Journal of Engineering and Applied Sciences 13 (Special Issue 4): 3817-3825, 2018
ISSN: 1816-949X

A Novel Object Tracking Method Using Binary Bat Algorithm

Hathiram Nenavath and Ravi Kumar Jatoth
Department of Electronics and Communication Engineering,
National Institute of Technology, 506004 Warangal, India

Abstract: The new metaheuristic algorithm used for optimization is Binary Bat Algorithm (BBA) which is motivated by echolocation features of bats whose pulse rates vary in loudness and emission. BBA has been recognized as one of the powerful algorithms to obtain a solution for a diversity of global optimization problems. In this research to solve the problem of tracking a novel binary bat algorithm based tracking architecture is proposed. The proposed BBA based tracking architecture is used to measure object tracking error, absolute error, tracking detection rate, root mean square error and time cost experimentally as parameters for BBA. The tracking results depicts that the BBA-based tracker can robustly track a random target in different challenging situations. To reveal the capability of tracker proposed in this research, a comparison of BBA based tracker and other trackers, viz., Alpha-Beta (α-β) filter, Linear Kalman Filter (LKF), Extended Kalman Filter (EKF) and Bat Algorithm (BA) is presented.

Key words: Binary bat algorithm, echolocation, object tracking error, optimization, pulse, target

INTRODUCTION

One of the principal operation in computer vision is object tracking and it has attracted the significant attention of computer vision research community, since, last few years. Even though, visual tracking has been studied for more than a few decades and significant evolution has been made in recent years (Sokhandan and Monadjemi, 2016; Comaniciu *et al*., 2003; Hare *et al*., 2016; Wei Chen *et al*., 2016; Yi *et al*., 2016; Gao *et al*., 2013) it remains a challenging problem. The reason is that object tracking has many real world applications, including visual surveillance, video communication and compression, navigation, display technology, high-level video analysis, traffic control, metrology, video editing, augmented reality and human-computer interfaces to medical imaging (Cannons, 2008; Yilmaz *et al*., 2006) and so on. However, as real world scenarios involves many complex factors such as illumination variation, shape deformation, partial occlusion, camera motion, etc. object tracking becomes challenging process due to difficulties in handling these factors.

Object tracking in consecutive image frames can also be considered as one of the searching methods for the maximum analogous candidate area of the object by an effective depiction of the object (Gao *et al*., 2015). Thus, an effective search approach and a robust appearance model are essential for a tracker. In the recent era, excessive struggles have been expended on appearance model building. It usually comprises of two mechanisms, namely statistical modeling (Wu *et al*., 2015) and visual illustration. Visual illustration emphases on the building of robust object label using various forms of visual structures and it is usually distributed into local visual illustration (Zhou *et al*., 2009) and global visual illustration (Comaniciu *et al*., 2003). Statistical modeling focuses on exactly how active mathematical prototypes are constructed for object recognition with statistical learning method and these are categorized into three types which include generative-method (Zhang *et al*., 2014) discriminative-method (Bai and Tang, 2012) and a hybrid of both methods (Yang *et al*., 2009).

On the other hand, compared with the appearance model, less attention is paid to search approaches. In tracking, the object can be traced by probing and evaluating every probable candidate in the image. If the state space for visual tracking is enormous then for exploration it takes a lot of time. Hence, an effective object search approach for estimation of the state is necessary to decrease the possibility of the object's state. Generally, two different search approaches employed that are probabilistic approach and deterministic approach (Gao *et al*., 2012). The probabilistic approach uses the uncertainty modeling, Bayesian framework and inseminate the provisional concentrations to solve state problem in

Corresponding Author: Hathiram Nenavath, Department of Electronics and Communication Engineering, National Institute of Technology, 506004 Warangal, India

the tracking. For such approach Kalman filter is the one of the example. Deterministic approach locates the target in each of the frames by iteratively probing for a region that exploits the analogy between the target window and this region (Kalman, 1960). One of the representative deterministic approaches is mean shift (Comaniciu *et al*., 2003).

Generally, object tracking also considered as an optimization problem (Gao *et al*., 2013). The similarity function is the observation distance between the object and candidate and maximization of the similarity function in the candidate solution can be referred to as tracing the object in image frames. Apparently, more intelligent searching techniques are used to resolve visual tracking problems based on optimization algorithms by many researchers. For example, Zhang *et al*. (2008) proposed Particle Swarm Optimization (PSO) based tracking algorithm. By Zhang *et al*. (2008) the governing parameters for the progress of the particles present in the swarm were reorganized depending on the fitness values of the search agents and the proposed tracker was found to be more vigorous and operational than unscented particle filter and particle filter. Fourie *et al*. (2010) used Improved Harmony Search (IHS) algorithm based visual tracking system. By Fourie *et al*. (2010), the target location is attained using improved harmony search algorithm and it is showed that improved harmony search was capable of tracking feebly modeled objects in real time. A tracker based on Cuckoo Search (CS) was proposed by Gao *et al*. (2013) recently. In their research, a comparative study of optimization and visual tracking was done which shows that the performance of the proposed tracker was better in several challenging tracing instances.

It is well-known that with strong abilities to search, the bat is one of the extremely powerful animals in nature. Motivated by echolocation features of bats whose frequency varies in loudness and emission, Yang proposed the Bat Algorithm (BA) (Yang, 2010). Preliminary studies showed that the performance of BA was much better than PSO and Genetic Algorithms (GA) for several standard benchmark functions (Yang, 2010). Since, the development of BA, it has been useful in an extensively wide range of optimization applications which includes scheduling, electricity market, artificial-neural-network, etc.

In this research, object tracking is observed as a method of probing for a target object using several bats in consecutive images. The proposed BBA based tracking architecture is used to experimentally measure object tracking error, absolute error, tracking detection rate, root mean square error and time cost as parameters for BBA. To reveal the capability of tracker proposed in this research, a comparison of BBA based tracker and other trackers, viz., Alpha-Beta (α-β) filter (Crouse, 2015), linear kalman filter (Simon, 2010), Extended Kalman filter (EKF) (Simon, 2010) and Bat Algorithm (BA) (Gao *et al*., 2016) is presented.

MATERIALS AND METHODS

Bat algorithm: Yang proposed a metaheuristic algorithm for optimization named bat algorithm, based on microbat's echolocation. According to standard BA, the echolocation features of bats are admired as following 3 rules.

The distance is sensed by all bats, using a unique feature called echolocation. The difference between food/prey and background obstacles are identified by them in some delightful way. Bats start with a random flight with velocity v_i at position x_i with a constant frequency and loudness for searching food/prey. Bats can spontaneously regulate the frequency (or wavelength) of pulses emitted by them and modify pulse emission rate based on the closeness of the object. Though, the loudness may differ in several ways, it is presumed that the loudness changes from a high (positive) A_0 to a least fixed value A_{min}.

For every bat (say i), its position x_i^{t-1} and velocity v_i^{t-1} are defined in a d-dimensional search space and they are reorganized consecutively during iterations. The global search method is used to design current solutions x_i^t and velocities v_i^t at time step t as follow:

$$f_i = f_{min}+(f_{max}-f_{min})\,\beta \tag{1}$$

$$v_i^t = v_i^{t-1}+(x_i^{t-1}-x_\phi)\,f_i \tag{2}$$

$$x_i^t = x_i^{t-1}+v_i^t \tag{3}$$

Where:

β = A random value which is distributed uniformly between 0 and 1

x = The new global best solution

The frequency f_{min} and f_{min} are depends on dimension of the problem under consideration. For doing local search, after selection of best solution among the current solutions, a random walk model is used to generate a modified solution for every bat locally which is given by:

$$x_{new} = x_{old}+\varepsilon\, A^t \tag{4}$$

Where:

$\varepsilon \in (-1, 1)$ = A random number

x_{old} = The solution in the current optimization solution set

$A^t \le A_i^t$ = The average loudness of all the bats at time step t

The pulse emission r_i and the loudness A are reorganized to replicate that if the object is found, the bats increase the pulse emission and reduces the loudness by:

$$r_i^{t+1} = r_i^0 \,[1\text{-exp}\,(-\gamma\, t)],\; A_i^{t+1} = \omega\, A_i^t \tag{5}$$

where, r_i^t the primary pulse rate, increasing pulse frequency coefficient is denoted by ω the attenuation coefficient of pulse amplitude is denoted by γ. For any $0<\omega$, $\gamma<1$, we have:

$$A_i^t \to 0,\; r_i^t \to r_i^0 \text{ as } t \to \infty \tag{6}$$

Binary Bat Algorithm (BBA): A contour of a binary search space may be assumed as a hypercube. Binary optimization algorithm has search agents whose position can be simply shifted to farther and nearer vertices of the hypercube by interchanging different bits (Kennedy and Eberhart, 1997). Therefore, while constructing the binary form of BA, i.e., BBA (Mirjalili *et al.*, 2014) a few basic ideas of updating velocity and position need to be improved.

In the continuous form of BA, the artificial bats travel in the search space using velocity and position vectors in the continuous real domain. Subsequently by adding velocities to positions as shown in Eq. 3, the effortless implementation of the concept of position updating can be done. But, the significant updating process of position is not the same in a discrete binary space. In binary space as we deal with only two binary values, "0" and "1", the updating procedure of position cannot be performed by using Eq. 3. Hence, a better method needs to be found to utilize velocities to update agent's positions from "1" to "0" or "0" to "1". That is a link needs to be developed amid position and velocity besides reorganizing the updating procedure of position in Eq. 3.

Position updating in discrete binary search space is nothing but swapping between numbers "1" and "0". This swapping must be done on the basis of agent's velocity. In real space, the concept of velocity must be employed in order to revise the positions of agents in binary space. As stated by Rashedi *et al.* (2010), the concept is to vary an agent's position with the probability of its velocity. So, as to do this, it is essential to develop a Transfer Function (TF) for mapping values of velocity to those of probability for position updating. That is, a TF can be defined as the probability of shifting the elements of position vector's from "1" and "0" or "0" and "1". Thus, TF pushes particles to transfer into a binary space. According to Rashedi *et al.* (2010) for designing a TF, a superior way is to force the particles with greater velocity to shift their positions. A V-shaped TF and rule for updating position are proposed in order to perform this as shown in Eq. 7 and 8:

$$T\left(v_i^k\,(t)\right) = \left|\frac{2}{\pi}\arctan\left(\frac{\pi}{2}\, v_i^k\,(t)\right)\right| \tag{7}$$

$$x_i^k\,(t+1) = \begin{cases}\left(x_i^k\,(t)\right)^{-1} \text{ if r and} < T\,(v_i^k\,(t+1)) \\ x_i^k\,(t) \text{ r and} \ge T\,(v_i^k\,(t+1))\end{cases} \tag{8}$$

Where:

$x_i^k(t)$ and $v_i^k(t)$ = Indicates the position and velocity of particle

i = Iteration t in k-th dimension

$(x_i^k(t))^{-1}$ = The complement of $x_i^k(t)$

The advantage of this method is that these TFs do not force particles to take values of 1 or 0. That is, they stimulate particles to stay in their existing positions when their velocity is low or switch to its complement when the velocity is high. These are named v-shaped TFs and their group is also named the "v-shaped" family of TFs.

Here, Eq. 7 represents the TF for mapping the velocities of BBA's bats to the probabilities of switching the elements of their position vectors. Consequently to update position vectors, Eq. 8 is used. Algorithm 1 shows pseudo code of BBA.

Algorithm 1; Pseudocode of BBA:

```
Initialize the bat population: xi (i = 1, 2, ..., n) = rand (0 or 1) and vi = 0
Define pulse frequency fi
Initialize pulse rates ri and the loudness Ai
while (t<Max number of iterations)
        Adjusting frequency and updating velocities
        Calculate transfer function value using equation (7)
        Update positions using equation (8)
        if (rand>ri)
                Select a solution (G best) among the best solutions randomly
                Change some of the dimensions of position vector with some
        of the dimentions of G best
        end if
        Generate a new solution by flying randomly
        if (rand<Ai and f (xi)<f (Gbest)
                Accept the new solutions
                Increase rate of pulse emission ri and reduse loudness Ai
        end if
        Rank the bats and find the current Gbest
end while
```

BBA based tracking system: Tracking the object in videos or the issue of tracking of target in each frame can be inferred as problem of optimization. The similarity function (fitness function) is nothing but the observation distance between the candidate and the target. Tracing the target can be inferred as maximizing or minimizing the similarity function in the candidate solution. In this manner, tracking of object is an optimization problem which could be solved with the help of optimization methods.

Consider, the image being searched contains the target. A random group of target candidates is generated in BBA based tracker. The BBA based tracker plays a role in finding the best candidate solution. The tracking framework based on BBA algorithm is shown in the Fig. 1.

In the first frame, the target can be detected by some object detectors or user can also select it manually. After the target selection the state vector is initialized. In this reserach, X = (x, y, z) is the state vector in which (x, y) is the target pixel position in coordinates and s represents the scale parameter which restrains the object size. After the selection of target in the first frame, $X_0 = (x_0, y_0, 1)$ is initialize as state vector. Where x_0, y_0 is the target's initial location and s = 1.0 specifies that there is no scale variation in the first frame.

After the selection of the target and initialization of state vector, a dynamic model produces new candidate's state vectors. Here, the random walk model is applied considering that there is very little movement of the object among frames. The correlation between the appearance and the state of the object is interpreted using an observation model. In any tracker there is need of good observation model but it is difficult to choose a particular observation model for all tracking scenarios. In this study to improve the search performance of tracker the observation model we have chosen is standard kernel based spatial color histogram (Comaniciu *et al*., 2003).

Let $\{ci\}_{i=1,...,n}$ represents the target candidate's normalized pixel locations, centered at c in the present frame. r = $\{h_x, h_y)$ where h_x and h_y, respectively denote the width and the height of the target's rectangle. The kernel-based spatial color histogram can be mathematically represented as:

$$P_c^{(u)}\left(X_k\right) = C\sum_{i=1}^{M} k\left(\left\|\frac{c-c_i}{r}\right\|^2\right) d\left[b\left(c_i\right)\text{-}u\right],\ u = 1, ...\ m \quad (9)$$

where, δ is the Kronecker delta function. The function b: $R^2 \rightarrow \{1 ... m\}$ relates to the pixel at the location c_i and the index b (c_i) of its bin in the quantized feature space. K (x) is an isotropic kernel assigning a smaller weight to pixels farther from the center is the normalization constant:

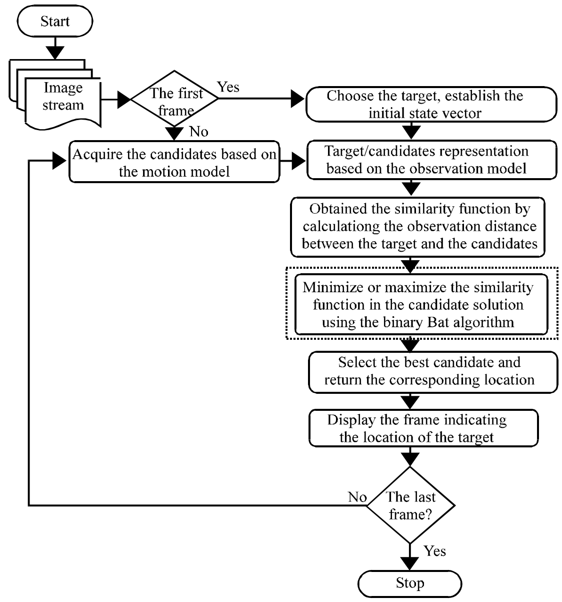

Fig. 1: Proposed BBA based tracking framework

$$C = 1/\sum_{i=1}^{M} k\left(\left\|\frac{c\text{-}c_i}{r}\right\|^2\right)$$

After the initialization of state vector to calculate the observation distance between the candidate and target a similarity (fitness) function is employed. To calculate the similarity between two histograms (Comaniciu *et al*., 2003; Gao *et al*., 2015). Bhattacharyya coefficient is applied which is given by:

$$B\left(h_1, h_2\right) = 1\text{-}\sum_{i=1}^{N} \sqrt{h_1\,(i)\,h_2\,(i)} \quad (10)$$

Where:

N = The number histogram bins

h_1 and h_2 = Shows the histograms among which comparison is done

The coefficient B (h_1, h_2) is large when the histograms are alike and small when they are different. The dashed box in Fig. 1 represents the optimization process. This is the core part of the optimization-based visual tracking algorithm. In this part an optimizer is adapted to select the candidate solution. This way the tracking can be carried out by maximizing or minimizing the similarity function. Each time the optimizer is queried for the target position, the frame is displayed to indicate the position of the target. The entire loop extends until no other frame is available.

RESULTS AND DISCUSSION

Experimental setup: In this research, we implemented our tracker in Matlab R 2013 a PC machine with Intel i7-3770 CPU (3.4 GHz) with 2GB memory which runs 29 fps in this platform. In addition, the self-made video consisting of JPEG image sequence with 720×1280 pixels per frame is taken using iPhone 6 camera. The environment for bad light condition is created by switching off all the lights in room (improper illumination) while the environment is considered as normal light condition when the room is properly illuminated. The distance considered for tracking the target object is approximately 6 m with target objects being balls of different sizes, i.e., small (B1), medium (B2) and big (B3) whose radii are 3.325, 4.9 and 8.5 cms, respectively. Table 1 shows the initial values of the basic parameters for the tracking algorithms.

Performance measures

Absolute Error (AE): AE is the magnitude of difference between the true value and the tracked value of the object:

$$AE = |X_{ture} - X_{tracked}| \tag{11}$$

Where:

x_{true} = True value of object parameters

$x_{tracked}$ = The tracked value of the object parameters

Root Mean Square Error (RMSE): RMSE is one of the most widely used full-reference quality assessment metric which is computed by square root of the average of squared intensity differences between tracked ($x_{tracked}$) and true image pixels (x_{true}):

$$RMSE = \sqrt{\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}(X_{tracked} - X_{true})^2} \tag{12}$$

where, N and M are the image dimensions.

Tracking Detection Rate (TDR): Tracking detection rate is the ratio of number of frames in which the object is detected to the total number of frames in which the object present:

$$TDR = \frac{\text{Object detected in number of frames}}{\text{Object present in number of frames}} \times 100 \tag{13}$$

Object Tracking Error (OTE): Object tracking error is the normal inconsistency in the centroid of the tracked object from its true value. It is given by:

$$OTE = \frac{\sqrt{\sum_{i=1}^{N}(X_{true} - X_{tracked})^2 - (Y_{true} - Y_{tracked})^2}}{N} \tag{14}$$

Table 1: Simulation parameters for trackers

α-β		LKF		EKF		BBA	
Parameters	Values	Parameters	Values	Parameters	Values	Parameters	Values
α	0.9	dt	1	dt	1	Number of artificial bats	20
β	0.005	A	$\begin{bmatrix}1&0&dt&0\\0&1&0&dt\\0&0&1&0\\0&0&0&1\end{bmatrix}$	A	$\begin{bmatrix}1&0&dt&0\\0&1&0&dt\\0&0&1&0\\0&0&0&1\end{bmatrix}$	f_{min}	0
Δ_k	1	H	$\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}$	H	$\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}$	f_{max}	20
-	-	R	$\begin{bmatrix}0.1524&0.0143\\0.055&0.0055\end{bmatrix}$	$\begin{bmatrix}0.2845&0.0045\\0.0045&0.0045\end{bmatrix}$	A	0.25	
-	-	Q	0.01*eye (4)	Q	0.01*eye (4)	r	0.5
-	-	P	100*eye (4)	P	100*eye (4)	β	ε [0,1]
						ε	ε [-1,1]

Table 2: Average parameters comparison for different size of balls using five trackers under normal and bad light conditions

	In Normal light condition															In bad light condition				
	For small size ball (B1)					For medium size ball (B2)					For big size ball (B3)					For small size ball (B1)				
Parameters	BBA	α-β	LKF	EKF	BA	BBA	α-β	LKF	EKF	BA	BBA	α-β	LKF	EKF	BA	BBA	α-β	LKF	EKF	BA
AE	0.11	0.98	0.12	0.76	0.12	0.10	0.74	0.21	1.25	0.17	0.17	1.31	0.63	1.34	0.17	0.09	0.45	0.11	1.05	0.11
RMSE	0.01	0.44	0.03	0.17	0.02	0.01	0.28	0.04	0.28	0.04	0.02	0.45	0.21	0.30	0.14	0.01	0.09	0.02	0.17	0.02
TDR (%)	100	96.2	100	100	100	100	91.6	100	91.6	100	100	93.1	100	93.1	100	100	85.7	100	85.7	100
OTE	0.02	1.35	0.04	0.16	0.03	0.02	0.69	0.05	0.29	0.03	0.06	1.01	0.48	0.29	0.20	0.01	0.22	0.02	0.14	0.02

Where:

X_{true} and Y_{true} = The actual 2D coordinates of the object

X_{traked} and $Y_{tracked}$ = The tracked 2D coordinates of the object

The RMSE, AE and OTEs of the BBA, α-β filter, LKF, EKF and BA for different size ball's dataset are given in Table 2 under normal light condition. As observed in Table 2, under the normal light condition, the average value of the RMSE was reduced after applying the BBA to small, medium and big size balls data. On average, the minimum RMSE reduction was 0.01, 0.01 and 0.02 for small, medium and big size ball's data, respectively under normal light conditions for the BBA. The average value of AE was reduced after applying the BBA to small, medium and big size ball's data. On average, the minimum AE was 0.11, 0.10 and 0.17 for small, medium and big size ball's data, respectively under normal light conditions for the BBA. Also, the average value of the OTE was reduced after applying the BBA to small, medium and big size balls data. On average, the minimum OTE was 0.02, 0.02 and 0.06 for small, medium and big size balls data, respectively under normal light conditions for the BBA.

The RMSE, AE and OTEs of the BBA, α-β filter, LKF, EKF and BA for only small size ball's dataset are given in Table 2 under bad light condition. As observed in Table 2, under the bad light condition, the average value of the AE, RMSE and OTE was reduced after applying the BBA to small ball data. On average, the minimum AE, RMSE and OTE were 0.09, 0.01 and 0.01 for small ball data, respectively under the bad light condition for the BBA. The best average values are emphasized in Table 2.

Figure 2 show five frames out of the tracking results for data obtained using B1-B3 under the normal light condition and the bad light condition of the BBA. The proposed approach was evaluated using a self-made database with the frame size being 720×1280. Frames shown in the figure were selected from the video while tracking the object continuously during its movement. The target object is marked by the bounding box (red circle). For B1 under the normal light condition in frames 24 and 30 of Fig. 2, the target object tracked using BBA is dominated by the red bounding box as the camera is static and the size of the object is reduced with distance. Even

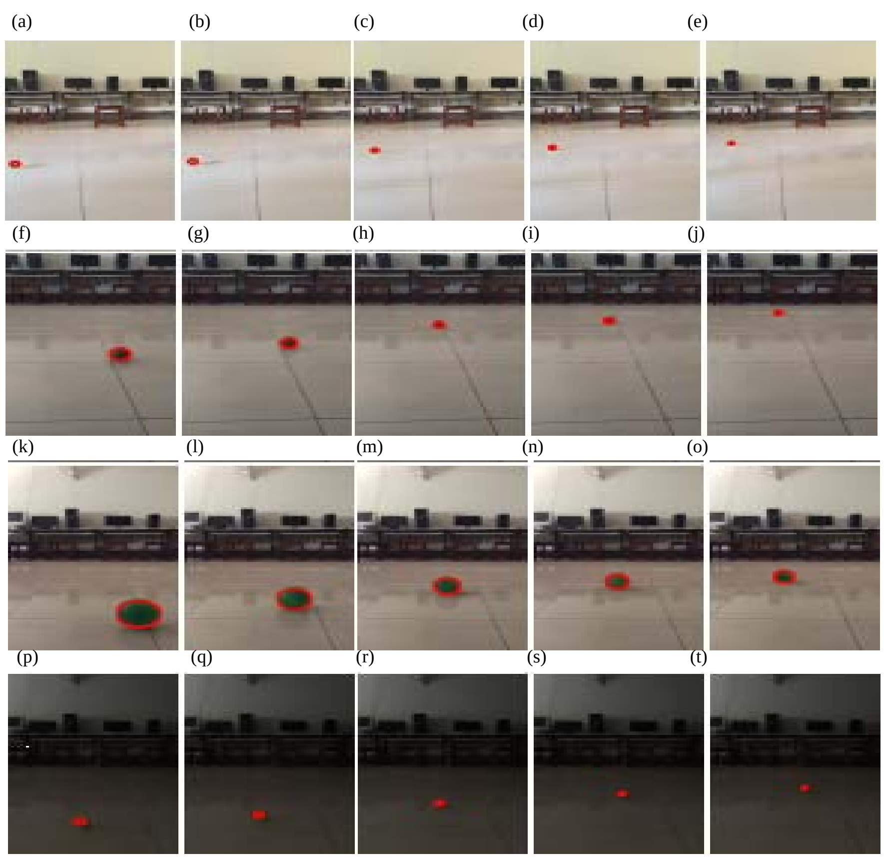

Fig. 2: Tracking result of BBA for B1 (6, 10, 21, 24 and 30 frames), B2 (11, 15, 31, 36 and 52 frames), B3 (16, 25, 39, 44 and 55 frames) in normal light condition and B1 (24, 28, 36, 42 and 51 frames) in bad light condition (From Top to bottom)

Fig. 3: Tracking result of BBA for David, Panda, Subway, Mountain Bike, Car and Crossing Video sequences (From Top to bottom)

Table 3: Average time cost of the five trackers for different size of balls under normal and bad light conditions

	Time cost (msec)				
Prameters/Image frames	BBA	α-β	LKF	EKF	BA
Normal light condition					
Small size ball (B1)	25.3	48.9	32.9	48.1	32.5
Medium size ball (B2)	23.2	27.0	29.7	29.1	29.3
Big size ball (B3)	23.7	30.0	28.7	28.4	27.8
Bad light condition					
Small size ball (B1)	19.0	25.6	24.9	25.7	23.6

though the target object is small, the proposed tracking algorithm can detect the object (B1) nevertheless. For B1 under the bad light condition in frames 42 and 51 of Fig. 2, the target object tracked using BBA is dominated by the red bounding box as the camera is static and the size of the object is reduced with distance. Even though the target object is small, the proposed tracking algorithm can detect the object (B1) under bad light condition. For analysing the time complexity, the mean time costs of the five trackers in the tracking process are evaluated and Table 3 shows the comparative results. It is evident from Table 3 that in all tracking examples (e.g., data obtained using B1-B3), the mean time cost of BBA is less than Alpha-Beta (α-β) filter, linear kalman filter, extended kalman filter and bat algorithm.

Table 4: Description of the tracking examples

Video	Frame numbers	Challenging factors
David	252	Occlusion, background clutters, out-of-plane rotation, deformation
Panda	1000	Scale variation, occlusion, deformation, in-plane rotation, out-of-plane rotation, out-of-view, low resolution
Subway	175	Occlusion, deformation, background clutters
Mountain	228	In-plane rotation, background clutters, bike out-of-plane rotation
Car	1500	Illumination variation, scale variation, background clutters
Crossing	120	Scale variation, background clutters, deformation, fast motion, out-of-plane rotation

Besides tracking the target object from data obtained using B1-B3, some videos (Wu *et al*., 2015) are selected and used for evaluation of BBA. (Note: the tracking examples are available on the website http://visual-tracking.net). As seen from Table 4 the targets in these examples are suffered from various challenging factors. Tracking result of BBA for David, Panda, Subway, Mountain Bike, Car and Crossing Video sequences are shown in Fig. 3 and experimental results depict that the BBA-based tracker is capable of tracking a random target in various challenging situations.

CONCLUSION

In this research, to solve the problem of tracking a novel binary bat algorithm based tracking architecture is proposed. Simulation results depicts that the BBA based tracker is capable of tracking an arbitrary target in different challenging environment robustly. We compare object tracking error, absolute error, tracking detection rate, root mean square error and time cost with four typical tracking algorithms including the Alpha-Beta (α-β) filter, linear kalman filter, extended kalman filter and bat algorithm. The experimental results shows that the BBA based tracker outperforms the Alpha-Beta (α-β) filter, linear kalman filter, extended kalman filter and bat algorithm.

According to the researcher's awareness, it is the commencement of using BBA for object tracking system and initial results showed that it is an influential tracking method. This research is considered for the only single object for tracking. In the near future it is expected to experiment on more proficient feature selection methods and to use BBA based tracking system for tracking multiple targets.

REFERENCES

Bai, Y. and M. Tang, 2012. Robust tracking via weakly supervised ranking SVM. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), June 16-21, 2012, IEEE, Providence, Rhode Island, USA., ISBN:978-1-4673-1226-4, pp: 1854-1861.

Cannons, K., 2008. A review of visual tracking. Technical Report CSE-2008-07, Department of Computer Science & Engineering, New York University, New York, USA. http://www.eecs.yorku.ca/research/techreports/2008/CSE-2008-07.pdf.

Chen, W., K. Zhang and Q. Liu, 2016. Robust visual tracking via patch based kernel correlation filters with adaptive multiple feature ensemble. Neurocomput., 214: 607-617.

Comaniciu, D., V. Ramesh and P. Meer, 2003. Kernel-based object tracking. IEEE. Trans. Pattern Anal. Mach. Intell., 25: 564-577.

Crouse, D.F., 2015. A general solution to optimal fixed-gain (á-â-ã etc.) filters. IEEE. Sig. Process. Lett., 22: 901-904.

Fourie, J., S. Mills and R. Green, 2010. Harmony filter: A robust visual tracking system using the improved harmony search algorithm. Image Vision Comput., 28: 1702-1716.

Gao, M.L., J. Shen, L.J. Yin, W. Liu and G.F. Zou *et al.*, 2016. A novel visual tracking method using bat algorithm. Neurocomput., 177: 612-619.

Gao, M.L., L.J. Yin, G.F. Zou, H.T. Li and W. Liu, 2015. Visual tracking method based on cuckoo search algorithm. Opt. Eng., 54: 073105-073105.

Gao, M.L., X. He, D. Luo and Y.M. Yu, 2012. Object tracking based on harmony search: Comparative study. J. Electron. Imaging, 21: 043001-043001.

Gao, M.L., X.H. He, D.S. Luo, J. Jiang and Q.Z. Teng, 2013. Object tracking using firefly algorithm. IET. Comput. Vis., 7: 227-237.

Hare, S., S. Golodetz, A. Saffari, V. Vineet and M.M. Cheng *et al.*, 2016. Struck: Structured output tracking with kernels. IEEE. Trans. Pattern Anal. Mach. Intell., 38: 2096-2109.

Kalman, R.E., 1960. A new approach to linear filtering and prediction. Trans. ASME J. Basic Eng., 82: 35-45.

Kennedy, J. and R.C. Eberhart, 1997. A discrete binary version of the particle swarm algorithm. Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics Computational Cybernetics and Simulation Vol. 5, October 12-15, 1997, IEEE, Orlando, Florida, ISBN:0-7803-4053-1, pp: 4104-4108.

Mirjalili, S., S.M. Mirjalili and X.S. Yang, 2014. Binary bat algorithm. Neural Comput. Applic., 25: 663-681.

Rashedi, E., H. Nezamabadi-Pour and S. Saryazdi, 2010. BGSA: Binary gravitational search algorithm. Nat. Comput., 9: 727-745.

Simon, D., 2010. Kalman filtering with state constraints: A survey of linear and nonlinear algorithms. IET Control Theory Appl., 4: 1303-1318.

Sokhandan, A. and A. Monadjemi, 2016. A novel biologically inspired computational framework for visual tracking task. Biol. Inspired Cognit. Archit., 18: 68-79.

Wu, Y., J. Lim and M.H. Yang, 2015. Object tracking benchmark. IEEE. Trans. Pattern Anal. Mach. Intell., 37: 1834-1848.

Yang, M., Z. Fan, J. Fan and Y. Wu, 2009. Tracking nonstationary visual appearances by data-driven adaptation. IEEE. Trans. Image Process., 18: 1633-1644.

Yang, X.S., 2010. A New Metaheuristic Bat-Inspired Algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Gonzalez, J.R., D.A. Pelta, C. Cruz, G. Terrazas and N. Krasnogor (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-12537-9, pp: 65-74.

Yi, S., N. Jiang, B. Feng, X. Wang and W. Liu, 2016. Online similarity learning for visual tracking. Inf. Sci., 364: 33-50.

Yilmaz, A., O. Javed and M. Shah, 2006. Object tracking: A survey. ACM Comput. Surv., Vol. 38. 10.1145/ 1177352.1177355.

Zhang, K., L. Zhang and M.H. Yang, 2014. Fast compressive tracking. IEEE. Trans. Pattern Anal. Mach. Intell., 36: 2002-2015.

Zhang, X., W. Hu, S. Maybank, X. Li and M. Zhu, 2008. Sequential particle swarm optimization for visual tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), June 23-28, 2008, IEEE, Anchorage, Alaska, ISBN: 978-1-4244-2242-5, pp: 1-8.

Zhou, H., Y. Yuan and C. Shi, 2009. Object tracking using SIFT features and mean shift. Comput. Vis. Image Understanding, 113: 345-352.