Reconnaissance par vision du type d'un véhicule automobile

P. Negri[1], X. Clady[1], M. Milgram[1], R. Poulenard[2]
[1]Institut Systèmes Intelligents et Robotique - Groupe PRC
Université Pierre et Marie Curie-Paris 6
[2]LPREditor, Montpellier, France
pablo.negri@lisif.jussieu.fr

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, limitée à sa calandre, nous en construisons une représentation à base de points de contour orientés. La classification est réalisée essentiellement en se fondant sur des algorithmes de votes. L'utilisation d'algorithmes de votes permet au système d'être robuste aux données manquantes ou erronées de la représentation. Un taux de reconnaissance de 93% est obtenu sur une base d'images prises en conditions réelles composée de 20 classes de type de véhicules. De plus, une caractérisation et une analyse du fonctionnement du système vis-à-vis de ses différents paramètres est proposée.

Mots-clés : Reconnaissance des formes, vision, classification multi-classes, méthode de votes.

1 INTRODUCTION

L'étude décrite dans cette communication est vouée à la reconnaissance du type (constructeur et modèle) des véhicules qui se présentent, par exemple, à l'entrée d'un parking public ou privé, à partir d'une image fournie par une caméra. Les deux tâches possibles du système sont l'identification du type à partir d'une base de données ou la vérification du type de véhicule (que celui à la sortie soit le même qu'à l'entrée). Nous rappelons que dans les problèmes d'identification, le système détermine l'identité à partir d'une base de données d'individus connus, tandis que dans des problèmes d'authentification, le système doit confirmer ou rejeter l'identité réclamée de l'entrée du système (qui peut être un visage [Zhao, 2003], une empreinte digitale, ou un véhicule,...). Les applications sont évidentes : la surveillance du parking ou le télépaiement.

Dans une situation réelle d'application, les images aux entrées de parking sont essentiellement cadrées de manière à rendre bien visibles les plaques d'immatriculation (cf. fig. 1). Aussi pour reconnaître le type des véhicules, nous allons utiliser leur image des calandres avant (que nous appelerons prototype). Par ailleurs, nous pouvons constater la présence d'une barrière masquant une partie de la calandre et ceci à une position (vis-à-vis de la calandre) que nous pouvons qualifier d'aléatoire puisqu'elle dépends de la distance à laquelle s'arrête le véhicule. 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. En effet, celles-ci sont très sensibles aux occlusions et aux changements de points de vue. De plus, pour ce type de représentation, un vecteur d'apparence de grande dimension est souvent requis, ce qui rends les algorithmes de classification utilisés très gourmands en mémoire et en temps de calcul. Nous avons donc opté pour une méthode basée sur des primitives.

FIG. 1 – Deux exemples de véhicules qui se présentent à l'entrée d'un parking. La barrière occulte une partie de la calandre.

Les calandres de différentes voitures ont des caractéristiques très variées d'un modèle à l'autre. Cependant, chaque modèle de véhicule a un gabarit constant, défini par le constructeur. Ceci rend les contours des calandres relativement constants d'une image d'un vehicule à celle d'un autre du même type. Afin de mieux discriminer ceux-ci, nous ajoutons à l'information du point de contour l'orientation du gradient en ce point.

Dans la suite de cet article, afin de situer nos travaux par rapport à l'existant, nous allons en premier lieu discuter des applications et des systèmes de reconnaissance par vision dans le contexte automobile. Ensuite, nous citerons plusieurs travaux en reconnaissance des formes utilisant aussi des représentations à base de contours orientés. Ceci nous permettra d'introduire la manière dont nous avons construit notre modèle de représentation de chaque type de véhicules à reconnaître, ainsi que notre fonction de classification. Cette fonction combine une mesure de distance et des scores des méthodes de votes. Plusieurs résultats, démontrant l'efficacité de notre méthode et caractérisant son fonctionnement en fonction de ses différents paramètres, seront exposés. Enfin, la conclusion de cet article sera l'occasion de proposer quelques perspectives

pour des travaux futurs.

2 CONTEXTE DE L'ÉTUDE

Reconnaissance de véhicules par vision. Les systèmes de vision pour la reconnaissance de véhicules s'insèrent 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 se présentant au devant du véhicule équipé. Elles doivent permettre le calcul des distances relatives et en informer le conducteur, afin de faire respecter les distances de sécurité et de prévenir les accidents. En second lieu, la télésurveillance des autoroutes et autres voies de circulation permet la mesure des flots de véhicules [Douret, 2004] et donc une meilleure gestion du traffic routier, et aussi de prévenir les services (urgences, police,...) concernés en cas d'incidents ou d'accidents. Enfin il existe des systèmes ayant pour rôle d'identifier les véhicules se présentant aux entrées ou sorties de zones télésurveillées.
La première de ces applications ne requiert qu'un système permettant de classer une zone d'intérêt (ROI - Region Of Interest) selon deux classes [Sun, 2006] : est-ce un véhicule ou non ? Les véhicules sont souvent localisés soit dans l'image ou soit dans l'espace 3D (utilisant une *bounding box*). La seconde peut utiliser en plus un modèle géométrique pour classer le véhicule en catégories : véhicules de tourisme, véhicules utilitaires, véhicules lourds... via des modèles paramétriques ou déformables, 2D ou 3D [Han, 2005, Ferryman, 1995, Dubuisson, 1996].
Enfin la troisième application, dédiée au contrôle d'accès, se base généralement sur la reconnaissance de la plaque d'immatriculation. Cet identifiant est en théorie suffisant pour identifier 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 fausse. Nous pensons qu'adjoindre à ce type de système, un système de reconnaissance du type de véhicule ne peut qu'accroître l'efficacité et les capacités d'un tel système. En effet, l'information du type de véhicule peut confirmer ou infirmer celle fournie par la reconnaissance de la plaque. Les cas suspects ou indécis seraient alors signalés à une personne compétente (gardien de parking, police,...).
A notre connaissance, seuls Petrovic et Cootes[Petrovic, 2004] ont abordé le problème de la reconnaissance automatique du type de véhicules par vision. Leur objectif est la création d'une étude comparative sur la représentation d'objets à structure rigide pour leur reconnaissance multi-classes. Ils ont obtenus divers résultats en appliquant différentes méthodes d'extraction de *features* sur une base d'images prises pour des véhicules en stationnement, avec la même prise de vue et différents éclairages. Une région d'intérêt dans l'image du véhicule, cadrée sur la calandre avant de celui-ci, est obtenue en référence aux coins de la plaque minéralogique. Ils utilisent deux types de distances : la distance euclidienne et un opérateur multiplicatif (défini par $d = 1 - f_1 f_2$, où f_i représente le vecteur d'apparence). Les meilleurs résultats sont obtenus pour les représentations à base de gradients (en conservant de manière implicite l'information d'orientation de celui-ci). Ceci s'explique par le fait que ce type de représentations est relativement indépendant de la couleur du véhicule et du contraste. Cela est souhaitable étant donné la grande variabilité des conditions des prises de vues des images.
Ces résultats et cette conclusion nous ont confortés dans le choix d'une représentation à base de contours orientés. Nous écartons cependant l'approche via apparence de Petrovic et Cootes du fait qu'ils n'ont pas tenus compte de la présence potentielle d'occlusion (due à une barrière). Nous avons donc sélectionné et développé une approche robuste à des informations manquantes ou erronées. En traitement d'images et vision par ordinateur, une catégorie de méthodes a démontré leur efficacité vis-à-vis d'une telle difficulté : les méthodes à base de votes. La méthode que nous allons décrire par la suite est donc essentiellement une méthode de vote utilisant pour votants les pixels de contour orientés. Mais avant, nous allons dresser un panorama de méthodes utilisant ce type de représentation afin d'y situer notre méthode.

Représentations à base de contours orientés. Parmi les méthodes utilisant ce type des représentations à contours orientés pour la reconnaissance d'objets, il faut distinguer celles qui les utilisent pour caractériser la forme globale de l'objet ou un ensemble de formes simples appartenant à l'objet, de celles qui les utilisent pour caractériser localement des points ou des zones d'intérêts sur l'objet.
Dans la première catégorie, nous pouvons citer les travaux d'Olson [Olson, 1997] qui, afin de modéliser des cibles de petites tailles aux bords irréguliers, représente les objets par leurs contours avec une orientation locale liée à chacun des pixels (direction du gradient, normale ou tangente). Une version de la mesure de Hausdorff qui incorpore l'information de la position et de l'orientation est employée pour localiser les potentielles positions de la cible dans l'image. Plus récemment, Jurie et al.[Jurie, 2004] ont développé une méthode originale de détection de formes d'intérêts (correspondant à des objets appris ; appartenant à une seule classe) en cherchant les convexités les plus remarquables du contour d'une forme. Cela peut se voir comme la détermination des cercles contenus dans la forme de l'objet. Chacun de ces cercles reçoit un poids dépendant de la quantité de points de contours qui sont proches de sa limite. Ce poids est calculé à partir de deux fonctions de coût : un terme d'énergie du gradient tangentiel et un terme d'entropie. Cette méthode de détection est robuste aux changements d'échelle, à des rotations et à des occlusions et repose essentiellement sur une décision selon un score de votes.
Dans la seconde catégorie, Hond [Hond, 1997] emploie les orientations du gradient pour la reconnaissance des visages. Le secteur d'identification est subdivisé en régions ou fenêtres. Un histogramme est produit pour chaque

fenêtre enregistrant le profil d'orientation du gradient. Cootes [Cootes, 2001] reprend son exemple mais, plutôt qu'employer un histogramme, ils calculent une mesure qui représente l'orientation locale en chaque pixel, ainsi qu'une indication de la fiabilité sur l'évaluation de son orientation. En particulier ils emploient une fonction non-linéaire de normalisation qui est choisie afin d'accentuer les bords probables et supprimer les mesures susceptibles d'être du bruit. Un vecteur formé des valeurs normalisées du gradient compose le modèle d'apparence du visage.
A la confluence de ces deux catégories, Belongie et al. [Belongie, 2002] utilisent des descripteurs de forme, nommés *context shapes*, décrivant la distribution du reste de la forme par rapport à un point donné sur le contour. Chercher la correspondance entre deux formes est alors équivalent à trouver pour chaque point sur une forme le point sur l'autre forme qui a un *context shape* similaire. Ils prolongent la correspondance à la forme entière en estimant une transformation qui aligne la carte de *contexts shapes* d'une forme sur l'autre. Dans la même lignée, Carmichael et al.[Carmichael, 2002] présentent une variation des *context shapes* afin de différencier l'objet du fond. Leur classificateur se compose d'un ensemble de noeuds reliés dans une structure d'arbre. Chaque noeud représente l'évaluation d'un *edge probe*, qui peut se penser comme une région gaussienne de variance σ^2 centrée en p (*probe center*). Le *edge probe* mesure la densité des pixels de contour dans le voisinage de p.
Nous situons nos travaux dans la première catégorie : c'est l'ensemble des votants (i.e. des points de contours) qui, d'une certaine façon, caractérise la forme d'objet. De plus, dans notre approche, la caractérisation locale de chaque pixel de contours est réduite au minimum. 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. Par contre, nous allons proposer de pondérer chaque point en fonction de son potentiel discriminatoire. La section qui suit, expose le processus de création des modèles à base de points de contour orientés, et le calcul de cette pondération.

3 CRÉATION DU MODÈLE

Dans cette section, nous décrivons comment nous construisons une représentation unique pour chaque type de véhicules. Nous appellons ici cette représentation, modèle. Nous baptisons Base de Connaissance, la liste des classes que le système est capable de reconnaître. Notre Base de Connaissance est composée de $K = 20$ classes de véhicules.

3.1 Bases d'Images

Nous disposons de deux bases d'images des voitures vues de face : une Base d'Apprentissage (TrB) et une Base de Test (TsB).
Afin de tester notre approche dans des conditions proches de celles industrielles, nous avons réaliser deux protocoles différents pour la collecte des échantillons qui vont constituer nos bases. Ainsi les images de la TrB ne sont pas prises dans les mêmes conditions que celles de la TsB. En effet, dans l'optique d'une application commerciale, il n'est pas envisageable que la TrB soit réalisée sur le site d'implantation du système ; aussi l'apprentissage doit être réalisé off-line sur une TrB indépendante du site (et donc de la TsB).
La Base d'Apprentissage est composée des images prises dans des parkings avec des appareils numériques. La résolution de ces images varie entre 1280x960 pixels et 1296x976 pixels. Les images sont soit en couleur soit en niveaux de gris. Nous avons choisi de se servir de ces images pour construire les modèles des types de véhicules, du fait de leur résolution de qualité supérieure (ce qui leur assure une plus grande portabilité sur les éventuels sites d'implantation du système).
La Base de Test est composée des images prises avec un caméscope numérique. La résolution des images de cette base va de 640x480 pixels à 720x576 pixels. Les images sont prises sous différents points de vue et sous différents éclairages. La première ligne de la figure 2 présente des exemples de la Base d'Apprentissage et la deuxième ligne les exemples de la Base de Test de la même classe.

FIG. 2 – Exemples de la Base d'Apprentissage et de la Base de Test.

En tout, nous avons réuni 173 images d'apprentissage et 480 de test, dont la répartition en 20 classes de type de véhicules est donnée dans le tableau 1.

Type de véhicule	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

TAB. 1 – Répartition de la Base de Connaissance.

3.2 Obtention du prototype

(a) (b)

FIG. 3 – (a) image originale, (b) prototype I.

A partir de l'image originale d'une voiture, nous obtenons une imagette de taille 252x600 où la plaque minéralogique est placée dans une position connue (cf. figure

3) via une transformation affine. Cette transformation est effectuée, que ce soit sur les images de la Base d'Apprentissage pour créer les modèles correspondant aux classes ou sur l'image de la Base de Test qui doit être classée. Nous considérons que le plan de la calandre et celui défini par les 4 coins de la plaque sont confondus. Les coins de la plaque sont obtenus via une méthode développée par LPREditor (pour plus d'informations, voir http ://www.lpreditor.com).

3.3 Contours de pixels orientés

Pour le calcul des contours orientés, nous utilisons des filtres de Sobel de taille 3x3 sur l'image prototype I en niveaux des gris, un pour la direction x, S_x, et l'autre pour la direction y, S_y. Le module du gradient est alors défini par :

$$G(x,y) = |S_x * I(x,y)| + |S_y * I(x,y)|$$

et son orientation par :

$$\phi(x,y) = tan^{-1}(\frac{S_x * I(x,y)}{S_y * I(x,y)})$$

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 $G(x,y)$. La valeur du seuil est obtenue en éliminant 85 % des pixels (les valeurs les plus faibles du module).
Dans la suite, nous nommerons $\mathbf{p} = [x,y]$ les pixels de contour appartenant à $\mathbf{E}_I$, où (x,y) est la position du point. Les orientations de ces pixels sont quantifiées pour qu'ils prennent des valeurs entières entre 0 et $N-1$ (ici N est égal à 4). Pour gérer les cas des voitures du même type mais de différentes couleurs (qui changent la polarité du contour), nous avons utilisé le module π au lieu du module 2π [Cootes, 2001]. Par abus de langage et pour simplifier l'écriture, $\phi(\mathbf{p})$ désignera dans la suite l'orientation quantifiée du gradient de I au point $\mathbf{p}$.

3.4 Création du modèle

Pour chaque type de véhicule, nous allons dans cette section déterminer un modèle unique et représentatif de sa classe à partir de toutes les imagettes 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 (voir le tableau 1).
Le pseudocode contenu dans la figure 4 décrit la procédure employée. Pour chaque classe, nous prenons un couple (i,j) parmi les n prototypes de la classe k. Les matrices de contours orientés $(\mathbf{E}_i, \mathbf{E}_j)$ sont alors évaluées. Nous définissons un accumulateur A_{ij} de taille 600x252xN. Ensuite, en prenant un point $\mathbf{p}_i$ de $\mathbf{E}_i$, on cherche dans $\mathbf{E}_j$ le plus proche $\mathbf{p}_j$ avec la même orientation. En respectant l'orientation du gradient, l'accumulateur A_{ij} est incrémenté au point milieu du segment $\overline{\mathbf{p}_i\mathbf{p}_j}$. Itérativement, ce procédé est répété pour tous les points $\mathbf{p}_i$ de $\mathbf{E}_i$, puis pour tous les couples possibles de l'ensemble de n exemples de la classe k. Enfin, une matrice

Soient n prototypes pour la classe $k \in$ **KnB**
pour i=1,...,n
  obtenir $\mathbf{E}_i$
  pour j=1,...,$n \wedge$ j $\neq$ i
    $A_{ij}(600, 252, N) = 0$
    détermine $\mathbf{E}_j$
    $\forall\, \mathbf{p}_i \in \mathbf{E}_i$, trouver $\mathbf{p}_j \in \mathbf{E}_j$ / $\phi(\mathbf{p}_i) = \phi(\mathbf{p}_j)$
      $\wedge \min_{\forall j} |\mathbf{p}_i\mathbf{p}_j| < t$
    $\mathbf{p}_m(x_m, y_m) = \frac{\overline{\mathbf{p}_i\mathbf{p}_j}}{2}$
    vote +1 pour l'élément du tableau $A_{ij}(x_m, y_m, \phi(\mathbf{p}_i))$
$A^k = \sum_{i,j} A_{ij}$
$\mathbf{M}^k = f_{max}(A^k)$

FIG. 4 – Pseudocode pour la création du modèle.

d'accumulation de la totalité des votes de la classe k est obtenue en faisant l'addition de tous les A_{ij} :

$$A^k = \sum_{i,j \wedge i \neq j} A_{ij}$$

Cet accumulateur distingue les points de contour les plus redondants dans les n prototypes, donc les plus représentatifs de la classe k, selon la quantité de votes. Pour obtenir un modèle unique (composé de points de contour orientés) de la classe k, on cherche itérativement le point $\mathbf{a}_m = [x,y,o]$ le plus voté de A^k, en faisant attention que les points soient distants d'au moins 5 pixels afin que leur distribution soit relativement homogène. Nous conservons la position et l'orientation du maximum $\mathbf{a}_m$ dans une matrice $\mathbf{M}^k$ (voir figure 5). La méthode est réitérée jusqu'à obtention du nombre de points voulu (ici 3500 points).

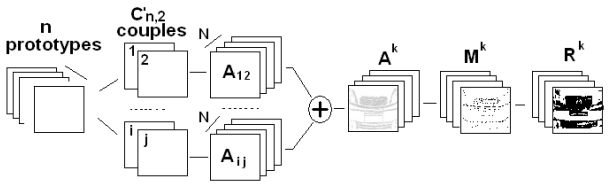


FIG. 5 – Création du modèle pour la classe k.

Quand il existe un seul exemple dans la base d'apprentissage, il n'est plus possible de créer l'accumulateur de votes A^k. Il faut donc utiliser une autre source d'information qui nous estime un poids pour chaque pixel de contour. Dans ce cas, l'accumulateur A^k est remplacé par le module du gradient $G(x,y)$ dans le procédé de génération du modèle.

3.5 Matrices de pondération

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 voisinage d'un point de ce modèle. Les autres sont construits selon ce modèle. Ils requierent donc la construction de cartes de voisinage.
Pour estimer ces 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. Les R_i^k sont regroupées dans une seule matrice R^k de dimension 600x252xN.

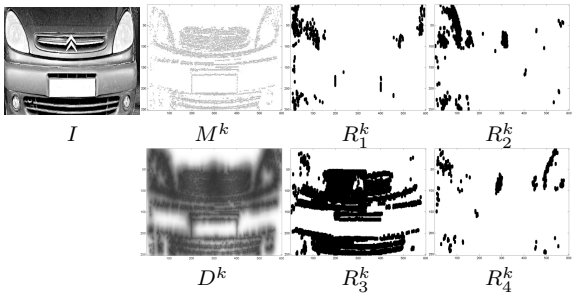


FIG. 6 – $\mathbf{M}^k$ est le modèle de points de contour orientés avec les N orientations superposées. D^k est la carte de distance de Chamfer de M^k. Les R_i^k sont les voisinages après application d'un seuil r pour chaque orientation du gradient i.

Afin d'optimiser la méthode de votes, nous allons calculer un poids w pour chaque point des régions définies dans R^k : les points les plus discriminants du modèle de $\mathbf{p}_m$ par rapport aux autres modèles, vont avoir un vote de poids plus important. Ces poids sont regroupés dans une matrice W_+^k. Les pixels du modèle k qui sont rarement présents dans les autres classes obtiennent des poids plus importants. Des poids faibles sont donnés aux points présents dans la plupart des autres classes. Ceci peut s'exprimer via l'équation suivante :

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

De la même façon que nous récompensons les modèles proches de $\mathbf{t}$ par le vote des points qui tombent dans les voisinages, nous pouvons aussi punir ceux qui ne le sont pas (via un vote des points qui ne tombent pas dans les voisinages). En considérant alors des poids négatifs, nous pouvons construire une matrice de pondération W_-^k :

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

4 CLASSIFICATION

La classification consiste à associer un exemple $\mathbf{t}$ dit de test à une classe k du dictionnaire des classes. Ceci sera réaliser via une fonction de discrimination. Elle se présente comme la combinaison de scores provenant de deux types de classifieurs. Le premier type consiste en trois processus de vote : les votes positifs de $\mathbf{t}$ vers k, les votes négatifs de $\mathbf{t}$ vers k et les votes de chacun des k vers le $\mathbf{t}$. Le deuxième classifieur évalue, pour chaque k, l'erreur en distance de mise en correspondance entre les points de contours orientés des modèles $\mathbf{M}^k$ et les points de $\mathbf{t}$.

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 $\mathbf{E}_t$. Parmi ces points, nous en sélectionnons T qui sont a priori les plus discriminants pour chaque classe. Ceci est réalisé en triant les points de $\mathbf{E}_t$ selon la matrice de pondération W_+^i avec $i = 1, ..., K$. Nous obtenons alors pour l'exemple $\mathbf{t}$, une matrice $\mathbf{P}_t$ de taille 600x252xN et construite telle que $\mathbf{P}_t$ a la valeur 1 dans la position des points sélectionnés et 0 ailleurs.

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 $\mathbf{M}^k$. Le voisinage d'un point d'un modèle $\mathbf{M}^k$ est défini comme un cercle de rayon r. Ceci peut être réalisé simplement en faisant le produit terme à terme 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) :

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

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 Distances de Chamfer pour $\mathbf{E}_t$. Après un seuillage des points qui se trouvent à une distance inférieure à r, nous obtenons la matrice des régions R^t de l'exemple te test. Par la suite, nous prenons les premiers T points de la liste M_k et construisons la matrice P^k. Chaque point de la matrice P^k est pondérée avec W_+^k :

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

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 [Dubuisson, 1994]. Celle-ci s'exprime de la manière suivante :

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

avec :

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

Afin de réaliser la fusion avec les autres scores, nous y appliquons une fonction décroissante pour attribuer un score maximum dans le cas d'une erreur minimale.

Fonction de discrimination Les quatre scores $\{v_+^k, v_-^k, v_+^t, d^k\}$ sont combinés dans la fonction discriminante $g_k(t)$ en une mesure de concordance entre l'exemple de test $\mathbf{t}$ et la classe k.

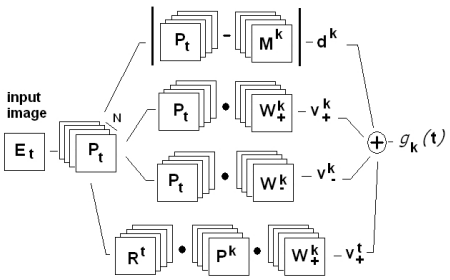


FIG. 7 – Obtention de la fonction de discrimination

Avant la fusion, une pseudo-distance de Mahalanobis est utilisée pour normaliser les valeurs des scores. On calcule la moyenne et l'écart type des votes positifs de $\mathbf{t}$ pour les différentes classes :

$$v_{+t}^{mh} = \frac{v_+^k - \mu_+}{\sigma_+}$$

De la même façon, on normalise les autres scores. La fonction de discrimination est définie alors par :

$$g_k(t) = \alpha_1\, v_{+k}^{mh} + \alpha_2\, v_{-k}^{mh} + \alpha_3\, v_{+t}^{mh} + \alpha_4\, d_k^{mh} \quad (1)$$

Les coefficients α_i peuvent donner des poids différents aux classifieurs. Dans notre système, nous fixons tous les α_i à 0.25.
La classe de l'exemple de test $\mathbf{t}$ est la classe qui obtient le plus grand score pour la fonction de discrimination :

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

5 RÉSULTATS ET CARACTÉRISATION

5.1 Résultats

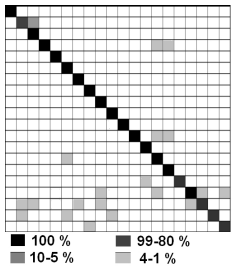
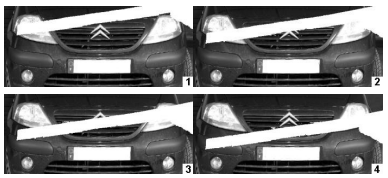


FIG. 8 – Matrice de confusion.

Classifieurs	Taux de reconnaissance
v_+	91.0 %
v_-	85.4 %
v_t	89.5 %
d	82.1 %
fusion	93.7 %

FIG. 9 – Taux de reconnaissance pour les différents classifieurs et la fusion des quatre.

Nous avons appliqué notre algorithme sur les 480 exemples de la Base de Test. Il identifie correctement 93.75 % des exemples[1]. La figure 8 montre les résultats obtenus pour l'ensemble des classes sous la forme d'une matrice de confusion. Plaçons nous, par exemple, dans la quatrième ligne de la matrice de confusion, qui corresponds au modèle Citroën Saxo (voir tab. 1). Du total de 21 exemples de test, 19 ont été classés correctement dans la classe Citroën Saxo. Les deux exemples restants ont été classifiés incorrectement : un exemple comme Clio D et l'autre comme Laguna.
Dans la figure 9, on peut vérifier que le résultat de la fusion des scores est meilleur que les résultats individuels. Les valeurs des α_i pourraient être optimisées dans l'équation (1) via par exemple un processus d'apprentissage. Mais, pour valider un tel algorithme d'apprentissage, nous aurons besoin de plus d'exemples de véhicules.
Un autre test simule la présence de la barrière en quatre positions différentes. La figure 10.(a) montre les différentes positions de celles-ci. Les résultats obtenus pour chaque position de la barrière sont exposés dans la figure 10.(b). Le faible taux de reconnaissance pour la $4^{\text{ème}}$ position peut s'expliquer par le fait que la barrière cache les points proches à la plaque minéralogique : ce sont des points importants pour la classification et sont aussi les plus stables à la transformation affine.



(a)

Position	Taux de reconnaissance
1	91.25 %
2	91.88 %
3	90.92 %
4	88.33 %

(b)

FIG. 10 – (a) les 4 positions de la barrière virtuelle, (b) Taux de reconnaissance.

Enfin, dans un dernier test, nous avons entraîné le système avec un seul exemple par classe dans la TrB. Le taux de reconnaissance alors obtenu est de 85.6%. Ce résultat illustre le fait que la présence de plusieurs exemples pour l'apprentissage permet de filtrer le bruit lors la création des modèles.

5.2 Caractérisation

Dans cette section, nous allons caractériser le comportement de notre système vis-à-vis des variations de ses différents paramètres, qui sont les suivants :
- T : le nombre de points utilisés dans la classification,
- N : la quantité d'orientations,
- r : le rayon du voisinage,
- la largeur de la plaque minéralogique dans le prototype (c'est-à-dire la résolution de celui-ci).

Dans cette étude, nous allons faire varier chacun d'entre eux, en fixant les autres.

Nombre de points de travail La figure 11 montre la variation du taux de reconnaissance en fonction du nombre de points de travail. Les résultats décrits dans la section précédente sont réalisés avec 2000 points. Il

[1] le taux présenté ici corresponds à la mesure de rappel en micro-précision introduite dans [Rijsbergen, 1979] ; les mesures de précision en micro et macro-précision sont respectivement de 95.43% et de 93.87% , celle de rappel en macro-précision est de 94.07%.

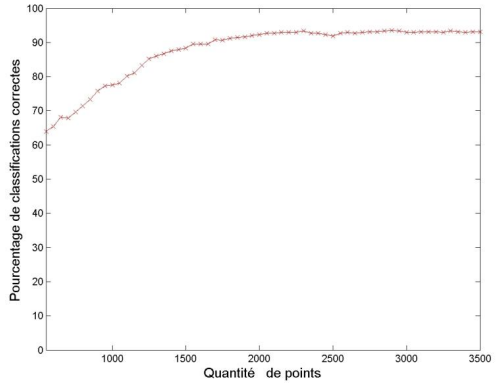


FIG. 11 – Variation du taux de reconnaissnce en fonction du nombre de points de travail.

s'agit d'un compromis entre un pourcentage de classifications correctes élevé et un temps de calcul acceptable. Mais il pourrait être avantageux de travailler en utilisant une quantitée inférieure de points, afin réduire le temps de calcul. Les valeurs des autres paramètres sont $N = 4$ et $r = 5$. La largeur de la plaque est fixée à 200 pixels.

Quantité d'orientations Nous allons faire la classification de la base de test en variant la quantité d'orientations. Les valeurs des autres paramètres sont fixées à $T = 2000$ et à $r = 5$. La largeur de la plaque est égale à 200 pixels. La figure 12 illustre le résultat obtenu. Comme nous pouvons le voir, il faut éviter de prendre un nombre d'orientations impair. En effet, un nombre d'orientations impair place certaines limites entre les classes d'orientation à $0°$, $90°$, $180°$ ou $270°$. Ainsi du fait du bruit, les pixels des contours horizontaux et verticaux auront tendance à osciller entre 2 classes d'orientation possibles. Or ce sont les points les plus nombreux sur un véhicule. Par ailleurs, nous pouvons aussi remarquer que si on prends un nombre d'orientations égal à 6 les résultats ne sont que légèrement meilleurs qu'avec un nombre d'orientations à 4. Le choix du nombre d'orientations se fera donc plus en fonction du temps de calcul et de la ressource mémoire disponible.

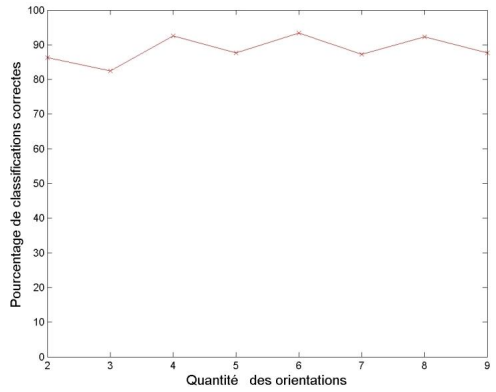


FIG. 12 – Variation du taux de reconnaissance en fonction de la quantité d'orientations.

Rayon de voisinage Dans ce test, nous allons faire varier le seuil appliqué à la distance de Chamfer pour obtenir la matrice M^k du modèle k.
Nous utilisons soit 550 points soit 2300 points et toujours 4 orientations. Le résultat est illustré dans la figure 13. D'une part, nous constatons que si on prends

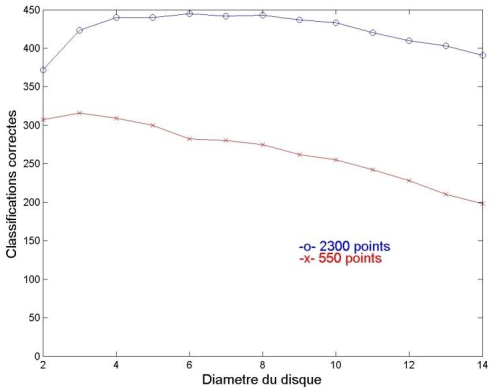


FIG. 13 – Variation du taux de reconnaissance en fonction du rayon de voisinage.

un rayon de voisinage trop grand, le nombre de classifications correctes diminue. En effet, il y a alors un recouvrement important entre les zones de voisinages des points de contours. D'autre part, plus le nombre de points est important, plus large est la plage des valeurs que l'on peut choisir sans influence notable sur le résultat final.

Variation de la taille du prototype Dans ce test, nous faisons varier la taille du prototype. Cette taille est fonction de la largeur de la plaque minéralogique. Dans la figure 15, cette largeur varie de 100 à 270 pixels. Les autres paramètres sont $T = 2000$, $N = 4$, $r = 5$.

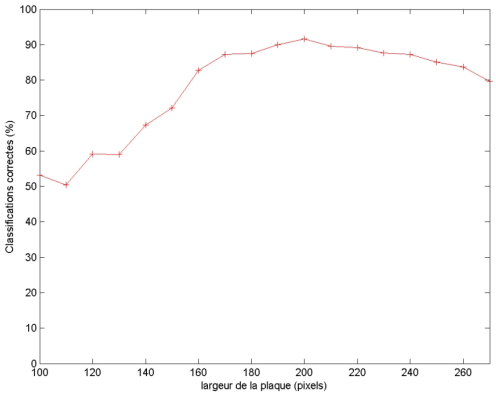


FIG. 14 – Variation du taux de reconnaissance en fonction de la largeur de la plaque d'immatriculation.

Pour analyser cette courbe, nous avons représenter dans la figure 15 l'histogramme représentant les quantités d'exemples de test en fonction de la taille de la plaque minéralogique dans les images originales.
Dans cet histogramme, nous pouvons observer que la majorité des plaques minéralogiques a une largeur inférieure à 200 pixels. En dessous de cette valeur, les images sont

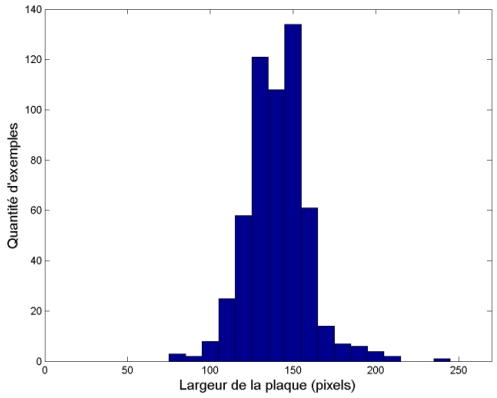


FIG. 15 – Histogramme représentant la taille des plaques minéralogiques dans les images originales pour les exemples de test.

sous-échantillonnées et nous perdons de l'information. En deça de 200 pixels, nous obtenons alors de faibles valeurs pour le taux de classifications correctes (fig. 14). Au-delà de 200 pixels, nous observons un effet de dégradation dans la classification dû à l'approximation bilinéaire employée lors de la Transformation Affine (voir sec. 3.2), qui est connue pour rendre floue une image agrandie et a donc une conséquence directe sur l'extraction des points de contour.

6 CONCLUSION

Cette article a présenté un système de votes pour une application de reconnaissance multi-classes du type de véhicule basé sur une représentation en points de contours orientés. Une fonction discriminante combine les scores obtenus de trois types de classifieurs basés sur les votes et une erreur en distance. Nous avons testé cette méthode sur un ensemble de 480 images de véhicules vus de face (prises en situation réelle) séparé en 20 classes. Notre taux de reconnaissance est supérieur à 93 %. Les résultats ont montré aussi que le système est robuste vis-à-vis des occlusions partielles de l'image.
Nos travaux futurs s'attacheront à augmenter le nombre de classes de la Base de Connaissance. Cette opération peut réduire le taux de reconnaissance : un test préliminaire sur 30 classes nous fournit un taux à 89.2 %. Nous espérons compenser cet effet en introduisant plus de classifieurs dans la fonction de discrimination. Une autre voie de recherche consiste à dessiner un arbre de décision hiérarchique [Gravila, 2000] où l'on pourra considérer des sous-classes contenant plusieurs types de véhicule.

BIBLIOGRAPHIE

[Belongie, 2002] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4) :509–522, 2002.

[Carmichael, 2002] O. Carmichael and M. Hebert. Object recognition by a cascade of edge probes. In *British Machine Vision Conference*, volume 1, pages 103–112, Cardiff, UK, September 2002. British Machine Vision Association.

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

[Douret, 2004] 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.

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

[Dubuisson, 1996] 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.

[Ferryman, 1995] 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.

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

[Han, 2005] 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.

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

[Jurie, 2004] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Computer Vision and Pattern Recognition*, Washington, DC, June-July 2004.

[Olson, 1997] 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.

[Petrovic, 2004] 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.

[Rijsbergen, 1979] C. J. Van Rijsbergen, Information Retrieval. *Butterworths, London, Second Edition*, 1979.

[Sun, 2006] 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.

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