Low-Complexity Driving Event Detection from Side Information of a 3D Video Encoder

Low-Complexity Driving Event Detection from Side Information of a 3D Video Encoder

Ruiliang Wang, Yang Li, Enrico Masala

Control and Computer Engineering Department, Politecnico di Torino
corso Duca degli Abruzzi 24 — 10129 Torino, Italy
ruiliang.wang@studenti.polito.it, yang.li@studenti.polito.it, masala@polito.it

***Abstract*—Mobile phones are often found in cars, for instance when they are used as navigation assistants. This work propose to use their camera, which is often already pointed to the road, to perform some low-complexity analysis of the driving context, with the final aim to detect potentially unsafe conditions. Since content understanding algorithms are typically too complex to run in real time on a mobile device, a driving event detection algorithm is presented based on the side information available from video encoders, which are a highly optimized application in mobile phones. A set of interesting and easy-to-extract features has been identified in the side information and then further reduced and adapted to the specific events of interest. A detection algorithm based on support vector machines has been designed and trained on several hours of video annotated by a human operator to extract the events of interest. The detection algorithm is shown to achieve a good identification rate for the considered events and feature sets. Moreover, results also show that the use of a stereoscopic camera significantly improves the performance of the detection algorithm in most cases.**

I. Introduction

Most of the people carry mobile phones with them at all times, including when they are in a car. Sometimes, mobile phones are also used as navigation assistants to provide directions. In this case, the device is almost always placed on a support attached to the dashboard or the windscreen. Therefore, it would be extremely interesting if such a device, with the rear camera typically pointing to the road, could be used to detect the driving context and alert in real time the driver about potentially unsafe conditions.

Although the idea is interesting in theory, in practice many algorithms developed for content understanding in such conditions (see, e.g., [1]) may be too computationally demanding to run on the mobile phone in real time, since the processing power of mobile devices is typically limited.

Despite those limitations, a particular family of heavy algorithms, i.e., video coding algorithms, can typically be run on mobile phones since the CPU and other specialized hardware has been optimized to run them in real time, so that users can record their video using such devices.

This work argues that some of the information produced by running differential encoding with motion compensation algorithms, which are the basis of all the widespread video compression techniques, could be used to gain some insight in the semantic of the scene when applied to a video captured from a camera pointed to the road. Even more, if stereoscopic devices, already commercially available on the market as smartphones, are used to capture the images, much better performance can be achieved since a second viewpoint provides additional information that cannot be easily detected from one viewpoint. Note also that, typically, the cost of mobile phones with stereoscopic capabilities is comparable with many non-stereoscopic good-quality smartphones, which many people are eager to buy in any case.

The underlying idea of this work is that the motion estimation process produces enough information to get at least a basic understanding of the driving context, so that the driver can be alerted in case of specific situations happen. Some preliminary results already showed that it is possible, in principle, to detect simple situations for the case of monoscopic video [2]. In this work we extend the approach to a larger and more interesting set of events and we investigate the possibility to use a stereoscopic camera, that we expect to be increasingly common on future mobile devices, to improve the accuracy of the detection algorithms. The context information is extracted by means of a classification algorithm, i.e., a linear support vector machine (SVM) [3], [4], with minimum complexity, using the motion vectors as input features.

Moreover, we also show that in addition to motion vectors, other features can also contribute to improve accuracy, such as the total frame size and encoding distortion, as well as the characteristics of each macroblock, such as the number of motion vectors that have been used to encode it or the distortion of each macroblock with respect to the original video sequence.

Note also that, although future cars, especially the more expensive ones, are expected to include an increasing number of cameras to support the driver for specific tasks, the presented solution is suitable for any car regardless of the on board sensors, since it can run on the mobile device without any additional support.

Several hours of stereoscopic video have been collected using a mobile phone attached to the windscreen of a car driving in different environments, e.g., urban, highway and motorway. Then, sequences have been manually annotated to identify interesting events in the video. This database has then been used to train the SVM-based algorithm to identify the

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00 ©*2013 IEEE.*

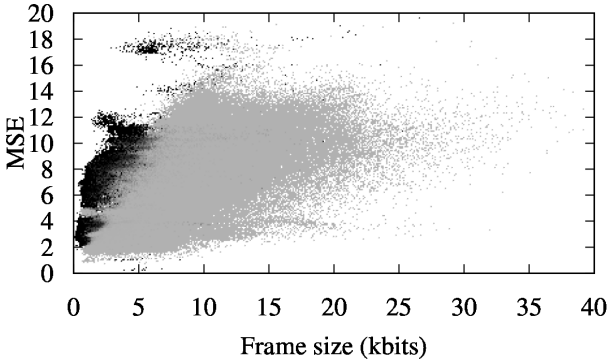

Fig. 1. Scatter plot of the frame size and distortion values for the static (black) and moving (grey) conditions. Darker greys are due to combinations found in both static and moving conditions (e.g., across the border between the two point clouds).

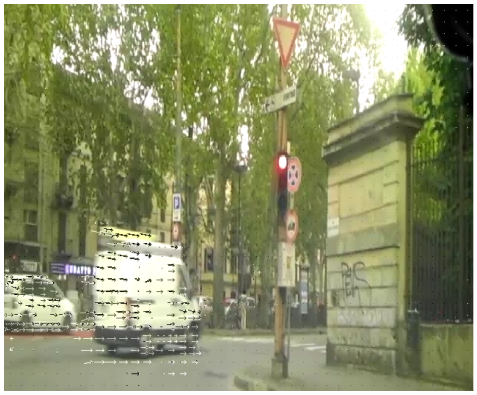

Fig. 2. Crossing vehicles at a traffic light stop. Several high-magnitude motion vectors are present even if the car is not moving.

events. The results of these experiments are very promising in terms of event detection accuracy.

The paper is organized as follows. Section II presents an investigation of the characteristics of some of the features that can be easily extracted as a side information from a video encoder and how to select them. Then, Section III explains how videos have been annotated and the resulting values have been used to train the proposed event detection system. The event detection algorithm is presented in Section IV, followed by simulation results in Section V. Finally conclusions are drawn in Section VI.

II. Feature Analysis and Selection

This section presents a preliminary analysis of the characteristics of very simple features that can be easily extracted as side information from a state-of-the-art video encoder, i.e., H.264/MPEG-4 AVC [5], such as frame and macroblock sizes and encoding distortion (MSE), macroblock types and motion vector components.

A simple example shows how some of those features can be effectively used to detect simple events. Consider, for instance, the average frame size and encoding distortion of each frame. Fig. 1 shows a scatter plot of those values, colored in black if the car is static or in grey if the car is moving. Although it appears intuitive that static scenes should require less bits to encode them, consider that there might be conditions in which such a result is not so obvious.

For instance, if the car is stopped at a traffic light and other cars cross the intersection, some motion, that can also have high intensity, may take place (the darker grey points between the black and grey clouds of points in Fig. 1). A sample of this situation is represented in Fig. 2 where a van is moving across the road while the car is stopped at the traffic light. Therefore, the frame size and distortion alone might not be suitable to reliably detect the movement in all cases. However, the availability of information for each single macroblock could help to better distinguish the moving from the static condition in such a case, e.g., noticing that the amount of macroblocks with higher size and MSE is limited to only a fraction of the total number.

However, a more systematic approach is needed to automatically analyze the relevance of the information that can be easily extracted as a side information from the video encoding process, such as the motion vector components of different macroblocks in the image. The motivation of an intelligent selection of information among the large available amount is twofold: maximizing the performance of the detection algorithm, as well as reducing its complexity.

Many different algorithms have been proposed in the literature concerning pattern analysis and classification for the important problem of feature selection. Several criteria have been used including, for instance, mutual information (MI) [6] between features and classes (which, in our case, correspond to the events that we want to detect).

The mutual information of two random variables is a quantitative indication of the statistical dependence between the two variables, that is the reduction in the uncertainty (as measured by Shannon's entropy) about one random variable yielded by the knowledge of the other one [7].

In principle, given a set of n features $F = (X_1, X_2, \ldots, X_n)$ and a target class C, the ultimate goal is to find the subset $S \subset F$ for a given $m < n$ which bears the highest amount of information I about the class C, i.e., which has the largest dependency on the target class (the so called max-dependency):

$$\arg\max_{S \subset F} I(S = (X_{k_1}, X_{k_2}, \ldots, X_{k_m}); C). \qquad (1)$$

When $m = 1$ the solution is trivial, i.e., the feature whose mutual information with the class is maximal. If more than one feature are involved, i.e., $m > 1$, the mutual information between the feature should be considered because the information of different features may partially overlap. Thus, maximizing Eq. (1) in this case is extremely difficult in practice, especially if m is large. Therefore, often the max-dependency problem in Eq. (1) is approximated with the simpler max-relevance:

$$\arg\max_{S \subset F} \frac{1}{|S|} \sum_{X_i \in S} I(X_i; C), \qquad (2)$$

TABLE I
FRAME-LEVEL FEATURES AVAILABLE AT LOW COST.

Name	Type
Frame size	integer
Encoding distortion (MSE)	float

TABLE II
MACROBLOCK-LEVEL FEATURES AVAILABLE AT LOW COST.

Name	Value type
Size	integer
MSE with respect to original	float
Motion vector components (x,y)	fractional (quarter-pel precision)
Motion vector magnitude	float (computed from x and y)
MB type	enumeration
Number of MVs per MB	integer

TABLE III
EVENTS TO DETECT.

Event	Description	Possible values	Probability of value
Moving	Car is moving	yes / no	90.4% (yes)
Lane change	Car is changing lane	yes / no	2.0% (yes)
Change direction	Direction when lane change is present	left / right	48.2% (left)
Queue	Many cars in front of the camera	yes/no	6.0% (yes)

i.e., taking the subset of the m features that individually maximizes the mutual information exchanged with the class, ignoring the mutual information among the features which, of course, would decrease the joint mutual information with the class. Although the max-relevance criterion may look simple, it proved to be reasonably effective for our aims. To compute mutual information we estimated all the probability mass functions by frequency counts on a training set, i.e., our test video sequences.

We considered features at both the frame and the macroblock level. They are listed in Table I and II for the frame and macroblock level respectively. All of them can be easily extracted during the encoding process. The events of interests considered in this work are listed in Table III which also shows the probability of occurrence of the possible values estimated by means of frequency counts. Note that the direction of the lane change is computed only when a lane change happens.

Fig. 3. Motion vectors in a sample scene when the car moves along the road.

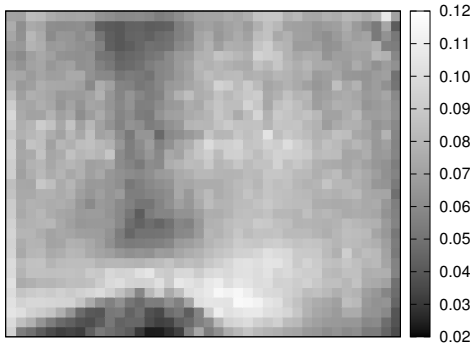

Fig. 4. Mutual information between the magnitude of each motion vector and the moving event.

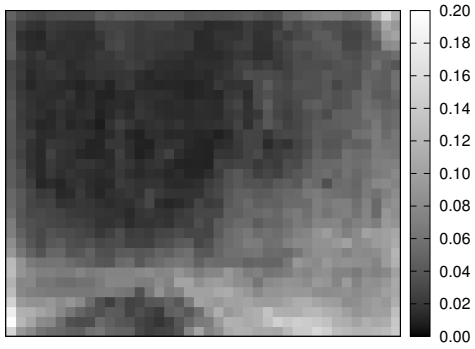

Fig. 5. Mutual information between the size of each macroblock and the moving event.

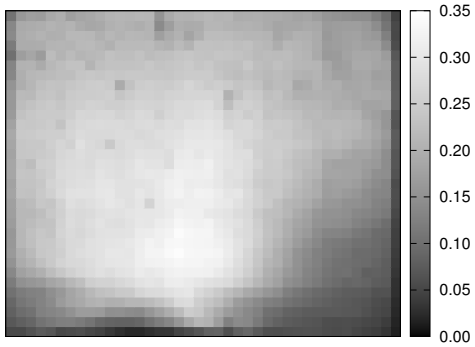

Fig. 6. Mutual information between the direction of each motion vector and the change in the direction of the car.

A mutual information value between features at the macroblock level and the events of interest can be computed, and the result can be graphically shown using different grey levels. These values are shown in Figures 4, 5, 6 for some combinations of features and events. Brighter colors represent higher mutual information values.

Figure 4 shows that the contribution of the magnitude of each macroblock to the detection of the motion event is high and almost equal from the sides, while in the central part the contribution is lower. This is due to perspective projection of the video scene on the camera sensor. When the car is moving forward, the majority of the motion vectors are located at the sides, since they are the part that presents higher apparent

motion due to the perspective. For instance, note the trees at the road size in the sample image shown in Fig. 3. The same image also shows that the contribution of the motion vectors present on the road surface are limited, probably due to the uniformity of the road surface that does not allow to compute motion vector information related to the actual camera motion in the real world.

Figure 5 shows that also macroblock sizes can be an interesting indicator. Large macroblock sizes suggest more difficulty in efficiently encode the residual information. Therefore, car motion is more likely to happen. The more difficult objects to encode are typically the one at the road side, hence the higher MI value in those areas.

In Fig. 6 the direction of the motion vectors is considered to determine the change in the direction of the car, i.e., left or right. As it can be expected, nearly all macroblocks equally contribute to this decision. In fact, when the car is changing lane or it is turning, the image approximately presents an apparent global motion that results into motion vectors spread over the whole picture, all of them with approximately the same direction.

The previous few examples showed that the MI approach can reliably be used to determine the macroblocks in the image that provide the highest contribution in identifying the event of interest. Moreover, MI is extremely valuable since it provides a quantitative measure of the contribution of each feature to the detection of the events of interest. MI provides guidance to perform a smart selection of macroblock used to identify a given event, as well as to reduce the number of features needed to perform the detection. This is important since the number of macroblocks for each frame is in the order of a thousand and the video frame rate can be high, e.g., about 30 frames per second. Thus, the MI value is used to sort the macroblocks a priori so that only a subset of them is used in the detection algorithm. The operation is repeated for each event of interest.

III. Video Annotation

In order to train the detection algorithm a large set of features-class pairs must be used. Therefore, video sequences have been manually inspected and annotated for the presence of the events of interest in the video by marking all the frames in which the event is present. For instance, for the case of the moving event, all the frames in which the car is moving have been marked as "yes" and the remaining ones as "no".

To efficiently perform this operation a software tool, named Anvil [8], has been used. The software, written in Java, has been specifically developed for the purpose of annotating multimedia sequences. Although it has been originally developed for speech and audio annotation, it also supports video and it is extremely flexible in defining the various event types that can be inserted in the annotation. All the configuration is based on XML files and it can also be set up by means of the graphical user interface. Fig. 7 shows a screenshot of the program, in which the bottom part shows a sample annotation of a segment of the video.

The program can export the result of the annotation in text files that can be later processed by means of some scripts that we developed for the purpose of correctly associating events and features of each single frame.

IV. Detection Algorithm

We employed a discriminative model, based on a binary linear Support Vector Machine (SVM) [3], [4]. For each event we try to detect, we map the two possible outcomes on the labels $-1, +1$, Then, a set of instance-label pairs has been generated on the basis of the annotation of the video. The training phase consists in solving the unconstrained optimization problem typical of the linear support vector machines. For this purpose, we employed the library described in [3]. The result of the previous step is a vector of weights to apply to the value of each feature to perform the classification.

Solving the previous optimization problem is computationally heavy especially if a high number of frames are involved, as in our case (about half a million). However, the algorithm need to be run only once. The process has been repeated for different feature sets and different events. Once the vector of weights has been determined, binary classification can be performed with very low complexity, by simply computing the value resulting from the application of the weights to the features and then using a zero threshold to decide between the two labels.

V. Results

The video sequences have been acquired by means of a mobile phone (HTC Evo 3D) with a stereoscopic camera fixed to the windscreen of a car. Several scenarios have been considered: urban, highway and motorway. More than four hours of video have been collected.

The video has been captured at high quality, 1280×720 pixels, 30 frames per second (fps), in stereoscopic mode. Then, the video has been cropped to avoid the uninformative part, i.e., the dashboard of the car, that does not change over time. Finally, the video has been compressed using the standard AVC test model software version JM11 [9] with a fixed quantization parameter equal to 28, using only I and P frames and GOP size equal to 30 frames. To avoid delay and additional complexity, B frames have not been used. The encoding software has been modified to extract the features of interest, e.g., macroblock size and motion vector components, during the encoding process. For the stereoscopic video case, the encoder has been adjusted to encode each right frame as a prediction with respect to the left one. Therefore, motion vectors for the right image constitute the information needed for disparity compensation. Although motion estimation may provide results unrelated to the actual disparity of the real objects present in the video scene, such "disparity" vectors proved to be sufficient to achieve interesting results for event detection. Note that we purposely avoided to use two references for the right frame, i.e., the right past frame and the current left frame. In this way simpler prediction algorithms

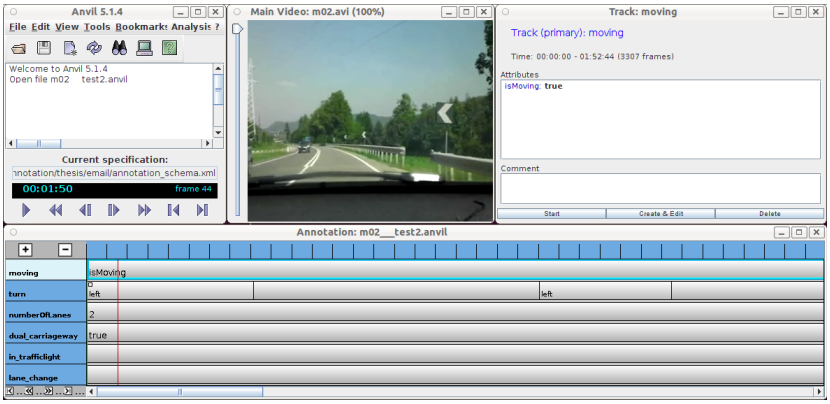

Fig. 7. Sample screenshot of the Anvil annotation tool. Content of the windows, from top left: log, image, event editing and annotation.

mechanisms and hardware can be used, therefore widening the applicability of our system to more devices.

We tested our algorithm by means of a 10-fold stratified cross-validation over a set of 451,980 frames. In other words, 10 cycles of training and testing are performed. Each time, about 90% of the dataset has been used as training data and the remaining part has been used to assess the performance of the generated SVM. This technique is typically employed to test the performance of learning algorithms when there is no clear subdivision between training and test sets. In our experiments each video has its peculiar characteristics, such as the driving environment, thus all the data have to be used. The data is automatically separated, each time, in training and testing by the cross-validation method.

The detection performance has been assessed in terms of the accuracy, precision and recall metrics. The accuracy value indicates how many times the classifier is right on average. More formally,

$$\text{Accuracy} = \frac{(\text{True positives}) + (\text{True negatives})}{(\text{Number of instances})}.$$

While the accuracy shows the average performance of the classifier, i.e., the average identification rate, both precision and recall complement this information. The formal definition of precision is:

$$\text{Precision} = \frac{(\text{True positives})}{(\text{True positives}) + (\text{False positives})},$$

i.e., it measures the fraction of times the event of interest is identified and the event is actually occurring. The recall metric, instead, computes the fraction of occurring events actually identified by the classifier, i.e.,

$$\text{Recall} = \frac{(\text{True positives})}{(\text{True positives}) + (\text{False negatives})}.$$

High recall values imply that, if an event occurs, it is very likely that the classifier will identify it. High precision, instead, indicates that most of the time the classifier is right when it decides that an event is taking place.

In the following, the most significant results are shown for some of the features available at the macroblock level. For each frame, we considered 200 features at the macroblock level, selected according to the MI criteria described at the end of Section II. Table IV shows the performance results in terms of accuracy, precision and recall for some interesting combinations of features and events. Note that, in all cases, in addition to the features specified in the table, the frame size and its total distortion (computed as the MSE with respect to the uncompressed version) have always been included as two additional features in the set. First, the case of monoscopic video has been considered. The monoscopic video is derived from the stereoscopic one by considering the left image only. Results in Table IV show that, for the specific combination of chosen features and events, significantly high values can be achieved in terms of all the three previous metrics, i.e., accuracy, prediction and recall.

We also experimented with the stereoscopic video to understand how much the addition of a second view can contribute to the performance of the detection algorithm. The expectation is that the relation with the first view (i.e., disparity information, although coarse) allows to detect information that cannot be easily extracted from a single view. In this second case, 100 features come from the left frame and 100 from the right frame. As in the previous experiment, the left and right image sizes and their total distortion are also included in the feature set. Therefore, the complexity is similar to the previous case. Results are shown in Table V. Significant improvements can be achieved by means of a second view (the difference in shown in brackets to simplify comparisons) for most of the events. This can be attributed to the fact that the disparity information can effectively provide additional indications, e.g., about the different depth of the objects in the scene. These cues probably make some the events of interest easier to detect. A decrease of precision is observed for the case of the moving event only, which is however compensated by the increase of the recall.

Since it is highly unlikely that events change their status at every frame, we also considered the case in which the detection of the event is attempted on a set of frames rather than a on single frame in isolation. For instance, a car which is changing lane takes a certain time to perform the operation, and it is not necessary to the detect the start and the end of

TABLE IV
PERFORMANCE OF THE DETECTION ALGORITHM FOR THE **MONOSCOPIC** CASE, ONE FRAME AT A TIME. VALUES EXPRESSED AS A PERCENTAGE.

Event	Features	Accuracy	Precision	Recall
Moving	mvx	95.8	91.3	72.0
Lane change	mvx	96.1	98.0	99.9
Change direction	mvx	98.9	99.9	99.1
Queue	#mv per MB	90.3	95.8	94.0

TABLE V
PERFORMANCE (PERCENTAGE) OF THE DETECTION ALGORITHM FOR THE **STEREOSCOPIC** CASE, **ONE** FRAME AT A TIME. IN BRACKETS: DIFFERENCE WITH RESPECT TO THE MONOSCOPIC CASE.

Event	Accuracy	Precision	Recall
Moving	96.8 (+1.0)	75.8 (-15.5)	89.4 (+17.5)
Lane change	98.0 (+1.9)	100 (+2.0)	98.0 (0.0)
Change direction	99.1 (+0.2)	100 (+0.1)	99.1 (0.0)
Queue	94.1 (+3.8)	97.3 (+1.5)	96.5 (+2.5)

TABLE VI
PERFORMANCE (PERCENTAGE) OF THE DETECTION ALGORITHM FOR THE **MONOSCOPIC** CASE, **FIVE** FRAMES AT A TIME. IN BRACKETS: DIFFERENCE WITH RESPECT TO ONE FRAME AT A TIME.

Event	Accuracy	Precision	Recall
Moving	94.6 (-0.8)	96.0 (+4.7)	64.6 (-7.4)
Lane change	98.3 (+2.2)	99.4 (+1.4)	98.0 (0.0)
Change direction	98.9 (+0.0)	99.8 (-0.1)	99.1 (0.0)
Queue	93.2 (+2.9)	99.1 (+3.3)	94.0 (0.0)

TABLE VII
PERFORMANCE (PERCENTAGE) OF THE DETECTION ALGORITHM FOR THE **STEREOSCOPIC** CASE, **FIVE** FRAMES AT A TIME. IN BRACKETS: DIFFERENCE WITH RESPECT TO MONOSCOPIC CASE.

Event	Accuracy	Precision	Recall
Moving	97.8 (+3.2)	82.8 (-13.2)	93.4 (+28.8)
Lane change	87.6 (-9.7)	89.1 (-10.3)	98.1 (+0.1)
Change direction	99.1 (+0.2)	100 (+0.2)	99.1 (0.0)
Queue	93.7 (+0.5)	99.6 (+0.5)	94.1 (+0.1)

the event with frame-level precision. Therefore, we considered sets of five consecutive frames by combining the annotation value with a majority decision as well as averaging the feature values. Thus the same number of features is retained and the complexity of the training and detection phase is not increased. Considering more than one frame at a time creates an intrinsic algorithmic delay in detecting the events, which is however very limited (i.e., 166.6 ms for 5 frames at 30 fps) and it can probably be considered acceptable for many applications. Table VI and VII report the results for both the case of monoscopic and stereoscopic video. They further improve with respect to the previous results that consider frames in isolation. However, some exceptions may exist, for instance in the case of the lane change. Therefore, for these cases alternative approaches could be pursued, for instance the analysis could be done one frame at a time and then some postprocessing on the result of the detection could be performed. For instance, sudden changes in the result of the event detection could be filtered considering past decisions, avoiding isolated changes for single frames. However, the results reported here, without postprocessing, allow to better appreciate the capabilities of the proposed system.

VI. CONCLUSION

This work presented a driving event detection algorithm based on the side information available from video compression algorithms run on both monoscopic and stereoscopic videos of the road as captured through the windscreen of a car. First, a set of interesting and easy-to-extract features has been identified in the side information. Then, by means of the mutual information values, the set of features has been reduced by selecting the most suitable ones for the specific events of interest. An SVM-based detection algorithm has been designed and trained on a set of video sequences comprising hundreds of thousand of frames where the events have been provided by means of manual annotation performed by a human operator. Results show that the detection algorithm achieves a very good identification rate for several events of interest. Moreover, the use of a 3D video provided by a stereoscopic camera significantly improves the performance of the detection algorithm. Future work will be devoted to investigate the possibility to identify more events related to object movements and to use more features at the same time.

REFERENCES

[1] D. Alonso, L. Salgado, and M. Nieto, "Robust vehicle detection through multidimensional classification for on board video based systems," in *Intl. Conf. on Image Processing (ICIP)*, vol. 4, Sep. 2007, pp. 321–324.

[2] E. Carotti and E. Masala, "Low-complexity driving event detection by analysis of video encoding side-information," in *Proc. Intl. Digital Signal Processing Workshop for In-Vehicle Systems*, Kiel, Germany, Sep. 2011.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[4] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S.S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. of the 25th international conference on Machine learning (ICML)*. Helsinki, Finland: ACM, 2008, pp. 408–415.

[5] *Advanced video coding for generic audiovisual services (AVC)*, ITU-T & ISO/IEC Std. H.264 & 14 496-10, May 2005.

[6] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[7] T.M. Cover and J.A. Thomas, *Entropy, Relative Entropy and Mutual Information*. John Wiley & Sons, Inc., 2001, pp. 12–49. [Online]. Available: http://dx.doi.org/10.1002/0471200611.ch2

[8] M. Kipp, "Anvil - a generic annotation tool for multimodal dialogue," in *Proc. of 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, Aalborg, Denmark, Sep. 2001, pp. 1367–1370.

[9] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG. (2007, Jan.) Joint Model Number 11.0 (JM-11.0). [Online]. Available: http://iphome.hhi.de/suehring/tml/download