FusionSORT: Fusion Methods for Online Multi-object Visual Tracking

Nathanael L. Baisa
School of Computer Science and Informatics
De Montfort University
Leicester LE1 9BH, UK
nathanael.baisa@dmu.ac.uk

***Abstract*—In this work, we investigate four different fusion methods for associating detections to tracklets in multi-object visual tracking. In addition to considering strong cues such as motion and appearance information, we also consider weak cues such as height intersection-over-union (height-IoU) and tracklet confidence information in the data association using different fusion methods. These fusion methods include minimum, weighted sum based on IoU, Kalman filter (KF) gating, and hadamard product of costs due to the different cues. We conduct extensive evaluations on validation sets of MOT17, MOT20 and DanceTrack datasets, and find out that the choice of a fusion method is key for data association in multi-object visual tracking. We hope that this investigative work helps the computer vision research community to use the right fusion method for data association in multi-object visual tracking. The source code is available at https://github.com/nathanlem1/FusionSORT.**

***Index Terms*—Multi-object tracking, Strong cues, Weak cues, Fusion methods, Data association**

I. Introduction

Multi-object visual tracking is currently an active research field in computer vision due to its wide range of applications including, but not limited to, intelligent surveillance, autonomous driving, robot navigation and augmented reality. Its main goal is to detect objects and recognize their identities in video stream in order to produce their trajectories. The most commonly adopted paradigm for multi-object visual tracking in computer vision is tracking-by-detection [1] [2] [3] [4] [5] [6] [7]. This is a result of the significant progress achieved in object detection algorithms powered by deep learning. In this tracking-by-detection paradigm, object detections are first obtained from object detector applied to video frames, which is considered as a detection step. This is then followed by a tracking step, where state estimation and data association are conducted. The motion prediction of a state estimation [8] predicts the bounding boxes of object tracklets in the next frame, and then the data association is performed between these tracklets and current detections to update the tracklets for generating trajectories of tracked objects over-time. The standard choice of the state estimation method for multi-object visual tracking is Kalman filter (KF) [8].

Data association is very challenging in multi-object visual tracking due to challenges such as miss-detections due to occlusions, appearance changes, or noisy detections. To solve the association task between the predicted tracklet bounding box and the detection bounding box, many works use either one or a combination of strong cues, motion and appearance information, since these cues provide powerful instance-level discrimination. The motion information is usually computed using intersection-over-union (IoU) and its variants [9] or Mahalanobis distance [10]. Normalized Euclidean distance between bounding box centers of detection and tracklet is also used in [7]. The appearance information is usually leveraged from trained deep learning models [11] [12]. The faster trackers such as [1] [4] [13] use only the motion information for the association task. Many trackers use a combination of motion and appearance information [2] [3] [5] [14] [15] [16] [17], generally with better performance but with compromised speed. In addition to the strong cues, some works also incorporate weak cues such as height state, confidence state and/or velocity direction for the association task [18] [19] [13] to compensate for the strong cues, especially in challenging situations such as occluded and crowded scenes. The key gap that is missing in the literature is the comparative study on the effectiveness of the strategies for fusing different cues such as strong cues and/or weak cues. Different trackers use different fusion methods; however, there is no work in the literature, to the best of our knowledge, which investigates and compares the different fusion methods for data association in multi-object visual tracking.

In this work, we investigate different fusion methods used in multi-object visual tracking and evaluate them extensively on different tracking datasets. Our tracker obeys the Simple, Online and Real-Time (SORT) characteristics; hence, we call our tracker FusionSORT. We design our tracker in such a way that we can flexibly use the different strong cues and/or weak cues with different fusion methods for the thorough investigation of our tracker through extensive experiments. Moreover, we elegantly incorporate tracklet confidence state into the state vector representation of the KF. In general, the main contributions of this paper are as follows:

1) We investigate four widely used different fusion methods for associating detections to tracklets in multi-object visual tracking, including minimum, weighted sum based on IoU, KF gating, and hadamard product of costs.

2) In addition to considering strong cues such as motion and appearance information, we also incorporate weak cues such as height-IoU and tracklet confidence information in the data association.
3) We conduct extensive evaluations on validation sets of MOT17, MOT20 and DanceTrack datasets, and demonstrate that the choice of a fusion method is key for data association in multi-object visual tracking.

The rest of this paper is organized as follows. After the discussion of related work in section II, our proposed method is explained in detail including strong and weak cues modelling and fusion methods in section III. The experimental setting and results are analyzed and compared in section IV, followed by the main conclusion in section V.

II. Related Work

We give an overview of related work on tracking-by-detection and data association.

A. Tracking-By-Detection

Tracking-by-detection is the most widely adopted paradigm for multi-object visual tracking in computer vision [1] [2] [3] [4] [5] [6] [7]. This is due to the significant progress achieved in object detection algorithms driven by deep learning. There are two steps in this tracking-by-detection paradigm: detection and tracking. In the detection step, object detection bounding boxes are obtained by applying object detector to video frames. This is then followed by a tracking step, where state estimation and data association are accomplished. Kalman filter (KF) [8] with a constant-velocity model for motion estimation is the commonly used state estimation method for multi-object visual tracking [1] [2] [20] due to its simplicity and efficiency. A Gaussian mixture probability hypothesis density (GM-PHD) filter has also been used in many works for multi-object visual tracking [7] [21] [22] [23]. These trackers have separate detection and tracking components. Recently, several joint trackers [15] [16] [24] [25] have been proposed which jointly train detection and some other components such as motion, embedding and association models. The primary advantage of these joint trackers is their low computational requirements combined with similar performance levels.

B. Data Association

Data association in multi-object visual tracking is highly challenging due to factors like missed detections caused by occlusions, changes in object appearance, and noisy input detections. Identifying the temporally stable properties of objects is crucial to effectively associate predicted tracklet bounding boxes to detection bounding boxes in video frames. Strong cues such as motion and appearance information provide powerful instance-level discrimination. The motion information is usually computed using IoU and its variants [9] or Mahalanobis distance [10]. Normalized Euclidean distance between bounding box centers of detection and tracklet is also used in [7]. The appearance information is usually leveraged from trained deep learning models [11] [12]. Specifically, deep appearance features are extracted from image patches determined by object detection boxes using an additional deep neural network in separate appearance-based trackers [2] [3] [4] [5]. Appearance models can also be trained jointly with object detectors in joint trackers [15] [16] [24] [25]. For the association task, cosine distance of the extracted deep appearance features is computed as appearance distance. Weak cues such as height state, confidence state and/or velocity direction can provide informative clues that help to compensate for the discrimination of strong cues such as motion and appearance information for associating predicted tracklet boxes to new detection boxes [18] [19] [13].

Combining these different sources of information for associating predicted tracket boxes to new detection boxes is very crucial. Though several trackers use only motion information [1] [4] [13] for data association with interesting performance, the performance can be improved by fusing different cues. Minimum fusion method is used in [5], where the minimum in each element of the cost matrices of motion and appearance is used to match tracklets to new detections. Several works [17] [18] [26] use weighted sum, with some variations, of motion and appearance information, in which motion information is computed using IoU. Another fusion method, which we call KF gating, is also based on weighted sum of cues and is used in [15] [16] [3]. However, the Mahalanobis distance is used in this fusion method instead of the IoU, and is subjected to KF gating. Hadamard product is also used in [22] [19] which computes element-wise multiplication of different costs. Hence, different trackers use different fusion methods to fuse different costs for associating tracklets to current detections, which is then solved by Hungarian algorithm [27] as bipartite graph matching. However, there is no work in the literature which thoroughly investigates these different fusion methods. In this work, we investigate four commonly used fusion methods and demonstrate that the choice of a fusion method is key for data association in multi-object visual tracking.

III. Method

In our tracker, we use Kalman filter (KF) [8] with a constant-velocity model for motion estimation of object tracklets in the image plane, similar to the other SORT methods [1] [2] [4] [5]. The cost matrices are computed by measuring the pairwise representation similarity between tracklets and detections for the association task, which is then solved by Hungarian algorithm [27] as bipartite graph matching. For computing the total cost matrix, we consider strong cues such as motion and appearance information as well as weak cues such as height-IoU and confidence information. Furthermore, our tracker incorporates camera-motion compensation (CMC), as used in [5] [28]. We adopted the two stage matching strategy, similar to previous works [4] [5], that conducts the first association using high-confident detections and then the second association using low-confident detections. Appearance information, and hence, the fusion methods are used only at the first association stage. The second association stage matches the low-confident detections to the remaining unassigned tracklets that have not been assigned to high-confident detections in the first association stage. In addition, the second association uses only the motion information, specifically IoU. Only the high-confident detections with scores above a given threshold are used for new track initialization. The low-confident detections are not utilized to start new tracks in order to avoid false-positive tracks that can be introduced from low-confident false positive detections.

For state vector representation, we extend the widely used standard KF in [5] with two additional states: the tracklet confidence (score) c and its velocity component $\dot{c}$, following the state vector derivation approach in [29]. Accordigly, the state vector is represented as in (1)

$$\begin{aligned}\mathbf{x}_k = \quad & [x_c(k), y_c(k), w(k), h(k), c(k), \\ & \dot{x}_c(k), \dot{y}_c(k), \dot{w}(k), \dot{h}(k), \dot{c}(k)]^T\end{aligned} \tag{1}$$

where (x_c, y_c) denote object tracklet box's center, while w, h and c represent the object tracklet box's width, height, and tracklet confidence, respectively. The velocity components are denoted by $\dot{x_c}$, $\dot{y_c}$, $\dot{w}$, $\dot{h}$ and $\dot{c}$. Tracklet box size can change dramatically when predicting inactive (lost) tracks for multiple frames without state update, which hinders re-activation after occlusion. To overcome this, we apply height preservation, setting the derivative $\dot{h}$ to zero before the KF prediction step, similar to [9] [5] [4]. Similarly, we also apply width preservation ($\dot{w} = 0$) and tracklet confidence preservation ($\dot{c} = 0$) before the KF prediction step in our experiments. Note that the tracklet confidence preservation has less impact on some sequences when compared to the others.

Similarly, the measurement vector is represented as in (2)

$$\mathbf{z}_k = [z_{xc}(k), z_{yc}(k), z_w(k), z_h(k), z_c(k)]^T \tag{2}$$

where (z_{xc}, z_{yc}) denote object detection box's center, while z_w, z_h and z_c represent the object detection box's width, height and score, respectively.

Following the extension of the above state and measurement vectors, we also extend the process noise covariance $\mathbf{Q}_k$ and the measurement noise covariance $\mathbf{R}_k$ matrices as in (3) and (4), respectively, which incorporate the tracklet confidence c and its velocity component $\dot{c}$. Following [2] [5], we use time-dependent $\mathbf{Q}_k$ and $\mathbf{R}_k$ which are expressed as functions of some estimated elements and some measurement elements.

$$\begin{aligned}\mathbf{Q}_k = \quad & diag((\sigma_p \hat{w}_{k-1|k-1})^2, (\sigma_p \hat{h}_{k-1|k-1})^2, \\ & (\sigma_p \hat{w}_{k-1|k-1})^2, (\sigma_p \hat{h}_{k-1|k-1})^2, \\ & (\sigma_p \hat{c}_{k-1|k-1})^2, (\sigma_v \hat{w}_{k-1|k-1})^2, \\ & (\sigma_v \hat{h}_{k-1|k-1})^2, (\sigma_v \hat{w}_{k-1|k-1})^2, \\ & (\sigma_v \hat{h}_{k-1|k-1})^2, (\sigma_v \hat{c}_{k-1|k-1})^2)\end{aligned} \tag{3}$$

$$\begin{aligned}\mathbf{R}_k = \quad & diag((\sigma_m \hat{w}_{k|k-1})^2, (\sigma_m \hat{h}_{k|k-1})^2, \\ & (\sigma_m \hat{w}_{k|k-1})^2, (\sigma_m \hat{h}_{k|k-1})^2, (\sigma_m \hat{c}_{k|k-1})^2)\end{aligned} \tag{4}$$

Following the works in [5] [2], we choose the noise factors as $\sigma_p = 0.05$, $\sigma_v = 0.00625$, and $\sigma_m = 0.05$, since our frame rate is also 30 fps. It is worth noting that we modified $\mathbf{Q}_k$ and $\mathbf{R}_k$ according to our slightly modified state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$, respectively. Noise Scale Adaptive (NSA) KF [3], $\mathbf{R}_{NSA} = (1 - z_c)\mathbf{R}$, did not help in our experiments. Ideally, the higher the detection confidence, the smaller the adapted measurement noise covariance $\mathbf{R}_{NSA}$ and the more influence has the detection on the track state update.

A. Strong Cues

The association task in multi-object visual tracking is primarily solved, explicitly or implicitly, by using strong cues such as motion and appearance information since these cues provide powerful instance-level discrimination. In this work, we consider both intersection-over-union (IoU) and Mahalanobis distance [10] as motion information.

Given two boxes as $b^1 = (x_1^1, y_1^1, x_2^1, y_2^1)$ and $b^2 = (x_1^2, y_1^2, x_2^2, y_2^2)$, where x_1 and y_1 represents the top-left corner and x_2 and y_2 represents the bottom-right corner, the conventional IoU based on area can be given as in (5).

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} \tag{5}$$

where B_1 and B_2 are the areas of the boxes b^1 and b^2, respectively.

It is possible to incorporate the uncertainty of the motion estimation into the distance measure since the KF is used as a motion model. Given a probability distribution f on $\mathcal{R}^N$, with mean $\mu = (\mu_1, \mu_2, \mu_3, \ldots, \mu_N)^\mathsf{T}$ and positive semi-definite covariance matrix S, the Mahalanobis distance of a point $\mathbf{z} = (z_1, z_2, z_3, \ldots, z_N)^\mathsf{T}$ from f is generally given as in (6)

$$d_M(\mathbf{z}, f) = \sqrt{(\mathbf{z} - \mu)^\mathsf{T} S^{-1} (\mathbf{z} - \mu)} \tag{6}$$

where $\mathbf{z}$ and μ correspond to measurement (detection) box center position and the projection of the estimated tracklet mean into measurement space, respectively, while f corresponds to a Gaussian predicted state distribution. S^{-1} denotes inverse of the projected tracklet state covariance matrix S into the measurement space. Hence, (μ, S) corresponds to a track state projected into the measurement space. We use the squared Mahalanobis distance $d_M^2(\mathbf{z}, f)$ in our experiments.

For appearance information, we exploited deep appearance representation, particularly using stronger baseline on top of BoT (SBS) [30] with the ResNeSt50 [31] as a backbone, from the FastReID library [11], as used in [5] [14]. For updating the matched tracklet appearance embedding e_i^k for the i-th tracklet at frame k, we use the exponential moving average (EMA) method, similar to [15] [5], as given in (7).

$$e_i^k = \alpha e_i^{k-1} + (1 - \alpha) f_i^k \tag{7}$$

where $\alpha = 0.9$ is a momentum term and f_i^k is the appearance embedding of the current matched detection. The appearance features are extracted only from the high-confident detections i.e. appearance features are used only in the first association step. We compute cosine similarity between the averaged tracklet appearance embedding vector e_i^k and the new detection embedding vector f_i^k to match them.

B. Weak Cues

Though strong cues are the widely used information for associating detections to tracklets in multi-object visual tracking, they suffer from degradation under challenging situations such as occluded and crowded scenes [18]. Hence, we employ weak cues such as confidence and height state to compensate for the strong cues. For incorporating height information, we compute height intersection-over-union (hIoU) given tracklet and detection bounding boxes.

Accordingly, given the two boxes as b^1 and b^2 above, the height-IoU (hIoU) can be computed as in (8)

$$hIoU = \frac{min(y_2^1, y_2^2) - max(y_1^1, y_1^2)}{max(y_2^1, y_2^2) - min(y_1^1, y_1^2)} \tag{8}$$

We compute the confidence cost as the absolute difference between the estimated tracklet confidence c_{trk} and detection confidence c_{det}, as given in (9).

$$C_c = |c_{trk} - c_{det}| \tag{9}$$

where the tracklet confidence c_{trk} is given in (1) as $c(k)$, and c_{det} is the detection score obtained from object detector applied to a video frame. Refer to (17) for explicit formulation of confidence cost.

C. Fusion Methods

The four fusion methods that we use in our investigative experimental analysis are described as follows. Note that the fusion methods are applied only for the first association step; the second association step uses only IoU for all the experiments.

1) Minimum: In this fusion method, we use the *minimum* in each element of the cost matrices of motion, appearance, height-IoU and confidence as the final value of the cost matrix C. IoU is used for computing the motion cost. We extend the minimum fusion method designed for motion and appearance costs in [5] to include the height-IoU and confidence costs as in (10), (11), (12) and (13).

$$\hat{d}_{i,j}^{cos} = \begin{cases} 0.5 \ . \ d_{i,j}^{cos} (d_{i,j}^{cos} < \theta_{emb}) \wedge (d_{i,j}^{iou} < \theta_{iou}) \\ 1, \ \text{otherwise} \end{cases} \tag{10}$$

$$\hat{d}_{i,j}^{hiou} = \begin{cases} d_{i,j}^{hiou} \wedge (d_{i,j}^{iou} < \theta_{iou}) \\ 1, \ \text{otherwise} \end{cases} \tag{11}$$

$$\hat{d}_{i,j}^{conf} = \begin{cases} d_{i,j}^{conf} \wedge (d_{i,j}^{iou} < \theta_{iou}) \\ 1, \ \text{otherwise} \end{cases} \tag{12}$$

$$C_{i,j} = min(\hat{d}_{i,j}^{cos}, d_{i,j}^{iou}, \hat{d}_{i,j}^{hiou}, \hat{d}_{i,j}^{conf}) \tag{13}$$

where $C_{i,j}$ denotes the (i, j) element of the total cost matrix C. $d_{i,j}^{cos}$, $d_{i,j}^{iou}$, $d_{i,j}^{hiou}$ and $d_{i,j}^{conf}$ are the cosine distance, IoU distance, height-IoU distance and confidence distance between i-th tracklet and j-th detection, respectively. $\wedge$ denotes logical 'and'. The minimum fusion is represented by *min* in (13). Note that cosine distance is 1 minus cosine similarity, and the other distances generally follow similar manner, and they are explicitly formulated as in (14), (15), (16), and (17).

$$d_{i,j}^{cos} = 1 - \frac{e_i^k . f_j^k}{\|e_i^k\|_2 \|f_j^k\|_2} \tag{14}$$

where $\| * \|_2$ is the 2-norm of its argument *, and . represents the dot product of tracklet averaged appearance embedding e_i^k and detection appearance embedding f_j^k. Note that $d_{i,j}^{cos} \in [0, 2]$.

$$d_{i,j}^{iou} = 1 - IoU_{i,j} \tag{15}$$

$d_{i,j}^{iou} \in [0, 1]$ is IoU distance, also called Jaccard distance.

$$d_{i,j}^{hiou} = 1 - hIoU_{i,j} \tag{16}$$

$$d_{i,j}^{conf} = |c_i^{trk} - c_j^{det}| \tag{17}$$

where $| * |$ is the absolute value of *. c_i^{trk} and c_j^{det} denote tracklet confidence and detection score, respectively. Note that $d_{i,j}^{hiou} \in [0, 1]$ and $d_{i,j}^{conf} \in [0, 1]$.

Accordingly, cost matrices C_a, C_m, C_h and C_c are constructed from $d_{i,j}^{cos}$, $d_{i,j}^{iou}$, $d_{i,j}^{hiou}$ and $d_{i,j}^{conf}$ to represent the appearance, motion, height-IoU and confidence costs, respectively. The IoU threshold θ_{iou} and appearance threshold θ_{emb} are set to 0.5 and 0.25, respectively. These thresholds are used to discard low cosine similarity or far away candidates in terms of IoU's value.

2) Weighted Sum: Given the appearance cost C_a, motion cost C_m, height-IoU cost C_h and confidence cost C_c, this method uses weighted sum for calculating the total cost matrix C as in (18). In this method, the motion cost is computed using IoU.

$$C = \lambda_1 C_m + \lambda_2 C_a + \lambda_3 C_h + \lambda_4 C_c \tag{18}$$

where the λ_1, λ_2, λ_3 and λ_4 are the weights for motion, appearance, height-IoU and confidence costs, respectively. We used the appearance threshold similar to (10); however, we did not apply the IoU threshold mask for the C_h and C_c costs as in (11) (12) since it slightly decreases the results. We set λ_1, λ_3 and λ_4 to 1.0, 0.1 and 0.1, respectively, for all of the datasets. λ_2 is set to 0.1 for MOT17 and MOT20 datasets and 0.2 for DanceTrack dataset.

3) KF Gating: Given the appearance cost C_a, motion cost C_m, height-IoU cost C_h and confidence cost C_c, this method also uses weighted sum for calculating the total cost matrix C as in (19). However, the motion cost is computed using Mahalanobis distance, rather than IoU, and it is subjected to KF gating [8] (preventing unlikely assignments) where the gating distance is computed between KF predicted state distribution and measurements (detections). A suitable Mahalanobis distance threshold can be obtained from a table for the 0.95 quantile of the chi-square distribution with N degrees of freedom. In our method, the chi-square distribution has 2 degrees of freedom since the distance computation is done with respect to the bounding box center position only i.e. (x_c, y_c).

$$C = \lambda(C_a + \lambda_h C_h + \lambda_c C_c) + (1 - \lambda)C_m \qquad (19)$$

where the weight factor λ is set to 0.98, and the λ_h and λ_c are the weights for the height-IoU and confidence costs, respectively. Both λ_h and λ_c are set to 0.2.

Even though the Mahalanobis distance is used for the first association step where fusion of different costs takes place, the IoU is used for the second association step for all the fusion methods including this fusion method to make a fair comparison. In our experiments, the IoU gives better overall performance than the Mahalanobis distance for the second association step, as shown in Table IV. Note that we did not apply any thresholding and 0.5 multiplication as in (10), (11), and (12) in this fusion method.

4) Hadamard Product: In this fusion method, the total cost matrix C is obtained by element-wise multiplication of all costs: appearance cost C_a, motion cost C_m, height-IoU cost C_h and confidence cost C_c, as in (20). In this method, the motion cost is computed using IoU.

$$C = C_a \odot C_m \odot C_h \odot C_c \qquad (20)$$

We used the appearance and IoU thresholds in a similar manner as in (10), (11) and (12), which improve the results.

IV. Experiments

A. Experimental Setting

1) Datasets: We conduct our investigative experiments on different multi-object visual tracking benchmarks, including MOT17 [32], MOT20 [33] and DanceTrack [34], which were captured under diverse scenarios. MOT17 was captured using both static and moving cameras and consists of seven train and seven test sequences, in which the motion is mostly linear. MOT20 consists of highly crowded four train and four test sequences which are used to evaluate trackers under dense objects and severe occlusions. DanceTrack stands out as one of the most demanding tracking benchmarks, characterized by a variety of non-linear motion patterns, frequent interactions, and significant occlusions. It contains 40, 25 and 35 videos of dancing humans for training, validation and testing. The MOT17 and MOT20 validation sets follow a widely adopted convention [35] [4] [5] [14] [18] where the train set is split into halves for training and validation since these datasets do not have a separate validation set i.e. they have only train and test sets. The DanceTrack has a separate validation set, in addition to train and test sets, which we use directly. Hence, our experimental analysis is based on the validation sets of these benchmarks.

2) Evaluation Metrics: We use different evaluation metrics for comparing tracking performance based on the different fusion methods, including Multiple Object Tracking Accuracy (MOTA) [36], Identification F1 (IDF1) [37] and Higher-Order Tracking Accuracy (HOTA) [38]. MOTA mainly focuses on evaluating the detection performance while IDF1 evaluates the identity association performance of a tracker. HOTA combines several sub-metrics that evaluate the tracker from different perspectives, providing a comprehensive assessment of the tracker performance, including detection, association, and localization into a single unified metric.

3) Implementation Details: Our visual tracking algorithm is implemented using Python and PyTorch deep learning framework, and run on Laptop with Intel(R) Core(TM) i7-10850H @ 2.70GHz, 16 GB RAM and NVIDIA GeForce RTX 2070 GPU. We use the publicly available YOLOX-X detector [39], trained by [4] for MOT17 and MOT20 datasets, and trained by [18] for DanceTrack dataset. For feature extraction, we used the publicly available models trained by [5] for MOT17 and MOT20 datasets and trained by [18] for DanceTrack dataset, based on FastReID library [11]. We use the same tracker parameters throughout our experimental analysis, which were mostly set empirically. Unless otherwise specified, we set high detection score threshold τ_1 to 0.6, which is used to separate high-confident detections from low-confident detections for the first association step. We set low detection score threshold τ_2 to 0.1 for the second association step. Detections with score lower than τ_2 are discarded. In the linear assignment step, the matching is rejected if the detection and the tracklet similarity is smaller than 0.2 for the first association step and smaller than 0.5 for the second association step. The detection score needs to be at least 0.7 to be considered for track initialization. The lost tracks (inactive tracks) are kept for 30 frames in case they appear again before they get deleted. Note that we do not output the boxes and identities of lost tracks, as in [4] [5]. In order to precisely investigate the effectiveness of the fusion methods, we do not apply any tracklet interpolation as a post-processing in our experiments.

B. Experimental Results

We compare FusionSORT on the validation set of MOT17, MOT20 and DanceTrack using different fusion methods in Table I, Table II and Table III, respectively.

1) MOT17: Different fusion methods are compared on validation set of MOT17 in Table I. As shown in this table, the highest values of HOTA, MOTA and IDF1 are obtained using minimum, weighted sum based on IoU and KF gating, respectively. Fusing appearance information to motion information generally improves performance when using minimum, weighted sum based on IoU and KF gating methods. However, the performance degrades when using hadamard fusion method since the hadamard fusion method treats both motion and appearance information with equal importance implicitly, where the contribution of the motion information is higher in this case. The incorporation of weak cues such as height-IoU and confidence information negatively affects the tracking performance when using the minimum fusion method. This happens because the minimum fusion method does not fully exploit the potential of all the cues since one of them is used for the association task, where the weak cues are expected to contribute minimally when compared to the strong cues. Similarly, significant performance decline is observed when fusing the weak cues with the strong cues using the hadamard fusion method since it implicitly treats all cues equally. The incorporation of weak cues improves the tracking performance when using the KF gating fusion method. Note that all the fusion methods use IoU for computing motion distance except the KF gating method which uses Mahalanobis distance.

TABLE I
EVALUATION ON MOT17 VALIDATION DATASET. MOTION DISTANCE (MOT) IS COMPUTED USING MAHALANOBIS DISTANCE FOR KF GATING FUSION METHOD AND USING IOU FOR THE OTHER METHODS. THE FIRST AND SECOND HIGHEST VALUES ARE HIGHLIGHTED BY RED AND BLUE, RESPECTIVELY.

Minimum	**MOT17**		
	MOTA	***IDF1***	***HOTA***
mot (iou)	78.421	81.999	69.345
mot, app	78.541	82.238	69.419
mot, app, hiou	78.209	80.937	68.586
mot, app, hiou, confidence	78.079	79.556	66.896
Weighted-sum			
mot (iou)	78.421	81.999	69.345
mot, app	78.584	81.877	69.379
mot, app, hiou	78.541	81.524	69.246
mot, app, hiou, confidence	78.432	81.15	68.812
KF-gating			
mot (mahalanobis)	76.749	69.256	60.581
mot, app	78.035	81.75	68.889
mot, app, hiou	78.072	82.249	69.29
mot, app, hiou, confidence	78.003	82.322	69.366
Hadamard			
mot (iou)	78.421	81.999	69.345
mot, app	78.311	80.718	68.38
mot, app, hiou	78.282	80.415	68.172
mot, app, hiou, confidence	78.261	80.045	67.882

2) MOT20: As can be seen in Table II, all fusion methods improve the tracking performance when we combine the appearance information with the motion information. Incorporating the weak cues such as height-IoU and confidence information improve the performance when using the weighted sum based on IoU and KF gating fusion methods. The highest values of HOTA, MOTA and IDF1 are obtained with the KF gating fusion method by combining both strong cues such as motion and appearance information and weak cues such as height-IoU and confidence information. However, combining weak cues with the strong cues degrades the tracking performance when using minimum and hadamard fusion methods.

3) DanceTrack: The comparison of different fusion methods on validation set of DanceTrack is given in Table III. As can be seen in this table, the minimum fusion method has outstanding overall performance, where the highest values of HOTA and IDF1 are obtained. The highest value of MOTA is obtained using the hadamard fusion method. The overall performance of the KF gating fusion method is lower on the DanceTrack dataset. In general, the fusion of the appearance information with the motion information improves the tracking performance when using all the fusion methods. However, integrating the weak clues decreases the performance when using the minimum and hadamard fusion methods, similar to on MOT17 and MOT20 datasets. Incorporating height-IoU decreases the tracking performance when using the weighted sum based on IoU and KF gating fusion methods; however, incorporating confidence information improves the performance, as shown in Table III.

TABLE II
EVALUATION ON MOT20 VALIDATION DATASET. MOTION DISTANCE (MOT) IS COMPUTED USING MAHALANOBIS DISTANCE FOR KF GATING FUSION METHOD AND USING IOU FOR THE OTHER METHODS. THE FIRST AND SECOND HIGHEST VALUES ARE HIGHLIGHTED BY RED AND BLUE, RESPECTIVELY.

Minimum	**MOT20**		
	MOTA	***IDF1***	***HOTA***
mot (iou)	72.722	73.794	57.815
mot, app	72.714	74.232	58.216
mot, app, hiou	72.701	73.653	57.69
mot, app, hiou, confidence	72.442	70.809	55.764
Weighted-sum			
mot (iou)	72.722	73.794	57.815
mot, app	72.682	74.505	58.262
mot, app, hiou	72.674	74.536	58.275
mot, app, hiou, confidence	72.706	74.626	58.368
KF-gating			
mot (mahalanobis)	71.932	60.222	48.78
mot, app	73.052	74.502	58.236
mot, app, hiou	72.993	74.724	58.418
mot, app, hiou, confidence	72.938	75.047	58.695
Hadamard			
mot (iou)	72.722	73.794	57.815
mot, app	72.747	74.155	58.091
mot, app, hiou	72.724	73.396	57.547
mot, app, hiou, confidence	72.7	73.271	57.432

V. CONCLUSION

In this paper, we investigate four commonly used different fusion methods for associating detections to tracklets in multi-object visual tracking by considering strong cues such as motion and appearance information as well as weak cues such as height intersection-over-union (height-IoU) and tracklet confidence information. The fusion methods include minimum, weighted sum based on IoU, Kalman filter (KF) gating, and hadamard product of costs due to the different cues. For computing confident cost, we elegantly incorporate tracklet confidence state into the state vector representation of the KF. Through extensive experiments on validation sets of MOT17, MOT20 and DanceTrack datasets, we find out that the different fusion methods have their own pros and cons. The minimum fusion method works reasonably well when fusing motion and appearance information; however, its performance decreases when incorporating weak cues. The incorporation of weak cues also degrades the tracking performance when using the hadamard fusion method. The tracking performance increases when using weak cues along with strong cues when using weighted sum and KF gating fusion methods though the performance of the KF gating fusion method is generally

TABLE III

Evaluation on DanceTrack validation dataset. Motion distance (mot) is computed using Mahalanobis distance for KF gating fusion method and using IoU for the other methods. The first and second highest values are highlighted by red and blue, respectively.

	DanceTrack		
Minimum	***MOTA***	***IDF1***	***HOTA***
mot (iou)	88.068	53.838	52.412
mot, app	88.168	60.105	58.474
mot, app, hiou	88.06	57.04	56.016
mot, app, hiou, confidence	87.922	53.538	54.47
Weighted-sum			
mot (iou)	88.068	53.838	52.412
mot, app	88.126	56.45	55.026
mot, app, hiou	88.118	54.393	54.066
mot, app, hiou, confidence	88.172	55.664	55.70
KF-gating			
mot (mahalanobis)	83.795	33.186	35.807
mot, app	86.419	47.098	49.08
mot, app, hiou	86.696	45.857	48.021
mot, app, hiou, confidence	86.716	46.393	48.695
Hadamard			
mot (iou)	88.068	53.838	52.412
mot, app	88.228	56.039	54.938
mot, app, hiou	88.106	53.107	52.237
mot, app, hiou, confidence	88.038	52.996	52.07

TABLE IV

Evaluation on MOT17 validation dataset for comparison of IoU and Mahalanobis distance for the second association of the KF gating fusion method. The first and second highest values are highlighted by red and blue, respectively.

	MOT17		
IoU	***MOTA***	***IDF1***	***HOTA***
mot	76.749	69.256	60.581
mot, app	78.035	81.75	68.889
mot, app, hiou	78.072	82.249	69.29
mot, app, hiou, confidence	78.003	82.322	69.366
Mahalanobis			
mot	76.708	68.453	60.158
mot, app	78.189	81.145	68.708
mot, app, hiou	78.115	81.613	69.02
mot, app, hiou, confidence	78.128	81.567	69.101

lower on DanceTrack dataset. Hence, the weighted sum based on IoU is more favourable when using weak cues along with the strong cues. We hope that this investigative work helps the computer vision research community to use the right fusion method with given cues for data association in multi-object visual tracking.

References

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[2] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[3] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, 2023.

[4] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," 2022.

[5] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," 2022. [Online]. Available: https://arxiv.org/abs/2206.14651

[6] J. Seidenschwarz, G. Braso, V. C. Serrano, I. Elezi, and L. Leal-Taixe, "Simple cues lead to a strong multi-object tracker," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, March 2023.

[7] N. L. Baisa, "Occlusion-robust online multi-object visual tracking using a GM-PHD filter with CNN-based re-identification," *Journal of Visual Communication and Image Representation*, vol. 80, p. 103279, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320321001814

[8] G. Welch and G. Bishop, "An introduction to the kalman filter," 2006.

[9] D. Stadler, "A detailed study of the association task in tracking-by-detection-based multi-person tracking," in *Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. Ed.: J. Beyerer*. KIT Scientific Publishing, 2023, pp. 59–85.

[10] R. De Maesschalck, D. Jouan-Rimbaud, and D. Massart, "The mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169743999000477

[11] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "FastReID: A pytorch toolbox for general instance re-identification," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 9664–9667. [Online]. Available: https://doi.org/10.1145/3581783.3613460

[12] N. L. Baisa, "Local-aware global attention network for person re-identification based on body and hand images," *Journal of Visual Communication and Image Representation*, vol. 103, p. 104207, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320324001639

[13] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9686–9696.

[14] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep-OC-SORT: Multi-pedestrian tracking by adaptive re-identification," *arXiv preprint arXiv:2302.11813*, 2023.

[15] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 107–122.

[16] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vision*, vol. 129, no. 11, p. 3069–3087, Nov. 2021. [Online]. Available: https://doi.org/10.1007/s11263-021-01513-4

[17] D. Stadler and J. Beyerer, "An improved association pipeline for multi-person tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023, pp. 3170–3179.

[18] M. Yang, G. Han, B. Yan, W. Zhang, J. Qi, H. Lu, and D. Wang, "Hybrid-SORT: Weak cues matter for online multi-object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 6504–6512.

[19] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple object tracking with high performance detection and appearance feature," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 36–42.

[20] C. Long, A. Haizhou, Z. Zijie, and S. Chong, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *ICME*, 2018.

[21] N. L. Baisa, "Online multi-object visual tracking using a GM-PHD filter with deep appearance learning," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–8.

[22] Z. Fu, F. Angelini, S. M. Naqvi, and J. A. Chambers, "GM-PHD filter based online multiple human tracking using deep discriminative correlation matching," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4299–4303.

[23] N. L. Baisa and A. Wallace, "Development of a N-type GM-PHD filter for multiple target, multiple type visual tracking," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 257–271, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320319300343

[24] H. Ren, S. Han, H. Ding, Z. Zhang, H. Wang, and F. Wang, "Focus on details: Online multi-object tracking with diverse fine-grained representation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 11 289–11 298.

[25] S. You, H. Yao, B.-K. Bao, and C. Xu, "UTM: A unified multiple object tracking model with identity-aware feature enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 21 876–21 886.

[26] H. Hashempoor, R. Koikara, and Y. D. Hwang, "FeatureSORT: Essential features for effective tracking," 2024. [Online]. Available: https://arxiv.org/abs/2407.04249

[27] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[28] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 941–951.

[29] N. L. Baisa, "Derivation of a constant velocity motion model for visual tracking," 2020. [Online]. Available: https://arxiv.org/abs/2005.00844

[30] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1487–1495.

[31] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "ResNeSt: Split-attention networks," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 2735–2745.

[32] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016. [Online]. Available: https://arxiv.org/abs/1603.00831

[33] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," 2020. [Online]. Available: https://arxiv.org/abs/2003.09003

[34] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, "DanceTrack: Multi-object tracking in uniform appearance and diverse motion," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 20 961–20 970.

[35] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 474–490.

[36] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, Jan. 2008. [Online]. Available: https://doi.org/10.1155/2008/246309

[37] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 17–35.

[38] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vision*, vol. 129, no. 2, p. 548–578, Feb. 2021. [Online]. Available: https://doi.org/10.1007/s11263-020-01375-2

[39] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021. [Online]. Available: https://arxiv.org/abs/2107.08430