

Research Article

Visual Object Tracking in RGB-D Data via Genetic Feature Learning

Ming-xin Jiang ¹, Xian-xian Luo ², Tao Hai,³ Hai-yan Wang,¹
Song Yang,¹ and Ahmed N. Abdalla¹

¹Jiangsu Laboratory of Lake Environment Remote Sensing Technologies, Huaiyin Institute of Technology, Huaian 223003, China

²Faculty of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou 362000, China

³Computer Science Department, Baoji University of Arts and Sciences, Shanxi 721031, China

Correspondence should be addressed to Ming-xin Jiang; jiangmingxin@126.com and Xian-xian Luo; luoxianxian@sina.com.cn

Received 26 December 2018; Revised 17 March 2019; Accepted 25 March 2019; Published 2 May 2019

Academic Editor: Marcin Mrugalski

Copyright © 2019 Ming-xin Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Visual object tracking is a fundamental component in many computer vision applications. Extracting robust features of object is one of the most important steps in tracking. As trackers, only formulated on RGB data, are usually affected by occlusions, appearance, or illumination variations, we propose a novel RGB-D tracking method based on genetic feature learning in this paper. Our approach addresses feature learning as an optimization problem. As owning the advantage of parallel computing, genetic algorithm (GA) has fast speed of convergence and excellent global optimization performance. At the same time, unlike handcrafted feature and deep learning methods, GA can be employed to solve the problem of feature representation without prior knowledge, and it has no use for a large number of parameters to be learned. The candidate solution in RGB or depth modality is represented as an encoding of an image in GA, and genetic feature is learned through population initialization, fitness evaluation, selection, crossover, and mutation. The proposed RGB-D tracker is evaluated on popular benchmark dataset, and experimental results indicate that our method achieves higher accuracy and faster tracking speed.

1. Introduction

Visual object tracking, as a hot research topic in computer vision, has been potential applications including intelligent surveillance systems [1], sport analysis [2], advanced assistance systems [3], etc. Despite its popularity, the existing trackers are mainly based on RGB information so that they have many limitations, how to accurately track the target object still remains challenging as variations in appearance or illumination, occlusions, and clutter [4–6]. As the introduction of low-cost RGB-D cameras, new algorithms that fuse color and depth cues to improve tracking performance have been proposed [7–9].

In general, most of trackers contain the following two parts: feature representation and search strategy. The existing methods of feature representation can be grouped into two categories: handcrafted features and deep learning features.

Song [10] proposed several different RGB-D trackers using both 2D and 3D model and released a large public

dataset which includes more than 100 videos. The best tracking performance was achieved by calculating HOG features on both color and depth data. In [11], authors proposed a real-time RGB-D tracker using depth information to handle scale and shape changes. Authors of [12] proposed an occlusion-aware RGB-D tracking approach based on particle filter which can handle complex and persistent occlusions. Authors of [13] proposed a tracker based on prototype budget maintenance; they explored local depth pattern to represent depth feature. A RGB-D tracker was presented by [14] using adaptive range-invariant depth models and spatiotemporal consistency constraints. The tracking methods introduced above are all based on handcrafted feature, which are usually designed by human experts and only achieve good performance in some particular domain.

To overcome these limitations, deep learning features have been applied to RGB-D tracker. A visual RGB-D tracker based on cross-modality Gaussian-Bernoulli deep Boltzmann Machines was introduced by [15]. Authors [16]

proposed a new RGB-D tracker which is upon kernelized correlation filter with deep features. In [17], an RGB-D hand tracking algorithm based on a deep learning framework was presented. A novel RGB-D tracker based on multimodal deep feature fusion was proposed by [18]. These RGB-D trackers are able to learn the features of objects automatically using deep learning; however, the number of parameters to be learned in these deep learning models is very large, which restricts their application.

As owning the advantage of parallel computing, evolutionary computation has fast speed of convergence and excellent global optimization performance. Within the field of evolutionary computation, genetic algorithm (GA) is the best known one that seeks the solution of a problem by simulating biological evolutionary mechanisms such as selection, crossover, and mutation. In recent years, evolutionary computation has been applied to the computer vision domain and proved to be powerful. Authors of [19] proposed an image segmentation algorithm based on a genetic framework. Lin et al. [20] proposed a coevolutionary genetic programming approach to learn composite features for object recognition. In [21], the authors performed an evolutionary optimization of hierarchical object recognition features.

Unlike handcrafted feature techniques, GA can be employed to solve the problem of feature representation without prior knowledge. At the same time, GA has no use for a large number of parameters to be learned. To the best of our knowledge, GA has not been utilized in RGB-D tracker by far. In this paper, we propose a visual object tracking method in RGB-D data via genetic feature learning. At first, the depth data is encoded into three channels of horizontal disparity, height above ground, and angle with gravity using HHA encoding for deriving geometrical information of target object. Then, candidate solution in RGB or depth modality is represented as an encoding of an image. Genetic feature is learned through population initialization, fitness evaluation, selection, crossover, and mutation. To fuse the information in RGB and depth modality, the sum of errors between genetic feature of target object in frame $t-1$ and the candidates in current frame are computed. Finally, the candidate with the minimum sum of errors in two modalities is output as the tracking result. The proposed RGB-D tracker is evaluated on popular benchmark dataset, and experimental results indicate that our method achieves a higher accuracy and faster tracking speed.

2. Proposed RGB-D Tracking Algorithm

At first, this section describes how to learn genetic feature for RGB-D data. Following that, the overall framework of our proposed RGB-D tracker is illustrated.

2.1. Individual Representation. As everyone knows, the geometric properties are very important for the primary mechanism of human visual system when recognizing or tracking the objects. Only using the plain RGB data, it is difficult to compute the geometric properties of objects with higher accuracy and greater reliability. It is fortunate that the depth

data can provide more accurate geometrical cues compared with only using RGB information. For deriving geometrical information explicitly, the depth is encoded into three channels of horizontal disparity, height above ground, and angle with gravity using HHA encoding [22] which emphasize the complementary discontinuities in the image.

In GA, encoding of chromosome is our first step to solve any problem. Following the work [23], we can know that individual, chromosome, candidate solution, or genotype is equivalent to represent the point in the space of possible solutions in GA domain. As after HHA encoding, the depth has three channels: disparity, height, and angle, which is similar to RGB image. In our paper, an individual/chromosome/candidate solution in RGB or depth modality includes 32 genes, each gene consists of 7 data items, and each data item contains 8 binary bits; accordingly, the dimension of a chromosome is 56×32 , as shown in Figure 1. The only difference is internal structure of data item in RGB or depth modality. Specifically, for a RGB image, the data item comprises three channels (R,G,B) and four values (X,Y,W,H) of a sliding window within a reconstructed image in GA domain, but for a depth image, three channels are disparity, height, and angle.

2.2. Genetic Feature Learning. The implementation of genetic feature learning is illustrated as Figure 2. Genetic feature learning comprises population initialization, fitness evaluation, selection, crossover, and mutation. Firstly, the initial population is generated randomly that is also called solution space. Then a candidate solution is generated within the solution space which is most adaptable to the environment according fitness evaluation. If this candidate solution meets termination condition, it will be output as the best individual. Otherwise, the crossover and mutation are performed on this candidate, then new population will be generated. Following that, a new candidate solution is selected from the new population which will be passed to the loop again. Finally, after a specific number of crossover, mutation, and selection of solutions, the GA will terminate and output the genetic features of the best individual.

2.2.1. Population Initialization. For an input image in RGB or depth modality X_{RGB} or X_{depth} , in the solution space, the population of individuals/chromosomes is expressed as $P_{\text{RGB}}(g) = \{C_j^g\}_{\text{RGB}}$ or $P_{\text{depth}}(g) = \{C_j^g\}_{\text{depth}}$, where C_j^g is the chromosome, $j=1, 2, \dots, M$, M is the population size, $g=1, 2, \dots, G$, and G denotes the number of generations. Accordingly, the initialization of population is $P_{\text{RGB}}(1) = \{C_j^1\}_{\text{RGB}}$ or $P_{\text{depth}}(1) = \{C_j^1\}_{\text{depth}}$. In order to cover the entire search space, individual of the initial population should be generated randomly according to a uniform distribution.

2.2.2. Fitness Evaluation. GA is dependent on the fitness function which can find out the quality of candidate solution such as measurable and returning the fitness value. They perform very well for large scale optimization problem. When inputting X_{RGB} or X_{depth} , there will be a reconstructed image $R_{X_{\text{RGB}}}(C_j)$, $j = 1, 2, \dots, M$ or $R_{X_{\text{depth}}}(C_j)$, $j = 1, 2, \dots, M$

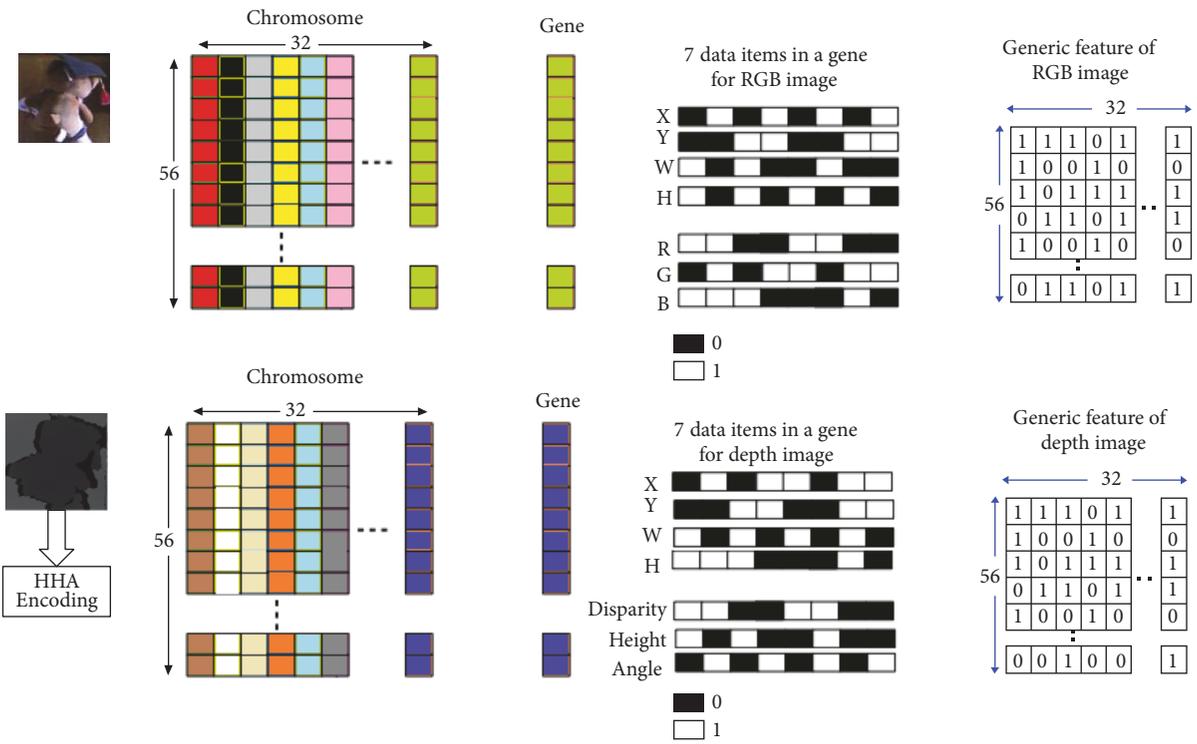


FIGURE 1: Individual representations in RGB and depth modality in GA.

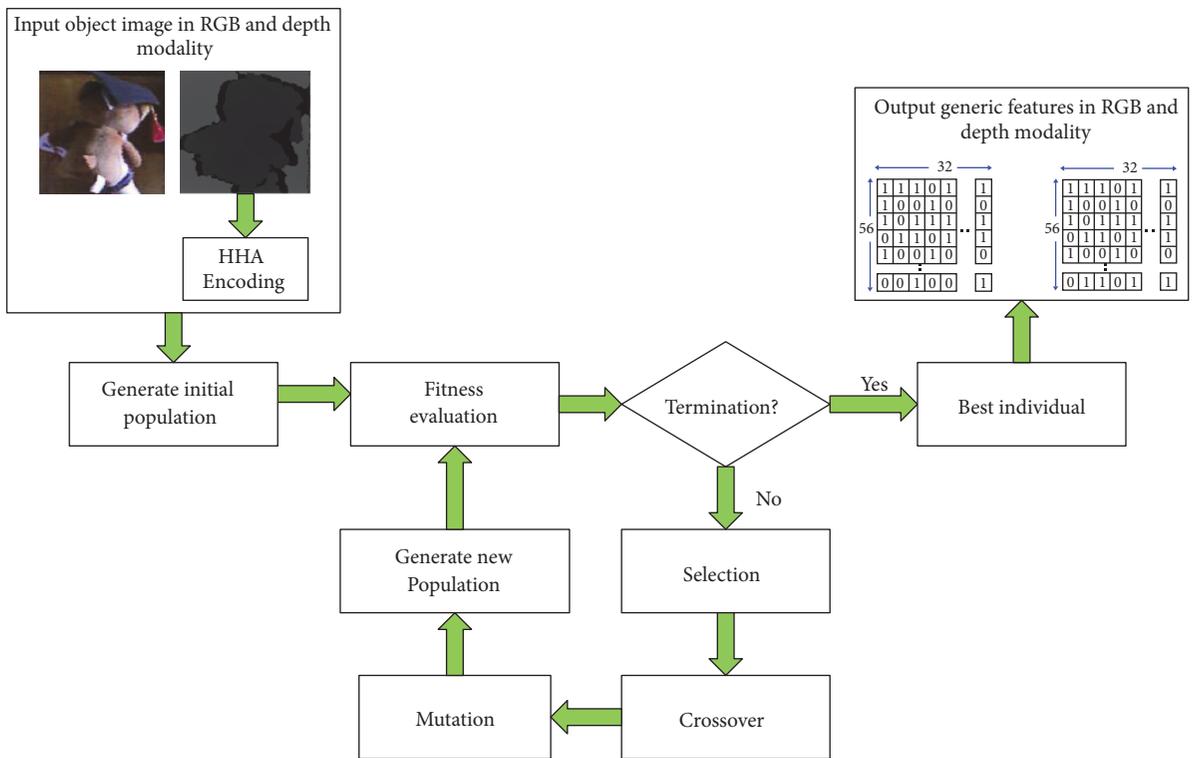


FIGURE 2: Flowchart shows how to calculate genetic feature in RGB and depth modality.

using the j th individual/chromosome/candidate solution C_j in the GA domain. The objective of genetic methods is to maximize the performance of candidate solutions evaluated by appropriate fitness function. For each candidate solution C_j , the corresponding fitness function is defined as follows:

$$\begin{aligned} f(R_{X_{\text{RGB}}}(C_j)) &= \|X_{\text{RGB}} - R_{X_{\text{RGB}}}(C_j)\|_2 \\ \text{or } f(R_{X_{\text{depth}}}(C_j)) &= \|X_{\text{depth}} - R_{X_{\text{depth}}}(C_j)\|_2 \end{aligned} \quad (1)$$

The best individual/chromosome/candidate solution p_i^{best} is calculated by

$$\begin{aligned} C_{j,\text{RGB}}^{\text{best}} &= \underset{C_j}{\operatorname{argmin}} f(R_{X_{\text{RGB}}}(C_j)) \\ \text{or } C_{j,\text{depth}}^{\text{best}} &= \underset{C_j}{\operatorname{argmin}} f(R_{X_{\text{depth}}}(C_j)) \end{aligned} \quad (2)$$

where $C_{j,\text{RGB}}^{\text{best}}$ or $C_{j,\text{depth}}^{\text{best}}$ is the genetic feature of X_{RGB} or X_{depth} ; for convenience, they are expressed as F^{RGB} and F^{depth} in the tracking algorithm.

2.2.3. Selection, Mutation, and Crossover Operation. Selection operation is also called reproduction operation, which is used to select the better individuals to be parents to the next generation. In selection method, we randomly choose the chromosome from the population according to fitness evaluation function (as (1) and (2)). Based on the fitness ranking, we use the ranking selection scheme to select better individuals from the current generation to be parents (p_1, p_2) of the next generation. In this process, high quality individuals have higher rankings and have higher chance to be selected.

In ranking selection [24], individuals are first sorted according to their fitness value and then the ranks are assigned to them. The worst individual has rank 1 and next has rank 2 and the best rank has M , where M is the size of population. The selection probability is then assigned linearly to the individuals according to their ranks.

$$P_j = \frac{1}{M} \left(m_{\text{worst}} + (m_{\text{best}} - m_{\text{worst}}) \frac{j-1}{M-1} \right), \quad (3)$$

$$j = 1, \dots, M$$

where P_j is the probability of selecting the j th individual, m_{worst}/M is probability of selecting the worst individual, and m_{best}/M is probability of selecting the best individual.

Crossover operation [25] is also called recombination, which is used to combine the genetic information of two parents (p_1, p_2) to produce new offspring (c_1, c_2) by selecting the crossover points with a crossover probability P_C . In our study, the swap node operator is used to generate new offspring, in which two nodes from a parent are randomly selected and swapped to create a second offspring.

Mutation will change in the DNA sequence of a cell's genome. Mutation operation is used to maintain genetic diversity from one generation of a population of chromosomes to the next generation. In our method, gene value is

altered in mutation operation by selecting and flipping a bit randomly.

2.3. Proposed RGB-D Tracker. The pipeline of the proposed RGB-D tracking approach is shown as Figure 3. The center location (x_{t-1}, y_{t-1}) of target object in frame $t-1$ is read as the center of the frame t , and a square with the side length of 32 pixels is drawn and taken as the search space. We sample N candidates within the search space in RGB and depth modality and calculate the genetic features of candidates $F_{t,i}^{\text{RGB}}$ and $F_{t,i}^{\text{depth}}$, $i=1, 2, \dots, N$. To fuse the information in RGB and depth modality, the errors between genetic feature of target object in frame $t-1$ and the candidates in current frame are computed, the sum of errors in RGB and depth modality are

$$\begin{aligned} E &= E^{\text{RGB}} + E^{\text{depth}} \\ &= \|F_{t,i}^{\text{RGB}} - F_{t-1}^{\text{RGB}}\|_2 + \|F_{t,i}^{\text{depth}} - F_{t-1}^{\text{depth}}\|_2, \end{aligned} \quad (4)$$

and the candidate with the minimum sum of errors in two modalities is output as the tracking result.

3. Experiments and Results

The performance of our proposed RGB-D tracker is evaluated using MATLAB R2016b platform on a server with a 12-core processor, 512GB Memory, and NVIDIA GeForce GT650m GPU running on Windows 10 operating system. Very few software resources for RGB-D object tracking are available publicly. We only use HOG detectors implemented in Opencv to run on the GPU. In our study, the population size $M=30$, the number of generations $G=200$, and the number of samples $N=100$. These values are determined after a lot of experiments for considering the balance of accuracy and speed of tracker. We conduct the experiments on the popular RGB-D tracking benchmark dataset: PTB dataset. The dataset includes 100 RGB-D videos.

3.1. Comparison of Success Rate. We compare our algorithm with the following state-of-the-art RGB-D trackers: Prin Tracker [10], DC-KCF Tracker [11], OAPF Tracker [12], and Berming Tracker [14]. The evaluation results are provided according to success rate (SR) as measurement, which is computed in terms of the overlap:

$$\text{overlap} = \frac{\text{area}(R_T \cap R_G)}{\text{area}(R_T \cup R_G)} \quad (5)$$

where R_G is the region of ground truth, R_T represents the region of the tracking result, and the larger scores mean more accurate tracking results.

The comparison results of SR for 11 attributes on the PTB dataset are shown in Table 1 for analyzing the performance of our RGB-D tracker. The PTB has been divided into different categorizations in terms of the characteristics of target objects, such as object type, size, speed of movement, presence of occlusions, and motion type. According to the results, we can see that our tracking algorithm based on

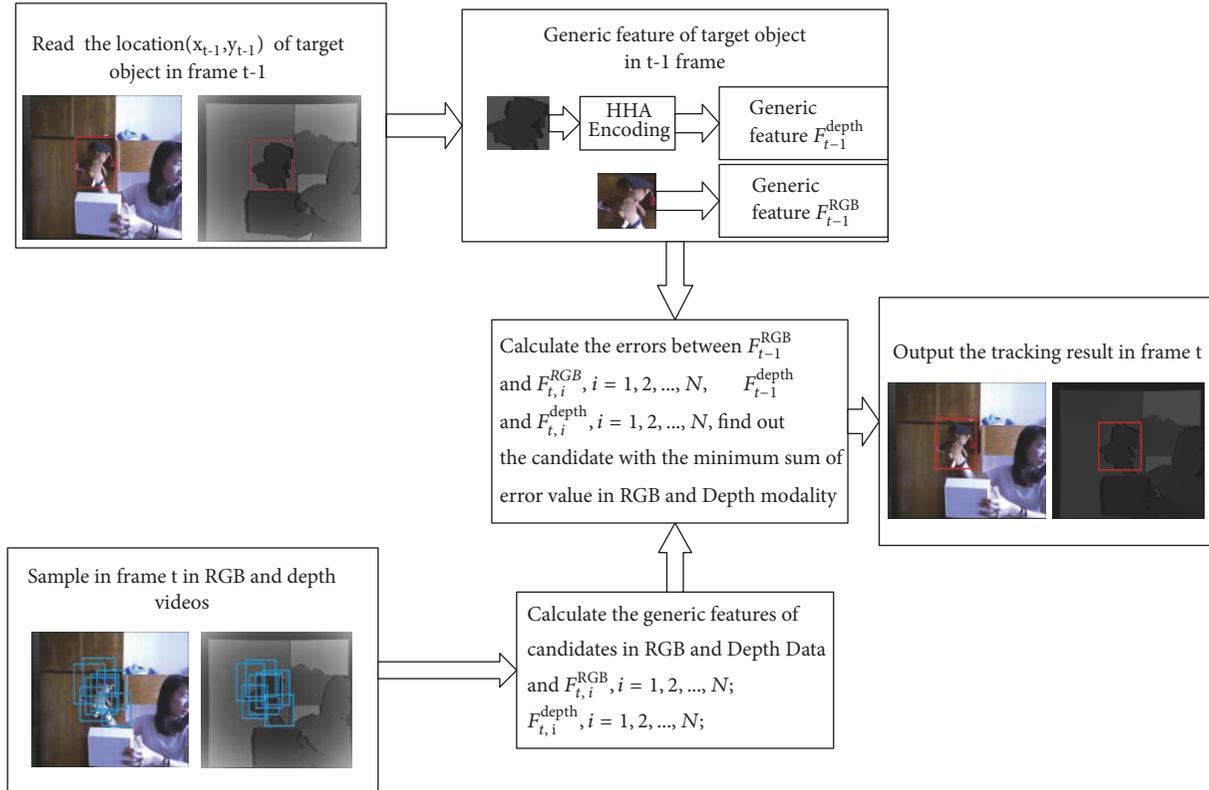


FIGURE 3: The pipeline of the proposed tracking approach.

TABLE 1: SR results on the PTB dataset.

Algorithm	Target type		Target size		Movement		Occlusion		Motion type		
	human	animal	rigid	large	small	slow	fast	yes	no	passive	active
Print Tracker	0.74	0.63	0.78	0.78	0.70	0.76	0.72	0.72	0.75	0.82	0.70
DS-KCF Tracker	0.67	0.61	0.76	0.69	0.70	0.75	0.67	0.63	0.78	0.79	0.66
OAPF Tracker	0.64	0.85	0.77	0.73	0.73	0.85	0.68	0.64	0.85	0.78	0.71
GBMTracker	0.80	0.72	0.77	0.80	0.71	0.82	0.77	0.81	0.82	0.79	0.69
Our Tracker	0.82	0.75	0.78	0.81	0.74	0.82	0.73	0.81	0.84	0.79	0.68

TABLE 2: The average running speed comparison results.

RGB-D Tracker	FPS
Print Tracker	0.25
DC-KCF Tracker	21.45
OAPF Tracker	1.60
Berming Tracker	16.5
Ours	20.65

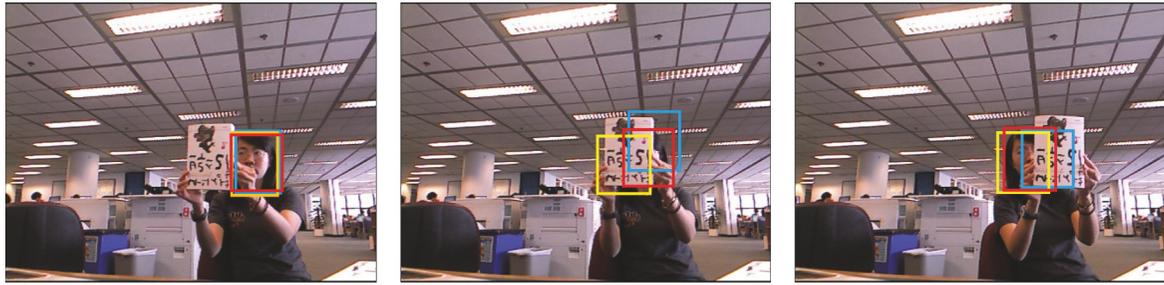
genetic feature learning obtains the highest SR in *human*, *rigid*, *large*, *small*, and *occlusion* and obtains the second best results in *animal*, *fast*, *no_occlusion*, *slow*, and *passive*, and obtains the fourth when motion type is *active*.

3.2. Comparison of Running Speed. The comparison results of average running speed on the PTB dataset which included 100

testing RGB-D videos are illustrated in Table 2. We speed up the computation by using NVIDIA GeForce GT650m GPU, and 512 GB memory is required to conduct these experiments as the GPU memory consumption. The speed of our method is only slower than DC-KCF Tracker, but the results of overall SR are better than it.

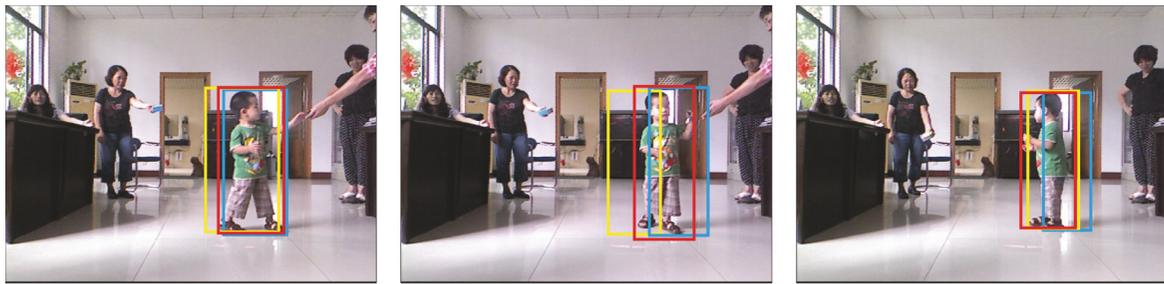
3.3. Comparison of Different Modality. The comparisons of tracking results using features in different modalities are given in the following Figure 4 to show the contribution of fusing genetic features in RGB and depth modality. Due to limited space, experimental results of four testing videos in the PTB dataset are selected in this section. The detailed descriptions for the selected videos are presented in Table 3.

Depth data is complementary to color information. 3D information included in depth data can reduce the effect of occlusions; at the same time, illumination variations have low impact to depth information. The above experimental results



RGB Depth RGB+Depth

(a) Book.move



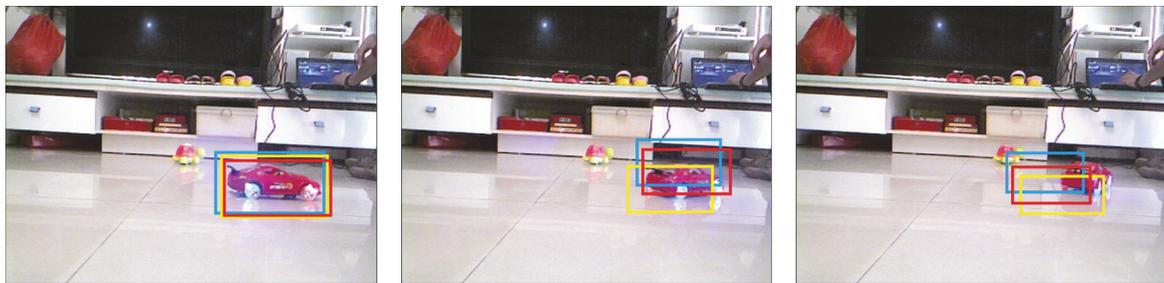
RGB Depth RGB+Depth

(b) Child



RGB Depth RGB+Depth

(c) Ball



RGB Depth RGB+Depth

(d) Toy-car

FIGURE 4: The tracking results of the selected test videos.

TABLE 3: Descriptions for the selected videos.

Test video	Challenge
Book_move	Full occlusion
Child	Appearance change
Ball	Illumination change
Toy-car	Move fast and turning

also show that fusing information in RGB and depth modality can improve the accuracy of tracking. Only using genetic feature in RGB videos, the tracking box drifts when appearance changes or occlusion occurs. At the same time, from Table 1 and Figure 4(d), we can find that the performance of our tracker is not good enough when the object moves fast or actively; we plan to do further researches about search strategy to improve it in the future.

4. Conclusion

In this paper, we have developed a RGB-D tracking method based on genetic feature learning, which can fuse the color and depth information for visual object tracking tasks. Our method treats feature learning as an optimization problem to be solved by GA. Experimental results manifest that our RGB-D tracker achieves higher accuracy and faster tracking speed. Compared with existing RGB-D tracking methods, the proposed genetic feature learning for tracking requires neither prior acknowledgement of object, data preprocessing, or learning a large set of parameters to obtain better performance. In future work, we will mainly focus on improving search strategy to improve our method.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by Department of Quanzhou Science and Technology (no. 2016N057), Major Program of Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant I8KJA520002, Project Funded by the Jiangsu Laboratory of Lake Environment Remote Sensing Technologies under Grant JSLERS-2018-005, Six Talent Peaks Project in Jiangsu Province under Grant 2016XYDXXJS-012, the Natural Science Foundation of Jiangsu Province under Grant BK20171267, the Fifth Issue 333 High-Level Talent Training Project of Jiangsu Province (BRA2018333), and 533 Talents Engineering Project in Huanan under Grant HAA201738.

References

- [1] X. Wang, "Intelligent multi-camera video surveillance: a review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.
- [2] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1704–1716, 2013.
- [3] C. Marina Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, 2018.
- [4] M. Manafifard, H. Ebadi, and H. Abrishami Moghaddam, "A survey on player tracking in soccer videos," *Computer Vision and Image Understanding*, vol. 159, pp. 19–46, 2017.
- [5] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.
- [6] D. Du, H. Qi, L. Wen, Q. Tian, Q. Huang, and S. Lyu, "Geometric hypergraph learning for visual tracking," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4182–4195, 2017.
- [7] W.-L. Zheng, S.-C. Shen, and B.-L. Lu, "Online depth image-based object tracking with sparse representation and object detection," *Neural Processing Letters*, vol. 45, pp. 1–14, 2016.
- [8] A. Ning, G. X. Zhao, and Z. G. Hou, "Online RGB-D tracking via detection-learning-segmentation," in *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, Cancun, Mexico, 2016.
- [9] T. Bagautdinov, F. Fleuret, and P. Fua, "Probability occupancy maps for occluded depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2829–2837, IEEE, Boston, MA, USA, June 2015.
- [10] S. Song and J. X. Xiao, "Tracking revisited using rgbd camera: unified benchmark and baselines," in *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Sydney, Australia, December 2013.
- [11] S. Hannuna, M. Camplani, J. Hall et al., "DS-KCF: a real-time tracker for RGB-D data," *Journal of Real-Time Image Processing*, pp. 1–20, 2016.
- [12] K. Meshgi, S.-I. Maeda, S. Oba, H. Skibbe, Y.-Z. Li, and S. Ishii, "An occlusion-aware particle filter tracker to handle complex and persistent occlusions," *Computer Vision and Image Understanding*, vol. 150, pp. 81–94, 2016.
- [13] S. Awwad and M. Piccardi, "Prototype-based budget maintenance for tracking in depth videos," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21117–21132, 2017.
- [14] J. Xiao, R. Stolkin, Y. Gao, and A. Leonardis, "Robust fusion of color and depth data for RGB-D target tracking using adaptive range-invariant depth models and spatio-temporal consistency constraints," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2485–2499, 2018.
- [15] M. Jiang, Z. Pan, and Z. Tang, "Visual object tracking based on cross-modality Gaussian-Bernoulli deep Boltzmann machines with RGB-D sensors," *Sensors*, vol. 17, no. 1, pp. 121–135, 2017.
- [16] S. Gu, Y. Lu, L. Zhang, and J. Zhang, "RGB-D tracking based on kernelized correlation filter with deep features," in *Proceedings of the International Conference on Neural Information Processing*, Lecture Notes in Computer Science, pp. 105–113, Springer International Publishing, Cham, Switzerland, 2017.
- [17] J. Sanchez-Riera, K. Srinivasan, K.-L. Hua, W.-H. Cheng, M. A. Hossain, and M. F. Alhamid, "Robust RGB-D hand tracking

- using deep learning priors,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2289–2301, 2018.
- [18] M. Jiang, C. Deng, M. Zhang, J. Shan, and H. Zhang, “Multimodal deep feature fusion (MMDFF) for RGB-D tracking,” *Complexity*, vol. 2018, Article ID 5676095, 8 pages, 2018.
- [19] J. Cardoso and L. Corte-Real, “Toward a generic evaluation of image segmentation,” *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1773–1782, 2005.
- [20] Y. Lin and B. Bhanu, “Evolutionary feature synthesis for object recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 156–171, 2005.
- [21] G. Schneider, H. Wersing, B. Sendhoff, and E. Körner, “Evolutionary optimization of a hierarchical object recognition model,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 3, pp. 426–437, 2005.
- [22] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” *European Conference on Computer Vision*, vol. 8695, no. 7, pp. 345–360, 2014.
- [23] S. A. A. Shah, M. Bennamoun, F. Boussaid, and L. While, “Evolutionary feature learning for 3-D object recognition,” *IEEE Access*, vol. 6, pp. 2434–2444, 2018.
- [24] P. Sharma, A. Wadhwa, and K. Komal, “Analysis of selection schemes for solving an optimization problem in genetic algorithm,” *International Journal of Computer Applications*, vol. 93, no. 11, pp. 1–3, 2014.
- [25] H. Nazif and L. S. Lee, “Optimised crossover genetic algorithm for capacitated vehicle routing problem,” *Applied Mathematical Modelling*, vol. 36, no. 5, pp. 2110–2117, 2012.

