

# Multiple Centroid-Based Multi-Object Tracking by Decision Making

Hao Zhou<sup>1,\*</sup>, Jian He<sup>2</sup> and Zhangqin Huang<sup>2</sup>

<sup>1</sup>Beijing University of Technology, China

<sup>2</sup>Xi'an Jiaotong University, China

\*Corresponding author

**Abstract**—Multiple object tracking(MOT) has many applications in different life scenes, such as autonomous driving and traffic management. A novel approach to object predictive tracking, which is based on the Multiple centroid, is proposed in this paper to overcome the problems of occlusion, scale changes and shape, which cause the situation to lose object. Firstly, we formulate the MOT problem as decision making in Markov Decision Processes, where the lifetime of an object is modeled with a Markov Decision Process. Secondly, the object region is selected as the tracking region in the video to initialize the tracking process. Several image sampling segments are made ready for counting the initial pixels with the gray of a continuous gradient features pixels, which have the same gray-scale changes in intensity, and the discrete pixel group centroid(center centroid and edge cen-troid) coordinate is fitting for a time-domain trajectory curve. Edges of spectrum and track of spectrum are calculated by the time-domain trajectory curve. thirdly, we also model the lifetime of every centroid of every object with a Markov Decision Process. Finally, we compare both states and spectrum of each centroid and find the best trajectory. We conduct experiments on the MOT Benchmark[1] to verify the effectiveness of our method.

**Keywords**—multiple object tracking; multiple centroid; edges of spectrum; track of spectrum

## I. INTRODUCTION

Multiple object tracking is one of the hot topics in the field of computer vision. It is a technology used to object identification, object location information extraction and analysis, object motion tracking from the image signal in real time. It has a very wide range of application in military industrial, security, and other fields; the development prospects are bright. Object motion randomness and complexity exist in the real environment, so its theoretical and applied research has some challenging. At present, there are still many problems to be resolved[2,3]. In order to resolve ambiguities in associating object detections and to overcome detection failures, most of these recent works [4, 5, 6, 7] process video sequences in a batch mode in which video frames from future time steps are also utilized to solve the data association problem.

Recently, there is a trend on learning to track that advocates the concept of injecting learning capabilities to MOT [8, 9]. Based on their learning schemes, we can categorize these methods into offline-learning methods and online-learning methods. In offline-learning, learning is performed before the actual tracking takes place. For instance, use super-vision from ground truth trajectories offline to learn a similarity function between detections and tracking for data association. As a result,

offline learning is static: it cannot take into account the dynamic status and the history of the target in data association, which is important to resolve ambiguities, especially when it needs to reassign missed or occluded objects when they appear again. In contrast, online-learning conducts learning during tracking. In multiple object tracking, problems are found such as too much time consuming in data processing and track loss of complex background. To elevate these problems, the multiple centroid theory of moving object was proposed in this work, concepts and definitions are given, and object tracking process based on multiple centroid is described.

## II. MULTIPLE CENTROID

The first concern is tailing problem of object motion in space. Assuming that the moving object is expressed by rectangle template, denoted as  $M$ , the lengths of the template are 2a and 2b. In three-dimensional coordinate system  $(x, y, z)$ , the tailing trajectory of rectangle template centroid is calibrated at time  $t$  the time-domain function of centroid trajectory  $\varepsilon$  is defined as  $\varepsilon = \tau(x, y, z, t)$ , and also at time-domain function  $l = v(x, y, z, t)$  represents the edge centroid trajectory  $l$ . The rectangle template is described by  $M_{x,y,z,t}$ , after a moving time of  $\Delta t$ , the template is  $M_{x,y,z,t+\Delta t}$ . Centroid during time  $\Delta t$  moves along the function  $\varepsilon$  from template  $M_{x,y,z,t+\Delta t}$  to  $M_{x,y,z,t+\Delta t}$ . After the moving, the left vertical lines of two templates are overlap. The relevant definitions and concepts are given as follows. The model diagram is shown in Figure 1.

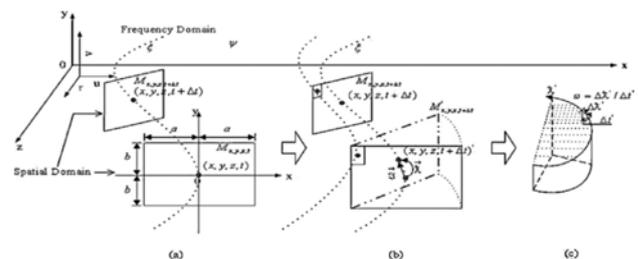


FIGURE I. MODEL CHART

### A. Definition 1

In spatial domain, the vector between two centers of mass is defined as spatial vector, denoted as  $\mathbf{3}$ , the modulus of spatial vector is  $|\mathbf{3}|$ .

B. Definition 2

In frequency domain  $\xi$ , the vector between two centers of mass is defined as frequency vector, denoted as  $\lambda$ , and the modulus of it is  $|\lambda|$ .

C. Definition 3

Precedent impact strength of moving target is assumed as  $\omega$ , where  $\omega = \Delta\lambda/\Delta t$ .

D. Definition 4

Centroid motion of the center of gravity formed the periodic tailing trajectory spectrum, it is defined as space spectrum, denoted by  $\zeta$ . By Fourier transform time-domain orbit  $\varepsilon$  of centroid obtains frequency domain spectrum, it can be used to represent  $\zeta$ :

$$\xi(u, v, r) = \int_0^{2a} \int_0^{2b} \int_0^{|\lambda|} \varepsilon(x, y, z) e^{-2\pi j(\frac{ux}{2a} + \frac{vy}{2b} + \frac{rz}{|\lambda|})} dx dy dz \quad (1)$$

E. Definition 5

Centroid motion of each edge formed the periodic (uniform or non-uniform) tailing trajectory spectrum, it is defined as edge spectrum, denoted by  $\psi$ . By Fourier transform time-domain orbit  $l$  of edge centroid obtains frequency domain spectrum, it can be used to represent  $\psi$ :

$$\psi^i(u, v, r) = \int_0^{2a} \int_0^{2b} \int_0^{|\lambda|} v^i(x, y, z) e^{-2\pi j(\frac{ux}{2a} + \frac{vy}{2b} + \frac{rz}{|\lambda|})} dx dy dz \quad (2)$$

Where  $i$  represents the number of edge center of mass, and  $i \in [0, 4ab]$ ,  $j = \sqrt{-1}$ , and  $u, v, r$  are spectrum vectors, among them  $u \in [0, 2a]$ ,  $v \in [0, 2b]$ ,  $r \in [0, |\lambda|]$ .

F. Definition 6

The frequency which changes continuously and flexibly in space spectrum and edge spectrum is defined as soft feature, denoted by  $T$ . The value range of  $T$  is the differentiable range of space spectrum  $\zeta(u, v, r)$  and edge spectrum  $\psi^i(u, v, r)$ :

$$T = \begin{bmatrix} (u_{1,1}, v_{1,1}, r_{1,1}) & \cdots & (u_{m,1}, v_{m,1}, r_{m,1}) \\ \vdots & \ddots & \vdots \\ (u_{1,n}, v_{1,n}, r_{1,n}) & \cdots & (u_{m,n}, v_{m,n}, r_{m,n}) \end{bmatrix} \times \cup \sum_{i=1}^q \begin{bmatrix} (u_{1,1}^i, v_{1,1}^i, r_{1,1}^i) & \cdots & (u_{q,1}^i, v_{q,1}^i, r_{q,1}^i) \\ \vdots & \ddots & \vdots \\ (u_{1,p}^i, v_{1,p}^i, r_{1,p}^i) & \cdots & (u_{q,p}^i, v_{q,p}^i, r_{q,p}^i) \end{bmatrix}, i \in [0, 4ab] \quad (3)$$

### III. MARKOV DECISION PROCESS

In our framework, the lifetime of a target is modeled with a Markov Decision Process (MDP). The MDP consists of the tuple  $(S, A, T(\cdot), R(\cdot))$ :

- The target state  $s \in S$  encodes the status of the target.
- The action  $a \in A$  which can be performed to a target.
- The state transition function  $T : S \times A \rightarrow S$  describes the effect of each action in each state.

- The real-valued reward function  $R : S \times A \rightarrow R$  defines the immediate reward received after executing action  $a$  to state  $s$ .

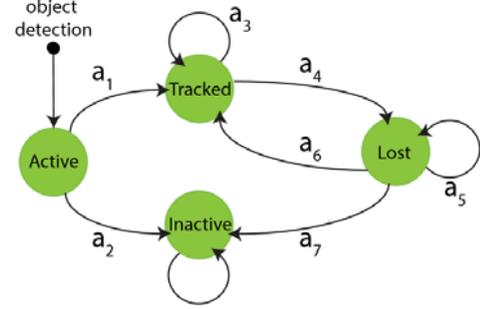


FIGURE II. THE TARGET MDP IN OUR FRAMEWORK

A. States

We partition the state space in the target MDP into four subspaces, i.e.,  $S = S_{Active} \cup S_{Tracked} \cup S_{Lost} \cup S_{Inactive}$ , where each subspace contains infinity number of states which encode the information of the target depending on the feature representation, such as appearance, location, size and history of the target. Figure 2 illustrates the transitions between the four subspaces. “Active” is the initial state for any target. Whenever an object is detected by the object detector, it enters an “Active” state. An active target can transition to “Tracked” or “Inactive”. Ideally, a true positive from object detector should transition to a “Tracked” state, while a false alarm should enter an “Inactive” state. A tracked target can keep tracked, or transition to “Lost” if the target is lost due to some reason, such as occlusion, or disappearance from the field of view of the camera. Likewise, a lost target can stay as lost, or go back to “Tracked” if it appears again, or transition to “Inactive” if it has been lost for a sufficiently long time. Finally, “Inactive” is the terminal state for any target, i.e., an inactive target stays as inactive forever.

B. Actions and Transition Function

Seven possible transitions are designed between the states of a target, which correspond to seven actions in our target MDP. Figure 2 illustrate these transitions and actions. In the MDP, all the actions are deterministic, i.e., given the current state and an action, we specify a new state for the target. For example, executing action  $a_4$  on a tracked target would transfer the target into a lost state, i.e.,  $T(s_{Tracked}, a_4) = s_{Lost}$ .

C. Reward Function

In our MDP, the reward function is not given but needs to be learned from training data, i.e., an inverse reinforcement learning problem [10], where we use ground truth trajectories of the targets as supervision.

### IV. POLICY

In MDP, a policy  $\pi$  is a mapping from the state space  $S$  to the action space  $A$ , i.e.,  $\pi : S \rightarrow A$ . Given the current state of the target, a policy determines which action to take. Equivalently, the decision making in MDP is performed by following a policy. The goal of policy learning is to find a policy which maximizes the total rewards obtained. In this section, we first describe our policies designed for the Active subspace and the Tracked

subspace, then we present a novel reinforcement learning algorithm to learn a good policy for data association in the Lost subspace.

#### A. Policy in an Active State

In an Active state  $s$ , the MDP makes the decision between transferring an object detection into a tracked or inactive target to deal with noisy detections. This decision making can be considered to be a preprocessing step before tracking. Strategies such as non-maximum suppression or thresholding detection scores are usually used. In our implementation, we train a binary Support Vector Machine (SVM) [11] offline to classify a detection into tracked or inactive using a normalized 5D feature vector  $\Phi_{Active}(s)$ , i.e., 2D coordinates, width, height and score of the detection, where training examples are collected from training video sequences. This is equivalent to learning the reward function in Active:

$$R_{Active}(s, a) = y(a)(w_{Active}^T \phi_{Active}(s) + b_{Active}) \quad (4)$$

#### B. Policy in a Tracked State

In a Tracked state, the MDP needs to decide whether to keep tracking the target or to transfer it into a lost state. As long as the target is not occluded and is in the camera's field of view, we should keep tracking it. Otherwise, it should be marked as lost. This decision making is related to the goal of single object tracking in the literature [12, 13, 14, 15]. Inspired by these works, we build an appearance model for the target online and use it to track the target. If the appearance model is able to successfully track the target in the next video frame, the MDP leaves the target in a tracked state. Otherwise, the target is transferred to a lost state. Our framework is general to utilize different approaches in building the appearance model. We describe our implementation based on the TLD tracker [14] in this work. This is equivalent to learning the reward function in Tracked:

$$R_{Tracked}(s, a) = \begin{cases} y(a), & \text{if } e_{medFB} < e_0 \text{ and } o_{mean} > o_0 \\ -y(a), & \text{otherwise} \end{cases} \quad (5)$$

#### C. Policy in a Lost State

In a Lost state, the MDP needs to decide whether to keep the target as lost, transition it to a tracked state, or mark it as inactive. We simply mark a lost target as inactive and terminate the tracking if the target has been lost for more than  $T_{Lost}$  frames. The challenging case is to make the decision between tracking the target and keeping it as lost. We treat it as a data association problem: in order to transfer a lost target into a tracked state, the target needs to be associated with one of the detections from the object detector, otherwise, the target is kept as lost. This is equivalent to learning the reward function in Lost:

$$R_{Lost}(s, a) = y(a) \left( \max_{k=1}^M \phi(t, d_k) + b \right) \quad (6)$$

### V. REVISE AND COMBINE

The flow-process diagram is shown in Figure 3.

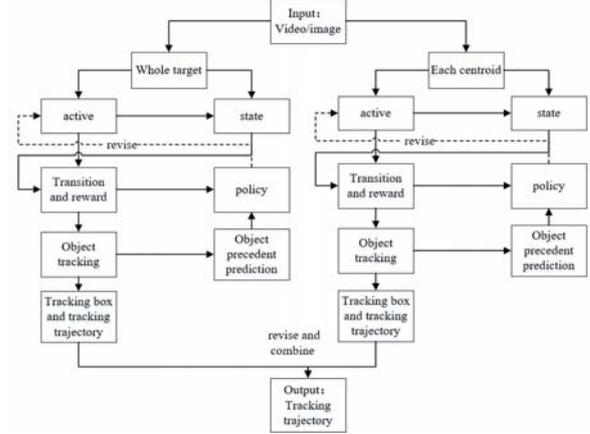


FIGURE III. THE FLOW-PROCESS DIAGRAM IN OUR FRAMEWORK.

### VI. EXPERIMENT AND ANALYSIS

Figure 4 shows the tracing effect of three moving persons. In this figure, there are some illustrations with representative results of continuous video tests. With different shooting angles, feather occlusion is produced when three objects in the test(NO.187, No.295). As the three moving objects merge a whole object visually, when edge separating of objects appears, High-dimensional Statistical Measure(HSM) and Probabilistic Fusion(PF) tracking algorithm of object occlusion proposed in [16,17] lose the objects. With the algorithm of precedent predication tracking based on soft feature put forward in this paper, discrete convolution effect of precedent impact strength is considered in the process of tracking, so locations of the moving objects are tracked and predicted accurately in the case of object occlusion(NO.179,NO.306).

Table 1 shows our tracking performance on the test set, where we compare our tracker (MDP REinforcement Learning, MDP REL) with the state-of-the-art methods tested on the MOT benchmark. As we can see from the table, our tracker improves 7% in MOTA compared with the second best published tracker, and achieves the best performance in terms of mostly tracked and mostly lost targets even though it works in the online mode. The superior performance demonstrates the advantage so four learning to track strategy with MDPs. Figure 5 shows sampled tracking results on the 11 sequences in the test set.

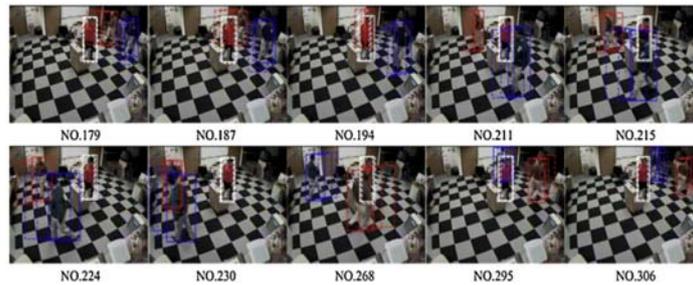


FIGURE IV. EXPERIMENT RESULTS OF THREE OBJECT TRACKING THROUGH SIMILAR BACKGROUND.

TABLE I. TRACKING PERFORMANCE ON THE TEST SET OF THE MOT BENCHMARK. MORE COMPARISONS ARE AVAILABLE AT [1]

Tracker	Tracking Mode	Learning Mode	MOTA	MOTP	MT	ML	FP	FN	IDS	Frag	Hz
DP.NMS [18]	Batch	N/A	14.5	70.8	6.0%	40.8%	13,171	34,814	4,537	3,090	<b>444.8</b>
TC.ODAL [19]	Online	Online	15.1	70.5	3.2%	55.8%	12,970	38,538	<b>637</b>	1,716	1.7
TBD [20]	Batch	Offline	15.9	70.9	6.4%	47.9%	14,943	34,777	1,939	1,963	0.7
SMOT [21]	Batch	N/A	18.2	71.2	2.8%	54.8%	8,780	40,310	1,148	2,132	2.7
RMOT [22]	Online	N/A	18.6	69.6	5.3%	53.3%	12,473	36,835	684	1,282	7.9
CEM [6]	Batch	N/A	19.3	70.7	8.5%	46.5%	14,180	34,591	813	1,023	1.1
SegTrack [23]	Batch	Offline	22.5	<b>71.7</b>	5.8%	63.9%	<b>7,890</b>	39,020	697	<b>737</b>	0.2
MotiCon [7]	Batch	Offline	23.1	70.9	4.7%	52.0%	10,404	35,844	1,018	1,061	1.4
MCMDP OFL	Online	Offline	30.1	71.6	10.4%	41.3%	8,789	33,479	690	1,301	0.8
MCMDP REL	Online	Online	<b>30.7</b>	71.3	<b>13.2%</b>	<b>38.4%</b>	9,717	<b>32,424</b>	681	1,500	1.1



FIGURE V. TRACKING RESULTS ON THE TEST SEQUENCES IN THE MOT BENCHMARK.

#### ACKNOWLEDGMENT

We thank Xiaohan Du for helpful discussions.

#### REFERENCES

- [1] Multiple object tracking benchmark. <http://motchallenge.net>. 8
- [2] M. Rezaeian, B.N. Vo, Error bounds for joint detection and estimation of a single object with random finite set observatio, *IEEE Trans: Signal Process.* 58(3) (2010)1493-1506.
- [3] D. Clark, B. Ristic, B.N. Vo, B.T. Vo, Bayesian multi-object filtering with amplitude feature likelihood for unknown object SNR, *IEEE Trans. Signal Process.* 58 (1) (2010) 26-37.
- [4] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806-1819, 2011.1, 2
- [5] A. A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *CVPR*, pages 186–183, 2013. 1.
- [6] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *TPAMI*, 36(1):58–72, 2014. 1, 2, 8.
- [7] L. Leal-Taix'e, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, pages 3542–3549, 2014. 1, 8
- [8] X.Song, J.Cui, H.Zha, and H.Zhao. Vision-basedmultipleinteracting targets tracking via on-line supervised learning. In *ECCV*, pages 642–655. 2008. 1, 2.
- [9] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960, 2009. 1, 2.
- [10] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *ICML*, pages 663–670, 2000. 3.
- [11] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992. 3.
- [12] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *TPAMI*, 33(8):1619–1632, 2011. 2, 3.
- [13] S.Hare, A.Saffari, andP.H.Torr. Struck: Structuredoutputtracking with kernels. In *ICCV*, pages 263–270, 2011. 2, 3, 4.
- [14] Z.Kalal, K.Mikolajczyk, and J.Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2012. 2, 3, 4.
- [15] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using acceleratedproximalgradientapproach. In *CVPR*,pages1830–1837, 2012. 2, 3, 4.
- [16] S. Boltz, E. Debrueve, M. Barlaud, High-dimensional statistical measure for region-of-interest tracking, *IEEE Trans. Image Process.* 18 (6) (2009) 1266–1283.
- [17] B.H. Han, L.S. Davis, Probabilistic fusion-based parameter estimation for visual tracking, *Comput. Vision Image Und.* 113 (2009) 435–445.
- [18] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201–1208, 2011. 2, 8.
- [19] S.-H.Baeand K.-J.Yoon. Robustonlinemulti-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, pages 1218–1225, 2014. 1, 2, 8.
- [20] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *TPAMI*, 36(5):1012–1025, 2014. 8.
- [21] C.Dicle, O.I.Camps, and M.Sznaier. Thewaytheymove: Tracking multiple targets with similar appearance. In *ICCV*,pages2304–2311, 2013. 8.

- [22] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multiobject tracking using motion context from multiple objects. In WACV, pages 33–40, 2015. 8.
- [23] A.Milan, L.Leal-Taix'e, K.Schindler, and I.Reid. Joint tracking and segmentation of multiple targets. In CVPR, pages 5397–5406, 2015. 8.