Journal of Intelligent & Fuzzy Systems 32 (2017) 1775–1786
DOI:10.3233/JIFS-152381
IOS Press

A novel object tracking algorithm by fusing color and depth information based on single valued neutrosophic cross-entropy

Keli Hu[a,*], Jun Ye[a], En Fan[a], Shigen Shen[a], Longjun Huang[a] and Jiatian Pi[b]
[a]*Department of Computer Science and Engineering, Shaoxing University, Shaoxing, Zhejiang, China*
[b]*Key Laboratory of Wireless Sensor Network and Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China*

Abstract. Although appearance based trackers have been greatly improved in the last decade, they are still struggling with some challenges like occlusion, blur, fast motion, deformation, etc. As known, occlusion is still one of the soundness challenges for visual tracking. Other challenges are also not fully resolved for the existed trackers. In this work, we focus on tackling the latter problem in both color and depth domains. Neutrosophic set (NS) is as a new branch of philosophy for dealing with incomplete, indeterminate and inconsistent information. In this paper, we utilize the single valued neutrosophic set (SVNS), which is a subclass of NS, to build a robust tracker. First, the color and depth histogram are employed as the appearance features, and both features are represented in the SVNS domain via three membership functions T, I, and F. Second, the single valued neutrosophic cross-entropy measure is utilized for fusing the color and depth information. Finally, a novel SVNS based MeanShift tracker is proposed. Applied to the video sequences without serious occlusion in the Princeton RGBD Tracking dataset, the performance of our method was compared with those by the state-of-the-art trackers. The results revealed that our method outperforms these trackers when dealing with challenging factors like blur, fast motion, deformation, illumination variation, and camera jitter.

Keywords: Object tracking, RGBD, fusion, single valued neutrosophic set, cross-entropy measure

1. Introduction

Object tracking has been extensively studied in computer vision due to its applications such as surveillance, human-computer interaction, video indexing, and traffic monitoring, to name a few.

While a lot of effort has been done in the past decades [30, 31, 33], it remains a very challenging task to build a robust tracking system to deal with the problems like occlusion, blur, fast motion, deformation, illumination variation, and rotation, etc.

There are mainly two ways to tackle those problems. One is utilizing robust features. Color model is frequently employed for tracking due to its robustness for confronting blur, deformation and rotation, etc. MeanShift [9] and CAMShift [7] employed color information to separate the object from the background. Both trackers perform well unless a similar color appears around the target. Cross-Bin metric [22], SIFT [36] and texture feature [6] were introduced into the mean shift based trackers, and the enhanced trackers outperform the corresponding traditional tracker. The other way is training an effective adaptive appearance model. Semi-supervised boosting [10] was employed for building a robust classifier; multiple instance learning [4] was introduced into the

*Corresponding author. Keli Hu, Department of Computer Science and Engineering, Shaoxing University, Shaoxing 312000, Zhejiang, China. Tel.: +86 575 88341512; E-mail: ancimoon@gmail.com.

classifier training procedure due to the interference of the inexact training instance; compressive sensing theory [18] was applied for developing effective and efficient appearance models for robust object tracking due to factors such as pose variation, illumination change, occlusion, and motion blur. In addition, Ross et al. [25] proposed the IVT method for dealing with appearance variation, other schemes like kernelized structured support vector machine [16], LGT [8] and TLD [19] also perform well. As known, though many efforts have been done for handling occlusion problem [1, 8, 18, 34], it is still one of the soundness challenges for visual tracking. Other challenges are also not fully resolved for the existed trackers [1–17].

All the trackers mentioned above are based on RGB information. The depth information is directly discarded. That is mainly caused by three reasons. Firstly, most cameras cannot provide depth information. Secondly, though a multi-camera stereo rig can achieve such a goal, 3D reconstruction remains a very challenging task [17]. Lastly, although depth sensors like Microsoft Kinect, Asus Xtion and PrimeSense can produce depth and RGB information, each of them has limitations. For example, all of them are sensitive for distance and illumination. Reliable depth information can only achieve in a limited range, e.g. 0.8–3.5 meters for Kinect. Besides, lack of effective scheme for fusing depth and RGB information is another main reason. Due to the fact that RGBD information can provide another dimension of information for object tracking, a lot of algorithms [2, 14, 15, 23, 27, 28] based on RGBD information have been proposed. However, most algorithms focused on tracking a specific target tracking, e.g. people [2, 14, 15, 23] or hand [28]. Few category free RGBD trackers are proposed. Due to problems like blur, fast motion, deformation, illumination variation and rotation, tracking an object without occlusion in a small area is still a very challenging job for both RGB and RGBD trackers. Thus, finding an effective way to improve existing category free tracker by using RGBD information is very meaningful [27].

Neutrosophic set (NS) [26] is as a new branch of philosophy to deal with the origin, nature and scope of neutralities. It has an inherent ability to handle the indeterminate information like the noise included in images [3, 11, 20, 35] and video sequences. Till now, NS has been successfully applied into many computer vision research fields, such as image segmentation [3, 11, 20, 35] and skeleton extraction [13]. For image segmentation applications, specific neutrosophic image was usually computed [3, 11, 20, 35]. A NS-based cost function between two neighboring voxels was proposed in [13]. In addition, the NS theory is also utilized for improving the clustering algorithm, such as c-means [12]. Decision-making can be regarded as a problem-solving activity terminated by a solution deemed to be satisfactory. A lot NS-based decision making methods [5, 21, 24, 32] were proposed. A single valued neutrosophic set (SVNS) [29] is an instance of a neutrosophic set and provides an additional possibility to represent uncertainty, imprecise, incomplete, and inconsistent information which exist in real world. Therefore, several SVNS-based algorithms, combining with some other metrics, were proposed for handling the multi-criteria decision making problem. Biswas, et al. [5] proposed TOPSIS method for multi-attribute group decision-making. Single valued neutrosophic cross-entropy was introduced in [32]. Fusing features from color and depth domain is also an indeterminate problem, and it can be translated into a kind of decision making problem. NS is still an open area for information fusion applications. Therefore, it is meaningful to form a bridge between NS theory and information fusion.

1.1. Proposed contribution

In this work, the proposed tracking algorithm based on RGBD data mainly exhibits three contributions. First, a more accurate ROI for initializing target model is achieved using depth-based segmentation. Secondly, the depth distance of the target between adjacent frames is incorporated into the back-projection to facilitate object discrimination. Finally, a color-depth fusion method based on single valued neutrosophic cross-entropy is proposed to enhance object tracking. To our own knowledge, it is the first time to introduce the NS theory into the visual object tracking domain.

The remainder of this paper is organized as follows. In Section 2, main steps and basic flow of our algorithm is first given, and then the details of the proposed algorithm are illustrated in the following subsections. Experimental evaluations and discussions are presented in Section 3, and Section 4 is the conclusion.

2. Problem formulation

In this section, we present the algorithmic details of this paper.

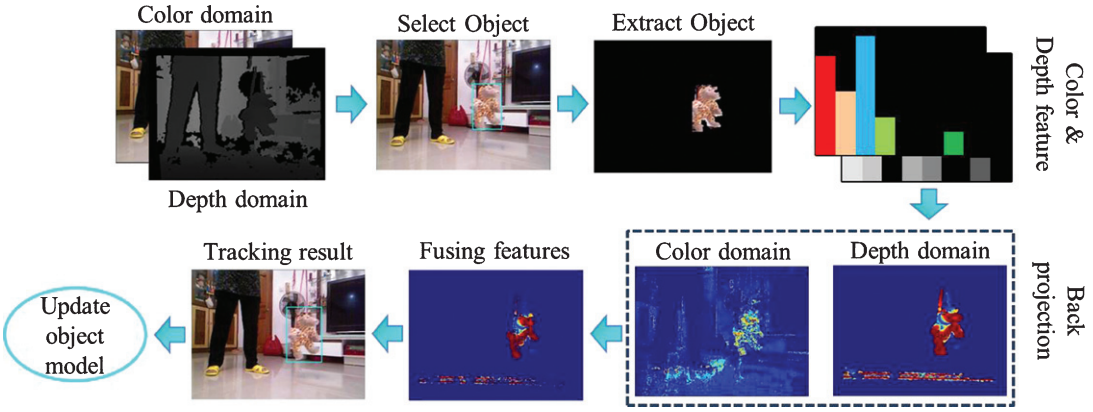

Fig. 1. Main steps of the proposed algorithm.

Table 1
Basic flow of the proposed tacking algorithm

Algorithm 1 SVNCE Tracking

Initialization
Input: 1-st video frame in the color and depth domain
1) Select an object on the image plane as the target
2) Select object seeds in the depth domain
3) Extract object from the image using the depth information
4) Calculate the corresponding color and depth histograms as object model
Tracking
Input: (t+1)-th video frame in the color and depth domain
1) Calculate back-projections in both color and depth domain
2) Represent both features in the NS domain via three membership subsets T, I, and F
3) Fusing color and depth information using the single valued neutrosophic cross-entropy method
4) Find the location of the object in the CAMShift framework
5) Update object model and seeds
Output: Tracking location

The main steps of our tracking system are summarized in Fig. 1. Algorithm 1 illustrates the basic flow of our algorithm, as shown in Table 1. Details of each main step of our algorithm are given in the following subsections.

2.1. Extracting object

The bounding box (as shown in Fig. 2) is always applied for indicating the location of the target by most trackers [30, 31, 33]. The color information in the bounding box is frequently employed to initialize the target's model [6, 7, 9, 22], represented as a color histogram. However, in addition to the target, the area in the bounding box sometimes contains background information. Thus, such a model could not represent the target's feature exactly.

Given an initial bounding box (as shown in the left part of Fig. 2), we try to extract the target's area with the help of the depth data. A depth-based method for extracting the target's area is proposed.

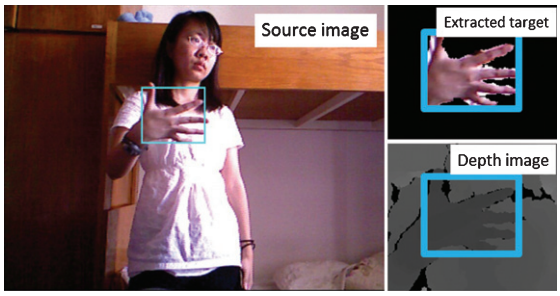

Fig. 2. Example of target extracting.

It is reasonable that we assume target area is the part which is closer to us. In addition, the target occupies most space of the bounding box. Then the target seeds can be automatically selected by

$$\mathbf{S} = \{\mathbf{x} | \, \mathbf{RD}(\mathbf{x}) = r_i, \; i = 1 \ldots n\} \tag{1}$$

where $\mathbf{RD}(\mathbf{x})$ is the distance rank in the bounding box, e.g., $\mathbf{RD}(\mathbf{x}) = 5\%$ means the depth at pixel location

x placed at the order 5% N_{Bbox} by sorting the depth of each point in the bounding box from near to far. Here, N_{Bbox} is the total number of the pixels in the bounding box.

Then the target region **A** owns n points, suppose **B** is the set of the points which borders at least one of the points of **A**, at each loop, target region is updated by

$$\mathbf{A}^{+} = \{\mathbf{x}| \, |\mathbf{D}(\mathbf{x}) - mean_{\mathbf{x'}\in\mathbf{A}}\{\mathbf{D}(\mathbf{x'})\}| < T, \mathbf{x} \in \mathbf{P}\} \tag{2}$$

where $\mathbf{A}^{+}$ is the newly added pixel set, **D(x')** is the depth at pixel location **x'**.

In this work, we set five ranks in Equations (1) for seed selection, where r_1, r_2, r_3, r_4, r_5 are set as 10%, 15%, 20%, 25%, 30% respectively. Experimental results have proved its robustness. As shown in Fig. 2, given a rough bounding box, our method can extract the target area much more exactly. The rough edge of the extracted target is mainly caused by the noise of the source data in the depth domain. Due to the RGB and depth data are not tightly aligned, thumb dislocation is also occurred.

2.2. Calculating back-projections

Back-projection is a probability distribution map with the same size as the input image. Each pixel value of the back-projection demonstrates the likelihood of the corresponding pixel located in the area of the tracked object on the current image plane. Before calculating the back-projection, we build the object model in both color and depth domain when the target area is extracted.

Let $\{\mathbf{x}_i\}_{i=1\ldots n}$ be the pixel locations in the region of the target area, the function b: $\mathrm{R}^2 \rightarrow \{1 \ldots m\}$ associates to the pixel at location $\mathbf{x}_i$ the index $b(\mathbf{x}_i)$ of its bin in the quantized feature space. The probability of the feature $u = 1, 2, \ldots, m$ in the object model is then computed by

$$\hat{q}_u^c = C\sum_{i=1}^{n}\delta\left[b^c(\mathbf{x}_i) - u\right],$$
$$\hat{q}_u^d = D\sum_{i=1}^{n}\delta\left[b^d(\mathbf{x}_i) - u\right] \tag{3}$$

where b^c is the transformation function in color domain, b^d is for the depth domain, δ is the Kronecker delta function. C and D is the normalization constant derived by imposing the conditions $\sum_{u=1}^{m^c}\hat{q}_u^c = 1$ and $\sum_{u=1}^{m^d}\hat{q}_u^d = 1$.

2.2.1. Calculating the back-projection in color domain

As shown in Algorithm 1, back-projections in both color and depth domain should be computed in every loop of the tracking procedure. The color histogram $\hat{q}_u^c$ is employed to calculate the back-projection in color domain:

$$\mathbf{P}_c(\mathbf{x}) = \hat{q}^c_{b^c(\mathbf{x})} \tag{4}$$

where **x** is the pixel location.

2.2.2. Calculating the back-projection in depth domain

We assume that the object need to be tracked is a relative low speed target. For a space point which is reconstructed by a RGBD sensor, it is reasonable to assign a higher probability value to it if it is closer to the location of the target at the previous time. In addition, only these pixels which are not very far from the previous location of the target on the image plane may belong to the target. Thus, if only the space restriction is considered, the probability of the pixel **x** obtained from the target at current time can be calculated by

$$\mathbf{P}_d(\mathbf{x}) = \frac{1}{2}erfc\left(4\frac{d(\mathbf{x},\mathbf{T})}{MAXD} - 2\right), \mathbf{x} \in 2\mathbf{R}_{pre} \tag{5}$$

where $MAXD$ is the maximum depth distance between the previous and current target's locations, $2\mathbf{R}_{pre}$ is the points set which is covered by a bounding box with the twice size than the previous one, but with the same center, and then $d(\mathbf{x}, \mathbf{T})$ is the distance between a point and the target, which is approximately calculated by

$$d(\mathbf{x}, \mathbf{T}) = \min\left(\left|\mathbf{D}(\mathbf{S}_{r_i}) - \mathbf{D}(\mathbf{x})\right|\right), \; i = 1 \ldots n \tag{6}$$

where $\mathbf{S}_{ri}$ is the i-th seed of the target.

2.3. Fusing color and depth information

Employing discriminative feature is one of the most critical factors for a robust tracker, and the method for selecting discriminative feature during the tracking process is still an open issue. A well discriminative feature owns the ability of effectively setting the target apart from the clutter background. Color and depth features are used by our tracker. However, similar depth or color may appear surrounding the target, and the tracker will fail if a bad feature is applied. To build a robust feature fusion mechanism, the single valued neutrosophic cross-entropy measure [32] is utilized here.

2.3.1. Cross-entropy measure of SVNSs for decision making

For a multicriteria decision-making problem, Let $A = \{A_1, A_2, \ldots, A_m\}$ be a set of alternatives and $C = \{C_1, C_2, \ldots, C_n\}$ be a set of criteria. Assume w_j is the weight of the criteria $\underline{C}_j$, $w_j \in [0, 1]$, and $\sum_{j=1}^{n} w_j = 1$. Then the character of the alternative A_i $(i = 1, 2 \ldots m)$ can be represented by the following SVNS information:

$$A_i = \left\{ \left\langle C_j, T_{C_j}(A_i), I_{C_j}(A_i), F_{C_j}(A_i) \right\rangle \middle| \, C_j \in C \right\}$$
$$i = 1, 2 \ldots m, \; j = 1, 2 \ldots n \tag{7}$$

where $T_{C_j}(A_i), I_{C_j}(A_i), F_{C_j}(A_i) \in [0, 1]$. $T_{C_j}(A_i)$ 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.

In the multicriteria decision-making problem, a weighted cross entropy measure between any alternative A_i and the ideal alternative $A^* = \{< 1, 0, 0 >, < 1, 0, 0 >, \ldots, < 1, 0, 0 >\}$ is proposed in SVNS domain [32] as follows:

$$\begin{aligned} D_i = & \sum_{j=1}^{n} w_j \left[\begin{array}{l} \log_2 \dfrac{1}{\frac{1}{2}(1 + T_{C_j}(A_i))} \\ + \log_2 \dfrac{1}{1 - \frac{1}{2} I_{C_j}(A_i)} + \log_2 \dfrac{1}{1 - \frac{1}{2} F_{C_j}(A_i)} \end{array} \right] \\ & + \sum_{j=1}^{n} w_j \left[\begin{array}{l} T_{C_j}(A_i) \log_2 \dfrac{T_{C_j}(A_i)}{\frac{1}{2}(1 + T_{C_j}(A_i))} \\ + (1 - T_{C_j}(A_i)) \log_2 \dfrac{1 - T_{C_j}(A_i)}{1 - \frac{1}{2}(1 + T_{C_j}(A_i))} \end{array} \right] \\ & + \sum_{j=1}^{n} w_j \left[I_{C_j}(A_i) + (1 - I_{C_j}(A_i)) \log_2 \frac{1 - I_{C_j}(A_i)}{1 - \frac{1}{2} I_{C_j}(A_i)} \right] \\ & + \sum_{j=1}^{n} w_j \left[F_{C_j}(A_i) + (1 - F_{C_j}(A_i)) \log_2 \frac{1 - F_{C_j}(A_i)}{1 - \frac{1}{2} F_{C_j}(A_i)} \right] \end{aligned} \tag{8}$$

The smaller the value of D_i is, the better the A_i is. That is, the alternative A_i with smaller D_i is closer to the idea alternative. Thus, after calculating each D_i, we can decide which alternative A_i is the best one.

2.3.2. Information fusion

Both color and depth features are expressed in the SVNS domain by $A_i = \{T_{C_j}(A_i), I_{C_j}(A_i), F_{C_j}(A_i)\}$. Each feature corresponds to an alternative A_i. A^c corresponds to the color feature, and A^d corresponds to the depth feature. For the proposition of color feature is a discriminative feature, $T(A^c)$, $I(A^c)$, $F(A^c)$ represent the probability of such a proposition is true, indeterminate and false degrees, respectively. Using the near region similarity criterion, we can define as:

$$T_{C_n}(A^c) = \sum_{u=1}^{m^c} \sqrt{\hat{q}_u^c \hat{p}_u^c} \tag{9}$$

$$I_{C_n}(A^c) = \sum_{u=1}^{m^c} \sqrt{\hat{q}_u^c \hat{p}_u^{c\prime}} \tag{10}$$

$$F_{C_n}(A^c) = 1 - T_{C_n}(A^c) \tag{11}$$

Equation (9) is the Bhattacharyya coefficient which is frequently employed as a similarity judgment [9], where $\hat{q}_u^c$ is the object model in the color domain, $\hat{p}_u^c$ is the histogram feature corresponding to the tracking location (a rectangle bounding box, region **G** in Fig. 3) in the previous frame.

The indeterminacy degree to which the alternative A^c satisfies or does not satisfy the criteria is defined in Equation (10), where $\hat{p}_u^{c\prime}$ corresponds to the near region $\mathbf{G}_n$, $\mathbf{G}_n = \alpha\mathbf{G} - \mathbf{G}$, as shown in Fig. 3. Both $\hat{p}_u^c$ and $\hat{p}_u^{c\prime}$ are computed by using Equation (3).

As the location estimated by the tracker may sometimes drifts from the target, to make the information fusion results robust to tracking location errors, we integrate the other condition, far region similarity criteria C_f, into the multicriteria decision-making problem.

In the SVNS, the three functions using the far region similarity criteria C_f are defined as:

$$\begin{aligned} T_{C_f}(A^c) &= T_{C_n}(A^c) \\ I_{C_f}(A^c) &= \sum_{u=1}^{m^c} \sqrt{\hat{q}_u^c \hat{p}_u^{c\prime\prime}} \\ F_{C_f}(A^c) &= F_{C_n}(A^c) \end{aligned} \tag{12}$$

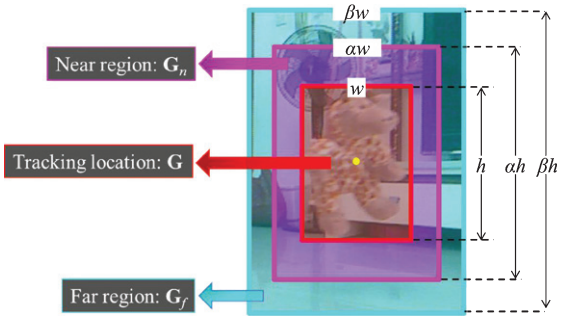

Fig. 3. Region illustration for information fusion.

where $\hat{p}_u^{c\prime\prime}$ is the histogram feature corresponding to the far region $\mathbf{G}_f$, $\mathbf{G}_f = \beta\mathbf{G} - \mathbf{G}_n$.

The method for computing the subsets T, I, and F for the depth feature is the same as the above steps. Similarly, for the proposition of depth feature is a discriminative feature, $T(A^d)$, $I(A^d)$, $F(A^d)$ represent the probability of such a proposition is true, indeterminate and false degrees, respectively. By applying the criteria C_n and C_f, the related functions are presented as follows:

$$T_{C_n}(A^d) = \sum_{u=1}^{m^d}\sqrt{\hat{q}_u^d \hat{p}_u^d},\ I_{C_n}(A^d) = \sum_{u=1}^{m^d}\sqrt{\hat{q}_u^d \hat{p}_u^{d\prime}},$$

$$F_{C_n}(A^d) = 1 - T_{C_n}(A^d)$$

$$T_{C_f}(A^d) = T_{C_n}(A^d),\ I_{C_f}(A^d) = \sum_{u=1}^{m^d}\sqrt{\hat{q}_u^d \hat{p}_u^{d\prime\prime}},$$

$$F_{C_f}(A^d) = F_{C_n}(A^d)$$

where $\hat{q}_u^d$ is the object depth model, $\hat{p}_u^{d\prime}$ and $\hat{p}_u^{d\prime\prime}$ is the corresponding depth histogram feature for the regions of G_n and G_f, respectively.

Substituting $T_{C_n}(A^c)$, $I_{C_n}(A^c)$, $F_{C_n}(A^c)$, $T_{C_f}(A^c)$, $I_{C_f}(A^c)$, $F_{C_f}(A^c)$, $T_{C_n}(A^d)$, $I_{C_n}(A^d)$, $F_{C_n}(A^d)$, $T_{C_f}(A^d)$, $I_{C_f}(A^d)$, $F_{C_f}(A^d)$ into Equation (8), we can obtain two values, D_c and D_d. Then we can decide which feature to choose. Considering a single feature may result in a confusing back-projection, we fusing features in both domains, the new back-projection after the fusion is defined as

$$\mathbf{P}(\mathbf{x}) = \begin{cases} \mathbf{P}_c(\mathbf{x}) & if\ D_c \geq D_d, \mathbf{P}_d(\mathbf{x}) \geq T_d \\ 0 & if\ D_c \geq D_d,\ \mathbf{P}_d(\mathbf{x}) < T_d \\ \mathbf{P}_d(\mathbf{x}) & if\ D_c < D_d, \mathbf{P}_c(\mathbf{x}) \geq T_c \\ 0 & if\ D_c < D_d,\ \mathbf{P}_c(\mathbf{x}) < T_c \end{cases} \quad (13)$$

where T_d and T_c are two thresholds. As shown in Equation (13), the feature with a smaller value of cross-entropy decides the final back-projection, and we call this feature the main feature. In addition, the left feature is utilized to remove the possible noise.

2.4. Tracking the object

The core of the CAMShift algorithm [7] is employed here. We choose the previous bounding box of the target as the mean shift window. Then the tracking location can be calculated by

$$x = \frac{M_{10}}{M_{00}},\ y = \frac{M_{01}}{M_{00}} \quad (14)$$

where $M_{10} = \sum_{\mathbf{x}\in BBox} x\mathbf{P}(\mathbf{x})$, $M_{01} = \sum_{\mathbf{x}\in BBox} y\mathbf{P}(\mathbf{x})$, and $M_{00} = \sum_{\mathbf{x}\in BBox} \mathbf{P}(\mathbf{x})$. Then, the size of the bounding box is $s = 2\sqrt{M_{00}/256}$.

2.5. Update object model

After finding the location of the target in the current frame, we begin to update the object model in both color and depth domain.

Before updating the object model, the object seed set is updated using Equation (1). Instead of considering all the pixels located in the bounding box, these pixels which satisfy $\mathbf{P}(\mathbf{x}) > T_s$ are considered for updating the object seeds during the tracking process.

By utilizing the object seeds, the target area can be well extracted with the help of the depth information. Thus a more exact color histogram can be computed at each time of the whole video sequence. Suppose $\hat{p}_u^c(t)$ is the color histogram corresponding to the extracted target area at time t, then the updated color model of the object can be calculated by

$$\hat{q}_u^c(t) = \lambda\hat{q}_u^c(t-1) + (1-\lambda)\hat{p}_u^c(t) \quad (15)$$

where $\lambda \in (0,\ 1)$.

Considering that the depth distribution of the target shifts faster than the color's and the extracted target area is relatively credible, we replace the previous object depth model by the new model:

$$\hat{q}_u^d(t) = \hat{p}_u^d(t) \quad (16)$$

3. Experimental result and discussions

We tested our algorithm on several challenging video sequences which are publicly available in the Princeton RGBD Tracking dataset. All of the sequences are captured from a Kinect sensor, and the information of both color and depth domain is provided. As mentioned at the beginning, we try to propose a robust algorithm for tackling challenging factors along with object tracking, such as blur, fast motion, deformation, illumination variation, and camera jitter, without considering occlusion. Thus, several sequences without serious occlusion challenge are selected as testing sequences.

To gauge absolute performance, we compare our results to four state-of-the-art trackers including CT [18], LGT [8], IVT [25] and TLD [19]. All the four tackers are based on the scheme of tracking-by-detection except LGT. Two layers are employed by the LGT tracker. Local patches which represent

the target's geometric deformation in the local layer are updated by using global visual properties, such as color, shape, and apparent local motion.

3.1. Setting parameters

For the proposed algorithm, five object seeds are kept during the tracking procedure, thus, in Equations (1), five ranks are selected, where r_1, r_2, r_3, r_4, r_5 are set as 10%, 15%, 20%, 25%, 30% respectively. The parameter T which decides the accuracy of the segmentation of the target area is set to 60 mm in Equation (2). The value of the parameter $MAXD$ in Equation (5) depends on the displacement of the target between adjacent frames. According to the attribution of the dataset employed in this work, $MAXD$ is set to 70 mm. In order to keep enough information in the fused back-projection map, both parameters T_c, T_d defined in Equation (13) should be assigned a relatively low value, all of them are set to 0.1 here. Finally, all parameters were kept constant for all experiments.

3.2. Evaluation criteria

Two kinds of evaluation criteria are considered. The center position error is plotted based on the location error metric and the success is plotted based on the overlap metric. The center position error is on the base of the Euclidean distance between the center location of the tracked target and the manually labeled ground truth in each frame.

The overlap score is defined as

$$s_i = \frac{area\left(\mathrm{ROI}_{T_i} \cap \mathrm{ROI}_{G_i}\right)}{area\left(\mathrm{ROI}_{T_i} \cup \mathrm{ROI}_{G_i}\right)} \tag{17}$$

where ROI_{Ti} is the target bounding box in the i-th frame and ROI_{Gi} is the corresponding ground truth bounding box. By setting an overlap score r which is defined as the minimum overlap ratio, one can decide whether an output is correct or not, the success ratio can be calculated by the following formula:

$$R = \sum_{i=1}^{N} u_i \Big/ N, \quad u_i = \begin{cases} 1 & if\ s_i > r \\ 0 & otherwise \end{cases} \tag{18}$$

where N is the number of frames.

3.3. Tracking results

The screen captures for some of the clips are shown in Figs. 4–8. The quantitative plots are given in Figs. 9, 10. Details of these video sequences are shown in Table 2. A more detailed discussion of the tracking results is described below.

Hand_no_occ sequence: This sequence highlights the challenges of target's deformation, illumination variation. As shown in Fig. 4, both CT and IVT trackers fails soon due to that much background area are judged as object model at the phase of tracker initialization. The TLD tracker performs better than CT and IVT. However, as shown in frame #61, TLD also lost the target on account of the serious deformation of the hand. On the contrary, the LGT tracker successfully tackled the problem of deformation. Overall,

Fig. 4. Screenshots of tracking results of the video sequence used for testing (*hand_no_occ*, target is selected in frame #1).

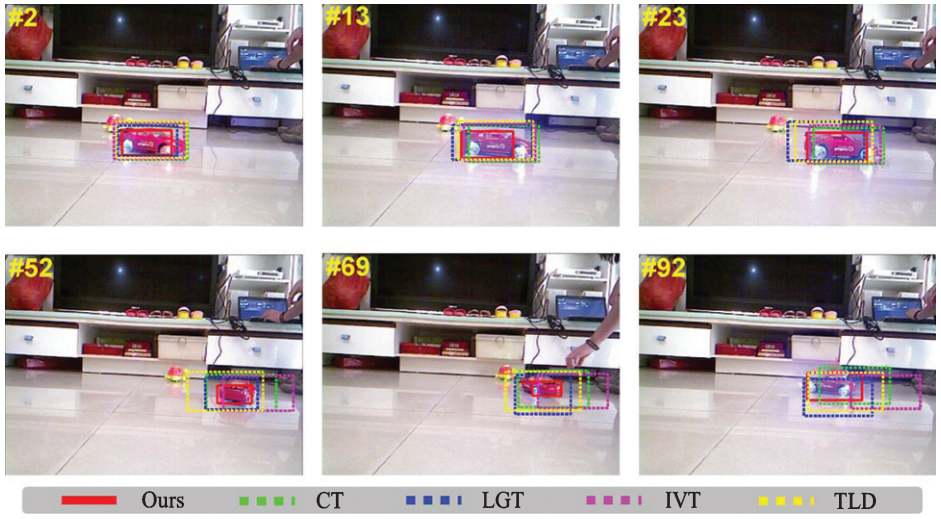

Fig. 5. Screenshots of tracking results of the video sequence used for testing (*toy_car_no*, target is selected in frame #1).

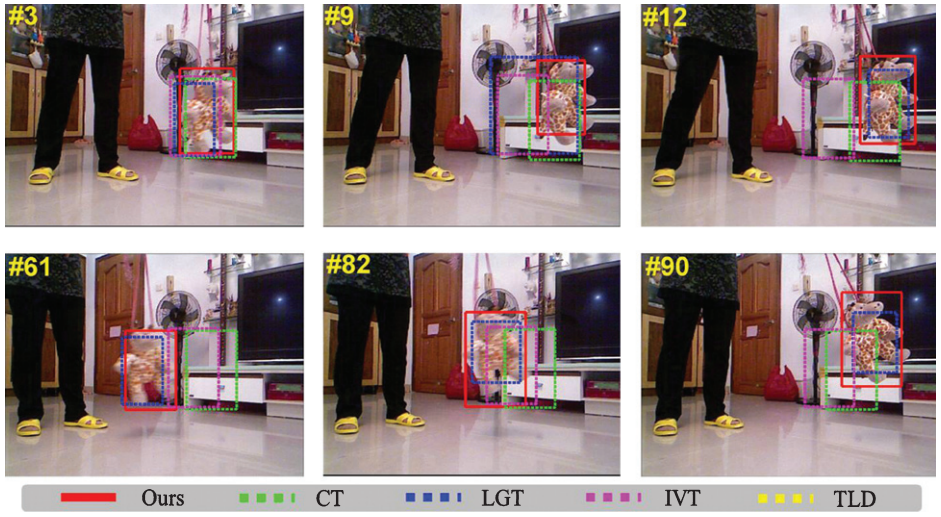

Fig. 6. Screenshots of tracking results of the video sequence used for testing (*toy_no*, target is selected in frame #1).

as shown in Fig. 9, our tracker gives the best performance. As seen in frame #18, #61, #134, #175, our tracker can produce a more accurate bounding box.

Toy_car_no sequence: This sequence presents challenging target rotation and light changes (lights mounted on the car winked sometimes, as seen in frame #13, #92 in Fig. 5). As shown in Fig. 5, all the trackers perform well before frame #23. In the course of turning, trackers except ours begin to lose the toy car. Both the feature model and the model updating method employed by CT, LGT, IVT and TLD cannot fit the serious change of the target's appearance and size, which leads to failures.

Toy_no sequence: Challenges of blur, fast motion and rotation are presented in this sequence. As shown in Fig. 6, the CT and IVT trackers have already failed in frame #9 on account of the factors of blur and fast motion. Both ours and the LGT tracker perform well throughout the sequence. However, as shown in frame #9, we can see from the estimated bounding boxes that the size of the object is often poorly estimated by the LGT tracker, which leads to failures. That is mainly because the sudden move of the target, and the update of local patched cannot follow such a rapid change.

Zball_no1 sequence: This sequence presents the challenges of illumination change, rapid motion, camera jitter, similar color and depth information.

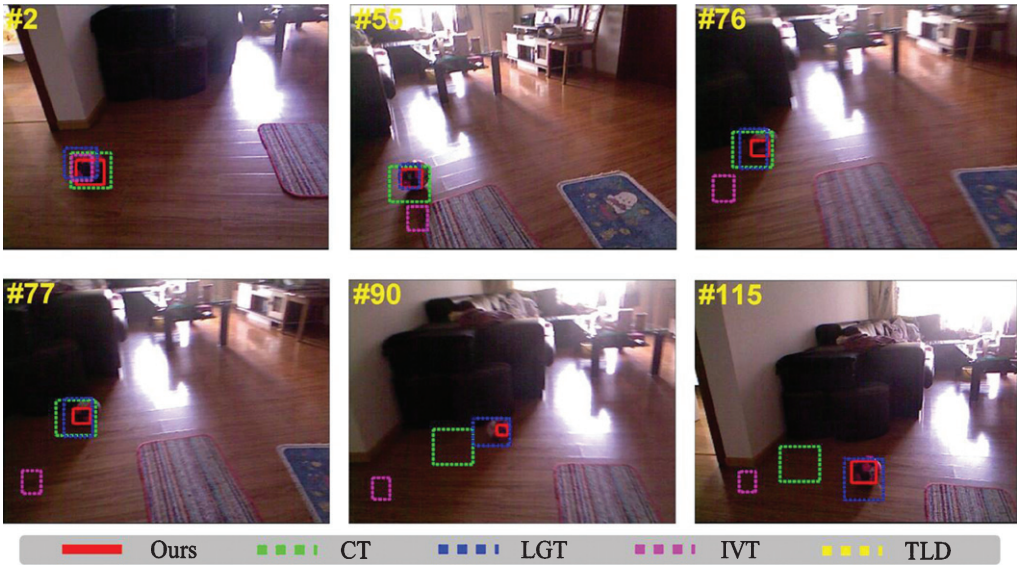

Fig. 7. Screenshots of tracking results of the video sequence used for testing (*zball_no1*, target is selected in frame #1).

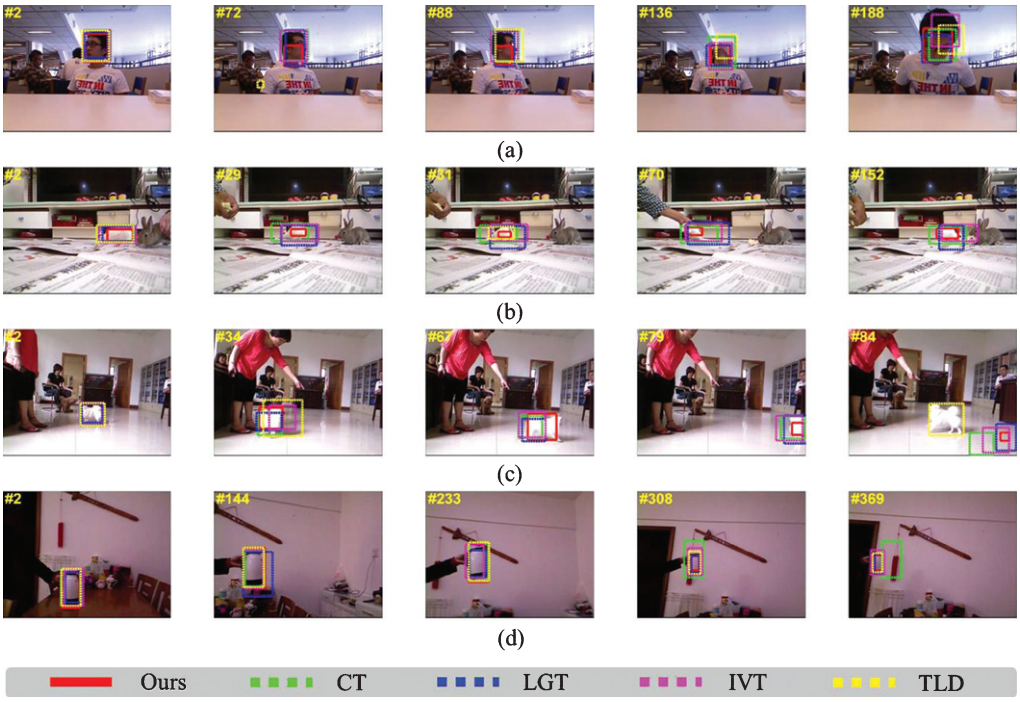

Fig. 8. Screenshots of tracking results of the video sequence used for testing (each target is selected in frame #1). (a) *new_ye_no_occ*; (b) *wr_no1*; (c) *wdog_no1*; (d) *zcup_move_1*.

As shown in Fig. 7, the TLD tracker fails soon. An inappropriate size of the bounding box is estimated by the IVT tracker, and it also fails soon (as seen in Fig. 9). As shown in frames #76 and #77 in Fig. 7, when the tracked ball rolled into the shadow of the sofa, both fast move and camera jitter happens. The CT tracker lost the target ball, and the LGT tracker cannot produce a proper bounding box because of the sudden movement and similar background. As seen in Figs. 7 and 9, our tracker performs best.

Other sequences: The tracking results of another four sequences are given in Figs. 8 and 10. The plot of

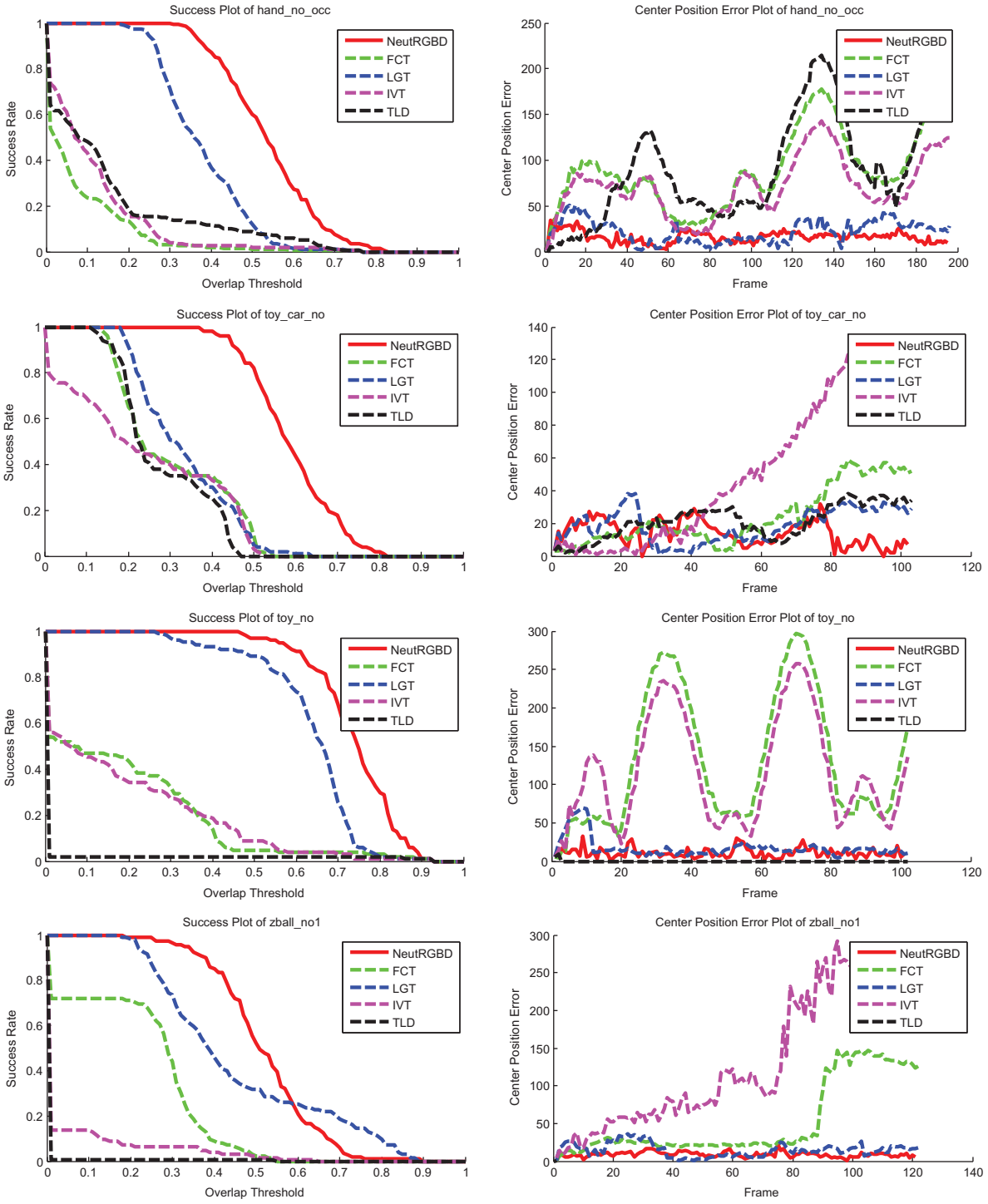

Fig. 9. Success and center position error plots of the sequence *hand_no_occ*, *toy_car_no*, *toy_no* and *zball_no1*.

the mean success rate of all the sequences is shown in Fig. 10. As seen in Figs. 8 and 10, the proposed tracker performs best among all the trackers compared in this work.

Analysis of our tracker: Figs. 9, 10 present the tracking results in terms of center location error and success rate. Our tracker achieves much better results than other trackers. By extracting target

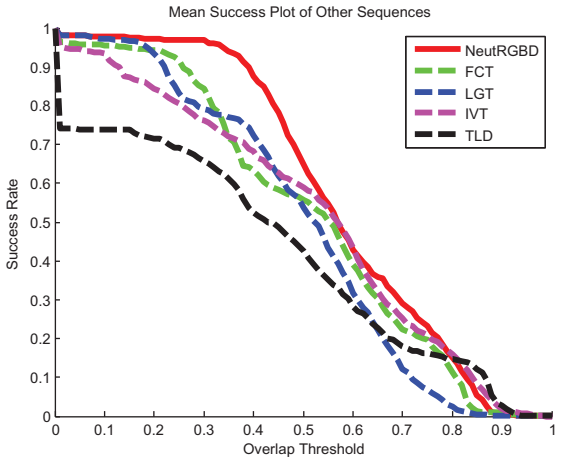

Fig. 10. Success and center position error plots of the sequence *hand_no_occ*, *toy_car_no*, *toy_no* and *zball_no1*.

Table 2
An overview of the video sequences

Sequence	Target	Challenges	Frames
hand_no_occ	hand	deformation, illumination variation	196
toy_car_no	toy car	rotation, illumination variation, scale change	103
toy_no	toy	blur, fast motion, rotation	102
zball_no1	ball	illumination variation, rapid motion, camera jitter	122
new_ye_no_occ	head	rolling, appearance change	235
wr_no1	rabit	rolling, blur, fast motion, appearance change	156
wdog_no1	dog	rolling, blur, fast motion, appearance chan	86
zcup_move_1	cup	scale change	370

area in depth domain, the noisy background information can be filtered (e.g. *hand_no_occ* and *toy_no*), and a more reliable object model can be achieved when illumination or object orientation changes (e.g. *toy_car_no* and *zball_no1*). By employing the multicriteria decision-making method in NS domain (the key of our tracker), the information fusion facilitates enhancing the robustness of the back-projection. Challenges of similar information in color and depth domain can be tackled in all of the sequences. Challenges of blur and fast motion are successfully tackled by using a relative large searching area and robust back-projection.

4. Conclusions

In this paper, we present a new scheme for tracking an object in RGBD domain. A distribution map in the depth domain is calculated by employing several object seeds. The object seeds are updated during the whole tracking procedure depending on the fused back-projection. The information fusion problem in both color and depth domain is translated into a multicriteria decision-making problem. Two kinds of criteria are proposed and the cross-entropy of SVNSs is utilized to tackling the information fusion problem. Such a discriminative back-projection leads to a more robust and efficient tracker. Experimental results on challenging video sequences demonstrate that our tracker achieves favorable performance when compared with several state-of-the-art algorithms. As discussed in this paper, we focus on the tracking task without serious occlusion. It will be our primary mission to try to tackle the occlusion problem through the RGBD information in future.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant no. 61603258, the Public Welfare Technology Application Research Project of Zhejiang Province in China under Grant no. 2016C31082, and the scientific research project of Shaoxing University under Grant no. 2014LG1009.

References

[1] A. Adam, E. Rivlin and I. Shimshoni, Robust Fragments-based Tracking using the Integral Histogram, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, 2006, pp. 798–805.

[2] E.J. Almazan and G.A. Jones, Tracking People across Multiple Non-overlapping RGB-D Sensors, in: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 831–837.

[3] A.M. Anter, A.E. Hassanien, M.A.A. ElSoud and M.F. Tolba, Neutrosophic sets and fuzzy C-means clustering for improving CT liver image segmentation, *Advances in Intelligent Systems and Computing* **303** (2014), 193–203.

[4] B. Babenko, Y. Ming-Hsuan and S. Belongie, Robust object tracking with online multiple instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33** (2011), 1619–1632.

[5] P. Biswas, S. Pramanik and B.C. Giri, TOPSIS method for multi-attribute group decision-making under single-valued neutrosophic environment, *Neural Computing and Applications* **27**(3) (2015), 727–737.

[6] F. Bousetouane, L. Dib and H. Snoussi, Improved mean shift integrating texture and color features for robust real time object tracking, *The Visual Computer* **29** (2013), 155–170.

[7] G.R. Bradski, Real time face and object tracking as a component of a perceptual user interface, in: *Proceedings of the*

Fourth IEEE Workshop on Applications of Computer Vision, 1998, pp. 214–219.

[8] L. Cehovin, M. Kristan and A. Leonardis, Robust visual tracking using an adaptive coupled-layer visual model, *IEEE Transactions on Pattern Analysis And Machine Intelligence* **35** (2013), 941–953.

[9] D. Comaniciu, V. Ramesh and P. Meer, Kernel-based object tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25** (2003), 564–577.

[10] H. Grabner, C. Leistner and H. Bischof, Semi-supervised on-Line Boosting for Robust Tracking, in: *European Conference on Computer Vision (ECCV)*, D. Forsyth, P. Torr and A. Zisserman, eds., Springer Berlin Heidelberg, Marseille, 2008, pp. 234–247.

[11] Y. Guo and A. Şengür, A novel image segmentation algorithm based on neutrosophic similarity clustering, *Applied Soft Computing Journal* **25** (2014), 391–398.

[12] Y. Guo and A. Sengur, NCM: Neutrosophic c-means clustering algorithm, *Pattern Recognition* **48** (2015), 2710–2724.

[13] Y. Guo and A. Sengur, A novel 3D skeleton algorithm based on neutrosophic cost function, *Applied Soft Computing Journal* **36** (2015), 210–217.

[14] J. Han, E.J. Pauwels, P.M. De Zeeuw and P.H.N. De With, Employing a RGB-D sensor for real-time tracking of humans across multiple re-entries in a smart environment, *IEEE Transactions on Consumer Electronics* **58** (2012), 255–263.

[15] J. Han, L. Shao, D. Xu and J. Shotton, Enhanced computer vision with Microsoft Kinect sensor: A review, *IEEE Transactions on Cybernetics* **43** (2013), 1318–1334.

[16] S. Hare, A. Saffari and P.H.S. Torr, Struck: Structured output tracking with kernels, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 263–270.

[17] X. Hu and P. Mordohai, Evaluation of stereo confidence indoors and outdoors, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1466–1473.

[18] Z. Kaihua, Z. Lei and Y. Ming-Hsuan, Fast Compressive Tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36** (2014), 2002–2015.

[19] Z. Kalal, K. Mikolajczyk and J. Matas, Tracking-learning-detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34** (2012), 1409–1422.

[20] E. Karabatak, Y. Guo and A. Sengur, Modified neutrosophic approach to color image segmentation, *Journal of Electronic Imaging* **22**(1) (2013), 4049–4068.

[21] A. Kharal, A neutrosophic multi-criteria decision making method, *New Mathematics and Natural Computation* **10** (2014), 143–162.

[22] I. Leichter, Mean shift trackers with cross-bin metrics, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34** (2011), 695–706.

[23] J. Liu, Y. Liu, Y. Cui and Y.Q. Chen, Real-time human detection and tracking in complex environments using single RGBD camera, in: *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 3088–3092.

[24] P. Majumdar, Neutrosophic sets and its applications to decision making, in: *Adaptation, Learning, and Optimization*, 2015, pp. 97–115.

[25] D. Ross, J. Lim, R.-S. Lin and M.-H. Yang, Incremental learning for robust visual tracking, *International Journal of Computer Vision* **77** (2008), 125–141.

[26] F. Smarandache, *Neutrosophy: Neutrosophic probability, set and logic*, in, American Research Press, Rehoboth, 1998, p. 105.

[27] S. Song and J. Xiao, Tracking revisited using RGBD camera: Unified benchmark and baselines, in: *IEEE International Conference on Computer Vision*, 2013, pp. 233–240.

[28] S. Sridhar, A. Oulasvirta and C. Theobalt, Interactive markerless articulated hand motion tracking using RGB and depth data, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2456–2463.

[29] H. Wang, F. Smarandache, Y. Zhang and R. Sunderraman, Single valued neutrosophic sets, *Multispace and Multistructure* **4** (2010), 410–413.

[30] Y. Wu, J. Lim and M.H. Yang, Online object tracking: A benchmark, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2411–2418.

[31] H. Yang, L. Shao, F. Zheng, L. Wang and Z. Song, Recent advances and trends in visual tracking: A review, *Neurocomputing* **74** (2011), 3823–3831.

[32] J. Ye, Single valued neutrosophic cross-entropy for multicriteria decision making problems, *Applied Mathematical Modelling* **38** (2014), 1170–1175.

[33] A. Yilmaz, O. Javed and M. Shah, Object tracking: A survey, *Acm Computing Surveys* **38** (2006), Article 13.

[34] A. Yilmaz, L. Xin and M. Shah, Contour-based object tracking with occlusion handling in video acquired using mobile cameras, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26** (2004), 1531–1536.

[35] M. Zhang, L. Zhang and H.D. Cheng, A neutrosophic approach to image segmentation based on watershed method, *Signal Processing* **90** (2010), 1510–1517.

[36] C. Zhu, *Video object tracking using SIFT and mean shift*, Master Thesis in Communication Engineering, 2011.