

MULTI-OBJECT TRACKING IN NON-STATIONARY VIDEO USING BACTERIAL FORAGING SWARMS

Hoang Nguyen, Bir Bhanu

University of California, Riverside
Center for Research in Intelligent Systems
Riverside, CA 92521 USA

ABSTRACT

One of the key problems in the field of image processing is object tracking in video. Multiple objects, occlusion, and non-stationary video are some of the challenges that one may face in developing an effective approach. A less-studied approach considers swarm intelligence. This paper presents a new and improved algorithm based on Bacterial Foraging Optimization in order to track multiple objects in real-time video exposed to full and partial occlusion, using video from both fixed and moving cameras. A comparison with various algorithms is provided.

Index Terms— swarm intelligence, multi-object tracking, non-stationary video

1. INTRODUCTION

Non-stationary video may result from active panning and tilting cameras, as well as from cameras mounted on moving platforms (such as from aerial, ground, or sea-faring vehicles). Tracking in non-stationary video is a challenging problem and approaches to addressing it include image stabilization and motion segmentation [1, 2]. This paper uses a different approach and presents an algorithm based on Bacterial Foraging Optimization (BFO) [3] which is used to track multiple objects in a single camera, treating camera motion as a part of the object motion.

2. RELATED WORK AND CONTRIBUTIONS

Traditional approaches to tracking include mean shift [4], CamShift [5], Kalman filters [6], and particle filters [7]. Mean shift and CamShift perform tracking based on distributions (such as tracking using a color histogram or texture features). Though simple to implement, they are not robust to factors such as fast movement or occlusion. Kalman filters are proven to be optimal in the sense of minimum square error, but only when the noise is Gaussian; the dynamical model used in Kalman filters is assumed to be linear and does not work well when the noise is multi-modal. Particle filters

evolve from extended Kalman filters and focus on dealing with non-linear dynamical models and multi-modal densities by sampling the prior probability. Thus, particle filters can recover from occlusion to some extent, provided there are enough particles on the image. In practice, however, the larger the number of particles, the higher the computational cost, thereby making real-time processing challenging.

In addition, other swarm intelligence algorithms have also been considered for tracking. For instance, Particle Swarm Optimization (PSO) [8] has been considered, both for single and multiple targets [9]. These approaches often involve running the algorithm to convergence for every single frame.

Unlike traditional optimization problems with stationary optima, tracking objects through video requires the algorithm to find the object not once, but for potentially every frame. With PSO, there are various approaches to dealing with moving objects. Approaches such as periodically giving a swarm amnesia have been considered [10]. Decaying the score of the best location by some percentage p_{decay} after each frame can also be used to force the swarm to continually search for a better location. Such methods are used to prevent the swarm from completely converging to a single point.

The contributions of this paper are:

- the development of a new and improved Bacterial Foraging algorithm with improved performance for multiple object tracking in video
- performance comparison with other swarm intelligence algorithms and a standard image processing/computer vision tracking algorithm such as CamShift [5]

3. TECHNICAL APPROACH

3.1. Tracking Multiple Objects in Video

Multi-object tracking is a challenging problem for a number of reasons. For instance, a tracker should be able to:

- handle partial and full occlusion
- scale to a potentially large number of objects

- operate in a real-time environment

Occlusion will likely occur as a result of object and scene interaction. In addition, the occlusion introduced by having multiple moving objects can occur arbitrarily; one cannot rely on pre-established entrance and exit points. Processing in real-time is also a highly desirable capability, as off-line analysis removes the ability of responding (such as panning or tilting to maintain the object in the camera’s field of view). There are other approaches to the problem, but Bacterial Foraging Optimization (BFO) has yet to be considered for multi-object tracking in video.

3.2. Bacterial Foraging Optimization for Tracking

Bacterial Foraging Optimization is a stochastic evolutionary search algorithm modeled after the behavior of *E.coli* bacteria [3]. A swarm consists of i bacteria particles or *agents* which “swim” and “tumble” through an environment looking for concentrations of food.

Swimming translates to each agent moving through the search space in steps of size C and a *run* consists of a series of (up to) N_s swims. After each swim, agents evaluate the fitness $J_{current}^i$ of the current position. At the end of each run, an agent tumbles, or uses its flagella to randomly rotate itself in a new direction. During a run, agents stop swimming when they notice food concentration begins to decrease ($J_{current}^i < J_{previous}^i$), otherwise swimming up to N_s times. In this way, the bacteria particles have programmed themselves to climb gradients.

Agents who cannot find food (or areas of high fitness) eventually starve to death. This is simulated in a reproduction step where the S_r agents with the worst fitness are removed. Conversely, the S_r agents with the best fitness go on to reproduce. In practice, this reproduction step essentially relocates the S_r worst agents to the location of the S_r best agents and keeps the size of the swarm constant.

Finally, all agents are subjected to an elimination-dispersal step where each agent is suddenly relocated with probability p_{ed} to a random location in the search space. This is done to simulate a dynamic environment which can displace agents (such as being introduced to a liquid). This important step ensures that the swarm does not fully converge and that the search space remains covered.

3.3. Details of Improved BFO Algorithm

While traditionally suited to searching for static or slow-moving global optima, BFO can also be adapted to tracking faster moving objects in video. We contribute three major modifications:

1. **Lookahead.** In classical Bacterial Foraging, agents stop swimming once the fitness gets worse. While it may be more realistic, it also leaves agents in a known suboptimal position (the previous position was better). This can

```

Initialize  $S$  agents randomly in the image;
for each frame do
  Evaluate fitness of previous best location;
  If fitness is good enough, stop for this frame;
  for up to  $N_r$  times do
    Initialize health of each agent to 0;
    for up to  $N_c$  times do
      for each agent without immunity do
        Evaluate fitness of current position;
        Choose a random direction;
        Move forward and evaluate fitness;
        Continue moving (up to  $N_s$  times) until
        fitness gets worse;
        Add fitness of each evaluation to
        agent’s health;
        Move 1 step back;
      Sort agents by health;
      Relocate  $S_r$  agents with worst health to
      location of  $S_r$  best agents with best health;
      Grant these agents  $S_r$  best agents immunity
      from movement;
    for each agent without immunity do
      With a probability of  $p_{ed}$ , relocate this agent to
      a random location;
  Reset immunity of all agents;

```

Algorithm 1: Improved Bacterial Foraging Algorithm for tracking.

be modified so that agents are given the ability of looking ahead and calculating the fitness of that position, and seeing whether or not they should move. This has the effect of agents preferring to stay within boundaries of higher fitness (which is useful when objects have a clear boundary or steep gradients; this is largely impacted by the step size C). Doing so allows the algorithm to move in bigger steps without worrying about overstepping or stepping out of an area of high fitness.

2. **Elitism.** In the classic BFO algorithm, all agents move at every reproduction step, including agents who had the best health on previous reproductions within the same frame. Since frames do not change across sequential reproduction steps, stopping the best agents (one from each reproduction) from further movement for the current frame has been found to improve the algorithm’s accuracy. In addition, the swarm’s best guess is selected from this group of agents (as opposed to the best agent from the last reproduction). Finally, granting these best agents with immunity to the elimination-dispersal step allows the swarm to keep its best agents on the target and having everyone else spread out.
3. **Early Termination.** The swarm is inherently insatiable and thus always moving; even agents who are already on

may the best location will continue to move in succeeding frames. This modification halts the swarm if the previous best location maintains a fitness which exceeds threshold T_t (which can be determined dynamically with a 2-class Bayesian classifier on the object’s previous best scores).

Algorithm 1 incorporates the enhancements to the algorithm for video tracking.

4. EXPERIMENTAL RESULTS

4.1. Data

A 30-second video was recorded using an Axis PTZ-215 camera at 30 FPS (900 JPEG frames) and 704x480 resolution (Figure 1). The video consists of 4 moving objects in camera in constant motion and includes partial and full occlusion of all targets.

The fitness function used in this paper is 20-bin color histogram [11] consisting of eight $B - G$ and eight $G - R$ bins (chrominance) and four $R + G + B$ bins (luminance), sampled in a 9x9 circle centered on the evaluated point.

4.2. Parameters

Due to Elitism of agents, it is beneficial to use a high elimination-dispersal probability $P_{ed} = 50\%$. This has the effect of freezing the best agents in place and spreading other agents out, thereby increasing the coverage of the search space. For classic BFO however, a low $p_{ed} = 2\%$ is chosen in order to allow the best agents to maintain their locations (without immunity to the elimination step, agents are easily dispersed from a target). Similarly, a larger step size $C = 10$ image pixels is selected in order to climb gradients more quickly, whereas it is set to $C = 2px$ in classical BFO since there is a high risk of overstepping out of a good area of fitness. Also, the number of chemotaxis steps $N_c = 1$ was experimentally determined after discovering that higher track accuracy could consistently be achieved (with the same amount of resources) by instead increasing N_r .

4.3. Implementation

The tested algorithms are implemented in C++ with OpenCV and are performed on a 2.4GHz Intel Core2 CPU with no additional multi-threading optimizations.

4.4. Evaluation Metrics

Track accuracy is measured as the percentage of frames in which the swarm correctly locates the object:

$$\text{accuracy} = \frac{\# \text{ of frames object correctly located}}{\text{total \# of frames object is visible}} \quad (1)$$

Ground truth object locations are first established off-line. Figure 2 shows an example fitness gradient and ground truth

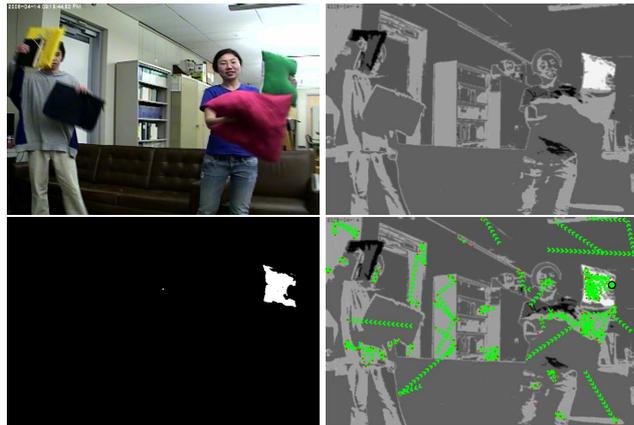


Fig. 2. From left to right: sample frame, fitness gradient for a green pillow, ground truth location of the green pillow, agents exploring and climbing the fitness gradient of the frame.

for an example tracked object. Frames per second (FPS) performance of each algorithm is also recorded (experiments are executed on offline data in order to achieve frame rates unrestricted by limitations of the camera hardware).

4.5. Results

Figure 3 shows the performance of the algorithms on the dataset. We see that the proposed BFO algorithm maintains a higher track accuracy against classical BFO and PSO with decay. Speed improvements can also be seen, with the proposed changes being capable of producing 65-92% speed improvements over classical BFO. Figure 4 shows the incremental performance gain of each proposed improvement on an example configuration. Lookahead capabilities introduces a significant gain for accuracy as agents remain located on a target (also resulting in a speed gain since fewer agents remain searching). Introducing elitism yields similar speed and accuracy gains (fewer agents continue searching and a larger pool of candidate positions to choose from, respectively). Early termination produces only modest accuracy gains while speed improvements are slightly more substantial.

Additionally, all the swarm intelligence algorithms outperformed the CamShift algorithm in track accuracy. This is because the algorithm expresses difficulty in reacquiring targets following periods of occlusion or once it has lost a target due to quick movement.

5. CONCLUSIONS

The results show that the proposed improvements to the BFO algorithm can be used to track objects through occlusion and camera movement, as well as out-perform Particle Swarm Optimization using fewer resources. Future work extends this approach to track multiple objects through multiple cameras.



Fig. 1. Four targets in a moving camera (red pillow, green pillow, person wearing a blue shirt, and yellow bag)

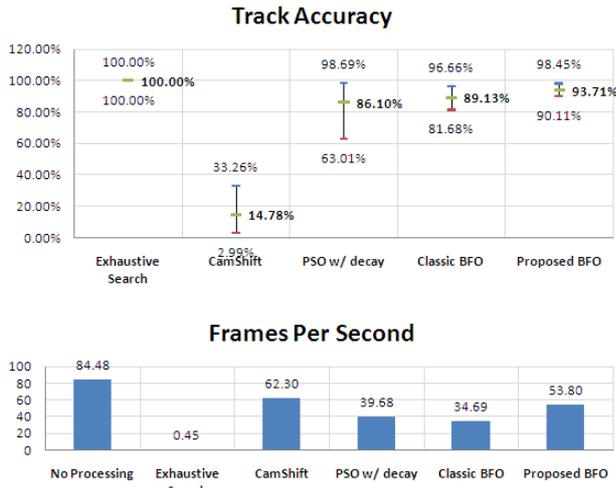


Fig. 3. Performance comparison on tracking 4 targets in a moving camera, averaged over 5 tests. Parameters: PSO (2,048 agents/target, 1% decay), Classic BFO (50 agents/target, $N_r = 10$, $N_c = 1$, $N_s = 25$, $p_{ed} = 1$, $C = 2px$), Proposed BFO (50 agents/target, $N_r = 10$, $N_c = 1$, $N_s = 25$, $p_{ed} = 50\%$, $C = 10px$)

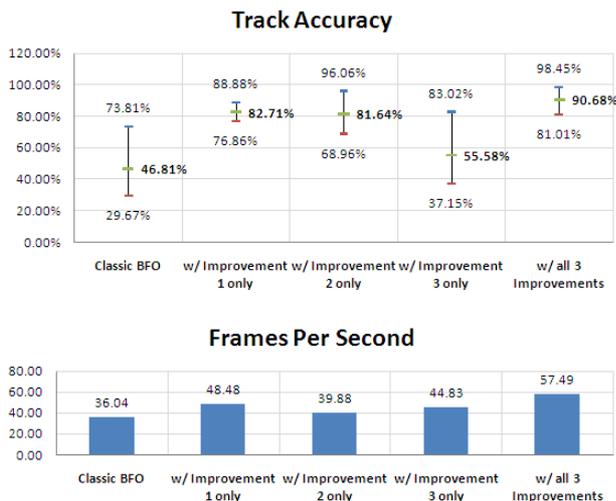


Fig. 4. Incremental contribution of each improvement on a BFO configuration. Parameters: 50 agents/target, $N_r = 10$, $N_c = 1$, $N_s = 25$, $p_{ed} = 15\%$, $C = 10px$

6. REFERENCES

- [1] Wei Ding, Zhenbang Gong, Shaorong Xie, and Hairong Zou, "Real-time vision-based object tracking from a moving platform in the air," *IEEE/RSJ Int. Conf. on Robots and Systems*, pp. 681–685, 2006.
- [2] Jinman Kang, I. Cohen, G. Medioni, and Chang Yuan, "Detection and tracking of moving objects from a moving platform in presence of strong parallax," *ICCV 2005*, vol. 1, pp. 10–17 Vol. 1, 2005.
- [3] Kevin M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. Vol. 22, No. 3, pp. 52–67, 2002.
- [4] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 142–149 vol.2, 2000.
- [5] Gary R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal Q2 '98*, 1998.
- [6] David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, August 2002.
- [7] Michael Isard and Andrew Blake, "Condensation conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, IEEE Int. Conf.*, vol. 4, pp. 1942–1948 vol.4, 1995.
- [9] Y. Owechko and S. Medasani, "Cognitive swarms for rapid detection of objects and associations in visual imagery," *Swarm Intelligence Symposium, IEEE*, pp. 420–423, 2005.
- [10] Anthony Jack Carlisle, *Applying the Particle Swarm Optimizer to Non-stationary Environments*, Ph.D. thesis, Auburn University, 2002.
- [11] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *CVPR 1998*, pp. 232–237, 1998.