Online multi-object tracking by detection based on generative appearance models

Dorra Riahi[a,*], Guillaume-Alexandre Bilodeau[a]

[a] *LITIV lab., Department of Computer and Software Engineering,*
Polytechnique Montréal,
P.O. Box 6079, Station Centre-ville, Montréal
(Québec), Canada, H3C 3A7

Abstract

This paper presents a robust online multiple object tracking (MOT) approach based on multiple features. Our approach is able to handle MOT problems, like long-term and heavy occlusions and close similarity between target appearance models. The proposed MOT algorithm is based on the concept of multi-feature fusion. It selects the best position of the tracked target by using a robust appearance model representation. The appearance model of a target is built with a color model, a sparse appearance model, a motion model and a spatial information model. In order to select the optimal candidate (detection response) of the target, we calculate a linear affinity function that integrates similarity scores coming from each feature. In our MOT system, we formulate the problem as a data association problem between a set of detections and a set of targets according to their joint probability values. The proposed method has been evaluated on public video sequences. Compared with the state-of-the-art, we demonstrate that our MOT framework achieves competitive results and is capable of handling several challenging problems.

Keywords: Multiple object tracking, Data association, Tracking by detection, Sparse appearance model, Multiple features.

[*] Corresponding author
Email addresses: `dorra.riahi@polymtl.ca` (Dorra Riahi), `gabilodeau@polymtl.ca` (Guillaume-Alexandre Bilodeau)

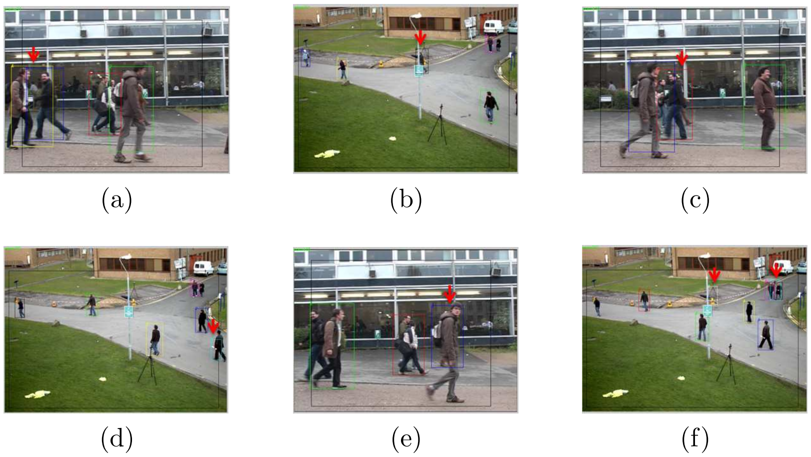
1. Introduction

Multiple object tracking (MOT) is used for many computer vision applications, such as robotics, video surveillance and activity recognition. Despite a steady increase in research focusing on MOT systems, it is still a challenging unsolved problem. Tracking an object is the task of predicting the target path during its presence in the field of view of a camera while multiple object tracking is the task of tracking a target and separating it from other similar objects to be tracked.

In order to perform the MOT task, several problems have to be addressed. In the recent years, MOT operates on detection responses coming from an object detector, typically a person detector. While this approach is less flexible than MOT based on background subtraction, it has the advantage of avoiding to have to deal with the fragmentation problem. The focus is thus more on the data association problem. Still many problems have to be solved.

One of the MOT problems comes from false detection responses where the target is not detected at all times (see figure 1 (a)-(c)). It depends on the quality of the technique used to extract detection responses. Another problem is related to occlusion. In crowded environments, we can find occlusion between similar targets (for example two persons), occlusion between a target and a fix object (for example an object from the background) and total occlusion where the target is totally invisible (see figure 1 (d)-(f)). In addition, the similarity of the appearance model of the targets can present a big challenge for MOT. Targets can have similar appearance, have similar movement and have the same size (see figure 1(a) person in green bounding box and person in yellow bounding box). The last MOT problem comes from the unknown number of targets, that is, the number of targets can change widely over time. A robust MOT is a tracking approach that can better handle the problems stated above by improving the detection responses, the appearance model of the target and the data association between targets and detection responses.

In this paper, we propose an online multi-object tracking in a multi-feature framework that addresses the aforementioned difficulties. MOT algorithms can be classified into two categories: online (or streaming) MOT and offline (or batch) MOT. Offline MOT uses information from past and future frames to predict the current position of targets while online MOT only uses information from past frames. Our proposed approach is an online



(a) (b) (c)

(d) (e) (f)

Figure 1: Typical situations showing MOT problems: (a),(b),(c) Occlusions indicated by the red arrow, and (d),(e),(f) False alarm and poorly localized detections indicated also by red arrows

MOT. We address the tracking of people using a person detector. However, our method can be applied to any pre-trained detector outputs. Our algorithm capitalizes on the strength of using multiple cues to build the appearance model of the target. This work demonstrates that an efficient way to ameliorate the performance of a MOT system is to use a robust target representation in addition to a good data association technique. This is justified by the fact that appearance modeling is a crucial component for associating targets and detections because the observation model can be highly dynamic and the complex interactions between similar targets may cause ambiguities.

A MOT process relies on two main components: the target appearance model and the data association strategy to select the best candidate for each target. These components are not trivial to design because it necessitates answers to many questions: How to decide what is the best candidate? When should we interpret a target as being occluded? Is the target partially or totally invisible? This requires an efficient representation of the target model, which is a priori unknown. The contributions of our work relate to both aspects: the appearance model of the target and the data association strategy. For the target representation, the appearance model is built using multiple cues coming from independent and complementary features: color histogram, motion histogram, sparse model and spatial information. A robust target representation is obtained that allows distinguishing targets from each other. Regarding the data association strategy, we adopted the Hungarian algorithm to associate detection responses with the set of targets frame-per-frame. Furthermore, to handle particular cases (like occlusion between targets, unknown number of the targets, etc.), we filter the associations (delete incorrect associations and add new associations) between the list of targets and the list of detection responses according to their state (occluded, active or hypothesized target). This way, we can manage the data association in order to select the best candidate (detection response) for the appropriate target. The main contributions of this paper are:

1. a novel MOT method that combines the strengths of many successful appearance models, namely sparse appearance model and locality sensitive histograms;

2. a data association between targets and candidates that is scored by an affinity function that fuses multiple cues coming from independent features;

3. an interpolation process for the target position that is based on spatial information. Thus, a target can be tracked even it is not detected or it is invisible for some time. The online interpolation of the position of the lost target is based on the history of movement of the target;

4. experimental results demonstrating that the proposed approach is applicable to a variety of tracking scenarios and that our approach outperforms several recent MOT approaches.

The rest of the paper is organized as follows. Section 2 reviews the state-of-the-art approach for MOT. Section 3 describes in detail the proposed approach. In section 4, we present experimental results for our MOT algorithm. Section 5 provides the main conclusions of our work.

2. Related works

As discussed previously, a MOT system can be improved either by improving the detection responses, the data association strategy or the appearance model of the target. Progress has been done recently on all these aspects.

Detection responses. To avoid the problems related to background subtraction (cluttered background, dynamic backgrounds, etc.), many works use an object detector outputs for their MOT system. In fact, if the task is to track one kind of object (like human, cars, etc.), it is more suitable to use an object detector, as the problem of object fragmentation is avoided. Some recent works use model-free single object trackers with an object detector to ameliorate the detection response outputs. In [1], authors use a particle filter tracker combined with a vote-based confidence map of an object detector. They use the detector as a confidence score. Breitenstein et al. [2] follow a tracking by detection approach for their MOT algorithm. The authors use particle filter outputs along with person detector outputs to handle occlusions and missing detections. The object detector is used in two ways: as a confidence score term through probabilistic votes for matching (ISM detectors) and to locate the targets (HOG detector). In a similar spirit, authors in [3] exploit a MOT framework based on combining tracking and detection. The tracker and the object detector are used as two independent identities and their outputs are integrated in the data association phase. In contrast to other tracking-by-detection approaches, this approach [3] works on results of both an object detector and multiple basic trackers. In [4], authors develop a MOT algorithm that uses object detection to supervise single object trackers. A Bayesian filtering based single object tracker is applied to every frame to predict the current position of the target. A human detector with high precision is associate with a person tracker based on their similarity score. The similarity score is calculated by combining multiple cues (color, shape, and texture) to build the observation models. However, the cues are human specific and focus on the upper part of the human body (face and torso). To get optimal maximizing assignments, authors use the Hungarian algorithm. If a detection is assigned to an existing trajectory, this detection will be used to update the corresponding trajectory. Else, a new trajectory will be initialized.

Data Association. In MOT systems, an additional challenge arises: it is the data association. In fact, it is the answer for the question of which detection should be assign to which target. Each detection response must be assigned to a target or discarded as a false alarm or added as a new target. In general, classical data association approaches are used like the Joint Probabilistic Data Association Filter (JPDAF) [5] and Multiple Hypotheses Tracking (MHT) [6]. They jointly consider all possible associations between targets and detection responses. Alternatively, the Hungarian algorithm [7] [4] and the greedy search algorithm can be used to recursively select the best assignment between a set of targets and the set of detections. More recently, tracking by tracklets approaches were exploited [8] [9] [10] [11]. This technique reframes data association process as a set of local trajectory fragments. For example, in [12], the authors propose a Latent Data Association approach where each detection is considered as its own target. So, the data association is re-formulated as a single Switching Linear Dynamical System (SLDS), i.e. linking these single detections (single targets) into longer trajectories. Yang et al. [11] introduced an online learning approach with a CRF model for tracking by tracklets approach. They add discriminative features to differentiate corresponding pairs of local tracklets. The CRF model is learned in each sliding window repeatedly. Each tracklet should be associated with one and only one tracklet. In other work done by Huang et al. [13], the data association between local tracklets is done in a hierarchical framework on three levels. In the first level, only single detection responses are matched. In the second level, short tracklets are combined to form longer tracklets. At the high level, occluded tracklets are re-assigned to handle the occlusion problem. In [10], authors proposed a MOT system by linking tracklets into long trajectories by finding a joint optimal assignment between global information (linking tracklets) and local information (linking detection responses). Trajectories are updated iteratively until convergence.

The work of [14] also exploit the notion of tracklets to achieve the data association step. They incorporate the benefit of person recognition to associate local tracklets. In fact, tracklets are classified into two categories: query tracklets and gallery tracklets. First of all, tracklets are generated by matching short trajectories of the targets (linking detection responses between two consecutives frames). After that, the tracklets are classified. A gallery tracklet is a tracklet which is longer than a threshold and is not covered by any other tracklet. In fact, the more a trajectory is long the more it is reliable. A query tracklet is a tracklet who is missing some feature of the target. The association of tracklets is based on three similarity scores: the motion, the time (as a step function) and the appearance where the motion cue is defined based on time gap between tracklets (the tail of the first tracklet and the head of the second one), the geometric position and the velocities of the tracklet.

Another work is proposed in [15] in which the data association is achieved in different levels: global data association (matching between trajectory), tensor approximation representation via a power iteration solution, optimization framework using context information (motion information). The data association step models the interaction energy between multiples and individual trajectories in an optimization framework using contextual information until convergence. The contextual information is based on two kind of motion descriptors. First, the low-level motion context (specific motion context) is generated based on the non-maximum suppression strategy (NMS). By estimating the motion consistency value (using the orientation similarity and the speed similarity between any two associated trajectories), the interaction between a pair of association is modeled. Second, the high-level motion context which is divided into two types: the motion interaction between association and tracklet (based on the average motion interaction between an association and neighboring tracklets) and the motion interaction between two associations (based on the temporal average of motion similarities between a pair of associations). The calculation of the low-level and the high-level motion context used the spatial displacement velocity vector (defined by the difference between spatial position). Their approach is similar to tracking by tracklets. The only difference is that the data association is done only between two tracklets in a short term (neighboring tracklets). So, it will have difficulty in handling the variation of the number of targets (exit and entry target).

In Fabio et al. work [16], a generic MOT method is proposed that is performed directly on confidence map. The confidence map is a representation of likely detection locations. In fact, a modified particle filter algorithm is applied on the confidence map. Besides the geometric position, the velocity and the intensity of the target, a target identity is integrated in the particle state. The ID state allows the approach to deal with unknown number of targets. The IDs assignment is performed using a Mean-Shift clustering supported by a GMM to obtain a robust matching of target identities within each cluster. To handle the ID mixing (specially in case of close targets), the ambiguity between targets IDs is resolved using an MRF (a Markov Random Field) of target birth and target death. Different to other approaches, the data association in [17] step is formulated into a minimisation problem. In fact, an energy function is estimated for each trajectory of targets. Then this energy function should be minimised to obtain a long trajectory (by linking smaller ones). Initially, authors use a Kalman Filter tracker to obtain initial trackers and then a greedy search based data association is applied to obtain initial trajectories. Thereafter, the minimization of the energy function is solved by executed different moving jump namely growing and shrinking of trajectories by adding some target location on the current trajectory or by weeding out incorrect targets from trajectories, merging (if the energy function of two paths is lower than the energy function of each one separately) and splitting (split a path in two smaller paths if the energy function of each path is lower than the original one), adding (if a detection is not assigned to an existing path, a new path should be created) and removing (a path is full deleted if its minimum energy function is above a threshold). The assignment step is not described in the paper but it is done indirectly using the appearance model and occlusion reasoning.

Appearance model. The appearance model of a target is the representation used to describe a region of interest. The appearance model can be based on target shape, color [18], motion properties [11] [19] and geometric properties [20]. Furthermore, the appearance model can be based on multiple features combined together. For example, in [21], for single object tracking, the appearance model is build using colour histogram and orientation histogram in a particle filter framework. In [20], the authors proposed a MOT algorithm dedicated to sport video sequences. The player appearance model is defined by a statistical and dynamical model (the position, the scale, the velocity and the optical flow). In Possegger et al. [22], they exploit geometric properties to create the appearance model of the target to handle the occlusion problems. They integrate the spatio-temporal evolution of occlusion regions, motion prediction and object detector reliability. Their work proved that geometric properties can help to handle occlusion between targets. In [8], the authors use three independent features to model each target which are color histograms, covariance matrices and histograms of gradients (HOG).

In [4], authors use multi-cues to build the appearance model but in a different manner. The model is highly specialized. Different appearance models are used to represent a particular part of the human body. The kernel-weighted color histogram is calculated for the head and the upper of torso region. The histogram consists of 8 bins for each color canal (R, G or B). To be robust to occlusions, two histograms are used to compare the dissimilarity: the first one is the last histogram of the target and the second one is the mean histogram of the target (created based on the average of the few latest histograms). The Bhattacharyya coefficient is applied to compare histograms. Besides, the head region is represented by an elliptical model. The intensity gradients vectors and the gradients are estimated for the ellipse (K= 36 normal vectors). The dissimilarity is then based on calculating the angle θ_k between the largest gradient and the k-th normal vector as:

$$1 - \frac{1}{K}\sum_{k=1}^{K} |cos(\theta_k))| \tag{1}$$

The last feature is a bag of local features that is extract on the upper part of torso region to capture the textural characteristics of this part. The features used are fast dense SIFT-like features on each grid (defined by 4×4 pixels). A local features based histogram is estimated on 256 clusters for each region. As the color histogram, the Bhattacharyya coefficient is used to measure the difference between histograms. Then a dissimilarity function is calculated as a linear and weighted combination of the dissimilarity functions of each cue. Although the appearance model is specific for each part of the upper region of the human body, it is difficult to build it. Indeed, the extraction of the head region and the upper part of the torso requires advanced strategies. This explain the fact that authors use a multi-view human head detector based on CNN (Convolutional Neural Network). However, it is not obvious to obtain the head region of the target (for example, in the case that the head of the person is occluded but the rest of the body of the person is visible in the video sequence) because this part of body is very likely to be occluded because it is small compared to the rest of the body. This MOT approach can be applied only for human tracking and for some special datasets. In contrast, the approach that we are proposing aims at describing the complete region of the object for better robustness to occlusion. Furthermore, we aim at proposing an appearance model that can be applied to a variety of objects.

Authors in [14] uses multiple cues to learn the appearance model. The used cues are the colour (RGB color histogram with 8 bins for each color canal), shape (HOG histogram) and texture (covariance matrices). A single descriptor is calculated for each support region via one feature. In fact, the person image is divided into a set of rectangles (654) respecting the constraints of the width and height ratio. So, the appearance descriptors are generated for each person image patches to calculate the similarity between targets. To compare the histograms, belonging to targets, the correlation coefficient is used. The final similarity function is a linear combination of each similarity measurement for each descriptor (where each descriptor has a weight which reflects its importance). Those descriptors are then trained using the standard Adaboost algorithm to sequentially select the best descriptor (the descriptor which gives the best comparison of the similarity). Indeed, the training data are collected by using the ground-truth of a dataset. A positive sample is defined by a pair of sample images belonging to the target and a negative sample is defined by a pair of sample images belonging to different targets. The similarity scores for positives and negatives samples are integrated into a standard Adaboost algorithm to learn the pool of features for different regions. According to [14], the color histogram descriptor on smaller regions is the most often selected while the covariance matrices are the least selected. The learning of the best descriptors is a kind of off-line learning. Thereby, the appearance model of the target requires prior knowledge of the structure of the target model.

The notion of multi-cues has a different use in the work of [23]. This work is based on fragTrack algorithm where each part of the objects is modeled separately. Each object fragment is represented by a cue. So, a multi-cue based approach is used to model multiple fragments for the object.

In [17], authors propose an energy function (or cost function) that offers a more complete representation of the target. In fact, authors give a robust representation for the target trajectory instead of representing directly the target. The energy function is calculated using: data term which allows to keep the trajectories close to the observations (obtained by estimated the localisation of the target relative to the detection localisation using an isotropic shaped function), dynamic term (a target motion constraint estimated by a constant velocity model), mutual exclusion term to avoid the case in which two targets come too close to each other (a penalty function is calculated based on the targets's volume intersections), trajectory persistence term (help to avoiding track fragmentation or abrupt track termination problems by using a sigmoid centered on the border of the tracking region) and a regularizing term to prevent the number of targets from growing (is calculated using the length of a trajectory and the number of targets). Besides those terms, the appearance model of the target is also added to calculate the energy function. An RGB color histogram with 16 bins is estimated on the Gaussian weighted region of the detection (to favor center pixels and delete the pixels along region borders). The construction of the appearance model of the trajectories requires the intrinsic and extrinsic camera parameters. In fact, besides the image coordinates, the target is defined by its real world coordinates.

The motion feature is widely used to build the appearance model. In [24], the motion model is the motion relation between two targets calculated using the position and the velocity difference. In other word, the relative motion model is a set of linked edges between different objects (including self-motion model for an object). To estimate the similarity score, a posterior probability is calculated bases on the relative motion models and their weights (calculated using event probabilities and observations). It is estimated with a Bayesian filter. Besides the relative motion models, the data association is achieved using the size similarity (ratio of the difference between the width and the height) and the appearance similarity (color histogram).

The approaches described above improve tracking performance in different ways, but can be quite complex because of using an object tracker (for tracking by using a model-free visual tracker) and a graph structure. In this work, we argue that creating a robust appearance model should be first addressed. In fact, for the data association step, the appearance model is used as input to estimate the affinity function for each target to be tracked. Also, to be robust to appearance model changes (like illumination and scale change), an update of the appearance model should be achieved.

By taking inspiration from previous work, we aim to improve MOT based on the three aspects described above. First, we follow a tracking by detection strategy. Secondly, we build a robust appearance model that combines intrinsic properties (color histogram and sparse representation) and motion properties (optical flow and geometric position). Finally, for simplicity, the optimal single-frame assignment is obtained by the Hungarian algorithm. A filtering step is done to handle association problems (the loss of the target, reappearance of the target, the exiting of the target and the entering of a new target in the scene) by deleting or adding some associations. For the false alarm detection, we can use the motion appearance model to interpolate the lost position of such target. After improving the appearance model, a target management step is achieved to alleviate the inter-occlusion (occlusion of targets with a fixed object in the scene) and intra-occlusion (occlusion of the current target with other targets) problem.

3. Proposed method

3.1. Motivation and overview

Our MOT method has the four steps outlined in figure 2. An object to be tracked is an ROI (region of interest) defined by a bounding box (rectangle) inside a frame. The set of target features is initialized with the features estimated on the detection responses in the first frame. The detection responses are found in each frame with a pre-trained person detector. In order to decrease the number of false detections, we filter the set of detection responses by removing those with inappropriate sizes or with lower classification confidence values. A set of a known number of tracks is initially built in which each target is defined by a state (see section 3.3.3) and a set of features. The set of targets will be updated dynamically to reflect appearance model changes and to handle MOT problems (as discuss in section 1). In addition to a color and a sparse representation model of the target, we also propose a motion model that includes optical flow feature and spatial feature. The motion model allows us to avoid false associations (or assignments) between targets and detection responses. For each frame, an affinity function is calculated which reflects the similarity between a target and a set of current candidates (a candidate is a detection response) based on their appearance model. More specifically, the appearance model of a target is defined by four features:

1. A color histogram H_c is used to encode the color information of the target. The Euclidean distance between histograms is used to evaluate the color similarity between targets and candidates.

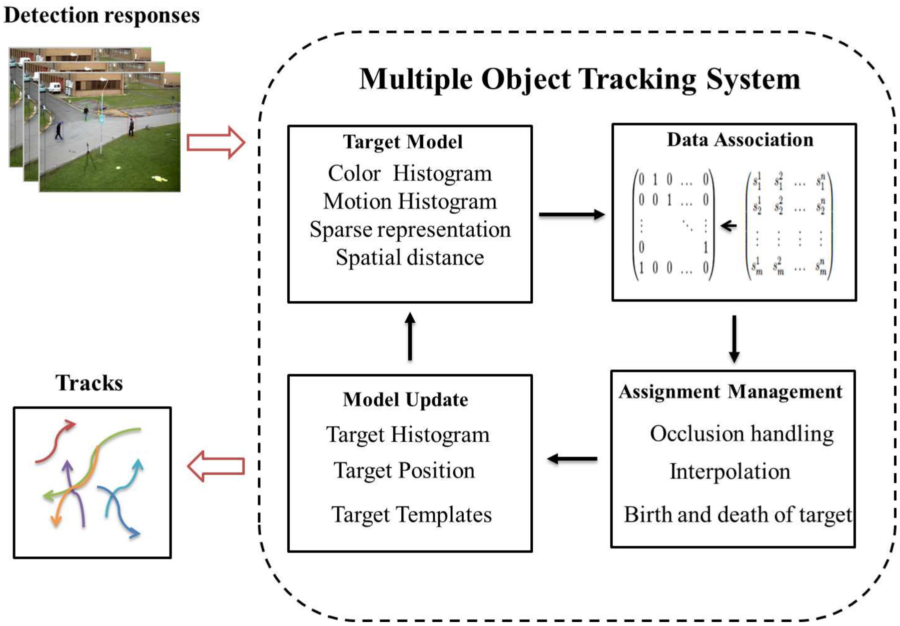


Figure 2: Method overview

2. A sparse representation error p reflects the projection error of the candidate in a template space of the target. In fact, each candidate is sparsely and linearly projected into target templates, which are linearly generated from the last bounding box of the target.

3. A histogram of oriented optical flow H_m is used to encode the motion properties of the target.

4. The spatial consistency $\vec{d}$ reflects the geometric correlation between target and the list of candidates in term of Euclidean distance between the target center point and the center point for each candidate.

The data association is a crucial task in our MOT framework. It is the task of associating existing targets (or trajectories) to different candidates (detection responses). Instead of doing the association in one step, the data association will be achieved in two steps or at two levels. In fact, we have two principal categories for the state of a target: occluded or active (visible). Active targets are matched in priority before occluded targets because we cannot know if an occluded target will be visible at that time or not. Data association of occluded targets is more uncertain. Therefore, fully visible targets will be assigned first. In other words, the data association is done in two hierarchical levels: active level and occluded level. All visible targets are assigned at the active level with all detection responses and the rest (occluded targets) are assigned at the occluded level with the not yet assigned set of detection responses later on. Then, all valid assignments between targets and detections are combined to achieve the global data association step. A global assignment matrix is then obtained. The assignment matrix is composed of 1 or 0 values: 0 if the assignment is not valid (a target is not matched with a detection) and 1 if the assignment is valid (a target is matched with a detection response). To handle occlusion problems, the assignment matrix should be filtered which means that if an assignment is not reliable, it should be deleted and if an assignment is reliable, it should be kept. This is achieved by creating a state for each target. Then, based on the state of the targets and the similarity score value, an assignment can be deleted or added. Data association is achieved by applying the Hungarian algorithm [22].

3.2. Multi-features based model

A target is represented by four independent descriptors that reflect the intrinsic properties (color and sparse appearance model) and the motion properties (optical flow and spatial feature). Each feature describes an object by considering different properties. In fact, the color reflects the distribution of the intensity value of the object, the sparse model reflects the linear combination of the intensity of the object into other intensity templates, the optical flow is the differential of the intensity values for the object and finally the spatial feature reflects the geometric characteristics. Although the color, sparse and the optical flow features are based on the color characteristics for their computation, we still consider them independent because they measure different properties of color (respectively, the color distribution, the organization of the color in a template, and color differential). Also, they are independent in the term of their decision. For example, if two objects have similar color feature, they will not necessary have similar motion feature or be sparsely projected with the same templates.

These descriptors are used together to define the similarity of the appearance model. Thus, we obtain a powerful discrimination of all tracked targets. We build a global appearance model F^t at each time t

$$F^t = [H_c, p, H_m, \vec{d}] \tag{2}$$

where H_c is the concatenation of locality sensitive histograms at each pixel, p is the probability error of the sparse projection, H_m is the oriented optical flow histogram and $\vec{d}$ is the vector of Euclidean distances between target and candidate center points.

3.2.1. Color appearance model

The color histogram is built at each pixel location of the bounding box of the target. We use a recent approach of histogram representation called locality sensitive histogram (LSH) [25]. As defined, the LSH is a set of local histograms at each pixel location. For object tracking application, target pixels inside a local neighborhood should not have an equal contribution. Pixels further away from the center should be weighted less than pixels closer to the target center. The LSH is the sum of weighted intensity values around a neighborhood region. Mathematically, let H_{px}^{E} the locality sensitive histogram at pixel px inside a neighborhood region E:

$$H_{px}^{E} = \sum_{q=1}^{px} \alpha^{|px-q|}.Q(I_q, b), b = 1, \cdots, B, \tag{3}$$

Where $\alpha \in [0, 1]$ is a parameter controlling the weight of pixel and $Q(I_q, b)$ is equal to zero except when intensity value I_q belongs to bin b. The LSH can be calculated based on the contribution of pixels from the left side (pixels on the left of pixel px) and the right side (pixels on the right of pixel px). So, the LSH can be written as:

$$H_{px}^{E}(b) = H_{px}^{E,left}(b) + H_{px}^{E,right}(b) - Q(I_{px}, b), \tag{4}$$

Where:

$$H_{px}^{E,left}(b) = Q(I_{px}, b) + \alpha.H_{px-1}^{E,left}(b), \tag{5}$$

$$H_{px}^{E,rigth}(b) = Q(I_{px}, b) + \alpha.H_{px+1}^{E,rigth}(b), \tag{6}$$

Pixels from the right side do not contribute to calculate $H_{px}^{E,left}$ and pixels from the left side do note contribute to calculate $H_{px}^{E,rigth}$. The LSH is then normalized at each pixel location. The normalization factor n_{px} at pixel px is:

$$n_{px} = \sum_{q=1}^{px} \alpha^{|px-q|} \tag{7}$$

The distance between two locality sensitive histograms can be computed as:

$$D(H_t, H_c) = \sum_{b=1}^{B}(|H_t(b) - H_c(b)|), \tag{8}$$

Where H_t is the target histogram and H_c is the candidate histogram.

3.2.2. Sparse representation model

Sparse appearance models have attracted a lot of attention in recent years. We adopted and modified the sparse representation technique developed in [26] to fit into our MOT framework. The sparse representation model aims at calculating the projection errors of the candidate model into the dictionary of target templates. The candidate is represented as a linear combination of the template set of the target. A target template dictionary is constructed by a set of templates generated by doing small translations around the target bounding box. There are two types of templates: main target templates and trivial templates (containing trivial pixels such as pixels from the background). A good target candidate is a candidate that can be efficiently represented by only the target templates, while, a bad target candidate is represented by a dense representation (represented by the use of many trivial templates), which reflects the dissimilarity to target template. In our sparse representation model, we sparsely projected the detection responses in a template space of the target. A vector of approximate errors of the sparse representation projections is then obtained. It reflects the similarity between the target sparse model and the candidate (detection response) model. Given the set of n target templates $T = \{t_1, t_2, ..., t_n\} \in \mathbb{R}^{d \times n}$, a candidate y is linearly projected into the target templates:

$$y = \vec{a}T = a_1 t_1 + a_2 t_2 + ... + a_n t_n, \tag{9}$$

Where $\vec{a} = (a_1, a_2, ..., a_n)' \in \mathbb{R}^n$ is the coefficient vector. To incorporate the effect of occlusion and noise on the target model, each candidate is represented by trivial templates in addition to the target templates. Trivial template is a matrix of zeros in which each row has only one nonzero entry. Then, equation (8) can be rewritten as:

$$y = \vec{a}T + \vec{e}I, \tag{10}$$

Where $I = \{i_1, i_2, ..., i_d\} \in \mathbb{R}^{d \times d}$ is a set of d trivial templates and $\vec{e} = (e_1, e_2, ..., e_d)' \in \mathbb{R}^d$ is the trivial coefficient vector. Note that the number of trivial templates is much larger than the number of target templates ($d >> n$) In sparse representation model, we can say that templates are positively related to the target depending to the sign of the coefficient in the vector $\vec{e}$. So, the nonnegativity constraint is taken into consideration by adding two kinds of trivial templates: negative and positive trivial templates. Consequently, equation (9) is rewritten as:

$$y = \vec{c}B, \tag{11}$$

Where $B = [T, I, -I] \in \mathbb{R}^{d\times(n+2d)}$ and $\vec{c} = [a, e^{+}, e^{-}]' \in \mathbb{R}^{(n+2d)}$.

Each candidate is then sparsely represented according to equation (10). The similarity between a target x and a candidate y is transform to a l_1 minimization problem :

$$min \left\| Bc - y_i \right\|_2^2 + \lambda \left\| c \right\|_1 ; s.t.\ c \geqslant 0 \tag{12}$$

Where $||.||_2$ and $||.||_1$ denote the l_2 and the l_1 norm used to solve the minimization problem and λ is a factor. The likelihood probability $p(y_i|x_t)$ between candidate sparse model y_i and target sparse model x_t at time t is then :

$$p(y_i|x_t) = \frac{1}{\tau}\exp[-\alpha \left\| y_i - cT \right\|_2^2], \tag{13}$$

Where c is the solution of equation (11), α is a constant, and τ is a normalization factor. A good candidate is a candidate that is approximated with small coefficients for the trivial templates and a bad candidate is a candidate for which the vector of coefficients is densely populated and the main approximation is done with trivial templates. The candidate with smallest projection error will have higher likelihood probability. An updating step for the target model is necessary to take into account local variation of the model (illumination, scale and pose changes). This is done by updating the template space according to the new bounding box of the target. If the tracking result is good, then a new set of template space will be generated from the target bounding box.

3.2.3. Motion appearance model

We propose to represent each target by its motion feature. We use the optical flow [27] to calculate this feature. To obtain the motion descriptor, we calculate the histogram of oriented optical flow (HOOF) [28]. First of all, the optical flow is calculated for each target bounding box. The calculation of the optical flow vector is done by solving a differential equation that describes the differential of intensity values at each pixel. So an optical flow vector $\vec{v} = [v_x, v_y]$ is obtained on each dimension (row and column). Then, each vector is binned according to its primary angle $\theta = \tan^1(\frac{v_y}{v_x})$ and weighted according to its magnitude $\sqrt{v_x^2 + v_y^2}$. The histogram of oriented optical flow is then normalized to be robust to scale variations. To use the HOOF histogram for computing candidates and target similarity, we compare the HOOF histograms with the following equation:

$$D(H_t^m, H_c^m) = \sum_{b=1}^{B} (|H_t^m(b) - H_c^m(b)|), \tag{14}$$

Where H_t^m is the target motion histogram and H_c^m is the candidate motion histogram.

3.2.4. Spatial model

The spatial information of a target enhances the study of the correlation of targets position over time. The spatial constraint is used in two steps of our algorithm: features extraction and data association steps to allow exploring the spatial relationships of a target with each candidate. The spatial information is used to avoid incorrect assignment with a far candidate and to observe the dynamic of each target. We encode the spatial information as geometric coordinates (i_x, i_y, w, h) of a target over time where (i_x, i_y) are the coordinate of the target, (w, h) are the width and the height of the target. The spatial similarity likelihood $\vec{d}$ is then the vector of Euclidean distances between center points of target and candidates:

$$d_i(j) = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}, \tag{15}$$

where (i_x, i_y) and (j_x, j_y) are the center coordinates of a target i and a candidate j respectively. Note that the spatial proximity is taken into account in the estimation of target and candidates similarity only in the case where there is no occlusion (the target is visible).

3.3. Data Association

The MOT problem is formulated as a data association problem. The data association is the step for finding the answer to the question: which detection should be assigned to which target. This step aims at matching the set of targets with the set of current candidates in order to define the current bounding box (the current position) of each target. The matching is done based on an affinity matrix (see section 3.3.2). Note that one target should be assigned to one and only one detection response. We follow a hierarchical matching process: step 1, matching only visible targets and step 2, matching only occluded targets (see algorithm 1). In order to handle occlusion and update the set of targets (adding new targets or deleting existing targets), a management step is done after the global data association.

3.3.1. Affinity function

To obtain a global similarity value, features are fused according to their weight. The global similarity map is thus created at time t to represent the target similarity considering all the features. Let $X^t = \{x_1^t, x_2^t, ..., x_n^t\}$ be the set of all tracked targets at time t and $Y^t = \{y_1^t, y_2^t, ..., y_m^t\}$ be the set of all detection responses at time t. The associated feature set $S = [s_1, s_2, s_3, s_4]$ combines affinity function measures from the different features, that is the color histogram, the sparse feature, the optical flow feature and the spatial feature. More precisely:

s_1 is the difference between color histograms (LSH) for each object (target and detection).

s_2 is the probability of the error of the sparse linear projection for the target model into the detection response templates.

s_3 is the difference between HOOF histograms (optical flow based histogram) for each object (target and detection).

s_4 is the spatial difference between the target position and the detection position in term of Euclidian distance.

The affinity function at frame t is then written as:

$$f_t(x_i^t, y_j^t) = \sum_k \alpha_k s_k(x_i^t, y_j^t), \tag{16}$$

where α_k denotes a weight for each feature and s_k represents the affinity function using the feature number k between the target state x_i^t and the detection response y_j^t. The weights α_k reflect the contribution of each feature to determine the similarity between targets and detection responses. They

Algorithm 1 Data association algorithm

```
- Compute the affinity function $f_t(x_i^t, y_j^t)$ for active targets and candidates
- Compute the assignment matrix by applying the Hungarian algorithm
for all valid assignments do
  if $f_t(x_i^t, y_j^t) > threshold$ then
    - Delete assignment
  end if
end for
- Compute the affinity function $f_t(x_i^t, y_j^t)$ for occluded targets and unassigned candidates
- Compute the assignment matrix by applying the Hungarian algorithm
for all valid assignments do
  if $f_t(x_i^t, y_j^t) > threshold$ then
    - Delete assignment
  end if
end for
if active target is not assigned then
  - target is set as occluded
end if
if occluded target is assigned then
  - target is set as active
end if
if candidate is not assigned and candidate is not in the in/out region then
  - candidate is set as hypothesized
end if
if candidate is not assigned and candidate is in the in/out region then
  - candidate is set as entering
end if
if candidate is not assigned and candidate stays is in the in/out region for more than $f$ frames then
  - candidate is set as exiting
end if
```

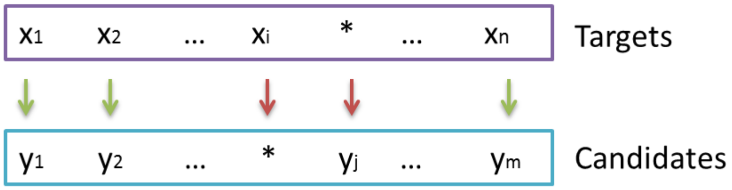


Figure 3: Targets Assignments

were calculated experimentally and are constant for all the tested videos. They are: 0.4 for color feature, 0.3 for sparse model feature, 0.1 for the optical flow feature and 0.2 for the geometric feature.

3.3.2. Hungarian algorithm

The optimal frame-by-frame assignment is achieved by using the Hungarian algorithm. The Hungarian algorithm finds the assignments that maximize the affinity function. First, an affinity matrix A_t at time t for each pair (x_i^t, y_j^t) is computed. $f_t(x_i^t, y_j^t)$ is the value in row number i and column number j. Then, the pair (x_*, y_*) with maximum score is iteratively selected for each row. An assignment matrix is then obtained. It contains 0 and 1 only for the selected matching pair. Only one selected pair per row.

3.3.3. Assignment management

Due to the variable number of targets over time, heavy occlusion between tracked targets and unreliable detection responses, MOT cannot be resolved by only a matching task. Thus, we exploit extra processing steps to handle such MOT problems. The challenging task is when a target is not assigned or a candidate is not labeled (see figure 3).

Target states. In addition to the geometric coordinate, the identifier and the set of features, a target can be defined also by its state. A state is used to distinguish visible targets from invisible ones, and new targets from exiting ones. Thus, a target can be:

1. *Active*. An active target is a visible target.
2. *Occluded*. An occluded target is a lost target caused by partial or total occlusion or false detection.
3. *Exiting*. An exiting target is a target that is temporarily out from the field of view of the camera.

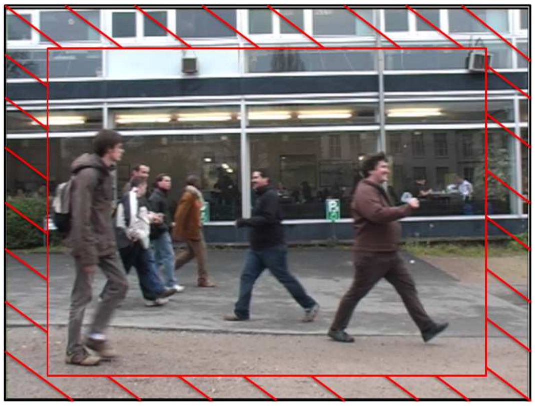


Figure 4: In/Out region

4. *Entering*. An entering target is a new target added to the set of current targets.

5. *Hypothesized*. A hypothesized target is a candidate that is not assigned. It can be a new target appearing in the middle of a frame, a false detection or an existing target that is already deleted.

Entering and exiting of targets is determined based on an in/out region. The in/out region is selected manually along frame borders in the first frame (see the hatched area in figure 4). If a candidate is detected inside the in/out area, it will be added to the set of targets as a new track in the entering state. If an existing target stays in this area for more than a given number of frames, the target will be deleted from the current set of tracks and it will be marked as exiting. Therefore, the number of targets changes over time because of the process of birth of target (adding a new track) and the death of target (deletion of an existing track). To handle occlusion, a target can be labeled as occluded or active. In the case of unassigned target, this target is marked as occluded. An occluded target can be set as active target only if it is assigned with a low similarity score (its affinity function exceeds a threshold).

Interpolation of lost targets Until now, the data association step is done between the set of detection responses and the set of current targets. It means that if a currently tracked target is not detected at time t, it will not be assigned (it will be set as occluded). To handle the

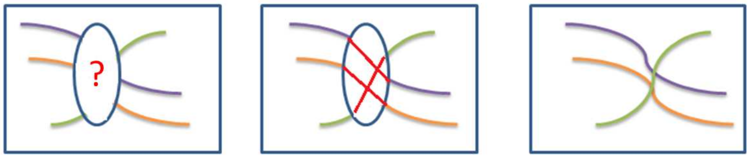


Figure 5: Interpolation step. First column: incomplete targets trajectories during the occlusion. Second column: Estimation of targets movements. Third column: complete target trajectories

problem of false detection, we propose to interpolate the lost position of the target. The interpolation is achieved based on the history of motion between two states of the target: occluded target and active target (see fig 5). First, the motion vector of the lost target is estimated based on the history of movement of the target over time. Let us consider a given target x_i^t at time t, x_i^t is occluded since t_{occ} time and it is set as active at the current frame t_{cur}. Assuming that the targets move with a linear constant motion, the motion vector between two consecutive times is:

$$\vec{dep}(t_1, t_2) = |(\vec{v}(t_1) - \vec{v}(t_2))/(t_1 - t_2)| \,, \tag{17}$$

Where $\vec{v}$ is the coordinate vector $[x, y]$ of the target at time t and $t_1, t_2 \in [t_{occ}, t_{cur} - 1]$. Then, the lost position (during the occlusion time) is estimated as:

$$pos_t(x_i) = pos_{t-1}(x_i) + \mu_{dep} \tag{18}$$

Where μ_{dep} is the mean value of $\vec{dep}$ during occlusion.

3.4. Model update

The appearance model changes during time because of many factors: scale change, pose change, illumination variation, etc. Thus, an update step is necessary. The update is done only when a good tracking is achieved. A good tracking is at a time when the matching score (the affinity function) exceed a threshold τ_{maj}. For the set of current targets, each feature is updated according to the new predicted position of the target.

4. Experiments

In this section, we present how our tracking framework helps to improve MOT performance.

Sequence	# frames	Persons	Resolution
TUD-CAMPUS	71	Up to 6	640x480
TUD-CROSSING	201	Up to 8	640x480
TUD-STADTMITTE	179	Up to 8	640x480
PETS2009-S2-L1	795	Up to 10	768x576

Table 1: Video sequence details

4.1. Experimental setup

We validated our proposed method on a variety of challenging video sequences: TUD Campus, TUD Crossing, TUD Stadtmitte and PETS2009 S2-L1 [29]. They are commonly used video sequences and they are very challenging for several reasons. First, they show walking pedestrians in an outdoor environment so lighting conditions are not controlled. Second, due to large field of view, people get very small when they are far from the camera making their tracking more challenging (PETS2009 video). Then, in TUD dataset, targets have a similar size and they walk with similar speeds. However, targets are frequently occluding each other (heavy inter-object occlusion) and are occluded by static objects. To obtain the detections, we use the detections originally provided with the videos [29]. For each detection response, the classification confidence term is provided. Video sequence details are given in table 1.

4.1.1. Evaluation metrics

Tracking performance is evaluated with the widely used CLEAR MOT metrics [30]. They return an accuracy score called (MOTA) that combines false positive, missed targets and identity switch errors, and a precision score called (MOTP) that is the average distance between ground truth and predicted target positions. In addition, the CLEAR MOT metrics includes: false negatives (FN), false positives (FP) and the number of identity switches (ID Sw).

4.1.2. Runtime

The proposed algorithm was implemented using Matlab language on an Intel Core i7 PC running at 3 GHz and with a 16 GB memory. Our code was no optimized. The speed of the implemented system depends on two major factors: the number and the size of detections and targets. A comparison of the speed computation time is shown in table 2. Note that the results given in table 2 represent the mean runtime for different datasets. For less crowded

Method	Proposed	[Breintenstein, 2011]	[Milan, 2014]	[Yoon, 2015]	[Poiesi, 2013]	[Kuo, 2010]
Runtime (s/f)	6.47	0.5	1	0.2	3	0.25

Table 2: Comparison of runtime performance.

video sequence like TUD-Campus, the runtime is about $5.5(sec/frame)$. In fact, the people appear near the camera so we have detections with large size. For crowded video sequence $PETS2009-S2L1$, the runtime is about $7.45(sec/frame)$. The most time consuming part of our approach is the construction of the appearance model, especially the LSH histogram.

4.1.3. The compared MOT algorithms

We evaluate our MOT approach by a comparison to recent state-of-the-art algorithms. Among the compared approaches, a first category studied MOT with the aim of improving detection responses using model-free tracker [2] [29], a second category aimed to ameliorate the data association technique [31] [12] [32], and a third category aimed to improve the appearance model [33] [34] [35]. The results, when available, are obtained from the authors' papers.

4.2. Experimental results

4.2.1. overall performance

Results are shown in table 3. In general, for all the performance metrics, our proposed approach outperforms other object trackers by achieving up to 84% of MOTA. Our MOTA are often higher than in the previous results. On PETS2009-S2-L1, TUD-Campus and TUD-Crossing, our algorithm outperforms the tracking by detection method of Breitenstein et al. [2] that uses outputs from particle filter trackers and HOG detector. This shows that using a robust appearance model allows to achieve better results than using a model-free tracker combined with a detector. On the other hand, on TUD-Campus and TUD-Crossing, we perform better than Riahi et al. method [35] which is based on improving the appearance model. This shows that besides a robust appearance model, a good strategy for assignments should be integrated. Our method also outperforms the tracking system proposed by [36]. On TUD-Stadtmitte and PETS2009-S2-L1, we achieved better MOTA than Segal et al. [12] MOT algorithm which uses an advanced technique to solve the data association task. It is possible to observe that our MOTA is higher than Gustavo et al. approach [34] by around 14% even if they use

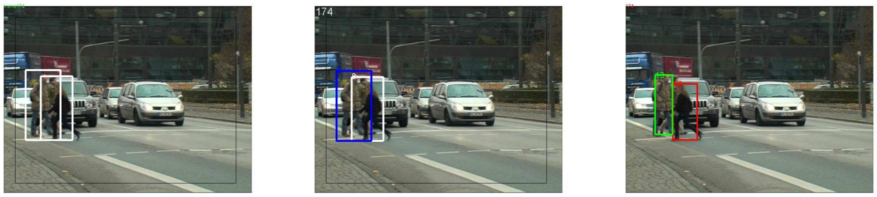


Figure 6: Detection responses, result, and ground-truth, respectively for frame 174 video TUD-CROSSING

multiple patches in their appearance model. Furthermore, we perform better than Yang et al. method [33] which includes background subtraction to handle occlusion. The MOT approach proposed in [17] tends to have more accuracy and precision compared to those of the compared approaches (include our MOT algorithm). This is natural because authors use a different and better set of detection responses. In fact, authors use linear SVM detector based on histograms of oriented gradients (HOG) and histograms of relative optic flow (HOF). Besides, our approach is applied on uncalibrated camera videos sequence while the proposed approach of [17] uses the camera parameters (intrinsic and extrinsic cameras parameters) to build the appearance model. Compared with [37], despite that MOT is performed using a discrete-continuous optimization based data association scheme, our MOTA is about 67% while their MOTA is about 61% on TUD-STADTMITTE video sequence. In [38], Sherrah et al. proposes a part based appearance model which represents the head and the whole body of a person. Our approach outperforms this approach on PETS2009 dataset. Regarding the precision value (MOTP), the performance is comparable to others methods. MOTP is limited by the precision of the detector. In the literature various detectors are used. Some better than others. In our case, we used the detections provided with the datasets, which are not necessarily the best. In fact, the value of MOTP depends on the distance between the predict object and the position of the object in the ground-truth. As we can see in figure 6, the predicted results are correct according to the detections responses but is not correct compared to the ground-truth. In this case, we obtain a lower value of true positive which is proportional to the MOTP value.

The results presented in table 3 emphasis the fact that the use of a robust appearance model with a simple technique of detection or data association can achieved better results. The robustness of our appearance model is coming from the use of sparse representation model in addition to other independent features.

4.2.2. Robustness of the appearance model

To fully evaluate the robustness of the proposed appearance model, we present the performance of each component. To this end, we evaluated all possible combinations of features on two video sequences: PETS2009-S2-L1 and TUD-Crossing. Table 4 and table 5 show the performance for each feature combination. When using all feature terms, the accuracy is the highest while the precision of the tracking remains about the same. When relying only on the motion feature, the MOT fails regularly, especially in the case of heavy and frequent occlusions (PETS2009-S2-L1). This is because the motion feature plays the role of distinguishing between motion directions of targets, not between target similarity. In fact, the motion feature can characterize an object and differentiate it from others objects only if it has a different motion appearance. In our case, we have many similar objects (pedestrians) who move with the same speed and in the same direction. So, many persons have similar motion feature. This why the motion feature is not as discriminative as other appearance models. It mostly allows us to distinguish people walking in different directions. However, in combination with other features, the motion direction often helps in removing assignment ambiguities. The false negative value is the smallest when using only color feature on TUD-Crossing but it is the smallest when using all features on PETS2009-S2-L1. This is explained by the fact that color feature can perform well depending on the number of targets and the level of difficulty of the occlusion. It can be seen that any combination performs better than using only one feature, like the combination of the color and the sparse features gives higher accuracy than using color or sparse feature only. In addition, the combination of sparse and motion features gives more accuracy than sparse or motion feature used alone.

4.2.3. Qualitative performance

Figure 7 depicts an example of the results of our approach on several videos, namely PETS2009-S2-L1, TUD-Stadtmitte, TUD-Crossing, TUD-Campus. We can see that our algorithm can handle heavy occlusion between people in cases of crowded scenes.

Dataset	Method	MOTA	MOTP	FN	FP	IDS
TUD-CAMPUS	Proposed	**78.18%**	*69%*	0%	13%	0
	[Riahi, 2014]	72%	**74%**	25 %	2%	1
	[Breitenstein, 2011]	*73%*	67%	26%	0.1%	2
TUD-CROSSING	Proposed	*78%*	66%	1%	8%	7
	[Riahi, 2014]	72%	**76%**	26%	1%	7
	[Breitenstein, 2011]	**84%**	*71%*	14%	1%	2
	[Andriyenko, 2011]	63%	75.5%	-	-	-
	[Pirsiavash, 2011]	63.3%	76.3%	-	-	-
	[Tang, 2014]	70.7%	77.1%	-	-	-
	[Segal, 2013]	74%	76%	-	-	-
TUD-STADTMITTE	Proposed	*67%*	57.26%	26%	6%	22
	[Andriyenko, 2011]	60.5%	*66%*	-	-	7
	[Milan, 2013]	56.2%	62%	-	-	15
	[Segal, 2013]	63%	**73%**	-	-	-
	[Milan, 2014]	**71%**	65.5%	-	-	4
	[Andriyenko, 2012]	61.8%	63.2%	-	-	4
PETS2009-S2-L1	Proposed	*84%*	66%	13%	2%	35
	[Yang, 2009]	76%	54%	-	-	-
	[Breitenstein, 2011]	80%	56%	-	-	-
	[Andriyenko, 2011]	80%	*76%*	-	-	15
	[Berclaz, 2006]	60%	66%	-	-	-
	[Fuhr, 2014]	70%	-	-	-	-
	[Milan, 2014]	**90%**	**80%**	-	-	11
	[Sherrah, 2013]	81.3%	74.4%	-	-	-
	[Bae, 2014]	80.34%	69.72%	-	-	3
	[Bae, 2014]	83%	69.59%	-	-	4

Table 3: Comparison of results on TUD and PETS2009 dataset. Best method in **red** and second best in *blue*

Features	MOTA	MOTP	FN	FP	IDS	Recall	Precision
All Features	***84%***	***66%***	***13%***	2%	***34***	***87%***	98%
Color Feature	76%	66%	21%	3%	34	78%	97%
Sparse Feature	45%	66%	40%	12%	130	57%	83%
Motion Feature	0%	65%	38%	46%	1178	37%	45%
Color + Motion	76%	66%	18%	5%	48	81%	94%
Color + Sparse	79%	***66%***	20%	***1%***	39	80%	***99%***
Sparse + Motion	62%	66%	17%	17%	166	79%	82%

Table 4: Results evaluation on each feature component of our approach for Pets2009-S2-L1. Best results are in ***red***

Features	MOTA	MOTP	FN	FP	IDS	Recall	Precision
All Features	***78%***	66%	15%	***2%***	45	81%	***97%***
Color Feature	73%	66%	***13%***	12%	22	***85%***	88%
Sparse Feature	43%	66%	50%	5%	24	75%	91%
Motion Feature	1%	***66%***	35%	42%	214	43 %	50 %
Color + Motion	68 %	66%	17%	12%	29	80%	87%
Color + Sparse	76%	66%	17%	5.98%	***11***	82%	***93%***
Sparse + Motion	68%	66%	23%	7%	20	75%	91%

Table 5: Results evaluation on each feature component of our approach for TUD-CROSSING. Best results are in ***red***

PETS2009-S2-L1. This video sequence contains especially challenging problems. First, targets are totally occluded by the traffic sign (see figure 10, first row) which influences on their appearance model. Second, some targets are suddenly stopping for a long time or moving in circle. As we can see in the figure (see figure 10 row 1), target with $id = 1$ stops for more than 100 frames. Our algorithm robustly handles the above problems by the increased power of our appearance model (using a unique fused appearance model) and our update strategy.

TUD-Dataset. For the three videos sequences of TUD-Dataset, most targets have the same size, the same cloths and they walk at similar speeds and in parallel directions. In these cases, our approach can handle assignment ambiguities by the management of the data association. In fact, a wrong assignment between targets and candidates will be deleted according to the descriptors similarity.

We present many scenarios to show how our approach is able to handle such difficult cases. To handle the problem of the missing detections, we follow an interpolation approach in which we can estimated the current position of the target even it is not detected. For example, in figure 8, the target (with the green bounding box) is not assigned by only applying the data association. But, after the interpolation step, we can observe that the green target is interpolated with success. In addition, our approach is able to keep good identity during multiple occlusions (see figure 9) and when the targets are much closer to each other (see figure 7 in row 4). Other scenario (see figure 10) shows that the identity of targets is not affected by the length of the occlusion. As we can see, the target with the red bounding box is successfully assigned during an occlusion of more than 100 frames. Finally, even with appearance model changes either by the scale changes (see figure 11) or the pose changes (see figure 12), our MOT can still identify the targets.

4.2.4. Sensitivity to the number of false detections

The results given in table 6 show that if we use the ground-truth as a set of detection responses, our method gives very high values of Clear MOT: 100% of accuracy and 100% of precision. Obtaining around 100% of accuracy for all tested datasets shows that our model is robust to MOT assignation problems namely similarity between target appearance model, heavy occlusion between targets and the birth and the death of targets. We also investigated the impact of different percentage of false detections on MOTA, Precision and

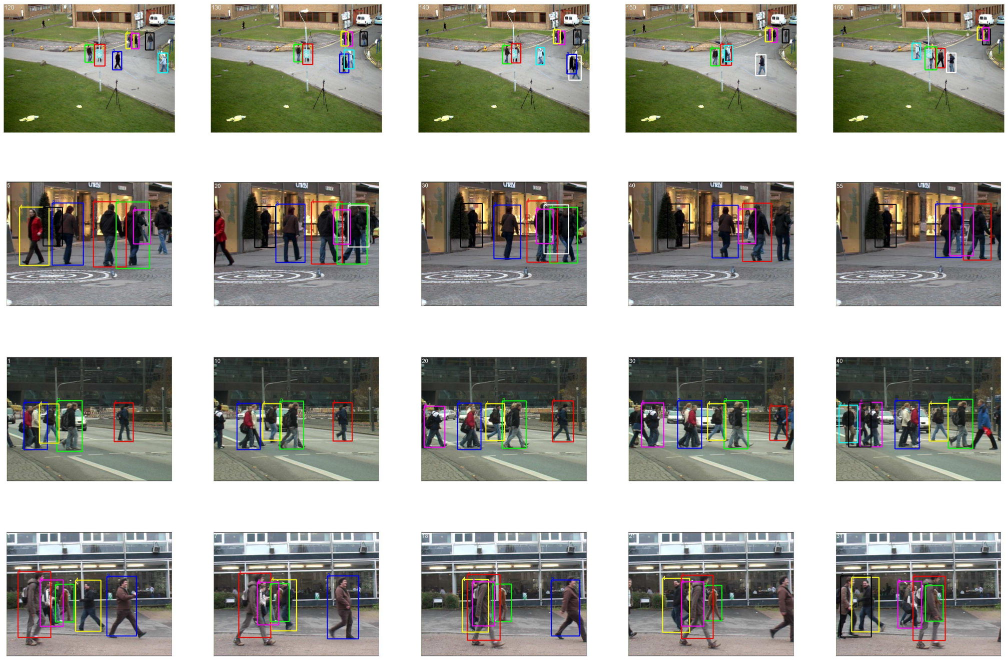


Figure 7:
Results for dataset. First row: PETS2009-S2-L1 (frames 120, 130, 140, 150 and 160), Second row: TUD-Stadtmitte (frames 5, 20, 30, 40 and 55), Third row: TUD-Crossing (frames 1, 10, 20, 30 and 40) and Fourth row: TUD-Campus (frames 1, 7, 18, 21 and 31)

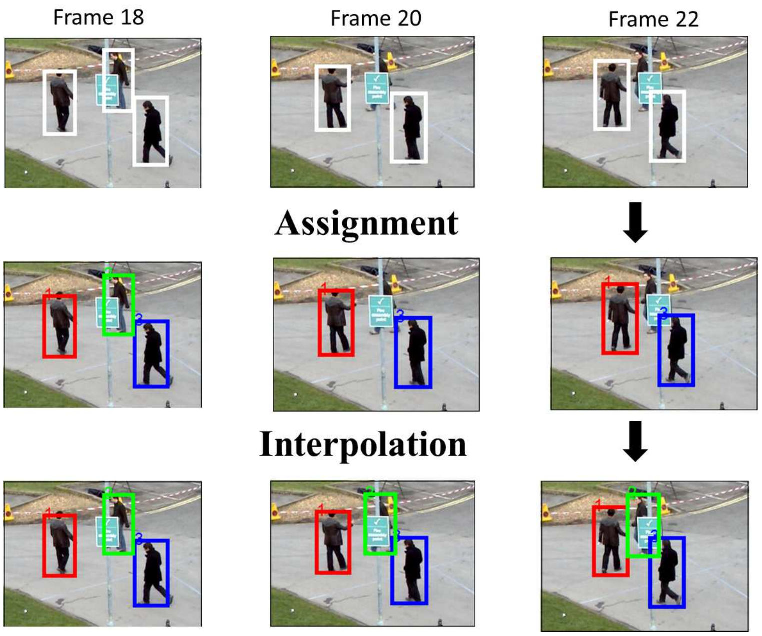


Figure 8: Interpolation of targets in the case of missing detections

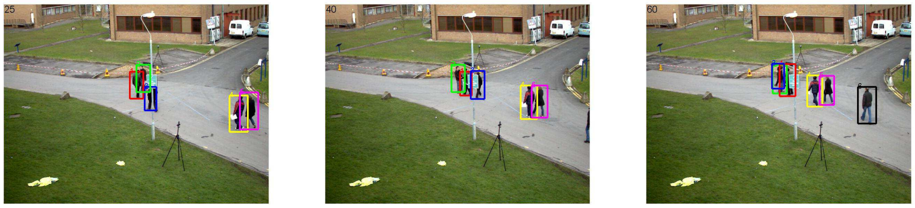


Figure 9: Keeping identity under multiple occlusions. Tracking results in frames 25, 40 and 60

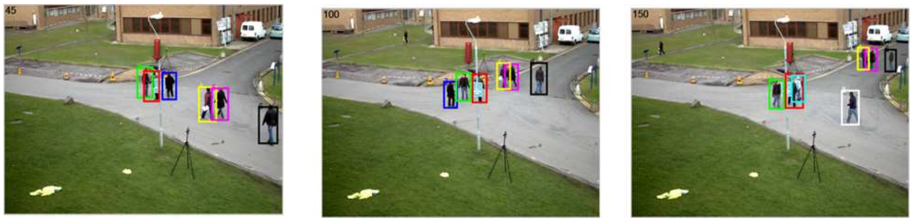


Figure 10: Keeping identity under long-term occlusion. Tracking results in frames 45, 100 and 150

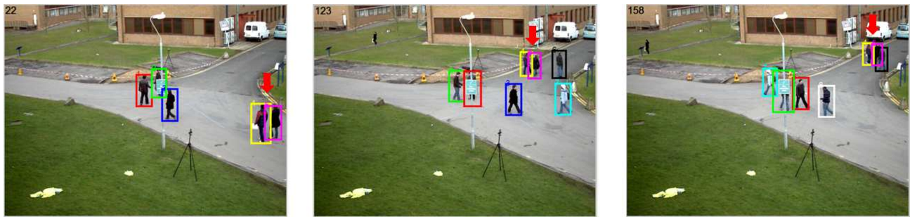


Figure 11: Keeping identity under scale changes. Tracking results in frames 22,123 and 158

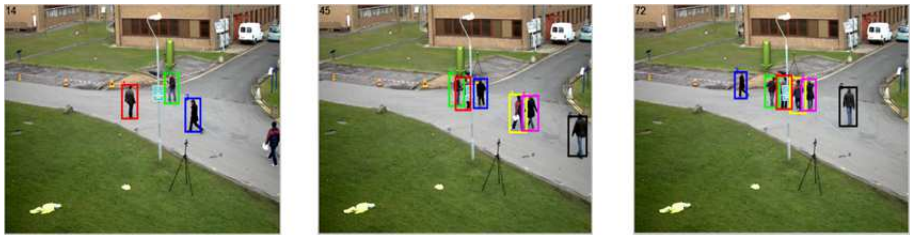


Figure 12: Keeping identity under pose change. Tracking results in frames 14, 45 and 72

Recall. We use three kinds of false detections: false negative detections, false positive detections and inaccurate detections. All the false detections are added randomly in different proportion 0%, 5%, 10%, 15%, 20%, 25% and 30%. We compare the performance of our proposed MOT with the following baselines:

> Baseline1: we implemented a version of our approach with no interpolation to show how the interpolation of a target can help to reduce the impact of false detection responses on the performance of our approach.
>
> Baseline2: we implemented a MOT approach which uses only the color feature to discriminate targets from each other. It demonstrates the impact of the feature fusion.
>
> Baseline3: we implemented a MOT approach which uses only the sparse representation feature to discriminate targets from each other. It demonstrates the impact of the feature fusion.

The graphs of figure 13 show that our proposed algorithm is more robust than the baselines. In fact, our approach maintains the best performance while the false detections change. In term of MOTA, we achieve results between 100% and 62% with false detection percentage between 0% and

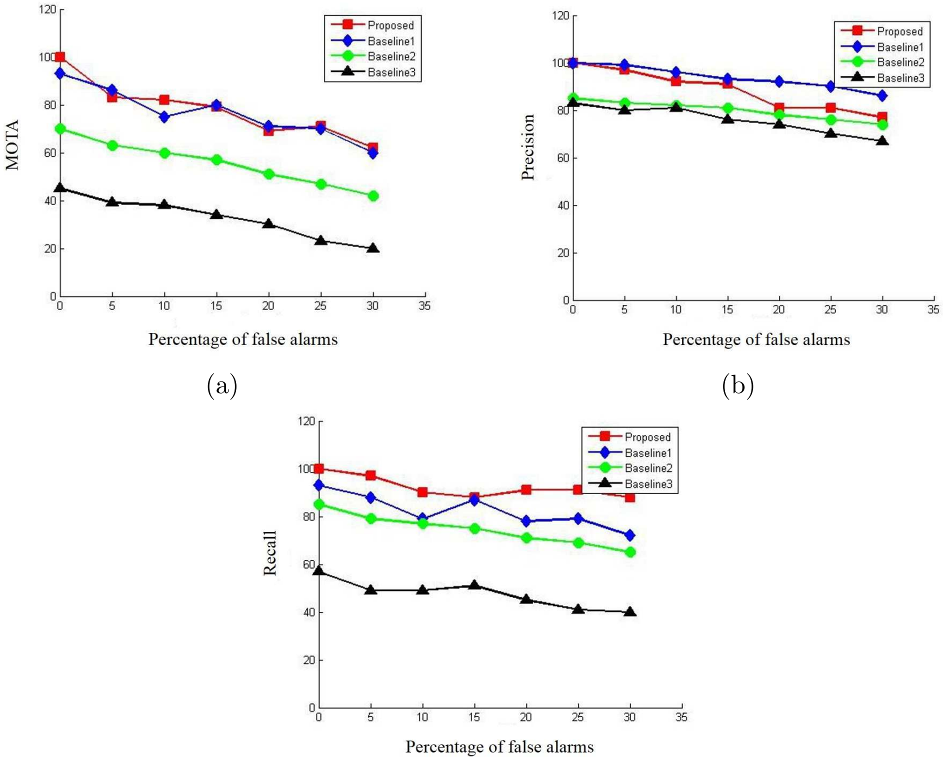
DataSets	MOTA	MOTP	FN	FP	IDS	Recall	Precision
TUD-CAMP	100%	100%	0%	0%	0	100%	100%
TUD-CROSS	97%	100%	3%	0%	1	97%	100%
TUD-STADM	100%	100%	0%	0%	0	100%	100%
PETS09-S2-L1	99.65%	97.27%	0%	0%	5	99.6%	100%

Table 6: Evaluation results using the ground-truth detection

30% while if we use only the color feature, the MOTA is under 70% and it decreases to 40% with very high percentage of bad detection responses (30%). Regarding baseline1, the performance is best than the other baselines but the use of interpolation still give the best performance. The precision is still high when the percentage of the false detections increase. The black and green curves in figure 13 (sparse and color features) demonstrate that the color feature is more discriminative than the sparse feature. It is because with pedestrian video sequences, all targets are walking, so the shapes of the targets change often and is less reliable. All curves are decreasing. It means that the performance of our MOT method depends to some extent on the quality of the detection responses. We can see that our approach is less sensitive to the false detections than the baselines. In fact, our proposed approach has the highest MOTA and Recall value.

5. Conclusion

In this work, we proposed a novel and robust MOT algorithm, based on the combination of independent features. Our features are: color histogram model, sparse appearance model, optical flow histogram and spatial model. Feature descriptors are integrated into a data association method where all targets are matched with all candidates under local geometric constraints, and with target states that handle the occlusion, birth and death of targets over time. To handle the occlusion problem, we propose a hierarchical data association process in which all the targets are divided into two sets: occluded and unoccluded targets. Each set is matched separately. In order to improve the detection responses quality, we incorporate an additional process in our framework, which is the interpolation of the position of the lost



(a)

(b)

(c)

Figure 13: Results evaluation: (a) Evaluation of MOTA, (b) Evaluation of Precision, (c) Evaluation of Recall

target. Our main contribution is to explore the capability of an appearance model that fuses independent descriptors and the use of a simple and robust data association framework. The proposed method is compared to several state-of-the-art approaches, which demonstrate the benefits of our method. Our method is competitive on all tested videos.

Acknowledgements

This work was supported by NSERC discovery grant No. 311869-2010.

6. References

[1] A. Yao, D. Uebersax, J. Gall, L. Van Gool, Tracking people in broadcast sports, in: Pattern Recognition, Springer, 2010, pp. 151–161.

[2] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool, Online multiperson tracking-by-detection from a single, uncalibrated camera, Pattern Analysis and Machine Intelligence(PAMI), IEEE Transactions on 33 (9) (2011) 1820–1833.

[3] X. Yan, X. Wu, I. A. Kakadiaris, S. K. Shah, To track or to detect? an ensemble framework for optimal selection, in: ECCV, Springer, 2012, pp. 594–607.

[4] M. Yang, F. Lv, W. Xu, Y. Gong, Detection driven adaptive multi-cue integration for multiple human tracking, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1554–1561.

[5] T. E. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, Oceanic Engineering, IEEE Journal of 8 (3) (1983) 173–184.

[6] D. B. Reid, An algorithm for tracking multiple targets, Automatic Control, IEEE Transactions on 24 (6) (1979) 843–854.

[7] H. W. Kuhn, The hungarian method for the assignment problem, Naval research logistics quarterly 2 (1-2) (1955) 83–97.

[8] C.-H. Kuo, C. Huang, R. Nevatia, Multi-target tracking by on-line learned discriminative appearance models, in: CVPR, IEEE, 2010, pp. 685–692.

[9] B. Wang, G. Wang, K. L. Chan, L. Wang, Tracklet association with online target-specific metric learning, in: CVPR, IEEE, 2014, pp. 1234–1241.

[10] S. Zhang, J. Wang, Z. Wang, Y. Gong, Y. Liu, Multi-target tracking by learning local-to-global trajectory models, Pattern Recognition.

[11] B. Yang, R. Nevatia, Multi-target tracking by online learning a crf model of appearance and motion patterns, International Journal of Computer Vision(IJCV) 107 (2) (2014) 203–217.

[12] A. V. Segal, I. Reid, Latent data association: Bayesian model selection for multi-target tracking, in: ICCV, IEEE, 2013, pp. 2904–2911.

[13] C. Huang, B. Wu, R. Nevatia, Robust object tracking by hierarchical association of detection responses, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 788–801.

[14] C.-H. Kuo, R. Nevatia, How does person identity recognition help multi-person tracking?, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1217–1224.

[15] X. Shi, H. Ling, W. Hu, C. Yuan, J. Xing, Multi-target tracking with motion context in tensor power iteration, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 3518–3525.

[16] F. Poiesi, R. Mazzon, A. Cavallaro, Multi-target tracking on confidence maps: An application to people tracking, Computer Vision and Image Understanding 117 (10) (2013) 1257–1272.

[17] A. Andriyenko, S. Roth, K. Schindler, Continuous energy minimization for multi-target tracking, IEEE TPAMI 35 (1) (2014) 1.

[18] S. Tang, M. Andriluka, B. Schiele, Detection and tracking of occluded people, International Journal of Computer Vision(IJCV) (2012) 1–12.

[19] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM computing surveys (CSUR) 38 (4) (2006) 13.

[20] A. Yao, D. Uebersax, J. Gall, L. Van Gool, Tracking people in broadcast sports, in: Pattern Recognition, Springer, 2010, pp. 151–161.

[21] E. Maggio, F. Smeraldi, A. Cavallaro, Combining colour and orientation for adaptive particle filter-based tracking., in: BMVC, 2005.

[22] H. Possegger, T. Mauthner, P. M. Roth, H. Bischof, Occlusion geodesics for online multi-object tracking, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 1306–1313.

[23] E. Erdem, S. Dubuisson, I. Bloch, Fragments based tracking with adaptive cue integration, Computer vision and image understanding 116 (7) (2012) 827–841.

[24] J. H. Yoon, M.-H. Yang, J. Lim, K.-J. Yoon, Bayesian multi-object tracking using motion context from multiple objects, in: Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on, IEEE, 2015, pp. 33–40.

[25] S. He, Q. Yang, R. W. Lau, J. Wang, M.-H. Yang, Visual tracking via locality sensitive histograms, in: CVPR, IEEE, 2013, pp. 2427–2434.

[26] C. Bao, Y. Wu, H. Ling, H. Ji, Real time l1 tracker using accelerated proximal gradient approach, in: CVPR, IEEE, 2012, pp. 1830–1837.

[27] B. K. Horn, B. G. Schunck, Determining optical flow, in: 1981 Technical Symposium East, International Society for Optics and Photonics, 1981, pp. 319–331.

[28] R. Chaudhry, A. Ravichandran, G. Hager, R. Vidal, Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1932–1939.

[29] A. Milan, K. Schindler, S. Roth, Detection-and trajectory-level exclusion in multiple object tracking, in: CVPR, IEEE, 2013, pp. 3682–3689.

[30] B. Keni, S. Rainer, Evaluating multiple object tracking performance: the clear mot metrics, EURASIP Journal on Image and Video Processing 2008.

[31] A. Andriyenko, K. Schindler, Multi-target tracking by continuous energy minimization, in: CVPR, IEEE, 2011, pp. 1265–1272.

[32] J. Berclaz, F. Fleuret, P. Fua, Robust people tracking with global trajectory optimization, in: CVPR, Vol. 1, IEEE, 2006, pp. 744–750.

[33] J. Yang, P. A. Vela, Z. Shi, J. Teizer, Probabilistic multiple people tracking through complex situations, in: 11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2009.

[34] G. Führ, C. R. Jung, Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras, Pattern Recognition Letters 39 (2014) 11–20.

[35] D. Riahi, G.-A. Bilodeau, Multiple feature fusion in the dempster-shafer framework for multi-object tracking, in: Computer and Robot Vision (CRV), IEEE, 2014, pp. 313–320.

[36] H. Pirsiavash, D. Ramanan, C. C. Fowlkes, Globally-optimal greedy algorithms for tracking a variable number of objects, in: CVPR, IEEE, 2011, pp. 1201–1208.

[37] A. Andriyenko, K. Schindler, S. Roth, Discrete-continuous optimization for multi-target tracking, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1926–1933.

[38] J. Sherrah, B. Ristic, D. Kamenetsky, A pedestrian multiple hypothesis tracker fusing head and body detections, in: Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on, IEEE, 2013, pp. 1–8.