A METHOD FOR TRACKING ROAD OBJECTS

Salma Kammoun Jarraya[1,2]

[1]Computer Science Deapartment, Faculty of Computing and Information Systems
King Abdulaziz University, Jeddah, Saudi Arabia.
[2]MIRACL-Sfax

ABSTRACT

In this paper, we present a new road traffic monitoring approach for a highway control and management system called RoadGuard. This system copes with several challenges in this type of applications in order to count and track road objects robustly. It adopts a novel approach for tracking road objects, first, to determine continuously their positions on the road, and then it uses vehicle positions to estimate their trajectories. The trajectory analysis provides vital information to control and manage highway traffic. In this paper, our main contribution is a tracking method based on a coherent strategy where both region and object information are used to establish objects correspondence over time. Our method operates in two phases: a spatial analysis that uses a multilevel region descriptors matching in order to identify object interactions and particular object states; and a continuous temporal analysis applied to cope with track management issues. As demonstrated experimentally, the proposed method can detect, track and count road objects accurately in highway videos that include several constraints. In addition, it produces effective and stable road objects tracking.

KEYWORDS

Foreground segmentation, Tracking Target, SIFT, Control and management system.

1. INTRODUCTION

The alarmingly increasing numbers of car accidents stim- ulated several research efforts to find counter measures. Among the explored solutions, computer vision software are being developed for highway control and management. These software rely essentially on tracking road objects to estimate the trajectory of moving objects over time. The information gathered by road object tracking helps in identifying their behavior in the observed scene. In addition, it can be used to collect statistical information about the traffic, which in turn can be used to control and manage the traffic to prevent road congestion and accidents.

In this highway traffic control and management context, the work presented in this paper aims at proposing a new method for tracking multiple rigid moving objects (*i.e.*, road objects) with different sizes and speeds in highway traffic videos. The videos are supposed to be acquired using a stationary camera with a large field of view, thus independently of the camera position. Our method relies on an automatic detection of moving objects. In addition, to handle the size and speed differences of the moving objects, our method has the merit of automatically accounting for possible state changes of the moving objects, interactions among them like occlusions, appearances of new objects and/or disappearances of existing objects.

The remainder of this paper is divided into four sections. In Section 2, we describe a brief state of the art in object tracking. Section 3 presents our proposed method. Section 4 highlights its advantages through the results of a quantitative and a qualitative evaluation. Finally, Section 5 recapitulates the presented work and outlines its extensions.

DOI: 10.5121/ijma.2018.10501

2. RELATED WORK

Tracking moving is one of the most challenging computer vision tasks. Several methods [24] [1] [3] [23] were proposed to deal with object tracking. In our study of existing methods, we focused on the modern methods called online trackers. More specifically, we did not consider pre-trained and offline trackers for which pre-processing steps are required.

Online tracking is a hard problem where all the information in the sequence is needed, especially in the initial frames [24]. The accuracy of these methods depends on both the constraints and context of the final-end application. The constraints are pertinent to the sensors (single or multiple, mobile or fixed), the observed scene (indoor and/or outdoor) and the tracked object(s) (single or multiple, rigid or nonrigid). It is worth noting that non-rigid object tracking methods allow tracking of both non-rigid and rigid objects (e.g. road objects) and can better deal with the various challenges. These advantages come however with a high computational time and a lack of genericity because these methods rely usually on silhouettes model that encodes the object nature and/or shape; the high computational time is unacceptable in our real-time application. Furthermore, existing rigid object tracking methods suffer from a low performance face to different challenges. Nonetheless, the low computational time of rigid object tracking methods motivated us to investigate this strategy for tracking road objects while improving their performance.

In addition to the constraints stemming from the application context, the methods reported in the literature differ in their object representation. According to the recent tracking survey papers [24] [1], the proposed methods can be classified into two categories of approaches: Model based (*cf.* [1] [3] [16] [17] and Features based (*cf.* [29] [6]). In both categories of approaches, the tracking strategy relies on matching information provided by features/models over time.

Model-based methods can be successfully used as long as they have accurate models for the different types of tracked vehicles. However, given their reduced timing complexity which is needed in real-time applications, tracking road objects is usually achieved using the features based approach. The method proposed in [26] requires no prior model, its main idea is to divide occluded vehicles into many small fragments (or patches) that are then grouped according to the clusters of motion vectors found by tracking feature points; as such, this method is applicable only to a high point of view. The method proposed in [6] tracks vehicles in real-time using Salient discriminative features (compactness, aspect ratio, and area ratio) and Euclidean distance to measure the distance between the centroids of two objects. Because the evaluation of this method [6] is only given for videos with low traffic, it is unclear how well it will perform for high traffic. In [8], a graph-based vehicle tracking method is used for building the correspondence between regions (compactness, aspect ratio). In [25], vehicle positions are predicted by Kalman Filter using feature vector (velocity, centroid position); Kalman Filter is a fast tracker and can deal with total occlusion. However, the performance of this method depends on a good quality of video since it requires an initialization step to detect vanishing point detection. Besides the aforementioned methods, several features-based methods [2][11][20][28] use descriptors points. Despite the popularity of descriptors points in other applications [29], few works used them to track road objects. Given their higher robustness compared to other features [29], in addition to their speed, we revisit the original idea of local features based on descriptors points to apply it for road object tracking.

Among the techniques used to compute descriptor points are Harris detector [5], KLT (Kanade-Lucas-Tomasi) detector [14] [27] and SIFT descriptor (Scale Invariant Feature Transform) [12] [13]. We performed a comparative study between these techniques according to a set of invariance criteria (Translation, Scale Changes, Image Rotation, Illumination changes,Image Locale Deformations, Affine Transformation), the results show that, unlike the other techniques, descriptors from SIFT are invariant to different criteria. In addition, from a theoretical point of view, SIFT can produce a great number of descriptor points, give a local image measurement that is robust to noises and to partial occlusions, and it can give distinctive points as well. Encouraged by the advantages of descriptors points, we have decided to adopt SIFT descriptor to track road object.

Within the application context of road traffic, the success of object tracking relies on the management of frequently object state changes (the lifecycle) as well as object interactions. The lifecycle of a road object starts by its appearance in the scene (state *'Entry'*) and ends by its disappearance (state *'Exit'*). In addition, during its presence in a scene, a road object can be in a normal state (*'Normal'*), a normal state with a high speed (*'Normal HS'*), stopped (*'Stopped'*), restarting motion after stopping (*'Re-moving'*).

Furthermore, during a lifecycle, two types of interactions among road objects can occur. The first interaction happens when two or many objects appear close to one another (*'Merge'*) causing partial or total occlusion. The second interaction results from two or many objects fragmentation (*'Split'*) after their merger.

In the literature, most of the proposed methods [28] [11] [2] track pre-selected (single and rarely multiple) specific object(s), the method proposed by Rahman *et al.* [20] is the exception. This latter proposes a multiple objects tracking detected automatically, their experiments show that the proposed method is dedicated to track two object in simple indoor scene. In addition, states changes of moving objects are not considered.

We have examined recent tracking road objects methods [6] [8] [25] and their challenges. Also, we have considered the work of [19] which has a good established reputation demonstrated by the number of times it has been cited. We summarize our observations in Table 1. In [8] and [6], Entry and Exit states of objects and the counting step are managed through a very small region of interest fixed manually: frame with five vehicles only two are tracked and counted. In [25], the authors claim that the graph association and weight assignment can deal with the various object states and interactions except Re-Moving and stopped objects. However, this paper gives neither a quantitative evaluation nor an explanation on how this can be done.

Table 1.Tracking road objects methods and their challenges

Challenge	**Paper [8]** **Cars and bus**	**Paper [6]** **Cars and bikes**	**Paper [25]** **Cars**	**Paper [19]** **Cars**	**Our method** **Road object**
Appearance [Entry, Re-Moving]	Yes{Entry}	Yes{Entry}	Yes{Entry}	–	Yes {Entry/Re-Moving}
Disappearance [Exit, Stop]	Yes{Exit}	Yes{Exit}	Yes{Exit}	Yes{Exit}	Yes {Exit/Stopped}
Merge	Yes	–	–	Yes	Yes
Split	Yes	–	–	Yes	Yes
Occlusion [Partial, total]	Yes{Partial}	–	Yes{Partial}	Yes {Partial, total}	Yes{Partial}
Counting	Yes	Yes	–	Yes	Yes

As shown in Table 1, Kalman filter and vanishing point detection are used to deal with Entry, Exit and occlusions in [25]. In [19], the proposed method is based on Kalman filter and vehicle contours to determine relation among objects. This kind of practice requires ”perfect” conditions of video acquisition. We can see also that the works [8] and [19] deal with Merge objects using spatial analysis which is not efficient for more than two merged objects. As recommended in [24], combining local (e.g. SIFT) and global features (e.g. template matching) can resolve complex, frequently merged objects.

From Table 1, we can conclude: (1) despite its popularity, SIFT-based tracking objects is not very much applied in the context of road object, and (2) the challenges related to road object tracking remain unsolved with a single method.

In this paper, focusing on tracking road objects for a straightforward surveillance and security application, our proposed method aims to track an unlimited number of rigid road objects (multiple rigid moving objects) with different sizes and speeds. More specifically, our objective is to propose an online tracking features based method that both is capable of overcoming various challenges and has a low computational time. In addition, the proposed method must take into consideration: (1) possible state changes and interactions of road objects, and (2) the appearance of a new or old object and the disappearance of existing object. To handle all these challenges, we propose a SIFT and Template Matching-based method to track road object at low and high point of views, even in the presence of severe object interactions and significant object state changes.

3. PROPOSED METHOD

Our proposed method for tracking road objects is based on the two main steps : (1) Spatial Analysis (AS) to manage objects' states and interactions for each input frame; and (2) continuous Temporal Analysis (CTA) to establish all objects tracked from the beginning up to an instant t and to generate objects' trajectories. Note that while AS produces object information based on consecutive frames (at time t-1 and t), CTA produces object information that is 'global' in time (from 0 to a frame t). Both processing rely on similarity measuring and matching.

3.1.Spatial Analysis (AS)

We adopt our fast and accurate moving object detection method described in [4] to obtain targets to be tracked. Let R_t^{cc} and R_{t-1}^{c} denote respectively the segmented regions from frames ξ_{01}^{t} andξ_{01}^{t-1} with cc ∈$\{1,...,m\}$ and c ∈$\{1,...,n\}$, n and m, are respectively the number of region in two successive frames.In STP, the $R_t^{cc}[cc \in \{1,...,m\}]$ and $R_{t-1}^{c}[c \in \{1,...,n\}]$ regions are used to manage objects states and interactions for each input frame, thus produces region correspondence ($R_{j=\{t-1,t\}}^{i=\{c,cc\}}.Cor$) and state ($R_{j=\{t-1,t\}}^{i=\{c,cc\}}.State$). Spatial analysis takes into account both the spatial attributes and multilevel region descriptors matching of $R_t^{cc}[cc \in\{1,...,m\}]$ and R_{t-1}^{c} $[c \in \{1,...,n\}]$. Each region R is represented by a set of attributes (Z(R) = ($\beta^{1...5}(R),\varphi(R)$)). Where $\beta^{1...5}(R)$ are 2D spatial attributes (*cf.* Figure 1) and $\varphi^{128}{}_{k}(R)$ is a *K-by-128* matrix, each row gives an invariant descriptor for one of the K key points. The descriptor is a vector of 128 values normalized to unit length. Regions correspondences ($R_{j=\{t-1,t\}}^{i=\{c,cc\}}.Cor$, $[cc \in\{1,...,m\}$, $c \in\{1,...,n\}]$) are initialized by-1.We project $\beta_t^{1}(R_t^{cc})$ $[cc \in\{1,...,m\}]$ onto area from$\beta_t^{2..5}(R_{t-1}^{c})$ $[c \in\{1,...,n\}]$, thusprovides correspondence for regions in *’Normal’* states and/or in *’Split’* interactions. Region in state *’Normal’* corresponds to the case where $\beta_t^{1}(R_t^{cc})$ belongs to only oneR_{t-1}^{c} area. The Split interaction corresponds to the case where β_t^{1} of two or more R_t^{cc}belong to oneR_{t-1}^{c} area. We associate regions R_t^{cc}andR_{t-1}^{c} according to equation 1.

$$\begin{cases} \quad if\ \ 'Norma'State\ then \\ \qquad R_t^{cc}.Cor = c \\ else\ if\ \ 'Split'Interraction\ then \\ \qquad R_t^{cc1,cc2,...}.Cor = c \end{cases} \quad (1)$$

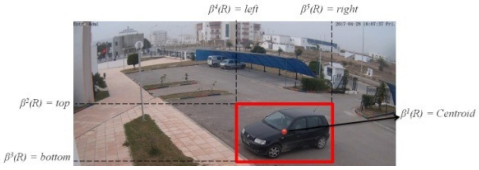

Figure 1. 2D Spatial attributes ($\beta^{1...5}(R)$)

A multilevel region descriptors matching is proposed for regions $R_{j=\{t-1,t\}}^{i=\{c,cc\}}$ withcorrespondences $(R_{j=\{t-1,t\}}^{i=\{c,cc\}}.Cor == -1)[cc \in\{1,...,m\}, c \in\{1,...,n\}]$. This step allows us to cope with region interaction (*'Merge'*) and states (*'Entry'*, *'Exit'*, *'Normal HS'*, *'Stopped'* and *'Re-Moving'*). We aim to select, for each region descriptors$(\phi_{i=1...k_1}^{128}(R_t^{cc}))$, its match to $(\phi_{k_2}^{128}(R_{t-1}^{c}))$ (equation 2).

$$\begin{cases} \text{R\_Match}\left(\phi_{\text{k1}}^{128}(R_t^{cc}), \phi_{k_2}^{128}(R_{t-1}^{c})\right) = 1 \\ \quad if\ any\ (Des_{Match} > 0) \end{cases} \quad (2)$$

There is matching (R*Match* = 1) between two regions in case of at least one descriptor match (*Des_Match* >0). Decision to select matched descriptors from $\phi_{k_2}^{128}(R_{t-1}^{c})$ is given by equation 3.

$$\begin{cases} \text{Des\_Match(i)} \quad = 1 \\ if\ Des - Match(i)\ =\ 1\ if\ (DPi(1) < 0.6\ *\ DPi(2)\ for\ i\ \in \{1, \dots, k1\} \end{cases} (3)$$

In our work, SIFT descriptors matching is based on dot products (DP^i[$i \in \{1,...,k_1\}$]) between unit vectors of descriptors (equation 4). Generic rules of the multilevel region descriptors matching is presented by Algorithm 1.

$$\text{DP}^i = \text{sort}(\text{arcosine}((\phi_{i=1...k_1}^{128}(R_t^{cc}))*(\phi_{k_2}^{128}(R_{t-1}^{c}))) \quad (4)$$

Three level matching levels are proposed: the first one is between $\phi_{k1}^{128}(R_t^{cc})$ and $\phi_{k_2}^{128}(R_{t-1}^{c})[cc\ \in \{1, \dots, m\}, c\ \in \{1, \dots, n\}]$ to identify regions with state *'NormalHS'* in case of$\phi_{k1}^{128}(R_t^{cc})$ matches to only one $\phi_{k_2}^{128}(R_{t-1}^{c})$ or prevent merging interaction (*'Merge'*) in case of $\phi_{k1}^{128}(R_t^{cc})$ matches to two or more $\phi_{k_2}^{128}\left(R_{t-1}^{c1,c2,...,cq}\right)$ (lines 1 to 4, Algorithm 1). The second one is between $\phi_{k1}^{128}(R_t^{cc})[cc\ \in \{1, \dots, m\}, c\ \in \{1, \dots, n\}]$and *Stopped*(*h*).$\varphi$.

Structure *Stopped*(*h*).φcorrespond to region of stopped objects in previous frames, thus, if they match, R_t^{cc}[$cc \in \{1,...,m\}$] are in state *'Re-Moving'*, otherwise they are in state 'Entry' (lines 5 to 8, Algorithm 1). The third matching is between $\phi_{k_2}^{128}(R_{t-1}^{c})$ and $\phi_{k3}^{128}(R_t^{cc})$ to identify stopped objects, otherwise means disappearance of$R_{t-1}^{c}('Exit')[c \in \{1, \dots, n\}]$ (lines 9 to 13, Algorithm 1). $\phi_{k3}^{128}(R_t^{c})$] corresponds to SIFT descriptors of$\beta_t^{2..5}(R_{t-1}^{c})$ projection onto current frame.

More precisely, attributes of objects in states *'Entry'*, *'Split'* and *'NormalHS'* are updated according to equation 5 . Objects in state *'Stopped'* are controlled by *Stopped*($O_{j=1...h}$).φ and objects in state *'Exit'* $\left(TrackingObject\{O_i.Z_t^{O_i}(Cor)\} == R_t^{cc}.Cor\right)OR(R_t^{cc}.Cor ==*)$ are killed.

$$\left(\begin{array}{l} If\ \left(TrackingObject\{O_i.Z_t^{O_i}(Cor)\} == R_t^{cc}.Cor\right) OR (R_t^{cc}.Cor == *) Then \\ \\ \qquad\qquad TrackingObject\{O_i.Z_t^{O_i}(Cor)\} = cc \\ \qquad TrackingObject\{O_i.Z_t^{O_i}(\beta^{1\ldots5})\} = \beta^{1\ldots5}(R_t^{cc}) \\ \qquad\quad TrackingObject\{O_i.Z_t^{O_i}(\phi)\} = \phi(R_t^{cc}) \end{array} \right. \qquad (5)$$

Algorithm 1 Multilevel region descriptors matching

```
Input:
-  $\phi_{k1}^{128}(R_t^{cc})$: Regions descriptors at t-1 [$cc$ ∈{1,...,$m$}]
-  $\phi_{k2}^{128}(R_{t-1}^{c})$)): Regions descriptors at t [$c$ ∈{1,...,$n$}]
-  Stopped($O_j$).$\varphi$: structure of objects in state 'Stopped' [$j$ ∈{1,...,$h$}]
Output:
-  $R_t^{cc}.State$: Regions states structure at t
-  $R_t^{cc}.Cor$:Regions matching structure at t
-  Stopped($O_j$).$\varphi$:structure of objects in state 'Stopped'
IfR_Match$(\phi_{k1}^{128}(R_t^{cc}), \phi_{k_2}^{128}(R_{t-1}^{c})^T)$Then
    If R_Match$(\phi_{k1}^{128}(R_t^{cc}), \phi_{k_2}^{128}(R_{t-1}^{c})^T)$ Then
        1. $R_t^{cc}.Cor = c$
        2. $R_t^{cc}.State = ' NormalHS'$
    Else if R_Match $\left(\phi_{k1}^{128}(R_t^{cc}), \phi_{k_2}^{128}\left(R_{t-1}^{c1,c2,..,cq}\right)^T\right)$
        1. $R_t^{cc}.Cor = c1, c2, \ldots, cq$
        2. $R_t^{cc}.State = ' Merge'$
    End if
Else
    If R_Match$(\phi_{k1}^{128}(R_t^{cc}), Stopped(O_j).\varphi)[cc \in \{1, \ldots, m\}]$ Then
        1. $R_t^{cc}.Cor = Stopped(O_j).Cor$
        2. $R_t^{cc}.State = ' Re - Moving'$
    Else
        1. $R_t^{cc}.Cor = *$
        2. $R_t^{cc}.State = ' Entry'$
    End If
    If R_Match $\left(\phi_{k2}^{128}(R_{t-1}^{c}), \phi_{k3}^{128}(R_t^{c})\right)$ $[c \in \{1, \ldots, n\}]$ Then
        1. $Stopped(O_{j+1}).Cor = c$
        2. $Stopped(O_{j+1}).State = ' Stopped'$
    Else
        1. $R_{t-1}^{c}.Cor = *$
        2. $R_{t-1}^{c}.State = ' Exit'$
    End If
End
```

3.2 Continuous Temporal Analysis (CTA)

CTA establishes all objects tracks (*TrackingObject*{O_i} [i∈{1,...,*ObjectCount*}]) from the beginning of the video stream until an instant*t*. It relies on $R_{t-1,t}^{c,cc}.Cor$and $R_{t-1,t}^{c,cc}.State[cc \in \{1, \ldots, m\}, c \in \{1, \ldots, n\}]$ to generate objects trajectories.

For each frame, LTP rule feedbacks objects (*TrackingObject*$\{O_i\}$ [$i\in$ *1,...,ObjectCount*]) and their corresponding regions to update tracked object attributes. Spatiotemporal attributes and descriptors of tracked object ($Z_t^{O_i} = (\beta^{2..5}(O_i), Cor, \varphi(Oi))$ are updated according to region/object association. The association between objects and their corresponding regions is based essentially on$R_t^{cc}.Cor$.

Note that attributes of objects in a merging region cannot easily be obtained since several objects share the same region. To deal with this problem, we use template matching based sum of squared difference to find 2D spatial attributes of each object, then, we compute their SIFT descriptors. Sum of squared difference is implemented using FFT (Fast Fourier Transform) based correlation.

4. EXPERIMENTAL RESULTS

In order to validate our contributions, we experimentally evaluated the proposed method to track road objects. We carried out two series of experiments whose results are presented in the second sub-section. We first clarify the experimental conditions, the used data set and validation conditions and techniques.

We used two road traffic sequences[1]recorded in typical conditions (*HighwayII*and *HighwayIII*). *'HighwayII'* shows a dense traffic of road objects that share some characteristics (color, size, ...). The distance between road objects is often very small, which produces partial occlusions and, hence, frequent interactions (*'Merge'* and *'Split'*). *'HighwayIII'* shows a dense traffic of road objects of different speeds often very fast, and with different types and sizes.

In the first experiment, we evaluated quantitatively and qualitatively the accuracy of our method for road objects tracking. Quantitative evaluations need *Ground-Truths (G-T)* which can be viewed as the *correct answer* for what exactly the algorithm is expected to produce; they are used to evaluate the obtained results quantitatively. However, since Ground-Truth tracks are not available for these sequences despite their celebrity, we had to develop a semi-automatic software to produce, for each sequence, several road object tracks from typical sequence parts. Four Parts from *HighwayII*and five parts from *HighwayIII*cover several challenges (frequent *'Merge'* and *'Split'* between road objects, dense traffic includes road objects in different size and speed, several states changes of road objects at the same time, high road objects speeds).

The evaluation is made through the calculation of the rates of *Centroid Error* [22] [18] [15] with regard to Ground-Truth (GT) of parts in both sequences (4 parts for *'HighwayII'* and 5 parts for *HighwayIII*). The *Centroid Error* rates are computed by the Euclidean distance (between two centroids) according to a *twopass matching scheme*: the first pass matches the system track to GT (*distanceSy*) to find false positive tracks , and the second pass matches the GT to system track (*distanceTrack*) to find false negative tracks. In typical results, the *Centroid Error rates* from the two passes are the same.In addition to the above quantitative metric, we also considered in our evaluation a second metric *'Two-pass many-to-many system to ground truth track matching'* [10] to measure how the system can deal with *'Merge'* and *'Split'* interactions. A *GT/system track* is matched to the *system/GT track* if there is both *temporal overlap* and *spatial overlap*. *Temporal overlap* is with respect to the duration of the system track. *Spatial overlap* is based on the centroid of the system lying inside the bounding box of the ground truth track. If multiple *GT-matches*, then this system track has *'Merge Error'* equal to matched GT-tracks. If multiple *system-matches*, then this GT track has *'Split Error'* equal to matched system- tracks.

[1] http://cvrr.ucsd.edu/aton/shadow/

In the second experiment, we evaluate our results by comparing our systemperformance with the following similar works: a Kalman filter based method [9], a well-known and referenced paper [19] and a recent proposed method [25]. In addition, the effectiveness and accuracy of the proposed method is demonstrated, in the experiment 3, through a Highway Control and Management System, called *RoadGuard*[7]. Semantic results of *RoadGuard*are based on counting and tracking moving vehicles starting by detecting the real moving objects. We next present and discuss results of the (1) Experiment 1: Quantitative and Qualitative evaluations, (2) Experiment 2: Comparison with related works, and (3) Experiment 3: *RoadGuard*

4.1.Experiment 1: Quantitative and Qualitative evaluations

As we can see in Figure 2, the four *HighwayII*parts echoed a very low average *distanceSys/distanceTrack*rate per frame. Table 2 summarizes the average *distanceSys/distanceTrack*and *FPT /FNT* rates for each sequence part: the *distanceSys/ - distanceTrack*echoed respectively between 0 and 6.163 pixels while *FPT* and *FNT* are between 0 and 6.19 percent.

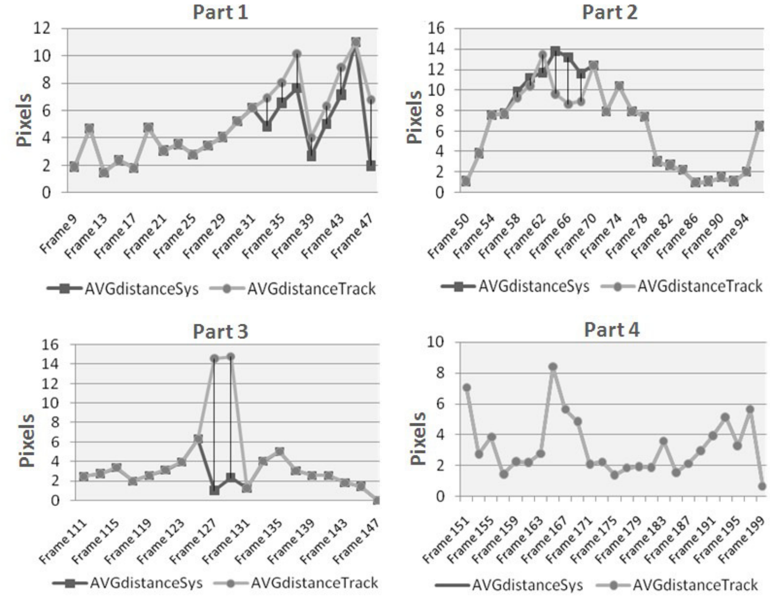

Figure 2**.** Average distanceSys/distanceTrack curves of the four parts from *HighwayII*

We have performed an experimental study to know how our system can deal with *'Merge'* and *'Split'* interactions. *'Merge Error'* and *'Split Error'* are computed for 11 tracks from *HighwayII*. *Temporal overlap* and *Spatial overlap* curves for the 11tracks are depicted in figure 3. For each track, both measures are computed firstly (A) from GT-Track-Matching and secondly (B) from System-Track-Matching.

Table 2.Average *AVGdistanceSys (AVGdS) /AVGdistanceTrack(AVGdT)* and *FPT / FNT* rates for *HighwayII*(HII)

HII part	AVGdS	AVGdT	FPT	FNT
HII Part 1	4.403	5.140	6.19	0
HII Part 2	6.618	6.163	0.83	0
HII Part 3	2.700	4.070	3.51	0
HII Part 4	0	3.283	0	0

There is a *'Merge Error'/'Split Error'* in the case of multiple *GT-matches/systemmatches*. More explicitly, if a curve from *GT-Track-Matching/ System-Track Matching* show more than peak with temporal overlap greater than 0.5. Our system achieves an average *'Merge Error rate'* of 9.09 percent per 11 tracks and an average *'Split Error rate'* of 0 percent.

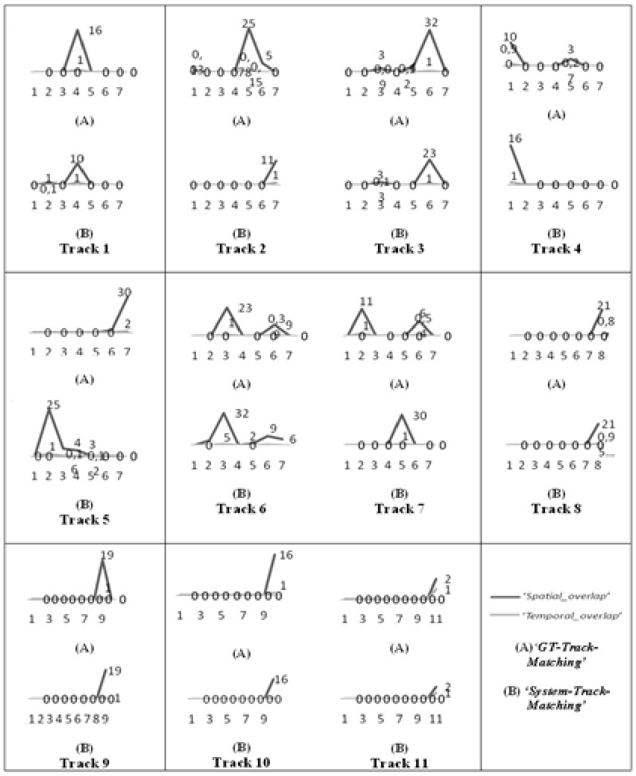

Figure 3.Temporal overlap and Spatial overlap of 11 tracks from *HighwayII*

The five *HighwayIII*parts (figure 4) showed a low average *distanceSys /distanceTrack*rate per part, respectively between 1.928 and 9,682 pixels. *FPT* and *FNT* are between 0 and 20,44 percent. The average *distanceSys/-distanceTrack*rates per frame are given in table 3.

Table 3.Average *AVGdistanceSys (AVGdS) /AVGdistanceTrack(AVGdT)* and *FPT / FNT* rates for *HighwayIII(HIII)*

HII part	AVGdS	AVGdT	FPT	FNT
HIII Part 1	1.928	7.547	5.77	0
HIII Part 2	8.795	0.685	3.85	0
HIII Part 3	3.181	9.682	10.08	9.52
HIII Part 4	7.361	5.247	0.60	20.44
HIII Part 5	1.505	5.245	1.67	0

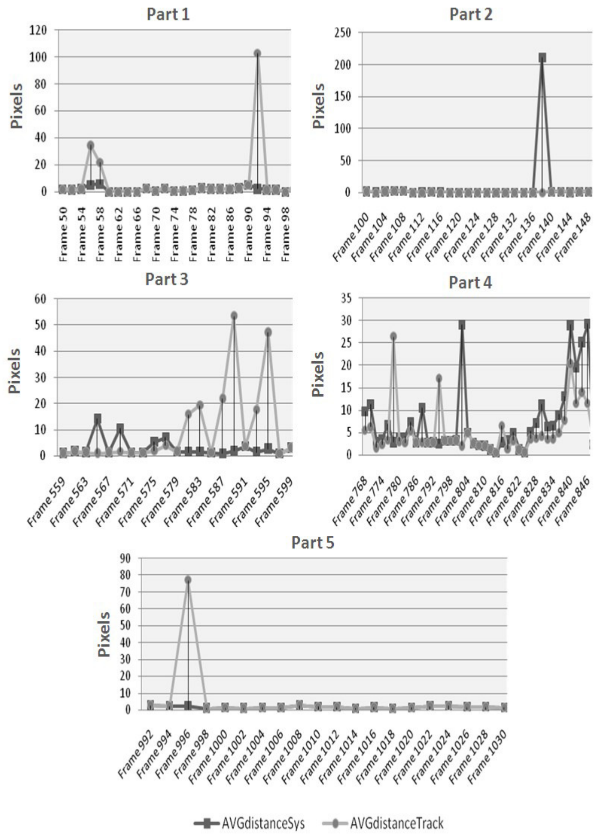

Figure 4.Average *distanceSys/distanceTrack*curves of each frames part 5 from *HighwayIII*

Figure5 present qualitative results on frames from *HighwayI* for tracking road objects (car, bike and person). Overall, our method produces good results in presence of merged objects, appearance of new object and disappearance of objects. In addition, Figure 5 presents the results of counting road objects (first column) in presence of Merged/Split objects. As we can see in the detection results (second column), merged objects appear as one object and a stopped object does not appear in the detection. Nevertheless, our tracking method counts correctly the existing road objects.

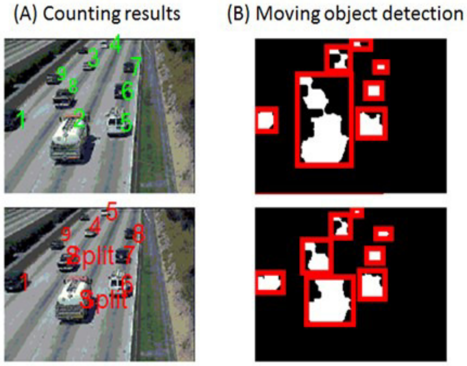

Figure 5.Results of counting road objects in presence of Merged/Split objects

4.2 Experiment 2: Comparison with related works

A complementary quantitative evaluation was performed by comparing our results (*Average distanceSys*/*distanceTrack*) with the results of Xin [9] on frames from *HighwayII*. Method of [9] is based on Kalman Filter, which is a probabilistic prediction rule to visual tracking. This method establish Kalman filter motion model with the features centroid and area of moving objects. Note that, the frame level tests are chosen from those containing objects in merge and split states.As illustrated in Figure 6, Kalman filter presents similar average *distanceSys/ distanceTrack rates* (∈ [1..4]) for the first frames part. However, they give high average *distanceSys/ distanceTrack*rates per frame (∈ [16..20]) in the second part which includes objects in merge and split states. In fact, Kalman filter fails to provide useable results in the presence of objects in merge and split states. On the same frames, our tracking method shows low average *distanceSys/distanceTrack*rates per frame (∈[3..4]).

For our application domain, the lack of open access to the codes, datasets and detailed descriptions of algorithms hinders the elaboration of a fair comparisonwith several methods and on large datasets.

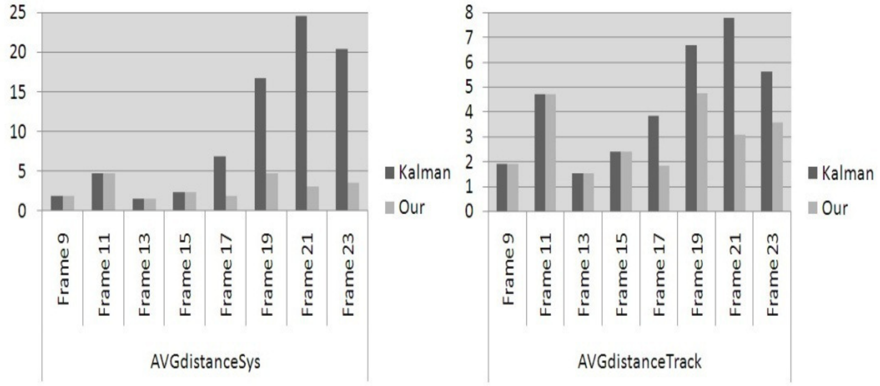

Figure 6.Comparative results with Kalman filter

Nevertheless to put our method in its context, a reasonable comparison or the performance can be deduced knowing the hardware, number and size of frames, and number of videos used in the evaluations. The performance can be analyzed in terms of the rates of Merge error, Split error, and Mean processing speed (MPS).

Table 4.Comparison with related works

Paper	Merge error	Split error	MPS	Accuracy
[25]	-	-	10 f/s	91.5%
[19]	2.5%	1.6%	10.99 f/s	96%
Our	9.09%	0%	11.02 f/s	97.92%

According to this evaluation (see Table 4), our method records the best accuracy with 97.92% for large number of frames (56 higher than [19]) because it is the only method that deals with stopped and Re-moving objects. In addition, our method is faster than the method of [19] and [25] which are considered in the literature as real time methods. The method of [25] achieves the best rate of Merge error 6.59% fewer than our rate for frames with high size and quality; but our method records a smaller rate for Split error.

4.3 Experiment 3: RoadGuard

RoadGuard is a highway control and management system we have implemented on a standard PC hardware. The control phase of *RoadGuard*is based on tracking vehicles in a defined Region Of Interest (ROI) and the emergency area. The management phase of the RoadGuard is based on counting vehicles in highways in order to obtain statistical information like the date and time of overloaded highways. The counting process is done in the ROI. The counter is incremented for each road object enters the ROI and decremented after its disappearance. In order to confirm the important effect of tracking road objects for highway control and management system, we integrate our proposed method in the *RoadGuard*process. ROI is obtained automatically by method of [21] (*cf.* figure 7). *RoadGuard*uses ROI to detect vehicles stopped in road.

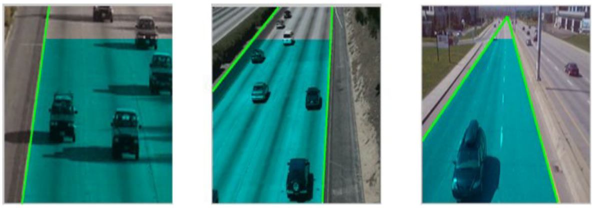

Figure 7.ROI of three highway sequences

*RoadGuard*achieves suitable rates for counting vehicles in *'HighwayII'* (4 parts) and *'HighwayIII'* (5 parts) (figure 8 and figure 9)when compared with the GT count (success rates: 90% for *'HighwayII'* and 82% for *'HighwayIII'*) .

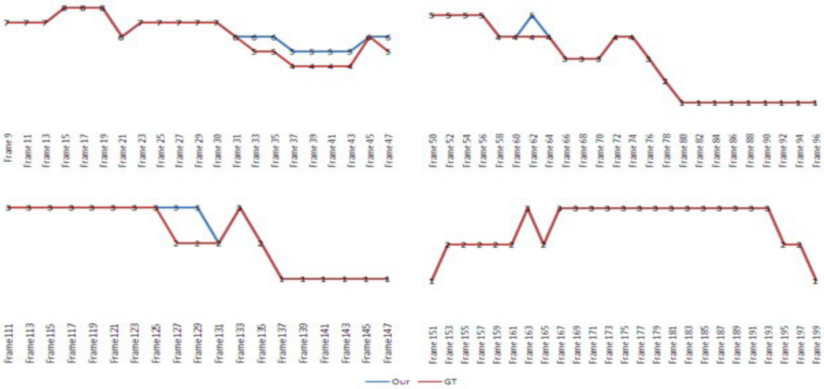

Figure 8.Counting results for four parts from *HighwayII*

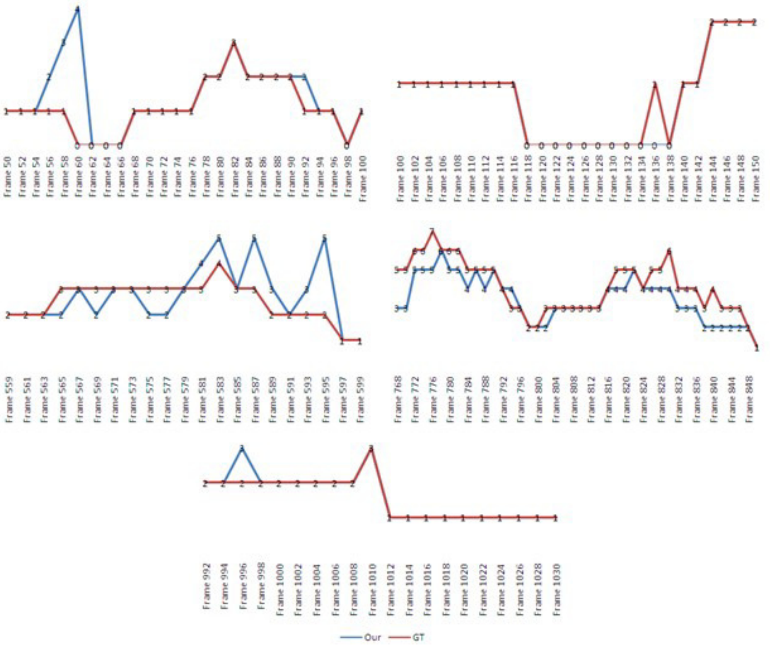

Figure 9. Counting results for five parts from *HighwayIII*

5. Conclusions

In this paper, we proposed a novel points-based method using the SIFT technique to track road objects in highway videos. Our method operates in two phases: Firstly, a a spatial analysis uses a multilevel region descriptors matching in order to identify object interactions and particular object states. Secondly, a continuous temporal analysis is applied to cope with track management issues. Our preliminary experimental evaluation on real highway sequences shows that our method can track multiple rigid moving objects (*i.e.*, road objects) with different sizes and speeds in traffic videos of highways. Moreover, the proposed method has the merit of automatically accounting for possible state changes of the moving objects, interactions among them like occlusions, appearance of new objects and disappearance of existing objects. Finally, this experimental study has demonstrated the practical usefulness of our contributions through a highways control and management system, called *RoadGuard*, which gives a good quality of results. Future works will focus on further interpretation of road objects' trajectories to identify suspect events.

References

[1] Z. Al-Ameen, G. Sulong, A. Rehman, M. Al-Rodhaan, T. Saba, & A. Al-Dhelaan, (2017)"Phasepreserving approach in denoising computed tomography medical images". CMBBE: Imaging & Visualization, 5(1):16–26.

[2] Y. Cheng-bo, Z. Jing, L. Yu-xuan, & Y. Ting, (2011)"Object tracking in the complex environmentbased on sift". 3rd International Conference on Communication Software and Networks, pages 150–153.

[3] X. Gao, Y. Su, X. Li, & D. Tao, (2010)"A review of active appearance models". IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(2):145–158, March.

[4] M. Hammami, S. K. Jarraya, & H. Ben-Abdallah, (2013)"On line background modeling for movingobject segmentation in dynamic scenes". Multimedia Tools and Applications, 63(3):899–926.

[5] C. Harris & M. Stephens. A combined corner and edge detector (1988)"Alvey vision conference".Volume: 15, Issue: Manchester, pages 147–151.

[6] D.-Y. Huang, C.-H. Chen, W.-C. Hu, S.-C. Yi, & Y.-F. Lin (2012),"Feature-based vehicle flowanalysis and measurement for a real-time traffic surveillance system". Journal of Information Hiding and Multimedia Signal Processing, 3(3):282–296.

[7] S. K. Jarraya, A. Ghorbel, A. Chaouachi, & M. Hammami (2011)"Roadguard - highway controland management system". In L. Mestetskiy and J. Braz, editors, VISAPP, pages 632–637. SciTePress.

[8] J. C. Lai, S. S. Huang, & C. C. Tseng (2010)"Image-based vehicle tracking and classification onthe highway". In The 2010 International Conference on Green Circuits and Systems, pages 666–670.

[9] X. Li, K. Wang, W. Wang, & Y. Li, (2010)"A multiple object tracking method using kalman filter". InThe IEEE International Conference on Information and Automation, pages 1862–1866.

[10] M. B. Lisa, W. S. Andrew, Y. li Tian, J. Connell, & A. Hampapur, (2005). "Performance evaluation of surveillance systems under varying conditions". IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance.

[11] Y. Liu, X. Wang, J. Yang, & L. Yao, (2011)"Multi-objects tracking and online identification basedon sift". International Conference on Multimedia Technology (ICMT),, pages 429–432.

[12] D. Lowe, (1999)"Object recognition from local scale-invariant features". The Proceedings of the Seventh IEEE International Conference on Computer Vision, pages 1150–1157.

[13] D. G. Lowe, (2004)"Distinctive image features from scale-invariant keypoints". International Journal of Computer Vision. Volume 60 Issue 2, pages 91–110.

[14] B. Lucas & T. Kanade, (1981)"An iterative image registration technique with an application tostereo vision". In Proceedings of the 7th international joint conference on Artificial intelligence, pages 674–679.

[15] T. Nawaz, A. Ellis, & J. Ferryman, (2017)"A method for performance diagnosis and evaluation ofvideo trackers". Signal, Image and Video Processing, 11(7):1287–1295.

[16] Y. Ouyang, (2017)"Structural sparse coding seeds–active appearance model for object tracking". Signal, Image and Video Processing, 11(6):1097–1104.

[17] G. Phadke& R. Velmurugan, (2017)"Mean lbp and modified fuzzy c-means weighted hybrid featurefor illumination invariant mean-shift tracking". Signal, Image and Video Processing, 11(4):665– 672, 11.

[18] G. Pingali& J. Segen, (1996)"Performance evaluation of people tracking systems". IEEE Workshop on Applications of Computer Vision, pages 33–38.

[19] R. Rad & M. Jamzad, (2005)"Real time classification and tracking of multiple vehicles in highways".Pattern Recognition Letters, 26(10):1597 – 1607.

[20] M. S. Rahman, A. Saha, & S. Khanum, (2009)"Multi-object tracking in video sequences based onbackground subtraction and sift feature matching". In Fourth International Conference on Computer Sciences and Convergence Information Technology, pages 457–462.

[21] N. B. Romdhane, M. Hammami, & H. Ben-Abdallah, (2011)"A comparative study of vision-basedlane detection methods". Advances Concepts for Intelligent Vision Systems, Volume 6915, pages 46–57.

[22] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, & R. Bolle, (2001)"Appearance models for occlusion handling". IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance.

[23] K. Shanmugapriya& R. S. M. Malar, (2017)"A multi-balanced hybrid optimization technique totrack objects using rough set theory". Signal, Image and Video Processing, 11(3):415–421.

[24] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, & M. Shah, (2014)"Visualtracking: An experimental survey". IEEE Trans. Pattern Anal. Mach. Intell., 36(7):1442–1468.

[25] J. Sochor& A. Herout, (2014)“Fully automated real-time vehicles detection and tracking with lanesanalysis”. In The 18th Central European Seminar on Computer Graphics, pages 666–670.

[26] B. Tamersoy& J. K. Aggarwal, (2010)“Counting vehicles in highway surveillance videos”. In 20th International Conference on Pattern Recognition, pages 3631–3635.

[27] C. Tomasi& T. Kanade, (1991)“Detection and tracking of point features technical report cmu-cs91-132. Technical report”, 1-22.

[28] Y. Yan, J. Wang, & C. Li, (2011)“Object tracking using sift features in a particle filter”. IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pages 384– 388.

[29] Y. Yang & Q. Cao, (2013), A fast feature points-based object tracking method for robot grasp.International Journal of Advanced Robotic Systems, 10(3):170.

AUTHOR

Salma Kammoun Jarraya received a Ph.D in Computer Science from Sfax University, Tunisia. She is a researcher in the MIRACL laboratory (Multimedia, InfoRmation systems and Advanced Computing Laboratory). Currently, she is Assistant Professor in computer science, CS Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA. Her research interests include computer vision, video and image processing. She has served on technical conference committees and as reviewer in many international conferences and journal