Swarm Strategy and Color Probability Density for Visual Tracking

BTSym2018 paper ID: 067

***Abstract*—This article introduces a new target tracker for videos and investigates the discriminatory power of color probability density in target identification. The nonparametric Kernel Density and Histogram estimators were used. The results showed that the performance of the proposed tracker is as robust as the Mean Shift, but the results of the proposed tracker are more stable. It was also observed that the target identification was best when the histogram was used to estimate the color probability density.**

Keywords – Visual Tracking, BBPSO, Mean Shift, Kernel, Histogram.

I. Introduction

Video target tracking is an active area of computational vision research and has a wide range of applications in real world problems [1]. In spite of the various proposed approaches, target tracking remains a challenging task, since several factors can affect the performance of the proposed models, such as object occlusion, non-rigid deformations, fast movements, variation in lighting, low processing time and rotations.

The tracking methods can be categorized into two groups [1]: type 1: Detection tracking that identifies the target in each frame by comparing the characteristics of the candidate targets and the target model such as Mean Shift [2]; type 2: Tracking by probabilistic methods that predict the unobserved position of the target in the current frame given the observed target positions in the previous frames, such as Kalman Filter [3] and Particle Filter [4].

The population-based stochastic optimization techniques have been employed along with the tracking algorithms of both groups [5] [6] [1] [7].

In this article, a type 1 tracker based on Bare Bones Particle Swarm Optimization algorithm [8], here called **TBBPSO** (T: Tracker) is proposed. In each frame of a video the **TBBPSO** spreads a cloud of particles in a region of the image until the highest fitness particle is found. Each particle represents a candidate target and the fitness is given by the quadratic distance of Bhattacharyya between the color probability densities of the model and the candidate targets.

In this work, the central objective is to investigate the performance of **TBBPSO** by analyzing the robustness and precision of the tracking. For this, a random sample of ten benchmark videos was obtained from *Visual Tracker Benchmark/Hanyang* [9] and the results were compared to the results of the Mean Shift tracker.

An investigation was also made on the ability to discriminate the color density of the analyzed objects as a characteristic of target identification. Two nonparametric density estimators were used, the histogram and the kernel, and the results were also compared.

The structure of this paper is as follows: section II presents the video target tracking method, the nonparametric density estimators, the histogram and the kernel estimator, and the Mean Shift algorithm. Section III presents the proposed tracker. The planned experiment, results and discussions are presented in section IV. Finally, section V presents the conclusions.

II. Tracking Methods

A. Target Tracking in Videos

In this work only type 1 trackers were considered. A review of type 2 trackers can be found in [10].

Type 1 trackers in general detect the target in the current frame of the video seeking the target candidate with the highest similarity to the target model. The measure of similarity is obtained by comparing the appearances of the targets witch are composed by a function of predetermined characteristics of the objects. For different characteristics, appearance models and measures of similarity, see [1] [10].

The characteristic adopted in this work is the color probability density of the targets that is a robust measure in relation to the rotations, deformations and the change of scale of the targets [11]. It is also adopted the appearance model as characteristic and the quadratic distance of Bhattacharyya [12] as measure of similarity.

B. Density Estimators

Let $f(x)$ be a probability density function (pdf) and $x_1, x_2, ..., x_n$ be an observed random sample of size n of $f(x)$. The histogram is a nonparametric density estimator and is given by the following equation:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} I_{B_k}(x_i), x \in B_k, \tag{1}$$

where B_k represents the interval or bin k, ($k = 1, ..., n_k$), h represents the width of the bins, and the indicator function $I_A(x)$ is defined as follows:

$$I_A(x) = \begin{cases} 1, & x \in A, \\ 0, & otherwise, \end{cases} \tag{2}$$

The parameter h, called the smoothing parameter, plays a predominant role in the estimation process. If h is large the histogram is very smoothed and hides regions of high probability density; if h is small it produces a histogram with much variation showing false modes. Therefore, finding a suitable intermediate value for it is critical.

We adopted the Sturges rule that determines the number of bins as follows:

$$n_k = \lceil 1 + 3.32 \log n \rceil, \tag{3}$$

where n is the sample size, log is the logarithm at base 10 and $\lceil x \rceil$ is the smallest integer greater than or equal to x.

Therefore, in equation 1, h can be estimated by

$$\hat{h} = \frac{R_X}{n_k}, \tag{4}$$

where R_X is the sample amplitude and n_k is given by equation 3 [13].

Although the histogram is a good tool for data analysis, it is not very efficient as a density estimator due mainly to the points of discontinuity at the borders between two consecutive bins. An alternative density estimation proposal is the kernel density estimator [14] [15] [13].

Given a random sample of size n of the density of probability $f(x)$, the kernel density estimator is defined as:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} \ker \left(\frac{x - x_i}{h} \right), \tag{5}$$

where the function $\ker(x)$ is called the kernel function.

A kernel function must be continuous on the support of variable X, the integral over the domain of X must be equal to 1 and $\ker(x)$ must be symmetric in origin, besides having the first two finite moments. In general, the kernel function is defined by a pdf which the Normal kernel is the most popular. Therefore, $\ker(x)$ inherits all the properties of the probability density, such as continuity, symmetry, and differentiability [16].

The smoothing parameter h, also called window size or bandwidth in the kernel estimation context, remains fundamental. There are other factors that influence the final quality of the kernel estimator, for example, the choice of the function $\ker(x)$. However, h is the main factor that affects the quality of the estimator.

The Mean Integrated Squared Error, $MISE$, measures the quality for both the kernel density and histogram estimators and is defined as:

$$MISE = \mathbb{E} \left[\int_{\Omega_X} \left(\hat{f}(x) - f(x) \right)^2 dx \right], \tag{6}$$

where $\mathbb{E}(X)$ is the expected value of the random variable X and Ω_X is the sample space of X.

An estimator of the h parameter that minimizes $MISE$ using the Normal kernel function is given by:

$$h^\star = \left(\frac{4}{3n} \right)^{1/5} \sigma \cong 1.06\, \sigma\, n^{-1/5}, \tag{7}$$

where σ^2 is the theoretical variance, so σ can be replaced by the sample standard deviation, s, or by $IQR/1.348$. IQR is the sample interquartile range [16].

For the multivariate case, given a d-dimensional random sample of size n of the multivariate probability density $f(\mathbf{x})$, the multivariate kernel density estimator is given by

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} |\mathbf{H}|^{-1/2} \ker_d \left(\mathbf{H}^{-1/2} (\mathbf{x} - \mathbf{x}_i) \right), \tag{8}$$

in which the smoothing parameter matrix $\mathbf{H}$ is a symmetric positive definite matrix of order d. In general, $\mathbf{H} = \mathbf{diag}[h_1^2, h_2^2, ..., h_d^2]$ or $\mathbf{H} = h\mathbf{I}$ with h a scalar smoothing parameter and $\mathbf{I}$ is the identity matrix of order d [17].

A multivariate kernel $\ker_d$ can be generated by the product of a univariate kernel $\ker_1$ or by the rotation of the univariate kernel $\ker_1$ in the $\mathbb{R}^d$.

C. The Mean Shift Algorithm

The Mean Shift tracker, called in this article by **TMS** (T: Tracker and MS: Mean Shift), is based on the Mean Shift algorithm [17] [2]. Its choice is due to the fact that it is a classic type 1 tracker that is already well tested and compared to several other trackers [18]. In addition, the **TMS** adopts kernel estimation of color probability density as a characteristic of targets and the Bhattacharyya distance as a measure of similarity between the model and candidate targets.

In Mean Shift, the targets are represented by the color space density and are estimated by a special kernel class, called the kernel profile, which produces a radially symmetric kernel. More precisely, the densities are estimated by the Epanechnikov kernel profile that guarantees a minimum MISE estimate [17].

The target model is represented by the color probability density q and, in subsequent frames, a candidate target is defined at the location y and is characterized by the color probability density $p(y)$. Both densities are estimated by the same kernel profile.

The Bhattacharyya coefficient, $\rho(y)$, is used as a function of similarity and is interpreted as a correlation score between the probability density of the target model, q, and the canditate target, $p(y)$.

The term $\rho(y)$ is spatially smooth due to radially symmetric kernel usage and plays the role of likelihood, so the maximum $\rho(y)$ location in the image indicates the presence of the target.

To find the maximum of $\rho(y)$ the gradient-based optimization procedure is used. The gradient density estimator is obtained by the gradient of the density estimator by locally exploiting the linearity of $\rho(y)$ by the Taylor expansion around y [2].

III. The Proposed Tracker

In recent years, bioinspired algorithms based on populations of individuals or particles have been used to solve many types of problems where conventional solutions are difficult or impossible to be achieved. There are two population-based approaches, Evolutionary Algorithms and Swarm Algorithms. The methods based on swarm intelligence have increasingly attracted researcher's attention due to the robustness and flexibility of the results, especially in dynamic environments.

The PSO algorithm [19] mimics the social behavior observed in a swarm of birds. The learning of particles (birds) is given in two ways: cognitive learning, based on past individual experience of each particle, and, social learning, based on the collective behavior observed in a swarm.

The particles are possible solutions of the problem of interest and are positioned in the space of solutions. An objective function, called fitness, assigns a measure of solution's quality to each particle. Therefore, the position of the particle of maximum fitness is considered the best solution to the problem.

The PSO version used in this work is the Bare Bones PSO, BBPSO [8] that has two parameters defined by the user: the number of particles N and the neighborhood topology.

The neighborhood topology affects the way particles propagate the information within the swarm. The global neighborhood topology all particles in the swarm communicate. The local neighborhood topology each particle communicates with a predetermined set of particles [20]. In this work the global neighborhood topology was adopted.

The BBPSO uses a probability density to update the position of the particles instead of adding a velocity to the position as it is done in the standard PSO. The equation for updating the position of the particles in BBPSO is given by

$$\mathbf{p}_i^\tau = \mu_i^\tau + \sigma_i^\tau \odot \mathbf{Z}, \tag{9}$$

where $\odot$ is the element by element vector product, and

$$\begin{array}{rcl} \mu_i^\tau & = & 0.5\left(pBest_i^{\tau-1} + gBest^{\tau-1}\right), \\ \sigma_i^\tau & = & |pBest_i^{\tau-1} - gBest^{\tau-1}|, \end{array} \tag{10}$$

with $pBest_i^\tau$ the position of greatest fitness of particle i up to iteration τ, $gBest^\tau$ being the position of the greatest fitness of all particles up to iteration τ and $\mathbf{Z}$ is a d-dimensional random vector with multivariate Normal distribution with vector of means $\mathbf{0}$ and matrix of covariances equal the identity matrix of order d, $\mathbf{I}$ ($\tau = 1, 2, ..., n_\tau$ e $i = 1, 2, ..., N$).

The proposed tracker, called **TBBPSO**, estimates the target in the current frame of all video frames by spreading a swarm of particles over a region of the image. These particles are candidate targets and they move in this region in search of the position of greatest similarity to the target model.

Each particle represents a window $[x, y, s_f]$, where (x, y) is an image point corresponding to the center of the rectangular bounding box representing the candidate target and s_f is a scaling factor that controls the length and height of the bounding boxes (in number of pixels).

Given a video, **TBBPSO** seeks to identify the position of the target in the current frame first by reducing the search space. Second, N randomly generated particles are scattered in this reduced search space. Then, the BBPSO algorithm is executed until the stop conditions are reached. The highest fitness particle indicates the presence of the target in the current frame.

The reduced search space is a rectangular region of the image bounded around the central position of the target bounding box estimated in the previous frame.

Table I
DESCRIPTION OF THE BENCHMARK VIDEOS

Video	Name	T	Challenges
1	BlurBody	334	SV, DEF, MB, FM, IPR
2	BlurCar2	585	SV, MB, FM
3	BlurFace	493	MB, FM, IPR
4	Bolt2	293	DEF, BC
5	Boy	602	SV, MB, FM, IPR, OPR
6	Couple	140	SV, DEF, FM, OPR, BC
7	Dog	127	SV, DEF, OPR
8	MountainBike	228	IPR, OPR, BC
9	Surfer	376	SV, FM, IPR, OPR, LR
10	Twinnings	472	SV, OPR
Total	–	3650	–

The fitness of the particles is given by the quadratic distance Bhattacharyya between the color probability densities of the model and the candidate targets,

$$\beta^2(p_i) = 1 - \rho(p_i), \tag{11}$$

where $\rho(p_i)$ is the Bhattacharyya coefficient of the particle p_i, $\forall i = 1, 2, ..., N$.

Here, the Bhattacharyya coefficient is calculated for each particle and $\rho(p_i)$ is not a spatial function of the local y image as it is done in Mean Shift.

The **TBBPSO** uses the density histogram estimator with h given by equation 4, calculed on the target model.

A second version of **TBBPSO** uses the kernel density estimator with h given by equation 7, and to differentiate from the original **TBBPSO**, this version is called **TKBBPSO**. In this case, since the characteristic space is continuous, the distance between the targets is calculated in a quantized space with n_k bins. The larger the n_k the better the detection of difference between the probability density and greater the computational cost. Here, $n_k = 300$ bins.

In this paper, we used $N = 25$ particles and the stop condition of the BBPSO algorithm is up to 10 iterations or the Bhattacharyya distance smaller than 0.005 in both versions of the proposed tracker.

IV. RESULTS AND DISCUSSIONS

In this section, experimental results on ten randomly selected benchmark videos in *Visual Tracker Benchmark/Hanyang* are shown and discussed. The Table I lists the videos and their main features: The number of frames, T, and the main challenges for tracking: MB indicates blurred motion, FM indicates fast target movement, IPR indicates target rotation in the image plane, OPR indicates rotation of the target outside the plane of the image, DEF indicates deformation of the object, BC indicates confusion of the target with the background of the image, SV indicates scale variation and LR indicates low image resolution [9].

Figure 1 shows the first frame of each video.

Recall (R), Precision (Pr), F_1 and Pascal (Pa) measurements were adopted to measure the quality of trackers performance [21]. They measure the percentage of overlapping of the windows corresponding to the manually marked target (ε) and the target estimated by the trackers (ξ). All measures

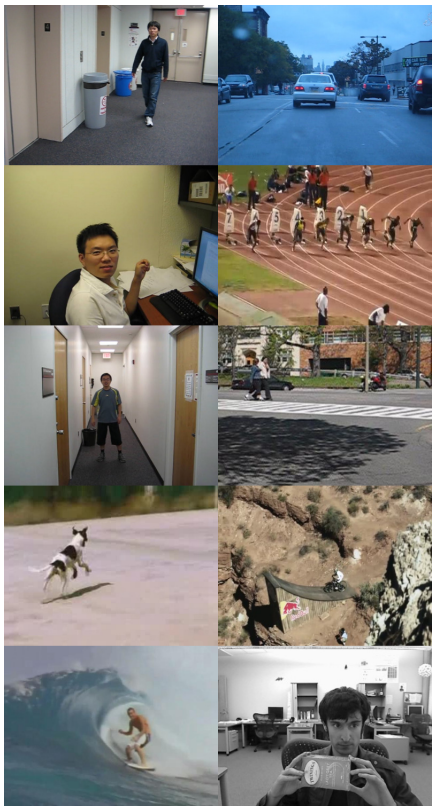

Figure 1. The frist frame of each video

range from 0%, for total lack of overlap, to 100%, for total overlap.

Recall measures how much ξ is covered by ε and is defined by

$$R(\xi,\varepsilon)=\frac{|\xi\cap\varepsilon|}{|\xi|}\times 100\% \tag{12}$$

where $|A|$ indicates the cardinality of the set A. Similarly, Precision measures how much ε is covered by ξ and is defined by

$$Pr(\xi,\varepsilon)=\frac{|\xi\cap\varepsilon|}{|\varepsilon|}\times 100\% \tag{13}$$

A good tracker has both Pr and R close to 100%, so the measure F_1, defined as the harmonic mean between Pr and R, measures the spatial quality of window overlays and is defined by

$$F(\xi,\varepsilon)=\frac{2Pr(\xi,\varepsilon)R(\xi,\varepsilon)}{Pr(\xi,\varepsilon)+R(\xi,\varepsilon)}\times 100\% \tag{14}$$

Pascal's measure determines the overlap of windows as follows:

$$Pa(\xi,\varepsilon)=\frac{|\xi\cap\varepsilon|}{|\xi\cup\varepsilon|}\times 100\% \tag{15}$$

Table II
PERFORMANCE MEASURES OF THE **TBBPSO**

Video	$Pa^{(1)}$	Pr	R	F_1	$Pa^{(2)}$
1	61.2	84.2	68.6	75.1	80.5
2	41.7	60.5	51.2	54.6	37.4
3	63.5	81.5	73.9	77.2	93.9
4	32.6	47.4	41.1	44.0	23.9
5	57.5	71.1	73.5	71.7	69.3
6	46.4	67.0	52.8	58.3	47.9
7	35.4	36.9	83.3	50.0	22.8
8	55.3	84.1	59.4	69.0	67.1
9	06.9	08.9	08.0	08.3	09.3
10	35.9	43.1	59.8	48.0	29.9
Mean	43.6	58.5	57.2	55.6	48.2
Median	44.0	63.8	59.6	56.5	42.7
Std	17.1	24.4	21.4	20.4	28.2
cv	39	42	37	37	58

The Pascal and F measurements are more robust than the Pr and R measurements since the Pr and R measurements can assume 100% values in two unwanted situations: When a tracker estimates the entire frame image as the target, in this case $Pr=100\%$, or when the tracker estimates a single pixel of the marked target, in this case, $R=100\%$. In general, Pascal's measure is used to consider whether a target was detected and this will be adopted in this work.

The target is considered detected in a given frame if the Pascal measure exceeds a threshold th. In this work, $th=50\%$.

All ten videos were used by each of the trackers, and the target model is the marked object in the first frame of each video.

An Intel Pentium dual-core processor of 1.86 GHz and 2 GB DDR2 was used. The trackers were programmed and executed using MATLAB. The **TMS** tracker was executed using the toolbox developed by Dollar [22]. Each video was run five times and was obtained the median of the values.

Tables II, III and IV present the results of the experiment performed, respectively, for the **TBBPSO**, **TKBBPSO** and **TMS** tracker. In Tables II, III and IV, for each video, $Pa^{(1)}$ is the median value of the variable Pa and $Pa^{(2)}$ is the percentage of frames detected by the variable Pa. cv represents the coefficient of variation in percentage values and Std is the standard deviation of the mean.

For all the trackers, the values observed in Tables II, III and IV show a remarkable variation of the results, cv ranges from 37% to 66% of the scale of the variables. This can be explained by the small sample size and by the different degree of challenge present in each video.

However, considering the medians of the variables $Pa^{(1)}$, F_1 and $Pa^{(2)}$, the largest percentage difference does not exceed 13% and this percentage does not reach 6% when the differences between means are observed. The median analysis is safer because they are more robust to outliers than meam ones. This empirical evidence points to a similar performance of the three trackers.

By observing the coefficients of variation of the variables

Table III
PERFORMANCE MEASURES OF THE **TKBBPSO**

Video	$Pa^{(1)}$	Pr	R	F_1	$Pa^{(2)}$
1	64.0	86.4	71.2	77.6	90.1
2	27.1	39.1	34.6	36.3	20.9
3	62.5	76.7	76.6	76.6	88.6
4	31.4	40.7	40.0	40.1	34.5
5	57.7	73.6	72.2	72.2	74.6
6	41.1	62.9	47.1	53.1	42.1
7	35.0	36.2	88.0	49.5	22.1
8	14.7	18.8	17.5	18.1	15.8
9	20.3	33.3	21.6	25.6	23.4
10	38.7	45.2	63.1	50.6	29.7
Mean	39.3	51.3	53.2	49.9	44.2
Median	36.9	42.9	55.1	50.1	32.1
Std	17.3	22.2	24.4	20.7	29.0
cv	44	43	46	41	66

Table IV
PERFORMANCE MEASURES OF THE **TMS**

Video	$Pa^{(1)}$	Pr	R	F_1	$Pa^{(2)}$
1	32.9	85.5	33.6	44.8	24.6
2	21.7	22.1	69.2	31.2	15.0
3	74.5	89.0	83.1	84.7	92.3
4	52.8	90.4	58.2	67.5	55.0
5	64.9	78.3	82.2	77.4	82.7
6	67.1	77.9	79.8	77.8	87.1
7	43.3	50.6	78.5	58.5	33.9
8	11.0	12.8	15.7	13.8	13.6
9	36.0	64.7	41.5	45.2	35.6
10	38.3	87.9	42.6	48.7	35.4
Mean	44.2	65.9	58.5	55.0	47.5
Median	40.8	78.1	63.7	53.6	35.5
Std	20.5	28.4	23.9	22.6	30.0
cv	46	43	41	41	63

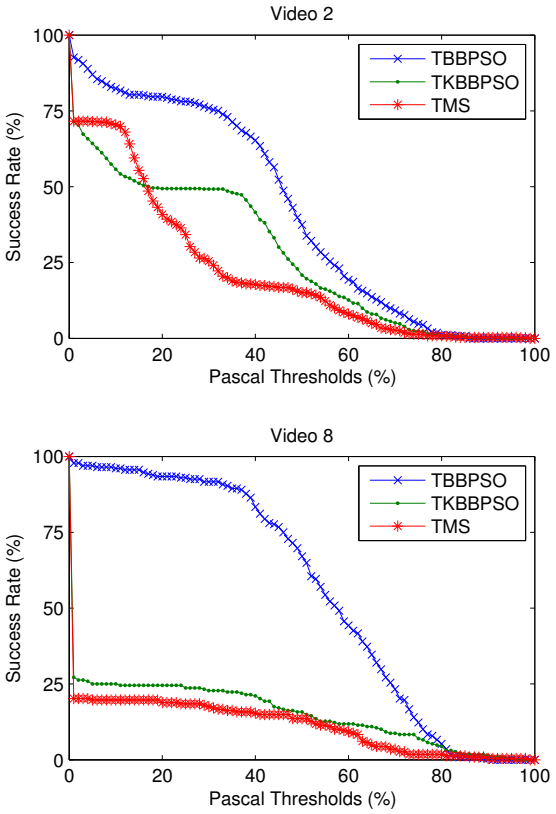

Figure 2. Detection Rate for Pascal thresholds

$Pa^{(1)}$ and $Pa^{(2)}$ we can observe a variation of 10% in relation to the scale, this implies a difference in the detection precision of the windows. In particular, the cv of the **TMS** is 7% and 5% larger than the cv of **TBBPSO** for $Pa^{(1)}$ and $Pa^{(2)}$, respectively. The cv of **TKBBPSO** is 7% and 8% higher than the cv of **TBBPSO** and the cv of the **TKBBPSO** and **TMS** trackers are very close for the respective variables. This indicates a better stability of detection in favor of **TBBPSO**.

Figure 2 shows the graph of success rate by Pascal thresholds for videos 2 and 8. The x-axis of the graph shows the Pascal thresholds ranging from 0% to 100% and the y-axis shows the success rate which is the percentage of video frames whose ξ and ε have overlaps that exceed the threshold *th*.

The curves show the performance of each one of the trackers (the blue curves refer to the **TBBPSO** tracker, green to **TKBBPSO** and red to **TMS**) and the more a curve stays at the top the better the tracker performance.

The **TMS** and **TKBBPSO** trackers detected the target in video 8 in less than 25% of the frames considering any Pascal threshold while the **TBBPSO** did not detect the target in less than 25% of the frames considering Pascal thresholds greater than 75%. Briefly, **TBBPSO** was superior to **TMS** and **TKBBPSO** for all Pascal thresholds. Also in video 2 **TBBPSO** outperformed **TMS** and **TKBBPSO** for all Pascal thresholds.

The **TBBPSO** performed better in scenarios where there were rotations, scale changes, fast movements and blurred movements of the target.

In general, in videos where **TBBPSO** was outperformed by **TMS**, the curves remained close to all thresholds. The analysis of the curves reinforces the fact that **TBBPSO** is more stable in tracking different videos, that is, the measures of Pascal generated by **TBBPSO** in different videos are more homogeneous.

To test the performance of trackers, each tracker will be considered an experimental treatment in which one intends to measure the effects on the videos (videos are experimental units). In this case, the results are an achievement of a randomized block experiment since all units are used in all treatments.

Since, on average, the sample correlation between F_1 and $Pa^{(1)}$ is 0.9885 with a standard deviation of 0.0034, there is no need to test the variable F_1. Therefore, a two-way ANOVA for the $Pa^{(1)}$ and $Pa^{(2)}$ variables will be done [23].

The nonparametric test χ^2_F of Friedman [24] was applied in the rank variable and if the null hypothesis, H_0, of the equality of the medians is rejected, Nemenyi Post-hoc tests [25] will

be applied to the difference variables among all possible pairs.

In both variables $Pa^{(1)}$ and $Pa^{(2)}$, the test did not rejected H_0 at the 5% level of significance. It is not possible to claim if there is a tracker with superior performance to the others. Also, it cannot be affirmed that using the kermel for density estimator improves the identification capacity of the targets in comparison to the histogram estimator, so it is concluded that the use of the histogram is more advantageous due to its simplicity and low computational cost.

V. Conclusions

The proposed tracker **TBBPSO**, based on collective intelligence strategy and target color probability density estimated by the histogram, was able to track any moving target in challenging scenarios with performance compatible with the Mean Shift. However, empirical evidence indicates that **TBBPSO** produces more stable results for diferents videos among the trackers surveyed, that is, the measures of Pascal generated by **TBBPSO** in different videos are more homogeneous.

The use of the kernel as an estimator of the color probability density of the targets does not improve the tracker's ability to identify targets when compared to the histogram. Therefore, the use of histogram is more advantageous due to its simplicity and low computational cost.

Future work will be done to improve the performance of **TBBPSO** by testing other isotropic kernels in the spatial domain and a new adaptive appearance model based on color densities.

References

[1] F. Sardari and M. E. Moghaddan, "A hybrid occlusion free object tracking method using particle filter and modified galaxy search meta-heuristic algorithm," *Applied Soft Computing*, 2017.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[3] S. Weng, C. Kuo, and S. Tu, "Video object tracking using adaptive kalman filter," *J. Visual Commun. Image Represent.*, 2006.

[4] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEEE Proceedings on Radar Signal Process*, 1993.

[5] A. M. A. Tawab, M. A. Abdelhalim, and S. E. D. Habib, "Efficient multi-feature pso for fast gray level object-tracking," *Applied Soft Computing*, 2014.

[6] M. L. Gao, L. L. Li, X. M. Sun, L. J. Yin, and H. T. Li, "Firefly algorithm (fa) based particle filter method for visual tracking," *Optik*, 2015.

[7] C. Bae, K. Kang, G. Liu, and Y. Y. Chung, "A novel real time video tracking framework using adaptive discrete swarm optimization." *Expert Systems With Applications*, 2016.

[8] J. Kennedy, "Bare bones particle swarm," *In Proceedings of IEEE on Swarm Intelligence Symposium*, 2003.

[9] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[10] G. M. Rao and C. Satyanarayana, "Object target tracking using particle filter: A survey," *I.J. Image, Graphics and Signal Processing*, 2013.

[11] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trend in visual tracking: A review," *J. Neurocomputing*, 2011.

[12] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Trans. Comm. Technology*, 1967.

[13] W. L. Martinez and A. R. Martinez, *Computational Statistic Handbook with MatLab*. Chapman & Hall/CRC, 2002.

[14] M. Rosemblatt, "Remarks on some nonparametric estimates of a density functions," *Annals of Mathematical Statistics*, 1956.

[15] E. Parzen, "On estimation of probability density function and mode," *Annals of Mathematical Statistics*, 1962.

[16] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[17] D. Comaniciu and P. Meer, "Mean shift: A roubust approach tward feature space analysis," *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[18] J. Dou and J. Li, "Robust visual tracking based on joint multi-feature histogram by integrating particle filter and mean shift," *Optik*, 2015.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," *In Proceedings of IEEE Interna-tional Conference on Neural Networks*, 1995.

[20] J. Kennedy, "Small world and mega-minds: Effests of neighborhood topology on particle swarm performance," *In Proceedings of the Congress on Evolutionary Computation*, 1999.

[21] M. Everingham, V. G. L., C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, 2010.

[22] P. Dollár, "Piotr's computer vision matlab toolbox (pmt)." [Online]. Available: https://github.com/pdollar/toolbox

[23] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, 2011.

[24] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, 1937.

[25] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, 2006.