

The efficient real- and non-real-time make and model recognition of cars

Remigiusz Baran · Andrzej Glowacz ·
Andrzej Matiolanski

Published online: 20 June 2013

© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract Make and Model recognition of cars (MMR) has become an important element of automatic vision based systems. Nowadays, MMR utility is commonly added to traffic monitoring (e.g. Licence Plate Recognition) or law enforcement surveillance systems. Facing the growing significance of Make and Model Recognition of cars we have designed and implemented two different MMR approaches. According to their disparate assumption data of these implementations one is obligated to estimate different car models in milliseconds (with a bit less emphasis placed on its accuracy) while the other is aimed first of all to reach higher classification accuracy. Both the implemented MMR approaches, called Real-Time and Visual Content Classification, respectively, are described in this paper in detail and with reference to other MMR methods presented in the literature. Analyses of their performance with respect to classification accuracy and, in case of the Real-Time approach, to its response time are also presented, discussed and finally concluded.

Keywords Make and model recognition of cars · SURF · SIFT · SVM · Content descriptors · Classification accuracy

R. Baran (✉)
Kielce University of Technology,
Faculty of Electrical Engineering, Automatics and Computer Science,
al. 1000-lecia P.P. 7, 25-314 Kielce, Poland
e-mail: r.baran@tu.kielce.pl

A. Glowacz · A. Matiolanski
AGH University of Science and Technology,
Department of Telecommunications, al. Mickiewicza 30, 30-059 Krakow, Poland

A. Glowacz
e-mail: aglowacz@agh.edu.pl

A. Matiolanski
e-mail: matiolanski@kt.agh.edu.pl

1 Introduction

Different automatic vision based vehicle detection and classification techniques are widely practiced nowadays. They are widespread used in traffic management and monitoring, automated parking garages, driver assistance and control access systems, as well in public as in commercial applications. Another useful adaptation of these techniques is surveillance for crime preventing and terrorism fighting. This category of vision based automatic vehicle detection and classification applications is important for public safety as well as for national security. The most common and well known application from the category of traffic management and monitoring is the Automatic Number Plate Recognition (ANPR). However, due to growing demand which meets mayor advance of technical capabilities, also other categories of vehicle classification have recently been added. Make and model recognition (MMR) of cars [1] is the main and relatively newly added functionality, among such as color and vehicle type distinguishing (between lorries and passenger cars for instance [22]). Unlike vehicle type distinguishing, MMR is aimed at correct identification of car make and model within a given type.

Although a number of scientific papers dedicated for MMR issue have been published recently, the problem of finding an efficient real-time and robust solution is still a challenge. To face this challenge we have prepared and examined two different MMR approaches. Frameworks of both of them as well as their goals, assumptions and obtained results are presented in this paper. For clarity of presentation we divided this paper into six main sections. Later in this section the MMR state-of-the art as well as our research motivation are presented. In Section 2, the training and test image sets as well as the preprocessing steps to both approaches are described in details. Section 3 presents the first of our methods which is a real-time solution combining Speeded-Up Robust Features (SURF) [13] detector and the Support Vector Machines (SVM) algorithm [20]. The second MMR scheme, which puts emphasis on recognition accuracy (regardless to processing time) using visual descriptors is presented in Section 4. In next two sections results of performed experiments and subsequently conclusions with an insight to possible future improvements to proposed solutions are discussed.

1.1 Literature review

Despite the fact that MMR frameworks have already been applied in selected security systems [2], the amount of relevant scientific literature is relatively small. Analysis of this literature allows to distinguish between two main categories of different approaches to MMR issue, which are generally feature- and appearance-based. As approaches proposed in this paper are feature-based, the appearance-based methods will not be discussed in this paper. Some of them are described in [31, 35, 44, 64]. In general, the feature-based approaches cover traditional classification methods (e.g. discriminant analysis [37], multivariate interpolation [25], Bayesian methods [14], and Support Vector Machines [8, 58]), as well as tools from the computational intelligence area (e.g. neural networks [19, 42, 54], and systems based on various combinations of neural networks, fuzzy sets and genetic algorithms [27, 28, 33], with a special emphasis on interpretability-oriented approaches [29, 30]).

One of the first feature-based approaches to the MMR problem has been presented in [53]. The authors of this paper have used a combination of a number of feature extraction algorithms to form vectors of feature descriptors of frontal views of cars allowing to distinguish between different car models. The classification method they have been applied was based on a simple k-Nearest Neighbor (kNN) algorithm [21]. Some different feature extraction algorithms (e.g. Canny edge detector, square mapped gradients) as well as different classifications methods (Naive Bayes) have also been investigated in [52]. In contradistinction to work of Petrovic and Cootes, selected feature extraction algorithms have not been investigated as their combination but separately. Similar idea, also based on edges extracted using Canny edge detector but with investigation of other classification methods (C4.5 decision tree and a feed-forward neural network), has been presented in [49].

A little more sophisticated approach, related to contours (not to edges) [11, 57] has been introduced in [50]. Negri et al. have proposed contours, or more precisely feature arrays containing contour information, as maps modelling different classes of cars in their MMR scheme. These feature arrays, called oriented-contours points matrices, are obtained as a results of complex process which starts from Sobel filtering and ends with special iterative voting procedure. The goal of this voting procedure is to find such contour points which are invariable for all the training images of a given class. The classification procedure of the proposed method uses three different measures (called votes) and the distance error evaluated between the oriented-contour points of the class model and the test sample. In other work—[19], the same authors, have proposed also a second classification stage in which above scores, employed as vectors, are evaluated with a kNN algorithm. Another contour based approach has been presented by Anthony in [10].

Methods described so far are based on edge detectors (e.g. Canny edge detector) operating in spatial domain. In contradistinction, Kazemi et al. [40] have investigated a three different transform domain feature extractors in their MMR scheme. The classifier they have used is a standard kNN algorithm. They have shown that the best recognition rate can be obtained using Curvelet Transform [17]. In [41] the same authors have proposed to combine the Discrete Curvelet Transform (DCT) [16] with multi-class SVM classifier [59]. They have shown this time that such a connection is more efficient, especially when SVM one-against-one strategy [34] is used. Rahati et al. [55] have repeated above research but with Contourlet Transform [24] instead of DCT. Further extension to this work has been done in [63] where a localized directional feature selection criterion, on the Contourlet Transform domain, has been used.

Another sub-class of feature-based methods is related to Scale Invariant Feature Transform (SIFT) [46]. Effectiveness of SIFT based MMR scheme has been investigated inter alia by Dlagnekov. In [23] he has proposed a simple SIFT matching algorithm in which SIFT descriptors calculated for a given query image are matched directly and separately to the descriptors calculated earlier for each of reference images. Estimation in his solution is achieved according to the distance measure defined in the SIFT scale space. He has also investigated other selected methods, including Eigencars and Shape Context Matching. The comparison he performed has shown that SIFT based methods are very promising. Similar investigations, related also to SIFT based methods, have been described in [36] and [62].

1.2 Research motivation

Our activity in the field of vehicle make and model recognition is originated from the INSIGMA R&D project [3] in which we are currently involved. One of the INSIGMA objectives is to develop a software which will be able to process video sequences, registered by surveillance cameras, in order to detect [26] and extract selected features of cars and, inter alia, to recognize their make and models. There are however different requirements related to this software. First of all it should be able to operate in controlled conditions as well in real-time (for rapid system interactions) as out of time limits (for post event system procedures). In case of the real-time part of this software it is expected to achieve a recommendation (related to the queried sample) in no more than few dozen of milliseconds and with classification accuracy not less than 90 %. On the other hand, the accuracy of the no time-limited part should be significantly higher. The imposed accuracy limits seems to be quite reasonable according to the results obtained by other researchers for real-time constraints and realistic data sets. Petrovic and Cootes for instance have reported the accuracy of 87,3 % [53]. Maximum performance of the oriented-contour based MMR system described in [19] was approximately 93 %. This high accuracy related performance has been achieved however using a fusion of different classifiers. Adaptation of localized contourlet features presented by Zafar et al. in [63] allows reaching the accuracy of even 94 %. It is necessary however to notice that a test dataset used in this work has contained only one or two images per each class. In case of four testing images, performance of Zafars approach has dropped to 85 %. There are however also other works where even higher accuracies, up to 100 %, have been reported. For instance a perfect MMR classifier has been proposed in [40], but the number of estimated classes in this case was limited to five. On the other hand, experiments performed by Dlagnekov from Belongie Research Group [4] have shown that SIFT based approaches can be also efficient. The classification accuracy he has reported in [23] for such a case was about 89 %. In our opinion however, the main disadvantage of his method is a simple matching algorithm which compares descriptors of query and reference images directly in one-to-one manner.

These as well as other requirements (more detailed explanation of all relevant system constraints, including time related ones, will be given in Section 2) have provoked us to develop two different MMR implementations.

As the base for the Real-Time approach we have chosen the SURF feature map extractor. We have chosen this because one feature map in use will grant faster performance than combination of two or more features. The SURF extractor we have selected however taking into consideration the promising results of Dlagnekov. Admittedly, Dlagnekov has performed his research using SIFT algorithm, but as it have been shown in [12, 39] “SURF is indeed superior to SIFT and all its different implementations”, especially in terms of runtime performance. Although SIFT indicates slightly better feature quality than SURF, differences between these two algorithms in this aspect are inessential. However, with regard to other aspects, e.g. robustness and scale invariance, both algorithms are comparable.

On the other hand, we have assumed that combination of few selected feature extractors will assure better recognition rate than in case of one selected extractor. Therefore our second approach (called Visual Content Classification approach) combines results from various visual and local features descriptors in one classifier. It gives us multipurpose method which can be adapted to different recognition sce-

narios. We have decided to use some well known descriptors such as SIFT and SURF in addition to promising feature extraction techniques as MPEG-7 descriptors [56].

2 Training and test image datasets

We have assumed that surveillance cameras providing video data to be processed by our approaches will be positioned over every traffic lane, including highways, countryside roads, streets, parking lots, etc. We have also assumed that resolution of MPEG-4 video sequences recorded by these cameras should not be less than 4CIF. In other words the expected minimal resolution of processed images (video frames) is 704×576 pixels. Taking into account the standard image sensor type (1/3 for instance) and the focal length of applied lens equal to e.g. 60 mm, size of the camera field of view (FOV), from the distance of about 40 m (131 ft), is 2.35×1.76 m (7×5 feet). FOV of the same size can be also obtained from the distance of about 5 m (16 ft), but with the focal length equal to 8 mm. Above relationships are illustrated in Fig. 1.

Such FOV assures that the standard automobile (its front or back side) covers approximately 75 % of the entire frame space. This, according to the predetermined video resolution, means however that size of a the car object to be processed is about 600×480 pixels. Such a car object is quite sufficient as well for our MMR approaches (as it will be indicated in relevant sections) as for other tasks related for instance to automatic license plate [38, 43] or vehicle-logo recognition [3] (as it is illustrated in Fig. 2).

With respect to above assumptions we have prepared two separate datasets where one is for training and the other one for testing purposes. These datasets have been used, in different ways however, for further evaluation of both of our approaches. Datasets contain images that have been collected in various lighting conditions over a period of twelve months. All of these images represent front sides of cars taken “en face” or at a little angle (less than 30 degrees). The training dataset contains a total number of 1360 images and corresponds to 17 different models of cars. In other words each model in our training dataset is represented by 80 images taken outdoor or downloaded from the Internet (fifty-fifty). The full list of car models as well as some examples of training images are depicted in Fig. 3.

Fig. 1 The predetermined field of view conditions

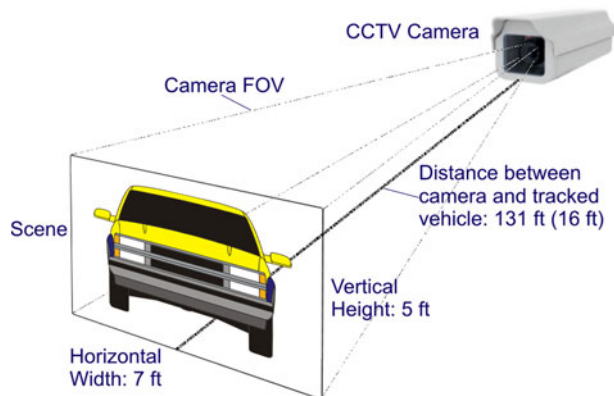




Fig. 2 Proportions of car number plate and logo objects within the 600×480 FOV and their minimal required resolutions—with respect to (a) [9] and (b) [15]

The test dataset is composed of 2499 images. These photos, like the training ones, have been taken as well from outdoor as from the Internet. In case of test dataset however, less attention has been paid to the quality and size of collected images. Therefore, some of these images are soft focus or dark as well as resolution of some of them is less than 600×480 . In addition, number of images corresponding to a given class is different. Their distribution over the full list of analyzed models is presented in Fig. 4.

According to another assumption, minimal frame rate of used surveillance cameras should be 25 fps.

2.1 Preprocessing the datasets

All images from the test dataset are initially processed using our implementation of Haar-like features detector [48, 60]. We use this detector to detect and then to extract



Fig. 3 List of car models covered by the training dataset and their selected examples

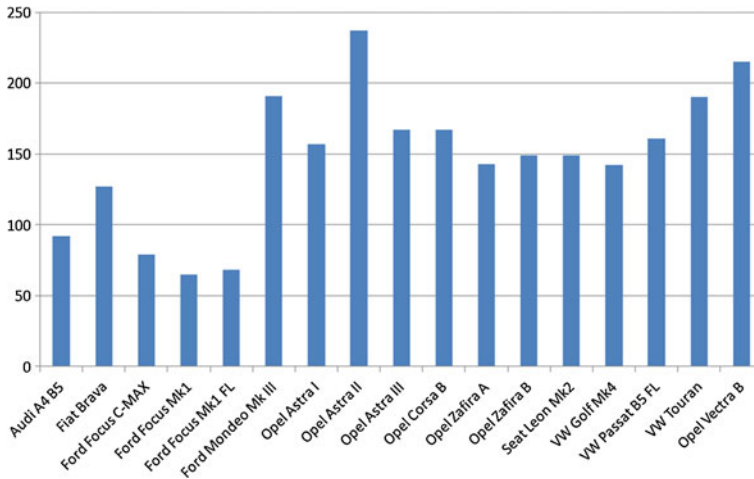


Fig. 4 Number of test images for each class

(by cropping an image) a Region of Interest (ROI) which is a rectangle containing grill part of a car together with its head and indicator lights, like in an example depicted in Fig. 5.

Our Haar-like detector implementation is based on the *haartraining* application from the Open Computer Vision Library (OpenCV) [5]. Numbers of positive and negative samples that we have collected for our classifier cascade training are 3838 and 8831, respectively. Collection of positive samples includes images that belong to the training dataset as well as images that correspond to car models from beyond of our analysis. The performance of our Haar-like features detector is given by Hits, Missed and False indicators shown in Table 1, where ‘Hits’ indicates the number of correct detections (equivalent to the true positive— TP), ‘Missed’ shows the number of missed detections (equivalent to the false negative— FN) and ‘False’ specifies the number of false detections (equivalent to the false positive— FP).

Values presented in Table 1 have been assigned during a test which have been performed on the total number of 1000 positive images. According to the following definition of the hit rate (HR)

$$HR = \frac{TP}{TP + FN}, \quad (1)$$

the hit rate for our ROI detector, with respect to performed test, is 97.2 %.

Fig. 5 A ROI example



Table 1 Hit rate indicators for ROI Haar-like detector

	Hits	Missed	False
Total	972	28	17

ROIs of the same kind are also detected and extracted from all of the training dataset images. These ROIs as reference images (RI) are then used to create a pattern database where its feature vectors are stored.

As a real video stream is a sequence of frames, the MMR approaches presented in our paper, especially the Real-Time approach, are ready to process each frame of the input video sequence provided by the CCD devices. As it has been described at the beginning to this section, all images from the test dataset (as well as from the training dataset), during our experiments, were initially processed by the aforementioned ROI detector. In real-time conditions, this ROI detector is also applied at the beginning of the whole process. Next steps of this process, described in details in next sections, are performed only if the detector returns the Region of Interest.

3 Real-time approach

Our Real-Time (RT) MMR approach, as it was already mentioned, is based on SURF detector. SURF is used at the beginning as well of training as of testing phase of this approach. However, SURF descriptors of selected interest points, representing the local, based on gradient distribution features of an image, stand for primary feature vectors for our approach, they are not vectors directly used in our classification task.

3.1 Training phase

SURF descriptors are computed independently for each of the reference images (ROIs) at the beginning of the training phase. Collection of these descriptors, taken as the whole, is then partitioned, using k-means method [47], into predetermined number of clusters. Set of clusters obtained in this way, called the vocabulary, is then saved as an XML file. Subsequently, SURF descriptors related to a given pattern image (and in consequence to a given model name) are iteratively assigned to the cluster centroids with the closest mean (in sense of Euclidean metric) with respect to the Lloyd's algorithm [45]. Above assignments create a so-called sparse vector of occurrence counts (SVoOC), which is also saved prior to the final step. Diagrams illustrating both phases of the RT approach are depicted in Fig. 6.

Procedure described up till now is a bit similar to the one presented in [6]. However, the final step in our approach is different. At the end of the training phase, histograms are used in general by SVM algorithm to construct a multi-dimensional hyperplane, that optimally separates the input data into predetermined number of classes.

In case of two class SVM, the problem of optimal separating the dataset (of size N) of training vectors x_i ($i = 1, 2, \dots, N$),

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_l, y_l)\}, \quad x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, \quad (2)$$

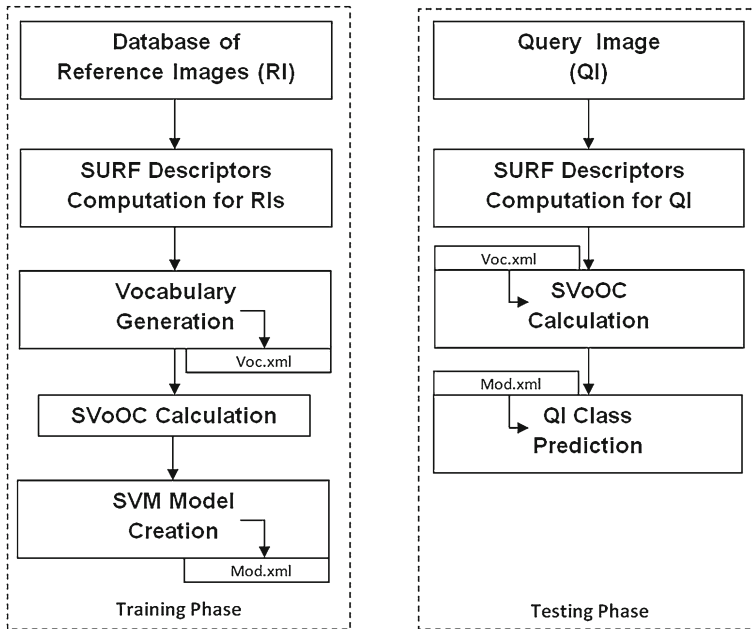


Fig. 6 Workflow of the RT approach

with a hyperplane,

$$\langle w, x \rangle + b = 0, \tag{3}$$

where: $w \in \mathbb{R}^n$ and b is a scalar, comes down to introducing no error while the maximal distance between the closest vector and the hyperplane is determined [32].

When a canonical form of hyperplane is considered [58], w and b are constrained by,

$$\min_i |\langle w, x_i \rangle + b| = 1. \tag{4}$$

Hence, it can be finally formulated that the optimal hyperplane is the one that minimises function $\Phi(w)$, given by the following equation,

$$\Phi(w) = \frac{1}{2} \|w\|^2. \tag{5}$$

In more general form, without restriction to the case where the training data is only linearly separated, optimal separating hyperplane must minimize the function [20],

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i, \tag{6}$$

where: C is a given value and ξ_i are a measure of misclassification errors.

There are many methods of constructing the multidimensional hyperplane that optimally separates the training data into predetermined number of classes. Our implementation of multi-class SVM classifier is based on LIBSVM library [18] where

the “one-against-one” algorithm is used. In such an approach a number of binary classifications is performed according to a voting strategy. These classifications generate a set of support vectors, called the SVM model, that are the closest to the optimal hyperplane. The SVM model is also saved in XML form during the training phase.

3.2 Testing phase

The vocabulary as well as the SVM model are read from relevant XML files in the first steps of the testing phase. Subsequently, a query (test) image (QI) is loaded and its SURF descriptors are computed. SURF descriptors are then assigned to vocabulary’s clusters with respect to the same iterative procedure as in the training phase. As a result, a new SVoOC vector is created. Finally, the SVM classifier estimates a class to which a query image belongs according to its SVoOC vector and the SVM model.

4 Visual content classification approach

The second mentioned method bases on information from visual content descriptors and local features. Our motivation was to create a highly efficient classification method which can be adapted with various classification scenarios. The main requirement for classification was to develop a method with great accuracy irrespective of database kinds. In opposite to the previously mentioned method, algorithm speed was not the main issue. In this article we use Visual Content Classification (VCC) method in MMR scenario. The VCC algorithm consists of two phases: training and testing (Fig. 7).

4.1 Training phase

All tasks used for preparation of reference database for further classification belong to this step. During the training phase calculations are done over the images from reference database (RD). The visual content descriptors belongs to MPEG-7 description standard. In current MMR scenario we use Edge Histogram descriptor [51]. However there is possible to use any other descriptor from MPEG-7 standard which can be useful in different (from MMR) scenarios. As the local features descriptors we use SIFT and SURF. Descriptors for RD are computing once for current database. Calculated feature vectors are used to specify a set of descriptors which provide the best results. Decreasing number of descriptors allows achieving good results with relatively lower computational cost. Numerical representations of descriptors are stored as binary vectors for shorter access time and easier processing.

4.2 Testing phase

The testing phase begins with query image (QI) which we would like to classify to one of reference database category. First step is to calculate descriptors for QI (the same descriptors as for RD). Having feature vectors for RD and QI distances between QI and images from RD are computing. As a result we gain complete list of distances

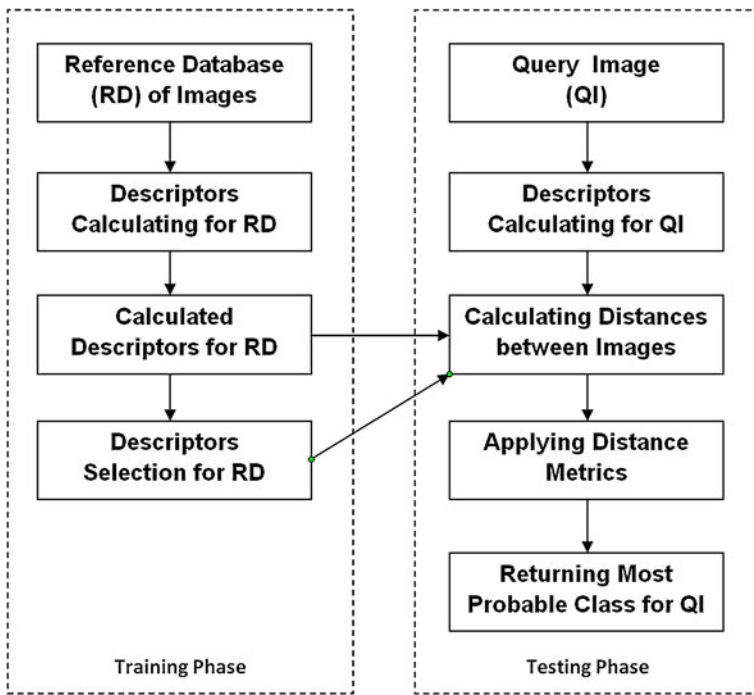


Fig. 7 Workflow of VCC framework

between query and reference images. To decide which category from RD is the most probable for each QI we apply distance metrics over the list of distances. We can use optionally one from three metrics.

$$M_1 = avg(dist((QI|A)), \tag{7}$$

where: $dist()$ is a distance between QI and images from each class of RD, A is a one of image classes from RD and $avg()$ is an average function.

$$M_2 = sum\left(\frac{\sum}{\|dist(A)\| \cdot A_n}\right), \tag{8}$$

where: $\|dist\|$ is a normalized distance between QI and each image from RD, A_n is a quantity of images in each class and \sum is a quantity of the whole RD,

$$M_3 = \frac{freq(A) \cdot \sum}{A_n}, \tag{9}$$

where: $freq(A)$ is a occurrence frequency of a particular class image in first N the most similar images from RD.

The M_1 represents an average distance between QI and the set of the most similar images from RD. It states that the smaller average distance in global means more probable similarity. The M_2 is the sum of inverse normalized distances to the RD images in respect to not equal image classes in RD. It shows which class from RD is the closest to QI. The M_3 counts frequency of appearance image from particular

RD class in the most similar to QI images. To reduce computations, distances from one hundred most similar images are taken into account in metrics counting process ($N=100$). The metrics are counted for each descriptor independently.

At this point, as a result a list of most similar classes to the QI is created according to particular descriptor. The last step is to merge results from various descriptors. For each RD type various weights can be established in order to achieve better results than with constant values.

5 Experimental results

Because the expected effectivity levels as well as assumed application areas are quite different for both presented MMR methods, we have decided to divide this section into two main subsections where performed tests and the obtained results are presented individually according to both proposed approaches.

5.1 RT approach accuracy and runtime performance

Analysis of the classification accuracy of the Real-Time approach have been performed according to the Overall Success Rate measure (*OSR*) defined as follows ([61]):

$$OSR = \frac{1}{n} \sum_{i=1}^k n_{i,i}, \quad (10)$$

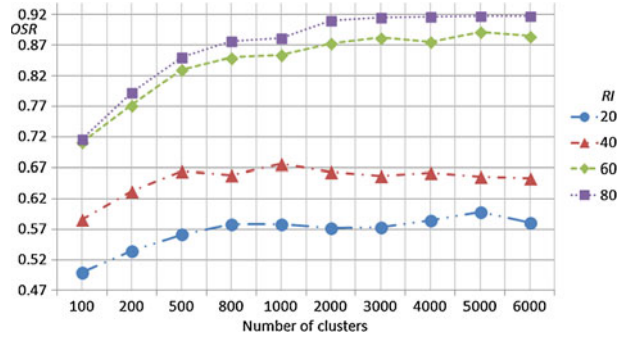
where: n is a number of test images, k is a number of estimated classes and $n_{i,i}$ —entries of the main diagonal of the confusion matrix.

The total number of test images selected for our experiments is 2499. We have decided however to check first how the *OSR* of our RT approach depends on various sizes of a test set (n). We have performed above analysis with respect to different

Table 2 *OSR* of RT approach according to various number of reference images (*RI*) and different number of clusters, for some selected different sizes of a test dataset (n)

<i>RI</i>	n	Number of clusters								
		100	200	500	1000	2000	3000	4000	5000	6000
20	1500	0.50	0.54	0.57	0.577	0.571	0.578	0.591	0.611	0.589
	2000	0.50	0.53	0.56	0.575	0.565	0.572	0.586	0.600	0.580
	2499	0.50	0.53	0.56	0.578	0.572	0.573	0.584	0.598	0.581
40	1500	0.59	0.62	0.66	0.673	0.659	0.662	0.669	0.660	0.656
	2000	0.59	0.63	0.66	0.673	0.659	0.659	0.659	0.655	0.651
	2499	0.59	0.63	0.66	0.676	0.663	0.657	0.663	0.655	0.654
60	1500	0.72	0.78	0.84	0.858	0.875	0.885	0.879	0.897	0.886
	2000	0.71	0.78	0.83	0.857	0.878	0.883	0.876	0.896	0.884
	2499	0.71	0.77	0.83	0.855	0.873	0.883	0.876	0.892	0.885
80	1500	0.72	0.80	0.85	0.885	0.909	0.917	0.915	0.913	0.915
	2000	0.72	0.80	0.85	0.883	0.913	0.915	0.915	0.914	0.919
	2499	0.72	0.79	0.85	0.882	0.910	0.916	0.917	0.918	0.918

Fig. 8 *OSR* versus various number of clusters, for different *RI* values ($n = 2499$)



amounts of reference images *RI* (where a given *RI* value represents the number of reference images per each car model) as well as various numbers of clusters creating the vocabulary.

The results of this analysis are depicted in Table 2.

Above analysis let us to notice that classification accuracy of our RT approach does not depend in general on a number of classified instances (n). There are very little dissimilarities in *OSR* distribution over different sizes of a test dataset for fixed *RI* value. This feature promises well, because it raises our hope to keep a steady accuracy level also after utilization of the RT approach. Table 2 shows however, that the classification accuracy depends on a number of reference images (*RI*) given per each car model as well as on a number of predefined clusters. Sample illustration of these dependences is depicted in Fig. 8.

Both, Fig. 8 and Table 2, show that the highest *OSR* values are obtained for *RI* equal to 80 and for a number of clusters which range from 3000 to 6000.

The bigger the number of clusters, the longer the durations of both the training and the testing phase. An illustration of this trend, in case of the testing phase, is depicted in Fig. 9.

In addition, Fig. 9 shows that runtime performance of the testing phase of the RT approach does not depend on the number of reference images. A similar relation is also obligatory for the training phase. Average durations of the whole testing phase for $RI = 80$ and for number of clusters bigger than 1000, are given in Table 3. Taking into account that the frame rate of the surveillance cameras have been assumed to 25 fps, we have decided finally to recommend 3000 as the most proper number of clusters for the RT approach.

Durations of the two main parts of the testing phase which are the SVoOC vector calculation (including previous operations of loading a query image and its

Fig. 9 Duration of the training phase (in ms) vs. various number of clusters ($n = 2499$)

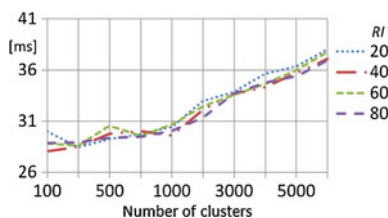


Table 3 Average durations of the testing phase for $RI = 80$ ($n = 2499$)

Number of clusters	2000	3000	4000	5000	6000
Duration [ms]	34.19	37.93	40.51	42.58	45.64

descriptors generation) and class prediction (see Fig. 6), for the above recommended RT approach configuration, are 33.69 ms and 4.24 ms, respectively. Above runtime performance outcomes have been achieved using the computer of parameters as follows: CPU—DualCore Intel Core i5 650, 3200 MHz (24×133), RAM—2x Kingston 2GB DDR3-1333 DDR3, system—Windows Server 2008 R2 Enterprise (64-bit).

The confusion matrix constructed for the recommended RT classifier is presented in Table 4. The true classes and the estimated classes, as defined by the RT classifier (see Fig. 3), are noted C_i and \hat{C}_i , respectively. Index i , according to the number of analyzed models, varies from 1 to 17.

Confusion matrix gives an additional information about classes which are commonly mislabeled one as another. As it is illustrated in Table 4, the RT MMR approach confuses, however very slightly, some similar models of cars. It is especially noticeable for cars which come from the same automaker. Car models confused mainly in the RT approach are as follows:

- Opel Astra II with Opel Astra I (in 13 in 237 cases),
- Opel Zafira A with Opel Astra II (in 10 in 143 cases),
- Ford Mondeo Mk III with Ford Focus C-MAX (in 12 in 191 cases),
- Ford Mondeo Mk III with Ford Focus Mk1 (in 10 in 191 cases),
- Opel Astra II with Opel Vectra B (in 15 in 237 cases).

Remaining cases of misclassifications are insignificant.

Table 4 Confusion matrix for $RI = 80$ and number of clusters equal to 3000 ($n = 2499$)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}
\hat{C}_1	90	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
\hat{C}_2	1	144	13	0	0	0	0	0	0	0	0	0	0	0	6	0	0
\hat{C}_3	0	1	193	0	0	0	0	0	0	0	1	1	0	2	2	10	1
\hat{C}_4	0	0	1	164	1	0	0	0	1	0	0	0	0	0	0	0	6
\hat{C}_5	0	3	2	1	119	3	0	1	3	2	2	0	1	0	2	3	0
\hat{C}_6	0	3	5	1	5	164	0	0	0	0	5	1	0	2	1	3	0
\hat{C}_7	0	1	1	0	0	0	78	0	2	0	1	12	0	3	0	0	0
\hat{C}_8	0	0	1	0	0	0	0	54	5	0	0	10	0	0	0	0	0
\hat{C}_9	0	0	0	0	0	0	0	7	51	0	0	5	0	0	0	0	0
\hat{C}_{10}	0	0	0	0	0	0	0	0	0	140	0	0	3	5	0	0	0
\hat{C}_{11}	0	1	1	0	0	0	0	0	0	0	138	0	0	0	0	0	0
\hat{C}_{12}	1	0	2	0	0	0	1	2	6	0	0	160	0	3	0	1	0
\hat{C}_{13}	0	0	0	0	0	0	0	0	0	0	0	0	155	2	0	0	0
\hat{C}_{14}	0	0	0	0	0	0	0	0	0	0	0	0	1	168	0	0	0
\hat{C}_{15}	0	3	15	0	0	0	0	1	0	0	1	2	0	4	204	1	0
\hat{C}_{16}	0	0	3	0	1	0	0	0	0	0	1	0	0	0	0	125	0
\hat{C}_{17}	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	142

Table 5 The VCC accuracy analysis according to different metrics and descriptors

Descriptors	M_1	M_2	M_3
Edge histogram	0.814	0.825	0.692
SIFT	0.804	0.812	0.706
SURF	0.802	0.779	0.683

5.2 VCC approach performance and accuracy

The Visual Content Classification approach uses the same database as RT one. Therefore accuracy and performance analyses have been performed with similar methods. For accuracy analysis previously defined OSR measure has been used. Because detecting cars with color features is not the point only three VCC descriptors are useful in MMR scenario. The following analysis is done with Edge Histogram (EH), SIFT and SURF.

Results of accuracy tests of VCC approach were presented in Table 5. The best results with single descriptor were achieved with M_2 . However in some cases differences between metrics are small, the changes have linear nature. The comparison shows advantage of M_2 but because of strict statistical differences all metrics are taken into further consideration.

With merging information from different descriptors more unique set of features are created. The obtained descriptor takes into account more heterogeneous object features which result in higher accuracy.

The descriptors can be merged with various weights. For specific RD different weights can provide better and worse results. In MMR scenario, local discrete grid is used to get optimal weights for each descriptor. The results presented in Table 6 prove that variant weights improves accuracy rate. These values should be computed for various RD independently.

In the same way as for RT approach, we present confusion matrix for VCC method—in Table 7.

The confusion matrix shows some similarity but also differences between approaches. The best result was achieved in c_6 model class (Opel Corsa B) in which accuracy is 100 %. The worst result (accuracy below 90 %) is recorded with C_3 class (Opel Astra II). The most common mislabels are:

- Opel Astra II with Opel Corsa B (in 3 in 237 cases),
- Opel Astra II with Opel Vectra B (in 4 in 237 cases),
- Opel Astra II with Opel Zafira A (in 3 in 237 cases),
- VW Touran with VW Passat B5 FL (in 3 in 190 cases).

The VCC is not dedicated for real-time analysis. The average time needed to process one frame is approximately 1.93 sec (in the same machine as for RT approach).

Table 6 The VCC accuracy analysis with weighted descriptors merging

Edge histogram	SIFT	SURF	M_1	M_2	M_3
1	1	1	0.913	0.939	0.840
0.6	0.7	0.5	0.946	0.951	0.832
0.7	0.8	0.5	0.937	0.972	0.822
0.6	0.6	0.8	0.918	0.945	0.881

Table 7 Confusion matrix for M_2 with weights: 0.7, 0.8, 0.5 for EH, SIFT, SURF, respectively ($n = 2499$)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}
\hat{C}_1	91	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
\hat{C}_2	1	152	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0
\hat{C}_3	0	1	211	0	0	0	0	0	0	0	0	1	0	0	0	2	0
\hat{C}_4	0	0	1	166	0	0	0	0	1	0	0	0	0	0	0	0	0
\hat{C}_5	0	0	2	0	124	0	0	1	0	2	0	0	0	1	0	2	0
\hat{C}_6	0	1	3	1	2	167	0	0	0	0	0	1	0	0	1	0	0
\hat{C}_7	0	1	1	0	0	0	78	0	0	0	1	0	0	0	0	0	1
\hat{C}_8	0	0	2	0	0	0	0	61	1	0	0	0	0	1	0	0	0
\hat{C}_9	0	0	0	0	0	0	1	1	65	0	0	1	0	0	0	0	1
\hat{C}_{10}	0	0	2	0	0	0	0	1	1	142	0	1	0	0	0	0	0
\hat{C}_{11}	0	1	0	0	0	0	0	0	1	0	148	0	0	0	0	0	0
\hat{C}_{12}	0	0	2	0	0	0	0	0	0	0	0	186	0	0	0	0	0
\hat{C}_{13}	0	0	0	0	0	0	0	0	0	0	0	0	160	3	0	1	2
\hat{C}_{14}	0	0	1	0	0	0	0	0	0	0	0	0	1	185	0	0	0
\hat{C}_{15}	0	1	4	0	0	0	0	1	1	0	0	0	0	0	212	1	0
\hat{C}_{16}	0	0	3	0	0	0	0	0	0	0	0	1	0	0	0	137	0
\hat{C}_{17}	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0	145

6 Conclusions and further work

The paper presents detailed description of two approaches for MMR scenario. The developed methods have been trained and experimentally tested on datasets prepared for training and testing (datasets can be downloaded from [7]). Datasets contain great number of cars front views grouped into 17 classes (one for each car model). The tests prove that RT approach and VCC one gives good results, 91.7 % and 97.2 % of correctly recognized classes, respectively. Taking into account database size and processing time (for RT approach) presented algorithms are an important contribution in car make and models recognition techniques. The results can be combined with the grill part of a car detector and use in surveillance applications.

The further work will put emphasis on increasing number of classes in RD. We are currently working on preparing new version with 54 different car models. Creating methods for larger database will be the next challenge.

Acknowledgements We want to address our special thanks to our colleague Mariusz Rychlik for his valuable contribution to this work. We would like to thanks also to Piotr Bogacki for his help in VCC algorithm implementation.

The work was co-financed by the European Union from the European regional Development Fund, as a part of the Innovative Economy Operational Programme INSIGMA no. POIG.01.01.02-00-062/09.

The numerical experiments reported in this paper have been performed using computational equipment purchased in the framework of the EU Operational Programme Innovative Economy (POIG.02.02.00-26-023/09-00) and the EU Operational Programme Development of Eastern Poland (POPW.01.03.00-26-016/09-00).

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. <http://www.orpixon.com/#!VehicleRecognition/c1dpm>. Viewed 8 April 2013
2. http://www.neurosoft.pl/index.php?lang=de&page_name=NeuroCar&wikipage=Neurosoft_Produkty_NeuroCar_Moduly. Viewed 8 April 2013
3. <http://insigma.kt.agh.edu.pl/>. Viewed 8 April 2013
4. http://vision.ucsd.edu/belongie-grp/research/carRec/car_rec.html. Viewed 8 April 2013
5. http://docs.opencv.org/doc/user_guide/ug_traincascade.html. Viewed 8 April 2013
6. <http://people.csail.mit.edu/fergus/icc2005/bagwords.html>. Viewed 8 April 2013
7. <http://research.wstkt.pl>. Viewed 8 April 2013
8. Ahmad N, An Y, Park J (2012) An intrinsic semantic framework for recognizing image objects. *Multimed Tools Appl* 57:423–438
9. Anagnostopoulos CNE, Anagnostopoulos IE, Loumos V, Kayafas E (2006) A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Trans Intell Transport Syst* 7(3):377–392
10. Anthony D (2005) More local structure information for make-model recognition. Dept. of Computer Science, University of California at San Diego. Taken from <http://www-cse.ucsd.edu>
11. Baran R, Dziech A (2010) Contour compression scheme combining spectral and spatial domain methods. *MAaM* 56(12):1409–1412
12. Bauer J, Sünderhauf N, Protzel P (2007) Comparing several implementations of two recently published feature detectors. In: Proc. of the int. conf. on intelligent and autonomous systems. IAV, Toulouse, pp 143–148
13. Bay H, Ess A, Tuytelaars T, VanGool L (2008) Speeded-up robust features (surf). *Comput Vis Image Underst* 110(3):346–359
14. Bernardo JM, Smith AFM (1994) Bayesian theory. Wiley, New York
15. Butzke M, Silva AG, Hounsell MDS, Pillon MA (2008) Automatic recognition of vehicle attributes- color classification and logo segmentation. In: HIFEN, vol 32, pp 293–300
16. Cands E, Demanet L, Donoho D, Ying L (2006) Fast discrete curvelet transforms. *Multiscale Model Simul* 5(3):861–899
17. Cands EJ, Donoho DL (2004) New tight frames of curvelets and optimal representations of objects with piecewise singularities. *Comm Pure Appl Math* 57(2):219–266
18. Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. *ACM Trans Intel Syst Technol* 2(3): 2:27:1–27:27
19. Clady X, Negri P, Milgram M, Poulénard R (2008) Multi-class vehicle type recognition system. In: Proc. of the 3rd IAPR workshop on artificial neural networks in pattern recognition, pp 228–239
20. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
21. Cover T, Hart P (1967) Nearest neighbor pattern classification. *IEEE Trans Inform Theory* 13(1):21–27
22. Dalka P, Czyzewski A (2010) Vehicle classification based on soft computing algorithms. In: Szczuka MS, Kryszkiewicz M, Ramanna S, Jensen R, Hu Q (eds) RSCTC, lecture notes in computer science, vol 6086. Springer, pp 70–79
23. Dlagnekov L (2005) Video-based car surveillance: license plate, make, and model recognition. Master's thesis, University of California, San Diego
24. Do MN, Vetterli M (2005) The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans Image Process* 14(12):2091–2106
25. Gasca M, Sauer T (2000) On the history of multivariate polynomial interpolation. *J Comput Appl Math* 122(1–2):23–35
26. Glowacz A, Mikrut Z, Pawlik P (2012) Video detection algorithm using an optical flow calculation method. In: Dziech A, Czyzewski A (eds) Multimedia communications, services and security, communications in computer and information science, vol 287. Springer, Berlin Heidelberg, pp 118–129

27. Gorzalczy MB (1999) On some idea of a neuro-fuzzy controller. *Inf Sci* 120:69–87
28. Gorzalczy MB (2002) Computational intelligence systems and applications, neuro-fuzzy and fuzzy neural synergisms. Physica-Verlag, Springer-Verlag Co., Heidelberg, New York
29. Gorzalczy MB, Rudziński F (2010) A modified Pittsburg approach to design a genetic fuzzy rule-based classifier from data. In: *Lecture notes in artificial intelligence*, vol 6113. Springer-Verlag, Berlin, Heidelberg, pp 88–96
30. Gorzalczy MB, Rudziński F (2012) Accuracy vs. interpretability of fuzzy rule-based classifiers: an evolutionary approach. In: *Lecture notes in computer science*, vol 7269. Springer-Verlag, Berlin, Heidelberg, pp 222–230
31. Gu H-Z, Lee S-Y (2012) A view-invariant and anti-reflection algorithm for car body extraction and color classification. *Multimed Tools Appl* 65(3):387–418
32. Gunn SR (1998) Support vector machines for classification and regression. Technical report, image speech and intelligent systems research group. University of Southampton
33. Gupta MM, Gorzalczy MB (1992) Fuzzy neuro-computational technique and its application to modeling and control. In: *Proc. of IEEE int. conf. on fuzzy systems*, pp 1271–1274
34. Hsu C, Lin C (2002) A comparison of methods for multi-class support vector machines. *IEEE Trans Neural Netw* 13(2):415–425
35. Huang H, Zhao Q, Jia Y, Tang S (2008) A 2dlda based algorithm for real time vehicle type recognition. In: *Proc of the 11th int. IEEE conf. on intelligent transportation systems*, pp 298–303
36. Iqbal U, Zamir SW, Shahid MH, Parwaiz K, Yasin M, Sarfraz MS (2010) Image based vehicle type identification. In: *Proc. of the int. conf. on information and emerging technologies*, pp 1–5
37. McLachlan GJ (2004) *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons, Canada
38. Janowski L, Kozłowski P, Baran R, Romaniak P, Glowacz A, Rusc T (2012) Quality assessment for a visual and automatic license plate recognition. *Multimed Tools Appl*. doi:10.1007/s11042-012-1199-5
39. Juan L, Gwun O (2009) A comparison of sift, pca-sift and surf. *IJIP* 3(4):143–152
40. Kazemi FM, Samadi S, Poorreza HR, Akbarzadeh-T M-R (2007) Vehicle recognition based on fourier, wavelet and curvelet transforms—a comparative study. *Int J Comput Sci Net Sec* 7(2):130–135
41. Kazemi FM, Samadi S, Poorreza HR, Akbarzadeh-T MR (2007) Vehicle recognition using curvelet transform and svm. In: *Proc. of the 4th int. conf. on information technology*, pp 516–521
42. Lee HJ (2006) Neural network approach to identify model of vehicles. In: *Proc. of the 3rd int. conf. on advances in neural networks*, vol 3, pp 66–72
43. Leszczuk M, Janowski L, Romaniak P, Glowacz A, Mirek R (2011) Quality assessment for a licence plate recognition task based on a video streamed in limited networking conditions. In: *Multimedia communications, services and security, communications in computer and information science*, vol 149. Springer, Berlin Heidelberg, pp 10–18
44. Li M, Yuan B (2005) 2d-lda: a statistical linear discriminant analysis for image matrix. *Pattern Recogn Lett* 26(5):527–532
45. Lloyd SP (1982) Least squares quantization in pcm. *IEEE Trans Inform Theory* 28(2): 129–137
46. Lowe DG (1999) Object recognition from local scale-invariant features. In: *Proc. of the ICCV '99*, vol 2. IEEE Computer Society, pp 1150–1157
47. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: *Proc. of the 5th berkeley symposium on mathematical statistics and probability*, vol 1, pp 281–297
48. Matiolanski A, Guzik P (2011) Automated optimization of object detection classifier using genetic algorithm. In: Dziech A, Czyzewski A (eds) *Multimedia communications, services and security, communications in computer and information science*, vol 149. Springer, Berlin Heidelberg, pp 158–164
49. Munroe DT, Madden MG (2005) Multi-class and single-class classification approaches to vehicle model recognition from images. In: *AICS-05*, pp 93–102
50. Negri P, Clady X, Milgram M, Poulenard R (2006) An oriented-contour point based voting algorithm for vehicle type classification. In: *Proc. of the 18th int. conf. on pattern recognition, ICPR'06*, pp 574–577
51. Park DK, Jeon YS, Won CS (2000) Efficient use of local edge histogram descriptor. In: *Proceedings of the 2000 ACM workshops on multimedia, MULTIMEDIA'00*. ACM, New York, pp 51–54

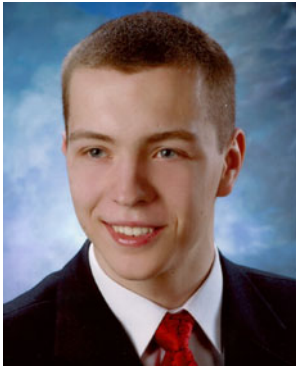
52. Pearce G, Pears N (2011) Automatic make and model recognition from frontal images of cars. In: Proc of the 8th IEEE int. conf. on advanced video and signal-based surveillance (AVSS), pp 373–378
53. Petrovic VS, Cootes TF (2004) Analysis of features for rigid structure vehicle type recognition. In: British machine vision conference, pp 587–596
54. Psyllos A, Anagnostopoulos CN, Kayafas E, Loumos V (2008) Image processing & artificial neural networks for vehicle make and model recognition. In: Proc. of the 10th int. conf. on applications of advanced technologies in transportation, vol 5, pp 4229–4243
55. Rahati S, Moravejian R, Mohamad E, Mohamad F (2008) Vehicle recognition using contourlet transform and svm. In: Proc. of the 5th int. conf. on information technology: new generations, pp 894–898
56. Salembier P, Sikora T (2002) Introduction to MPEG-7: multimedia content description interface. John Wiley & Sons, Inc., New York
57. Ukasha A, Dziech A, Elsharif E, Baran R (2009) An efficient method of contour compression. In: Proc. of the int. conf. on visualization, imaging and image processing, pp 213–218
58. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
59. Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10(5):988–999
60. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proc. of the 2001 IEEE computer society conference on computer vision and pattern recognition, vol 1, pp I–511–I–518
61. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco
62. Zafar I, Acar BS, Edirisinghe EA (2007) Vehicle make & model identification using scale invariant transforms. In: Proc. of the 7th IASTED international conference on visualization, imaging and image processing, pp 271–276
63. Zafar I, Edirisinghe EA, Acar BS (2009) Localized contourlet features in vehicle make and model recognition. In: Proc. of SPIE-IS&T electronic imaging, image processing: machine vision applications II, vol 7251, pp 725105–725105–9
64. Zafar I, Edirisinghe EA, Acar S, Bez HE (2007) Two-dimensional statistical linear discriminant analysis for real-time robust vehicle-type recognition. In: Proc. of SPIE-IS&T electronic imaging, real-time image processing 2007, vol 6496, pp 649602–649602–8



Remigiusz Baran received the M. Sc. degree in Electrical Engineering from the Faculty of Electrical and Control Engineering, Kielce University of Technology in 1993 and the PhD degree in Telecommunications from the Faculty of Electrical, Control, Electronic and Computer Engineering, AGH University of Science and Technology in Krakow in 2004. He actively participates (or has participated) in EU and national projects, e.g. in INDECT, INSIGMA, CALIBRATE, INTOM, OASIS Archive, InWAS. His fields of scientific and professional interest are related to embedded DSP systems as well as digital image processing with image compression, feature extraction and object recognition emphasis.



Andrzej Glowacz received PhD in Telecommunications from the AGH University of Science and Technology, in 2007. Currently he is Assistant Professor in AGH University. Andrzej Glowacz has been working in numerous industrial works, and the EU research projects including: INSIGMA, INTOM (in both as Deputy Project Coordinator), INDECT, Future Internet Engineering, GAMA, OASIS Archive, CARMEN, DAIDALOS, DAIDALOS 2, EuroNGI and EuroFGI. His main professional areas are: intelligent information systems, multimedia systems, pattern recognition, wireless QoS, modern transport protocols and advanced systems programming. He is author of over seventy scientific papers and technical reports, he also serves as a reviewer of international journals and conferences.



Andrzej Matiolanski He is a PhD student at the Department of Telecommunications of AGH University of Science and Technology. He has received his M.Sc. degree from the Faculty of Physics and Applied Computer Science of AGH University of Science and Technology in 2010. His research interests include computer vision, image processing and super-resolution algorithms. He has actively participated in both national and international research projects like INDECT, INSIGMA and MAYDAY EURO 2012.