



Article

Online Visual Tracking of Weighted Multiple Instance Learning via Neutrosophic Similarity-Based Objectness Estimation

Keli Hu ¹, Wei He ², Jun Ye ¹, Liping Zhao ¹, Hua Peng ^{1,3} and Jiatian Pi ^{4,*}

¹ Department of Computer Science and Engineering, Shaoxing University, Shaoxing 312000, China; ancimoon@gmail.com (K.H.); yejun@usx.edu.cn (J.Y.); zhaoliping_jian@126.com (L.Z.); penghua_47@163.com (H.P.)

² Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China; wei.he@mail.sim.ac.cn

³ College of Information Science and Engineering, Jishou University, Jishou 416000, China

⁴ College of Computer and Information Science, Chongqing Normal University, Chongqing 400047, China

* Correspondence: pijiatian@cqnu.edu.cn

Received: 30 May 2019; Accepted: 20 June 2019; Published: 25 June 2019



Abstract: An online neutrosophic similarity-based objectness tracking with a weighted multiple instance learning algorithm (NeutWMIL) is proposed. Each training sample is extracted surrounding the object location, and the distribution of these samples is symmetric. To provide a more robust weight for each sample in the positive bag, the asymmetry of the importance of the samples is considered. The neutrosophic similarity-based objectness estimation with object properties (super straddling) is applied. The neutrosophic theory is a new branch of philosophy for dealing with incomplete, indeterminate, and inconsistent information. By considering the surrounding information of the object, a single valued neutrosophic set (SVNS)-based segmentation parameter selection method is proposed, to produce a well-built set of superpixels which can better explain the object area at each frame. Then, the intersection and shape-distance criteria are proposed for weighting each superpixel in the SVNS domain, mainly via three membership functions, T (truth), I (indeterminacy), and F (falsity), for each criterion. After filtering out the superpixels with low response, the newly defined neutrosophic weights are utilized for weighting each sample. Furthermore, the objectness estimation information is also applied for estimating and alleviating the problem of tracking drift. Experimental results on challenging benchmark video sequences reveal the superior performance of our algorithm when confronting appearance changes and background clutters.

Keywords: visual tracking; neutrosophic weight; objectness; weighted multiple instance learning

1. Introduction

The task for visual object tracking is to estimate the object location at each frame in a video sequence. Such a kind of visual analysis has been widely studied in computer vision due to its application in many fields, e.g., video surveillance, human–computer interaction, autonomous driving, and traffic monitoring [1,2]. While a lot of effort has been made, and numerous tracking algorithms have been proposed in past decades, it is still a very challenging task due to the factors of illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, rotation, and background clutters, etc. Most of these factors can cause appearance changes, which are very challenging for the visual tracker.

To adjust these challenges adaptively, it is quite important for a robust object tracker to employ an effective appearance model. In general, the existing appearance models can be mainly divided into

two categories, the generative model [3–9] and the discriminative model [10–18]. For the generative model, the tracking is cast as finding the most similar region to the learned object appearance. The tracking result at each frame can be used to update the appearance model. The MeanShift tracker [3] is one of the most influential methods due to its high efficiency and its robustness when confronting the challenges of radical changes and deformation. However, the MeanShift tracker is very sensitive when there is a similar color surrounding the target, because only the histogram feature is applied for building the appearance model. To improve the performance, some other features are introduced into the MeanShift tracking technology, e.g., cross-bin metric [5], background information [6], and depth feature [7]. The IVT tracker [4] employs incremental principal component analysis to represent and update the appearance model. Unlike generative models, both positive and negative samples are utilized by discriminative models, and the object tracking is posed as the task of binary classification to discriminate the target from its surroundings.

Generally, the robustness of generative models is inferior to discriminative models to some extent. Generative models always utilize the foreground information to model the object appearance, and the surrounding information is not considered. Thus, generative models are more likely to drift when handling appearance changes in complex environments. For discriminative models, the positive and negative samples extracted in the first frame are utilized for initialization. For a new arriving frame, the object location is estimated by the afore-trained classifier. The classifier will be updated by using the newly collected positive and negative samples. A boosting method was used for online feature selection in [10]. Only the sample located at the estimated object location is employed as the positive sample. Some negative examples are extracted around the neighborhood of the estimated location. This often causes drift and error accumulation problems when the tracked location drifts a little from the real location. In [11], only the samples extracted from the first frame were labeled for training the classifier, and the samples generated in the subsequent frames were all unlabeled. The SemiBoost tracker [11] may drift in interframe motion on account of the smooth motion assumption.

To tackle the problem of imprecise sampling, online multiple instance learning (MIL) was firstly proposed for object tracking in [12,13]. The MIL tracker uses the positive and negative bags to update the classifier at each frame. The positive bag contains several instances extracted from the close neighborhood of the object location. It has been revealed that the MIL tracker alleviates the drift problem, however, the limitation still exists. Several trackers have been developed within the framework of MIL [14–18]. A more effective and efficient algorithm was proposed, using important prior information, such as instance labels and the most correct positive instance [14], due to the fact that discriminative information about the importance of the positive examples is not considered in the MIL tracker. By the assumption that the tracking result in the current frame is the most positive sample, Zhang et al. [15] employed the Euclidean distance between each positive instance and the estimated object location as the importance of each positive sample. The weight distribution of the samples is centrally symmetric. In [16], the chaotic theory was introduced into MIL-based tracking. The fractal dimension of the dynamic model was adjusted as instance weight. There are still drawbacks for the proposed weighting algorithms. Once the tracked result drifts from the real object location, the assumption of the most positive sample will not be satisfied. Weights will be wrongly utilized to update the classifier. The objectness measure [19] is applied for judging the importance of each instance [17]. The objectness measure [20] owns the ability of judging the probability of a given window containing a whole object. Instead of using the distance measure [15], Yang et al. [17] integrated the objectness estimation into the calculation of the instance importance. Experimental results have demonstrated the power of the objectness measure once it is introduced into the MIL-based tracker. As we know, the robustness of the objectness estimation highly depends on the segmentation result [20,21]. For the application of visual tracking, the environment change occurs almost at each frame. It is essential to propose a scheme to let the segmentation result adaptively adjust different scenes. In addition, some noisy superpixels can distract the objectness-based instance weighting.

Thus, a method for filtering the noisy superpixel should also be seriously considered. However, much uncertain information must be considered when we try to tackle these two serious issues.

Neutrosophic set (NS) [22] is a new branch of philosophy to deal with the origin, nature, and scope of neutralities. It has been widely used in dealing with uncertain information [23]. Due to this, the NS theory has been successfully introduced into many applications, such as medical diagnosis [24], skeleton extraction [25], image segmentation [26–31], and object tracking [7–9,32]. SVNS (single valued neutrosophic set) [33] is a subclass of the neutrosophic set with a finite interval for practical usage. Both the cosine and tangent similarity measures were applied for medical diagnosis in [24]. For the application of image segmentation, the source image is usually transformed into the NS domain, and will be described by the T, I, and F membership set [26,31]. The cosine similarity measure was also utilized in [26,31]. Furthermore, the neutrosophic set-based MeanShift and c-means clustering methods were proposed to earn a more robust segmentation result [27,28,30].

Guided by the above idea, we propose an online visual tracking of weighted multiple instance learning via a neutrosophic similarity-based objectness estimation. First, to produce a well-built set of superpixels at each frame, we propose a method of neutrosophic set-based segmentation parameter selection. The information surrounding object location is taken into consideration. Second, the intersection and shape-distance criteria are utilized for evaluating each superpixel, and three membership functions, T, I, and F, are proposed for each criterion. Then, the neutrosophic set-based multi-criteria similarity score is utilized to facilitate the superpixel filter. The importance of each training instance is evaluated by the estimation of the filtered objectness measure. Third, the information of the objectness estimation is applied for alleviating the problem of tracking drift. Empirical results on challenging video sequences demonstrate our NeutWMIL tracker can robustly track the target. To our own knowledge, this is the first time the NS theory has been introduced into the visual object tracking of a discriminative model.

The paper is organized as follows: in Section 2, we introduce related work. Section 3 introduces our tracking system, where the basic flow is first given, and then the principle of our method and its advantages are illustrated in the following subsections. Section 4 gives the detailed experiment setup and results. Finally, Section 5 concludes the paper.

2. Related Work

Benefitting from the discriminative model, many trackers based on such kind of framework have been proposed [10–18]. Several studies revealed that the method for weighting the training samples plays quite an important role for improving the robustness of a tracker [14–18]. The asymmetry of the importance of the samples is considered, and the objectness measure [19] is applied for judging the importance of each instance [17], which is collected for training the multiple instance learning-based tracker. The results revealed that the performance of the MIL tracker was highly improved [17]. However, the parameters for segmentation were not adaptively updated by considering the surroundings of the tracked object, and all the regions were considered equally for calculating the objectness.

To overcome the above problem, we tried to utilize the NS theory to deal with the related uncertainty problems, due to the ability of the NS theory of handling uncertain information [23], as well as the requirement for enhancing the robustness of the visual trackers. Several works have been done introducing the NS theory into the tracking issue [7–9,32]. In order to fuse both the color and depth features, three membership functions, T, I, and F, were proposed to deal with the uncertainty problem for judging the robustness of each feature [7]. The single valued neutrosophic cross-entropy measure [34] was finally utilized for feature fusion. By considering the drawbacks of the traditional MeanShift tracker [3], Hu et al. [8] integrated the background information into the bin importance of the histogram feature by using the neutrosophic descriptions regarding the uncertain issue of feature enhancement. Furthermore, Hu et al. [9] proposed the element-weighted neutrosophic correlation coefficient and utilized it to improve the CAMShift tracker. It has been revealed that the

proposed trackers are more robust than traditional ones when the neutrosophic theory is utilized [7–9]. Fan et al. [32] proposed a neutrosophic hough transform-based track initiation method to solve the uncertain track initiation problem, and the results demonstrated that it is superior to the traditional hough transform-based track initiation in a complex surveillance environment.

This work is quite different from the proposed NS-base trackers in that it is the first time the NS theory has been introduced into the visual object tracking of discriminative model. Though many methods for image segmentation in NS domain have been proposed, this is also the first work that has proposed to tackle the problem of segmentation parameter selection and superpixel filtering, for the purpose of enhancing the robustness of a tracker.

3. Problem Formulation

3.1. System Overview

The basic flow of the online visual tracking of weighted multiple instance learning via neutrosophic similarity-based objectness estimation is shown in Figure 1. Before the tracking process, the classifier of multiple instance learning is initialized with the same method as the weighted multiple instance learning [15]. Suppose $l_t^* \in R^2$ is the center of the object location at frame t , the object location is represented by a rectangular bounding box. To produce robust objectness [19,20] information for tracking, N segmentation results of frame t are firstly calculated by utilizing N parameter tuples for segmentation. Then we used the SVNS-based segmentation parameter selection method to select the parameter tuple, which can achieve the best segmentation result for the representation of the object area. When the next frame arrives, we calculated its segmentation result by using the selected segmentation parameter tuple. Each region in the segmentation result corresponds to a superpixel. The neutrosophic set-based superpixel filter was employed to filter out the noisy superpixel when measuring the objectness. The sliding window method was used to calculate two confidence maps. The classifier confidence map was calculated based on the afore-trained classifier. The Neut-Objectness confidence map was calculated based on the filtered objectness measure. Finally, each maximum value of the two kinds of confidence maps was utilized to decide $l_{t+1}^* \in R^2$, which is the center of the object location at frame $t+1$. The scale of the rectangular bounding box was fixed in this work. To update the classifier parameters, the filtered objectness measure was applied for weighting the training samples. We first cropped M positive instances within the circular region centering at l_{t+1}^* to form a positive bag $\mathbf{X}^+ = \{x | \|l_{t+1}(x) - l_{t+1}^*\| < \alpha\}$, and the distribution is symmetric, then L negative instances were cropped from an annular region with radius $\alpha < \xi < \beta$ to form a negative bag $\mathbf{X}^- = \{x | \alpha < \|l_{t+1}(x) - l_{t+1}^*\| < \beta\}$, where $l_{t+1}(x)$ is the location of instance x at frame $t+1$. The instances in the positive bag were weighted by the filtered objectness measure, and the negative instances were weighted equally. All the instances were represented by W Haar-like features. Each Harr-like feature corresponded to a weak classifier. All the weighted instances were employed to train the W weak classifiers, and finally K weak classifiers were chosen for constructing a strong classifier \mathbf{H}_k . For each frame, we chose the updated \mathbf{H}_k as the new classifier for tracking. For a new arriving frame, the above procedures were repeated.

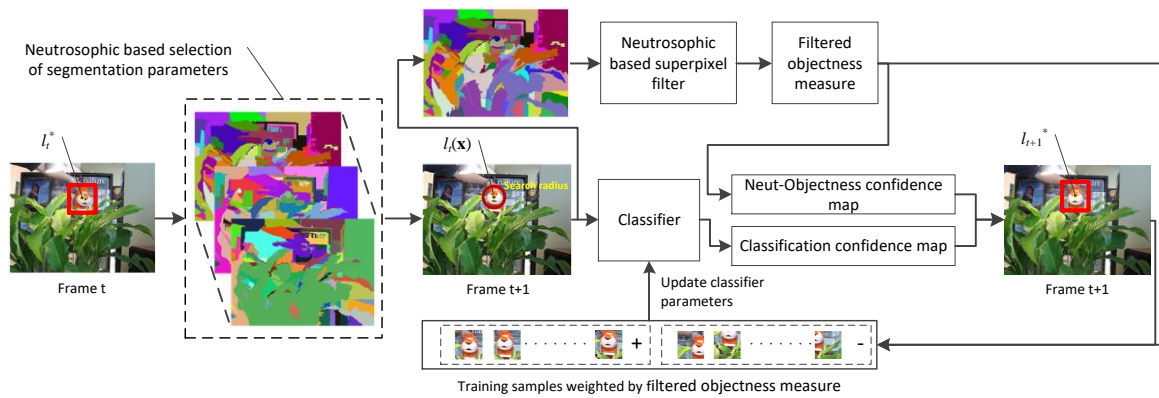


Figure 1. The flow chart of online visual tracking with NeutWMIL.

3.2. Objectness Measure

The objectness measure [20] is always utilized in the research area of object detection for quantifying the degree of an image window containing an object. The color contrast, edge density, and superpixel straddling (SS) methods were proposed in [20], and it has been proven that the SS measure is suitable for object tracking [17]. The SS-based objectness for a window O_w can be calculated by:

$$ss(O_w, T_i) = 1 - \sum_{s \in S(T_i)} \frac{\min(|s \setminus O_w|, |s \cap O_w|)}{|O_w|}, \quad (1)$$

where $S(T_i)$ is the superpixel set obtained by the literature [21], with the segmentation parameter tuple T_i . For each superpixel s , $|s \setminus O_w|$ computes its area outside O_w and $|s \cap O_w|$ calculates its area inside O_w . From Equation (1), we can get determine that a superpixel contributes less when it straddles the window O_w . The superpixels inside the window O_w contribute the most. $ss(O_w, T_i)$ achieves to 1 when there is not any superpixel straddling the window O_w .

As seen in Equation (1), the segmentation parameter tuple T_i is also an important parameter. T_i is a set of parameters that can affect the segmentation result greatly. Different segmentation algorithms always relate to different parameters. For the efficient graph-based image segmentation algorithm [21] employed in this work, each tuple $T_i = \{\sigma, k, m\}$ contained three parameters including σ (used to smooth the input image before segmenting it), k (value for the threshold function), and m (minimum component size enforced by post-processing). As shown in Figure 2, when we use the same segmentation algorithm [21] but employ different segmentation parameter tuples, the segmentation results are quite different. In addition, with the three different segmentation results, the corresponding superpixel sets are also quite different from each other. From Equation (1), we can find that the calculation of SS-based objectness highly depends on the shape and distribution of the superpixels in the image. To apply the objectness measure for weighting the training samples, we then proposed a neutrosophic set-based scheme to handle such a problem.

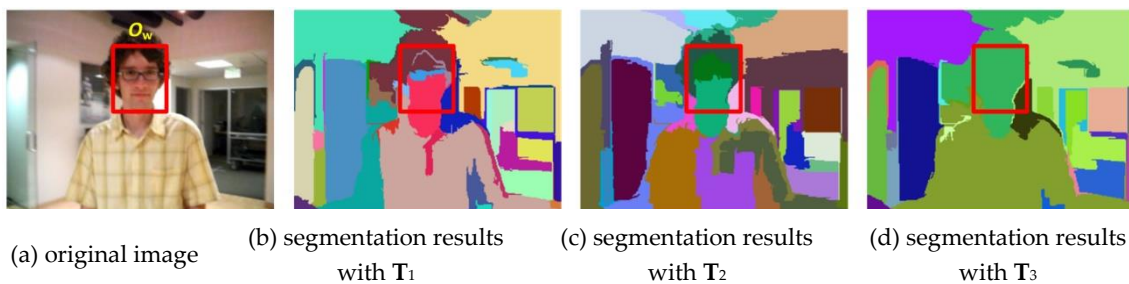


Figure 2. Segmentation results with different segmentation parameter tuples. $T_1 = \{0.4, 450, 150\}$, $T_2 = \{0.5, 250, 400\}$, $T_3 = \{0.6, 550, 250\}$.

3.3. Neutrosophic Set-Based Segmentation Parameter Selection

To apply the objectness information for enhancing the tracker, a suitable segmentation parameter tuple should first be chosen. A well-built superpixel set is one of the most important preconditions for producing reliable objectness measures for the application of visual tracking. A well-built superpixel set should have the ability to enhance the objectness response at the object location. In addition, the symmetric surrounding information should also be considered seriously. The neutrosophic theory has shown its ability to deal with uncertain situations, and the neutrosophic similarity score between SVN_Ss is applied in this work.

The neutrosophic theory was firstly proposed by Smarandache [22]. The original neutrosophic theory is difficult to use for tackling practical problems. SVN_S [33] is a subclass of the neutrosophic set with a finite interval, and it is proposed for practical usage. Let $\mathbf{A} = \{A_1, A_2, \dots, A_m\}$, which denotes a set of alternatives. For a multiple criteria neutrosophic situation, the alternatives A_i can be represented as:

$$A_i = \left\{ \left\langle C_j, T_{C_j}(A_i), I_{C_j}(A_i), F_{C_j}(A_i) \right\rangle \right\}, i = 1, \dots, m; j = 1, \dots, n, \tag{2}$$

where $\mathbf{C} = \{C_1, C_2, \dots, C_n\}$ is a set of criteria, $T_{C_j}(A_i)$ denotes the degree to which the alternative A_i satisfies the criterion C_j , $I_{C_j}(A_i)$ indicates the indeterminacy degree to which the alternative A_i satisfies or does not satisfy the criterion C_j , $F_{C_j}(A_i)$ indicates the degree to which the alternative A_i does not satisfy the criterion C_j , $T_{C_j}(A_i) \in [0, 1]$, $I_{C_j}(A_i) \in [0, 1]$, $F_{C_j}(A_i) \in [0, 1]$.

Suppose $A^* = \left\{ \left\langle C_j, T_{C_j}(A^*), I_{C_j}(A^*), F_{C_j}(A^*) \right\rangle \right\}$ is an ideal alternative with the criteria C_j . The cosine similarity score between A_i and A^* is defined by [35]:

$$S_{\cos}(A_i, A^*) = \sum_{j=1}^n w_j \frac{T_{C_j}(A_i)T_{C_j}(A^*) + I_{C_j}(A_i)I_{C_j}(A^*) + F_{C_j}(A_i)F_{C_j}(A^*)}{\sqrt{T_{C_j}^2(A_i) + I_{C_j}^2(A_i) + F_{C_j}^2(A_i)} \sqrt{T_{C_j}^2(A^*) + I_{C_j}^2(A^*) + F_{C_j}^2(A^*)}}. \tag{3}$$

The corresponding tangent similarity score is defined as [24]:

$$S_{\tan}(A_i, A^*) = \sum_{j=1}^n w_j \tan \left[\frac{\pi}{12} \left(\frac{|T_{C_j}(A_i) - T_{C_j}(A^*)| + |I_{C_j}(A_i) - I_{C_j}(A^*)| + |F_{C_j}(A_i) - F_{C_j}(A^*)|}{|I_{C_j}(A_i) - I_{C_j}(A^*)| + |F_{C_j}(A_i) - F_{C_j}(A^*)|} \right) \right], \tag{4}$$

where w_j is the weight for each criterion, and $w_j \in [0, 1]$, $\sum_j w_j = 1$. Both the cosine and tangent measures have been successfully employed for medical diagnoses [24] and some visual analysis missions [7,26]. In this work, the neutrosophic similarity score was utilized for fusing information, and these two measures were tested separately in the experimental section.

As we want to use the objectness measure to weight the training samples, the weights of the samples close to the object location should be enhanced. Considering such a problem, the object location objectness enhancing criterion was proposed. For the segmentation parameter tuple \mathbf{T}_k , the corresponding membership functions $T_O(\mathbf{T}_k)$ (truth), $I_O(\mathbf{T}_k)$ (indeterminacy), and $F_O(\mathbf{T}_k)$ (falsity) were defined as:

$$T_O(\mathbf{T}_i) = ss(O_{l^*}, \mathbf{T}_i); \tag{5}$$

$$I_O(\mathbf{T}_i) = 1 - \min \left(\frac{1}{C} \frac{\sum_{j=1}^C |ss(O_j(r), \mathbf{T}_i) - ss(O_{l^*}, \mathbf{T}_i)|}{ss(O_{l^*}, \mathbf{T}_i)}, 1 \right); \tag{6}$$

$$F_O(\mathbf{T}_i) = 1 - T_O(\mathbf{T}_i), \tag{7}$$

where O_{l^*} is the rectangular window corresponding to the object location and \mathbf{T}_i is the the i -th segmentation parameter tuple. As shown in Figure 3, $O_j(r)$ is a square window centered at the j -th pixel of the C uniform sampled pixels on the boundary of the square window with the edge length of

$2r+1$ (pixel) centered at l^* , the distribution of $O_j(r)$ is symmetric, l^* is the center of the object location, and $O_j(r)$ has the same size as the objects.

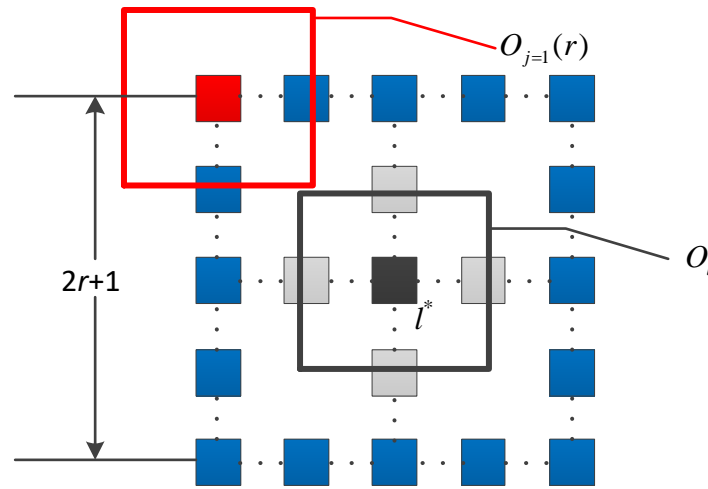


Figure 3. The illustration of $O_j(r)$ mentioned in Equation (6).

Let $A^* = \{(C_O, 1, 0, 0)\}$ denote the ideal alternative with the object location objectness enhancing criterion. Substituting Equations (5)–(7) into Equation (3), we obtain the cosine similarity score between the choice of \mathbf{T}_i and the ideal choice:

$$ltup_{\cos}(\mathbf{T}_i) = \frac{T_O(A_i)}{\sqrt{T_O^2(\mathbf{T}_i) + I_O^2(\mathbf{T}_i) + F_O^2(\mathbf{T}_i)}}. \tag{8}$$

When substituting Equation (5) into Equation (4), we obtain the corresponding tangent similarity score:

$$ltup_{\tan}(\mathbf{T}_i) = \tan\left[\frac{\pi}{12}(|I_O(\mathbf{T}_i)| + 2|F_O(\mathbf{T}_i)|)\right]. \tag{9}$$

Suppose we have N segmentation parameter tuples waiting for selection, the tuple with the maximum similarity score is chosen as the right tuple \mathbf{T}_{sel} . For the cosine and tangent measures, the selections may be different from each other.

3.4. Neutrosophic Set-Based Superpixel Filter

As shown in Equation (1) and Figure 2, all the superpixels were taken into the consideration for the estimation of the objectness. However, such a simple method may bring in some noise for the objectness estimation. For instance, the superpixel far from the object or the superpixel that is too large size may disturb the objectness of the tracked object.

By considering the uncertain information, the intersection and shape–distance criteria were considered in this work. For the intersection criteria, the corresponding true, indeterminate, and faulty membership functions are defined as follows:

$$T_{int}(i) = \max\left(\frac{|s_i \cap O_l|}{|s_i|}, \frac{|s_i \cap O_l|}{|O_l|}\right); \tag{10}$$

$$I_{int}(i) = \frac{|s_i \setminus O_l|}{|s_i|}; \tag{11}$$

$$F_{int}(i) = 1 - T_{int}(i), \tag{12}$$

where O_l is the rectangular window, which corresponds to the object location calculated by the afore-trained classifier before the modification, suppose $\mathbf{S}(\mathbf{T}_{sel})$ is the superpixel set obtained by the literature [21] with the selected segmentation tuple \mathbf{T}_{sel} , and s_i is the i -th superpixel included in $\mathbf{S}(\mathbf{T}_{sel})$.

A superpixel is more likely to satisfy the intersection criteria when it is located in the estimated object location. The corresponding uncertain probability increases with the area outside the object location.

To enhance the robustness of the neutrosophic set-based method for judging a superpixel, the shape and the distance information were considered. Using the shape–distance criteria, we can further give the definitions:

$$T_{sd}(i) = \min\left(f\left(\frac{w_{si}}{w_{ol}}\right), f\left(\frac{h_{si}}{h_{ol}}\right)\right); \quad (13)$$

$$I_{sd}(i) = 1 - e^{-|x_{si}-l|/D}; \quad (14)$$

$$F_{sd}(i) = 1 - T_{sd}(i), \quad (15)$$

where w_{si} and h_{si} are the width and height, respectively, of the tight rectangular bounding box of the super pixel s_i , w_{ol} and h_{ol} are the width and height corresponding to the object window O_l , x_{si} is the centroid location of s_i , l is the center of the O_l , and D is half the length of the O_l diagonal. The function $f(x)$ in $T_{sd}(i)$ is defined as:

$$f(x) = \begin{cases} \frac{1}{2}\text{erfc}(5x - 7) & x \geq 1 \\ \frac{1}{2}\text{erfc}(-2.2x + 0.2) & 0 \leq x < 1 \end{cases} . \quad (16)$$

The domain of $f(x)$ is positive real numbers, and then $f(x)$ is manually designed for the purpose of decreasing the value of $T_{sd}(i)$ when the width or the height of s_i is larger or smaller than the w_{ol} or h_{ol} . As seen in Figure 4, when $x > 1$, the response of $f(x)$ decreases slowly in the intervals of [1,1.2] and [1.6,1.8], but decreases sharply during the interval of [1.2,1.6]. The reason for such a design is that we wanted to keep the information of those superpixels with a relative similar size to the object, and try to discard the superpixels with a much larger width or height than the object. As shown in Figure 4, the response has decreased at the value of less than 0.1 when x equals 1.6. However, we choose a different solution when $x < 1$. The response of $f(x)$ decreases much slower than in the interval of $x > 1$, because a small superpixel may be one of the real parts of the object. We tried to keep the information of such superpixels.

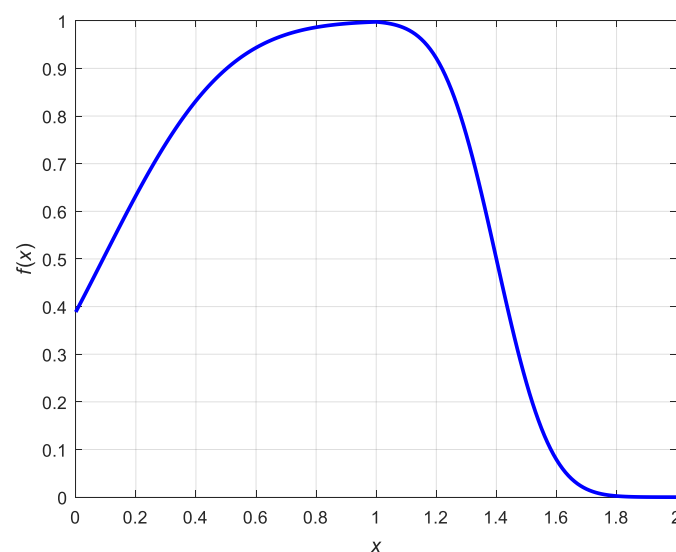


Figure 4. The plot of the function $f(x)$ defined in Equation (16).

As seen in Equation (13) and Equation (14), the shape factor is mainly considered when calculating the truth response, and the distance factor is primarily considered as the indeterminate information.

Let $A^* = \{\langle C_{int}, 1, 0, 0 \rangle, \langle C_{sd}, 1, 0, 0 \rangle\}$ denote the ideal alternative with the intersection and shape–distance criteria. Substituting Equations (10)–(16) into Equation (3), we obtain the cosine similarity score between the choice of s_i and the ideal superpixel:

$$ls_{\cos}(i) = w_{cint} \frac{T_{int}(i)}{\sqrt{T_{int}^2(i) + I_{int}^2(i) + F_{int}^2(i)}} + w_{csd} \frac{T_{sd}(i)}{\sqrt{T_{sd}^2(i) + I_{sd}^2(i) + F_{sd}^2(i)}}, \tag{17}$$

where $w_{cint}, w_{csd} \in [0, 1], w_{cint} + w_{csd} = 1$. When substituting Equation (5) into Equation (4), we obtain the corresponding tangent similarity score:

$$ls_{\tan}(i) = w_{tint} \tan\left[\frac{\pi}{12}(|I_{int}(i)| + 2|F_{int}(i)|)\right] + w_{tsd} \tan\left[\frac{\pi}{12}(|I_{sd}(i)| + 2|F_{sd}(i)|)\right], \tag{18}$$

where $w_{tint}, w_{tsd} \in [0, 1], w_{tint} + w_{tsd} = 1$.

By employing the similarity score, we finally give the definition of the neutrosophic set-based superpixel filter:

$$\mathbf{H}_i = \begin{cases} 1 & ls(i) > \gamma \\ 0 & \text{else} \end{cases}, \tag{19}$$

where \mathbf{H}_i is the filter response for the superpixel s_i , γ is a threshold parameter, and $ls(i)$ is the cosine or tangent similarity score calculated by Equation (17) or Equation (18).

The results of the superpixel filter are applied for estimating the neutrosophic set-based objectness, then the definition of the filtered objectness measure is given by:

$$nss(O_W, \mathbf{T}_{sel}) = 1 - \sum_{s_i \in \mathbf{S}(\mathbf{T}_{sel})} \frac{\min(|s_i \setminus O_W|, |s_i \cap O_W|)}{|O_W|} \mathbf{H}_i, \tag{20}$$

where $\mathbf{S}(\mathbf{T}_{sel})$ is the superpixel set obtained by using the selected segmentation parameter tuple \mathbf{T}_{sel} and O_W is a rectangular window with the same size as the object.

3.5. Object Localization

For the task of visual tracking, the precision of the object location given by the tracker is very important for the following tracking procedure. An inaccurate location may disturb the updating of the classifier, and the tracking result may drift from the real object location in the following frames, even leading to failure. To improve the robustness of the object localization, both the Neut-Objectness confidence map and the classification confidence map were employed in this work.

The classification confidence map is calculated by using the afore-trained classifier \mathbf{H}_k . All the windows whose center is located within the circle area for searching are employed as candidates, suppose sr denotes the searching radius. The scale of each window is the same as the tracked object. The \mathbf{H}_k response of each candidate window finally forms the classification map.

The neutrosophic-based objectness measure is utilized to modify the location l , which fully depends on \mathbf{H}_k . We calculated the Neut-Objectness confidence map by using the similar manner of the calculation of the classification confidence map, but the response of the filtered objectness measure is employed instead of the \mathbf{H}_k response. Let l_{nss} denote the center location of the candidate window with a maximum value within the Neut-Objectness confidence map, and the corresponding response is $nss(O_w(l_{nss}), \mathbf{T}_{sel})$. The fused object location is calculated by

$$l^* = \begin{cases} \lambda l + (1 - \lambda)l_{nss} & \text{if } nss(O_w(l_{nss}), \mathbf{T}_{sel}) > \tau_1, \text{ and } \mathbf{H}_k(l_{nss}) > \tau_2 \\ l & \text{else} \end{cases}, \tag{21}$$

where l denotes the location, corresponding to the center of the candidate window with the maximum value within the classification confidence map, $\mathbf{H}_k(l_{nss})$ is the response of \mathbf{H}_k for the window centered at l_{nss} , λ is the ratio parameter, and τ_1 and τ_2 are threshold parameters for $\lambda, \tau_1, \tau_2 \in [0, 1]$.

As seen in Equation (21), we modify the location calculated by the afore-trained classifier only when the maximum of the Neut-Objectness confidence map and the \mathbf{H}_k response at the corresponding location achieves a relative high value. Such a method can effectively remove the interference, which may be caused by an objectness estimation that is not stable enough.

3.6. Weighted Multiple Instance Learning

In this work, we considered the importance of the instances in the positive bag \mathbf{X}^+ during the learning process. The weight of the j -th positive instance in \mathbf{X}^+ is obtained by using the filtered objectness measure:

$$w_j = \begin{cases} nss(O_{W_j}, \mathbf{T}_{sel}) & \text{if } nss(O_w(l_{nss}), \mathbf{T}_{sel}) > \tau_1 \\ 1 & \text{else} \end{cases}, \quad (22)$$

where O_{W_j} is the window corresponding to the j -th instance in \mathbf{X}^+ . Then, the positive bag probability is computed by [15,17]:

$$p(y = 1 | \mathbf{X}^+) = \sum_{j=1}^M w_j p(y_1 = 1 | x_{1j}), \quad (23)$$

where $p(y_1 = 1 | x_{1j})$ is the posterior probability for the positive sample x_{1j} , and x_{1j} denotes the j -th instance in the positive bag.

Comparing the weight methods employed in WMIL [15] and ONMIL [17], we used the neutrosophic set-based objectness measure to calculate the weight of each positive instance. In the WMIL tracker, the weight is computed mainly based on the Euclidean distance between the center of the instance and the estimated object location. When the tracking result drifts from the real object location, those real positive instances will be assigned as a relative low weight because of the long distance, which is contrary to the fact. For the ONMIL tracker, the traditional objectness estimation with superpixel straddling [20] is directly employed as the weight of each instance. As we have discussed above, the traditional objectness estimation is highly relevant to the segmentation results. When the scene changes during the tracking process, a weak objectness measure may be obtained if an inappropriate set of superpixels is used. In addition, for the task of visual tracking, some superpixels may also disturb the objectness response of the tracked object area. In our method, we first employed an online selection method of segmentation parameter tuples to produce a well-built result of a superpixel set. Secondly, we proposed a neutrosophic set-based superpixel filter to enhance the objectness estimation for the tracking application. Finally, when calculating the weight for each instance, the filtered objectness measurements were utilized for weighting when the corresponding response results were robust enough.

The posterior probability of labeling x_{ij} to be positive is defined as [13]:

$$p(y = 1 | x_{ij}) = \sigma(\mathbf{H}_k(x_{ij})), \quad (24)$$

where $i \in \{0, 1\}$, as has been mentioned above, x_{1j} denotes the j -th instance in the positive bag, x_{0j} denotes the j -th instance in the negative bag, and σ is the sigmoid function, $\sigma(x) = 1 / (1 + e^{-x})$.

The strong classifier \mathbf{H}_k in Equation (24) is defined as:

$$\mathbf{H}_k(x_{ij}) = \ln \left(\frac{p(\mathbf{f}(x_{ij}) | y = 1)p(y = 1)}{p(\mathbf{f}(x_{ij}) | y = 0)p(y = 0)} \right) = \sum_{k=1}^K h_k(x_{ij}), \quad (25)$$

where $f(x_{ij})$ is a set of Haar-like features corresponding to the weak classifier $h_k(x_{ij})$, $f(x_{ij}) = (f_1(x), \dots, f_K(x))^T$. We assume the features in $f(x_{ij})$ are independent and assume uniform prior $p(y = 0) = p(y = 1)$ as MIL tracker [13]. Then, the $h_k(x_{ij})$ is described as [13]:

$$h_k(x_{ij}) = \ln \left(\frac{p(f_k(x_{ij})|y = 1)}{p(f_k(x_{ij})|y = 0)} \right). \quad (26)$$

The conditional distribution in $h_k(\cdot)$ is also defined as a Gaussian function as the MIL tracker, that is:

$$\begin{aligned} p(f_k(x_{ij})|y = 1) &\sim N(\mu_1, \sigma_1) \\ p(f_k(x_{ij})|y = 0) &\sim N(\mu_0, \sigma_0) \end{aligned} \quad (27)$$

Like the WMIL tracker [15], the parameters μ_1 and σ_1 are updated as follows:

$$\begin{aligned} \mu_1 &\leftarrow \eta\mu_1 + (1 - \eta)\bar{\mu} \\ \sigma_1 &\leftarrow \sqrt{\eta(\sigma_1)^2 + (1 - \eta)\frac{1}{M} \sum_{j=0|y_i=1}^{M-1} (f_k(x_{ij}) - \bar{\mu})^2 + \eta(1 - \eta)(\mu_1 - \bar{\mu})^2} \end{aligned} \quad (28)$$

where η is the learning rate, M is the number of positive samples, and $\bar{\mu} = \frac{1}{M} \sum_{j=0|y_i=1}^{M-1} f_k(x_{ij})$ is the average of the k -th feature extracted from the M positive samples. The parameters μ_0 and σ_0 are updated by employing the same rules.

Similar to the WMIL tracker, the bag log-likelihood function is defined as [15]:

$$\uparrow(\mathbf{H}) = \sum_{i=0}^1 \left(y_i \log \left(\sum_{j=0}^{M-1} w_j p(y = 1|x_{1j}) \right) + (1 - y_i) \log \left(\sum_{j=M}^{M+L-1} (1 - p(y = 1|x_{0j})) \right) \right), \quad (29)$$

where L is the number of negative samples, w_j is the weight of the j -th positive instance defined in Equation (22), y_i is the label of the training bag, y_i equals to 1 when the bag is positive, and y_i is set as 0 when the bag is negative.

As the method utilized in MIL [13], WMIL [15], and ONMIL trackers [17], our tracker maintains W weak classifiers in the pool $\Phi = \{h_1, h_2, \dots, h_W\}$. At each frame, the K weak classifier with strong classification ability is selected to form \mathbf{H}_k . In the WMIL tracker, a more efficient criterion was proposed. Similar rules are employed here. The scheme for selecting K weak classifier is given below [15].

$$h_k = \operatorname{argmax}_{h \in \Phi} \langle h, \nabla \uparrow(\mathbf{H}) \rangle_{\mathbf{H} = \mathbf{H}_{k-1}}, \quad (30)$$

where [15]:

$$\nabla \uparrow(\mathbf{H})(x_{ij}) = y_i \frac{w_j \sigma(\mathbf{H}(x_{ij}))(1 - \sigma(\mathbf{H}(x_{ij})))}{\sum_{m=0}^{M-1} w_m \sigma(\mathbf{H}(x_{im}))} - (1 - y_i) \frac{\sigma(\mathbf{H}(x_{ij}))(1 - \sigma(\mathbf{H}(x_{ij})))}{\sum_{m=M}^{M+L-1} (1 - \sigma(\mathbf{H}(x_{im})))}, \quad (31)$$

where w_j and w_m are the weights of the corresponding samples, and they are calculated by Equation (22).

Finally, the main steps of the proposed NeutWMIL tracker are shown in Algorithm 1.

Algorithm 1 Online neutrosophic similarity-based objectness tracking with weighted multiple instance learning algorithm (NeutWMIL)

Initialization:

- (1) Initialize the region of the tracked object in the first frame.
- (2) Initialize the MIL-based classifier \mathbf{H}_k by employing the training bags surrounding the object location.

Online tracking:

- (1) Select suitable segmentation parameter tuple \mathbf{T}_{sel} by utilizing the method of neutrosophic set-based segmentation parameter selection.
 - (2) Read a new frame of the video sequence.
 - (3) Calculate the superpixel set for the current frame with the selected tuple \mathbf{T}_{sel} .
 - (4) Compute the estimation of the filtered objectness surrounding the location obtained by the afore-trained classifier \mathbf{H}_k , and get the final object location l^* in this frame by Equation (21).
 - (5) Compute the neutrosophic set-based weight by Equation (22).
 - (6) Crop M positive instances within the circular region centering at l^* to form a positive bag, then crop L negative instances from an annular region to form a negative bag.
 - (7) Update the parameters of W weak classifiers by Equation (28).
 - (8) Select K weak classifiers by Equation (30) and form the current classifier \mathbf{H}_k .
 - (9) Go to step (1) until the end.
-

4. Experiments

In this section, we compared the proposed NeutWMIL tracker with six trackers on 20 challenging video sequences. The six trackers were the NeutanWMIL, ONWMIL, WMIL [15], MIL [13], OAB [10], and SemiB[11] trackers. Specifically, the NeutWMIL and the NeutanWMIL trackers were two kinds of the proposed tracking algorithm. The only difference is that the NeutWMIL tracker employed the cosine similarity measure, and the NeutanWMIL tracker used the tangent measure when the neutrosophic similarity estimation was needed during the tracking process. For the ONWMIL tracker, we implemented it using a similar instance weighting method to that proposed in [17]. The objectness estimation was directly applied for weighting each instance for the ONWMIL tracker, and the segmentation parameters were kept constant. The instance-weighting scheme was the only difference when compared with the NeutWMIL and the NeutanWMIL trackers. The source codes of the WMIL, MIL, OAB, and SemiB trackers are all publicly available, and the default parameter settings were utilized.

4.1. Parameter Setting

For the NeutWMIL, NeutanWMIL, and ONWMIL trackers, all the parameters related to the WMIL tracker were set as they are mentioned in the publicly available source codes. For instance, we set $\alpha = 4$, the search radius sr was set to 25, and β was set to $1.5sr$, the number of the cropped negative instances L was set to 50, the number of the weak classifiers $W = 150$, and the strong classifier \mathbf{H}_k maintained $K = 15$ weak classifiers. Three segmentation parameter tuples were chosen— $\mathbf{T}_1 = \{0.4, 450, 150\}$, $\mathbf{T}_2 = \{0.5, 500, 200\}$, and $\mathbf{T}_3 = \{0.6, 550, 250\}$. When we performed the tuple selection algorithm, the parameter r defined in Equation (6) was set to 8, and C was set to 4, which means the four corners of the square window with an edge length of 17 pixels were considered for evaluating the indeterminate estimation. For the neutrosophic set-based superpixel filter, when calculating the similarity score of each superpixel by Equations (17)–(18), we set $w_{cint} = w_{csd} = w_{tint} = w_{tsd} = 0.5$, which meant each criterion was treated equally. The threshold parameter γ in Equation (19) decided how many superpixels could pass the filter. To fully use the superpixel information and to filter the noisy superpixel in the meantime, a near median value 0.4 was set to γ . For the object location step, the location estimated by the trained classifier was more robust than the filtered objectness-based result statistically. However, when a relatively high response of the filtered objectness was received, as well as the response of the classifier, the objectness results were usually robust enough, and a more accurate result could be

achieved by fusing these two kinds of results. By considering this, for the parameters in Equation (21), we set $\tau_1 = 0.3$ and $\tau_2 = 0.3$, and then the fusing ratio λ is set to 0.3. Finally, all parameters were kept constant for all the experiments.

4.2. Evaluation Criteria

We used two kinds of evaluation criteria, one was center location error, and the other was the success ratio based on the overlap metric. For the center location error metric, the Euclidean distance between the estimated object location and the manually labeled ground truth was considered. The overlap score is defined as:

$$p_i = \frac{|ROI'_i \cap ROI_i|}{|ROI'_i \cup ROI_i|} \quad (32)$$

where ROI'_i is the object area estimated by the tracker in the i -th frame, ROI_i corresponds to the ground truth. Giving a threshold u , we can say the result is correct if $p_i > u$. Suppose FNs is the number of the frames, then the success ratio is calculated by:

$$R = \sum_{i=1}^{FNs} q_i / FN_s, \quad (33)$$

where:

$$q_i = \begin{cases} 1 & \text{if } p_i > u \\ 0 & \text{else} \end{cases} \quad (34)$$

4.3. Quantitative Analysis

Details of each tested video sequence are given in Table 1. 14,163 frames are evaluated here. The challenge degree grew with the increase in the NC value. The average center location errors for the tested trackers are shown in Table 2. We can see that the NeutWMIL tracker ran the best in 13 sequences, and it performed the second best in four sequences. The NeutanWMIL tracker performed the best in four sequences and the second best in six sequences. The results of the corresponding average overlap ratio are given in Table 3. We can see the NeutWMIL tracker had the best performance in 12 sequences and the second best in six sequences. The NeutanWMIL tracker performed the best in five sequences and the second best in eight sequences. Figure 5 shows the plot of average success of the evaluated trackers through all the sequence. We can see that the NeutWMIL tracker performed the best, and the NeutanWMIL tracker had the second best performance. We can also find that there is a big gap between the NeutWMIL tracker and the WMIL tracker, as well as the MIL tracker. Based on the last line of both Tables 2 and 3, a conclusion can also be drawn that the proposed NeutWMIL tracker performs the best, and the proposed NeutanWMIL tracker performs the second best.

Table 1. An overview of the 20 tested video sequences. (Total number of evaluated frames is 14,163)

Sequence	IV	SV	OCC	DEF	MB	FM	IPR	OPR	OV	BC	LR	FNs	NC
Freeman1		Y					Y	Y				326	3
Mountain-Bike							Y	Y		Y		228	3
Vase		Y				Y	Y					271	3
Sylvester	Y						Y	Y				1345	3
Rubik		Y	Y				Y	Y				1997	4
Gym		Y		Y			Y	Y				767	4
Football			Y				Y	Y		Y		362	4
Boy		Y			Y	Y	Y	Y				602	5
Couple		Y		Y		Y		Y		Y		140	5
BlurBody		Y		Y	Y	Y	Y					334	5
Basketball	Y	Y		Y				Y		Y		725	5
Doll	Y	Y	Y				Y	Y				3872	5
FleetFace		Y		Y	Y	Y	Y	Y				707	6
Coke	Y		Y			Y	Y	Y		Y		291	6
David	Y	Y	Y	Y	Y		Y	Y				471	7
ClifBar		Y	Y		Y	Y	Y		Y	Y		472	7
Tiger1	Y		Y	Y	Y	Y	Y	Y				354	7
Biker		Y	Y	Y	Y	Y	Y	Y	Y		Y	142	7
Tiger2	Y		Y	Y	Y	Y	Y	Y	Y			365	8
Soccer	Y	Y	Y		Y	Y	Y	Y		Y		392	8

Note: IV: Illumination Variation, SV: Scale Variation, OCC: Occlusion, DEF: Deformation, MB: Motion Blur, FM: Fast Motion, IPR: In-Plane Rotation, OPR: Out-of-Plane Rotation, OV: Out-of-View, BC: Background Clutters, and LR: Low Resolution, FNs: Frames, NC: Number of types of challenge.

Table 2. The average center location errors (in pixels) for the compared trackers (bold red fonts indicate the best performance, while the italic blue fonts indicate the second best ones).

Sequence	Neut-WMIL	Neutan-WMIL	ON-WMIL	WMIL	MIL	OAB	SemiB
Freeman1	14.30	16.70	17.80	<i>15.64</i>	17.06	66.12	54.69
MountainBike	7.89	15.02	29.80	120.05	<i>8.07</i>	12.96	44.39
Vase	21.97	21.53	22.62	<i>21.08</i>	15.04	34.58	32.05
Sylvester	7.75	<i>8.79</i>	8.96	18.37	17.49	12.18	22.75
Rubik	14.49	41.30	80.97	84.44	<i>22.56</i>	33.74	53.82
Gym	11.47	29.60	62.04	123.95	20.71	<i>15.24</i>	23.60
Football	12.86	<i>12.54</i>	16.79	14.38	11.66	171.91	96.91
Boy	7.54	<i>7.38</i>	19.63	7.88	108.24	3.43	56.03
Couple	9.70	7.88	35.68	35.92	34.80	33.86	102.71
BlurBody	33.91	36.44	<i>35.07</i>	85.45	81.99	59.44	108.71
Basketball	<i>10.76</i>	10.22	17.69	25.65	107.05	145.94	158.50
Doll	28.18	82.91	<i>47.29</i>	74.80	70.37	127.33	52.73
FleetFace	29.66	50.11	69.94	109.24	50.68	<i>44.27</i>	69.49
Coke	<i>23.54</i>	32.47	43.24	46.60	113.62	17.64	50.93
David	18.97	38.47	46.67	20.22	24.34	71.55	55.41
ClifBar	17.85	9.65	<i>10.50</i>	20.08	23.47	32.87	74.98
Tiger1	<i>14.52</i>	13.94	25.10	73.96	84.24	42.01	60.78
Biker	10.30	<i>11.51</i>	36.35	20.15	27.54	92.87	93.26
Tiger2	17.28	<i>19.19</i>	50.53	40.29	<i>21.93</i>	58.31	68.10
Soccer	28.35	88.16	57.80	101.44	51.88	99.71	92.07
average	17.06	<i>27.69</i>	36.72	52.98	45.64	58.80	68.59

Table 3. The average overlap ratio for the compared trackers (bold red fonts indicate the best performance, while the italic blue fonts indicate the second best ones).

Sequence	Neut-WMIL	Neutan-WMIL	ON-WMIL	WMIL	MIL	OAB	SemiB
Freeman1	0.299	<i>0.281</i>	0.243	<i>0.281</i>	0.260	0.201	0.170
MountainBike	0.705	0.607	0.554	0.380	0.701	0.621	0.258
Vase	0.314	<i>0.312</i>	0.308	0.307	0.312	0.275	0.236
Sylvester	0.696	<i>0.675</i>	0.657	0.547	0.514	0.612	0.478
Rubik	0.553	0.457	0.191	0.236	<i>0.490</i>	0.385	0.288
Gym	0.474	0.366	0.172	0.074	0.399	0.438	0.269
Football	0.574	<i>0.583</i>	0.412	0.508	0.604	0.237	0.149
Boy	0.662	<i>0.670</i>	0.375	0.591	0.296	0.780	0.272
Couple	<i>0.594</i>	0.617	0.462	0.456	0.486	0.279	0.078
BlurBody	0.525	<i>0.496</i>	0.506	0.298	0.281	0.375	0.193
Basketball	<i>0.655</i>	0.667	0.502	0.529	0.213	0.037	0.048
Doll	0.356	0.164	<i>0.275</i>	0.141	0.231	0.051	0.270
FleetFace	<i>0.568</i>	0.552	0.459	0.305	0.579	0.560	0.415
Coke	<i>0.521</i>	0.448	0.277	0.234	0.047	0.551	0.176
David	0.407	0.309	0.234	<i>0.395</i>	0.346	0.212	0.238
ClifBar	0.426	0.508	<i>0.467</i>	0.403	0.373	0.264	0.224
Tiger1	<i>0.657</i>	0.671	0.517	0.128	0.168	0.526	0.303
Biker	<i>0.437</i>	0.445	0.246	0.246	0.259	0.241	0.244
Tiger2	0.610	<i>0.571</i>	0.309	0.292	0.505	0.250	0.202
Soccer	0.321	0.101	0.204	0.145	<i>0.221</i>	0.103	0.101
average	0.52	<i>0.47</i>	0.37	0.32	0.36	0.35	0.23

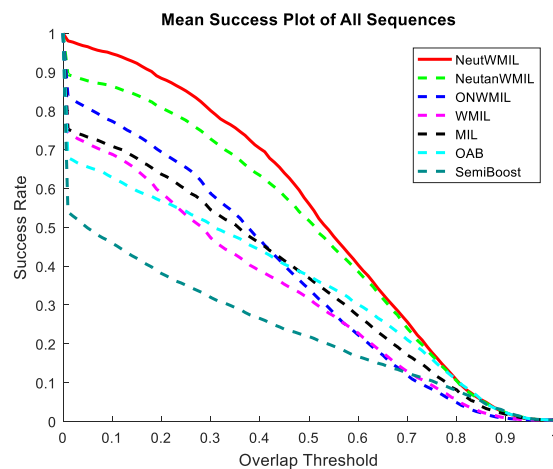


Figure 5. The plot of average success for all the tested sequences.

4.4. Qualitative Analysis

We chose thirteen representative sequences among the tested sequences for the Qualitative analysis. As seen in Tables 1–3, the names of the selected 13 sequences are shown in bold font in each table. Based on the challenge degree of these sequences, they were separated into three groups for analysis. The success and center position error plots for each sequence can be seen in Figures 6–8. Some sampled tracking results are shown in Figures 9–11.

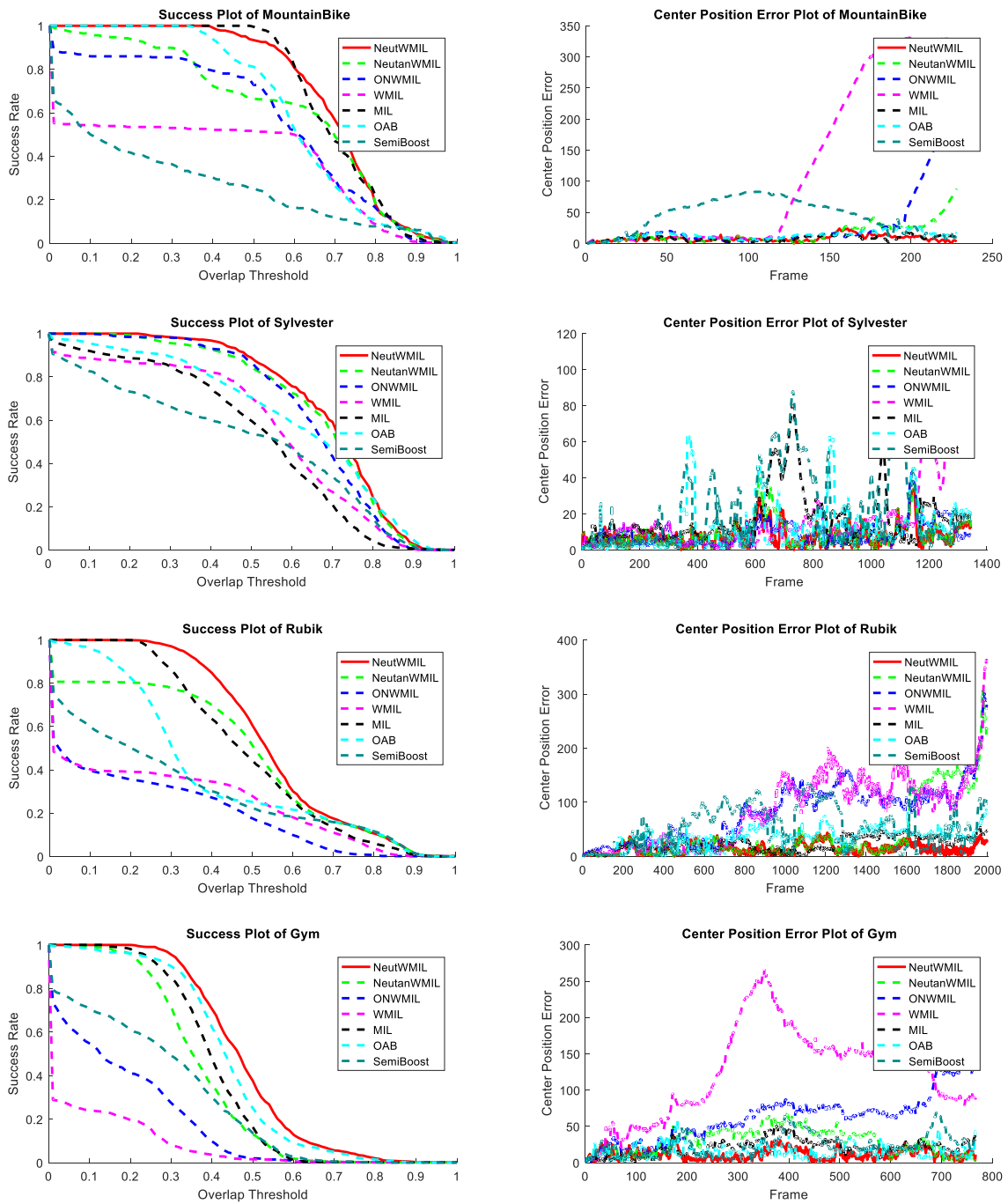


Figure 6. Success and center position error plots of tested sequences *MountainBike*, *Sylvester*, *Rubik*, and *Gym*.

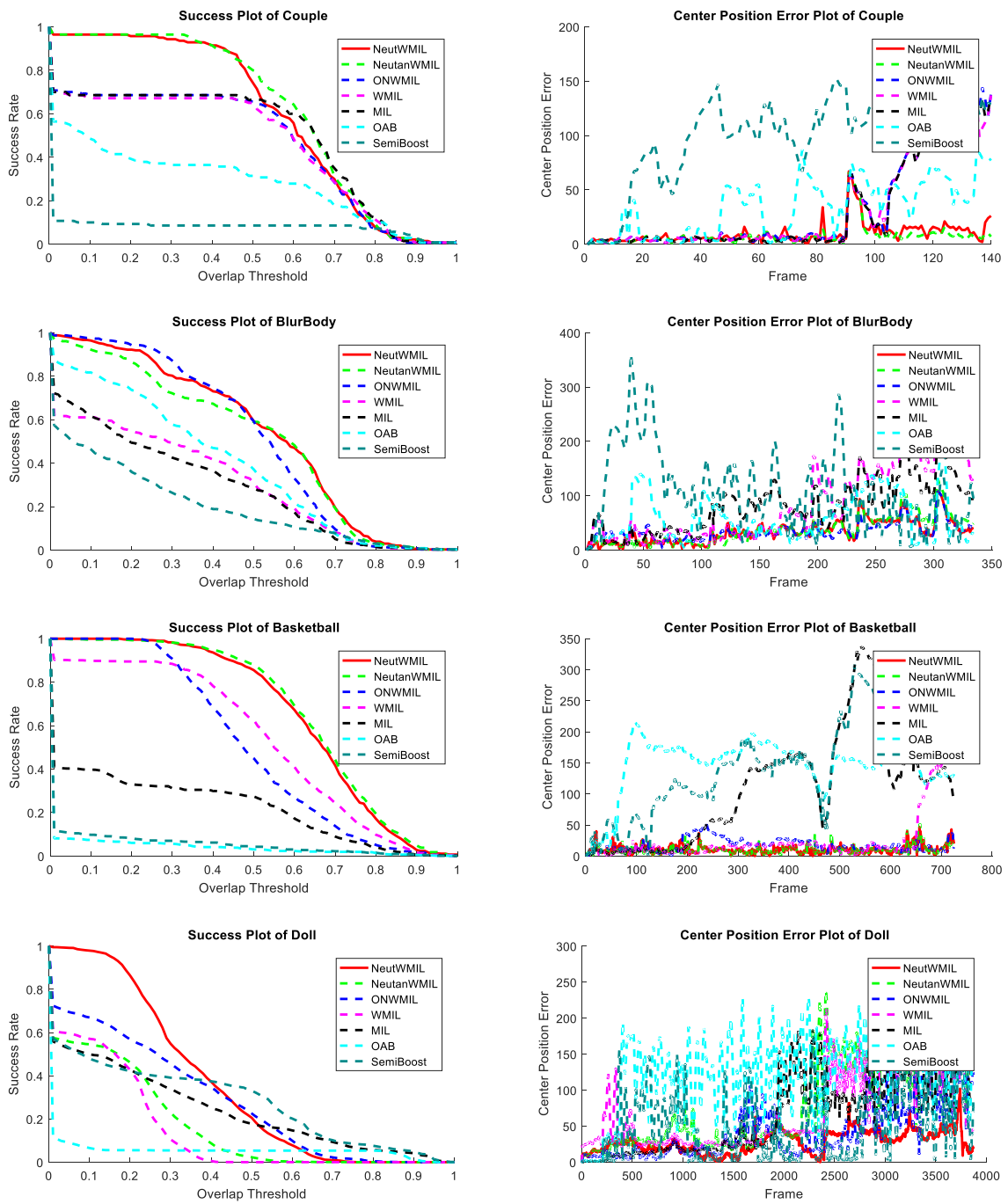


Figure 7. Success and center position error plots of tested sequences *Couple*, *BlurBody*, *Basketball*, and *Doll*.

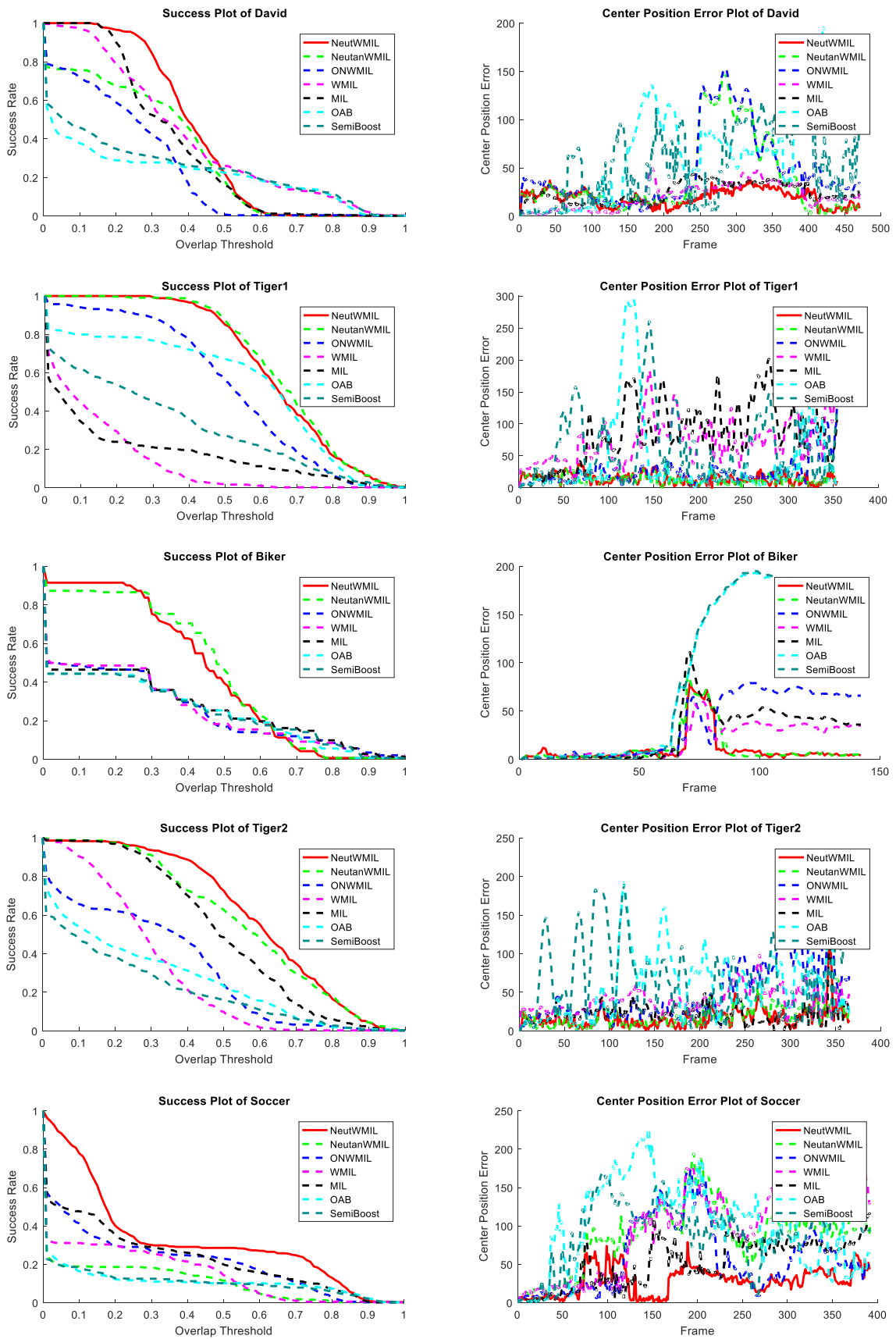


Figure 8. Success and center position error plots of tested sequences *David*, *Tiger1*, *Biker*, *Tiger2*, and *Soccer*.

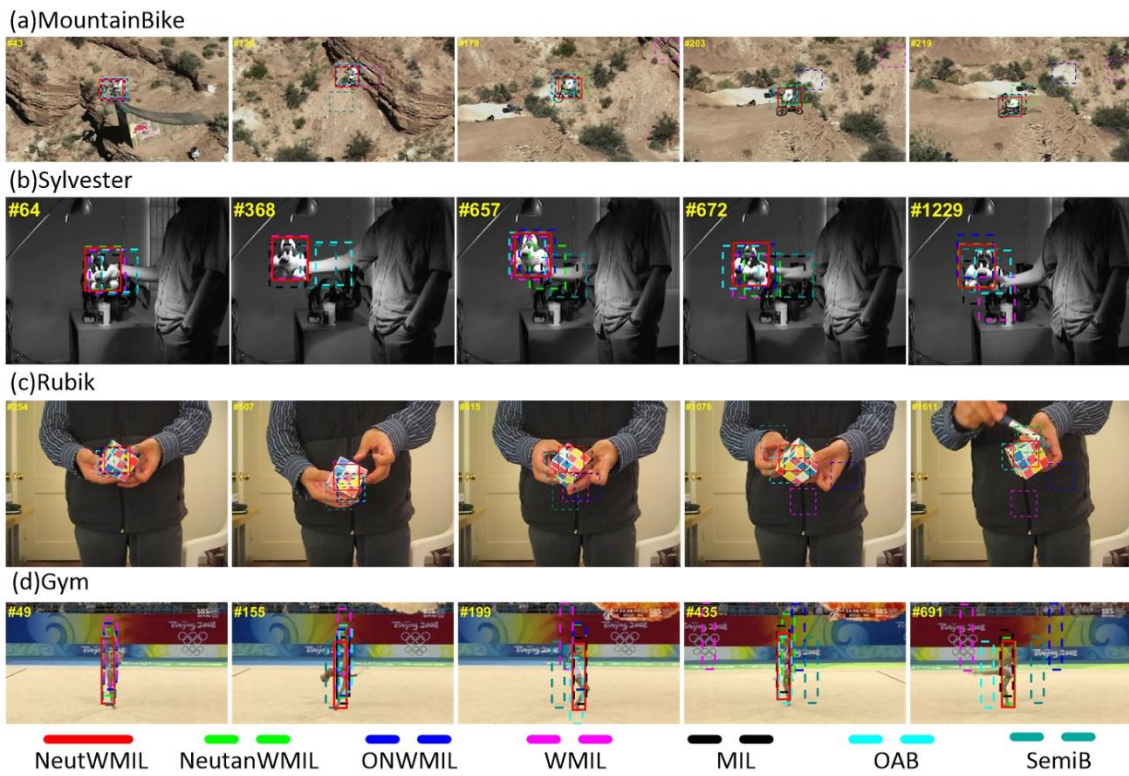


Figure 9. Sampled tracking results for tested sequences *MountainBike* (a), *Sylvester* (b), *Rubik* (c), and *Gym* (d).



Figure 10. Sampled tracking results for tested sequences *Couple* (a), *BlurBody* (b), *Basketball* (c), and *Doll* (d).



Figure 11. Sampled tracking results for tested sequences *David* (a), *Tiger1* (b), *Biker* (c), *Tiger2* (d), and *Soccer* (e).

4.4.1. MountainBike, Sylvester, Rubik, and Gym

As shown in Table 1, the first two sequences were with three types of challenges. The main challenges were in-plane and out-plane rotation, and there was a background clutter challenge in the *MountainBike* sequence, and illumination variation challenge in the *Sylvester* sequence. As seen in Figure 9a, the SemiB tracker yielded a severe drift problem at frames #43, #126, and #179. The WMIL tracker failed due to the background clutter, as shown by frames #126 and #179. When the biker passed by the area with more challenging background clutter, the NeutanWMIL and ONWMIL trackers produced wrong object locations, as seen in frame #219. The OAB and SemiB trackers drifted away on account of the in-plane and out-plane challenge, as shown by frames #64 and #368 in Figure 9b. Due to the rotation and illumination variation challenges, the trackers drifted from the object except the NeutWMIL tracker, as shown by frames #657 and #672. The center position errors of all the trackers at each frame are shown in Figure 6. We can see the NeutWMIL tracker outperformed all the others, the results in Tables 2 and 3 also support this conclusion. For the last two sequences, besides the scale variation, in-plane and out-plane rotation challenges, the occlusion was another challenging issue for the *Rubik* sequence. For the *Gym* sequence, there was also a deformation challenge. All the trackers drifted away due to the in-plane and out-plane rotation, as shown by frames #254, #507, #815, and #1078 in Figure 9c, except the NeutWMIL and NeutanWMIL trackers. As shown in Figure 6, the plot of center position error shows that NeutanWMIL tracker performed well until frame #1600, and the drift was mainly caused by the occlusion, as shown by frame #1611. Considering the deformation challenge,

all the trackers except the NeutWMIL tracker were affected in some degree in the *Gym* sequence. Drifts can be seen in Figure 9d, the plot in Figure 6 also reveals the drift problem.

4.4.2. Couple, BlurBody, Basketball, and Doll

This group of sequences was more challenging than the previous one. The fast motion of the camera and background clutter were the most challenging problems in the *Couple* sequence. As shown in Figure 7, Table 2, and Table 3, the NeutanWMIL tracker performed the best in the *Couple* sequence. The OAB and SemiB trackers could not adaptively adjust to these challenges, and they drifted from the 'couple', as shown in Figure 10a. As seen in Figure 7, we can see that all the trackers drifted away severely near frame #90, because a very fast motion occurred to the camera. However, as shown in Figure 10a, the 'couple' was re-tracked by the NeutWMIL and NeutanWMIL trackers. For the *BlurBody* sequence, besides the fast motion, the most challenging problems were motion blur and scale variation. As shown in Figure 7, the SemiB tracker drifted away at the very beginning. The related screenshots can be seen by frames #12 and #16 in Figure 10b. As shown by frames #124 and #203, the WMIL, MIL, and OAB trackers drifted for the scale variation and motion blur challenges. From the plots shown in Figure 7, we can find that the NeutWMIL and NeutanWMIL trackers performed the best, the ONWMIL tracker also performed well to some extent. There was background clutter, out-plane rotation, deformation, and scale variation challenges in the *Basketball* sequence. The OAB and SemiB trackers failed quickly when a player appeared nearby the target, as shown by frame #39 in Figure 10c. The MIL tracker also drifted from the target player after he passed by several other players, as shown by frames #258 and #468. The WMIL tracker ran well until the player passed by several other players with similar color, as shown by frame #664. As shown in Figure 7, Table 2, and Table 3, the NeutanWMIL tracker ran the best for both the *BlurBody* and *Basketball* sequences, but the gap between the NeutanWMIL tracker and the NeutWMIL tracker is very small. Challenges like illumination variation, scale variation, occlusion, and rotation were included in the *Doll* sequence. The WMIL, OAB, and SemiB trackers drifted away quickly, as shown by frames #185, #225, and #742. The NeutanWMIL and ONWMIL trackers could not adaptively adjust to these challenges and drifted from the doll, as shown by frames #1414 and #2324. The proposed NeutWMIL tracker can yield more stable and more accurate results than the other six trackers.

4.4.3. David, Tiger1, Biker, Tiger2, and Soccer

This group had the most challenging sequences. Seven types of challenge were included in the *David* sequence. Details can be seen in Table 1. The OAB tracker was distracted by the wall because of the similar color, the SemiB tracker drifted from the target mainly due to the challenge of illumination variation, as shown by frames #117 and #155 in Figure 11a. The NeutanWMIL and ONWMIL trackers were also distracted by the wall because of the illumination change and the scale variation, as well as the similar texture between the painting and the target. As seen in Figure 7, Figure 11a, Table 2, and Table 3, the NeutWMIL tracker produced a more robust estimation of object location. For the sequences of *Tiger1* and *Tiger2*, seven types of challenge were included in the *Tiger1* sequence, and *Tiger2* consisted of eight types of challenge. The SemiB, OAB, WMIL, MIL, and ONWMIL trackers could not adjust to these challenges well, as shown by frames #21, #95, #128, #143, and #235 in Figure 11b, as well as frames #64, #99, #280, and #353 shown in Figure 11d. The performance gap between the NeutanWMIL tracker and the NeutWMIL tracker was very small in the sequences of *Tiger1* and *Tiger2*, as shown in Figure 7, Table 2, and Table 3. The *Biker* sequence mainly contained challenges like fast motion, scale variation, out-plane rotation, out of view, and low resolution. The OAB and SemiB trackers drifted from the target mainly due to the challenges of scale variation and head rotation, as shown by frames #64 and #66 in Figure 11c. All the trackers failed nearby frame #70, because of a very fast move from the target, as shown by frame #72 in Figure 11c. However, the NeutanWMIL and NeutWMIL trackers snapped to the target again when the target was located within the searching area, as shown by frames #84 and #136. The *Soccer* sequence had the most challenging problems among the tested

sequences. As shown by frame #37 in Figure 11e, the OAB and SemiB trackers drifted away because of the challenges of motion blur and rotation. As seen in Figure 8, we can find the estimation results of the NeutWMIL and NeutanWMIL trackers were not stable between the frame #70 and #120. This is mainly due to the severe occlusion during such a period. The ONWMIL, WMIL, and MIL trackers drifted away after such a long-term occlusion. When the target appeared, the NeutWMIL snapped to the target again, and performed the best in the following frames, as shown by frames #127 and #155.

4.5. Discussion

With the above analysis, we can find that the proposed NeutWMIL tracker had the best performance. This was mainly due to three contributions of this work. First, by employing the proposed object location objectness enhancing criterion, a more appropriate tuple of segmentation parameter was selected, and then a more suitable superpixel set for measuring the objectness was produced, which helped the tracker to adjust for different scenes during the tracking process. Secondly, the intersection and shape–distance criteria were proposed for constructing the superpixel filter, which helped the tracker filter out those superpixels which may disturb the tracker, and more reliable sample weights were produced for updating the classifier. Thirdly, the object location was finally decided by fusing the information of the Neut-Objectness confidence map and the classification confidence map, which could also enhance the robustness of the tracker.

As shown in Figure 5, Table 2, and Table 3, we can see that there is a big gap between the proposed NeutWMIL and the WMIL tracker. The main reason for this is the NeutWMIL tracker utilized the neutrosophic set-based objectness weighting algorithm. Weights that are more robust can be produced when there is a small drift from the real object location. The comparison result also shows that the NeutWMIL tracker performed better than the NeutanWMIL tracker. As we have mentioned, the only difference between these two trackers is the usage of different neutrosophic similarity measures. The cosine similarity measure is applied for the NeutWMIL tracker, and the tangent similarity measure is employed by the NeutanWMIL tracker. Regarding Equations (8) and (9) and Equations (17) and (18), it can be found that the truth element is enhanced for the cosine measure, and the I and F elements are only applied for decreasing the similarity response. Such a scheme seems more suitable for the judging application in this work. However, all the T, I, and F elements are treated equally for the tangent measure. As shown in Figure 5, Table 2, and Table 3, a relatively big gap exists between the NeutWMIL tracker and the ONWMIL tracker. This result reveals that the proposed weighting method contributed much more to the tracker, rather than using the objectness estimation directly.

5. Conclusions

In this paper, we presented a novel algorithm of neutrosophic similarity-based objectness tracking via weighted multiple instance learning. This is the first time the NS theory has been introduced into the visual object tracking of a discriminative model. To produce more reliable sample weights, object location objectness enhancing criterion was first proposed for the selection of segmentation parameter tuples. Then, the intersection and the shape–distance criteria were proposed for filtering out the unreliable superpixels. Three membership functions, T, I, and F, for each criterion were given by considering uncertain issues. Finally, we proposed a method of filtered objectness measure and also utilized such a neutrosophic set-based measure to modify the location calculated by the afore-trained classifier. Experimental results on challenging video sequences demonstrated the superiority of the proposed tracker to state-of-the-art discriminative trackers in accuracy and stability. Moreover, our parameter selection technique can be extended to any other segmentation algorithm if its performance can be affected by some parameters. Our sample weighting scheme can be also extended to other discriminative trackers to easily help them update the classifier using robust sample weights.

Author Contributions: Conceptualization, investigation and writing, K.H.; review and editing, W.H., J.Y., L.Z., H.P., and J.P.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61603258, Grant 61703280, and Grant 61662025, and in part by the Natural Science Foundation of Zhejiang Province under Grant LY19F020015.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, 13. [[CrossRef](#)]
2. Yang, H.; Shao, L.; Zheng, F.; Wang, L.; Song, Z. Recent advances and trends in visual tracking: A review. *Neurocomputing* **2011**, *74*, 3823–3831. [[CrossRef](#)]
3. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577. [[CrossRef](#)]
4. Ross, D.; Lim, J.; Lin, R.-S.; Yang, M.-H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [[CrossRef](#)]
5. Leichter, I. Mean shift trackers with cross-bin metrics. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 695–706. [[CrossRef](#)] [[PubMed](#)]
6. Vojir, T.; Noskova, J.; Matas, J. Robust scale-adaptive mean-shift for tracking. *Pattern Recogn. Lett.* **2014**, *49*, 250–258. [[CrossRef](#)]
7. Hu, K.; Ye, J.; Fan, E.; Shen, S.; Huang, L.; Pi, J. A novel object tracking algorithm by fusing color and depth information based on single valued neutrosophic cross-entropy. *J. Intell. Fuzzy Syst.* **2017**, *32*, 1775–1786. [[CrossRef](#)]
8. Hu, K.; Fan, E.; Ye, J.; Fan, C.; Shen, S.; Gu, Y. Neutrosophic similarity score based weighted histogram for robust mean-shift tracking. *Information* **2017**, *8*, 122. [[CrossRef](#)]
9. Hu, K.; Fan, E.; Ye, J.; Pi, J.; Zhao, L.; Shen, S. Element-weighted neutrosophic correlation coefficient and its application in improving camshift tracker in rgb-d video. *Information* **2018**, *9*, 126. [[CrossRef](#)]
10. Grabner, H.; Grabner, M.; Bischof, H. *Real-Time Tracking Via On-Line Boosting*; Chantler, M., Fisher, B., Trucco, M., Eds.; BMVA Press: Graz, Austria, 2006; pp. 6.1–6.10.
11. Grabner, H.; Leistner, C.; Bischof, H. *Semi-Supervised On-Line Boosting for Robust Tracking*; European Conference on Computer Vision (ECCV); Forsyth, D., Torr, P., Zisserman, A., Eds.; Springer Berlin Heidelberg: Marseille, France, 2008; pp. 234–247.
12. Babenko, B.; Ming-Hsuan, Y.; Belongie, S. Visual tracking with online multiple instance learning. In Proceedings of the IEEE Conference on Computer Vision Pattern Recognition (CVPR) 2009, Miami, FL, USA, 20–25 June 2009; pp. 983–990.
13. Babenko, B.; Ming-Hsuan, Y.; Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632. [[CrossRef](#)]
14. Kaihua, Z.; Lei, Z.; Ming-Hsuan, Y. Real-time object tracking via online discriminative feature selection. *IEEE Trans. Image Process.* **2013**, *22*, 4664–4677.
15. Zhang, K.; Song, H. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recogn.* **2013**, *46*, 397–411. [[CrossRef](#)]
16. Abdechiri, M.; Faez, K.; Amindavar, H. Visual object tracking with online weighted chaotic multiple instance learning. *Neurocomputing* **2017**, *247*, 16–30. [[CrossRef](#)]
17. Yang, H.; Qu, S.; Zhu, F.; Zheng, Z. Robust objectness tracking with weighted multiple instance learning algorithm. *Neurocomputing* **2018**, *288*, 43–53. [[CrossRef](#)]
18. Hu, K.; Zhang, X.; Gu, Y.; Wang, Y. Fusing target information from multiple views for robust visual tracking. *IET Comput. Vis.* **2014**, *8*, 86–97. [[CrossRef](#)]
19. Alexe, B.; Deselaers, T.; Ferrari, V. What is an object? In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 73–80.
20. Alexe, B.; Deselaers, T.; Ferrari, V. Measuring the objectness of image windows. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2189–2202. [[CrossRef](#)] [[PubMed](#)]
21. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient graph-based image segmentation. *Int. J. Comput. Vision* **2004**, *59*, 167–181. [[CrossRef](#)]
22. Smarandache, F. *Neutrosophy: Neutrosophic Probability, Set and Logic*; American Research Press: Rehoboth, MA, USA, 1998; p. 105.

23. Peng, X.; Dai, J. A bibliometric analysis of neutrosophic set: Two decades review from 1998 to 2017. *Artif. Intell. Rev.* **2018**, *52*, 1–57. [[CrossRef](#)]
24. Ye, J.; Fu, J. Multi-period medical diagnosis method using a single valued neutrosophic similarity measure based on tangent function. *Comput. Methods Programs Biomed.* **2016**, *123*, 142–149. [[CrossRef](#)]
25. Guo, Y.; Sengur, A. A novel 3d skeleton algorithm based on neutrosophic cost function. *Appl. Soft Comput. J.* **2015**, *36*, 210–217. [[CrossRef](#)]
26. Guo, Y.; Şengür, A.; Ye, J. A novel image thresholding algorithm based on neutrosophic similarity score. *Meas. J. Int. Meas. Confed.* **2014**, *58*, 175–186. [[CrossRef](#)]
27. Guo, Y.; Şengür, A.; Akbulut, Y.; Shipley, A. An effective color image segmentation approach using neutrosophic adaptive mean shift clustering. *Meas. J. Int. Meas. Confed.* **2018**, *119*, 28–40. [[CrossRef](#)]
28. Guo, Y.; Xia, R.; Şengür, A.; Polat, K. A novel image segmentation approach based on neutrosophic c-means clustering and indeterminacy filtering. *Neural Comput. Appl.* **2017**, *28*, 3009–3019. [[CrossRef](#)]
29. Rashno, A.; Koozekanani, D.D.; Drayna, P.M.; Nazari, B.; Sadri, S.; Rabbani, H.; Parhi, K.K. Fully automated segmentation of fluid/cyst regions in optical coherence tomography images with diabetic macular edema using neutrosophic sets and graph algorithms. *IEEE Trans. Biomed. Eng.* **2018**, *65*, 989–1001. [[CrossRef](#)] [[PubMed](#)]
30. Ashour, A.S.; Guo, Y.; Kucukkulahli, E.; Erdogan, P.; Polat, K. A hybrid dermoscopy images segmentation approach based on neutrosophic clustering and histogram estimation. *Appl. Soft Comput.* **2018**, *69*, 426–434. [[CrossRef](#)]
31. Guo, Y.; Şengür, A. A novel image segmentation algorithm based on neutrosophic similarity clustering. *Appl. Soft Comput. J.* **2014**, *25*, 391–398. [[CrossRef](#)]
32. Fan, E.; Xie, W.; Pei, J.; Hu, K.; Li, X. Neutrosophic hough transform-based track initiation method for multiple target tracking. *IEEE Access* **2018**, *6*, 16068–16080. [[CrossRef](#)]
33. Wang, H.; Smarandache, F.; Zhang, Y.; Sunderraman, R. Single valued neutrosophic sets. *Multispace Multistructure* **2010**, *4*, 410–413.
34. Ye, J. Single valued neutrosophic cross-entropy for multicriteria decision making problems. *Appl. Math. Model.* **2014**, *38*, 1170–1175. [[CrossRef](#)]
35. Ye, J. Vector similarity measures of simplified neutrosophic sets and their application in multicriteria decision making. *Int. J. Fuzzy Syst.* **2014**, *16*, 204–211.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).