An Oriented-Contour Point Based Voting Algorithm for Vehicle Type Classification

Pablo Negri, Xavier Clady, Maurice Milgram
Université Pierre et Marie Curie-Paris6,
EA 2385, Paris, F-75005 France
pablo.negri@lisif.jussieu.fr

Raphael Poulenard
LPREditor
Montpellier, France
raphael.poulenard@lpreditor.com

Abstract

This communication deals with an Oriented-Contour Point based voting algorithm for multiclass vehicle type identification (make and model). The system obtains similar results for equivalent recognition frameworks with different feature selections [8]. Results also show the method to be robust to partial occlusion.

1. Introduction

Vision-based license plate recognition is often used to check incoming (or outcoming) cars in parking or toll road. To increase robustness of such systems, we propose to combine it with other process dedicated to identify vehicle type (make and model). The aim of the system described in this article will be the vehicle type identification from a vehicle greyscale frontal image.

Many vision-based Intelligent Transport Systems are dedicated to detect, track or recognize vehicles in image sequences. Embedded cameras detect obstacles and compute distances from the equipped vehicle [11]. Surveillance road monitoring measures traffic flow [2, 10]. Vehicles are localized in an image using 2D or 3D bounding box [11, 6] or geometrical models which classify vehicles in categories suchs sedans, minivans, SUV[1] or trucks[4, 3].

One paper deals with a similar problem: Petrovic and Cootes [8] test various features for vehicle type classification. Their decision module is based on a simple Euclidean measure (with or without PCA pre-stage). 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 colour. Others works [5, 1] had took the edge orientations for the recognition of different patterns like faces.

In this paper, a vehicle type is a class represented by an Oriented-Contour Points based model. We have also take into account occlusions (tollgate) hiding a part of the vehicle and making inadequate simple appearance-based methods. We shall see that in spite of the presence of tollgate, our system does not need to change the training set or apply time-consuming reconstruction process.

Our classifier is based on a voting algorithm and on a Euclidean edge distance. For an input image, a discriminant function gives a *score* to each class in the system's type list. The input then is identified as the best match in the type list; that is simply the class with the highest *score*.

Next section explains how we define the model. Section 3 employs this model to obtain the discriminant function. Results are presented in the following section. We finish with conclusions and prospects.

2. Model Creation

During the initial phase of our algorithm, we produce an Oriented-Contour Point based model for all the K vehicle type classes composing the system knowledge. We call Knowledge Base (**KnB**) the list of classes the system is able to recognize.

2.1. Images Databases & Confusion Matrix

All ours experiments have been carried out on the Training Set (**TrB**) and on the Test Set (**TsB**). The **TrB** set is used to produce the oriented-contour point models of the vehicle classes. While the **TsB** sample is used to evaluate the accuracy of the classification system. **TrB** is composed of high quality frontal vehicle images captured in different car parks. On the other hand, **TsB** includes outdoor frontal vehicle images under different light conditions and at a lower resolution. In figure 1, the upper row shows samples from

[1] SUV or Sport Utility Vehicle is a type of passenger vehicle like 4x4.

TrB while the bottom row shows the corresponding vehicle class of the **TsB**. Our classification system will be applied

Figure 1. Samples from the TrB and the TsB.

to a Confusion Matrix (see table 1). Formally, we select a finite set $\mathcal{K}$ of $K = 20$ classes. For the multiclass classification problem, each example $t \in \mathcal{T}$ is assigned toa single class $k \in \mathcal{K}$, so that labelled examples are pairs (t,k). The system objective is to find a function $G : \mathcal{T} \rightarrow \mathcal{K}$ which matches a newly 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. Number of images.

2.2. Prototype image

We create a canonical rear-viewed vehicle image from the four corner points of the license plate {**A**,**B**,**C**,**D**} (see fig. 2). The image templates are called prototypes and in the present work are 600 * 252 pixels (rows * columns). In order to correct the orientation of the original image (see examples in fig. 1), an affine transformation moves original points {**A**,**B**,**C**,**D**} to the desired {**A'**,**B'**,**C'**,**D'**} reference position, considering the vehicle grille and the license plate in the same plane. Bigger ROIs, (i.e. with roofs, windshields and wheels), do not respect this assumption, so the affine transformation produces big mistakes in the reconstruction. The LPREditor's license plate recognition system provides us the corners of the vehicle license plate (see at http://www.lpreditor.com for details).

(a) (b)

Figure 2. (a) original image, (b) prototype I.

The Sobel operator is used to calculate the magnitude and orientation of the I gradient greyscale prototype ($|\nabla g_I|, \phi_I$). We obtain an oriented-contours points matrix $\mathbf{E}_I$ after an histogram based threshold process. We consider each edge point $\mathbf{p}_i$ of $\mathbf{E}_I$ as a vector in $\Re^3$: $\mathbf{p}_i$=[x,y,o]', where (x, y) is the point position, and o is the gradient orientation of $\mathbf{p}_i$ [7]. We sample the gradient orientations to N bins. To manage the cases of vehicles of the same type but with different colours, we use the modulus π instead of the modulus 2π [1]. For our application $N = 4$.

2.3. Model Features

Oriented-Contour points features array Each class in the **KnB** is represented by n prototypes in the **TrB**, n varies from class to class, some classes are represented by a single prototype.

Superposing the n prototypes of the class k, we find a map of the redundant oriented-contour points. This feature map of Oriented-Contour based points models this class in the **KnB**. The algorithm treats the n prototypes of the class k in the **TrB** by couples (having $C_{n,2}$ couples at all). Let

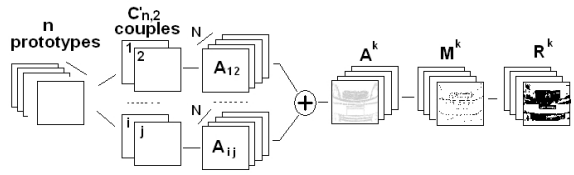


Figure 3. Model creation.

be ($\mathbf{E}_i, \mathbf{E}_j$) a 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 $\mathbf{M}^k$. So, the map $\mathbf{M}^k$ 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 determined from the maximum values of the gradient magnitude $|\nabla g_I|$.

Weighted Matrix A Rosenfeld transformation [9] is applied to determine the distance from every picture element

to the given $\mathbf{M}^k$ set. The figure 4 shows the four R_i^k Chamfer region matrix (one for each gradient orientation) obtained after thresholding the Chamfer chart matrix D_i^k when the distances are smaller than r.

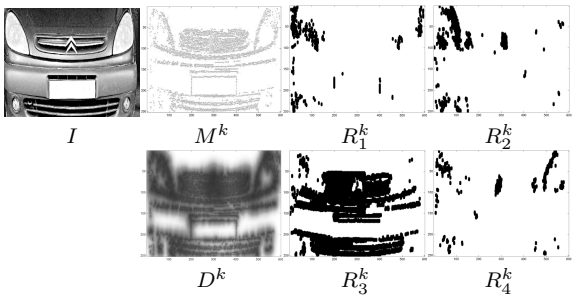


Figure 4. Obtaining Chamfer region matrix.

Two weighted regions maps 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 discrimination power of the corresponding oriented-contour points. The points of k, which are rarely present in the others classes, obtain the highest weights:

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

where the binary function f_p equals **1** if the argument is 1, and 0 otherwise. Similarly, 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: $W_-^k = \frac{1}{K} \sum_{i=1 \wedge i \neq k}^{K} f_n(R^k - R^i)$, where the binary function f_n equals **1** if the argument is -1, and 0 otherwise.

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

3. Classification

This section develops the methods to classify the test samples using the models $\mathcal{M}_k$. A new instance t is classified using the winner takes all rule: $G(t) = ArgMax\{g_1(t), ..., g_K(t)\}$. Two types of matching scores compose the g_k. The first obtains a score based on three kind of votes (positive ,negative and class votes) for each class k. The second score evaluates the distance between the oriented-contour points of the model $\mathbf{M}^k$ to the oriented-contour points of t.

An Oriented-Contour Points matrix $\mathbf{E}_t$ (section 2.2) is calculated for each example t. We randomly select T points from $\mathbf{E}_t$. These points are regrouped in an 600x252xN matrix $\mathbf{P}_t$. The value of T is a compromise between the computing time and a good rate of correct classifications.

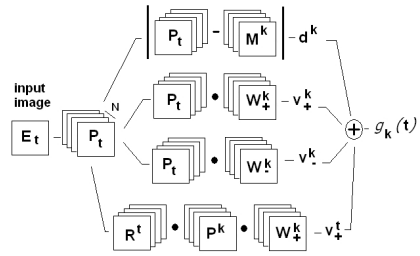


Figure 5. Obtaining the discriminant function.

Positive votes The methodology 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. This neighbourhood representation is modelled in the 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 of 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 in equation (1) where [•] is the dot product.

$$v_+^k = \sum_x \sum_y \sum_o P_t \bullet W_+^k \tag{1}$$

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: $v_-^k = \sum_x \sum_y \sum_o P_t \bullet W_-^k$

Votes to test We calculate the votes from the models to the sample test. In short, the method is the same as the one detailed in the preceding section. We first build the chart of Chamfer Distances for $\mathbf{E}_t$. We keep the regions around the oriented-contour points of $\mathbf{E}_t$ which are at 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 P^k. Each point of the matrix P^k is weighted by the matrix W_+^k. Total votes from the class k to the sample test t are calculated as follows: $v_+^t = \sum_x \sum_y \sum_o R^t \bullet P_k \bullet W_+^k$

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. Furthermore, values in the error vector have to be processed by a decreasing function considering that in the vote vectors we search for the maximum and for the error vector we search for the minimum.

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-Mahalanobis distance normalizes the scores: $\bar{v} = (v - \mu)/\sigma$, where (μ, σ) are the mean and the standard deviation of v. The matching function is defined as:

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

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 label k is determined from:

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

4. Results

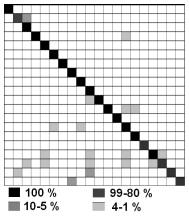
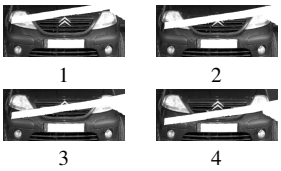


Figure 6. Confusion Matrix results.

Matching Scores	Mean Success
v_+	91.0 %
v_-	85.4 %
v_t	89.5 %
d	82.1 %
fusion	92.4 %

Figure 7. Matching scores results.

Successive tests on the Confusion Matrix shown that the Oriented-Contour Voting Algorithm correctly identifies in average 92.4% samples from the Confusion Matrix. A standard deviation of $\pm 1.1\%$ in the average result is the consequence of the randomised selection of the Oriented-Contour Points from the $\mathbf{E}_t$.



(a)

Position	Mean Success	Deviation
1	93.1 %	± 0.9
2	92.2 %	± 1.1
3	90.9 %	± 0.7
4	88.5 %	± 1.4

(b)

Figure 8. (a) the four positions of a virtual tollgate, (b) results for the virtual tollgate.

Results in figure 7 show that the fusion rule obtains better results than each individual match score. Additional experiment, where the Knowledge Base is composed by only one prototype for every class, results in an average recognition rate of 85.6%. This result can be explained as follows: the presence of multiple prototypes allows to filter edge noise. Another experiment simulates the presence of the tollgate at four different locations, hiding 15% of the pattern I (see fig. 8.a). The average results and the standard deviations for each tollgate location are showed in the table of figure 8.

5. Conclusions

We present in this paper a voting algorithm for a multiclass vehicle type recognition system based on Oriented-Contour Points. Similar results as [8] are obtained utilizing a different feature selection and classification method for a smaller number of classes. Results show that the method is robust to partial occlusions. Furthermore, this algorithm can be implemented in real time application due to the fast matrix operations of image libraries like *OpenCV*. Future works will be oriented to improve the selection of the Oriented-Countour Points of the sample test in the classification phase instead of random selection. A confidence criterion for the discriminant function could be developed to evaluate the results. This criterion is necessary in order to combine our system with a license plate recognition system and for rejection purpose.

References

[1] T. Cootes and C. Taylor. On representing edge structure for model matching. In *CVPR*, 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 *ICPR04*, pages III: 334–337, 2004.

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

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

[5] D. Hond and L. Spacek. Distinctive descriptions for face processing. In *BMVC*, University of Essex, UK, 1997.

[6] A. Lai, G. Fung, and N. Yung. Vehicle type classification from visual-based dimension estimation. In *IEEE Int. Transp. Syst. Conf.*, pages 201–206, 2001.

[7] 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.

[8] V. Petrovic and T. Cootes. Analysis of features for rigid structure vehicle type recognition. In *BMVC*, volume 2, pages 587–596, 2004.

[9] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *Ass. Comp. Mach.*, 54(1):471–494, 1966.

[10] S. L. Skszek. State-of-the-art report on non-traditional traffic counting methods. Technical report, FHWA-AZ-01-503, October 2001.

[11] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *PAMI*, 28(5):694–711, May 2006.