PRECISE REGRESSION FOR BOUNDING BOX CORRECTION FOR IMPROVED TRACKING BASED ON DEEP REINFORCEMENT LEARNING

Yifan Jiang *Hyunhak Shin* *Hanseok Ko*

School of Electrical and Computer Engineering, Korea University, Seoul, Korea

ABSTRACT

In this paper, we propose a precise regression approach for correcting imprecise bounding box using deep reinforcement learning. Object tracking task essentially builds trajectory of a moving object based on detection and tracking algorithms and its current state is indicated by having the object encapsulated with a bounding box corresponding to its position and size. However due to the imperfect detection and tracking algorithms operating in complex scene, it is difficult to obtain the precise bounding box as errors frequently occur producing oversized, partial, and false bounding box, respectively. To correct the error, we train an intelligent agent that move the bounding box to the right position and scale it to its correct size matching to that of the true target. The agent is trained by deep Q-learning and evaluated on several state-of-the-art multiple object tracking approaches. The experimental results demonstrate that our proposed framework can effectively eliminate the object tracking bounding box error and its robustness is verified by realizing improved tracking performance in complex scene.

Index Terms— Bounding box, regression, reinforcement learning, object tracking

1. INTRODUCTION

Object tracking task in video sequence is a crucial problem for many real time computer vision applications such as video surveillance, robot navigation, and driverless vehicles [1, 2, 3]. While object tracking focuses on estimating the targets current state by building its trajectory over time, the result is typically represented with a bounding box which indicates the position and size of the target. However, the state-of-art tracking accuracy is still limited to poor performance due to presence of complex scenes and frequent change of target appearance. Poor performance of object tracking can be attributed largely by the objects encapsulation with inaccurate bounding box in the form of oversized, partial and false position as shown in Figure 1.

To mitigate the poor tracking problem, several methods have been proposed to correct the inaccurate bounding boxes. Most of them are based on regression approaches. For these methods, the bounding box with error is typically given and then the algorithm sequentially moves the bounding box towards its true position step by step. Also, coarse-to-fine optimization methods can be considered as another kind of error correction approach since they use a regression approach to localize target, but their algorithms starting point is not the error location but a relatively large bounding box region that includes the intended target, then it tries to sequentially move and zoom in closer to the target. Recently, such regression approaches [4, 5, 6, 7, 8, 9] have been widely used in various object detection tasks.

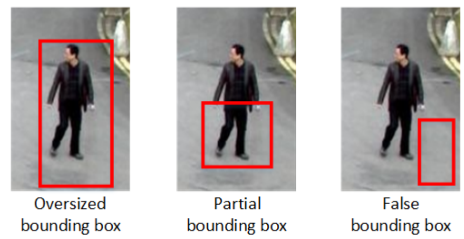


Fig. 1. Three categories of bounding box error.

Regression based bounding box correction approaches [4, 5, 6, 7] generally focus on reducing the error between detection based bounding box position and true target position. For these approaches, correction starts from the bounding box region with error and moves toward the direction whose distance is close to the ground truth which is provided by an object detection algorithm. Then, the correction algorithm proceeds with the iteration to move the bounding box toward the target location step by step, finally stopping at the optimal position. Thanks to advancement of convolution neural networks (CNNs) at full speed recently, bounding box correction schemes can exploit the deep features which are extracted by CNNs to capture and map the bounding box region better and find an optimal path to get closer to target quickly [4, 5, 6]. However, almost all of these approaches are based on a simple linear regression model that makes them weak to find the best path and leads to a non-convergence.

In order to overcome the drawbacks brought by the simple linear regression model, the reinforcement learning based object detection approaches [8, 9] have been developed. The regression usually starts from the whole image or a relatively large region. For these regression methods, reinforcement

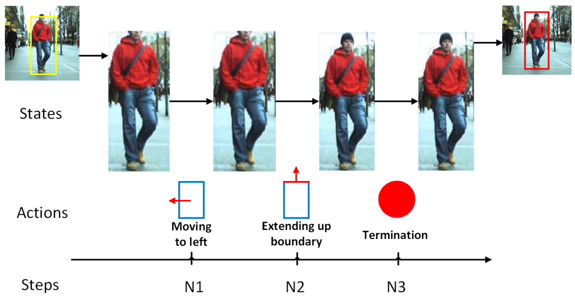


Fig. 2. An example regression progress for correcting an error tracking result with a sequence of actions.

learning techniques give them a better fitting performance when facing with complex scenes. But they tend to spend more steps to reach its optimal position since they ignore the information between frames. In short, rather than being exploited in the field of object detection, regression methods are more suitable for object tracking task since information from previous frame can be fully utilized to initiate a regression procedure.

In this paper, we propose a robust bounding box regression method with deep reinforcement learning that learns to correct error bounding box for multiple object tracking task in consideration of the aforementioned limitations of previous methods. To avoid the shortage of simple linear model, a novel deep Q-network is used to learn the way of finding out optimal regression policy efficiently. Considering the characteristics of object tracking task, we present a sample generation method to formulate the deep reinforcement model to object tracking bounding box regression problem. Also, the precise actions and unique reward mechanism are developed to reach to pixel-level regression accuracy. Finally, since our proposed method has different starting point selection mechanism compared to the reinforcement learning based object detection methods [8, 9], we compare the performance of the proposed method with the bounding box correction approaches in [4, 5, 6, 7].

2. PROPOSED METHOD

Error bounding box correction task essentially can be modeled as a framework of Markov decision process (MDP) because the resulting outcome is partly random and partly under the control of a decision maker. We can exploit this hypothesis to model an agent to make the sequence of decisions. We set a single bounding box region as environment (or observation), so that the agent can make actions to move the bounding box according to the environment. Our proposed method follows a neighborhood search strategy, which starts from a random region near by previous target location and then adjusts position and size to correct target. Figure 2 illustrates a part of the regression process during tracking a pedestrian.

2.1. MDP formulation

Several important parameters in the proposed MDP formulation are as follows.

Actions There are 13 possible actions which can be categorized into movement actions (e.g. 4 actions), scale actions (e.g. 8 actions) and termination action (e.g. 1 action). Movement actions represent moving the bounding box horizontally or vertically for one pixel. Scale actions indicate extending or shrinking one pixel for either boundary of bounding box. Lastly, termination action indicates that the bounding box has already reached correct place and the regression process should be stopped.

States States in our work can be divided into two parts; feature vector and memory vector. The feature vector is the Pool5 layer feature map of VGG-16 [10] from current bounding box region. The memory vector consists of the last 10 actions which the agent has already performed in search for an object. Since past 10 actions are encoded with one-hot format, the memory vector is represented as a 130 dimension vector.

Reward Reward strategy of the proposed method closely follows the Caicedo and Juans work [8]. To adjust the object tracking task, a specific case is needed. Hence, we set the threshold with a constant and $\tau = 0.9$, while other parameters stay the same as in [8].

2.2. Deep Q-learning for bounding box regression

In order to correct the error bounding box for object tracking tasks, the main problem is to establish an optimal policy for the agent to regress the incorrect bounding box to its correct position. Hence, this problem is suitable for being formulated with a reinforcement learning framework which can be solved by Q-learning method [11].

Following the Q-learning method, agent makes decisions according Q value $Q^\pi(s_t, a_t)$ which is determined by current state s_t and chosen action a_t. With Bellman equation (1), Q value can be represented as an iteratively updating format, where r_t is the received reward after taking action a_t, γ is a discount factor and $\max_{a_{t+1}} Q^\pi(s_{t+1}, a_{t+1})$ term indicates future reward, finally π is the optimal policy which learned at training stage. Due to the insurmountable curse of dimensionality, we employ the recently enhanced CNN based deep Q-learning (DQN) method developed by Mnih *et al.* [12].

$$Q^\pi(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q^\pi(s_{t+1}, a_{t+1}) \qquad (1)$$

Figure 3 depicts the proposed Q-network model. In this model, a pre-trained VGG16 model is used to extract the features from input image. Employing a pre-trained CNN model makes the policy learning process go faster than using the original image as input. Then an action memory vector combines the input image with the feature map from Pool5 layer of VGG16 model and feeds the result into the Q-network.

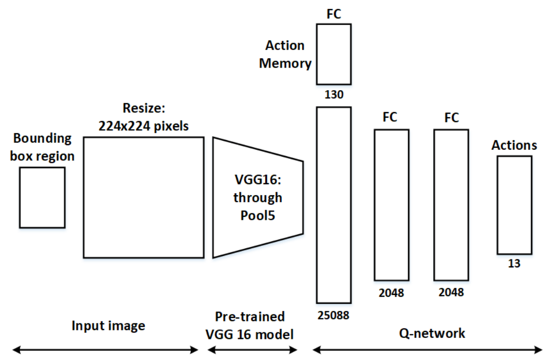


Fig. 3. Bounding box regression model.

Finally, since higher regression accuracy needs with higher action complexity require the proposed model generate better fitting performance, we increase the size of Q-network to 2048 units rather than 1024 units in [8, 9].

2.3. Training bounding box regression agent

Samples generation Object in reality tends to move smoothly. That means the object is close in distance and its size would change slightly if it is located in neighbor frames in a video. In order to generate sufficient and high quality training samples, we follow this motion smoothness idea which was introduced by Held *et al* [13]. Under this motion smoothness hypothesis, changes of object location and size obey Laplace distribution with mean of 0 and 1 respectively. Also, filtering of unreasonable samples would be necessary, such as those samples being too small or too far away from previous location.

Training strategy Every training iteration starts from a randomly sample generation, ϵ-greedy policy [11] is used to enlarge path searching range by randomly choosing actions. Here, it is initiated with 1 then decreased to 0.1 by steps of 0.05 every 5 epochs. In order to explore for more available paths, the step limit is set as 100 at each iterative process.

DQN parameters Experiment replay method is widely used to train DQN model since it can avoid inefficient learning drawback from exploiting consecutive experience and help to make learning progress more stable. Here we set the replay pool size as 1000. Another important parameter is discount factor γ, which indicate how much the future information will be taken into consideration. Here we use a relative high γ which is 0.9 since there is a limitation step at each regression iteration and we only are interested in current rather than future rewards.

Training parameters The VGG16 model is pre-trained by ImageNet [14] database. Q-network is initiated randomly from a uniform distribution and trained with learning rate as 1e-6 and Adam [15] optimizer. Finally, each target-specific model is trained with 100 epochs and batch size of 100.

2.4. Testing bounding box regression agent

For testing, it is necessary to create a new DQN agent for each target since it might be totally different among the appearance of targets. Each regression iteration is limited under 100 steps to save processing time. If regression step is over 100 but terminal action still does not appear, we consider this regression iteration is failed. Otherwise, if the termination action comes out within 100 steps, we consider it as a successful iteration. In the experiment, we find that sometimes agent tends to be stuck into a local optimum (if the bounding box region takes the same pair of actions for 10 steps continuously) caused by high complexity of the environment. In this case, the bounding box region is randomly moved to an arbitrary direction of 5 pixels to get it out of the local optimum.

One regression procedure will be considered as successful when three times continuously successfully regression iteration takes place. The final corrected result would be set as the average of the three regression iteration results. If a regression procedure is failed, the tracking result would not be corrected and another agent goes on for next target.

In order to increase adaptability of new target which agent has never seen before, an online fine-tuning method is exploited. For every 10 frames of each specific target, the model would be updated with the same procedure as training stage. However, models share and fix their pre-trained CNN parameters and only update the parameters of Q-networks online.

3. EXPERIMENT RESULTS

We compare proposed method on several state-of-the-art multi-object tracking approaches with 2D MOT 2015 [16] dataset from MOT challenge to finish the comparison of performance. The experiments environment: GTX TITAN Xp GPU using Keras 2.0.6 with Tensorflow 1.3.0 backend.

3.1. Evaluation metrics

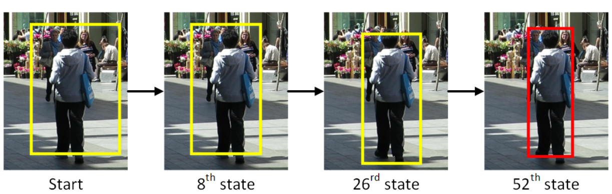
Here CLEAR MOT metrics [21] and a new ID switches evaluation metric [22] are used to evaluate the performance of trackers. Multiple object tracking precision (MOTA↑) and ID F1 Score (IDF1↑) reflect the integrated accuracy of tracking results from bounding box and IDs identification respectively. The mostly trajectories (MT↑) and the most lost trajectories (ML↓) indicate degree of overlap between tracking results and ground truths. HZ(↑) means the speed of tracker in frames per second. ↑ and ↓ each means that higher number and lower number is better respectively.

3.2. Results analysis

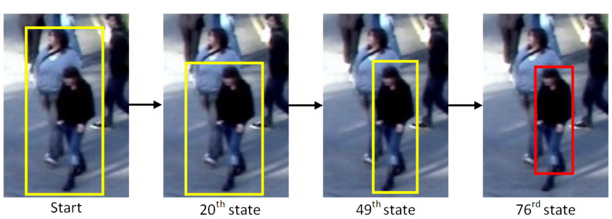
Table 1 shows the performance comparison of the proposed method and other regression methods on several state-of-the-art multi-object tracking approaches. As we can see, the method which we proposed can effectively improve the

Table 1. Performance comparison of original state-of-the-art multi-object tracking methods and methods with regression approach; while the best evaluation metric is in bold.

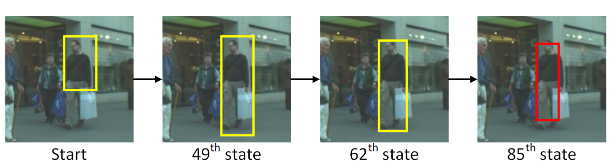
Object tracking method	Regression method	MOTA(%)	IDF1(%)	MT(%)	ML(%)	HZ
AMIR15 [17]	OURS	**40.1**	46.0	**18.4**	**23.0**	0.7
	Girshick, Ren *et al.* [4, 5, 7]	37.4	46.0	15.4	26.5	1.6
	He *et al.* [6]	38.6	46.0	17.9	25.7	1.5
	None	37.6	46.0	15.8	26.8	**1.9**
HybridDAT [18]	OURS	**42.3**	47.7	13.6	**39.7**	3.1
	Girshick, Ren *et al.* [4, 5, 7]	36.0	47.7	11.5	42.6	4.0
	He *et al.* [6]	37.4	47.7	**13.8**	40.0	4.0
	None	35.0	47.7	11.4	42.2	**4.6**
AM [19]	OURS	**39.8**	48.3	**15.1**	**39.9**	0.2
	Girshick, Ren *et al.* [4, 5, 7]	34.5	48.3	11.8	43.3	0.4
	He *et al.* [6]	36.6	48.3	14.0	43.0	0.4
	None	34.3	48.3	11.4	43.4	**0.5**
MDP [20]	OURS	**32.3**	44.7	**16.7**	**36.0**	0.5
	Girshick, Ren *et al.* [4, 5, 7]	32.0	44.7	11.8	39.7	0.8
	He *et al.* [6]	**32.3**	44.7	15.8	38.4	1.0
	None	30.3	44.7	13.0	38.4	**1.1**



(a) ADL-Rundle-3 160^{th} frame (AM [19])



(b) PETS09-S2L2 314^{th} frame (MDP [20])



(c) ETH-Linthescher 133^{rd} frame (AMIR15 [17])

Fig. 4. Example sequences of regression procedure.

precision of object trackers and make them corrected in challenging locations. In terms of MOTA, MT and ML, they represent the accuracy of tracking results. Our proposed method performs better than other competing conventional methods such that all the accuracy evaluation metrics show improved. For IDF1 metric, it evaluates the tracker distinguishing ability among targets. Because all the trackers give ID results before regression procedure so that regression has no influence on IDF1 metric. Every regression procedure has step limitation so the time consumption is acceptable.

Some examples of regression results are depicted in Figure 4.

4. CONCLUSIONS

We developed and presented a precise bounding box regression approach to correct imprecise bounding box so that tracking result becomes improved in object tracking task. In order to handle the bounding box errors, our proposed method employed deep reinforcement learning algorithm to learn about how to explore for the optimal regression path between error bounding box and ground truth. Experimental results indicate that the proposed regression method can correct error bounding box effectively and definitely increase the tracking accuracy of state-of-the-art object trackers.

Acknowledgements: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1A2B4012720).

5. REFERENCES

[1] Jaeyong Ju, Daehun Kim, Bonhwa Ku, David K Han, and Hanseok Ko, "Online multi-object tracking based on hierarchical association framework," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 34–42.

[2] Anton Milan, Stefan Roth, and Konrad Schindler, "Continuous energy minimization for multitarget tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, 2014.

[3] Jaeyong Ju, Daehun Kim, Bonhwa Ku, David K Han, and Hanseok Ko, "Online multi-person tracking with two-stage data association and online appearance model learning," *IET Computer Vision*, vol. 11, no. 1, pp. 87–95, 2016.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[5] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*. Springer, 2014, pp. 346–361.

[8] Juan C Caicedo and Svetlana Lazebnik, "Active object localization with deep reinforcement learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2488–2496.

[9] Miriam Bellver, Xavier Giró-i Nieto, Ferran Marqués, and Jordi Torres, "Hierarchical object detection with deep reinforcement learning," *arXiv preprint arXiv:1611.03718*, 2016.

[10] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, vol. 1, MIT press Cambridge, 1998.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[13] David Held, Sebastian Thrun, and Silvio Savarese, "Learning to track at 100 fps with deep regression networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 749–765.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[15] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, Apr. 2015, arXiv: 1504.01942.

[17] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," *arXiv preprint arXiv:1701.01909*, 2017.

[18] Min Yang, Yuwei Wu, and Yunde Jia, "A hybrid data association framework for robust online multi-object tracking," *arXiv preprint arXiv:1703.10764*, 2017.

[19] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," *arXiv preprint arXiv:1708.02843*, 2017.

[20] Yu Xiang, Alexandre Alahi, and Silvio Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4705–4713.

[21] Keni Bernardin and Rainer Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 246309, 2008.

[22] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 17–35.