Image-based classification of driving scenes by Hierarchical Principal Component Classification (HPCC)

Robert Kastner, Frank Schneider, Thomas Michalke, Jannik Fritsch, Christian Goerick

2009

Image-based classification of driving scenes by Hierarchical Principal Component Classification (HPCC)

Robert Kastner[⋆], Frank Schneider[⋆], Thomas Michalke[⋆], Jannik Fritsch[⋄], Christian Goerick[⋄]

[⋆]Darmstadt University of Technology
Institute for Automatic Control
D-64283 Darmstadt, Germany
{robert.kastner,
thomas.michalke}
@rtr.tu-darmstadt.de

[⋄]Honda Research Institute Europe GmbH
D-63073 Offenbach, Germany
{jannik.fritsch,
christian.goerick}
@honda-ri.de

***Abstract*— State-of-the-art advanced driver assistance systems (ADAS) typically focus on single tasks and therefore, have functionalities with clearly defined application areas. Although said ADAS functions (e.g. lane departure warning) show good performance, they lack general usability, as e.g. different modes of operation for highways and country roads. This paper presents a real-time capable approach, which classifies the driving scene by using the newly developed Hierarchical Principal Component Classification (HPCC). Based on that, an ADAS gets information about the current scene context and is able to activate different operation modes. Exemplarily, the algorithm was trained on three different categories (highways, country roads, and inner city), but can be applied to any number and type of categories. Evaluation results on 9000 images show the reliability of the approach and mark it as a crucial step towards more sophisticated high level applications.**

Keywords: driver assistance, scene classification, scene context

I. INTRODUCTION

The growing importance of driver assistance systems for further decreasing the number of traffic accidents is a widely acknowledged fact. Along with that, the complexity of tasks these Advanced Driver Assistance Systems have to handle grows likewise, leading to complex systems that use information fusion of many sensory devices and processing results. Nevertheless, currently available systems focus only on restricted application areas, like e.g. highways. Information about the current scene context is not included. However, said applications make a number of assumptions based on the designated context, which results in a predefined set of rules and parameters. Normally, these rules and parameters restrict the usage of these systems to one application area only. Nevertheless, different sets of rules/parameters could allow the application to work in diverse scene contexts by the simple modification of the basic assumptions. Therefore, knowledge about the scene is valuable for improving higher level applications, allowing more dedicated and also diverse reactions in different surroundings.

In this paper, we present a fast and robust approach for the classification of the scene. The algorithm is independent of potentially outdated map data as well as errors and inaccuracies of Global Positioning System data. Furthermore, it is independent of precoded context information in the map data, because it requires only an image of the scene. Based on the novel Hierarchical Principal Component Classification (HPCC), the algorithm reliably classifies the scene (over 97%) in various, complex scenarios (evaluated on 9000 images). As the evaluation will show, the presented approach is an important step towards further understanding of the driving situation, as well as the basis for applications with a diverse range of operation.

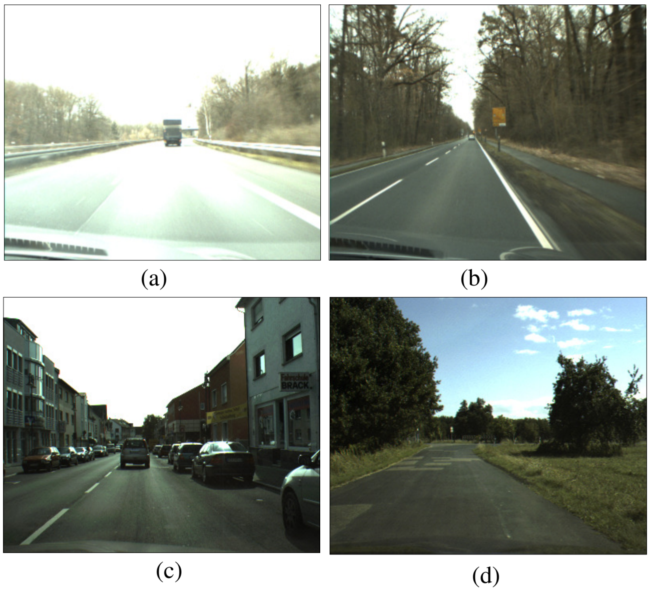


Fig. 1. Examples for the different scene categories and illumination conditions: (a) Highway, (b) Country road, (c) Inner city, (d) Country road.

II. RELATED WORK

The concept of visual scene classification has gained increasing interrest in recent years. To this end, numerous publications handle this topic in a general fashion, but only a few are designed for the traffic domain. The general procedure of existing algorithms is similar. Therefore, a number of features from the image are extracted, which will either be processed at once or gradually extracted from the image depending on previous results. Afterwards, a classification is done. A possible classification approach compares extracted features with mean values derived during a training phase. Publications of visual scene classification, mostly use a number of selected categories, like e.g. indoor/outdoor, landscape/city, and coast/landscape/forest/mountains. Typical low level features for the classification are color histograms (see [1], [2], [3]), texture orientation (see [3], [4], [5]) or a combination of these features.

Another approach is the usage of special features on the intermediate level, like grayscales, color spots, and SIFT descriptors as done by [6], using probabilistic Latent Semantic Analysis (pLSA) for the classification. A further intermediate level approach was proposed by [7], which segmented the image by its RGB, HSV and texture values, by assigning each pixel to one of six categories like water, sand, sky, etc. Afterwards, the pixels of the same category are transformed to regions, segmenting the overall picture. The regions and their spatial relation to each other are used for the classification. Oliva and Torralba (see [8], [9]) have shown a more promising approach for the classification of the scene based on the usage of the frequency domain. Thus, they introduced so called Discriminant Spectral Templates (DST), which are generated from a large number of sample images. Therefore, the spectrum of an image is sampled by a number of frequency selective filters and assigned to each of the classes (like artificial/natural, open/closed, expanded/enclosed, etc., see [10]) with a certain value between 0 and 1 corresponding to their respective membership. Hence, each class has a continuous scale describing the membership of the image to the class. The sampled frequency results for each of the features allows the generation of templates, which describe all relevant frequencies with their corresponding intensity for a certain class. However, the proposed method by Oliva and Torralba is on the one hand continuous regarding the obtained classification results, which makes a crisp decision for a certain class difficult. On the other hand the approach is not discriminative enough in case of similar categories as present in the car domain.

Publications that handle scene classification applicable for driver assistance systems are sparse. One of these is the work by [11], which uses features of the HSV-color space to identify the number of image pixels having the color of bricks, grass, etc. These features are evaluated by a set of fuzzy rules to classify the scene. Nevertheless, most of the categories (in the general case) differ quite strongly regarding their visual features (e.g. landscape/city), in contrast to typical scenes in the traffic domain. A typical scene for an ADAS will always contain street in front of the car. The upper middle part often shows sky. Objects can always occlude the view on the scene and there are no unique objects for different categories, which would simplify the task. Additionally, the classification has to deal with changes in lighting conditions and also great variety within a scene category, e.g. a country road through a forest compared to a country road surrounded by grassland (see Fig. 1 for exemplary categories).

III. SYSTEM DESCRIPTION - SCENE CLASSIFICATION

In the following, a rough overview of our approach for scene classification is given (see Fig. 2). Thereafter, all processing steps and their theoretical background are described in more detail.

The overall system can be divided in three main parts. The first one is the preprocessing, where adaptations of the image take place to reduce the amount of data, as well as to reduce the influence of changes in lighting conditions (see [8]). In the last step of the preprocessing, the image is divided in 16 equally sized square sub parts, which will be processed independently in the following feature extraction part. In the second part a feature extraction is done for obtaining the relevant information of the image. As already mentioned, our system uses the frequency domain to get a compact representation of the data. To this end, for each of the 16 subparts the DFT is used to compute a spectrum. Afterwards, each spectrum is sampled (i.e. weighted) with a number of Gaussian filters, which is inspired by [12], that used oriented Gabor filters for the sampling of the spectrum. The sampled data is normalized with the computed mean and variance values from the training phase, to provide comparability between the different scales of the sub parts. To further reduce the amount of data, while keeping the relevant information, the principal component analysis (PCA) is applied. Finally, in the third part the classification takes place, which is independent of the previous steps, because different methods could be chosen. Therefore, two versions of the newly developed HPCC have been evaluated.

A. Preprocessing

The input of the system is a 400x300 RGB image, which in the first step will be converted to a grayscale image. Afterwards, the influence of different lighting conditions is reduced. To this end, the image is rescaled using a logarithmic curve to change the distribution of the intensity values. This is motivated by early processes in the human vision pathway and boosts the contrast in dark regions in order to compensate the effects of the limited capabilities of exposure control in digital cameras. More specifically, digital cameras use an intensity average over the complete image to compute the exposure, which leads to low contrast in dark regions. Therefore, the intensity values are rescaled to cover the complete dynamic range (see Fig. 3b). The next step is a high pass filtering to reduce low spatial frequencies, attenuating large differences in contrast as well as side effects of the DFT. For that reason, the image is filtered with a filter whose frequency response is given in Eq. (1) (see [13]).

$$H(k,l) = \begin{cases} 1 & \text{for } k = 0 \cap l = 0 \\ 1 - \left(1 - 0.9e^{-\frac{k^2+l^2}{156.25}}\right) & \text{otherwise} \end{cases} \tag{1}$$

The filtering leads to a uniformly distributed intensity over the complete image (see Fig. 3c). Afterwards, the image is resized to 256x256 pixel, which leads to a compression of

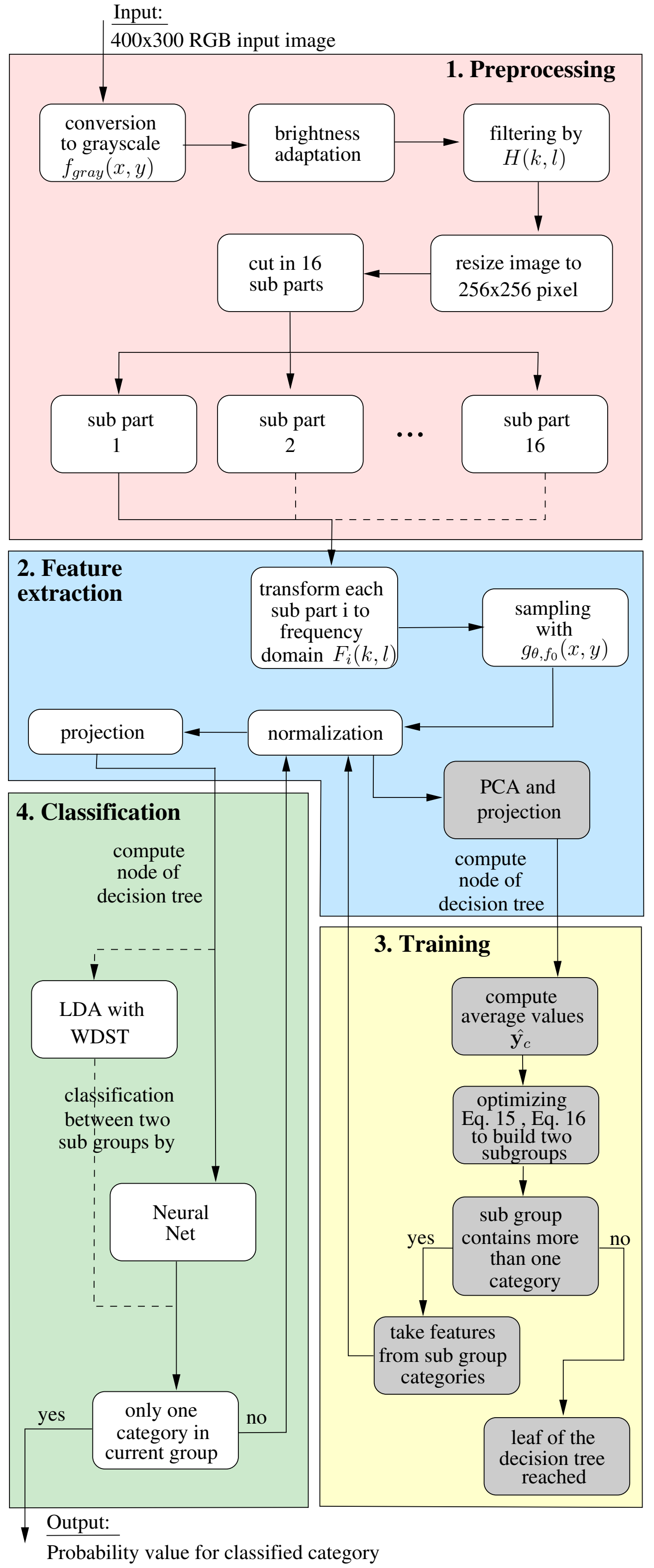

Input:
400x300 RGB input image
1. Preprocessing
conversion to grayscale $f_{gray}(x,y)$
brightness adaptation
filtering by $H(k,l)$
resize image to 256x256 pixel
cut in 16 sub parts
sub part 1
sub part 2
...
sub part 16
2. Feature extraction
transform each sub part i to frequency domain $F_i(k,l)$
sampling with $g_{\theta,f_0}(x,y)$
normalization
projection
PCA and projection
compute node of decision tree
3. Training
compute average values $\hat{\mathbf{y}}_c$
optimizing Eq. 15 , Eq. 16 to build two subgroups
sub group contains more than one category
yes
no
take features from sub group categories
leaf of the decision tree reached
4. Classification
compute node of decision tree
LDA with WDST
classification between two sub groups by
Neural Net
only one category in current group
yes
no
Output:
Probability value for classified category

Fig. 2. System structure for scene classification (gray system modules are only used in the training phase)

the horizontal axis and an elongation of the vertical axis in the frequency domain. Since the resize is carried out during the training as well as execution phase the classification result is not effected. The final step is a division of the image in 16 equally sized square sub parts of the size 64x64 pixel (see Fig. 3d). In the following, each of the sub parts is independently transformed to the frequency domain. The division in sub parts is done to draw conclusions from characteristic amplitude values at certain image regions, which would not be possible if the overall image would be directly transformed to the frequency domain. Hence, each of the sub parts can be described as a complex spatial frequency function separated in magnitude and phase (see Eq. (2)).

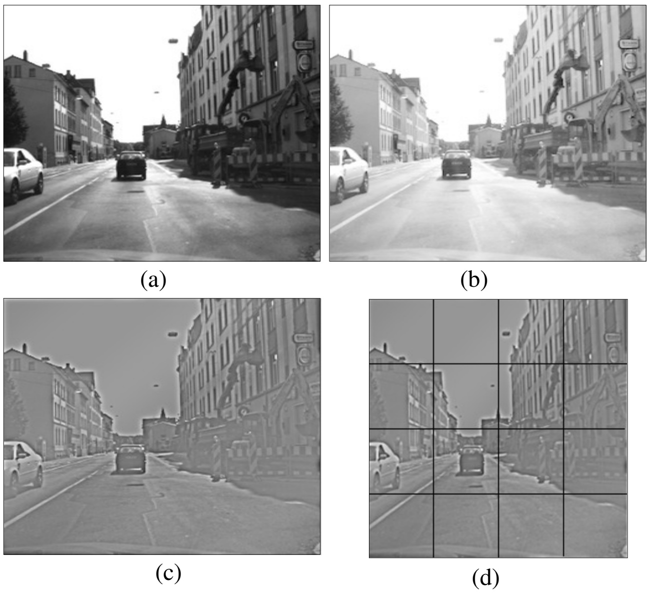


Fig. 3. Different steps of the preprocessing stage: (a) Grayscale Image, (b) Image after adaptation of intensity, (c) Image after suppression of low frequencies, (d) Rescaled image with subparts

$$F_i(k,l) = |F_i(k,l)|e^{\phi(k,l)} \tag{2}$$

In the following only the magnitude of the spectrum will be used and the phase information will be neglected.

B. Feature Extraction

The second part of the algorithm handles the extraction of relevant information from each of the 16 spectra. Each spectrum is handled independently, while the procedure for all the spectra is the same. When aiming at a real-time implementation a spectrum with 64x64 values is a too high dimensional description of the image and unsuitable for a direct classification. Additionally, the knowledge of single magnitude values at a certain frequency and phase is not of interest, because changes of object positions in the image will also cause small changes at the energy level of the spectrum. Therefore, the energy distribution of certain areas of the spectrum will be evaluated, instead of single magnitude values. Another point is the general applicability of the approach, if single magnitude values at a certain position and phase would be used, a possible overfit to the training data might occur, which interferes with the goal of reaching a good generalisation of the classification. Therefore, the spectrum is sampled by a number of Gaussian filters, which are scaled, rotated and shifted to get different resolutions for different frequencies and can be interpreted as a weighted mean of the sampled areas. Each of the 16 spectra is filtered with an array of 100 Gauss filters (see Fig. 4b), which have been adapted to the spectrum providing a high resolution at low frequencies and a low resolution at high frequencies. A single filter kernel (see Fig. 4a) can be described by the Gauss function (see [14] for details) shown in Eq. (3).

$$g_{\theta,f_0}(x,y) = Ae^{-(a(x-x_0)^2+2b(x-x_0)(y-y_0)+c(y-y_0)^2)} \quad (3)$$

For the rotation of the filter kernel by the angle θ the following parameters a, b and c have to be adapted (see Eq. (4), Eq. (5) and Eq. (6)).

$$a = \frac{\cos^2\theta}{2\sigma_x^2} + \frac{\sin^2\theta}{2\sigma_y^2} \quad (4)$$

$$b = -\frac{\sin 2\theta}{4\sigma_x^2} + \frac{\sin 2\theta}{4\sigma_y^2} \quad (5)$$

$$c = \frac{\sin^2\theta}{2\sigma_x^2} + \frac{\cos^2\theta}{2\sigma_y^2} \quad (6)$$

Finally, the width and height of a filter kernel can be adapted by σ_x and σ_y and where chosen in a way that the -3dB border frequencies of the filters touch each other. The shifting of the filter kernel's center is carried out by the adaptation of x_0 and y_0, defined by the spatial frequency f_0, as well as the angle θ (see Eq. (7) and Eq. (8)).

$$x_0 = f_0\frac{N}{2}\cos\theta \quad (7)$$

$$y_0 = f_0\frac{M}{2}\sin\theta \quad (8)$$

For each of the sub parts i the result of the sampling is a vector $\tilde{y}_i$ $(i = 1\ldots16)$ with 100x1 dimensions. The overall result are 16 vectors having 100 values each, making in total a number of 1600 attributes per image.

Before a further reduction of the data can be done, each sub part i should be normalized to generate a similar range of values for each vector element h of $\tilde{y}_i(h)$. For the normalization, the mean and standard deviation vectors have to be estimated for each of the sub parts from the training data. The database for the training holds N images, which results in 16 matrices $\tilde{\mathbf{Y}}_i$ for each of the sub parts (i) with an overall dimension of $100 \times N$. A row j of matrix $\tilde{\mathbf{Y}}_i$ for sub part i contains the data of $\tilde{y}_i$ sampled from image j. Hence, the mean estimator (see [15] for details on this concept) for each vector element h from the sampling vector $\tilde{y}_i$ can be described by Eq. (9).

$$\bar{y}_i(h) = E(\mu_i(h)) = \frac{1}{N}\sum_{n=1}^{N} y_i^n(h) \quad (9)$$

And the standard deviation (see [15]) with Eq. (10) were n denotes the image.

$$s_i(h) = \sqrt{E(\sigma_i^2(h))} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N} y_i^n(h) - \bar{y}_i(h)} \quad (10)$$

The normalization of a sub part i and element h of the feature vector $\tilde{y}_i(h)$ is carried out with Eq. (11).

$$y_i(h) = \frac{\tilde{y}_i(h) - \bar{y}_i(h)}{s_i(h)} \quad (11)$$

For the normalization of the overall training data (all elements (h) of subpart i) and therefore, $\tilde{\mathbf{Y}}_i$ see Eq. (12).

$$\mathbf{Y}_i = \frac{\tilde{\mathbf{Y}}_i - \mathbf{I}^{N\times1}\cdot\bar{\mathbf{y}}_i^T}{\mathbf{I}^{N\times1}\cdot\mathbf{s}_i^T} \quad (12)$$

Where $\mathbf{I}^{N\times P}$ is the unit matrix of dimension $N \times P$, $\bar{\mathbf{y}}_i$ the mean vector and $\mathbf{s}_i$ the variance vector of sub part i.

As already mentioned, a further reduction of the extracted features is carried out based on the results of a PCA (during the training phase). In the following, the PCA (see [16]) will be computed for each matrix $\mathbf{Y}_i$ (containing data of N training images of subpart i), which can be seen as a coordinate transformation, where the first new coordinate axis aligns along the maximum variance of the data, the second axis along the second largest variance and so on. To this end, the result of the PCA is the transformation matrix $\mathbf{\Gamma}_i$, where each column contains the coefficients for a single principal component and the columns are in descending order of their importance. To reduce the size of each $y_i(h)$ only the first v principal components will be used for the projection. Therefore, a shortened transformation matrix $\hat{\mathbf{\Gamma}}_i$, containing only the first v principal components, is constructed.

The final step of the feature extraction is the projection of the matrices $\mathbf{Y}_i$ to their new coordinate system by Eq. (13).

$$\hat{\mathbf{Y}}_i = \mathbf{Y}_i \cdot \hat{\mathbf{\Gamma}}_i \quad (13)$$

Similarly, the transformation is carried out for a single feature vector $\mathbf{y}_i$ by Eq. (14), which is the input for the following classification step.

$$\hat{\mathbf{y}}_i = \mathbf{y}_i \cdot \hat{\mathbf{\Gamma}}_i. \quad (14)$$

C. Training and Classification

Due to limitations in space only the HPCC will be explained in detail, while the underlying methods are not described here, but can be found in the literature. Details on classification approaches as the Linear Discriminant Analysis (LDA) with Windowed Discriminant Spectral Template (WDST) can be found in [10] and for details on neural networks please refer to [17].

In most cases a classification algorithm tries to distinguish between all possible categories in one step, by simply extracting the principal components from all categories. A different approach was developed for the Hierarchical Principal Component Classification (HPCC) of the here described system. More specifically, a decision tree is build, always

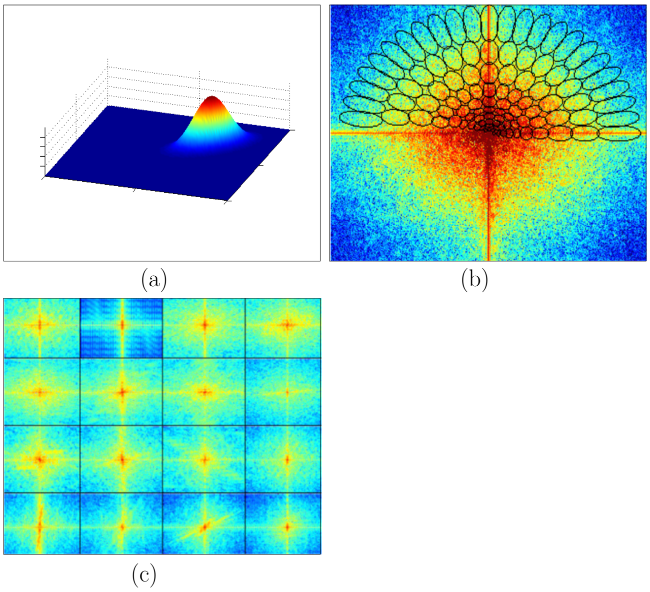


Fig. 4. (a) Two dimensional Gauss function, (b) Spectrum and filter array with -3dB level lines , (c) Image after transformation to frequency domain

separating between two groups and each group contains a number of categories (at least one). At the first view, this is not a novelty if the classification is based on the same data or the same principal components. But this changes if during each classification step only the principal components from the current two groups are used. At first, the average values $\hat{\mathbf{y}}_c$ of the feature vectors in principal component space for each category are computed. On the basis of these average values the categories will be divided in two groups $A, B \subseteq C$ (with A, B satisfying $A \cap B = \emptyset$ and $A \cup B = C$) optimising the criterion of Eq. (15).

$$\min_{A,B \subseteq C} \left(\sum_{i,j \in A, i \neq j} |\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j|_2 + \sum_{i,j \in B, i \neq j} |\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j|_2 \right) \quad (15)$$

Whereas $|\mathbf{x} - \mathbf{y}|_2$ describes the Euclidean Distance between vector $\mathbf{x}$ and $\mathbf{y}$ (see [16]) . At the same time, the Euclidean Distance between the average values of group A and B should become maximal (for a mathematical description, see Eq. (16)).

$$\max_{A,B \subseteq C} \left| \frac{1}{|A|} \sum_{i \in A} \hat{\mathbf{y}}_i - \frac{1}{|B|} \sum_{j \in B} \hat{\mathbf{y}}_j \right|_2 \quad (16)$$

The separation in groups A and B is the initial step at the first node of the decision tree. If for example group A contains more than one category the procedure above is repeated. The normalization is now done with the mean and standard deviation of the categories which are part of group A and combined in matrix $\mathbf{Y}_A$. Again, the principal components are computed and the matrices projected to the principal component sub space of group A with $\hat{\mathbf{Y}}_{\mathbf{cA}} = \hat{\mathbf{\Gamma}}_{\mathbf{A}} \cdot \mathbf{Y}_{\mathbf{c}}$. Also the average values $\hat{\mathbf{y}}_{cA}$ of the categories are calculated and the previous criteria (Eq. (15), Eq. (16)) optimised to sub divide the group A into AA and AB. This method is applied for group B as well as for all sub groups containing more than one category. The total number of categories is initially defined by the training data. The advantage of the method is the specific calculation of the principal components at each node of the decision tree. Only at the first node the overall training data is used to form two groups, which are maximally heterogeneous to each other, but at the same time maximally homogeneous for the members of the group (concerning the average values $\hat{\mathbf{y}}_c$ of the categories). The classification at each node can either be carried out with LDA and WDST (see [10]) or a neural net (see [17]).
Hence, at each node a specific classification task is trained. The overall system with the classification step is depicted in Figure 2. In a nutshell, a training and subsequent classification is carried out in the following way. As first step, a preprocessing on the complete training data is performed, afterwards each image with its sub parts is transformed independently and their spectra sampled. In the second step, the decision tree is learned. Thereby the matrices for the transformation to principal component space as well as mean and standard deviation vectors for each node are stored. The last step is the classification of an image, therefore, the features of the image are extracted and stored in the sampling vector $\tilde{y}_i$. At each node, the sampling vector is normalized with the corresponding mean and standard deviation vectors and afterwards projected to the principal component space. The result is classified by one of the named methods and assigned to one of the sub groups. This procedure is repeated until a leaf of the tree is reached, which results in the classification of the image to the category of the leaf. In another variant of the HPCC a decision tree for each of the sub parts from an image is generated. Thereby, 16 decision trees are build for each of the sub parts and finally a majority voting is carried out for the 16 results. In the following sections, the approach of a single classification tree for the overall image is called HPCC 1 and the approach with 16 single classifications is called HPCC 2.

IV. RESULTS

In this section, we evaluate the performance of our system by training and test with a total of 10800 images, taken from several image streams. The images were manually assigned to one of the categories highway, country road and inner city (see Fig. 1). The images show various scenes, some containing cars, trucks, and pedestrians, others do not contain traffic relevant objects. The scenes also show different lighting conditions, some are dark and others are bright. The training was conducted with 600 images per category. Afterwards, the evaluation was done by five independent runs with 1800 images each (also 600 images per category). The approach is implemented with Matlab and was evaluated on a 1,83 GHz Intel Centrino Duo, having 1 GB Ram and running Windows XP. Only one of the CPU cores was used for the computation. The results of the HPCC on the different test sets showed similar performance on the classification rate (variation of 2%). Table I shows the average results over the five evaluation runs. To draw a comparison, a system

Method	correct classification			total
	Highway	Country road	City	
HPCC 1 with LDA/WDST	98,37	94,03	96,17	96,19
HPCC 1 with Neural Net	98,60	97,77	97,30	97,89
HPCC 2 with LDA/WDST	98,73	84,70	92,80	92,11
HPCC 2 with Neural Net	98,63	96,60	94,93	96,72

TABLE I

AVERAGE RESULTS OF THE CLASSIFICATION ON ALL TEST SETS.

Method	Mean computation time (s)
HPCC 1 with LDA/WDST	3,775
HPCC 1 with Neural Net	2,446
HPCC 2 with LDA/WDST	3,728
HPCC 2 with Neural Net	4,071

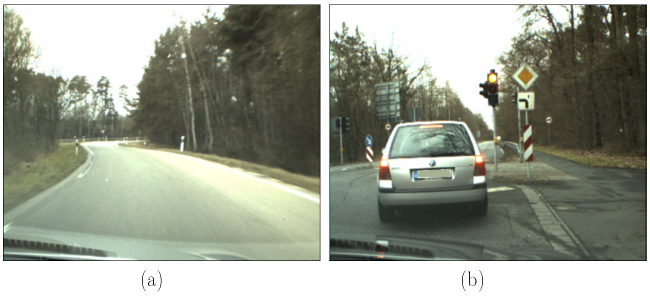
TABLE II

AVERAGE COMPUTATION TIME PER IMAGE.

only comprising of LDA with WDST was set-up. Hence, the LDA/WDST system generates a template for each categorie without subparts. Therefore, it is comparable to the work of Oliva and Torralba [8] and showed a total result of nearly 68% on a single evaluation run. Compared to our results, the HPCC 1 with a neural net reliably classifies the scene with over 97% accuracy and also requires the fewest computation time. To this end, the HPCC is able to provide the current scene context as an input for higher level applications. The computation time of the different versions, for the Matlab implementation, is given in Table II. As the experience shows a factor of 100 can be gained with a native C implementation, so the approach should be capable of 25Hz frame rate. The images being incorrectly classified (see Fig. 5 for two examples), show on the one hand, largely covered areas, due to cars and trucks in front. On the other hand, they show ambiguous scenes, which would also be difficult for a human to classify correctly, without temporal integration. This is shown in the two examples, were Figure 5a shows the driveway to a highway, which is not a typical scene for highway but belongs already to the category. The other Figure 5b shows a stop at a traffic light on a country road, which underlines the smooth transition between the different categories. Nevertheless, so far no temporal integration of the results is carried out, which can increase the classification rate even further and suppress outliers.

V. SUMMARY

This paper describes a generic and fast method for scene classification. Information on the scene context is an important step towards the understanding of complex scenarios for future high level applications. The proposed scene classification approach allows the building of ADAS supporting diverse modes of operation. This facilitates building robust safety-relevant algorithms as trajectory planning and active collision avoidance.


(a)

(b)

Fig. 5. Two example images resulting in a wrong classification: (a) Driveway to a highway as country road, (b) Country road as inner city

In our future work, we plan to incorporate the proposed approach in our biologically motivated driver assistance system (described in [18]) to get it running online and in real-time on our prototype vehicle. Additionally we plan to embed the scene classification into our attention-based system approach for scene analysis (see [19]).

REFERENCES

[1] N. Serrano, A. Savakis, and J. Luo, "A computationally efficient approach to indoor/outdoor scene classification," *Proc. of the Int. Conference on Pattern Recognition*, vol. 04, pp. 146–149, 2002.

[2] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," in *IEEE International Workshop on Content-based Access of Image and Video Databases, in conjunction with ICCV'98*, 1998, pp. 42–51.

[3] A. Vailaya, A. Jain, and H. J. Zhang, "On image classification: City vs. landscape," in *IEEE Workshop on Content - Based Access of Image and Video Libraries*, 1998, pp. 3–9.

[4] M. Gorkani and R. W. Picard, "Texture orientation for sorting photos "at a glance"," *ICPR-A*, vol. 94, pp. 459–464, 1994.

[5] K. Hotta, *Computer Vision Systems*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, vol. 5008/2008, ch. Scene Classification Based on Multi-resolution Orientation Histogram of Gabor Features, pp. 291–301.

[6] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via plsa," *Proceedings of the European Conference on Computer Vision*, 2006.

[7] D. Gökalp and S. Aksoy, "Scene classification using bag-of-regions representations," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, vol. 1, 2007, pp. 1–8.

[8] A. Oliva, A. Torralba, A. Guerin-Dugu, and J. Herault, "Global semantic classification of scenes using power spectrum templates," 1999.

[9] A. Oliva, "Gist of the scene," in *The Encyclopedia of Neurobiology of Attention*, L. Itti, G. Rees, and J. K. Tsotsos, Eds. San Diego, CA: Elsevier, 2005, pp. 251–256.

[10] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[11] M. Wilson, "Interclass fuzzy rule generation for road scene recognition from colour images," *Proc. of the 9th Int. Conference on Computer Analysis of Images and Patterns*, vol. 2124, pp. 692–699, 2001.

[12] A. Oliva and P. Schyns, "Coarse blobs or fine edges? evidence that information diagnosticity changes the perception of complex visual stimuli," *Cognitive Psychology*, vol. 34, pp. 72–107, 1997.

[13] F. Wahl, *Digitale Bildsignalverarbeitung*, 2nd ed. Berlin, Heidelberg: Springer, 1989.

[14] M. Lighthill, *An introduction to fourier analysis and generalized functions*, 1st ed. Cambridge, MA, USA: Cambridge University Press, 1958.

[15] L. Fahrmeir, R. Künstler, I. Pigeot, and G. Tutz, *Statistik - Der Weg zur Datenanalyse*, 6th ed. Heidelberg, Berlin: Springer, 2007.

[16] K. Backhaus, W. Plinke, B. Erichson, and R. Weiber, *Multivariate Analysemethoden*, 11st ed. Berlin, Heidelberg: Springer, 2006.

[17] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River: Prentice Hall, 2002.

[18] J. Fritsch, T. Michalke, A. Gepperth, S. Bone, F. Waibel, M. Kleinehagenbrock, J. Gayko, and C. Goerick, "Towards a human-like vision system for driver assistance," *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 275–282, June 2008.

[19] T. Michalke, R. Kastner, J. Adamy, S. Bone, F. Waibel, M. Kleinehagenbrock, J. Gayko, A. Gepperth, J. Fritsch, and C. Goerick, "An attention-based system approach for scene analysis in driver assistance," *at - Automatisierungstechnik*, vol. 56, no. 11, pp. 575–584, 2008.