Multimed Tools Appl (2016) 75:10471–10493
DOI 10.1007/s11042-015-3151-y

A smart camera for the surveillance of vehicles in intelligent transportation systems

Remigiusz Baran[1] · Tomasz Rusc[2] · Paweł Fornalski[3]

Received: 4 November 2014 / Revised: 16 November 2015 / Accepted: 9 December 2015 /
Published online: 26 December 2015

Abstract The paper presents a smart camera aimed at security and law enforcement applications for intelligent transportation systems. An extended background is presented first as a scholar literature review. The smart camera components and their capabilities for automatic detection and recognition of selected parameters of cars, as well as different aspects of the system efficiency, are described and discussed in detail in subsequent sections. Smart features of make and model recognition (MMR), license plate recognition (LPR) and color recognition (CR) are highlighted as the main benefits of the system. Their implementations, flowcharts and recognition rates are described, discussed and finally reported in detail. In addition to MMR, three different approaches, referred to as bag-of-features, scalable vocabulary tree and pyramid match, are also considered. The conclusion includes a discussion of the smart camera system efficiency as a whole, with an insight into potential future improvements.

Keywords Intelligent camera · Surveillance of vehicles · Color and make and model recognition · License plate recognition · Intelligent transportation system

✉ Remigiusz Baran
r.baran@tu.kielce.pl

Tomasz Rusc
tomasz.rusc@ujk.edu.pl

Paweł Fornalski
pawelf2@o2.pl

[1] Faculty of Electrical Engineering, Automatics and Computer Science, Kielce University of Technology, al. 1000-lecia P.P. 7, 25-314 Kielce, Poland

[2] Institute of Physics, Jan Kochanowski University, ul. Swietokrzyska 15, 25-406 Kielce, Poland

[3] Department of Telecommunications, AGH University of Science and Technology, al. Mickiewicza 30, 30-059, Krakow, Poland

1 Introduction

The digital camera market experienced a major boom in the late 1990s and early 2000s. This is due to technological advancements in chip manufacturing, progress in embedded system design, the coming-of-age of CMOS (complementary metal oxide semiconductor) image sensors, and so on [65]. In particular, the development of CMOS image sensors – cheaper to manufacture than CCDs – boosted this growth. Together with stand-alone digital cameras and camera phones, accessibility and demand for smart cameras also increased. According to [65], While the primary function of a normal camera is to provide video for monitoring and recording, smart cameras are usually designed to perform specific, repetitive, high-speed and high-accuracy tasks. Machine vision and intelligent video surveillance systems (IVSS) are the most common applications.

In general, surveillance camera systems aim to observe a given area in order to increase safety and security. In [67], a surveillance system for the detection of individuals within a dense crowd from a scene captured by a time-of-flight camera is presented. It makes it possible to detect and track every person's movement, and to analyze this movement to compare it to the behavior of the entire crowd. Dedicated software enhances these capabilities by providing analysis of the situation, for example. Smart cameras are also widely used in numerous road transportation systems, including traffic management, surveillance, security and law enforcement, automated parking garages [13], driver assistance and control access systems, etc. A state-of-the-art application related to self-guided and driverless transport vehicles is presented in [62]. The most common and well-known application from the category of traffic surveillance and law enforcement is license plate recognition (LPR) [39]. Due to growing demand, other categories of vehicle classification have also been added recently. Make and model recognition (MMR) [59] and color recognition (CR) of cars are relatively new functionalities.

The smart camera system presented in this paper also belongs to the category of traffic surveillance and law enforcement applications. According to the goals of the INSIGMA R&D project [38], under which the presented system has been developed, it also incorporates the three functionalities mentioned above: LPR, MMR and CR.

For clarity of presentation, the rest of the paper is organized as follows. The current section presents an extensive literature review within the framework of the subject matter. In Section 2, the overall architecture of the presented smart camera system is introduced. The MMR, LPR and CR components of the system are presented in detail in Sections 3, 4 and 5, respectively. In Section 6, the system's efficiency is reported and discussed. Conclusions, with an insight to potential future improvements, are drawn in Section 7.

1.1 Literature review

As mentioned in Section 1, numerous computer vision approaches and their applications are used in various current video-based roadway transportation systems. Due to their extensive capabilities, such systems are categorized as intelligent transportation systems (ITS) by researchers [68] and legislators around the world [46].

Various approaches to ITS and different aspects of their architectures are presented in detail in [48]. Methods related to traffic surveillance including tracking and recognition of vehicles, traffic flow monitoring [12] and driver assistance applications are also discussed in the paper. Typical driver assistance applications address lane departure and pedestrian detection problems. Traffic flow monitoring applications may prove to be useful in traffic optimization and road incident management systems. For example, they are able to evaluate the length of traffic queues [26] or estimate critical flow time periods [14].

The use of traffic cameras for security and law enforcement purposes has many practical benefits. First of all, video sequences recorded by such cameras can be used as evidence for police forces or insurance companies. They can be browsed to review events of interest at different points in time. Moreover, when a given event has been registered by a number of cameras, it can be analyzed from different views. As well as post-hoc analysis, views registered by traffic cameras are usually monitored in real time by human operators in control centers.

Computer vision techniques can significantly expand the capabilities mentioned above. Segmentation, extraction of salient regions, feature-based detection and classification, video indexing and retrieval, etc., can radically increase the number of factors taken into account during the analysis and, in this way, appreciably improve its accuracy. This helps operators avoid making wrong decisions, as they are supported by automatically-generated alarms and advised by powerful content-oriented analysis engines.

As mentioned in Section 1, the smart camera system presented in this paper can also be included in the category of security and law enforcement applications. Development of the presented architecture and the related research form a part of the INSIGMA R&D project [38] in which the authors of this paper are currently involved. One of INSIGMA's objectives is to develop software which will be able to process video sequences registered by surveillance cameras in order to detect and recognize selected features of cars, including vehicle manufacturer and model, number plates and color.

Recognition of vehicle number plates, known as LPR (or as automatic number plate recognition (ANPR), especially in the UK), is one of the most popular and the earliest available application from that category. Most of the existing LPR systems use similar schemes, which usually include the following successive processing steps: preprocessing, plate detection, localization and horizontal alignment, and character segmentation and recognition. The preprocessing step is generally required to improve the quality of the processed images. This may address objectives such as shadow removal, characters enhancement, background suppression, strengthening of edges, etc. These goals are usually achieved by various binarization methods, including the Otsu binarization [37], adaptive binarization techniques such as variable thresholding [16] or the Sauvola method [2], and other non-adaptive methods, as in [64]. Strengthening of edges is achieved by combining selected binarization methods with techniques including greying, normalizing, histogram equalization, etc., as reported in [41]. Other preprocessing objectives, such as noise removal and general image enhancement, are achieved by applying wavelet-based filters [55] and the top-hat transform [5], respectively.

There are also many different approaches to license plate detection and localization. One of the simplest (albeit least efficient) methods is based on histograms obtained as a result of horizontal and vertical projections through the image [37]. In [40] the density-based region growing method is also shown as being capable of detecting license plates. In [70], connected component analysis followed by the labeling technique is reported as an efficient method. Various edge detection algorithms, including the Canny edge detector [82] and Roberts cross operator [73], have also been found to be effective. Other approaches to license plate detection and localization are based on different types of salient features including SIFT [17], discrete wavelet transform [72], neural networks [28,33], etc.

Since objects of interest in video footage are usually distorted, an additional step of horizontal alignment of license plates must follow to improve detection and localization. A number of techniques can be applied to correct the skew of localized and extracted plates. The most effective are the Hough transform [4] and a method based on appropriate geometric constraints, as reported in [42].

The step following the successful skew correction is character segmentation. There are many different approaches to this task, some of which refer to horizontal and vertical projections through the extracted license plate image. Used alone or in combination with selected geometrical constraints (related to assumptions about the height and width of characters) are reported as being effective in [74] and [61], respectively. A grey-level quantization combined with appropriate morphology analysis were also performed to locate and separate individual characters [40]. Another technique examined within the framework of this subject [78] is connected component analysis. In [79], it was shown that an in-depth analysis based on a combination of selected binarization methods results in good character segmentation. In [84], characters, even when they are adhesive or cracked, are accurately extracted thanks to the spatial scalability of their contours. Characters extracted in this way are then segmented using a matching algorithm using adaptive templates.

The final step is character recognition. The most popular approaches to this task are based on different models of neural networks, including artificial neural networks (ANNs) [27], probabilistic neural networks (PNNs) [54], and back propagation neural networks (BPNN) [83]. Within the category of machine learning methods, support vector machine (SVM) [18] -based approaches are also popular, as reported for example in [6]. Among other methods, template matching [77] and optical character recognition (OCR) [81] are also frequently used. Comprehensive surveys of LPR techniques can be found in [23] and [63].

Despite the fact that MMR frameworks are already being applied in selected security systems [51], the volume of related scientific literature is relatively low. This is most likely due to commercialization.

One of the first approaches to the MMR issue was presented in [58], where a combination of different types of features, extracted from frontal views of cars, was used to distinguish between different car models. Selected feature extraction algorithms (e.g., Canny edge detector, square mapped gradients, etc.) and various classifications methods (e.g., naive Bayes) were investigated in [57]. Another contour oriented approach [85] is reported in [50]. In this approach, contours, extracted using the Sobel filter, are transformed into complex feature arrays where only the contour points common to all images from the training set (of a given class) are represented. Such feature arrays, known as oriented-contour point matrices, are input into the classification procedure which uses four different measures including distance errors between oriented-contour points of the class model and the sample being examined. Another contour-based solution is presented in [3].

Methods described so far are based on features extracted from the spatial domain. There are also methods which operate in different transform domains [9,25]. An example of such an approach was presented in [43] where the discrete curvelet transform (DCT) [15] was shown; this transform domain feature extractor provides the best recognition rate out of the three being studied. In [43], the DCT was combined with a standard *k*-nearest neighbor (*k*NN) algorithm [19]. According to results reported in [35], SVN gives better results when combined with DCT, especially when the SVM one-against-one strategy is used. Similar research based on the contourlet transform [22] is presented in [60].

Other valuable approaches to MMR are also related to the scale invariant feature transform (SIFT) [49]. The effectiveness of SIFT-based MMR schemes was investigated and reported by the research team of Prof. Serge J. Belongie [11]. A simple matching algorithm, where SIFT descriptors computed for a given query image are matched directly, one by one, with descriptors determined for each of the reference images, is presented in [21]. This and other reports confirm that approaches based on SIFT [80] or the speeded-up robust features (SURF) method [34] are also promising for solving the MMR problem.

Vehicle color recognition (VCR) in outdoor conditions remains an unsolved problem. This is mainly due to lighting conditions, shadows and reflections of sunlight on the shiny vehicle surface. These problems make finding a suitable solution challenging.

In [31] a tri-state architecture including a Separating and Re-Merging (SARM) algorithm to effectively extract the car body and classify the vehicle color in challenging cases with unknown car type, unknown viewpoint, and no homogeneous light reflection conditions. In [24], in turn, different features, selected to represent various color spaces, and different classification methods (*k*NN, ANNs, and SVM), were analyzed with regard to the VCR task. The features were computed according to two selected views of the car: a smooth hood piece, and semi-frontal view. Sunlight reflections and filtering out vehicle parts irrelevant to the color recognition problem were the subjects of research reported in [36]. An effective approach based on color histograms and template matching was reported in [45]. The main objective of this approach was to find the required number of histogram bins. Color histograms, combined with principal component analysis (PCA), were examined in [56]. A different SVM based approach was proposed in [76]. The video color classification (VCC) algorithm presented in this paper was based on refining the foreground mask for removing the undesired region.

2 General assumptions and system architecture

According to the INSIGMA project's objectives, it has been assumed that smart surveillance cameras will be positioned over every traffic lane, including highways, streets, parking lots, etc. It has also been assumed that the resolution of M-JPEG video sequences recorded by these cameras should not be less than 4CIF. In other words, the expected minimum resolution of processed 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 for example 60 mm, the size of the camera field of view (FOV) from a distance of about 40 m is 2.35×1.76 m. FOV of the same size can be also obtained from a distance of about 5 m, but with a focal length equal to 8 mm. These relationships are illustrated in Fig. 1.

At its core, our smart camera implementation (known as the iCamera system), is a JVM-based system built using the Spring Framework [66]. It works on the Ubuntu 13.04 64 bit operating system. Despite this, to increase the system efficiency, its specialized modules (MMR, LPR, CR) have been written in C. To use their functionalities as well as to exchange data between them and the system backbone (the Camera Core), the Java Native Interface (JNI) framework has been used.

As illustrated in Fig. 2, the Camera Core receives the video stream from the Camera IP, and decodes and passes it on to subsequent modules. The decoded video frames are initially passed on to the Global Detection and Extraction (GDE) module. The task of this module is to detect

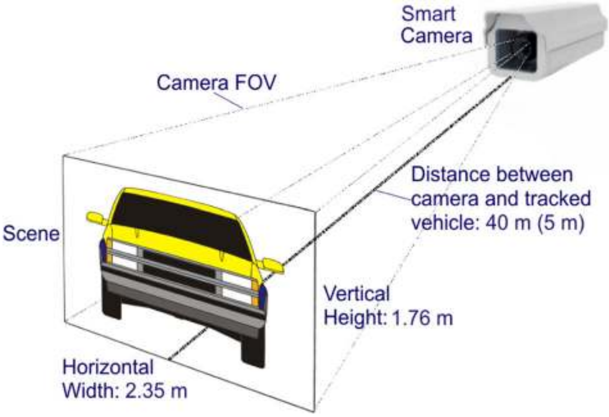


Fig. 1 Predetermined parameters of the camera's FOV [76]

(on a video frame from the camera) and then to extract (by cropping this frame) two Regions of Interest (ROIs). One of them - a sub-image containing the grill part of a car together with its headlights and indicator lights is for the MMR and CR modules. The other – a sub-image limited to the license plate area is for the LPR module.

Both ROIs are detected using two different Haar-like detectors, which have been trained concurrently according to MMR (CR) and LPR needs. More details of Haar-like detectors have been reported in [8]. Successful ROI detection (equivalent to car detection in FOV) causes the GDE to activate the MMR, CR and LPR modules.

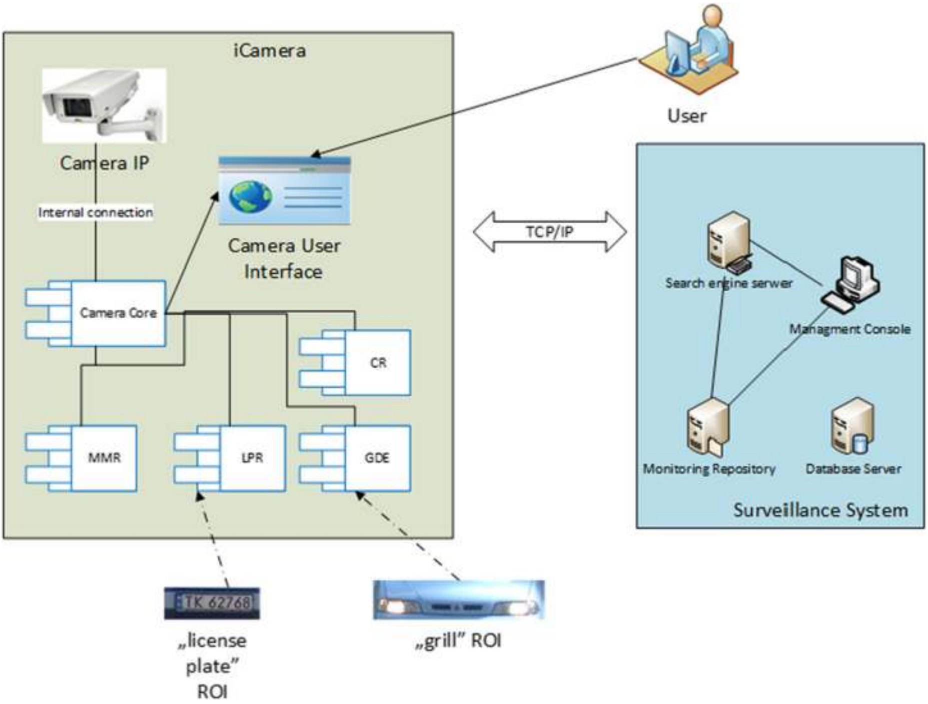


Fig. 2 Overall iCamera system architecture

After activation, the MMR, CR and LPR modules individually process ROIs passed to them from the GDE, and send the results of this processing back to the Camera Core. These results are metadata depending on the module which generates them.

In the case of MMR the returned metadata contains an alias name identifying the make and model of the car, which have been predicted by the classifier built into the module. In the case of LPR however, the metadata contains the text read from the license plate by the embedded OCR tool. This information like that is passed as Exif fields in the XML format. Selected examples of these fields are shown in Fig. 3.

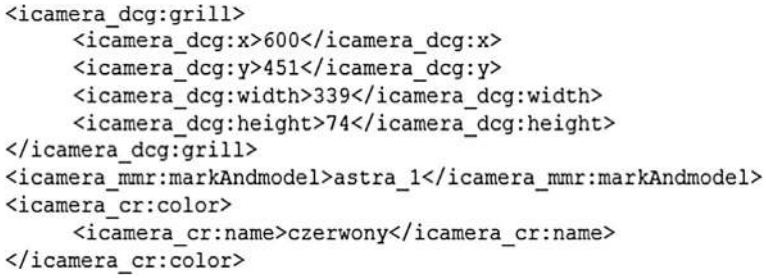
The input video stream is supplemented by additional data during its passage through the iCamera system. The added data are the previously mentioned Exif fields. The video stream extended in this way is finally passed to the user interface. The interface allows the user to control the iCamera system by for example stopping and starting the video streaming and enabling or disabling each of the specialized modules. Its current look is shown in Fig. 4.

The core thread of the iCamera system aggregates the prediction/OCR results referring to the same vehicle from successive neighboring video frames to obtain statistics and finally increase the system accuracy. Final accuracy in this case is proportional to the camera frame rate. However, the greater the frame rate the shorter the frame processing time. Of course, there are also other factors, for instance the efficiency of the hardware platform used, which impact the system's performance as well as its accuracy. These, and other aspects referring to iCamera system's efficiency, are discussed in Section 6.

3 Make and model recognition

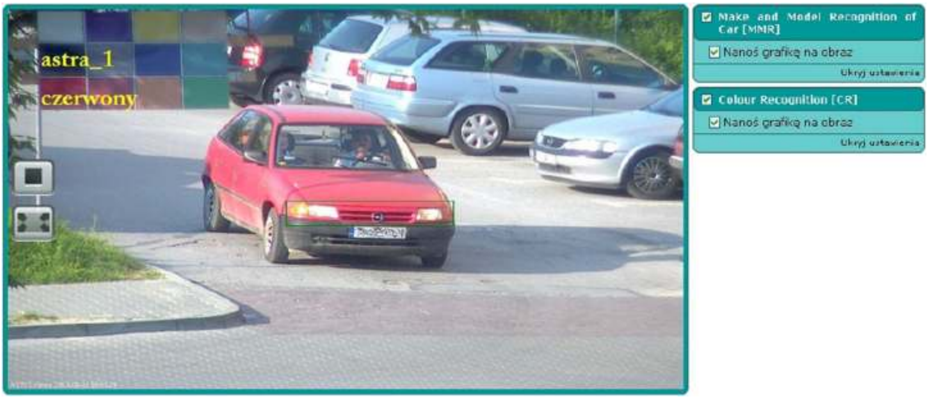
The MMR approach presented in this paper has originated in the Real-Time (RT) solution described in detail in [8] and, like that one is a feature-based classification procedure where makes and models of cars are predicted according to the Speeded Up Robust Feature (SURF) [10] descriptors. These descriptors are calculated for salient points (also known as key- or interest points) found in an image and, as such, can be treated as local features of an image, which enables identification of objects in the analyzed scene. In contrast to the approach described in [8], where the *k*-means clustering [29] was used in combination with the Support Vector Machines (SVM) [18], the scheme presented in this paper is based on the Scalable Vocabulary Tree (SVT) technique [52].

The reason why we started looking for a different solution for the MMR module was the duration of the training phase, which, in the case of the RT approach [8] (hereinafter referred as



```
<icamera_dcg:grill>
      <icamera_dcg:x>600</icamera_dcg:x>
      <icamera_dcg:y>451</icamera_dcg:y>
      <icamera_dcg:width>339</icamera_dcg:width>
      <icamera_dcg:height>74</icamera_dcg:height>
</icamera_dcg:grill>
<icamera_mmr:markAndmodel>astra_1</icamera_mmr:markAndmodel>
<icamera_cr:color>
      <icamera_cr:name>czerwony</icamera_cr:name>
</icamera_cr:color>
```

Fig. 3 Exif fields with metadata returned by GDE, MMR and CR modules, respectively. (The icamera_cr field contains value returned by the CR module. "Czerwony" is the polish name of the red color.)



☑ Make and Model Recognition of Car [MMR]

☑ Nanoś grafikę na obraz

Ukryj ustawienia

☑ Colour Recognition [CR]

☑ Nanoś grafikę na obraz

Ukryj ustawienia

Fig. 4 Sample look of the iCamera User Interface (Captions "astra_1" and "czerwony", displayed at the top left of the screen are the metadata returned by the MMR and CR modules, respectively. Because the LPR module is off at the moment – it has been deactivated using checkboxes situated at the top right of the screen, there is no information about registration numbers

SVM-based approach), is pretty long. In addition, it extends rapidly together with a growing number of trained classes in the SVM model, gaining finally an unacceptable level. To size the problem, for 17 classes for instance (the case reported in [8]), duration of the training phase takes about 20 h; for 45 car models it takes nearly 3 weeks (on the same computer system).

To solve this problem, the two following approaches have been examined in addition to the SVM-based one. One of them is just the SVT-based approach mentioned above, while the other one, based on LIBPMK - a Pyramid Match Toolkit [47], is related to Grauman and Darrell's Pyramid Match algorithm" [30]. The VocabTree2, by Noah Snavely [71] is in turn, an example of useful library where SVT implementation is popularized [1].

SVT is a technique which evolved from Bag-of-Features method [20] (in fact the RT approach was also organized according to a Bag-of-Features scheme). SVT is used to create a hierarchy of features, which are organized as tree nodes. To each of such tree nodes a cluster of features is assigned. As well as tree nodes, which are getting smaller at the successive tree levels, clusters of features assigned to these nodes are also getting smaller. As the successive levels of the tree are created, clusters of features are subdivided into smaller ones, using *k*-means algorithm. Despite this fact, SVT is able to generate a big codebook with light-weight calculations.

In VocabTree2 implementation [71], vocabulary tree is organized according to two main parameters, which are as follows:

- "depth", which determines the depth (number of levels) of the tree,
- "branching_factor", which defines the number of children of each node of the tree ("branching_factor" is equated with *k*).

At the beginning, the vocabulary tree (VT) is just an "empty" structure of cluster centers and their Voronoi regions" [52], defined at each level by "branching_factor". Although VT is built as a result of hierarchical quantization of a common feature space (the bag-of-features), where all the descriptor vectors calculated for all the training images are thrown in advance, features are not assigned to its tree nodes until the next step, known as an online phase. During this next step, each descriptor vector from the bag-of-features is propagated down the VT structure and compared with every cluster center, one by one. Information about the descriptor vector, and thereby about the image that it comes from, is assigned to the closest cluster – a selected node of the tree in fact. Number of vectors assigned to each cluster builds a weighted relation between tree nodes and each image from the training dataset. Above-mentioned relation forms an "image database" used later for fast image classification or retrieval. To classify a query image using (or retrieve it from) such a database, its descriptor vectors must be first quantized in a similar way to this applied when vocabulary tree is created. Next, an appropriate weighting scheme has to be employed, which, in the case of the SVT approach, is the TF-IDF (Term Frequency – Inverse Document Frequency). With regard to the TF-IDF scheme, visual words that occur frequently in an image but are rare in other images got higher weights. After TF and IDF scores for the images in the database are accumulated, a set of top matches (of matching a query image to the database) is retrieved.

In the SVT-based approach, likewise in the SVM-based one, training images, known as the reference images (RI), are sub-images containing grill parts of cars together with their headlights and indicator lights. The same type of sub-images, known as grill ROIs, are used in the query (testing) phase, when the SURF descriptors are determined for the analyzed image, known in turn as the query image (QI).

Diagram which represents in general both training and testing phases of the presented SVT-based MMR approach is shown in Fig. 5.

One of the most significant advantages of the SVT approach is very compact representation of an image patch in the "image database". It is "simply one or two integers which should be contrasted to the hundreds of bytes or floats used for a descriptor vectors" [52]. This compact representation is also the most important difference between SVT approach and Grauman and Darrell's Pyramid Match algorithm [30], which has been also taken into examination.

The Pyramid Match (PM) is a partial matching approximation carried out between two sets of feature vectors, where feature vectors are for instance vectors of local features (e.g., SURF features) extracted from regions around salient - interest points in an image. PM algorithm uses "a multi-dimensional, multi-resolution histogram pyramid to partition the feature space into

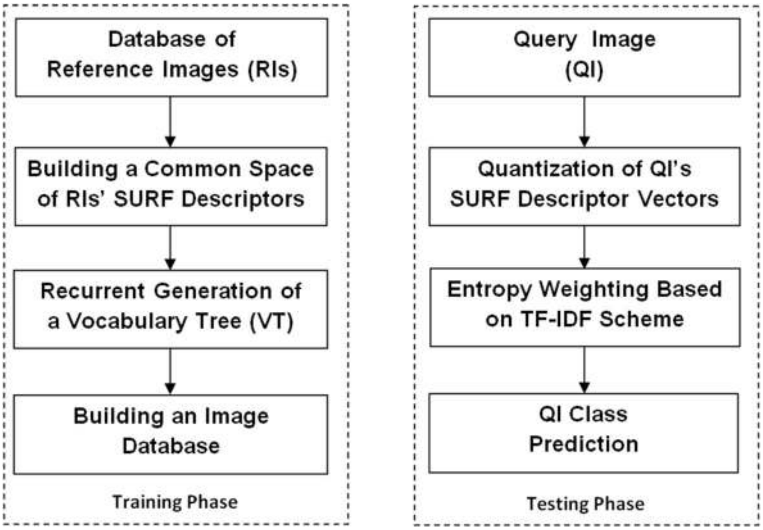

Database of Reference Images (RIs)
Building a Common Space of RIs' SURF Descriptors
Recurrent Generation of a Vocabulary Tree (VT)
Building an Image Database
Training Phase
Query Image (QI)
Quantization of QI's SURF Descriptor Vectors
Entropy Weighting Based on TF-IDF Scheme
QI Class Prediction
Testing Phase

Fig. 5 Workflow of the SVT-based MMR approach

increasingly larger regions" [44]. Partitions in the pyramid, known also as bins, continue to grow in size from very small to larger ones while the successive bins enclose ever-greater subspaces of the entire feature space. Partition being at the top level of the pyramid, engulfs the entire feature space. If any two feature points from any two feature vectors are contained inside the same bin, they are counted as matched. In addition, the size of that bin indicates the farthest possible distance between these two points. Such an approach is in contradiction with clustering method proposed in the SVM-based approach, where distances between any feature points have to be computed. In general, this is also why matching using the PM algorithm is potentially faster than algorithms which compute the distances. The computational time of matching two pyramids in PM algorithm "is linear in the number of features" [44].

Pyramids in LIBPMK Toolkit [47] are created according to two main parameters:

- "finest_side_length", which determines the length of one side of the smallest bin,
- "side_length_factor", which defines the value by which the length of a side of a bin increases from one level to another.

The SVM-based approach (reported in details in [8]) and the MMR schemes organized according to aforementioned PM and SVT techniques have been examined and compared with regard to the classification accuracy as well as durations of their training and testing phases. Overall results of performed examinations are given in Table 1. Analysis of the classification accuracy was performed according to the Overall Success Rate measure (*OSR*), which is defined as follows [75]:

$$OSR = \frac{1}{n}\sum_{i=1}^{k} n_{i,i} \tag{1}$$

where:

n is the number of test images, k is the number of estimated classes (different car models) and $n_{i;i}$ are entries of the main diagonal of a confusion matrix.

Table 1 *OSR* and average durations of the training (D_{TR}) and testing (D_{TE}) phases

	SVM-based[a]	SVT-based[b]	PM-based[c]
	for $k=17$		
D_{TR}^{3} [h]	20	0.5	0.3
D_{TE}^{3} [ms]	38	115	160
OSR	0.91	0.87	0.75
	for $k=45$		
D_{TR}^{3} [h]	512	0.67	0.5
D_{TE}^{3} [ms]	150	123	184
OSR	0.81	0.82	0.57

[a] for number of clusters equal to 3000

[b] for depth = 8 and branching_factor = 300

[c] for finest_side_length = 55 and side_lenght_factor = 42

Results presented in Table 1 have been obtained during experiments performed on a computer system with parameters as follows: DualCore Intel Core i5 650 processor, 4GB of DDR3-1333 RAM. The number of reference images (RI) as well as the number of test (query) images (QI) were the same for all the examined methods and amount to: RI=1360 (for 17 car models) and 3600 (for 45 car models), QI=4865.

Analysis of results presented in Table 1 shows that the SVT-based MMR approach is the most appropriate approach for larger sets of different car models to recognize while the SVM-based one is preferable for smaller sets. It is noticeable also that the PM-based solution has no advantages over the two above-mentioned.

Use of the Scalable Vocabulary Tree algorithm in the case of large number of car models (e.g., 45) allows the MMR module of the presented iCamera system to efficiently distinguish between them. First of all, use of the SVT method in that case gives the OSR rate even higher (0.82) than this possible to achieve in the SVM-based scheme (0.81). The second fact is that the SVT-based approach guarantees the D_{TR} time which is many times shorter than in the case of SVM-based one. For 45 car models for instance, the training phase for the SVT-based approach takes only 40 min, while for the SVM-based one – 512 h, what makes this solution impractical in that case. In fact, the PM-based scheme is even better in this field, but the two remaining parameters (D_{TE} and *OSR*) debase it, in the case of smaller sets of different car models as well.

In the case of smaller sets, relation between the SVT and SVM-based approaches changes to the advantage of the latter. Although the SVT-based scheme still beats the SVM-based one with respect to average duration of the training phase, the D_{TR} time for the SVM-based approach (e.g., for 17 different car models) can be accepted. The most important thing, however, is that the *OSR* rate for the SVM-based scheme is much better than in case of SVT-based one.

In conclusion to the above-considered, we can state that:

- in the case of smaller sets of different car models to predict, the SVM-based approach seems to be the most perspective due to its pretty good *OSR* rate,
- in the case of larger sets, in turn, the SVT-based scheme should be applied, with regard to its better *OSR* rate as well as very short duration of its training phase.

As the final supplement, it is valuable to add that all the results presented and discussed in this section were obtained with the use of SURF implementation patterned on OpenCV v. 2.3.1 [53].

4 License plate recognition

Automatic recognition of license plates is performed using the Tesseract OCR tool [69]. Tesseract is a powerful open source OCR specially designated to read text from various image formats. The ability to use custom-created training sets is a significant advantage of Tesseract. An OCR application can be oriented to a given specific font type thanks to this ability. As reported in [32], the recognition accuracy of Tesseract, when used to digitize antique books, is comparable with the well-known commercial ABBYY Fine Reader package. According to our tests, this accuracy is also comparable with that achieved by the OCR Reader – a part of

Matrox Image Library [39]. Subsequent steps of the algorithm built in the LPR module are illustrated in Fig. 6.

During the first preprocessing step, the license plate ROI taken from the GDE module is converted to a grayscale image, then blurred using the Gaussian filter and finally filtered by applying noise removal morphological operations. After that, a binarization using the Otsu method combined with dilation operation is applied. In the next step, the Canny Edge Detector, followed by the selected contour extraction method, were used to reject a frame surrounding the white license plate area as well as the elements outside this frame. This step allows extraction of an area limited only to the gray license plate numbers on the light background. To extract the numbers properly, the adaptive binarization procedure, with the binarization threshold determined according to the neighborhood of successive pixels, is used. Finally, filters based on factors computed according to contour properties of the extracted objects, are applied to remove elements which differ significantly from license plate numbers they are too wide or too tall.

Selected examples of license plate ROIs and results of the last step mentioned above are shown in Fig. 7.

The results depicted in Fig. 7 show that the accuracy of the LPR algorithm strongly depends on the quality of the input ROI. However, statistical evaluation (taking into account a given number of successive frames with the same license plate), applied after the last OCR step, can significantly increase this accuracy.

The success rate of the LPR algorithm, given as the proportion of correctly recognized license plates among all test images (the test set used in the reported experiments contained 700 images) are as follows:

- with no statistics – 76.43 %,
- with statistics (based on 15 successive images) – 95 %.

5 Color recognition

The color recognition task is performed according to the procedure illustrated in Fig. 8. Inputs to this procedure are the “grill” ROI and the Color Reference Table (CRT). The Color Reference Table is a color palette defined with regard to colors used by car manufacturers

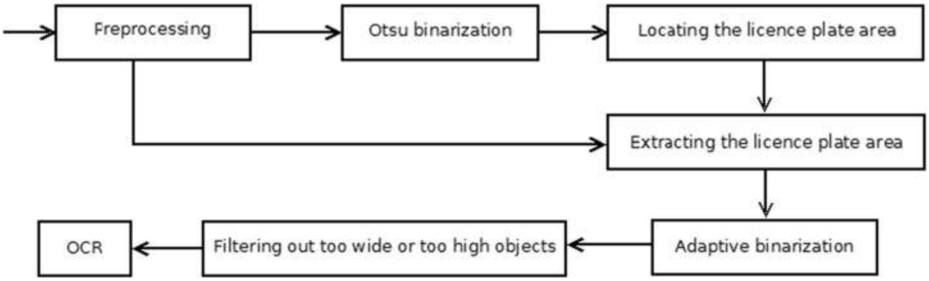

Preprocessing
Otsu binarization
Locating the licence plate area
Extracting the licence plate area
Adaptive binarization
Filtering out too wide or too high objects
OCR

Fig. 6 Workflow of the LPR procedure

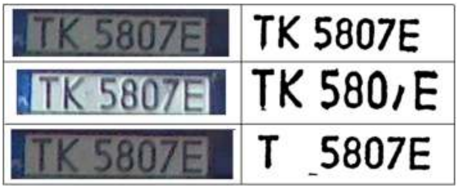


Fig. 7 License plate ROIs (on the *left*) and results of their processing (on the *right*)

(currently as well as in the past) and the human perception of colors. It consists of eight colors described as ranges of RGB values and indexed as follows:

1. Pink – Red,
2. Brown – Orange – Beige,
3. Golden – Olive – Yellow,
4. Green – Lime,
5. Caesious – Blue Navy-blue,
6. Black,
7. White,
8. Gray-Silver.

The idea of vehicle color mapping is present in the literature, as for instance in [24] and [45]. The color recognition approach presented in this paper maps vehicle colors to 8 classes. We have decided to use such an mapping scheme with regard to 2013 Color Popularity Report given by Axalta Coating Systems [7]. According to this report, as well as to the next one for 2014, the colors we have selected refer to the 8 most popular vehicle colors all over the world.

The color recognition algorithm begins with the “White balance” filtering step. The filter applied in this step uses the color of the road surface to modify the color curves, depending on the weather or lighting conditions. Surface images are registered (by the GDE module) every given period of time, when recognition modules are disabled (when no car is present in the camera FOV). The next step changes the color space from RGB to CIELAB according to the requirements of dominant color analysis which is carried out after that. Dominant color

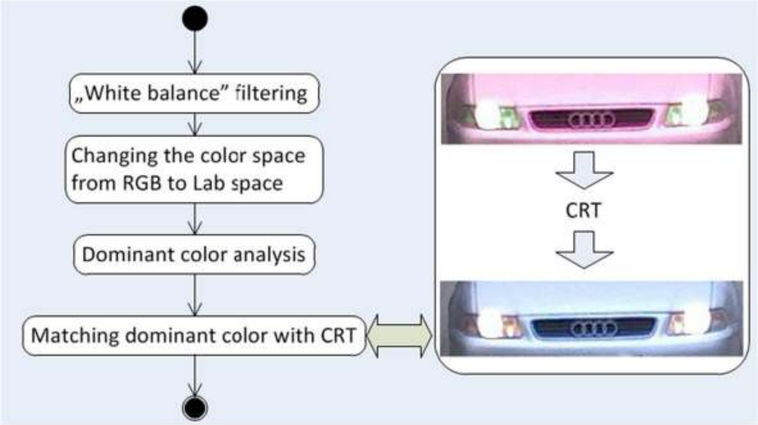

„White balance" filtering
Changing the color space from RGB to Lab space
Dominant color analysis
Matching dominant color with CRT
CRT

Fig. 8 Diagram of the CR algorithm

analysis is performed with regard to Dominant Color Descriptor (DCD) implementation based on MPEG-7.

In the final step, the dominant color in the analyzed ROI is converted back from CIELAB to RGB space and referred to colors from CRT. The result of this reference is returned as the name of the predicted color.

Success rates relating to individual colors from the CRT table obtained with or without the use of the white balance filter are illustrated in Fig. 9.

Figure 9 shows that the success rate of the CR module differs according to the color in the CRT table. The highest rate is obtained in the case of the Pink–Red color range, while the lowest is in Gray color range. Figure 9 also shows that results are slightly better when the white balance filter except in the case of the Pink–Red color range.

6 System efficiency

As mentioned in Section 2, the efficiency of the iCamera system depends mainly on the performance parameters of the applied CPU architecture. To analyze this dependency as well as to verify system assumptions and requirements, the following x86 platforms have been selected:

- Intel i5: CPU - Dual Core Intel Core i5 650, 3200 MHz, RAM - 4GB DDR3-1333 DDR3, system - Windows Server 2008 R2 Enterprise (64-bit),
- ATOM N270: CPU - ATOM N270, 1,6 GHz, RAM - 1GB DDR2 SDRAM 533.0 MHz, system - Linux Debian 3.2.0-4-686-pae,
- AMD Zacate: CPU - Dual-Core AMD Zacate E350/350D APU, 800 MHz, RAM - 4GB DDR3, system - Linux Ubuntu 13.04-desktop-amd64.

At the moment, the iCamera system implements serial computation. Taking this into account, the total processing time of a single frame is the sum of the times required to decode the video stream, create the frame object, detect and extract two types of ROIs and perform recognition tasks by the MMR, LPR and CR modules. There are, of course, other processes,

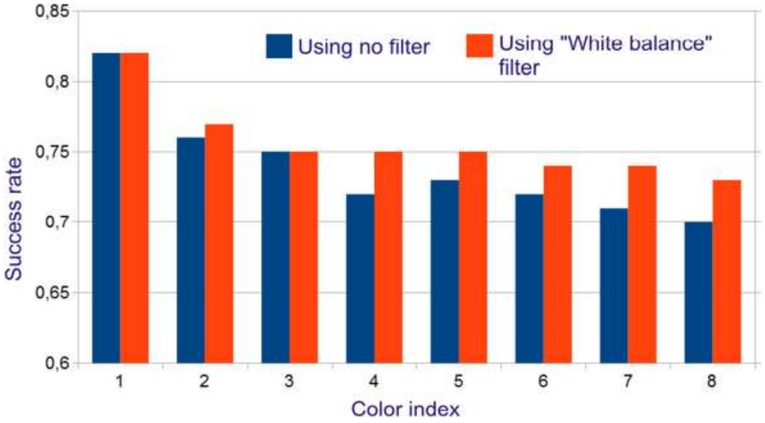


Fig. 9 Accuracy of the CR module

such as those related to statistical evaluation as well as many others connected with internal communication and video stream servicing.

While the duration of recognition tasks varies depending on the analyzed content, the time consumed by the remaining processes is constant and hinges only on hardware performance. There is, however, one exception to the above rule. Because of the stable and very small size of the license plate ROI (about 214×44 pixels) [76] and the recurrent nature of its content, the duration of the LPR task (T_{LPR}) varies very slightly. In accordance with performed tests we can assume that T_{LPR}, regardless of the platform used, is not larger than 20 ms. Similarly, we can take for granted that the remaining processes (T_{RP}), except MMR and CR tasks, take no more than 15 ms.

In the case of the MMR module, the times required to process the single QI image and return prediction about make and model of the analyzed vehicles on examined platforms (T_{MMR}), for 17 different car models, are illustrated in Fig. 10.

Times needed to complete the task by the CR module (T_{CR}) are portrayed in turn in Fig. 11.

Charts presented in Figs. 10 and 11 lead to the simple conclusion that the parameters of the Intel i5 architecture give better performance than other selected platforms. However, these charts also allow us to evaluate which frame rate of the IP camera would be the most appropriate.

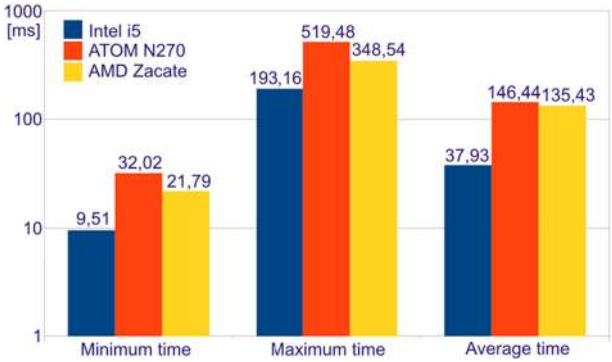
Taking into consideration the times reported earlier in this section, the average duration of processing the single QI image in the iCamera system (T_{QI}) is as follows:

$$T_{QI} = T_{RP} + T_{MMR} + T_{LPR} + T_{CR} \approx 90 \text{ ms.} \tag{2}$$

This means that the iCamera system is capable, when implemented on the Intel i5 platform or similar, to serially compute 11 frames of resolution 4CIF. Respectively, the frame rate of the IP camera can be 10 or 11 fps. This meets our assumptions about statistical evaluation, because to increase the accuracy we need to predict by analyzing at least 10 frames.

Results depicted above give the summary of the iCamera system time constraints. Considerations presented in Sections 3, 4 and 5 provide data on success rates of the MMR, LPR and CR modules, respectively. To show in turn how these success rates and processing times, related to each of the modules mentioned above, compare with other systems briefly discussed in the Literature Review, tables presented below have been drawn up. Commercial systems have been excluded from this comparison due to the fact that their technological bases are strictly confidential in general. According to the fact that in majority of selected papers the

Fig. 10 Times required to complete the MMR task



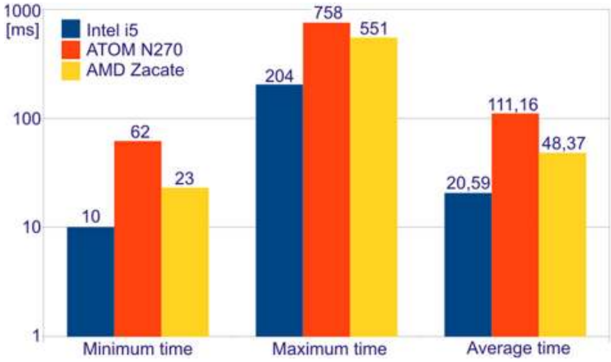


Fig. 11 Times needed to complete the CR task

only reported rates were success rates, we have also resigned from depicting the false positive and false negative rates.

As it is depicted in Table 2, the iCamera's MMR system is pretty effective in comparison to other ones. There are however approaches where reported success rate is higher than in case of the iCamera system, but either number of predicted models is much less, as in [43] and [60] or other conditions, e.g., in which test images were acquired, make the classification task easier, as in [57]. Compared to the architecture presented in this paper, the MMR system described in [34] reports a better average success rate which is over 98 %. It is worth to notice however that the time required to complete the MMR task is for this approach about 10 ms longer than for the iCamera system in the case of 17 different car models.

Table 3 shows that the iCamera's LPR system, if applied with statistics, overtakes other approaches taken into the comparison with regard to both average success rate and processing time. The only approach which is comparable to the iCamera system is the one reported in [74].

Table 2 Overall performance of the MMR systems in the literature and in the iCamera (iC) system

Ref.	Average Success Rate [%]	Average Processing Time [s]	Architecture	Comments
[57]	96.00[1,2]	not reported	Intel Core 2 Duo E6600 PC with 2GB of RAM	[1]for 74 different car models [2]small pose variations and small distances from the car
[43]	92.00[3,9] 97.00[4,9] 100.00[5,9]	not reported		[3]using Wavelet [4]using FFT [5]using Curvelet [9]for 5 different car models
[60]	81.00[6,9] 92.00[7,9] 99.00[8,9]	not reported		[6]using SVM (one versus all) [7]using KNN method [8]using SVM (one versus one)
[21]	89.50[10]	not reported		[10]SIFT matching - for 38 different car models
[34]	98.43[11]	0.048[11]	Intel Core I5 3337U 1.8 GHz and 4-G RAM	[11]for 29 different car models
iC	91.60[12] 81.00[13]	0.038[12] 0.123[13]	Intel Core i5 650, 4GB of DDR3-1333 RAM,	for 17[12] (45[13]) car models and SVM[12] (SVT[13])-based approach

Table 3 Overall performance of the LPR systems in the literature and in the iCamera (iC) system

Ref.	Average Success Rate [%]	Average Processing Time [s]	Processor and Platform	Comments
[16]	93.70(98.8)[1]	2.4	Pentium IV-1.6 GHz PC	[1]for short distances (<= 3 m)
[2]	85.54[2]	0.276	Pentium IV at 3.0 GHz with 512 MB RAM	[2]overall, for all sample sets
[74]	93.54	0.284	Intel Core 1.8G, 1.5G RAM, Microsoft Windows XP	
[54]	91.00[3]	0.100	Intel ® Core ™ 2 Duo Processor CPU P8400 (2.26GHz, 2267 MHz)	[3]for small dataset consisted of 260 photos of cars
[83]	93.10	not reported		
[77]	65.97[4]	not reported		[4]for very small dataset (130 samples)
iC	95.00[5] 76.43[6]	0.020	e.g., Dual-Core AMD Zacate E350/350D APU, 800 MHz, RAM - 4GB DDR3, system - Linux Ubuntu 13.04-desktop-amd64	[5]with statistics [6]without statistics

Data depicted in Table 4 show that approach proposed in [31] is superior to all other methods. The success rate for this "tri-state" approach and for dataset which includes car images taken from real traffic streams (to simulate the ITS scenario) is 97 %. Success rates for other approaches range in general from 82 to 88 %. Our algorithm overtakes only the one reported in [76].

However, all the CR systems compared in Table 4 with the iCamera's one classify the color of the vehicle into one of seven colors. Our approach differs from them significantly because it classifies the color of the vehicle into one of eight colors. This makes the classification task for our approach harder and probably it is one of the reasons which decreases its average success

Table 4 Overall performance of the CR systems in the literature and in the iCamera (iC) system

Ref.	Average Success Rate [%]	Average Processing Time [s]	Processor and Platform	Comments
[31] ITS	91.00[1] 94.00[2] 97.00[3]	0.015[1] 0.021[2] 0.057[3]	AMD Athlon II X2240,2.0GB RAM, 2.81 GHz PC	[1]homo SVM method, [2]removal rule method, [3]tri-state method
[24]	83.30[4] 82.20[5] 83.92[6]	not reported		[4]*k*NN classifier [5]ANN model [6]SVM method 7 colors; ca. 2000 dataset
[45]	88.34[7] 87.83[8]	not reported		[7]with city-block distance [8]with Euclidean distance 7 colors; 350 dataset.
[76]	72.30	not reported		
iC	75.63[9] 73.86[10]	0,020	Intel Core i5 650, 4GB of DDR3-1333 RAM, 64-bit Windows Server	[9]with White balance filter [10]with no filter

rate. Anyway, according to our CR system, we need to admit that there is a space to improve it. So far, its major advantage is pretty short processing time.

Results presented in Tables 2, 3 and 4 and referred to the iCamera system have been obtained using the same test dataset which consisted of 3000 images taken by the network camera positioned over crossroads at the university campus. These test images have been collected in various lighting conditions over a period of 2 years. A sample test image is presented in Fig. 4.

Another dataset has been gathered for training purposes of the MMR module. According to the fact that in our MMR system each model is represented by 80 images (as it was briefly mentioned in Section 3), this training dataset contained a total number of 3600 images referred to 45 different models of cars. Its sub-set, consisted of 1360 images, refers in turn to 17 models, listed exactly in [8]. All the training images were taken outdoor or downloaded from the Internet (50-50) and represented front sides of cars taken “en face” or at a little angle (less than 30°).

7 Conclusions and future work

In summary, the novel contributions of this paper are as follows:

- analysis and comparison of three different state-of-the-art approaches (SVM, SVT and PM-based ones) according to their effectiveness in real-time MMR applications with regard to the number of different car models,
- novel vehicle color recognition algorithm based on dominant color analysis, performed in the lab space, and mapping scheme applied with regard to the 8 most popular vehicle colors all over the world,
- very efficient LPR approach taking successfully the advantage of statistical evaluation,
- detailed comparison of applied MMR, LPR and CR schemes with other relevant solutions,
- comprehensive the iCamera system capable to efficiently, and in real time, identify license plate numbers, recognize selected makes and models of cars and classify real colors of cars into eight predefined categories.

The current prototype implementation of the iCamera system presented in this paper are is suitable for a wide range of traffic monitoring applications. As shown in Section 2, assumptions made about camera settings were intended to monitor every individual traffic lane of city center streets, countryside roads, highways, parking lots, etc. The goal of traffic monitoring performed by the iCamera system is first of all to create opportunities for identifying offenders in traffic accidents, especially in cases where the offender has fled the scene. Relying on evidence given by witnesses of such accidents, authorized services (e.g., municipal ones) can use the material recorded by the multiple iCamera systems distributed over main crossroads in the city to look for the car (a black Ford Mondeo for instance) which, according to the time of event as well as distance from the scene of the accident, is likely to be responsible. Cars selected this way can then be verified according to their license plate numbers returned by the system.

To make the above capabilities useful, the iCamera system must ensure an adequate level of efficiency. The results presented in Section 6 as well as the success rate factors reported in accordance to accuracies of the MMR, LPR and CR modules confirm the iCamera system’s utilities in this kind of surveillance applications.

There are however some ways to increase this efficiency.

According to performance parameters, the increase can be obtained in two ways:

- by substituting serial computing with parallel computing;
- by applying GPU-accelerated computing instead of CPU only.

Our experiments show that using a mixed CPU/GPU architecture combined with OpenCL (Open Computing Language) implementations can increase system performance by more than 5-fold. Moreover, it is reasonable to assume that parallel computing will also be able to accelerate the system at least twice.

According to the accuracies of the recognition modules, the system's efficiency can be improved first of all by increasing the number of frames taken into account by the statistical evaluation procedure. To achieve this, the iCamera system has to be able to process more than 10 fps as is the case at the moment. This aspect is however, strongly connected to performance parameters. We hope that proper implementation of both the technologies listed above will allow the frame rate of the applied cameras to be increased at least 25 fps, and in this way improve significantly the system's efficiency as a whole.

Acknowledgments This work was supported by the European Regional Development Fund under the Innovative Economy Operational Programme, INSIGMA project no. POIG.01.01.02- 00-062/09. We want also to address our special thanks to our colleague Mariusz Rychlik from The University of Computer Engineering and Telecommunications in Kielce (Poland) for his valuable contributions to this work.

References

1. Agarwal S, Snavely N, Simon I, Seitz SM, Szeliski R, Ieee (2009) Building Rome in a day. In: 12th IEEE International Conference on Computer Vision, Kyoto, JAPAN, 2009. Sep 29-Oct 02 2009. IEEE International Conference on Computer Vision. pp 72–79. doi:10.1109/iccv.2009.5459148
2. Anagnostopoulos CNE, Anagnostopoulos IE, Loumos V, Kayafas E (2006) A license plate-recognition algorithm for intelligent transportation system applications. IEEE Trans Intell Transp Syst 7(3):377–392. doi:10.1109/tits.2006.880641
3. Anthony D (2005) More local structure information for make-model recognition. Dept. of Computer Science, University of California at San Diego
4. Arulmozhi K, Perumal AS, Priyadarsini TCS, Nallaperumal K, IEEE (2012) image refinement using skew angle detection and correction for indian license plates. 2012 I.E. Int Conf Comput Intell Comput Res (Iccic):718–721
5. Arulmozhi K, Perumal AS, Sanooj P, Nallaperumal K, IEEE (2012) Application of top hat transform technique on indian license plate image localization. 2012 I.E. International Conference on Computational Intelligence and Computing Research (Iccic):708–711
6. Ashtari AH, Nordin MJ, Fathy M (2014) An Iranian license plate recognition system based on color features. IEEE Trans Intell Transp Syst 15(4):1690–1705. doi:10.1109/tits.2014.2304515
7. Axalta. http://www.axaltacs.com/content/dam/AP/Axalta/Korea/Axalta%202013%20Global%20Color%20Popularity%20Report.pdf. Viewed 15 July 2015
8. Baran R, Glowacz A, Matiolanski A (2013) The efficient real-and non-real-time make and model recognition of cars. Multimedia Tools and Applications. doi:10.1007/s11042-013-1545-2
9. Baran R, Wiraszka D, Dziech W (2000) Scalar quantization in the pwl transform spectrum domain. In: Proc. of the int. conf. of Mathematical Methods in Electromagnetic Theory, MMET2000, pp 218–221,
10. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-Up Robust Features (SURF). Comput Vis Image Underst 110(3):346–359. doi:10.1016/j.cviu.2007.09.014

11. Belongie. http://vision.ucsd.edu/belongie-grp/research/carRec/car_rec.html. Viewed 15 July 2015
12. Bulan O, Bernal EA, Loce RP (2013) Efficient processing of transportation surveillance videos in the compressed domain. J Electron Imaging 22 (4). doi:10.1117/1.jei.22.4.041116
13. Bulan O, Loce RP, Wu W, Wang Y, Bernal EA, Fan Z (2013) Video-based real-time on-street parking occupancy detection system. J Electron Imaging 22 (4). doi:10.1117/1.jei.22.4.041109
14. Caliendo C, Guida M (2012) Microsimulation approach for predicting crashes at unsignalized intersections using traffic conflicts. J Transp Eng-Asce 138(12):1453–1467. doi:10.1061/(asce)te.1943-5436.0000473
15. Candes E, Demanet L, Donoho D, Ying L (2006) Fast discrete curvelet transforms. Multiscale Model Simul 5(3):861–899. doi:10.1137/05064182x
16. Chang SL, Chen LS, Chung YC, Chen SW (2004) Automatic license plate recognition. IEEE Trans Intell Transp Syst 5(1):42–53. doi:10.1109/tits.2004.825086
17. Chen LH, Hung YL, Su CW (2013) integration of keypoints and edges for image retrieval. Int J Pattern Recognit Artif Intell 27 (8). doi:10.1142/s0218001413550136
18. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297. doi:10.1023/a:1022627411411
19. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. Ieee Trans Inf Theory 13(1):21–+. doi:10.1109/tit.1967.1053964
20. Csurka G. DCR, Fan L., Willamowski J., Bray C. (2004) Visual categorization with bags of keypoints. Paper presented at the Workshop on Statistical Learning in Computer Vision, ECCV
21. Dlagnekov L (2005) Video-based car surveillance: License plate, make, and model recognition. Master's thesis. University of California, San Diego
22. Do MN, Vetterli M (2005) The contourlet transform: an efficient directional multiresolution image representation. Ieee Trans Image Process 14(12):2091–2106. doi:10.1109/tip.2005.859376
23. Du S, Ibrahim M, Shehata M, Badawy W (2013) Automatic License Plate Recognition (ALPR): a state-of-the-art review. IEEE Trans Circ Syst Video Technol 23(2):322–336. doi:10.1109/tcsvt.2012.2203741
24. Dule E, Gokmen M, Beratoglu MS (2010) A convenient feature vector construction for vehicle color recognition. recent advances in neural networks, Fuzzy Systems & Evolutionary Computing
25. Dziech W, Baran R, Wiraszka D (2000) Signal compression based on zonal selection methods. In: Proc. of the int. conf. of Mathematical Methods in Electromagnetic Theory, MMET2000, pp 224–226,
26. Fathy M, Siyal MY (1995) An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis. Pattern Recogn Lett 16(12):1321–1330. doi:10.1016/0167-8655(95)00081-x
27. Ghazal M, Hajjdiab H, IEEE (2013) license plate automatic detection and recognition using level sets and neural networks. 2013 First International Conference on Communications Signal Processing, and Their Applications (Iccspa'13)
28. Gorzalczany MB, Rudzinski F (2006) Cluster analysis via dynamic self-organizing neural networks. In: Rutkowski L, Tadeusiewicz R, Zadeh LA, Zurada J (eds) Artificial Intelligence and Soft Computing - Icaisc 2006, Proceedings, vol 4029. Lecture Notes in Computer Science, pp 593–602
29. Gorzalczany MB, Rudzinski F (2008) WWW-newsgroup-document clustering by means of dynamic self-organizing neural networks. In: Rutkowski L, Tadeusiewicz R, Zadeh LA, Zurada JM (eds) Artificial Intelligence and Soft Computing - Icaisc 2008, Proceedings, vol 5097. Lecture Notes in Artificial Intelligence. pp 40–51
30. Grauman K, Darrell T (2007) The pyramid match kernel: efficient learning with sets of features. J Mach Learn Res 8:725–760
31. Gu H-Z, Lee S-Y (2013) A view-invariant and anti-reflection algorithm for car body extraction and color classification. Multimedia Tools and Applications 65:387–418. doi:10.1007/s11042-012-0996-1
32. Heliński M. KM, Parkoła T (2012) Report on the comparison of tesseract and Abbyy Finereader OCR Engines Paper presented at the PCSS, Poznan
33. Hong WJ, Kim MW, Oh I-S (2013) Learning-based detection of license plate using SIFT and neural network. Inst Electron Eng Korea 50(8):187–195. doi:10.5573/ieek.2013.50.8.187
34. Hsieh J-W, Chen L-C, Chen D-Y (2014) Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition. Ieee Transactions on Intelligent Transportation Systems 15(1):6–20. doi:10.1109/tits.2013.2294646
35. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. Ieee Trans Neural Netw 13(2):415–425
36. Hu W, Yang J, Bai L, Yao L (2013) A new approach for vehicle color recognition based on specular-free image. In: 6th International Conference on Machine Vision (ICMV), London, 2013, Nov 16–17 2013. Proceedings of SPIE. doi:90670a10.1117/12.2051976
37. Huang Y-P, Chen C-H, Chang Y-T, Sandnes FE (2009) An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition. Expert Syst Appl 36(5):9260–9267. doi:10.1016/j.eswa.2008.12.006

38. Insigma. http://kt.agh.edu.pl/en/projekt/294. Viewed 15 July 2015
39. Janowski L, Kozlowski P, Baran R, Romaniak P, Glowacz A, Rusc T (2014) Quality assessment for a visual and automatic license plate recognition. Multimed Tools Appl 68(1):23–40. doi:10.1007/s11042-012-1199-5
40. Jiao J, Ye Q, Huang Q (2009) A configurable method for multi-style license plate recognition. Pattern Recogn 42(3):358–369. doi:10.1016/j.patcog.2008.08.016
41. Jin L, Xian H, Bie J, Sun Y, Hou H, Niu Q (2012) License plate recognition algorithm for passenger cars in Chinese residential areas. Sensors 12(6):8355–8370. doi:10.3390/s120608355
42. Ju Z, Wang P (2012) License plate image skew correction algorithm based on geometric constraint 1–4
43. Kazemi FM, Samadi S, Poorreza HR, Akbarzadeh-T M-R (2007) Vehicle recognition based on fourier, wavelet and curvelet transforms - a comparative study. International Conference on Information Technology, Proceedings, pp 939–940
44. KG (2006) Matching sets of features for efficient retrieval and recognition. MIT
45. Kim K-J, Park S-M, Choi Y-J, Ieee Computer SOC (2008) Deciding the number of color histogram bins for vehicle color recognition. 2008 Ieee Asia-Pacific Services Computing Conference, Vols 1–3, Proceedings. doi:10.1109/apscc.2008.207
46. lex.europa.eu (2010) DIRECTIVE 2010/40/EU
47. libpmk. http://people.csail.mit.edu/jjl/libpmk/. Viewed 15 July 2015
48. Loce RP, Bernal EA, Wu WC, Bala R (2013) Computer vision in roadway transportation systems: a survey. J Electron Imaging 22(4):24. doi:10.1117/1.jei.22.4.041121
49. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. doi:10.1023/b:visi.0000029664.99615.94
50. Negri P, Clady X, Milgram M, Poulenard R (2006) An oriented-contour point based voting algorithm for vehicle type classification. In: 18th International Conference on Pattern Recognition (ICPR 2006), Hong Kong, pp 574–577
51. Neurocar. http://en.neurosoft.pl/products/neurocar/. Viewed 15 July 2015
52. Nister D, Stewenius H (2006) Scalable recognition with a vocabulary tree. In: Conference on Computer Vision and Pattern Recognition, New York, 2161{2168
53. OpenCV. http://opencv.org/documentation.html. Viewed 15 July 2015
54. Ozturk F, Ozen F (2012) A New license plate recognition system based on probabilistic neural networks. First World Conf Innov Comput Sci (Insode 2011) 1:124–128. doi:10.1016/j.protcy.2012.02.024
55. Pan J, Yuan Z (2008) Research on license plate detection based on wavelet. Adv Intell Comput Theor Appl Proc: Asp Contemp Intell Comput Tech 15:440–446
56. Park S-M, Kim K-J (2008) PCA-SVM based vehicle color recognition. Korea Information Processing Society Transactions on Software and Data Engineering (PartB) 15(4):285–292. doi:10.3745/KIPSTB.2008.15-B.4.285
57. Pearce G, Pears, N. (2011) Automatic make and model recognition from frontal images of cars. Paper presented at the 8th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS)
58. Petrovic VS, Cootes TF (2004) Vehicle type recognition with match refinement. In: 17th International Conference on Pattern Recognition (ICPR), British Machine Vis Assoc, Cambridge, 2004, pp 95–98. doi:10.1109/icpr.2004.1334477
59. Psyllos A, Anagnostopoulos CN, Kayafas E (2011) Vehicle model recognition from frontal view image measurements. Comput Stand Interfaces 33(2):142–151. doi:10.1016/j.csi.2010.06.005
60. Rahati S, Moravejian R, Kazemi EM, Kazemi FM (2008) Vehicle recognition using contourlet transform and SVM. Proceedings of the Fifth International Conference on Information Technology: New Generations:894–898. doi:10.1109/itng.2008.136
61. Ren C, Song J (2013) A character segmentation method based on character structural features and projection. Fifth International Conference on Digital Image Processing (Icdip 2013) 8878. doi:10.1117/12.2030579
62. SaLsA. http://www.salsa-autonomik.de/. Viewed 15 July 2015
63. Sanap PR, Narote SP (2010) License plate recognition system-survey. In: International Conference on Methods and Models in Science and Technology, Chandrigarh, INDIA, 2010. Dec 25–26 2010. AIP Conference Proceedings. pp 255–260
64. Shapiro V, Gluhchev G, Dimov D (2006) Towards a multinational car license plate recognition system. Mach Vis Appl 17(3):173–183. doi:10.1007/s00138-006-0023-5
65. Shi YLS (2006) Smart cameras: a review. CCTV Focus
66. Spring. http://springsource.io. Viewed 15 July 2015
67. Stahlschmidt C, Gavriilidis A, Velten J, Kummert A (2013) People detection and tracking from a Top-view position using a time-of-flight camera. Multimed Commun Serv Secur, Mcss 368:213–223
68. Szeto WY (2014) Dynamic modeling for intelligent transportation system applications. J Intell Transp Syst 18(4):323–326. doi:10.1080/15472450.2013.834770
69. Tesseract. http://tesseract-ocr.repairfaq.org/. Viewed 15 July 2015

70. Vishwanath N, Somasundaram S, Ravi MRR, Nallaperumal NK, IEEE (2012) Connected component analysis for indian license plate infra-red and color image character segmentation. 2012 I.E. International Conference on Computational Intelligence and Computing Research (Iccic):743–746
71. VocabTree2. https://github.com/snavely/VocabTree2. Viewed 15 July 2015
72. Wang Y-R, Lin W-H, Horng S-J (2009) Fast license plate localization using discrete wavelet transform. Algoritm Architectures Parallel Process Proc 5574:408–415
73. Wang A, Liu X, IEEE (2012) Vehicle license plate location based on improved roberts operator and mathematical morphology. Proceedings of the 2012 Second International Conference on Instrumentation & Measurement, Computer, Communication and Control (Imccc 2012):995–998. doi:10.1109/imccc.2012.237
74. Wen Y, Lu Y, Yan J, Zhou Z, von Deneen KM, Shi P (2011) An algorithm for license plate recognition applied to intelligent transportation system. IEEE Trans Intell Transp Syst 12(3):830–845. doi:10.1109/tits.2011.2114346
75. Witten IH, Frank E (2005) Data mining: practical machine learning tools and tech-niques, 2nd edn. Morgan Kaufmann, San Francisco
76. Wu Y-T, Kao J-H, Shih M-Y (2010) A vehicle color classification method for video surveillance system concerning model-based background subtraction. Advances in Multimedia Information Processing-Pcm Pt I 6297:369–380
77. Yang J, Hu B, Yu J, An J, Xiong G, IEEE (2013) a license plate recognition system based on machine vision. 2013 I.E. International Conference on Service Operations and Logistics, and Informatics
78. Yoon Y, Ban K-D, Yoon H, Kim J, IEEE (2012) Blob detection and filtering for character segmentation of license plates. 2012 I.E. 14th International Workshop on Multimedia Signal Processing (Mmsp):349–353
79. Yoon Y, Ban K-D, Yoon H, Lee J, Kim J (2013) Best combination of binarization methods for license plate character segmentation. Etri J 35(3):491–500. doi:10.4218/etrij.13.0112.0545
80. Zafar I, Acar BS, Edirisinghe EA (2007) Vehicle make & model identification using scale invariant transforms. Proceedings of the Seventh IASTED International Conference on Visualization, Imaging, and Image Processing:271–276
81. Zhai X, Bensaali F, Sotudeh R (2013) Real-time optical character recognition on field programmable gate array for automatic number plate recognition system. Iet Circ Devices Syst 7(6):337–344. doi:10.1049/iet-cds.2012.0339
82. Zhai B-F, Xu E, Wang Q, Li Y (2011) Characters edge detection method of vehicle license tag based on canny operator. 2011 Second International Conference on Education and Sports Education
83. Zhang L, Shi X, Xia Y, Mao K (2013) A multi-filter based license plate localization and recognition framework. 2013 Ninth International Conference on Natural Computation (Icnc):702–707
84. Zhang Y, Zha Z, Bai L (2013) A license plate character segmentation method based on character contour and template matching. Meas Technol Eng Res Ind 1–3(333–335):974–979. doi:10.4028/www.scientific.net/AMM.333-335.974
85. Ziolko M, Sypka P, Dziech A, Baran R, Peric N, Petrovic I, Butkovic Z (2005) Contour transmultiplexing. ISIE 2005: Proc IEEE Int Symp Ind Electron 1–4:1167–1170

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 Ph.D. 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.

Tomasz Rusc received the M. Sc. degree in Telecommunications from the Faculty of Electrical and Control Engineering, Kielce University of Technology in 1998 and the Ph.D. degree in Institute for Circuit and System Theory from the Faculty of Engineering, Christian-Albrechts-University in Kiel in 2006. The main areas of his scientific interest are digital image processing including filter bank design, and semantic modelling and searching of multimedia content.

Pawel Fornalski received his M. Sc. degree in Informatics from the Faculty of Electrical Engineering, Automatic Control and Computer Science, Kielce University of Technology in 2004. He actively participates (or has participated) in EU and national projects, e.g., in INDECT, INSIGMA, OASIS Archive, InWAS. His fields of scientific and professional interest are related mainly to search engines and big data processing applications as well as digital image processing including especially such computer vision tasks as object detection and recognition.