VEHICLE CLASSIFICATION ON MULTI-SENSOR SMART CAMERAS USING FEATURE- AND DECISION-FUSION

Andreas Klausner, Allan Tengg

Institute for Technical Informatics
Graz University of Technology
Austria, Graz, 8010
{klausner,tengg}@iti.tugraz.at

Bernhard Rinner

Institute of Networked and Embedded Systems
Klagenfurt University
Austria, Klagenfurt, 9020
bernhard.rinner@uni-klu.ac.at

ABSTRACT

In the proposed project we are working towards multi-sensor smart cameras, i.e., we augment vision-based cameras by additional sensors such as infrared and audio and, thus, transform a single smart camera into an embedded multi-sensor node. Our software framework for embedded online data fusion, called I-SENSE, which supports data fusion on different levels of data abstraction is presented.

Further our fusion model is presented with the focus set on four main parts, namely (i) the acoustic and visual feature extraction, (ii) feature based data fusion and the feature selection algorithm, (iii) feature based decision modeling based on Support Vector Machines (SVM) and (iv) decision modeling based on a modified Dempster-Shafer approach is discussed. Finally we demonstrate the feasibility of our multi-level data fusion approach with experimental results of our "vehicle classification" case study.

Index Terms— sensor data fusion, multi-level fusion, vehicle classification, smart camera, traffic surveillance

1. INTRODUCTION

In the future the world will be faced with a tremendous increase in the number of vehicles on roads. Future traffic monitoring systems will therefore play an important role to improve the throughput and safety of roads and satisfy the trend towards integration of sensor, computing and communication technology into everydays life. Currently, monitoring systems which capture traffic data from a large sensory network, by implementing a centralized architecture are usually vision-based. Furthermore, they require continuous human supervision which is extremely expensive and often unreliable.

In [1] we have demonstrated the potential of combining the scientific research areas multi-sensor data fusion and pervasive embedded computing by presenting an autonomous traffic monitoring system. Visual data is fused with data from other sensors such as audio sensor, radar, infrared or inductive loop sensors and therefore transform a single smart camera to an decentralized multi-sensor smart camera. A novel combination of the following three approaches lead to an embedded and reusable multi-sensor data fusion framework:

First, the full data oriented software fusion model (cp. section 3) enables multi-task operations in a single system capable of traffic monitoring, vehicle identification and classification. Furthermore, this model is a approach to step back from the many application-specific and system-specific data-fusion solutions to a generic model for an effective data fusion-problem solution, and is therefore reusable for many applications with classification and observation tasks.

Second, multi-level data fusion is performed by combining data from different sensors at different levels of abstraction. These fusion levels are differentiated according to the (i) amount of information they provide and (ii) computational complexity. Therefore, dynamically reconfiguration based on actual communication and computation loads is possible.

Third, the developed vision and acoustic feature extraction algorithm as well as fusion methods are implemented on a distributed embedded platform which makes them wider applicable and supports real-time operation.

A sparse least-squares SVM (LS-SVM) [2] is used for feature based decision modeling and is therefore extended for calculating probability mass distributions based on certain and uncertain classification results. In [3] we present a data oriented fusion model and the feature based data fusion method. The shown results show the importance of a feature selection process for selecting the features out of a pool of candidates with the best class separability abilities. The contribution of this article is to introduce a generic fusion model and suitable data fusion methods in order to realize a generic and reusable framework for online data fusion application on an embedded smart cameras on different levels of data abstraction.

The remainder of the paper is organized as follows: Section 2 gives an short review on related work. Section 3 discusses our model for multi-sensor data fusion. Section 4 deals

This project has been partially supported by the Austrian Research Promotion Agency.

1-4244-1354-0/07/$25.00 ©2007 IEEE

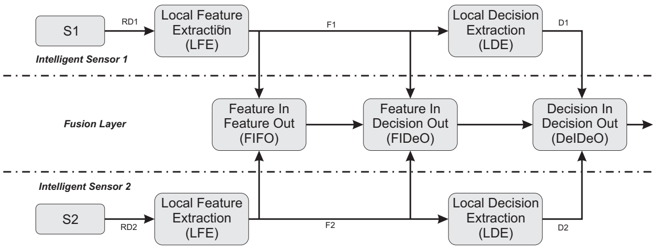
S1
RD1
Local Feature Extraction (LFE)
F1
Local Decision Extraction (LDE)
D1
Intelligent Sensor 1
Fusion Layer
Feature In Feature Out (FIFO)
Feature In Decision Out (FIDeO)
Decision In Decision Out (DeIDeO)
Intelligent Sensor 2
S2
RD2
Local Feature Extraction (LFE)
F2
Local Decision Extraction (LDE)
D2

Fig. 1. Relevant tasks of our Fusion Model characterized by input- and output data

with the visual feature extraction tasks while section 5 treats the acoustic feature extraction tasks. In section 6 we discuss the feature fusion and the feature selection tasks. Section 7 deals with the decision modeling process while section 8 presents some experimental results of our approach, and section 9 concludes the paper with a short discussion of our approach.

2. RELATED WORK

Our idea of developing a high-performance data fusion architecture originates from the SmartCam project [4]. We extend a single-source smart camera to a distributed embedded sensor node (consisting of a network processor and various digital signal processors) capable of fusing data from various heterogeneous sensors. There exists a large variety of multi sensor fusion applications, and therefore most solutions are currently application-specific [5, 6].

Our proposed approach is different because it is a step back from these application-specific and system-specific solutions to a generic and reusable model for online data fusion. Furthermore, most solutions are fusing data only on an individual level and are therefore limited in reusability for other applications. Our approach supports data-fusion based on *raw-data level*, *feature level* and *decision level* depending on the current situation.

Examples for single-level fusion solutions are given in (i) [7, 8] in cases of decision fusion and (ii) [9] in cases of feature fusion, just to mention a few. However, only a few architectures serve multi-level data fusion but they have to much overhead and are, therefore, not suitable for embedded systems.

3. FUSION MODEL

Figure 1 presents the relevant parts of our data oriented generic software fusion model. The complete model and further details are given in [2]. In figure 1 we show an example of two physical sensors, labelled with $S1$ and $S2$ (i.e. audio sensor and visual sensor).

The *Sensing Unit* represents the intelligent sensor which consists of the physical sensor itself and a suitable data preprocessor.

A *Local feature extraction Unit (LFE)* is used to extract a single-source feature vector based on color information, structural information, spectral information or acoustic information of an observed object. These individual feature vectors are input to a feature based data fusion process, namely the *Feature in feature out (FIFO)* process, in order to achieve a joint feature vector estimate based on multiple sensors. Therefore, a *Feature Selector* (cp. section 6) is used to select a suitable set out of a large amount of candidate features.

The *Local decision extraction Unit (LDE)* or the *Feature in decision out unit* are used to extract decisions from objectives features either from a single source or multiple sources based on Support Vector Machines (cp. section 7.1). Therefore, a modified Least-Squares Support Vector Machine (LS-SVM) [2] is used to gather suitable information for a decision fusion process, namely the *Decision in decision out unit (DeIDeO)* unit based on statistical Dempster-Shafer methods (cp. section 7.2).

4. VISUAL FEATURES EXTRACTION

In this section the feature extraction task based on visual features is explained. For our feature extractor we adopted the ideas of Viola and Jones [10] to build a multi-class classificator and improved it using *RealBoost* [11]. Simple *Haar-like* and *center-surround* features are used in the implementation. Additional gradient-based information is used in order to generate robust features and calculate them in real-time on an embedded platform. The boosting approach mainly is used to extract the most powerful features while during feature regression phase the overall interpretation of their values could be left to the feature based decision modeling process (cp. section 7.1).

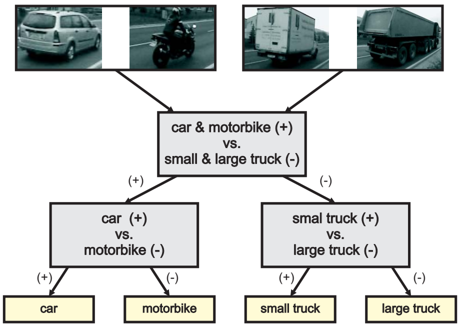

Fig. 2. Visual feature extraction tree. Individual feature extractor are each trained with positive and negative training examples.

To establish a multi-class feature extractor we build a tree structure consisting of several binary extractors (cp. figure 2). We chose the tree scheme to improve computational efficiency.

The result of each stage is given as a set of features with suitable class discrimination abilities. Further information about visual feature extraction can be found in [3].

5. ACOUSTIC FEATURE EXTRACTION

The acoustic feature extractor [12] uses various signal processing algorithms to generate feature candidates. Major issues for the algorithms are robustness against wind noise and non-stationary signal behavior of moving vehicles, and computational feasibility for real time implementation. Acoustic signatures of moving vehicles often behave non-stationary due to variations in engine rotations, gear changes and Doppler effects. Additional wind or environmental noise impact significantly reduces signal to noise ratio. Features from spectral analysis techniques such as harmonic line association [13] assume stationary signals, and thus, often perform poor in our recognition scenario. An optimized feature subset is selected out of the pool of candidate features provided by the algorithms. Audio processing is performed at sampling frequency $f_S = 8kHz$ to afford analysis of broad band signal properties, such as the tire noise. The final optimized subset is build as shown in section 6 and consists of features chosen from the algorithms described in the following sections.

5.1. Cepstral Features

Cepstral analysis techniques estimate the spectral envelope of a signal and are widely spread in speech recognition and musical instrument classification systems. Cepstral coefficients (CCs) are features with good information packing properties and low mutual correlation. They can be estimated efficiently via linear predictive (LP) analysis by conversion of the coefficients into LP based cepstral coefficients. The LP parameters a_i are directly obtained as system of equations from the autocorrelation function $r(k)$, by solving the so called Yule-Walker equations [14]:

$$\sum_{i=1}^{p} a_i r\left(|k-i|\right) = r(k), \tag{1}$$

where p denotes the selected model order. A recursive method for solving these equations is the Levinson-Durbin algorithm. The CCs are then calculated by the following recursion:

$$c_m = \begin{cases} \ln\left(r(0)\right) & \text{for } m = 0 \\ a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right) c_k a_{m-k} & \text{for } m \geq 1 \end{cases} \tag{2}$$

where $a_0 = 1$ and $a_k = 0$ for $k > p$ and p denotes the selected model order. This method avoids any signal transformation and thus, computational effort is highly reduced in comparison to CCs calculated from Fourier transform, since we only require a few CCs to capture the slowly varying properties of the spectrum, i.e. the spectral envelope. The whole signal energy, for example, is contained only in the first coefficient (c_0 term), which yields excellent class discrimination between cars and trucks. The first 60 coefficients are calculated from the same number of autoregressive parameters, which guarantees sufficiently detailed signal description. They are finally used as candidates for the feature selection stage.

5.2. Spectral features

Spectral features [15] include signal attributes that describe average energies, positions and spreads in frequency domain, such as the spectral centroid, signal bandwidth, spectral flux, or band energy ratios. Due to the frame processing of audio signals, spectral features are always given as vectors that reflect signal behavior over time. Thus, statistical moments such as mean, variance and median values are utilized to support spectral features with information about non-stationary signal behavior. Multiresolution analysis was also carried out using discrete wavelet transform with *Daubechier-10 wavelets*. The energy content of each resolution level can be used to localize signal energy and to provide information about spectral composition.

5.3. Time Domain Features

These features describe short time energy behavior of the audio signals, zero crossing rates and features achieved by correlation analysis methods. As single energy values provide only local information, again mean and variance must be calculated as statistical moments to capture long term signal characteristics. The zero crossing rate counts the number of zero crossings of a signal within the specified time interval. It is a measure for the noisiness of a signal, and hence it provides useful information about the presence of tire noise, especially with cars at higher vehicle speed. Cross-correlation analysis was performed with two microphones placed along the road side. Point like sound sources produce interference patterns when passing the sensors, which can be investigated by image processing and thus used for feature extraction.

6. FEATURE FUSION - FEATURE SELECTION

This section deals with the fusion of features from different sensors (cp. section 4 and 5) and the selection of a suitable set out of a pool of candidate features. After feature generation often a very large number of candidate features must be reduced to a sufficient minimum as the SVM classifier can only handle limited number of input features. Some of these candidate features may give reliable class discriminatory information while others do not carry any relevant information and hence, must be excluded as they could mislead the classifier. This task is not trivial since features that provide good classification information only achieve little improve when combined in a feature vector, because of a high mutual correlation. Conversely, the correct combination of features with little class discriminatory abilities may carry out good results.

A genetic algorithm (GA) [16] is used as a search method that follows the principals of natural biological evolution. A set of possible candidate features, is generated at the beginning and used to produce optimized solutions by repeatedly modifying its candidate solutions, the individuals, over successive generations.We provide two feature selection fitness algorithms (i) based on class separability criterions calculated for feature subsets and (ii) based on the classification result of the classifier itself.

6.1. Class separability measures

Different class separability measures have been developed as efficient feature selection criterions in various feature subset searching methods. The major drawback is that they do not always reflect the classifier behavior, and thus yield only suboptimal classification results. Better performance is usually achieved by including the classifier into the selection process and using the classification error rate directly as separability criterion, but this step also includes a high computational cost. Therefore, separability measures are especially important when pre-selecting features out of a large amount of candidate features. Two different measures are implemented in our *I-SENSE* framework:

Bhattacharyya Distance: The Bhattacharyya distance is derived from Bayes decision theory, which assumes multivariate Gaussian distribution for the underlying probability densities:

$$\begin{aligned} B_{ij} = \tfrac{1}{8}(\mu_i - \mu_j)^T \left(\tfrac{\Sigma_i + \Sigma_j}{2}\right)^{-1} (\mu_i - \mu_j) \\ + \tfrac{1}{2} \ln \left(\frac{\left|\frac{\Sigma_i+\Sigma_j}{2}\right|}{\sqrt{|\Sigma_i||\Sigma_j|}} \right) \end{aligned} \quad (3)$$

with Σ denoting the covariance matrix and μ is the mean vector, calculated for the selected feature subset over all objects of the classes i and j, and $|\ \cdots\ |$ as the determinant of the respective term. This criterion consists of two additive terms, one that weights feature vectors with different mean values and the other for different variances. In a multiple class case with M classes, the average Bhattacharyya distance is calculated as follows:

$$Bhat = \sum_{i=1}^{M} \sum_{j \neq i}^{M} B_{ij} P_i P_j \quad (4)$$

where P_i is the appearance probability of class i.

Scatter Matrices: Unlike the above separability measure, the scatter matrices criterion does not assume Gaussian probability distribution for individual features, but investigates how feature vector samples are scattered in the feature vector space. Several scatter matrices are implemeted for that purpose:

- Within-class scatter matrix:

$$S_w = \sum_{i=1}^{M} P_i S_i \quad (5)$$

- Mixture scatter matrix:

$$S_m = E\left\{(x - \mu_0)(x - \mu_0)^T\right\} \quad (6)$$

where $S_i = E\left\{(x - \mu_i)(x - \mu_i)^T\right\}$ denotes the covariance matrix for class i and $\mu_0 = \sum_{i=1}^{M} P_i \mu_i$ denotes the global mean vector. S_m represents the covariance matrix with respect to the global mean. The traces of the matrices can be used to define three different criterions for class separability:

$$J = \frac{|S_m|}{|S_w|} \quad (7)$$

where J-criterion is invariant under linear transformations of the feature space.

Both class separation measures perform well as fitness function in the GA based *Feature Selector*.

6.2. SVM classifier error measures

The second method act as follows: for each feature vector combination, the classification error probability of the classifier is estimated and the one with minimum error selected. That means increased complexity and computational demand for the feature selection process, but on the other hand direct inclusion of the classifier into the optimization process.

As the fitness function is being minimized by the GA, the error rate of the classifier is interpreted as a reciprocal measure for the fitness. That is, for each individual the SVM is trained with part of the database samples and then tested with the other one, yielding the error rate as fitness value. A single train & test run of the SVM does not necessarily lead to a reliable classification result, since feature data can be noised and the result highly depends on the selected training samples. Classifier results can be significantly scattered when using different training data. Hence, the SVM must be trained and tested several times with randomly chosen samples from the database to ensure an accurate average result for the selected features.

feature type	candidate features	error SVM	error CSM
$acoustic$	108	$14.3\%(20/108)$	$18.1\%(20/108)$
$visual$	56	$11.8\%(24/56)$	$13.5\%(21/56)$
$fused$	164	$8.3\%(20/164)$	$11.3\%(21/164)$

Table 1. Feature selection results for individual features and combined candidate features, in brackets the selection ratio is given (e.g 24/56 means 24 features are selected from 56 candidates)

6.3. Feature selection results

Our experiments have shown that the best way to select suitable class separation features from a set of candidate features is to use the classifier itself to obtain the classification error minimized by a GA. The main bottleneck for this method is the limited computational power. Using the method mentioned above as Class Separability Measures (CSM) leads to sufficient results while reducing the necessary time by factor approx. 100.

The table (cp. table 1) show the results from GA based on classifier behaviour compared with error rates of feature subsets derived from optimization runs with scatter matrices criterion J. The presented percentages correspond to the median value of 20 random permutations from the database (i.e. over 2000 cars, 850 large trucks and 520 small trucks) where 50% of each vehicle class are used as learning data and the other 50% are used for evaluation.

7. DECISION MODELING

The decision modeling process is provided as a generic software framework which allows online data fusion on a distributed embedded system with limited memory resources. In our multi-level data fusion framework Support Vector Machines (SVM), proposed by Vapnik [17] as well as a decision fusion approach based on Dempster-Shafer theory of evidence [18] are used as classification method for decision modeling.

7.1. SVM - Feature based decision modeling

For large sets of training data, common SVM learning strategies are not feasible, especially on embedded platforms because of their restricted time and memory resources. Therefore, a modified version of the original SVM, the so called Least Squares Support Vector Machine (LS-SVM) [19] is used for decision modeling in our framework. The main characteristic of LS-SVMs is the lower computational complexity compared with original SVMs, without any quality loss in the classification results.

The extraction of support vectors from a given training dataset is comparable with the problem formulation of finding the most significant vectors in a given data set. The optimal solution for solving this task should combine the following features. It should (i) be fast, (ii) lead to a sparse solution (i.e. low number of support vectors) and (iii) produce good classification results.

In [2] we propose a modified nearest neighbor technique for an intelligent preselection of learning data in order to reduce the training set and therefore reduce the number of support vectors which are then used by the LS-SVM classifier. The remaining datasets are used as support vectors for a LS-SVM classifier to find the decision boundary between two classes in the learning process. Using our approach leads to a sparse LS-SVM classifier with good classification results (about 2% higher error rate compared to standard SVM) and lower computational costs (about 70% faster than Standard SVM) and lower memory costs (about 55% less data for storage compared to LS-SVM).

The learned function by our LS-SVM approach can be given as:

$$f(x) = \sum_i y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{8}$$

where i are the indices of the support vectors out of the training data set, y are the label of the support vectors, α_i are the non-zero Langrange multipliers and b is the bias. It is seen that both training and evaluation of $f(x)$ can be done in terms of arbitrary kernel functions $k(\mathbf{x}, \mathbf{x}_i) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}_i) \rangle$. In cases of classification we modified the interpreting of the given result by equation 8 in a way that we introduce a threshold value to distinguish between certain and uncertain classification. In both cases a probability measure for the classification correctness is calculated – by interpreting the distance from the actual data point to the separating hyperplane. By using a simple one-against-one technique the classifier output is modified in a way that a list of all possible classes and a corresponding probability measure is given as well as the a resulting uncertainty instead of investigating only the probable class. This output style is suitable for our DS based decision fusion approach.

7.2. DS combination - Decision based decision modeling

This section deals with the fusion of decisions from individual sensors. The main objective is to use the complementary information from different single-source classifiers to fuse these classification result into a single decision or more precise into a matrix of uncertainty intervals for each possible proposition the so called “frame of discernment Θ”. Here, we use a distance mass function of our SVM based classifier as our DS belief function.

The DS theory of evidence is a powerful tool representing uncertain knowledge. Therefore this theory have inspired many researchers to investigate different aspects related to uncertainty and lack of knowledge and their application to real world problems. In the following a short overview about the DS theory basics should be given and our method for weighted DS combination is introduced.

The hypotheses set Θ, is indent to represent a set of mutually exclusive prepositions. The evidence on a subset $X \subset \Theta$ is represented with a *basic belief assignment* $m(X) \geq 0$ and satisfies the following:

$$\sum_{X \subseteq \Theta} m(X) = 1, \text{ and } m(\phi) = 0 \tag{9}$$

where ϕ is the empty set. The *belief function* $Bel(X)$ gives the amount of evidence which implies the observation of X. This function is defined on the frame of discernment by the relation:

$$Bel(X) = \sum_{A \subseteq X} m(A) \tag{10}$$

where A is a subset of X. The basic belief assignment $m(.)$ is a generalization of a probability density function whereas a belief function is a generalization of a probability distribution function. The *plausibility function* $Pl(X)$ can be seen as the amount of evidence which does not refute X and is given as:

$$Pl(X) = \sum_{A \cap X \neq \phi} m(A) \tag{11}$$

where the plausibility can be computed as the sum of all basic belief assignments belonging to the subset A that have a non-null intersection with X.

Consider two basic belief assignments $m_1(.)$ and $m_2(.)$ the DS rule of combination combines these pieces of evidence from independent sources to obtain the belief committed to $X \subset \Theta$ according to the following formula:

$$\begin{aligned} m(X) &= m_1(X) \oplus m_2(X) \\ &= \frac{\sum_{A_i \cap B_j = X} m_1(A_i) m_2(B_j)}{1 - \sum_{A_i \cap B_j = X} m_1(A_i) m_2(B_j)}, X \neq \phi \end{aligned} \tag{12}$$

Implementation of equation 12 implies that we trust all sensors equally. This approach can cause problems if the DS fusion system is not properly designed and is therefore suitable only for situations when both sensors(i.e. two sensors are involved into the data fusion process) have the same accuracy estimates or in situations where the basic belief assignments over the frame of discernment Θ can reflect the ignorance going with the observations. Due to building a generalizable sensor fusion framework working with sensors of different accuracy we introduce a weighted combination rule. The basic idea is supposing that we know how a sensor performs historically in similar situations, we can use the historical correctness rates to decide how much we trust in a sensors actual estimation. By using this approach we modify the original DS combination rule to handle cases of sensors with unequal confidence. Equation 12 changes therefore to:

$$\begin{aligned} m(X) &= m_1(X) \oplus m_2(X) \\ &= \frac{\sum_{A_i \cap B_j = X} w_1 m_1(A_i) w_2 m_2(B_j)}{1 - \sum_{A_i \cap B_j = X} w_1 m_1(A_i) w_2 m_2(B_j)}, X \neq \phi \end{aligned} \tag{13}$$

where w are the corresponding estimation correctness rates in history.

8. EXPERIMENTAL RESULTS

In this section we present the results of our vehicle classification case study. We have implemented a simple multi-class classifier by using the One-against-All technique. The database consists of 3 different classes: (i) large trucks, (ii) small trucks and (iii) cars. Further vehicle classes (motorcycles and busses) are available but the number of samples in these classes are rather low in comparison with the other classes and therefore we decide to use only these three classes to demonstrate the feasibility of our multi-level data fusion approach. We generate a database with at least 250 samples in each class (2140 cars, 264 small trucks and 554 large trucks) – 30% are used as training samples and the other 70% are used as evaluation samples. The samples for training and evaluation are chosen randomly from the database.

For all classifications we used our LS-SVM with support vector preselection and radial basis function (RBF) as the kernel function. The results presented in the following figures are average values from 20 runs with random selection of training datasets and random selection of the evaluation set (Note: Both sets are disjunctive).

Firstly we demonstrate the vehicle classification results of our single sensors with the proposed LS-SVM classifier. The box plots shown in figure 3a indicate that class separation only with acoustic features is quite difficult especially in cases of small trucks. Using a confusion (cp. table2) matrix for this case study experiment indicates that the classification system can make the distinction between cars and other vehicles pretty well but has troubles distinguishing between the two types of trucks. Figure 3b indicate that class separation only based on visual features work overall pretty well. Using a confusion matrix (cp. table3) for this case indicates that the classification system can make the distinction between cars and other vehicles pretty well but has troubles distinguishing between the two types of trucks – but still less troubles than classification based on acoustic data only.

Figures 3c and 3d indicate that fusing data from various sensors help to improve the robustness and confidence as well as to reduce ambiguity and uncertainty of the processed vehicle classification. In figure 3c and table 4 we show that our approach for fusing data at decision level is advantageous in comparison to single sensor classification in the overall classification result as well as the individual class separation abilities (e.g. the classification of small trucks can be increased by at least 7% and the classification of large trucks increases

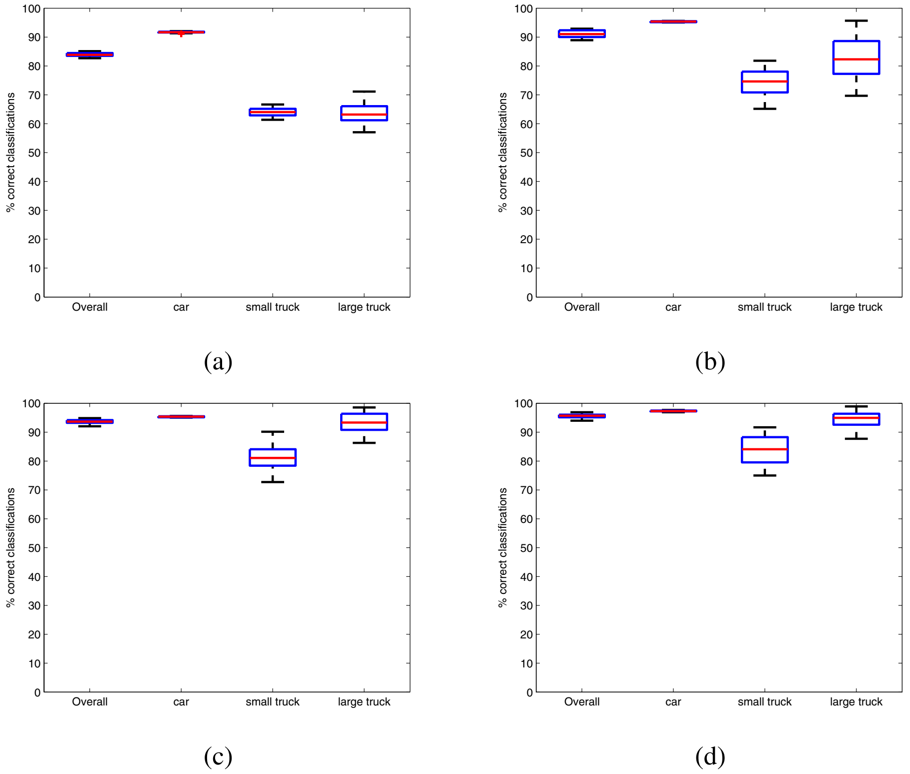

Fig. 3. Classification result with LS-SVM based on (a) acoustic features only, (b) visual features only, (c) DS fused decisions from individual sensors and (d) fused features from both sensors and accurate feature selection; Note: the given boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data.

	car	**s. truck**	**l. truck**
car	1377	82	39
s. truck	13	117	55
l. truck	25	99	264

Table 2. Confusion matrix of the acoustic feature based classification result

	car	**s. truck**	**l. truck**
car	1426	48	24
s. truck	25	140	20
l. truck	3	26	359

Table 3. Confusion matrix of the visual feature based classification result

by at least 10% in comparison to single sensor classification). In our case study we use a weight $w_1 = w_2$ valued by 0.5 in order to trust in both sensors equally. In figure 3d and table 5 we show that our approach for fusing data at feature level is advantageous in comparison to single sensor classification as well as in decisions gained by fusing single sensor decisions by the proposed DS approach, discussed in section 7.2. Furthermore, fusing data on feature level allows to decrease the number of learning samples in order to gain same classification results than with single source data [2]. Note, that classification based on single sensor decisions needs less memory and communication requirements than using feature based classification and therefore both approaches are suitable in an multi-level sensor fusion framework – depending on the current situation and the available computational and memory resources.

9. DISCUSSION

Vehicle classification is one of the most important tasks in traffic surveillance systems. Our aim is to develop a generic software fusion framework which allows online data fusion of different types of sensors on different levels of abstraction.

	car	s. truck	l. truck
car	1431	42	25
s. truck	11	153	21
l. truck	1	9	378

Table 4. Confusion matrix of the decision fused classification

	car	s. truck	l. truck
car	1447	33	18
s. truck	5	170	10
l. truck	2	2	384

Table 5. Confusion matrix of the feature fused classification

A multi-sensor smart camera serves as key component for this implementation. A training data preselection approach [2] makes learning of large training datasets possible even as these embedded systems with restricted memory and time resources.

We show that our approach for multi-level data fusion is advantageous in comparison to single source vehicle classification (e.g. based on visual sensors only) by demonstrating the feasibility in our case study experiment. The results of our experiment demonstrate that the advantage is twofold. First, the classification error rate decreases by using our modified LS-SVM classifier and data fusion based on feature level with a suitable feature selection algorithm proposed in this paper. Second, the decision based data fusion enabled on a modified LS-SVM output of single source decision extractions and the proposed Dempster-Shafer framework can be used to obtain quite similar classification results while reducing the amount of necessary communication and storage data – which is quite important in embedded systems.

We plan to extend our database by a bus class and motorbike class. Further acoustic feature extraction algorithm will be implemented in real-time as DSP based software tasks.

10. REFERENCES

[1] A. Klausner, B. Rinner, and A. Tengg, "I-SENSE: Intelligent embedded multi-sensor fusion," in *Proceedings of the 4th IEEE International Workshop on Intelligent Solutions in Embedded Systems (WISES)*, Vienna, Austria, June 2006, pp. 105–116.

[2] A. Klausner, A. Tengg, and B. Rinner, "Enhanced Least Squares Support Vector Machines for Decision Modeling in a Multi-Sensor Fusion Framework," in *Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition (AIPR-07)*, Orlando, US, July 2007.

[3] A. Klausner, A. Tengg, C. Leistner, S. Erb, and B. Rinner, "An audio-visual sensor fusion approach for feature based vehicle identification," in *Proceedings of the International Conference on Advanced Video and Signal based Surveillance (AVSS-07)*, London, GB, September 2007.

[4] Michael Bramberger, Andreas Doblander, Arnold Maier, Bernhard Rinner, and Helmut Schwabach, "Distributed Embedded Smart Cameras for Surveillance Applications," *Computer*, vol. 39, no. 2, pp. 68–75, Feb. 2006.

[5] R. Chellappa, G. Qian, and Q. Zheng, "Vehicle detection and tracking using acoustic and video sensors," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004.(ICASSP 04)*, vol. 3, pp. 793–796, 2004.

[6] H. Durrant-Whyte and M. Stevens, "Data fusion in decentralised sensing networks," in *Proceedings of the International Conference on Information Fusion, 2001*, 2001, vol. 3, pp. 19–24.

[7] B. Moshiri, M. R. Asharif, and R. Hosein Nezhad, "Pseudo information measure: A new concept for extension of bayesian fusion in robotic map building," *Information Fusion*, vol. 3, no. 1, pp. 51–68, 2002.

[8] V. Koval, V. Turchenko, V. Kochan, A. Sachenko, and G. Markowsky, "Smart license plate recognition system based on image processing using neural network," in *Proceedings of the Second IEEE International Workshop on Advanced Computing Systems: Technology and Applications, 2003*, 2003, pp. 123–127.

[9] A.H. Gunatilaka and B.A. Baertlein, "Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 577–589, June 2001.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE,vol. 1, pp. 511-518*, vol. 1, no. 3, pp. 511–518, 2001.

[11] Robert E. Shapire and Yoram Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning, vol. 37, no. 3, pp. 297-336*, 1999.

[12] A. Klausner, A. Tengg, S. Erb, and B. Rinner, "Dsp based acoustic vehicle classification for multi-sensor real-time traffic surveillance," in *Proceedings of EUSIPCO 07*, Poznań, Poland, September 2007.

[13] M.E. Hohil, J.R. Heberley, J. Chang, and A. Rotolo, "Vehicle counting and classification algorithms for unattended acoustic sensors," *SPIE - The International Society for Optical Engineering*, vol. 5019, pp. 99–110, 2003.

[14] L. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice hall, Englewood Cliffs, NJ, USA, 1978.

[15] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Elsevier Academic Press, San Diego, US, 2006.

[16] M. Mitchell, *An introduction to genetic algorithms*, MIT Press, Cambridge, MA, 1996.

[17] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, US, 1998.

[18] A. P. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society*, vol. 30, pp. 205–247, 1968.

[19] J.A.K Suykens, P. Van Dooren, B. De Moor, and J. Vandewalle, "Least squares support vector machine classifiers: a large scale algorithm," *European Conference on Circuit Theory and Design (ECCTD'99)*, pp. 839–842, 1999.