Abstract—Understanding the intention of other road users is a key requirement for autonomous driving. In this regard, one particularly relevant cue is a flashing turn signal, since it gives an important hint regarding the intended driving direction of another vehicle in the next few seconds. As such, turn signals can be considered as one of the first methods invented for car-to-car communication. In contrast to modern radio-based approaches, turn signals are installed in almost every vehicle. However, only image-based methods are able to detect, recognize and understand those signals.

In this paper, we present a new method to recognize turn signals of other vehicles in images. Our approach builds upon a robust vehicle detector and involves three major steps applied to each detected vehicle: light spot detection, feature extraction through FFT-based analysis of the temporal signal behavior at each detected light spot, and AdaBoost classification of the extracted feature set.

In our experiments, we use solely virtually-generated data for training and evaluate the proposed approach on a large 30 minute real-world image sequence. Our results indicate competitive performance at real-time speeds.

I. INTRODUCTION

In daily traffic, turn signals are amongst the most important indicators to understand and predict the intention of other road users. A flashing turn signal notifies other traffic participants of a potentially critical situation, such as lane changes, turn maneuvers, or the end of a traffic jam. Given that humans are capable of recognizing abrupt changes in the environment very quickly, locating and understanding turn signals does not pose a significant problem under regular lighting conditions. However, developing a robust machine vision approach to tackle that particular problem is a very complex task. This complexity mainly arises from the requirement to detect a very local and small illumination change in fast moving traffic scenarios at long distances with imperfect in-vehicle camera sensors.

Besides approaches that focus on the detection of temporally-consistent rear and brake lights, e.g. [1]–[3], there is not much published work on the recognition of flashing turn signals, to the best of our knowledge: Li et al. [4] divide detected cars into a left and a right side. On each side, a threshold is applied to detect and count the number of amber pixels. Based on this number, the turn signal is defined as either on or off.

A more sophisticated method is presented by Almagambetov et al. [5]. Here, light pairs on detected cars are located and tracked using a Kalman filter. When a significant change in luminance is detected, an analysis determines whether both sides of the car changed the luminance (they assume the brake lights are on) or only one side (turn signal is on). Unfortunately, they have a very limited test set containing only two cars that use a turn signal.

In contrast to previous approaches, we advocate to additionally take the frequency of illumination changes into account. This turn signal frequency is defined in paragraph 6.5.9. of the Regulation No 48 of the Economic Commission for Europe of the United Nations (UN/ECE) with $90 \pm 30$ times per minute [6], which is $1.5 \pm 0.5$ Hz. Assuming a frequency of $1.5$ Hz, one flashing interval lasts $\frac{2}{3}$ s. To be both fast and robust, we aim to recognize a flashing turn light no later than after three flashes. Thus, we base our analysis on a temporal window of two seconds. As shown in Figure 1, our presented approach consists of five successive steps: In this work, we present a new method for locating and classifying turn lights in traffic scenes.

1) Locate objects (cars, trucks, motorcycles) in the current...
Fig. 2. Long-range vehicle detection and tracking results.

frame
2) Detect light spots on each located object
3) Extract descriptor: the history of the intensity at the
located light spots
4) Transform descriptor (i.e. FFT)
5) Classify each descriptor and stabilize the results over
time

Robust object detection and localization is the foundation
for the proposed method. We do not focus on this problem
in this paper and do not require any particular object detection
method among many possible approaches, e.g. [7]. For
vehicle detection and tracking, we follow a recent approach,
as presented in [8]. The light sources are located using a
Gaussian mixture model as clustering method. As feature
descriptor, we use the discrete Fourier transform of the signal
at the light source of the past few seconds. Finally, AdaBoost
is used for classification.

The remainder of this paper is organized as follows. First,
we show how to get a localization of the vehicles in an
image. In the same section, we present how to remove the
background and how to stabilize the tracked vehicle over
time. Subsequently, the feature extraction and transformation,
and the classification is described. Finally, we evaluate our
proposed method on a 30 minutes sequence on a highway
with more than 2000 vehicles.

II. OBJECT LOCALIZATION AND PREPROCESSING

In this section, we first describe how the objects (vehicles)
are roughly located. Note that the method presented in this
paper is rather independent from the actual method used
for object detection. Furthermore, we show the background
elimination and finally the stabilization of the tracked object.

A. Vehicle detection and tracking

For vehicle detection and tracking, we apply the approach
proposed in [8] at distances of up to 200 m from the ego-
vehicle. In such a long-range scenario, precise depth and
velocity estimation is very difficult due to large disparity
noise, given our camera setup. Hence, we cannot apply
stereo-based ROI generation, e.g. [9] and rely on a very
fast monocular vehicle detector, i.e. a Viola-Jones cascade
detector [10]. Its main purpose is to create regions-of-interest
for subsequent strong Mixture-of-Experts classifiers [11],
that are applied as a verification step to every detection of the
cascade detector. Hence, we can easily tolerate the inferior
detection performance of the Viola-Jones cascade framework
compared to state-of-the-art, and exploit its unrivaled speed
as an initial detection module.

Each detection of the cascade detector is classified by
powerful multi-cue object classifiers. Here, we are using a
Mixture-of-Experts scheme that operates on a diverse set of
image features and modalities inspired by [11]. In particular,
we couple gradient-based features such as histograms of
oriented gradients (HoG) [12] with texture-based features
such as local binary patterns (LBP) or local receptive fields
(LRF) [13]. Furthermore, all features operate both on gray-
level intensity as well as dense disparity images to fully
exploit the orthogonal characteristics of both modalities [11].
Multiple classifier responses at similar locations and scales
are addressed by applying mean-shift-based non-maximum
suppression to the individual detections.

Precise distance and velocity estimation of detected ob-
jects throughout the full distance range poses extreme de-
mands on the accuracy of stereo matching as well as camera
 calibration. In order to obtain optimal disparity estimates and
temporal correspondences, we perform an additional careful
correlation analysis both spatially (left and right image) and
temporally. An example result for detection and tracking is
shown in Figure 2.

B. Background subtraction

The area around a localized vehicle changes over time.
Due to the fact that we focus on objects during a specific time
interval, undesirable effects like the illumination adjustment
in between sky and trees in an alley, can lead to false positive
results. Therefore, it is reasonable to extract the background.
Since we use a stereo camera system, we are able to exclude
pixels with outlying depth information in the bounding box.
A disparity map is computed by applying dense semi-global
matching [14]. All pixels further than one meter away from
the median of all pixels in a patch are marked as background
and excluded for all further steps. Figure 3 demonstrates
the foreground segmentation for a given input and disparity
image.

C. Track stabilization

For the following steps, we need a robust track of each car.
Due to the tiny size of some turn lights, a small shift of only
a few pixels over time leads to significant different features
Fig. 4. Image stabilization for an image pair. Using the feature correspondences in the two images of the same object at different times, the transformation maps one image to another. Figures 4f and 4g visualize the quality of the mapping by applying the absolute difference of the input patches and of the first input patch and the transformed second input patch, respectively.

and consequently to a wrong classification (cf. Section III and Section IV). For computing correspondences between points in multiple images, we use the well-known Kanade-Lucas-Tomasi feature tracker (KLT) from [15]. With these correspondences, we can compute the transformation $T_i(P_j)$ between two image patches $P_i$ and $P_j$. We only consider translation, rotation and scale changes. Therefore, we can describe $T_i$ as a matrix multiplication using homogeneous coordinates for each pixel $p_j$ of patch $P_j$:

$$p'_j = T_i(p_j) = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \cdot p_j .$$

The transformation $T_i$ is estimated by using the RANSAC algorithm for each relevant image pair [16]. An example stabilization is shown in Figure 4.

III. FEATURE CALCULATION

In this section, we describe how to locate light spot candidates. Furthermore, we show how to extract the features and how to transform a feature descriptor to obtain a representation of the frequencies in the signal.

A. Light spot detection

We assume an active turn light as the light source, represented by a spot of light in the image. Therefore, we are interested in detecting such light spots on the located objects. To do so, we compute the absolute difference image between the located object at the current frame and the same vehicle half of the flashing interval previously. A Gaussian mixture model is used as the clustering method on this difference image. We assume a flashing frequency of 1.5 Hz, corresponding to an interval length of $\frac{2}{3}$ s. For a faster and more stable computation, we compute only the diagonal covariance matrix and use a MAP estimation instead of an ML estimation (following Perronnin et al. [17]). If the variance of a cluster is too large, it is split into two new clusters with slightly different means and the expectation-maximization algorithm is restarted with those values.

On an ideal difference image there should only be significant clusters if a light is flashing, otherwise there should be nothing. Unfortunately, the background subtraction and the shift correction is not optimal in every case, which results in locating some additional spots like brake or position lamps, reflections of the sun, or rapid background changes. Those kinds of false positives are eliminated later on in the classification step Section IV.

B. Feature extraction

A feature descriptor is calculated for each detected light spot. First, pixels belonging to the current spot are segmented by grouping similar pixels in the neighborhood of the cluster center. Second, for each frame of the past $t$ seconds the average intensity of all those pixels is computed. We assume a frequency of 1.5 Hz and consider the last $t = 2$ s (the last 3 intervals). With this and a frame rate of 16.6 fps we get a descriptor with 33 entries. To get a more robust description of the signal, we also use the signal of the last interval and the two last intervals. Finally, we have three descriptors with sizes 11, 22, and 33, respectively. An example signal for $t = 2$ s is shown in Figure 6.

C. Feature transformation

In the next step, we transform each of the three feature descriptors using the discrete Fourier transform to represent the signal in the frequency domain. To reduce the side effects (boundary treatment) of transforming a finite input signal $s$ with length $|s|$ into the frequency domain, it is weighted using a Hann window $w$ (raised-cosine window) [18]

$$s'(t) = w(t) \cdot s(t) ,$$

Fig. 5. Regions of interest of the same object in the last two seconds.
with $t \in 1 \ldots |s|$ and

$$w(t) = 0.5 \left( 1 - \cos \left( \frac{2 \pi t}{|s| - 1} \right) \right).$$

(3)

The discrete Fourier transform $f$ of the signal $s'$ is defined as:

$$f(k) = \sum_{t=0}^{|s'|-1} e^{-2 \pi i \frac{k}{|s|} \cdot s'(t)}.$$  

(4)

For a faster computation, we apply the fast Fourier transform (FFT). Figure 7 shows the Fourier transform of the signal from Figure 6. Finally, all three Fourier transformed signals are concatenated to one feature vector as input for the following steps (cf. Section IV).

### IV. Classification

In this section, we describe our training data, the AdaBoost classifier, and a temporal stabilization of the classification results.

#### A. Training data generation

Labeling a sufficient amount of training data for turn signal recognition is very difficult. We recorded some sequences on German highways and evaluated the occurrence of flashing vehicles. On average, less than three vehicles per minute use their turn signal. For an adequate training set of about 10000 flashing turn lights, we need at least 50 hours of road scenes with each turn light accurately labeled. Due to this effort, we artificially generate a large amount of training data by synthesizing the signal. Since we do not know whether the light illumination behavior is sinusoidal

$$s_{\sin}(t) = \cos(2 \pi \phi (t - t_0)),$$

(5)

or a square wave signal

$$s_{\text{rect}}(t) = \begin{cases} 1 & \text{if } s_{\sin}(t) \geq 0, \\ -1 & \text{else} \end{cases},$$

(6)

with the frequency $\phi = 1.5 \text{Hz} \pm 0.1 \text{Hz}$ and a shift $t_0$, we use both as positive examples. Finally, some Gaussian noise is added to the signal. As negative examples, we generate some random signals and also sinusoidal and square wave signals which do not fulfill the previous mentioned conditions. All those signals are also transformed as shown in Section III-C. Figure 8 shows some positive random generated feature descriptor examples.

#### B. AdaBoost

Due to its fast performance, we choose AdaBoost [19] for classification. This method linearly combines $a \in m$ different weak classifiers $c_a$ into a single strong one $H$ (called a cascade)

$$H(f) = \sum_{a=1}^m w_a \cdot c_a(f),$$

(7)

in which $w_a$ is the weight for the weak classifier $c_a$. In our case a weak classifier $c_a$ is a simple threshold for any feature to split the feature space into two disjunct sets.
To train an AdaBoost cascade, the best split is iteratively estimated by evaluating all possible splits. The evaluation criterion for each split $j$ is defined by the minimum error $\epsilon_a$

$$\epsilon_a = \sum_i \omega_{a,i} |c_{a,j}(f_i) - y_i|,$$  

for all descriptors $f_i$ and the corresponding ground truth label $y_i$. The weights $\omega_i$ for each descriptor are equally distributed for the first iteration and updated after each iteration by

$$\omega_{a+1,i} = \omega_{a,i} \cdot \beta_a^{1 - e_i},$$

with

$$\beta_a = \frac{\epsilon_a}{1 - \epsilon_a},$$

and $e_i = 0$ if the descriptor $f_i$ is classified correctly or $e_i = 1$ otherwise.

As an advanced modification, one can use a soft cascade of weak classifiers as in Bourdev et al. [20], which is much more efficient for unbalanced scenarios.

C. Temporal stabilization

To stabilize the classification output and to reduce the amount of false positive detections, we apply temporal smoothing. Therefore, we analyze the classification results of the last interval ($\frac{2}{3}$ s) for the current object. If there is a flashing light detected in more than 50%, the final returned state of our method is “flashing left” or “flashing right,” respectively. In all other cases the state “not flashing” is returned.

V. Experiments

To evaluate our method, we use a 30 minute long color sequence of a highway in Germany with a frame rate of 16.6 fps and a resolution of 1176 $\times$ 640 pixels. Our object localization method finds 2656 different objects. 53 out of those 2656 vehicles are using their turn signal. Due to long distances, perspective, and occlusion, 27 cars use their turn signal, but are not recognized and tracked by our object localization method.

The evaluation measure is defined as follows. If our method recognizes a turn signal at least once for an object in the correct direction while the ground truth data assigns a flashing turn signal to the same object, we count it as a true positive. Every time our method recognizes a turn signal and no turn light is flashing with respect to the ground truth, we count it as a false positive.

In Figure 9, we show the results for multiple decision functions of the AdaBoost classifier. For further analysis, we concentrate on two points of the graph:

- Scenario 1: the best result without any false positives and
- Scenario 2: the best recognition rate with 26 false positive detections in total (i.e. 0.87 false positives per minute)

Both scenarios are marked in Figure 9 by a green circle. We achieve recognition rates between 64.15% for Scenario 1 and up to 86.79% for Scenario 2. Due to the fact that our method is not frame- but time-based, we decided to present the false positive rate in detected events per minute.

To our surprise, the quality of the recognition does not correlate with the distance of the vehicle using the turn signal. Figure 10a shows the result for Scenario 1. All events between 30 m and 40 m and almost all events between 100 m and 150 m are well recognized. On the other side events less than 20 m and between 50 m and 60 m are poorly classified. Furthermore, Figure 10b shows the results for Scenario 2, in which the missed positives are almost equally distributed over the different distances. Please note that the higher miss rate for objects closer than 20 m are due to the fact that those vehicles are often completing an overtaking maneuver.

Much more important than the distance is the amount of flashing intervals. As shown in Figure 11a for Scenario 1 we have problems with vehicles flashing only once or twice (only 25% recognition). The results for vehicles using the turn signal exactly three times are much higher (60%) and for more than three times almost every flashing signal is found (93%). In average, we recognize an event after 1.55 s after the turn signal starts to flash the first time. For Scenario 2, we come to a similar observation. Figure 11b shows that 62.5% of vehicles flashing only once or twice, 86.7% of vehicles flashing exactly three times, and all other vehicles are recognized. For Scenario 2, we recognize a turn signal much faster, in an average time of 1.21 s.

VI. Conclusion

In this work, we presented a novel method for locating and classifying vehicle turn signals. Besides the stable localization of vehicles, we have shown how to detect light spots. Furthermore, we described how to extract and classify features to distinguish flashing turn signals from other light spots.

In our experiments on a huge dataset, we have demonstrated that our method can be used in practical scenarios without many false positive results. Furthermore, we have shown that our method is independent of the distance from the analyzed vehicle. However, the results greatly depend on the number of flashes by the turn signal. If the turn signal
Since many lights in the images appear as simply huge white spots, a camera with higher color fidelity would be very useful, and for further work, it is recommended to find a faster and more robust method for the light spot detection. Furthermore, a camera with higher color fidelity would be very useful, since many lights in the images appear as simply huge white spots.

**References**


