Video Object Tracking with Heuristic Optimization Methods

Chin-Shiuh Shieh, Yong-Shixa Jhan, Yuan-Li Liu, Mong-Fong Horng, and Tsair-Fwu Lee
Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan, R.O.C.
Email: csshieh@nkust.edu.tw

***Abstract*—Object tracking is a common and essential task in video processing. This study approaches the object tracking problem using heuristic optimization methods. HSV color space is used as features for object matching. We evaluate the performance of particle filter, particle swarm optimization and grey wolf optimizer. Tracking rate, tracking accuracy and tracking time are important criteria in our comparative study. Experimental results reveal that particle swarm optimization prevails in object tracking applications.**

***Index Terms*—HSV color space, particle filter, particle swarm optimization, grey wolf optimizer**

I. Introduction

As computer vision technology getting mature, we rely on it more and more in our daily life. Road-side cameras, dash-cams and mobile phones play the role of deterrence in criminal prevention. They also provide effective evidence in criminal investigation. However, examining recorded or real-time video by human being is a time-consuming and error-prone process. As a result, intelligent monitoring systems with computer vision technology receive considerable attention in recent years.

Various technologies are involved in an intelligent monitoring system. We focus on object tracking of recorded video in this study. It aims to keep tracking of a particular object in each frame of a video clip.

In the beginning of an object tracking process, user specifies a region, a block for instance, containing the object of interesting in the first frame. System needs to find out where the object is located in the second frame. In most systems, a number of guesses/predictions will be made. In the second frame, the regions of predicted locations will be sampled and matched against the specified object. A predicted position with highest matching score will be taken as the location of the target in the second frame. The process repeats for every subsequent frame.

II. Image Processing

Object tracking is to locate the position of a specified object in each frame of a video clip. If the object to be tracked is not a rigid object, the shape, orientation and texture may vary from frame to frame. This makes the tracking task challenging.

To this problem, histogram of HSV color space is used as features for object matching in our study. The Bhattacharyya distance between histograms of target object and predicted image block is used as the fitness value for particle swarm optimization and grey wolf optimizer.

A. Color Spaces

RGB color space is commonly used in image processing. However, for the purpose of object tracking, RGB color space might not be the best option, since the target might be unidentifiable when the brightness changed due to shadow or illumination change.

HSV color space is a promising alternative to RGB color space for object tracking applications. In HSV color space, as shown in Fig. 1, each image pixel consists of 3 components, namely Hue h, Saturation s and Value v, where Value stands for brightness or lightness. Hue has its value from 0° to 360°. Saturation and brightness have their values between 0 and 1. In practice, each component of a pixel is mapped to integers from 0 to 255. That is, the color information of an image pixel is coded using 24 bits.

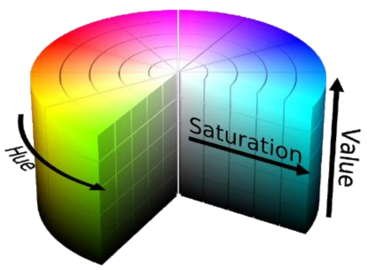

Figure 1. HSV color space.

B. Feature Descriptor

To match target image block and predicted image block, feature vectors are extracted from the raw pixels in the blocks. In our study, the features used for object matching have two parts. The first one is a vector formed using the histogram of hue values. The dimension is 360. The second part is a vector formed with the histogram of grey values. The dimension is 256.

HSV system has an isolated hue component which is an invariant for different illumination conditions. However, a grey-value pixel has a zero-value hue. To

Manuscript received July 12, 2018; revised December 11, 2018.

doi: 10.18178/joig.6.2.95-99

retain the information conveyed in grey-value pixels, we include the histogram of grey values as the second part of the feature vectors.

Given HVS values, the grey value g of a pixel is defined as follows:

$$g = \begin{cases} 0, & \text{if } v = 0 \\ v \cdot 255, & \text{if } s = 0 \end{cases}$$

Fig. 2 is an illustration of the cascaded feature vector with dimensionality 616.

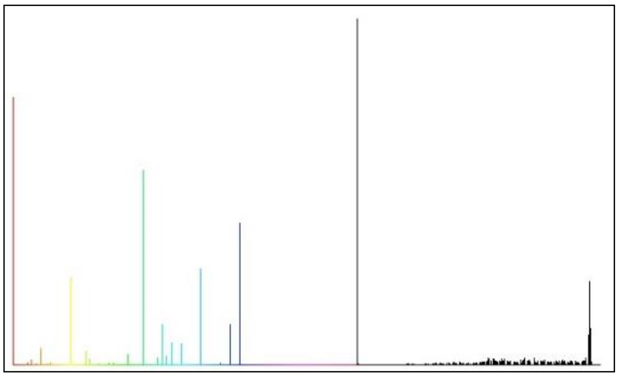

Figure 2. Feature vector formed by cascading histograms of hue and grey values.

Texture is another important attribute in identifying objects. However, the above-mentioned feature vector alone can't capture this vital information. An example is given in Fig. 3. Those two image blocks in Fig. 3 have quite different textures but share exactly the same feature vector.

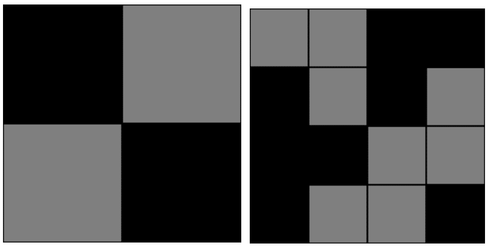

Figure 3. Image blocks with different textures have the same color histogram.

To this problem, we partition an image block into 4 sub-blocks, as illustrated in Fig. 4. Each sub-block has its own 616-dimensional feature vector. Feature vectors from sub-blocks are then cascaded to form a 2464-dimensional feature, which will be used for image block matching in our experiments.

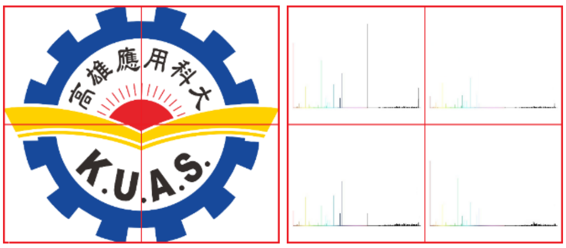

Figure 4. Sub-block partitioning.

C. Measurement of Similarity

Bhattacharyya distance is commonly used to measure the similarity of 2 items in statistics. It is defined as follows:

$$D_B(p, q) = -\ln(BC(p, q))$$

where $BC(p, q)$ is the Bhattacharyya coefficient, evaluated according to:

$$BC(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)}$$

Bhattacharyya coefficient is used to evaluate the similarity between 2 probability distributions. The feature vectors in this study can be transformed into probability mass functions after normalization. As a result, Bhattacharyya distance can then be used to measure the similarity between the target image block and a predicted image block. The transformation is done according to the followings:

$$PMF_u = \frac{H_u}{H}$$

where PMF_u is the probability of *u*-th hue value, and $H = \sum_{u=1}^{U} H_u$ is the summation of all histogram values. It equals to the number of pixels in an image block. The Bhattacharyya distance between target image and predicted image serves as the fitness value in particle swarm optimization and grey wolf optimizer.

III. Heuristic Optimization Methods

Heuristic optimization methods are well embraced in recent year for their potential in solving real-world, complex problems. Although global optima can't be guaranteed, in practice, near-optimal solutions can be found within justified computational cost. In this article, we examine the feasibility of the application of heuristic optimization methods to the video object tracking problem. Two heuristic optimization methods, particle swarm optimization (PSO) and grey wolf optimizer (GWO), and one classical method, particle filter (PF), are included in our comparative study.

A. Particle Swarm Optimization

Proposed by J. Kennedy and R. C. Eberthart, particle swam optimization [1]-[5], is a population-based heuristic optimization method. It imitates the interaction among individuals in a school of fishes or a flock of birds. In PSO, an individual is called a particle. Particles wander in the search space to conduct a biased random search. The location of each particle is a potential solution. At a given location, a particle has its fitness value according to the defined fitness function. Particles share with each other their experience on fitness values at different locations.

At each iteration, every particle updates its location in order to move toward more promising region. The movement is biased by the best experience of the particle itself and the best experience of the entire population.

$$v^{t+1} = w \cdot v^t + c_1 \cdot \mathrm{RAND} \cdot (\mathrm{pbest} - x^t) + c_2 \cdot \mathrm{RAND} \cdot (\mathrm{gbest} - x^t)$$

$$x^{t+1} = v^{t+1} + x^t$$

where x^t and v^t are location and velocity of a particle at iteration *t*. RAND is a random number from $[0,1)$. pbest and gbest are locations of best experience of individual and population, respectively. c_1 and c_2 are weighting factors for individual experience and population experience.

w is the inertial weight. It controls the tradeoff between exploration and exploitation. In order to have better convergence characteristic, it is in general an adaptive parameter varies according to the following setting:

$$w = W_u - (W_u - W_l) \cdot \left(\frac{g}{G}\right)$$

where W_u and W_l are upper and lower bounds for w. 0.9 and 0.4 are typical settings. g is the current iteration and G is the maximal number of iteration.

B. Grey Wolf Optimizer

Grey wolf optimizer was pioneered by Mirjalili in 2014 [6], [7]. Grey wolf is on the top of food chain and lives in group. There are in general 5 to 12 wolfs in a wolf pack. Grey wolf has strict social hierarchy. In the hierarchy, there are α wolf, β wolf, δ wolf, and ω wolfs in descending order. As a model for optimization, the α wolf, β wolf and δ wolf are the top 3 best solutions. The rest wolfs are ω wolfs serving as candidate solutions. The movement of ω wolfs are guided by α, β and δ wolfs.

In the hunting process, there are 3 phases, namely encircling, hunting and attacking. The location of a wolf is updated according to:

$$X(t+1) = X_p(t) - A \cdot D$$

$$D = \left|C \cdot X_p(t) - X(t)\right|$$

where *t* is the index of iteration; $X(t)$ and $X_p(t)$ are the locations of a wolf and the prey at iteration t; *D* is the distance between the wolf and the prey. *A* and *C* are random coefficients:

$$A = 2a \cdot r_1 - a$$

$$C = 2 \cdot r_2$$

$$a = 2 - 2 \cdot \frac{t}{\max}$$

where r_1 and r_2 are random numbers from $[0,1)$. max is the maximal number of iterations. As a result, a linearly decreases from 2 to 0 as *t* increases.

During the hunting process, the prey runs away as wolfs approach. As shown in Fig. 5, the new position of the prey can be estimated as follows:

$$\begin{cases} D_\alpha = \left|C_1 \cdot X_\alpha(t) - X(t)\right| \\ D_\beta = \left|C_2 \cdot X_\beta(t) - X(t)\right| \\ D_\delta = \left|C_3 \cdot X_\delta(t) - X(t)\right| \end{cases}$$

$$\begin{cases} X_1 = X_\alpha(t) - A_1 \cdot D_\alpha \\ X_2 = X_\beta(t) - A_2 \cdot D_\beta \\ X_3 = X_\delta(t) - A_3 \cdot D_\delta \end{cases}$$

$$X_p(t+1) = \frac{(X_1 + X_2 + X_3)}{3}$$

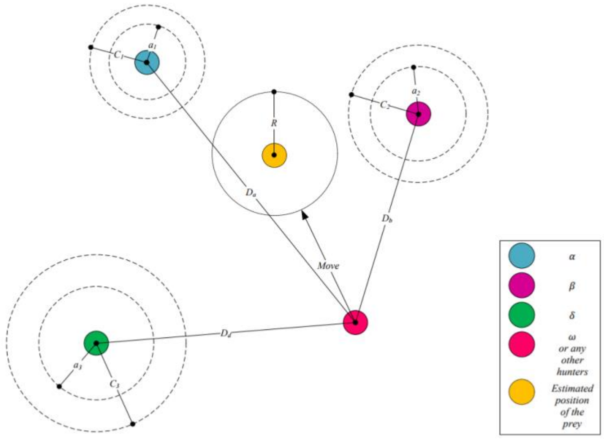

Figure 5. Illustration of position updating from [6].

IV. Experiments

We experiment the performance of Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO) and Particle Filter (PF) [8]-[12] in video object tracking. Since stochastic processes are involved in all 3 algorithms, we report the average of 10 runs with different random seeds. The maximal number of iteration for each run is 100.

A. Performance Metrics

Capability of successful tracking is the most fundamental performance indicator. For each frame, the tracking is considered failed if there is no overlap between the target image block and the predicted image block. The successful tracking rate is defined as:

$$\mathrm{Rate}_s = \frac{\mathrm{frame}_a - \mathrm{frame}_f}{\mathrm{frame}_a} \times 100\%$$

where $frame_a$ is the number of frames in a video clip and $frame_f$ is the number failed frames.

We also observe the average time it takes for different algorithm to converge to a best solution in a frame.

In order to measure the tracking accuracy, we define the tracking error as follows:

$$D_x = \left|O_x - M_x\right|$$

$$D_y = \left|O_y - M_y\right|$$

$$D = \sqrt{{D_x}^2 + {D_y}^2}$$

where (O_x, O_y) and (M_x, M_y) are the coordination of the target image block *O* and the predicted image block *M*.

B. Experimental Results

Table I presents the result when population size is 30. It can be seen that PSO and GWO take much longer execution time than PF does. There are also considerable tracking errors. It implies that both PSO and GWO are likely to be trapped at local optima. It could be a result of insufficient diversity due to the small population size.

TABLE I. POPULATION SIZE = 30

	PSO	GWO	PF
x-axis Tracking Error	57.65	59.28	46.44
x-axis Tracking Error	53.1	48.34	93.91
Tracking Error	86.72	82.83	110.58
Execution Time	205.09	224.82	57.34
Tracking Rate	65.08%	67.46%	57.14%

The performance has significant improvement as the population size increase to 50, as shown in Table II.

TABLE II. POPULATION SIZE = 50

	PSO	GWO	PF
x-axis Tracking Error	25.27	29.55	48.29
x-axis Tracking Error	23.36	27.19	101.27
Tracking Error	37.95	43.73	116.18
Execution Time	281.63	359.78	73.04
Tracking Rate	91.27%	88.89%	52.38%

As the population size increase from 30 to 50, there is a 67% increase in the storage. PSO has a 37% increase in execution time, while GWO has a 60% increase. It means that, for larger population size, although tracking rate for GWO is improved. Its execution time is also increased. As for PF, there is no clear improvement, neither in tracking rate nor tracking accuracy.

TABLE III. POPULATION SIZE = 50

	PSO	GWO	PF
x-axis Tracking Error	9.06	12.9	41.15
x-axis Tracking Error	8.11	9.78	89.18
Tracking Error	13.44	17.77	102.66
Execution Time	472.65	758.19	71.91
Tracking Rate	99.99%	99.99%	53.97%

Table III reports the experimental results when the population size is set to 100. Tracking errors with PSO and GWO reduce to below 10 pixels. Their tracking rates are nearly 100%. Referring to Table II and III, when the population size doubled, the execution time of PSO is increased by 67%, while GWO has a 107% increase. The tracking rate and accuracy of GWO are improved for larger population size. However, the computational cost is also increased. For a population size of 30, PSO's tracking accuracy is slightly inferior to that of GWO. For a population size of 50 or 100, PSO prevails in all aspects.

V. CONCLUSION

This article reports our experience in video object tracking using heuristic optimization methods. Feature vectors based on histograms of hue and grey values are used for object matching. A comparative study had been made among PSO, GWO and PF. Experimental results show that heuristic optimization is a feasible approach to the video object tracking problem. Nearly 100% tracking rate is achievable when the population size is large enough. PSO is superior to GWO and PF, at least with our experiment settings.

ACKNOWLEDGMENT

We are grateful to the support from the Ministry of Science and Technology, Taiwan, ROC, under the grant number MOST 106-2221-E-151-007 -.

Corresponding author: Chin-Shiuh Shieh; E-mail: csshieh@nkust.edu.tw.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[2] Z. Zhan, J. Zhang, Y. Li, and H. S. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 6, pp. 1362-1381, 2009.

[3] R. Ugolotti, Y. S. G. Nashed, P. Mesejo, S. Ivekovi, L. Mussi, and S. Cagnoni, "Particle swarm optimization and differential evolution for model-based object detection," *Applied Soft Computing*, vol. 13, no. 16, pp. 3092-3105, 2013.

[4] R. Serajeh, K. Faez, and A. E. Ghahnavieh, "Robust multiple human tracking using particle swarm optimization and the Kalman filter on full occlusion conditions," in *Proc. First Iranian Conference on Pattern Recognition and Image Analysis*, 2013, pp. 1-4.

[5] G. Liu, Y. Y. Chung, and W. Yeh, "A simplified swarm optimization for object tracking," in *Proc. International Joint Conference on Neural Networks*, 2016, pp. 169-176.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[7] Y. Bai, J. Chen, and Z. Zhan, "Analysis and research of improved grey wolf optimization algorithm," *Computer Science and Application*, vol. 7, no. 6, pp. 562-571, 2017.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transaction on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[9] P. D. Moral, "Nonlinear filtering: Interacting particle resolution," *Markov Processes and Related Fields*, vol. 2, no. 4, pp. 555-580, 1996.

[10] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107-113, 1993.

[11] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic system," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032-1044, 1998.

[12] G. Xia and S. A. Ludwig, "Object-tracking based on particle filter using particle swarm optimization with density estimation," in *Proc. IEEE Congress on Evolutionary Computation*, 2016, pp. 4151-4158.

Chin-Shiuh Shieh received his Ph.D. degree from Department of Computer Science and Engineering, National Sun Yat-Sen University, Taiwan, in 2009. He now serves as an associate professor in Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Taiwan. His research interests include computer networking, embedded system and computational intelligence.

Yong-Shixa Jhan received his M.S. degree from Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Taiwan, in 2018. His research interests include image processing and computational intelligence.

Yuan-Li Liu received his M.S. degree from Department of Electronic Engineering, I-SHOU University. Taiwan, in 2004. He now is a PhD student in Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Taiwan, and serves as an assistant researcher in NCSIST, Taiwan. His research interests include machine learning, and Artificial Intelligence.

Mong-Hong Horng received his PhD degree in Computer Science and Information Engineering from National Cheng Kung University, Taiwan in 2003. Currently, he is a Professor with Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Taiwan. His current research interests include computational intelligence, embedded systems and computer networks.

Tsair-Fwu Lee is currently a professor at the Medical Physics and Informatics Laboratory of Electronics Engineering Department at National Kaohsiung University of Science Technology. Dr. Lee has published more than 70 peer-reviewed journals and 100 conference papers. His major research interests area: medical physics and medical informatics, signals and systems, radiation oncology, radiation biophysical modeling, evolutionary optimization and intelligent algorithms.