Expert Systems with Applications 42 (2015) 7263–7275

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

# An integrated system for vehicle tracking and classification

Sebastiano Battiato [a], Giovanni Maria Farinella [a], Antonino Furnari [a,*], Giovanni Puglisi [b], Anique Snijders [c], Jelmer Spiekstra [c]

[a] University of Catania, Department of Mathematics and Computer Science, Italy
[b] University of Cagliari, Department of Mathematics and Computer Science, Italy
[c] Q-Free, The Netherlands

ARTICLE INFO

*Article history:*
Available online 29 May 2015

*Keywords:*
Traffic monitoring
Vehicle tracking
Vehicle classification
Data-driven
Template matching

ABSTRACT

We present a unified system for vehicle tracking and classification which has been developed with a data-driven approach on real-world data. The main purpose of the system is the tracking of the vehicles to understand lane changes, gates transits and other behaviors useful for traffic analysis. The discrimination of the vehicles into two classes (cars vs. trucks) is also required for electronic truck-tolling. Both tracking and classification are performed online by a system made up of two components (tracker and classifier) plus a controller which automatically adapts the configuration of the system to the observed conditions. Experiments show that the proposed system outperforms the state-of-the-art algorithms on the considered data.



## 1. Introduction

Video traffic monitoring is a popular application domain in Computer Vision. In this context, algorithms are often designed to detect, re-identify, count, track or classify vehicles (Bas, Tekalp, & Salman, 2007; Hsieh, 2006; Zhou, Gao, & Zhang, 2007), while others are designed to improve the safety of the driver (Aouaouda, Chadli, Boukhnifer, & Karimi, 2014; Aouaouda, Chadli, & Karimi, 2014; Dahmani, Chadli, Rabhi, & El Hajjaji, 2013; Saifia, Chadli, Karimi, & Labiod, 2014). We tackle the specific tracking and classification tasks presenting a unified system for the online tracking and classification of vehicles. The system has been designed and tested to work with real-world data acquired by Q-Free[1] and can be used for a series of traffic-related applications ranging from road charging to law enforcement, electronic toll collection and truck tolling.

Many approaches to visual object tracking are available in the literature (Arnold et al., 2013; Maggio & Cavallaro, 2011). Each strategy is formulated by making assumptions on the application domain and choosing a suitable object representation and a frame-by-frame localization procedure. A method to update the target representation during the tracking is usually required, especially when the target is subject to geometric and photometric transformations (pose changes, deformations, illumination changes, etc.) (Maggio & Cavallaro, 2011). The most straightforward approach is probably the Template Matching, where the object is assumed to be rigid and it is represented as an image patch (the template) (Maggio & Cavallaro, 2011; Yilmaz, Javed, & Shah, 2006). If no pose changes are considered, the object is searched in the neighborhood of the last known position by maximizing a chosen similarity function (e.g., Sum of Squared Differences (SSD), Normalized Cross Correlation (NCC), etc.) between the template and candidate image patches. If pose changes are considered, the Lucas–Kanade affine tracker can be used (Baker, Gross, Ishikawa, & Matthews, 2004; Lucas & Kanade, 1981). In this case the pose changes are modeled as a set of affine transformations and the target is localized by estimating the transformation parameters which maximize the Sum of Squared Differences (SSD) between the template and the transformed version of the candidate. In other approaches the object is represented as a set of local feature points which are tracked independently (Maggio & Cavallaro, 2011; Tomasi & Kanade, 1991). This allows the algorithm to naturally deal with the object deformations since no global rigid coherence is required among the key-points. In order to track each key-point, the sparse optical flow can be computed assuming that the changes of the pixel intensities are about entirely due to motion and not to possible lighting changes in the

* Corresponding author.
*E-mail addresses:* battiato@dmi.unict.it (S. Battiato), gfarinella@dmi.unict.it (G.M. Farinella), furnari@dmi.unict.it (A. Furnari), puglisi@unica.it (G. Puglisi), anique.snijders@q-free.com (A. Snijders), jelmer.spiekstra@q-free.com (J. Spiekstra).

[1] Q-Free (http://www.q-free.com/) is a global supplier of solutions and products for Road User Charging and Advanced Transportation Management having applications mainly within electronic toll collection for road financing, congestion charging, truck-tolling, law enforcement and parking/access control.

http://dx.doi.org/10.1016/j.eswa.2015.05.055
0957-4174/

scene (brightness constancy assumption Horn & Schunck, 1981). The Lucas–Kanade optical flow algorithm (Lucas & Kanade, 1981) is often used to compute the optical flow. It requires the key-points to satisfy both spatial and temporal coherence constraints. In Tomasi and Kanade (1991) it is stated a criterion to choose which points may be selected as key-points in order to improve the performances of the tracker (specifically corners or points taken from a highly textured area of the image). In some cases the set of feature points can be directly "tracked" for specific application contexts (e.g., video stabilization (Battiato, Gallo, Puglisi, & Scellato, 2007), human computer interaction (Farinella & Rustico, 2008), traffic conflict analysis (Battiato, Cafiso, Di Graziano, Farinella, & Giudice, 2013)). In Comaniciu, Ramesh, and Meer (2003a) and in Bradski (1998) the object is represented by describing the image region in which it is contained as a n-bins histogram in the hue feature space. The object is then localized by maximizing a similarity function between the object representation of the current frame and the representation of the target candidate with respect to its position. In Bradski (1998) the CAMShift algorithm is proposed. A probability image is built back-projecting the target object hue histogram onto the current frame in order to obtain a map of the most probable object positions. The object is localized searching for local maxima of the probability map in the neighborhood of the last known position of the target using the Mean-Shift procedure (Comaniciu, Ramesh, & Meer, 2003b; Fukunaga & Hostetler, 1975). In Comaniciu et al. (2003a) the Kernel Based Object Tracking method is presented. A similarity measure based on the Bhattacharyya coefficient is derived. This measure provides a similarity score between the representation of the target object and the one of the candidate found at a given position. The localization is performed by maximizing the similarity measure with respect to the target candidate position using the Mean-Shift procedure. Other methods consider an extended appearance model and solve the tracking task as a classification problem: Arnold et al. (2013), Kalal, Matas, and Mikolajczyk (2009, 2012) and Hare, Saffari, and Torr (2011). In Kalal et al. (2012) TLD is proposed, a hybrid approach capable of tracking the object, learning its appearance and detecting it after its eventual disappearance from the scene. The tracker component is a Lucas–Kanade based tracker which tracks a set of feature points obtained using a regular grid constructed over the target object. The trajectory in the feature space is modeled by two parallel processes that extend and refine an online model (the learning component). A detector component runs in parallel with the tracker in order to enable re-initialization after its failure.

This paper is the extension of our previous work (Battiato et al., 2014) where a first version of the algorithm for vehicle tracking was presented. Here we discuss the tracking algorithm in more details and provide a comparative analysis with respect to the state-of-the-art. Moreover we add a module for online vehicle classification and integrate the two components into a unified system for traffic monitoring purposes.

The rest of the paper is organized as follows: in Section 2 we provide an overview of the system and analyze the reference data. Sections 3 and 4 present the proposed tracking and classification components respectively. In Section 5 we discuss the controller component. In Section 6 we report the experimental settings and discuss the results. Section 7 draws the conclusions.

## 2. System overview and reference data

The goal of the proposed system is to correctly track the vehicles during their transit through the road. We also want to classify the vehicles into two main classes: tall vehicles (e.g., trucks, buses, etc.) and short vehicles (e.g., cars, vans, etc.). We assume that the detection of the vehicles is performed by an external module based on plate detection and recognition plus background/foreground segmentation.[2] Both tracking and classification are performed online on real-world data. The system is composed of two main components: a tracker and a classifier. The tracker is based on template matching and is augmented with four additional modules tailored to cope with the specific variabilities exhibited by the data. The classifier is based on a supervised machine learning technique trained on a dataset containing both real and artificial examples in order to consider a number of variabilities during the learning process. A controller is introduced to optimize the performances of the tracker by turning the modules on or off on the basis of the feedback received by them. Fig. 1 shows the overall schema of the proposed system. The tracker component consists of four modules plus the classic template matching technique which is used to obtain an initial estimate of the bounding box of the vehicle. The output of the tracker component is used to update the template and to keep track of the vehicle position. The classifier component extracts an image patch of the vehicle from the current frame using the estimated bounding box. This is done once in the whole vehicle transit as explained in Section 4. Finally, the controller optimizes the performances of the tracker component by enabling or disabling each module on the basis of the estimated trajectories (current and past positions) and the predicted class.

The overall components have been designed using a data driven approach, hence an analysis of the application context is necessary before discussing the details of the developed system in the next sections. The reference data consists in video sequences related to real video traffic monitoring which have been acquired by Q-Free. The sequences exhibit high variability in terms of lighting changes, contrast changes and distortion.

Specifically the input data are the result of a preprocessing stage on sequences originally acquired through cameras mounted on the top of the road. The preprocessing stage produces a normalized, low resolution representation of the scene where the distance between neighboring pixels is constant in the real world. An example of the preprocessing results is shown in Fig. 2. The sequences have been acquired in different places and under different lighting, weather and environment conditions and are identified by a keyword summarizing the main variabilities that the system should cope with, namely: Low Contrast, Light Changes, Leading Shadows, Stop And Go + Turn, Rain and Stop And Go. These sequences are considered for both tracking and classification. Three more sequences are introduced for classification purposes only and are identified by the keywords: Sequence 1, Sequence 2, Sequence 3. These sequences are useful to learn new variabilities for the classification and allow to get a larger number of examples of tall vehicles. In order to perform quantitative evaluations, the sequences have been manually labeled, annotating for each vehicle transit the number of the starting frame, the initial bounding box, the number of the final frame and the vehicle class. Specifically each transit $T_i$ is associated to a label $l_i$, where $l_i = 1$ for the short vehicles (e.g., cars), while $l_i = 2$ for the tall vehicles (e.g., trucks).

Table 1 shows the number of vehicles which have been labeled in each sequences and the corresponding classes. It should be noted that the ratio between the number of tall vehicles and the number of short ones is approximately equals to $1:5$, while ideally we would like to work with a balanced (i.e., $1:1$ ratio) set. The effects of using a balanced dataset are discussed in Section 6. The overall data contain 1208 vehicle transits in total.

In analyzing the application context, we highlight the following relevant characteristics of the reference data:

[2] The plate detection and recognition module is already commercialized by Q-Free.