user-friendly interface as well as the provided simple video tutorial and a user guide. The ease of annotating states and behaviors comes from the explicit interface features for individual as well as group targets in the tool. Additionally, ViTBAT offers an effective separate visualization for annotated states as well as behaviors of individual/group targets by using a distinctive coloring schema that facilitates distinguishing among targets even in crowded scenes (Fig. 12). The image zoom in/out feature facilitates an accurate annotation particularly for small targets and further allows playing/replaying the video in zoomed form. Moreover, the minimization of annotation effort in ViTBAT comes due a more effective use of linear interpolation between a sparse set of annotations selected as reference by the user. The use of linear interpolation is shown to work well in most common scenes [27]. Furthermore, ViTBAT is made to annotate the state/behavior of a single target/group at a time. This allows the users to focus their attention on the target of interest as it might be difficult to follow movements of multiple targets at a time particularly in crowded and highly dynamic scenes. However, the user can easily switch among different targets/groups anytime.

Table 2. List of criteria for subjective assessment of ViTBAT and ViPER.

Criterion	Description
C1	Ease of understanding how to use the tool
C2	Simplicity/friendliness of the user interface
C3	Ease of annotating states of individual targets as well as groups
C4	Ease of annotating behaviors of individual targets as well as groups
C5	Ease of correcting wrong annotations
C6	Quality of annotation visualization
C7	Effort needed for performing annotation

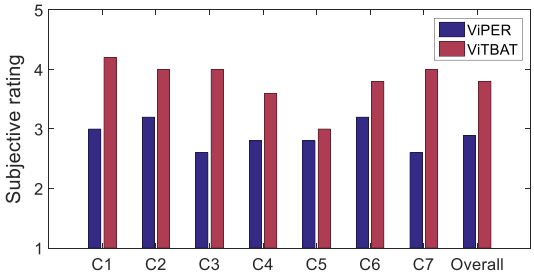


Figure 11. Subjective comparison between ViPER and ViTBAT on a series of criteria (C1 to C7) as listed in Table 2.

5. Conclusions

This paper presented a new tool (ViTBAT) that, unlike existing tools, enables a more explicit annotation of states and behaviors of both individual and group targets. On a series of criteria covering the key features of user friendliness, annotation quality and annotation effort, ViTBAT is found to be more effective than the well-known ViPER tool when judged by a sample of skilled people in a statistically significant initial subjective assessment. The dedicated ViTBAT website (https://vitbat.weebly.com) makes available the tool to the research community, which also offers a detailed video tutorial on its usage. The website is aimed to serve as a platform to interact with users and, as a future work, we would continue to improve the tool based on the received users' feedback. Moreover, in the future we also intend to perform a more comprehensive comparison of ViTBAT with ViPER (and other related tools) with a much larger and a more diverse sample of human subjects.

(a)

(b)

Figure 12. (a) Example of point-based annotation of states of small targets state in a crowded scene from Train Station dataset [30]: left: full image; right: zoomed-in part of image. (b) Example of groups' targets annotations in a crowded scene from Students003 dataset [25]: left $k = 1$; right: $k = 160$.

References

[1] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[2] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini. An interactive tool for manual, semi-automatic and automatic video annotation. *Computer Vision and Image Understanding*, 131:88–99, February 2015.

[3] T. A. Biresaw, A. Cavallaro, and C. S. Regazzoni. Correlation-based self-correcting tracking. *Neurocomputing*, 152(0):345 – 358, March 2015.

[4] T. A. Biresaw and C. S. Regazzoni. A Bayesian network for online evaluation of sparse features based multitarget tracking. In *Proc. of IEEE Int. Conf. on Image Processing*, pages 429–432, September 2012.

[5] D. Comaniciu, V. Ramesh, P. Meer, S. Member, and S. Member. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[6] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. de Polavieja, L. P. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, and U. Brose. Automated image-based tracking and its application in ecology. *Trends in ecology & evolution*, 29(7):417–428, 2014.

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[8] D. Doermann and D. Mihalcik. Tools and techniques for video performance evaluation. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, volume 4, pages 167–170, Sep 2000.

[9] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. of Int. Conf. on Computer Communications and Networks*, pages 65–72, October 2005.

[10] C. Garate, S. Zaidenberg, J. Badie, and F. Bremond. Group tracking and behavior recognition in long video surveillance sequences. In *Proc. of Int. Conf. on Computer Vision Theory and Applications*, volume 2, pages 396–402, January 2014.

[11] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Trans. on Signal Processing*, 50(2):425–437, 2002.

[12] J. Halberstadt, J. C. Jackson, D. Bilkey, J. Jong, H. Whitehouse, C. McNaughton, and S. Zollmann. Incipient social groups: An analysis via in-vivo behavioral tracking. *PLoS ONE*, 11(3):1–14, 3 2016.

[13] T. Holy. Generate maximally perceptually-distinct colors. http://www.mathworks.com/matlabcentral/fileexchange/29702. Last accessed: March 2016.

[14] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods*, 10(1):64–67, Jan 2013.

[15] M. J. V. Leach, R. Baxter, N. M. Robertson, and E. P. Sparks. Detecting social groups in crowded surveillance videos using visual attention. In *Proc. of Conf. on Computer Vision and Pattern Recognition Workshops*, June 2014.

[16] A. Li, M. Lin, Y. Wu, M. H. Yang, and S. Yan. Nus-pro: A new visual tracking challenge. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(2):335–349, February 2016.

[17] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *Proc. of Conf. on Computer Vision and Pattern Recognition Workshops*, pages 735–742, June 2013.

[18] T. Nawaz, F. Poiesi, and A. Cavallaro. Measures of effective video tracking. *IEEE Trans. on Image Processing*, 23(1):376–388, January 2014.

[19] T. H. Nawaz. *Ground-truth-based trajectory evaluation in videos*. PhD thesis, Queen Mary University of London, UK, 2014.

[20] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June 2010.

[21] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010. Last accessed: March 2016.

[22] J. C. SanMiguel, A. Cavallaro, and J. M. Martinez. Adaptive online performance evaluation of video trackers. *IEEE Trans. on Image Processing*, 21(5):2812–2823, 2012.

[23] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[24] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. on Signal Processing*, 56(8):3447–3457, August 2008.

[25] J. Sochman and D. C. Hogg. Who knows who - inverting the social force model for finding groups. In *Proc. of Int. Conf. on Computer Vision Workshops*, Barcelona, 2011.

[26] F. Solera, S. Calderara, and R. Cucchiara. Socially constrained structural learning for groups detection in crowd. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(5):995–1008, May 2016.

[27] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *Int. Journal of Computer Vision*, 101(1):184–204, January 2013.

[28] H. Wu, A. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1443–1458, August 2010.

[29] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 1451–1458, September 2009.

[30] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, Providence, 2012.