Multiclass Vehicle Type Recognition System using an Oriented Points Set based Model

Pablo Negri[1], Xavier Clady[1], Maurice Milgram[1], Raphael Poulenard[2] and Lionel Prevost[1]
[1] Institut Systèmes Intelligents et Robotique
Groupe Perception et Réseaux Connexionnistes
Université Pierre et Marie Curie-Paris 6
pablo.negri@lisif.jussieu.fr
[2] LPREditor, Montpellier, France

***Abstract*— This paper presents a framework for multi-class vehicle type identification based on oriented contour points. The decision process combines four classifiers: three voting algorithms and a distance error. This method have been tested on a realistic data set obtaining similar results for equivalent recognition frameworks with different features selections [11]. A confidence criterion is also calculated to validate the system results. The system also show to be robust to partial occlusions.**

***Keywords*— Pattern Recognition, Multiclass Classification, Oriented Contour Poins, Voting Algorithm.**

I. Introduction

Many vision based Intelligent Transport Systems are dedicated to detect, track or recognize vehicles in image sequences. Three main applications can be distinguished. Firstly, embedded cameras allow to detect obstacles and to compute their distances to the equiped vehicle [12]. Secondly, road monitoring measures traffic flow [2], notifies the health services in case of an accident or informes the police in case of a driving fault. Finally, Vehicle based access control systems for buildings or outdoor sites have to authentify incoming (or outgoing) cars [11]. The first application has to classify region-of-interest (ROI) in two classes: vehicles or background. Vehicles are localized in an image with 2D or 3D bounding box [12], [9]. The second one can use geometric models to classify vehicles in some categories such sedans, minivans or SUV. These 2D or 3D geometric models are defined by deformable or parametric vehicle templates [7], [4], [5].

The third application often uses only the recognition of a small part of the vehicle: the license plate. It is enough to identify a vehicle but, in practice, licence plate recognizer can provide wrong information due to poor image quality or fake plate. Combining such a system with other processes dedicated to identify the vehicle type (brand and model) can increase the authentication accuracy and robustness. This paper addresses the vehicle type identification problem from a vehicle greyscale frontal image.

As far as we know, only one paper deals with a similar problem. In a recent recognition framework for rigid object recognition, Petrovic and Cootes [11] test various features for vehicle type classification. Their decision module is based on two distance measures (with or without Principal Component Analysis feature selection): the dot product $d = 1 - f_1 f_2$ and the Euclidean measure $d = |f_1 - f_2|$, where f_i are the feature vectors. The dot product gives slighthly outperforming results. Best results are obtained with gradient-based representations. These results can be explained because the vehicle rigid structure is standardized by the manufacturer for each model. The relevant information contained in contour edge and orientation is independent of the vehicle color. Others works [8], [10], [1] had taken the edge orientations for the recognition of different patterns like faces or small targets with irregular borders.

Fig. 1. Realistic vehicle images with tollgate presence.

In this paper, a multi-class recognition system is developed using the oriented-contour pixels to represent each vehicle class. From a vehicle frontal view, the system identifies the instance as the most similar model class in the data base. The classification is based on a voting process and a Euclidean edge distance. The algorithm has to deal with partial occlusions as tollgates can hide part of the vehicle (see fig. 1), making quite inadequate appearance-based methods. In spite of tollgate presence, our system does not have to change its parameters or apply time-consuming reconstruction process.

In section II, we explain how we define a model for each class in the data base using oriented-contour points. Section III employs this model to obtain a *score* measuring the similarity between the input instance and the database classes. A Bayes confidence criterion is then proposed in order to reject classifier response or to combine it with a license plate recognition system. Experimental results are presented in section IV. Last section is devoted to conclusions and prospects.

II. Model Creation

During the training stage, we produce a model for all the K vehicle type classes composing the system knowledge. The list

of classes the system is able to recognize is called Knowledge Base (**KnB**). In our system, the Knowledge Base will be the **20** vehicle type classes of the Confusion Matrix.

A. Images Databases

All ours experiments have been carried out on the Training Base (**TrB**) and on the Test Base (**TsB**). The **TrB** samples are used to produce the oriented-contour point models of the vehicle classes. The **TsB** samples are utilized to evaluate the accuracy of the classification system. The **TrB** is composed of high quality frontal vehicle images captured in different car parks. On the other hand, the **TsB** is made out of outdoor nearly frontal vehicle images under different light conditions and at a lower resolution. In figure 2, the upper row shows samples from **TrB** and in the bottom row, the figure shows the corresponding vehicle type class of the **TsB**.

Fig. 2. samples from the training database **TrB** (upper row) and the corresponding vehicle type class in the test database **TsB**.

B. Confusion matrix

Our classification system will be applied to a data base (see table 1). Formally, we select a finite set $\mathcal{K}$ of $K = 20$ classes. For the multiclass classification problem, each example $t \in \mathcal{T}$ (**TsB**) is assigned a single class $k \in \mathcal{K}$, so that labelled examples are pairs (t,k). The system goal is to find a function $G : \mathcal{T} \rightarrow \mathcal{K}$ which matches a unknonw example (t,k) minimizing the probability that $k \neq G(t)$.

Model	Citroen Berlingo A	Citroen C3	Citroen Picasso B	Citroen Saxo B	Ford Focus A	Peugeot 206 B	Peugeot 307	Peugeot 405	Peugeot 406 B	Renault 19 B	Renault Clio A	Renault Clio C	Renault Clio D	Renault Laguna B	Renault Scenic B	Renault Scenic C	Renault Twingo A	Renault Twingo B	VW Golf C	VW Golf D	**Total**
TrB	1	1	4	9	5	10	12	3	6	4	7	15	32	5	5	5	19	21	4	5	173
TsB	11	20	19	21	13	21	28	17	23	20	21	37	62	22	20	14	33	33	22	23	480

Table 1. The database sets.

C. Prototype image

We create a canonical rear-viewed vehicle image I from the four corner points of the license plate {**A**,**B**,**C**,**D**} (see fig. 3). The image templates are called prototypes and in the present work are 600 * 252 pixels. A ROI defined by the points {**A**,**B**,**C**,**D**} is independent of the vehicle location in the image and the scale (fig.3.a). In order to correct the orientation of the original image (see example in fig.2), an affine transformation moves original points {**A**,**B**,**C**,**D**} to the desired {**A'**,**B'**,**C'**,**D'**} reference position, considering the vehicle grid and the license plate in the same plane. A license plate recognition system provides the corners of the vehicle license plate.

(a) (b)

Fig. 3. (a) original image, (b) prototype I.

The Sobel operator is used to compute the gradient's magnitude and orientation of the greyscale prototype I ($|\nabla g_I|$,ϕ_I). An oriented-contour points matrix $\mathbf{E}_I$ is obtained using an histogram-based threshold process. Each edge point $\mathbf{p}_i$ of $\mathbf{E}_I$ is considered as a vector in $\Re^3$: $\mathbf{p}_i$=[x,y,o]^T, where (x,y) are the coordinates and o the gradient orientation of $\mathbf{p}_i$ [10]. We sample the gradient orientations to N bins. To manage the cases of vehicles of the same type but with different colors, the modulus π is used instead of the modulus 2π [1]. In the present application, $N = 4$.

D. Model Features

1) Oriented-Contour points features array: Each class in the **KnB** is represented by n prototypes in the **TrB**. This quantity n varies from one class to another. Some classes are defined with only one prototype.

Superposing the n prototypes of the class k, we find an array of the redundant oriented-contour points. This feature array of Oriented-contour points models this class in the **KnB**. The algorithm operates the n prototypes of the class k in the **TrB** by couples (having $C_{n,2}$ couples at all). Let be $(\mathbf{E}_i, \mathbf{E}_j)$ a

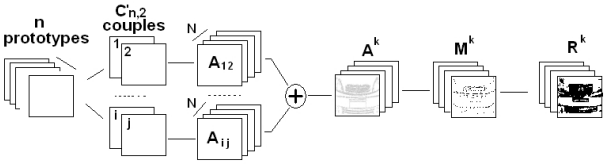


Fig. 4. Model creation.

couple of Oriented-Contour Points matrix of the prototypes 1 and 2 from the k class. We define an 600x252xN accumulator matrix A_{ij} and the vote process is as follow: a) taking a point $\mathbf{p}_i$ of $\mathbf{E}_i$, we seek in $\mathbf{E}_j$ the nearest point $\mathbf{p}_j$ with the same gradient orientation; b) the algorithm increments the accumulator A_{ij} in the middle point of $\overline{\mathbf{p}_i\mathbf{p}_j}$ at the same gradient orientation; c) the procedure is repeated for all the points $\mathbf{p}_i$ of $\mathbf{E}_i$. Considering the addition of all A_{ij} we obtain the accumulator array A^k: $A^k = \sum_{i,j} A_{ij}$. The most voted points $\mathbf{a}_m$=[x,y,o] of A^k are selected iteratively. We impose a distance of 5 pixels between the $\mathbf{a}_m$ in order to obtain a homogeneous distribution of the model points. We store $\mathbf{a}_m$ in a feature array $\mathbf{M}^k$. The array $\mathbf{M}^k$ contains the Oriented-

Contour Points that are rather stable through the n samples of the class k.

When $n = 1$, the accumulator matrix A^k cannot be computed: the feature array $\mathbf{M}^k$ is then determined from the maximum values of the gradient magnitude $|\nabla g_I|$.

2) Weighted Matrix: The Chamfer distance is applied to determine the distance from every pixel to the given $\mathbf{M}^k$ set (fig. 5). This figure shows the four R_i^k Chamfer region matrix (one for each gradient orientation) obtained after thresholding the Chamfer chart matrix D_i^k with the distances smaller than r.

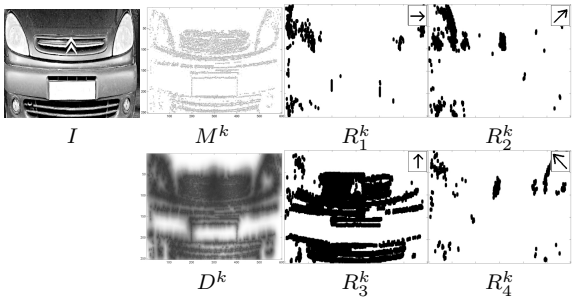


Fig. 5. Obtaining Chamfer region matrix.

Two weighted region arrays W_+^k and W_-^k will be created for each class k. W_+^k is based on the R^k region matrix where each pixel has a weight related to the discriminative power of the corresponding oriented-contour point. Pixel rarely present in other classes obtains highest weights. We assign a lower weight to the points present in the majority of the Knowledge Base classes.

$$W_+^k = \frac{1}{K-1} \sum_{i,i \neq k} (R^k - R^i \cap R^k)$$

W_-^k gives a negative weight to the points of the other models which are not present in the matrix R^k of the model k. Pixels that are present in most of the other classes obtain higher weights. In the other hand, pixel present in a few classes get lower weights.

$$W_-^k = -\frac{1}{K-1} \sum_{i,i \neq k} (R^i - R^i \cap R^k)$$

The K classes in the **KnB** are modelled by $\{\mathcal{M}_1, ..., \mathcal{M}_K\}$, where each $\mathcal{M}_k = \{\mathbf{M}^k, W_+^k, W_-^k\}$.

III. Classification

This section develops the methods to classify the samples providing from the database set using the models $\mathcal{M}_k$. A new instance t is evaluated on the classification function $G(t) = ArgMax\{g_1(t), ..., g_K(t)\}$ using a *winner-take-all* rule. The example t is labelled by $k \in \mathcal{K}$ with the highest score of the g_k. Two types of matching scores compose the g_k (see fig. 6). The first obtains a score based on three kind of votes (positive, negative and class votes) for each class k. The second score computes the distance between the oriented-contour points of the model $\mathbf{M}^k$ to the oriented-contour points of t. We prouve that the fusion of all these matching scores improves the classification results (see fig. 9).

Obtaining an image prototype of a sample t from the Test Base, we calculate the corresponding oriented-contour points matrix $\mathbf{E}_t$ (section II-C). Considering the large number of points in $\mathbf{E}_t$, we have to choose a limited set of T points. The value of T is a compromise between the computing time and a good rate of correct classifications (in our algorithm, $T = 3500$). To select these points, we construct a sorted list of the prototype positions (x, y, o). We sort in decreasing order, the values of the weighted arrays W_+^i $i = 1, ..., K$, placing the discriminant pixels (highest values) in the firsts positions of the list. Looking iteratively if the pixels in the list are present in $\mathbf{E}_t$, we add the T points in $\mathbf{P}_t$.

A. Designing the discriminant function

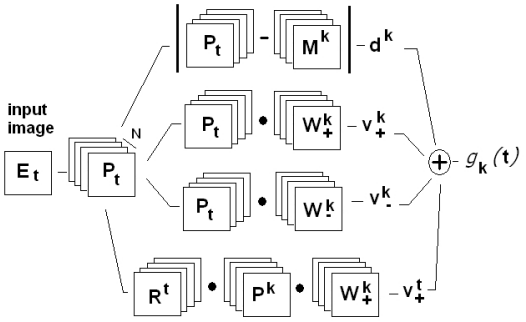


Fig. 6. Obtaining the discriminant function.

1) Positive votes: The methode consists in accumulating votes for the class k, whenever a point of $\mathbf{P}_t$ falls in a neighbourhood of a $\mathbf{M}^k$ point. We define the neighbourhood of the point $\mathbf{M}^k$ as a circle of radius r around the point of interest. These neighbourhood are computed like Chamfer regions R_i^k. Moreover, each point of $\mathbf{P}_t$ votes for the class k with a different weight depending on its value in the matrix W_+^k.

The nonzero points of the dot product ([•] in the equation 1) of $\mathbf{P}_t$ and W_+^k correspond to the points of $\mathbf{P}_t$, that belong to a neighbourhood of the $\mathbf{M}^k$'s points. Thereafter, we calculate the amount of positive votes:

$$v_+^k = \sum_x \sum_y \sum_o \mathbf{P}_t \bullet W_+^k \qquad (1)$$

2) Negative votes: The negative votes take into account the points of $\mathbf{P}_t$ that did not fall into the neighbourhood of the $\mathbf{M}_k$ points. We punish the class k by accumulating these points weighted by the matrix W_-^k. The amount of negative votes is defined as:

$$v_-^k = \sum_x \sum_y \sum_o \mathbf{P}_t \bullet W_-^k$$

3) Votes to test: We compute the votes from the models to the sample test. The method is similar to the one detailed in the preceding section. We first build the Chamfer Distances map for $\mathbf{E}_t$. We keep the regions around the oriented-contour points of $\mathbf{E}_t$ which are a distance lower than r pixels in the matrix R^t. Then, randomly selecting T points from the array $\mathbf{M}_k$, we obtain a representation of this set in an array $\mathbf{P}^k$. Each point of the matrix $\mathbf{P}^k$ is weighted by the matrix W_+^k. Total votes from the class k to the sample test t are calculated as:

$$v_+^t = \sum_x \sum_y \sum_o R^t \bullet \mathbf{P}^k \ \bullet \ W_+^k$$

4) Distance Error: The last score is the error measure of matching the $\mathbf{P}_t$ points with their nearest point in $\mathbf{M}_k$. Calculating the average of all the minimal distances, we obtain the error distance d^k [3] :

$$H(\mathbf{P}_t, \mathbf{M}_k) = max(h(\mathbf{P}_t, \mathbf{M}_k), h(\mathbf{M}_k, \mathbf{P}_t))$$

with :

$$h(\mathbf{P}_t, \mathbf{M}_k) = mean_{a \in \mathbf{P}_t}(min_{b \in \mathbf{M}_k} \|a - b\|))$$

Furthermore, we applied a decreasing function on the values of the error vector.

5) Discriminant Function: The four matching scores $\{v_+^k, v_-^k, v_+^t, d^k\}$ are combined in a discriminant function $g_k(t)$ matching the sample test t to the class k. A pseudo-distance of Mahalanobis normalizes the scores: $\bar{v} = (v - \mu)/\sigma$, where (μ, σ) are the mean and the standard deviation of v. The discriminant function is defined as:

$$g_k(t) = \alpha_1 \ \bar{v}_+^k + \alpha_2 \ \bar{v}_-^k + \alpha_3 \ \bar{v}_+^k + \alpha_4 \ \bar{d}^k \qquad (2)$$

The α_i are coefficients which weight each classifier. In our system, we give the same value for all α_i.

Finally, given the test sample t, its class label k is determined from:

$$k = G(t) = ArgMax\{g_1(t), ...g_K(t)\}$$

B. Confidence criterion

The multiclass identification system, developed in the preceding section, always gives a label for an input instance. This section studies the cases when a new instance is not present in the **KnB**. A confidence criteria allows us to accept or reject the response of the classification function $G(t)$. Rejected samples can be submitted to an exhaustive classifier or a fusion process (e.g. logo detection).

We construct a new Test Base with vehicle samples included in the **KnB** and new vehicle classes not present in the data base. Applying the classification function $G(t)$ over all the Test Base samples, we obtain 3 different responses: **CC**, the correct classifications; **BC**, bad classifications, the misclassified samples of the **KnB** and **FC**, the false classifications, when the type is not present in the **KnB**.

We place the $G(t)$ results in a 2D graphic. The y-axis represents the highest score of the $G(t)$ function. The x-axis is the difference between the two highest scores of $G(t)$.

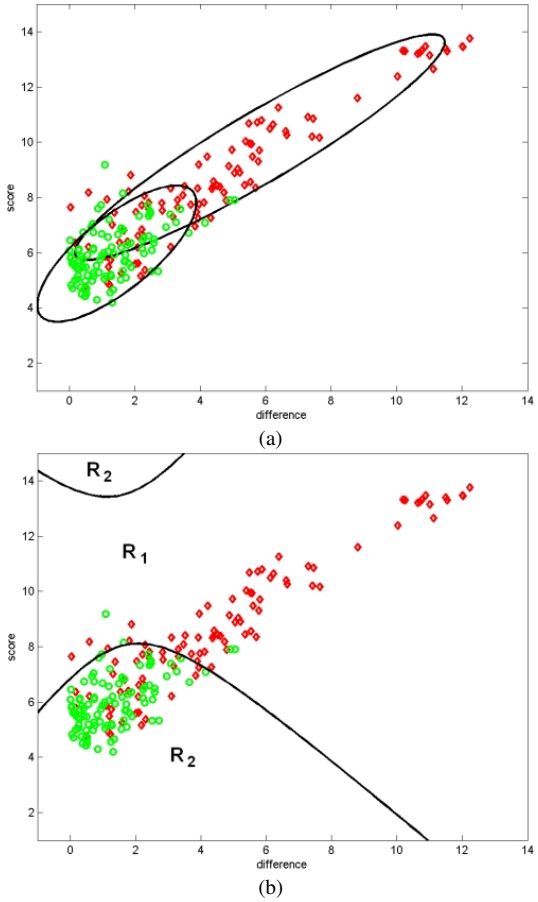


Fig. 7. (a) is a 2D representation of the results of the discriminant function $G(t)$, and the covariance ellipses of the two clusters, the diamonds represent the **CC** and the cercles the **BC+FC**, (b) is the frontier separating the **CC** region (R_1) and the **BC+FC** region (R_2) with $\lambda = 1$.

Even if the two clusters are not separable in the 2D space, the Bayes rule found the optimal frontier, considering they have binomial distributions(w_1 and w_2 representing the **CC** samples and the regrouped **BC+FC** samples respectively):

$$P(w_i|\mathbf{x}) = \frac{p(\mathbf{x}|w_i) \cdot P(w_i)}{\sum p(\mathbf{x}|w_i) \cdot P(w_i)}$$

For each w_i, we can obtain: $P(w_i)$ the *a priori* probability of this cluster, and $p(\mathbf{x}|w_i)$ the probability density function. The normal bi−variable distribution is defined by $N(\mu_i, \Sigma_i)$. Both covariance matrix have differents values for each cluster (see fig. 7.a). Applying the logarithms we obtain a discriminant function replacing the $p(\mathbf{x}|w_i)$ for the normal bi−variable :

$$f_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \cdots$$
$$- \frac{d}{2} log(2\pi) - \frac{1}{2} log|\Sigma_i| + log(\pi_i) \qquad (3)$$

where π_i = $P(w_i)$.

Two discriminants functions are obtained : f_1 (**CC** samples) and f_2 (**BC+FC** samples). For a x point, the rule 4 is evaluated in order to determine if the sample is placed in the **CC** region **R1**, or otherwise.

$$f_1(x) < \lambda f_2(x) \tag{4}$$

This criterion maximizes the quantity of **CC** samples placed in **R1** and at the same time minimize the **BC** and **FC** samples placed in this region.

IV. Results

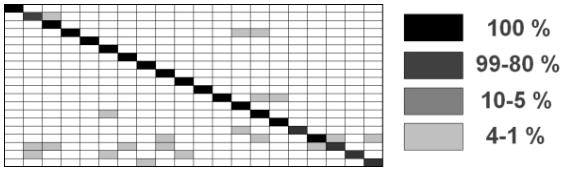


Fig. 8. classification results in the Confusion Matrix.

The Oriented-Contour Voting Algorithm correctly identifies 93.75 % of samples from the Confusion Matrix (fig. 8). The Confusion Matrix respects the order defined in the table 1 and the percentages are computed on the rows (e.g. in the fourth row -class Citroën Saxo in the table 1)- 19 samples are correctly classified and two are labelled as Renault Clio D and Renault Laguna).

In figure 9 we clearly see that the fusion of the four matching scores obtains the best results in the classification. Better rates could be obtained: a training algorithm could optimised the α_i values, but we have to acquire more frontal view vehicle samples.

The ROC (receiver-operating-characteristic) curve showed in the figure 10, is obtained varying λ in equation (4). For $\lambda = 0.41$, we obtain 91.46 % of **CC** and 20.13 % of **BC+FC** in **R1**. This means 84.79 % of **CC** from the total and 1.46 % of **BC**. For this value of λ, only 12.84 % of the retained samples (those placed in **R1**) are the false alarms. The final λ value depends on the application requirements: i.e., if the final system is designed in order to be combined with others identification methods (license plate recognition, electronic cards, etc.), higher rejection ratios could be admitted (the global error will be compensated for the combination).

Another test simulates the presence of the tollgate at four different locations: in a car park access control system it is

Matching scores	Correct Classifications (%)
v_+	92.08
v_+,v_-	91.25
v_+,v_t	90.42
v_+,d	89.17
v_+,v_-,v_t	93.54
v_+,v_-,d	89.79
v_+,v_-,v_t,d	93.75

Fig. 9. Matching scores fusion results.

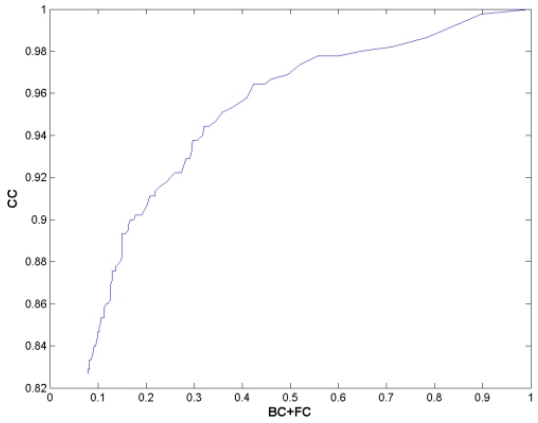
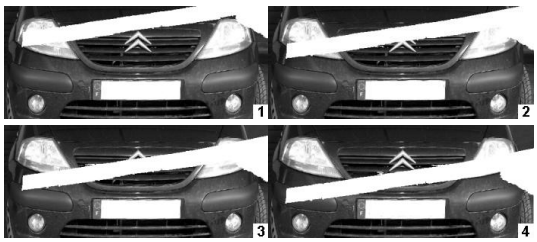


Fig. 10. ROC Curve varying λ in equation 4.

very difficult to define the relative vertical position between the barrier and the vehicle even if the license plate is always visible (see fig. 11.a). The results for each tollgate location are showed in figure 11.b. The lowest results (for the 4th position) comes from the fact that the virtual barrier hides the nearest part of the license plate. These points contain important information for the classification and they also are the more stable points before the affine transformation.



(a)

Toll Position	Correct Classifications (%)
1	91.25 %
2	91.88 %
3	90.92 %
4	88.33 %

(b)

Fig. 11. (a) the four positions of a virtual tollgate, hiding a 15% of the pattern I, and (b) the Oriented-Contour Points Set based algorithm results.

V. Conclusions

This article has presented a voting system for the multiclass vehicle type recognition based on Oriented-Contour Points Set. Each vehicle class is composed from one or many grayscale frontal images of one vehicle type (manufacturer and model). A discriminant function combines the scores provided from three voting based classifiers and an error distance. We have tested this method on a realistic data sets of 480 frontal images of cars. Our recognition rate is over 93%. The results showed that the method is robust to a partial occlusion of the patterns. Furthermore a confidence criterion will allow to combine it with others process like license plate recognition systems or to reject vehicles which are not in the **KnB**.

Without occlusion, our system obtains similar results as [11]. Future works will be oriented to increase the number of classes in our **KnB**. This operation would reduce the classification rates: a preliminary test with 30 classes gives a recognition rate of over 89.2%. But we expect to compensate this effect in adding more classifiers. Our voting classifiers are closed to a weighted Modified Hausdorff Distance [3], with heaviside step functions applied on a Distance Transform. So we would test others linear or nonlinear functions to design new classifiers. Another way consists in creating subclasses with more than one vehicle type and designing specific classifiers for each subclasses and for the set of subclasses. We would obtain a hierarchical decision tree [6].

REFERENCES

[1] T. Cootes and C. Taylor. On representing edge structure for model matching. In *Conference on Vision and Pattern Recognition*, volume 1, pages 1114–1119, Hawai, USA, December 2001.

[2] J. Douret and R. Benosman. A multi-cameras 3d volumetric method for outdoor scenes: a road traffic monitoring application. In *International Conference on Pattern Recognition*, pages III: 334–337, 2004.

[3] MP. Dubuisson and A. Jain. A modified hausdorff distance for object matching. In *International Conference on Pattern Recognition*, volume A, pages 566–569, 1994.

[4] MP. Dubuisson-Jolly, S. Lakshmanan, and A. Jain. Vehicle segmentation and classification using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):293–308, 1996.

[5] J. M. Ferryman, A. D. Worrall, G. D. Sullivan, and K. D. Baker. A generic deformable model for vehicle recognition. In *British Machine Vision Conference*, pages 127–136, 1995.

[6] D.M. Gravila. A pedestrian detection from a moving vehicle. In *European Conference on Computer Vision*, pages 37–49, 2000.

[7] D. Han, M.J. Leotta, D.B. Cooper, and J.L. Mundy. Vehicle class recognition from video-based on 3d curve probes. In *VS-PETS*, pages 285–292, 2005.

[8] D. Hond and L. Spacek. Distinctive descriptions for face processing. In *British Machine Vision Conference*, University of Essex, UK, 1997.

[9] A.H.S. Lai, G.S.K. Fung, and N.H.C. Yung. Vehicle type classification from visual-based dimension estimation. In *IEEE International System Conference*, pages 201-206, 2001.

[10] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, 1997.

[11] V.S. Petrovic and T.F. Cootes. Analysis of features for rigid structure vehicle type recognition. In *British Machine Vision Conference*, volume 2, pages 587–596, 2004.

[12] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711, May 2006.

[13] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.