

Reconnaissance multiclassées de type de véhicules à l'aide d'algorithme de votes sur des contours orientés

Xavier CLADY¹, Pablo Augusto NEGRI¹, Maurice MILGRAM¹, Raphaël POULENARD²

¹Université Pierre et Marie Curie-Paris6, ISIR, CNRS FRE 2507
BC 252, 4 place jussieu, 75252 Paris Cedex 05, France

²LPR Editor, Montpellier, France

pablo.negri@lisif.jussieu.fr, xavier.clady@upmc.fr
maurice.milgram@upmc.fr, raphael.poulenard@lpreditor.com

Résumé – Cette communication présente un système de reconnaissance du type (constructeur, modèle) de véhicules par vision. A partir d'une vue de face avant d'un véhicule, nous construisons une représentation de celle-ci basée sur les points de contour orientés. La classification est réalisée essentiellement en se fondant sur des algorithmes de votes. La classe d'un véhicule est déterminée selon celle de son plus proche voisin dans l'espace des votes. Plusieurs résultats effectués sur des bases d'images prises en conditions réelles (contenant 50 types de véhicules différents) sont présentés et analysés : le taux de reconnaissance dépasse les 90%.

Abstract – This article deals with a system for multiclass vehicle type identification (manufacturer and model). From a frontal vehicle image, an oriented points set is computed in order to match it with vehicle type models. The decision process uses voting algorithms. A simple nearest neighbour classification is used to determine the vehicle type associated with each voting vector. This method has been tested on a realistic data set (containing 50 different vehicle type classes) : the recognition rate reaches over 90%.

1 Introduction

Les systèmes de vision pour la reconnaissance de véhicules s'intègrent dans diverses applications dites Systèmes de Transports Intelligents. Trois principales doivent être distinguées. En premier lieu, des caméras embarquées sont dédiées à la détection des véhicules dits obstacles [7] se présentant au devant du véhicule équipé pour le calcul des distances relatives. En second lieu, la télésurveillance des autoroutes et autres voies de circulation permet la mesure des flots de véhicules [1]. Enfin il existe des systèmes ayant pour rôle d'identifier les véhicules se présentant aux entrées ou sorties de zones télésurveillées (parking, télépaiement,...).

Ces derniers utilisent généralement la lecture automatique de la plaque d'immatriculation. Cet identifiant est en théorie suffisant pour reconnaître le véhicule. Cependant, en pratique, les systèmes de vision s'y conformant uniquement, peuvent renvoyer des informations erronées, soit du fait d'une mauvaise qualité d'image soit d'une plaque illisible ou fautive. Nous pensons qu'y adjoindre, un système de reconnaissance du type de véhicule (constructeur et modèle) ne peut qu'en accroître leur efficacité et leurs capacités. En effet, cette information peut confirmer ou infirmer celle fournie par la reconnaissance de la plaque (cf. figure 1).

A notre connaissance, seuls Petrovic et Cootes [6] ont abordé un problème similaire dans une étude comparative sur la représentation par apparence d'objets à structure rigide. Ils ont appliqués différentes méthodes d'extraction

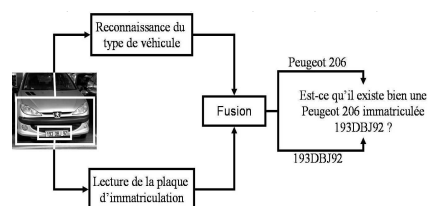


FIG. 1 – Système d'identification de véhicule fusionnant la reconnaissance de son type et la lecture de sa plaque d'immatriculation.



FIG. 2 – Exemple d'images.

de *features* sur une base d'images de véhicules en stationnement. Dans une situation réelle d'application, les images sont essentiellement cadrées de manière à rendre bien visibles les plaques d'immatriculation (cf. fig. 2). Aussi pour reconnaître le type des véhicules, nous allons utiliser leur image des calandres avant (que nous appellerons prototype). Par ailleurs, nous pouvons constater la présence d'une barrière masquant une partie de la calandre. De plus, les images peuvent être prises de différents points de vue (suivant le site d'implantation du système). Enfin, dans ce type d'application, un fonctionnement en temps réel est souvent requis. Toutes ces contraintes rendent inadéquates les méthodes via apparence, comme celles décrites dans [6]. En effet, celles-ci sont très sensibles aux occlusions et aux changements de points de vue, et sont souvent gourmandes en mémoire et en temps de calcul. Nous avons donc opté pour une méthode basée sur des primitives.

Chaque modèle de véhicule a un gabarit constant, défini par le constructeur. Ceci rend les contours des calandres relativement invariants d'une image d'un véhicule à celle d'un autre du même type. Notre méthode de classification se base essentiellement sur un processus de votes avec une caractérisation locale de chaque pixel de contours orientés réduite au minimum (positions et orientation). Ceci est motivé par le fait que dans les méthodes de votes, c'est le nombre qui fait la force : si la caractérisation des points d'intérêts (i.e. des votants) est trop sélective, le nombre de votants diminue. De plus, la sélection du modèle ne limite pas à celui ayant reçu le plus de votes. Dans cette communication, nous proposons de représenter chaque modèle par un ou plusieurs échantillons dans l'espace de votes. Un processus de plus proche voisin dans cet espace nous permet de décider à quel modèle appartient l'image testée.

Dans la suite de cet article, nous allons d'abord introduire la manière dont nous avons construit notre modèle de représentation de chaque type de véhicules à reconnaître. Plus nous expliciterons notre méthode de classification. Plusieurs résultats, démontrant l'efficacité de notre méthode seront exposés. Enfin, la conclusion de cet article sera l'occasion de proposer quelques perspectives.

2 Création du modèle

Dans cette section, nous décrivons comment nous construisons un modèle unique pour chaque type de véhicules. Notre Base de Connaissance est composée de $K = 50$ classes de véhicules. Nous disposons de deux bases d'images en niveaux de gris des voitures vues de face : une Base d'Apprentissage de 291 images (TrB) et une Base de Test de 830 images (TsB). Ces bases sont composées d'images prises dans des parkings avec des appareils numériques ainsi que d'images prises au niveau d'un feux tricolores avec un caméscope numérique. Les images sont prises sous différents points de vue et sous différents éclairages (cf. fig. 3).



FIG. 3 – Exemples de notre base d'images.

2.1 Prototype et pixels de contours orientés

A partir de l'image originale d'une voiture, nous obtenons une imagerie de taille 252x600 où la plaque minéralogique est placée dans une position connue (cf. figure 4) via une transformation affine. Les coins de la plaque sont obtenus via une méthode développée par LPREditor (pour plus d'informations, voir <http://www.lpreditor.com>).

Pour le calcul des contours orientés, nous utilisons des filtres de Sobel de taille 3x3 sur l'image prototype I . Une matrice \mathbf{E}_I représentant les contours du prototype I est déterminée après une opération de seuillage sur les valeurs du module du gradient. Dans la suite, nous nommerons $\mathbf{p} = [x, y]$ les pixels de contour appartenant à \mathbf{E}_I , où (x, y)



FIG. 4 – (a) image originale, (b) prototype I .

est la position du point. Les orientations, notées $\phi(\mathbf{p})$, sont quantifiées pour qu'ils prennent des valeurs entières entre 0 et $N-1$ (ici N est égal à 4), en tant compte d'un modulo π au lieu du module 2π pour gérer les cas des voitures du même type mais de différentes couleurs.

2.2 Un modèle unique par type

Pour chaque type de véhicule, nous allons dans cette section déterminer un modèle unique et représentatif de sa classe à partir de toutes les imageries de véhicules de ce type. En effet, une classe k est représentée dans la Base d'Apprentissage par n prototypes. Cette quantité varie d'une classe à l'autre. Un processus d'accumulation, similaire à un filtrage inter-images et illustré dans le pseudocode de la figure 5, permet de ne retenir que les pixels redondants d'une image à l'autre pour un même type. Les pixels retenus sont regroupés dans une matrice notée \mathbf{M}^k via une fonction f_{max} qui sélectionnent les pixels ayant reçu le plus de votes dans A^k tout en leur préservant une distribution homogène (en faisant attention que les points soient distants d'au moins 5 pixels). Le lecteur pourra consulter [5] pour en obtenir plus de détails.

```

Soient  $n$  prototypes pour la classe  $k$  parmi les  $K$  modèles
 $A^k(600, 252, N) = 0$ , matrice d'accumulation
pour  $i=1, \dots, n$ 
  obtenir  $\mathbf{E}_i$ 
  pour  $j=1, \dots, n$  et  $j \neq i$ 
    détermine  $\mathbf{E}_j$ 
     $\forall \mathbf{p}_i \in \mathbf{E}_i$ , trouver  $\mathbf{p}_j \in \mathbf{E}_j / \phi(\mathbf{p}_i) = \phi(\mathbf{p}_j)$ 
    et  $\min_{\forall j} |\mathbf{p}_i - \mathbf{p}_j| < t$ 
    détermine  $\mathbf{p}_m(x_m, y_m)$  tel que  $\overline{\mathbf{p}_i \mathbf{p}_j} = \frac{\mathbf{p}_i \mathbf{p}_j}{2}$ 
    vote+1 pour l'élément du tableau  $A^k(x_m, y_m, \phi(\mathbf{p}_i))$ 
 $\mathbf{M}^k = f_{max}(A^k)$ 

```

FIG. 5 – Pseudocode pour la création du modèle.

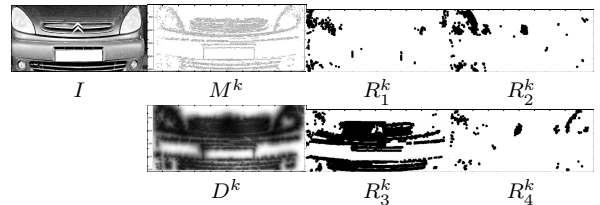


FIG. 6 – \mathbf{M}^k est le modèle de points de contour orientés avec les N orientations superposées. D^k est la carte de distance de Chamfer de \mathbf{M}^k . Les R_i^k sont les voisinages après application d'un seuil r pour chaque orientation du gradient i .

Comme nous allons le voir dans la section suivante, notre algorithme de classification est essentiellement basé sur plusieurs calculs de votes. Un de ceux-ci repose sur le principe suivant : chaque point de contour de l'exemple de test \mathbf{t} vote pour un modèle s'il tombe dans le voisi-

nage d'un point de ce modèle. Les autres sont construits selon ce modèle. Pour estimer les voisinages, la distance de Chamfer est appliquée à \mathbf{M}^k pour déterminer les cartes des distances D_i^k par rapport à ses éléments (où i représente l'orientation et k la classe de voiture). La figure 6 montre les quatre R_i^k matrices de régions de Chamfer obtenues après le seuillage de la matrice D_i^k en gardant les distances plus petites que $r = 5$. Les R_i^k sont regroupées dans une seule matrice R^k de dimension $600 \times 252 \times N$. Afin d'optimiser la méthode de votes, une pondération des points est calculée sous forme de matrices notées W_{\pm}^k (afin que les points les plus discriminants du modèle de \mathbf{p}_m par rapport aux autres modèles, aient un vote de poids plus important) : $W_+^k = \frac{1}{K-1} \sum_{i,i \neq k} (R^k - R^i \cap R^k)$ et $W_-^k = -\frac{1}{K-1} \sum_{i,i \neq k} (R^i - R^i \cap R^k)$

3 Classification

La classification consiste à associer un exemple \mathbf{t} dit de test à une classe k du dictionnaire des classes. Nous appliquons à \mathbf{t} les mêmes opérations que lors de la création des modèles. Nous obtenons donc une matrice des points de contour orientés \mathbf{E}_t , ainsi qu'une matrice \mathbf{P}_t de taille $600 \times 252 \times N$ et construite telle que \mathbf{P}_t a la valeur 1 dans la position des points sélectionnés et 0 ailleurs.

Un premier processus consiste à faire voter l'exemple pour chaque classe. Un second se sert de l'ensemble des votes comme d'un vecteur caractéristiques du modèle. Ces votes se présentent comme la combinaison de scores provenant de quatre classifieurs.

Votes positifs Le score dit de votes positifs est incrémenté si un point de \mathbf{P}_t tombe dans le voisinage d'un point de \mathbf{M}^k . Ceci peut être réalisé simplement en faisant le produit terme à terme (noté \bullet) entre \mathbf{P}_t et W_+^k :

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

Votes négatifs Les votes négatifs prennent en compte les points du \mathbf{P}_t qui ne sont pas tombés dans le voisinage de \mathbf{M}_k . Nous punissons la classe k en accumulant ces points (pondérés par la matrice W_-^k) avec une implémentation similaire à la précédente.

Votes vers le test Ensuite, nous allons calculer les votes des modèles vers l'exemple de test. Sommairement, la méthode est la même que celle détaillée dans la section précédente. Nous construisons la matrice des régions R^t pour \mathbf{E}_t . Par la suite, nous construisons la matrice P^k pour chaque modèle k . Le calcul de ce vote se déclinera aussi comme une multiplication de matrices.

Erreur de distance Le dernier score est issu de l'erreur en distance de mise en correspondance entre les points de \mathbf{P}_t avec les points plus proches de \mathbf{M}_k pour la classe k . Elle est connue comme la distance de Hausdorff modifiée [2] à laquelle nous y appliquons une fonction décroissante.

Classification Une première stratégie consiste à combiner ces scores, que nous nommons $\{v_{+k}^{mh}, v_{-k}^{mh}, v_{+t}^{mh}, d_k^{mh}\}$ après une normalisation de type mahalanobis, en une fonction discriminante $g_k(t)$ qui mesure la concordance entre l'exemple de test \mathbf{t} et la classe k :

$$g_k(t) = \alpha_1 v_{+k}^{mh} + \alpha_2 v_{-k}^{mh} + \alpha_3 v_{+t}^{mh} + \alpha_4 d_k^{mh}$$

Les coefficients α_i sont fixés ici à 0.25. La classe de l'exemple de test \mathbf{t} est alors celle qui obtient le plus grand score pour la fonction de discrimination :

$$\mathbf{t} \in c/c = \text{ArgMax}_{k=1..K} g_k(\mathbf{t})$$

La seconde stratégie développée consiste à considérer soit :

- $v(t) = [v_{+k}^{mh}, v_{-k}^{mh}, v_{+t}^{mh}, d_k^{mh}]_{k=1..K}$ comme un vecteur appartenant à un espace de dimension $200 (= 4 * K)$, noté Ω_{sf} (sf dénote qu'il n'y a pas eu de fusion des scores comme dans la fonction de discrimination).
- $v^{ACPX}(t) = [v_{+k}^{mh}, v_{-k}^{mh}, v_{+t}^{mh}, d_k^{mh}]_{k=1..K}^{ACPX}$ comme un vecteur appartenant à un espace de dimension X , noté Ω_{sf}^{ACPX} , qui est une réduction de l'espace précédent après une analyse en composantes principales.
- $g(t) = [g_k(t)]_{k=1..K}$ comme un vecteur appartenant à un espace de dimension $50 (= K)$, noté Ω_{af} (af dénote qu'il y a eu fusion des scores).

La classe de t sera donc celle de son plus proche voisin dans ces espaces. Il faut alors disposer d'échantillons représentatifs de chaque classe dans ces espaces. Pour cela, nous avons procédé par validation croisée : la Base de Test est décomposée en 2 parties égales. La première servira de référence dans l'espace considéré, dit de votes. Les échantillons de la seconde seront classés selon le processus de plus proche voisin vis-à-vis des échantillons de la première.

4 Résultats et Discussion

Avec la première stratégie, le système identifie seulement 80,2% des 830 exemples, ce qui correspond au taux de reconnaissance en microprécision [8] ; le taux de reconnaissance en macroprécision, c'est-à-dire la moyenne des taux de reconnaissance individuels, est de 69,4%.

La seconde stratégie réalise de meilleurs taux de reconnaissance (en moyenne sur 100 tirages aléatoires) :

- pour un premier espace Ω_{sf} , 93,1% en microprécision et 83,5% en macroprécision.
- pour un second espace Ω_{sf}^{ACPX50} , 86,2% en microprécision et 78,8% en macroprécision.
- pour un troisième espace Ω_{af} , 90,6% en microprécision et 86,4% en macroprécision.

La figure 7 représente les courbes *Cumulative Match Characteristic* (CMC¹) obtenues pour les espaces Ω_{sf} et Ω_{af} . Nous pouvons constater que la seconde stratégie sans fusion des scores et sans ACP donne clairement de meilleurs résultats, mais au prix d'un coût algorithmique supplémentaire car on travaille dans un espace à plus grande dimension. Comme nous le montre la figure 8, il faut conser-

¹La courbe CMC est utilisée pour mesurer la performance d'un système d'identification. Elle donne le taux de reconnaissance en fonction du rang.

ver une dimension $X > 100$ pour obtenir des taux similaires à ceux obtenus avec fusion. De plus, remarquons que la fusion peut être encore améliorée². Par ailleurs, signalons que les taux de reconnaissance varient en fonction de la proportion (et donc du nombre) d'échantillons pris dans la TsB comme références lors du processus de validation croisée, comme nous le montre la figure 9.

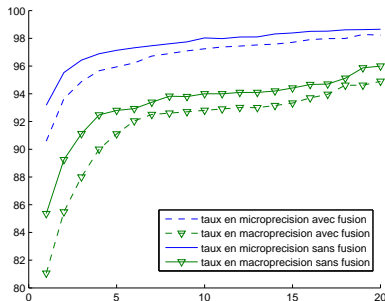


FIG. 7 – Courbes CMC ; les courbes en lignes pleines représentent les taux obtenus dans l'espace Ω_{sf} et les courbes en pointillés ceux dans l'espace Ω_{af} .

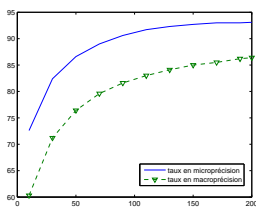


FIG. 8 – Evolution des taux de reconnaissance sur Ω_{sf}^{ACPX} en fonction de la dimension X .

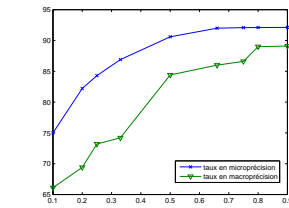


FIG. 9 – Evolution des taux de reconnaissance (dans Ω_{af}) en fonction de la proportion d'échantillons.



Position	première stratégie	Ω_{sf}	Ω_{sf}^{ACPX50}	Ω_{af}
1	84,0 %	87,3 %	87,1 %	89,0 %
2	78,5 %	84,5 %	84,1 %	85,6 %
3	78,6 %	84,5 %	83,8 %	85,3 %
4	80,2 %	87,5 %	85,9 %	87,4 %

FIG. 10 – les 4 positions de la barrière virtuelle (en haut) et leur taux de reconnaissance respectifs (en bas).

Dans un autre test, nous avons simulé la présence de la barrière en quatre positions différentes (cf. fig. 10) afin vérifier la robustesse de notre méthode aux occlusions. A noter que les taux de reconnaissance selon la première stratégie sont meilleurs lorsque la barrière couvre le haut de l'image. Cette partie est en effet constituée essentiellement de la carrosserie et contient donc beaucoup de points de contours qui sont issus de reflets (cf. fig. 12) et qui altèrent le processus de reconnaissance. Ceci est corroboré

²Un processus d'apprentissage permettrait une plus grande acuité dans le choix des coefficients α_i ; cela nécessiterait bien entendu une base de données plus importante.

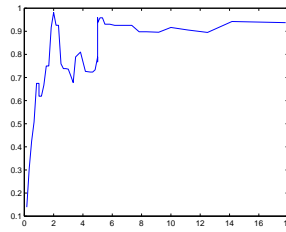


FIG. 11 – Evolution moyenne des taux individuels en fonction du nombre d'images considérées lors de la création des modèles.

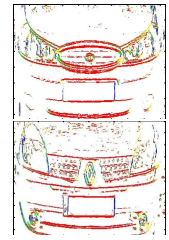


FIG. 12 – Exemples d'extraction de contours orientés.

par la figure 11 qui représente la moyenne des taux individuels selon le nombre d'images d'apprentissage considérées par modèle. Il en faut un certain nombre (>5) pour le processus de création de modèles, filtre efficacement les contours dus aux bruits ou aux reflets sur la carrosserie. Nous pensons qu'un filtrage intra-image, en plus de celui inter-images proposé, améliorerait nos résultats.

5 Conclusion

Cette communication a présenté une méthode de reconnaissance multi-classes du type de véhicule basé sur une représentation en points de contours orientés. Une fonction discriminante combine des scores obtenus de trois types de classificateurs basés sur les votes et une erreur en distance. Cette fonction nous permet de constituer un vecteur caractéristique afin de classer chaque image test selon la classe de son plus proche voisin dans l'espace de votes. Un taux de reconnaissance supérieur à 90% est obtenu sur des images prises en conditions réelles. Le système est robuste vis-à-vis des occlusions partielles de l'image. Nos travaux futurs s'attacheront à mieux filtrer les contours introduits par le bruit ou les reflets sur la carrosserie. Une autre voie de recherche consisterait à dessiner un arbre de décision [3, 4].

Références

- [1] 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.
- [2] MP. Dubuisson and A. Jain. A modified hausdorff distance for object matching. In *ICPR*, volume A, pages 566–569, 1994.
- [3] D.M. Gravila. A pedestrian detection from a moving vehicle. In *European Conference on Computer Vision*, pages 37–49, 2000.
- [4] R. Isukapalli and A.M. Elgammal. Learning policies for efficiently identifying objects of many classes. In *ICPR*, pages III : 356–361, 2006.
- [5] P.A. Negri, X. Clady, M. Milgram, and R. Poulénard. An oriented-contour point based voting algorithm for vehicle type classification. In *ICPR*, 2006.
- [6] V.S. Petrovic and T.F. Cootes. Analysis of features for rigid structure vehicle type recognition. In *BMVC*, volume 2, pages 587–596, 2004.
- [7] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection : A review. *PAMI*, 28(5) :694–711, May 2006.
- [8] C. J. Van Rijsbergen. *Information Retrieval*. Butterworths, London, Second Edition, 1979.