Vehicle Tracking based on Customized Template Matching

Sebastiano Battiato[1], Giovanni Maria Farinella[1], Antonino Furnari[1], Giovanni Puglisi[1], Anique Snijders[2] and Jelmer Spiekstra[2]
[1]*Università degli Studi di Catania, Dipartimento di Matematica e Informatica, Catania, Italy*
[2]*Q-Free, Beilen, Netherlands*

Keywords: Vehicle Tracking.

Abstract: In this paper we present a template matching based vehicle tracking algorithm designed for traffic analysis purposes. The proposed approach could be integrated in a system able to understand lane changes, gate passages and other behaviours useful for traffic analysis. After reviewing some state-of-the-art object tracking techniques, the proposed approach is presented as a customization of the template matching algorithm by introducing different modules designed to solve specific issues of the application context. The experiments are performed on a dataset compound by real-world cases of vehicle traffic acquired in different scene contexts (e.g., highway, urban, etc.) and weather conditions (e.g., raining, snowing, etc.). The performances of the proposed approach are compared with respect to a baseline technique based on background-foreground separation.

1 INTRODUCTION

Object tracking strategies are formulated by making some assumptions on the application domain and choosing a suitable *object representation* and a frame-by-frame *localization* method. The object representation is usually updated during the tracking, especially when the target object is subject to geometric and photometric transformations (object deformations, light changes, etc.) (Maggio and Cavallaro, 2011).

In the *Template Matching* based strategies (Maggio and Cavallaro, 2011; Yilmaz et al., 2006), the object is represented as an image patch (the template) and is usually assumed to be rigid. In the simplest settings, the object is searched in a neighbourhood window of the object's *last known position* by maximizing a chosen similarity function between image patches. When target changes of pose are considered, the *Lucas-Kanade* affine tracker can be used (Lucas et al., 1981; Baker and Matthews, 2004). In the *Local Feature Points* based strategies (Tomasi and Kanade, 1991) the object is represented as a set of *key-points* which are tracked independently by estimating their *motion vectors* at each frame. In order to track each *key-point*, a *sparse optical flow* is usually computed considering the (*brightness constancy assumption* (Horn and Schunck, 1981)). The *Lucas-Kanade Optical Flow* (Lucas et al., 1981) algorithm is often used to compute the *optical flow* and requires the *key-points* to satisfy both *spatial* and *temporal* coherence. In some cases the set of feature points can be directly "tracked" for specific application contexts (e.g., video stabilization (Battiato et al., 2007), human computer interaction (Farinella and Rustico, 2008), traffic conflict analysis (Battiato et al., 2013)). In the *Region Based* techniques (Comaniciu et al., 2003) the object is represented by describing the image region in which it is contained as a quantized probability distribution (e.g., a *n-bins* histogram) with respect to a given *feature space* (e.g., the hue space). In (Bradski, 1998) the CAMShift algorithm is proposed and it is suggested to build a probability image 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 finding the probability image relative peak in the neighbourhood of the last known position using the *Mean-Shift* procedure (Comaniciu and Meer, 2002). In (Comaniciu et al., 2003) a similarity measure is derived based on the Bhattacharyya coefficient providing a similarity score between the target object representation and the one of the candidate found at a given position. By using the *Mean-Shift* procedure(Comaniciu and Meer, 2002), the similarity measure is maximized with respect to the target candidate

Battiato S., Farinella G., Furnari A., Puglisi G., Snijders A. and Spiekstra J..
Vehicle Tracking based on Customized Template Matching.
DOI: 10.5220/0004872607550760
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (PANORAMA-2014), pages 755-760
ISBN: 978-989-758-004-8

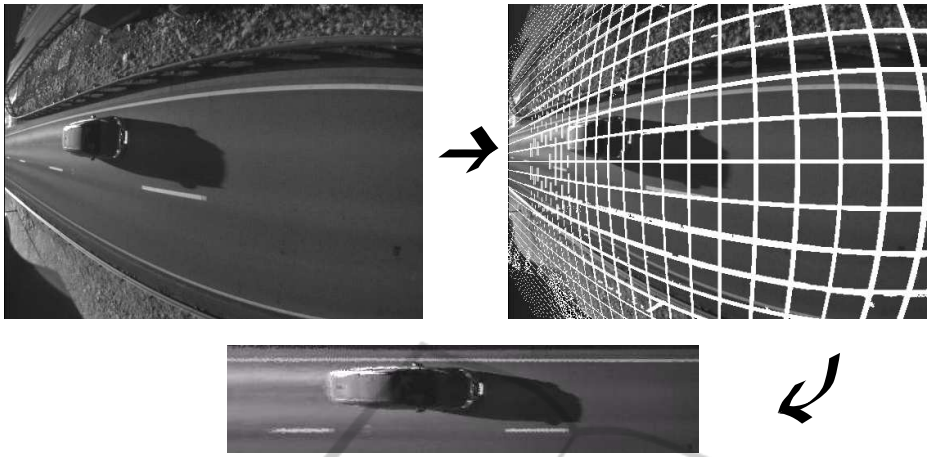


Figure 1: Preprocessing stage and generation of a normalized representation of the scene where the distance between neighbouring pixels is constant in the real world.

In this paper we present a customized vehicle tracking algorithm based on template matching. The proposed algorithm is tested on real video sequences which are characterized by high variability in terms of perspective, light and contrast changes, object distortion and presence of artefacts. The input sequences are the result of a preprocessing stage which filters out the camera distortion. An example of such preprocessing is reported in Figure 1. In designing the proposed algorithm the data have played the main role. In this paper we report the rationale beyond the build method making connections between the adopted strategies and the real video sequences.

The remainder of the paper is organized as follows: in Section 2 the reference video data are discussed, whereas Section 3 presents the proposed approach. Section 4 describes the experiments and the way we have measured the performances of the algorithms. Finally, Section 5 reports the conclusions and the directions for future works.

2 APPLICATION CONTEXT AND REFERENCE DATA

The goal of our work is to correctly track each vehicle from the beginning of the scene to the end, assuming that an external detection module based on plate recognition gives to us the position of the front part of the vehicle in the first frame in which the plate is detected. The dataset used in the experiments consists of six video sequences related to real video traffic monitoring which have been acquired by Q-Free.[1] 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 which produces a normalized, low resolution representation of the scene where the distance between neighbouring pixels is constant in the real world (Figure 1). 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 characteristic that the tracker should deal with, namely: LOW CONTRAST, LIGHT CHANGES, LEADING SHADOWS, STOP AND GO + TURN, RAIN and STOP AND GO. The overall sequences contain 1168 vehicle transits in total.

3 PROPOSED APPROACH

The proposed approach is based on the general *template matching* scheme: at the initialization step, assuming that the plate detection and recognition module returned the current vehicle position in the form of a bounding box, the template is extracted as a portion of the current frame and the object position is set to the bounding box centre; at each frame, a search window is centred at the object last known position and a number of candidates centred at each point of the search window and having the same size as the template are extracted. The object current position is then set to the one which maximizes the similarity score between the target template and the candidate one ac-

[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.

cording to a selected similarity measure; at the end of the search, the vehicle representation is updated extracting a new template at the current vehicle position. We use this general scheme (Maggio and Cavallaro, 2011) as a baseline and augment it by adding some domain-specific customizations in the form of modules which can be dynamically switched on (or off) by a controller. There are four proposed modules: *Multicorrelation*, *Template Drift and Refinement*, *Background Subtraction* and *Selective Update*.

In the following we summarize the scope of each module used to extend the basic template matching procedure providing related details. All the parameters' values are reported in Section 4.

The presence of artefacts (see Figure 2 (a)) contributes to radical changes of the vehicles' appearance between consecutive frames. In such cases the similarity between the current instance of the object and its representation can be low, thus making the tracker less accurate and possibly leading to a failure. In order to reduce the influence of the artefacts, we act as if it were an occlusion problem introducing an alternative way to compute the similarity between two image patches which is referred to as *Multicorrelation*: both the *template* and the *candidate* are divided into nine regular *blocks*. A similarity score (e.g., Normalized Cross Correlation) is so computed between each couple of corresponding *blocks* and the final score is obtained by averaging the nine subwindows similarity values. A statistical analysis of the similarity score values highlighted that when the issue shown in Figure 2 (a) arises, the *similarity measure* computed in the regular way tends to be lower than a given threshold t_m. So we use the *multicorrelation similarity measure* only when the regular *similarity score* is under the given threshold. Figure 2 (c) shows the result of the multicorrelation approach.

The presence of *light*, *perspective*, *contrast* changes and *distortion*, joined with the continuous update of the *template*, generate the *template drift* problem in the form of the progressive inclusion of the background into the *template model*. This effect is shown in Figure 2 (b).

In order to reduce the *template drift*, a *refinement* is performed at the end of the basic template matching search. The *refinement* is based on the assumption that the object is stretched horizontally by effect of the distortion introduced in the preprocessing stage (see Figure 1). According to this assumption, we adopt the following strategy: given the current frame and the template model found at the previous frame, we search for a version of the object at a smaller horizontal scale, obtaining a smaller tracking box which will be properly enlarged backward in order to fit the original template dimensions. Searching for the object at different horizontal scales would make the algorithm much slower, so, in order to improve performances, we first perform a *regular search* (i.e., without any *refinement*) in order to obtain an *initial guess*, afterwards we search for the *best match* among a number of *candidates* obtained discarding the rightmost pixels (the ones which are more likely to contain background information) and horizontally-scaled versions of the template. The results of the technique are shown in Figure 2 (d).

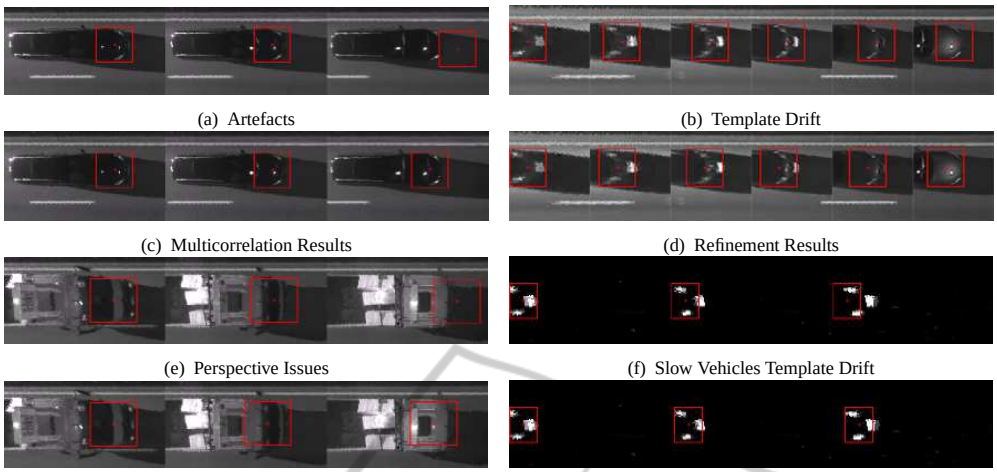
When tracking *tall vehicles*, the *perspective issue* shown in Figure 2 (e) arises: the radical change of the vehicle appearance in consecutive frames leads to the progressive inclusion of the background inside the template model up to the eventual failure of the tracker. In order to correct this behaviour, after a regular search, we perform a *background aware refinement* sliding the tracking window backward in order to remove the background pixels in the front of the tracking box through a rough *background subtraction* technique based on subsequent frames subtraction and thresholding. The results of the technique are shown in Figure 2 (g).

The continuous update of the vehicle representation induces the template drift problem in those sequences in which the vehicles move slowly. An example of this problem is shown in Figure 2 (f). Since the vehicle moves very slowly and considering that the object changes of appearance between two consecutive frames are slight, a shifted version of the template still returns a high similarity score, while the continuous update favourites the propagation of a wrong vehicle representation. In order to correct this behaviour, we update the object representation only when it is significantly different from the old one, i.e., when the *similarity score* is under a fixed threshold t_u. Figure 2 (h) shows the results of the *selective update* mechanism.

Due to the different operations involved in the specific modules, we found the performances of the modules to be dependent on the vehicle speed. In order to maximize the performances of the overall algorithm on the data, we distinguish between *high-speed* (60 km/h or more) and *low-speed* (less than 60 km/h) vehicles and introduce a *controller component* which dynamically enables or disables the modules.

4 EXPERIMENTAL SETTINGS AND RESULTS

All the experiments have been performed on the dataset described in Section 2. The sequences have



(a) Artefacts (b) Template Drift

(c) Multicorrelation Results (d) Refinement Results

(e) Perspective Issues (f) Slow Vehicles Template Drift

(g) Background Subtraction Results (h) Selective Update Results

Figure 2: The figure shows the domain specific issues (a, b, e, f) and the results of the modules introduced to deal with them (c, d, g, h).

been manually labelled annotating for each vehicle *transit*, the bounding box of the starting frame and the final frame. This information is used to initialize the proposed tracker,[2] which is then executed in the subsequent frames till the last frame of the transit is processed. After running the different compared trackers, an examination is needed to manually mark each tracked transit as "successful" or "failed". We have also manually annotated the first frame of failure. The algorithm parameters have been tuned through a statistical analysis in order to maximize the performances on the data. The *Normalized Cross Correlation* is used as *similarity measure* for template matching, the search window size is 20 $px \times 12$ px wide, in order to handle vehicles with a maximum horizontal speed of 381 km/h and a maximum vertical speed of 32 km/h. The search is performed using an asymmetrical window (forward only) in order to reduce the computation (the vehicles can only move forward or stay still). As we cannot predict an exact horizontal scaling factor, in the refinement stage, multiple *scaling factors* have to be explored. Since in the given context a scaling factor of 0.02 corresponds to less than 1 px, which is the best precision we can achieve, and considering that a statistical analysis pointed out that in most cases the best scaling factor is in the range $[0.90, 1]$, the scaling factors are taken form this range at step of 0.02. Both the *multicorrelation* and the *selective update* thresholds are set to $t_m = t_u = 0.8$.

In order to analyse the trackers performances, two evaluation methods are used:

Transit based Accuracy (TBA): focused on the ability to correctly track the vehicle in all the frames of his transit. This measure is defined as:

$$TBA = \frac{1}{N}\sum_{i=0}^{N-1} s_t(T_i) \quad (1)$$

where N is the total number of transits, $\{T_i\}_{i\in[0,N-1]}$ are the transits and

$$s_t(T_i) = \begin{cases} 1 & \text{if the tracking has no errors} \\ 0 & \text{otherwise} \end{cases} ; \quad (2)$$

Longevity based Accuracy (LBA): focused on the tracker longevity, i.e., the mean transit percentage correctly tracked before a possible failure. This measure is defined as:

$$LBA = \frac{1}{N}\sum_{i=0}^{N-1} s_l(T_i)\ . \quad (3)$$

where N and T_i are defined as above,

$$s_l(T_i) = \frac{m_i}{n_i}\ , \quad (4)$$

m_i is the number of frames in which the vehicle is tracked correctly in transit T_i and n_i is the total number of frames in T_i.

For sake of comparison we have considered the following approaches: the *CAMShift* algorithm (Bradski, 1998) gives poor results since the initialization step in the intensity domain fails. This is due to the simplicity of the image representation which doesn't ensure the maximization of the similarity measure between the target representation and the candidate one. The *Kernel-Based Object Tracking* algorithm (Comaniciu et al., 2003) succeeds in the initialization step but fails in the tracking due to the poor

[2]We assume that the bounding box is given by another module related to the plate detection and recognition already present in the systems.

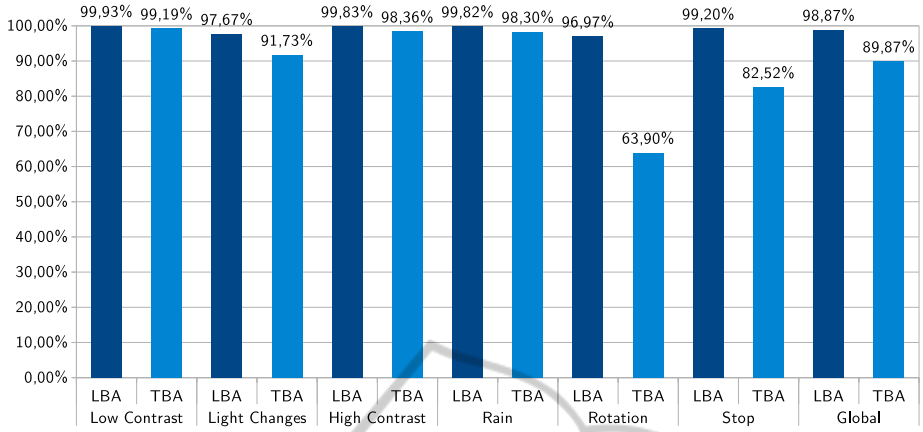


Figure 3: The results of the proposed technique on the sequences identified by corresponding keywords.

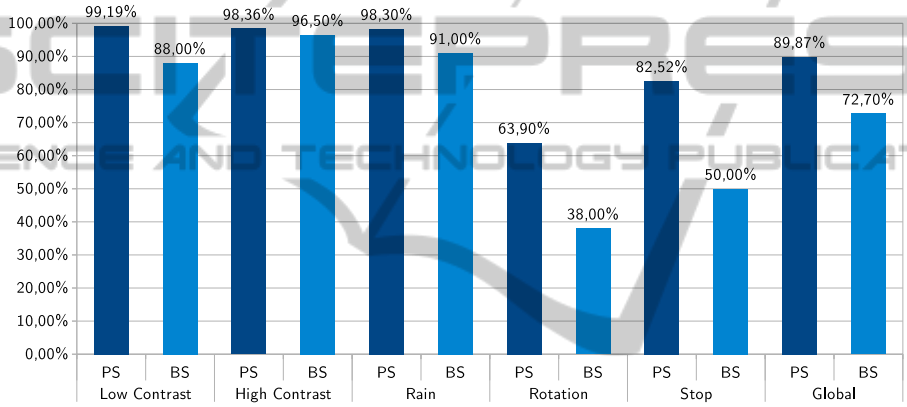


Figure 4: The results of the proposed technique (PS) vs a simple background-foreground separation pipeline (BS) according to the *TBA* measurement (see Section 4).

separation between the object and the background in the feature space (intensity values). Both *CAMShift* and *Kernel-Based Object Tracking* do fail in the *gradient orientations feature space* since the similarity measure is not a smooth function (no gradient based optimizations are possible).

Figure 3 shows the results of the proposed approach for each sequence (identified by its relative keyword as described in Section 2) and the global accuracy according to the *TBA* and the *LBA* measuring methods. The introduction of the two measuring methods can be justified observing that they measure two different qualities of the tracker. In the STOP and ROTATION sequences, it can be noticed that the *TBA* values are consistently lower than the related *LBA* values. This happens because the tracker correctly tracks the object for the most part of the scene (obtaining a high *LBA* score) systematically failing in the last frames of the transit due to poor lighting (which gives a zero-weight to the transit in the *TBA* settings). Figure 4 compares the results of our *technique* with respect to the results of a typical *background--foreground separation* pipeline based on *first order time derivative* and *gradient difference*, according to the *TBA* measurement.

5 CONCLUSIONS

In this paper we have proposed a template matching based method for vehicle tracking applications. The classical template matching algorithm has been customized to be able to cope with a series of challenging conditions related to real word sequences such as high variability in perspective, light and contrast changes, object distortions and artefacts in the scene. The effectiveness of our approach has been then demonstrated through a series of experiments in critical conditions and comparisons with respect to a baseline technique. Future work will be devoted to compare the proposed tracker with respect to recent techniques (e.g., TLD (Kalal et al., 2009)) as well so to include

a module able to discriminate among different kinds of vehicles (e.g., car, truck) in order to collect useful statistics for the traffic analysis.

ACKNOWLEDGEMENTS

This work has been performed in the project PANORAMA, co-funded by grants from Belgium, Italy, France, the Netherlands, and the United Kingdom, and the ENIAC Joint Undertaking.

REFERENCES

Emilio Maggio and Andrea Cavallaro. *Video tracking: theory and practice*. Wiley, 2011.

Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.

Bruce D. Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ., 1991.

Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

Sebastiano Battiato, Giovanni Gallo, Giovanni Puglisi, and Salvatore Scellato. Sift features tracking for video stabilization. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 825–830, 2007.

Giovanni M. Farinella and Eugenio Rustico. Low cost finger tracking on flat surfaces. *Eurographics Italian Chapter Conference 2008 - Proceedings*, pp. 43-48, 2008.

Sebastiano Battiato, Stefano Cafiso, Alessandro Di Graziano, Giovanni M. Farinella, and Oliver Giudice. Road traffic conflict analysis from geo-referenced stereo sequences. *International Conference on Image Analysis and Processing, Lecture Notes in Computer Science LNCS 8156*, pp. 381-390, 2013.

Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.

Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. Online learning of robust object detectors during unstable tracking. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1417–1424. IEEE, 2009.