

Reconnaissance par vision du type d'un véhicule automobile

Vehicle make and model identification using vision system

X. Clady¹, P. Negri¹, M. Milgram¹ et R. Poulenard²

¹ Université Pierre et Marie Curie-Paris 6, Institut des Systèmes Intelligents et de Robotique – CNRS,
4, place Jussieu, 75005 Paris, France

² LPRÉditor, Montpellier, France
xavier.clady@upmc.fr

Manuscrit reçu le 11 mars 2008

Résumé et mots clés

Cet article présente un système de reconnaissance du type (constructeur, modèle) de véhicules par vision. À 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. Nous avons donc construit une fonction de discrimination qui combine 3 votes et une distance, et agit comme une mesure de similarité entre chaque modèle et l'image de véhicule testée. Deux stratégies de décision ont été testées. La première associe à une image de calandre avant du véhicule, le modèle qui a obtenu la valeur la plus importante en sortie de la fonction. Une seconde stratégie regroupe toutes les sorties en un vecteur. La décision est alors prise *via* un algorithme de plus proche voisin dans un espace dit de votes. Avec la première stratégie, 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. Cependant ce taux chute à 80 % lorsque le nombre de modèles passe à 50 classes. Pour le même nombre de classes, la seconde stratégie permet d'obtenir un taux supérieur à 90 %.

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

Abstract and key words

Vehicle make and model identification

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 distances from the equipped vehicle. Secondly, road monitoring measures traffic flow, notifies the health services in case of an accident or informs the police in case of a driving fault. Finally, Vehicle based access control systems for buildings or outdoor sites have to authenticate incoming (or outgoing) cars.

Rather than these two systems, the third one uses often only the recognition of a small part of vehicle: the license plate. It is enough to identify a vehicle, but in practice the vision based number plate recognition system can provide a wrong information, due to a poor image quality or a fake plate. Combining such systems with others process dedicated to identify vehicle type (brand and model) the authentication can be increased in robustness. This paper addresses the identification problem of a vehicle type from a vehicle greyscale frontal image: the input of the system is an unknown vehicle class, that the system has to determine from a data base. This multiclass recognition system is developed using

the oriented-contour pixels to represent each vehicle class. The system analyses a vehicle frontal view identifying the instance as the most similar model class in the data base. The classification is based on voting process and a Euclidean edge distance. The algorithm have to deal with partial occlusions. Tollgates hide a part of the vehicle and making inadequate the appearance-based methods. In spite of tollgate presence, our system doesn't have to change the training base or apply time-consuming reconstruction process.

Model Creation

During the initial phase of our algorithm, we produce a model for all the **50** vehicle types classes composing the list of classes the system is capable to recognize is called Knowledge Base (**KnB**). All ours experiments have been carried out on the Training Base (**TrB**) and on the Test Base (**TsB**). The **TrB** samples (291 images) are used to produce the oriented-contour point models of the vehicle classes. While the **TsB** samples (830 images) are utilized to evaluate the performance of the classification system. These databases are composed of frontal vehicle views, captured in different car parks, under different light conditions and different points of view.

We create a canonical rear-viewed vehicle image I from the four corner points of the license plate. The image templates are called prototypes and in the present work are $600 * 252$ pixels (rows * columns). A license plate recognition system provides the corners of the vehicle license plate. The Sobel operator is used to compute the gradient's magnitude and orientation of the greyscale prototype I ($|\nabla g_I|, \phi_I$). An oriented-contours points matrix \mathbf{E}_I is obtained using an histogram based threshold process.

Each class in the **KnB** is represented by n prototypes in the **TrB**. This quantity n varies from class to class, having some defined with one prototype only. Superposing the n prototypes of the class k , we find an array of the redundant oriented-contour points \mathbf{M}^k . This feature array of Oriented-Contour based points models this class in the **KnB**.

Two weighted regions arrays W_+^k and W_-^k will be created for each class k . Each pixel point has a weight related to the discrimination power of the corresponding oriented-contour points. Pixel points rarely present in the others classes obtain highest weights. We give lowest weights to the points present in the majority of the Knowledge Base classes.

W_-^k gives a negative weight to the points of the other models. Pixel points that are present in most of the other classes obtain highest weight values. In the other hand, pixel points present in few classes get lowest weight values.

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

Classification

This section develops the methods to classify the samples providing from the $\mathcal{T}f\mathcal{B}$ using the models \mathcal{M}_k . Obtaining the image prototype of the sample t from the Test Base, we calculate the oriented-contour points matrix \mathbf{E}_t .

The matching scores

Three kind of vote based mathching scores (positive, negative and class votes) for each class k are defined.

The Positive votes 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. 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 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 . For voting 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 . 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 . 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 .

Classification Strategies

We have developed two strategies for classification. The first combines the scores in a discriminant function. The second creates voting vector spaces from the scores : the decision is based on a nearest-neighbor process.

First Strategy: 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 a fusion of scores:

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

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 class label k is determined from: $k = G(t) = \text{ArgMax}\{g_1(t), \dots, g_K(t)\}$.

Second Strategy: Voting Spaces

We construct vector spaces with the results from the voting process. We define:

- $v(t) = [v_{+k}^{mh}, v_{-k}^{mh}, v_{+t}^{mh}, d_k^{mh}]_{k=1..K}$ as a vector in a $200 (= 4 * K)$ dimension space, called Ω_{wf} ($wf =$ without fusion).

- $v^{PCAX}(t) = [v_{+k}^{mh}, v_{-k}^{mh}, v_{+t}^{mh}, d_k^{mh}]_{k=1..K}^{PCAX}$ as a vector in a X dimension space, called Ω_{wf}^{PCAX} , (with a Principal Component Analysis pre-stage).

- $g(t) = [g_k(t)]_{k=1..K}$ as a vector in a $50 (= K)$ dimension space, called Ω_f ($f =$ with fusion).

In these spaces, given the test sample t , its class label is determined as the nearest-neighbor class. It needs reference samples. We use a cross-validation process : the test database is decomposed in two equal parts. The first is used as references. The second is used for the test.

Results and Conclusions

With the first strategy, the system correctly identifies 80,2% of 830 test samples. The mean of the recognition rates per class is 69,4%. The second strategy obtains better results (in mean, with 100 randomly different repartitions). In the first space, Ω_{wf} , we obtain 93,1% for the correctly identification rate (83,5% for the mean of the recognition rates per class). In a second space, Ω_{wf}^{PCAX50} , we obtain 86,2% for the correctly identification rate (78,8% for the mean of the recognition rates per class). In the last space Ω_f , we obtain 90,6% for the correctly identification rate (86,4% for the mean of the recognition rates per class). Another test simulates the presence of a tollgate at four different locations. The better recognition are obtained if the virtual tollgate hides the upper part of the images: a lot of noise points are extracted from this part. These points perturb the recognition system. They are filtered if the number of images used in the model creation is sufficient (> 5).

The results presented allow to conclude that the second strategy obtains better results, particularly if the fusion scores are used.

Pattern Recognition, Vision, Multiclass recognition, Voting Algorithm.

1. Introduction

L'étude décrite dans cet article 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 [27], 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 appellerons 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 dépend 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 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 rend les algorithmes de classification utilisés très gourmands en mémoire et en temps de calcul. Nous avons donc opté pour une méthode utilisant des primitives, de type points de contour orientés.

En effet, 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 véhicule 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. Ensuite nous aborderons comment à partir de ce modèle, nous avons construit une fonction de discrimination. Cette fonction combine une mesure de distance et des scores des méthodes de votes. Elle permet de mesurer la similarité d'une image de calandre avant avec chaque type de véhicule (chaque modèle). Pour la décision (l'association d'un type à l'image), nous avons développé deux stratégies différentes. La première consiste à associer l'image au type ayant obtenu la plus grande valeur en sortie de la fonction de discrimination. La seconde consiste à regrouper toutes les valeurs en sortie de la fonction en un vecteur ; nous proposons donc de représenter chaque type par un ou plusieurs échantillons dans un espace dit de votes. Un processus de plus proche voisin nous permet alors de décider à quel modèle appartient l'image testée. 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 [15] se présentant au devant du véhicule équipé. Elles doivent permettre le calcul des distances relatives et en



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

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 [4] et donc une meilleure gestion du trafic 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 [23]: est-ce un véhicule ou non ? Les véhicules sont souvent localisés soit dans l'image 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 [10, 8, 7].

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 (cf. figure 2). Les cas suspects ou indécis seraient alors signalés à une personne compétente (gardien de parking, police,...).

À notre connaissance, peu de travaux se sont intéressés récemment à une problématique similaire. En 2004, Petrovic et Cootes [19, 20] ont abordé pour la première fois 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 une reconnaissance multi-classes. Ils ont obtenu 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). Les auteurs de [25] reprennent ces travaux en étudiant l'apport d'une analyse en composantes discriminantes (*Linear Discriminant Analysis*). Daniel T. Munroe *et al.* [21] a étudié les techniques de classification via apprentissage appliquées à des descripteurs issus d'un détecteur de contour de type Canny. L. Dlagnekov [5] utilise des descripteurs de type *Scale Invariant Feature Transform* (SIFT [17]) pour corrélérer des points d'intérêt. Zafar *et al.* [26] ont proposé un algorithme similaire. David A. Torres [24] a étendu le travail de Dlagnekov en utilisant des descripteurs à base de segment de contour, au lieu des SIFT. Dans leur travail [13], Kazemi *et al.* ont étudié l'utilisation de la Transformée de Fourier (*Fourier Transform*), des Transformées en Ondelettes ou en Courbelettes Discrètes (*Discrete Wavelet Transform* et *Discrete Curvelet Transform*). D'une manière générale, tous ces travaux utilisent des descripteurs à base de gradients ou de contours. 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. Leurs résultats et cette conclusion nous ont confortés dans le choix d'une représentation à base de contours orientés. Nous avons particulièrement orienté nos recherches de sorte à développer une approche robuste à des informations manquantes ou erronées du fait de la présence potentielle d'une barrière occultant une partie de l'image. 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

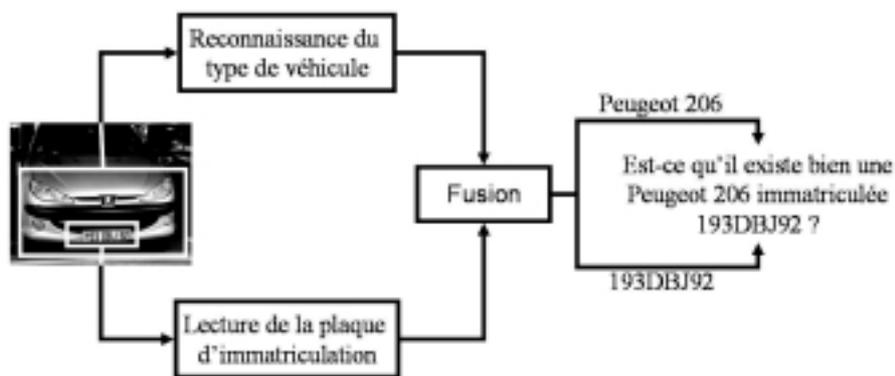


Figure 2. Système d'identification de véhicule fusionnant la reconnaissance de son type et la lecture automatique de sa plaque d'immatriculation.

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 de 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 [18] 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.* [12] 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 [11] 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 [3] 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.

À la confluence de ces deux catégories, Belongie *et al.* [1] 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 *context shapes* d'une forme sur l'autre. Dans la même lignée, Carmichael *et al.* [2] 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 en 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 appelons 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 classes de véhicules; nous notons cet ensemble de classes $\Omega(K)$.

3.1. Obtention du prototype

À 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 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>).

3.2. 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:

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

Figure 3. (a) image originale, (b) prototype I .

et son orientation par :

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

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'elles 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 [3]. 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} .

3.3. 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 $600 \times 252 \times N$. 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 . Pour chaque classe, il y a donc C_n^2 combinaisons de couples possibles. Enfin, une matrice d'accumulation de la totalité des votes de la classe k est obtenue en faisant l'addition de tous les A_{ij} :

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

Soient n prototypes pour la classe $k \in \mathbf{KnB}$

pour $i = 1, \dots, n$

obtenir \mathbf{E}_i

pour $j = 1, \dots, n$ et $j \neq i$

$A_{ij}(600, 252, N) = 0$

détermine \mathbf{E}_j

$\forall \mathbf{p}_i \in \mathbf{E}_i$, trouver $\mathbf{p}_j \in \mathbf{E}_j / \phi(\mathbf{p}_i) = \phi(\mathbf{p}_j)$

et $\min_{\forall j} |\mathbf{p}_i \mathbf{p}_j| < t$

$\mathbf{p}_m(x_m, y_m) = \frac{\overline{\mathbf{p}_i \mathbf{p}_j}}{2}$

vote +1 pour l'élément du tableau

$A_{ij}(x_m, y_m, \phi(\mathbf{p}_i))$

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

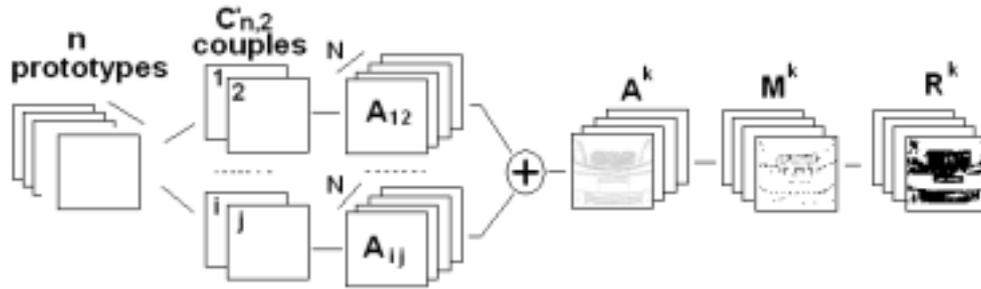
$\mathbf{M}^k = f_{\max}(A^k)$

Figure 4. Pseudocode pour la création du modèle.

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). 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.

3.4. Matrices de pondération

Notre algorithme de classification utilise essentiellement 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


 Figure 5. Création du modèle pour la classe k .

un modèle s'il tombe dans le voisinage d'un point de ce modèle. Les autres sont construits selon ce modèle. Il requière 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 $600 \times 252 \times N$.

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 réalisé *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 $i = 1, \dots, K$. Nous obtenons alors pour l'exemple \mathbf{t} , 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.

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 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 en 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 .

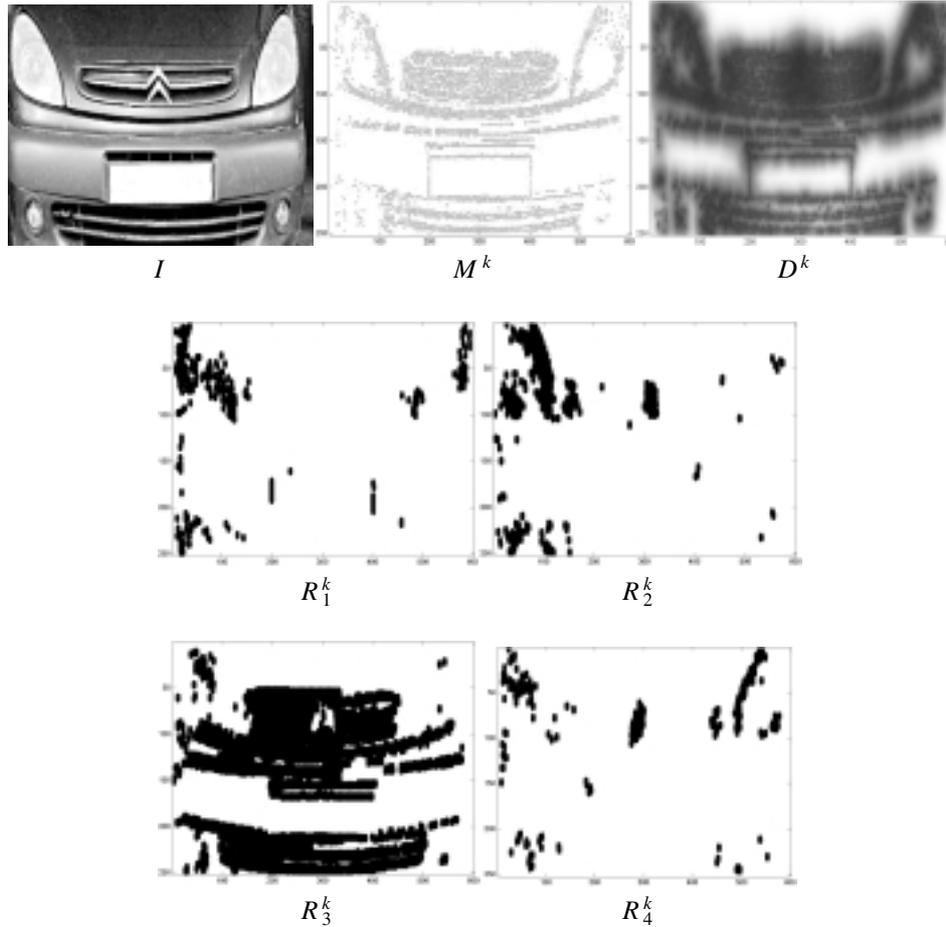


Figure 6. \mathbf{M}^k est le modèle de points de contour orientés avec les N orientations superposées. D^k est la carte de distance de Chamfer de \mathbf{M}^k . Les R_i^k sont les voisinages après application d'un seuil r pour chaque orientation du gradient i .

Elle est connue comme la distance de Hausdorff modifiée [6]. 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 :

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

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. L'erreur en distance alors obtenue est notée d^k .

Fonction de discrimination Une première stratégie consiste à combiner ces scores $\{v_+^k, v_-^k, v_+^t, d^k\}$ dans la fonction discriminante $g_k(t)$ en une mesure de concordance entre l'exemple de test \mathbf{t} et la classe k .

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 :

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

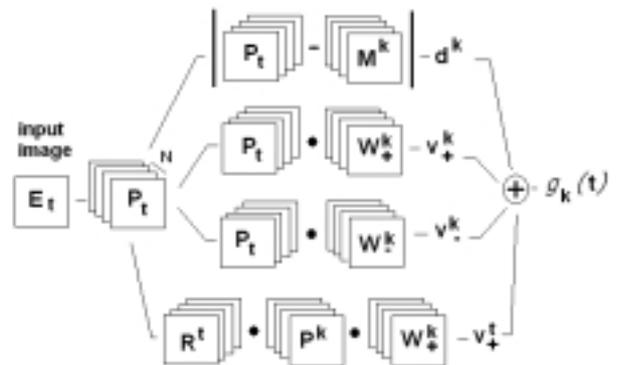


Figure 7. Obtention de la fonction de discrimination.

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 classificateurs. 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 :

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

Espaces de vote 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.

5. Résultats et Caractérisation

Dans cette section, nous allons commenter plusieurs résultats obtenus à l'aide du système décrit précédemment. Ces résultats ont été essentiellement réalisés sur 2 Bases de Connaissances différentes : une composée de 20 modèles (*cf.* tab 1) et une autre de 50 modèles. La première servira essentiellement à illustrer le fonctionnement du classifieur en n'utilisant que la première stratégie. La seconde nous permettra d'analyser plus finement l'apport de la seconde stratégie. Enfin, nous discuterons des perspectives possibles afin d'améliorer les performances du système.

5.1. Bases d'Images

Nous disposons de deux bases d'images des voitures en niveaux de gris et 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 réalisé 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 1280×960 pixels et 1296×976 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. La Base de Test est composée des images prises avec un caméscope numérique. La résolution des images de cette base va de 640×480 pixels à 720×576 pixels. Les images sont prises sous différents points de vue et sous différents éclairages. La première ligne de la figure 8 présente des exemples de la Base d'Apprentissage et la deuxième ligne les exemples de la Base de Test de la même classe.

5.2. Résultats de la première stratégie

Nous avons appliqué notre algorithme sur les 480 exemples de la Base de Test. Il identifie correctement 93.75 % des exemples ; ce qui correspond au taux de reconnaissance en micro-précision [22]. Le taux en macro-précision, *i.e.* la moyenne des taux par classe, est de 94.07 %. La figure 9 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 correspond 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.



Figure 8. Exemples de la Base d'Apprentissage et de la Base de Test.

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

Tableau 1. Répartition de la Base de Connaissance avec 20 modèles.

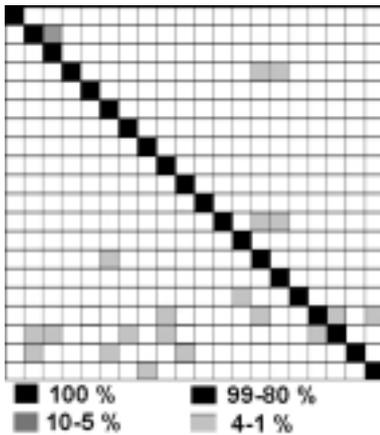


Figure 9. Matrice de confusion

Classifieurs	Taux de reconnaissance
v_+	91.0 %
v_-	85.4 %
v_t	89.5 %
d	82.1 %
fusion	97.7 %

Figure 10. Taux de reconnaissance pour les différents classifieurs et la fusion des quatre.

Dans la figure 10, 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 aurions besoin de plus d'exemples de véhicules.

5.3. Caractérisation de la première stratégie

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),
- K : le nombre de modèles.

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 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 quantité inférieure de points, afin de 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.

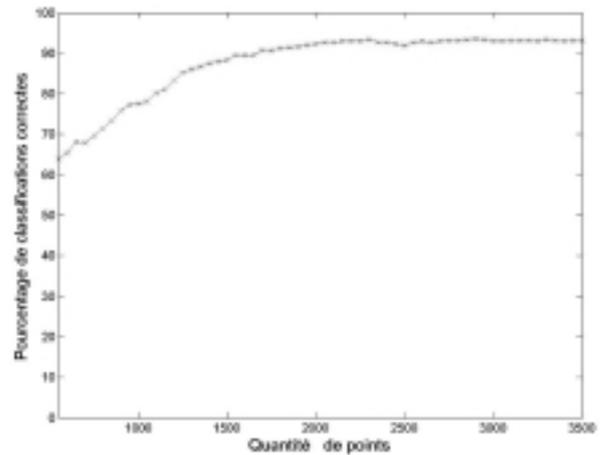


Figure 11. Variation du taux de reconnaissance en fonction du nombre de points de travail.

Quantité d'orientations Nous allons réaliser 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.

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

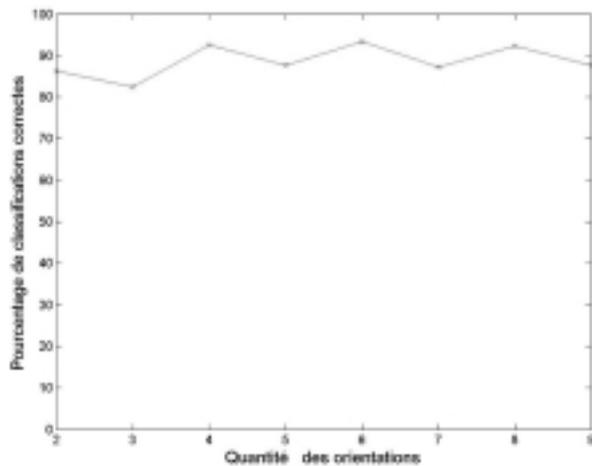


Figure 12. Variation du taux de reconnaissance en fonction de la quantité d'orientations.

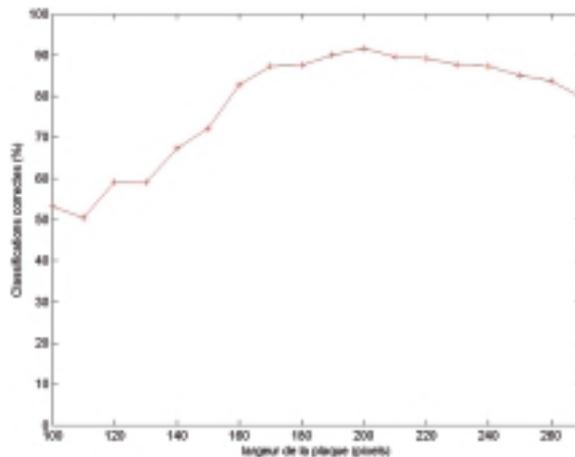


Figure 14. Variation du taux de reconnaissance en fonction de la largeur de la plaque d'immatriculation.

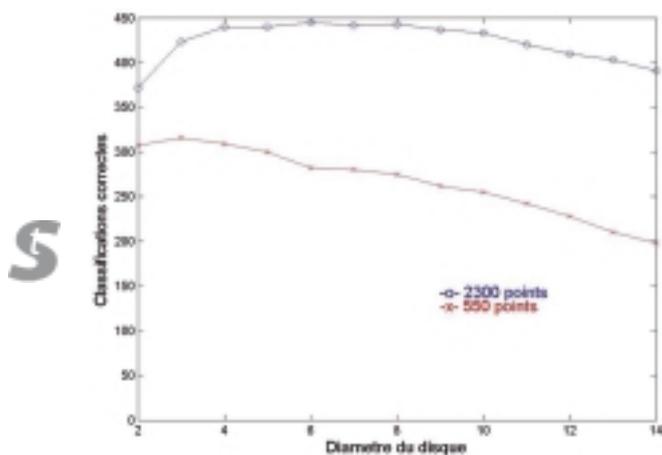


Figure 13. Variation du taux de reconnaissance en fonction du rayon de voisinage.

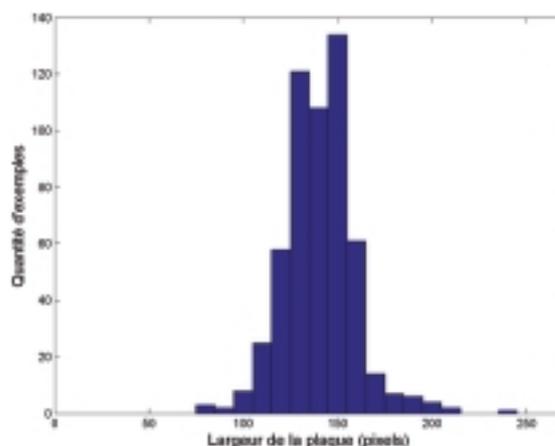


Figure 15. Histogramme représentant la taille des plaques minéralogiques dans les images originales pour les exemples de test.

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 prend 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 $T = 2000$, $N = 4$, $r = 5$.

Pour analyser cette courbe, nous avons représenté 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 sous-échantillonnées et nous perdons de l'information. En deçà 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. 3.1), qui est connue pour rendre floue une image agrandie et a donc une conséquence directe sur l'extraction des points de contour.

Variation du nombre de modèles Nous avons vu que le système de reconnaissance obtient des résultats intéressants avec la première stratégie. Cependant, nous constatons aussi une forte baisse de ces performances lorsque le nombre de modèles augmente dans notre Base de Connaissances, comme l'indique aussi la figure 16. Comme nous allons le voir dans la section

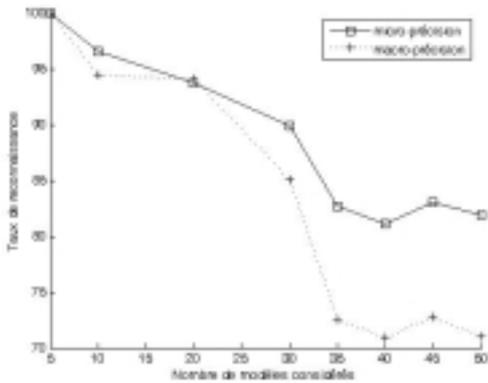


Figure 16. Taux de reconnaissance en fonction de la taille K de la Base de Connaissances.

suiuante, la seconde stratégie permet d'améliorer nettement les performances.

5.4. Résultats de la seconde stratégie

Avec la première stratégie, utilisée avec la seconde Base de Connaissances composée de 50 modèles, le système identifie seulement 83,2 % des 830 exemples de la Base de Test (la Base d'Apprentissage est alors composée de 291 exemples); le taux de reconnaissance en macro-précision est de 71,4 %.

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

- pour un premier espace Ω_{sf} , 93,1 % en micro-précision et 83,5 % en macro-précision.
- pour un second espace Ω_{sf}^{ACP50} , 86,2 % en micro-précision et 78,8 % en macro-précision.
- pour un troisième espace Ω_{af} , 90,6 % en micro-précision et 86,4 % en macro-précision.

La figure 17 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 18, il faut conserver une dimension $X > 100$ pour obtenir des taux similaires à ceux obtenus avec fusion. De plus, remarquons que la fusion peut être encore améliorée.

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 19. Bien entendu, plus il y a d'échantillons de références, plus long est le processus de plus proche voisin; pour compenser cela, il est important de travailler dans l'espace de plus faible dimension possible et donnant les meilleurs résultats possibles (comme l'espace avec fusion).

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

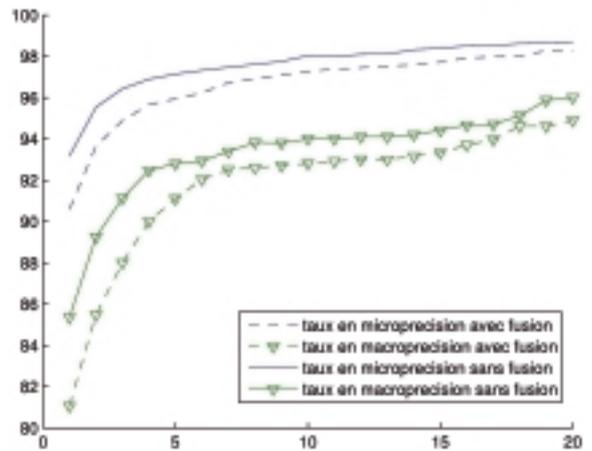


Figure 17. Courbes CMC; les courbes en lignes pleines représentent les taux obtenus dans l'espace Ω_{sf} et les courbes en pointillés ceux dans l'espace Ω_{af} .

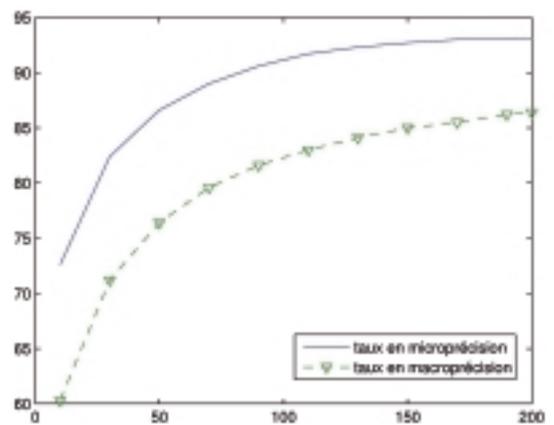


Figure 18. Évolution des taux de reconnaissance sur Ω_{sf}^{ACPX} en fonction de la dimension X .

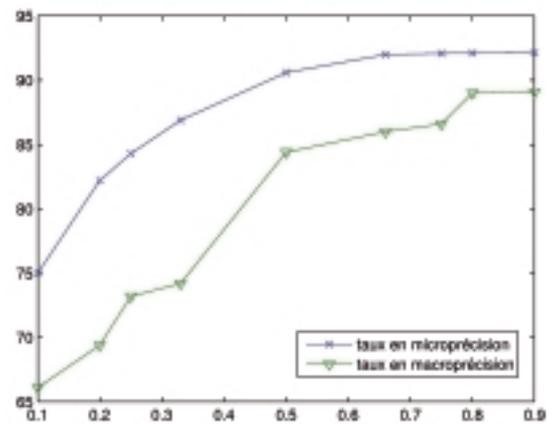


Figure 19. Évolution des taux de reconnaissance (dans Ω_{af}) en fonction de la proportion d'échantillons.



Position	première stratégie	Ω_{sf}	Ω_{sf}^{ACP50}	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 %

Figure 20. 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. 22) afin vérifier la robustesse de notre méthode aux occlusions.

À 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. 22) et qui altèrent le processus de reconnaissance. Ceci est corroboré par la figure 21 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.

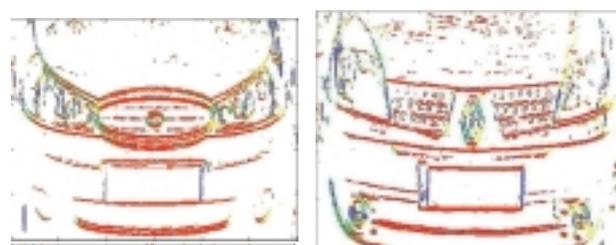


Figure 22. Exemples d'extraction de contours orientés.

découlent directement de celles de ces classifieurs. D'ailleurs, ceci explique que les taux de reconnaissance en présence d'occlusion sont globalement plus élevés dans l'espace de votes avec fusion que sans fusion. Le fait de combiner les 4 classifieurs dans la même fonction de discrimination permet d'obtenir des éléments des vecteurs représentatifs de chaque modèle plus robustes aux occlusions (comme c'est déjà le cas sans occlusions, cf. figure 10).

Ainsi globalement, pour la seconde stratégie comme pour la première, la fusion des 4 classifieurs est plus intéressante car elle permet d'être plus robuste et de travailler dans un espace à plus faible dimension.

6. Conclusions

Cet article a présenté un système de votes pour une application de reconnaissance multi-classes du type de véhicule utilisant 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. Cette fonction nous permet de constituer un vecteur caractéristique afin de classer chaque image test selon la classe de son plus proche voisin. 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

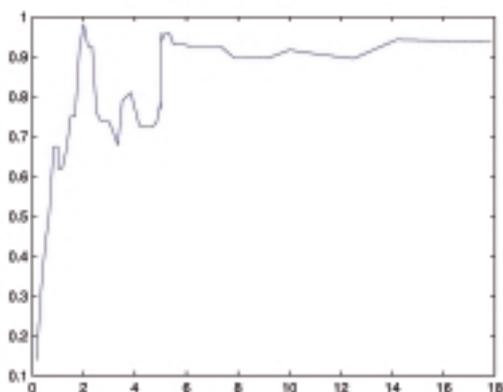


Figure 21. Évolution moyenne des taux individuels en fonction du nombre d'images considérées lors de la création des modèles.

Remarquons aussi que les taux de reconnaissance de la première stratégie peuvent être directement corrélés avec ceux de la seconde. Lorsque ces taux sont élevés avec la première, ils le sont également dans les espaces de vote, et *vice versa*. En fait, les deux stratégies de classification utilisent les mêmes classifieurs (votes et distance) à l'origine ; aussi leurs robustesses

voie de recherche consiste à dessiner un arbre de décision hiérarchique [9, 14]. Par ailleurs, rappelons que ce système serait destiné à améliorer l'identification de véhicules dans des applications de contrôle d'accès. Dans l'avenir, nous comptons donc combiner notre approche avec une lecture automatique de plaque d'immatriculation, tel que cela est exprimé dans la figure 2. Nous pensons qu'un tel processus de fusion d'informations viendra encore améliorer les résultats obtenus dans cet article : l'information de la lecture automatique de la plaque venant corriger éventuellement certaines ambiguïtés dans la reconnaissance de type, et réciproquement. Il reste d'autres aspects à parfaire. Notamment notre méthode nécessite une détection de la plaque d'immatriculation relativement précise, ce qui limite sa généralité. En effet, bien qu'efficace, elle n'est applicable que pour des objets à structures rigides et possédant un élément géométriquement normalisé. Une autre approche consisterait à développer une représentation robuste aux changements de perspective, voire indépendante de la géométrie. Un espace de représentation, inspiré de celui des sacs de mots visuels [16], pourrait être par exemple envisagé.

Références

- [1] 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.
- [2] 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.
- [3] T. COOTES and C. TAYLOR, On representing edge structure for model matching. In *Computer Vision and Pattern Recognition*, volume 1, pages 1114-1119, Hawaii, USA, December 2001.
- [4] 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.
- [5] L. DLAGNEKOV, Video-based car surveillance : License plate make and model recognition. *Master Thesis. University of California at San Diego*, 2005.
- [6] MP. DUBUISSON and A. JAIN, A modified hausdorff distance for object matching. In *International Conference on Pattern Recognition*, volume A, pages 566-569, 1994.
- [7] 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.
- [8] 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.
- [9] D.M. GRAVILA, A pedestrian detection from a moving vehicle. In *European Conference on Computer Vision*, pages 37-49, 2000.
- [10] 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.
- [11] D. HOND and L. SPACEK, Distinctive descriptions for face processing. In *Machine Vision Conference*, University of Essex, UK, 1997.
- [12] 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.
- [13] F. M. KAZEMI, S. SAMADI, H. R. POOREZA and M. R. AKBARZADEH-T, Vehicle Recognition Based on Fourier, Wavelets and Curvelet Transforms – a Comparative study. In *IEEE International Conference on Information Technology, INTNG'07*, 2007.
- [14] P. NEGRI, X. CLADY, M. MILGRAM and R. POULENARD, Système de reconnaissance multiclassées de type de véhicule. In *Orasis. Obernai, France*, 2007.
- [15] P. A. NEGRI, X. CLADY, S. MUHAMMAD HANIF and L. PREVOST, A cascade of boosted generative and discriminative classifiers for vehicle detection. In *EURASIP Journal on Advances in Signal Processing (Special Issue : Machine Learning in Image Processing)*, 12 Pages, 2008.
- [16] E. NOWAK and F. JURIE, Learning visual similarity measures for comparing never seen objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [17] D. G. LOWE, Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, 60(2):91-110, 2004.
- [18] 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.
- [19] 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.
- [20] V.S. PETROVIC and T.F. COOTES, Vehicle Type Recognition with Match Refinement. In *International Conference on Pattern Recognition*, volume 3, pages 95-98, 2004.
- [21] D.T. MUNROE and M.G. MADDEN, Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images. *AICS*, 2005.
- [22] C. J. Van RIJSBERGEN, Information Retrieval. *Butterworths, London, Second Edition*, 1979.
- [23] 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.
- [24] D.A. TORRES, More Local Structure for Make-Model Recognition. *Technical Rapport, University of California, San Diego*, 2007.
- [25] I. ZAFAR, E.A. EDIRISINGHE, B.S. ACAR and H.E. BEZ, Two Dimensional Statistical Linear Discriminant Analysis for Real-Time Robust Vehicle Type Recognition. *SPIE*, vol.6496, 2007.
- [26] I. ZAFAR, B.S. ACAR and E.A. EDIRISINGHE, Vehicle Make and Model Identification using Scale Invariant Transforms. *IATED*, 2007.
- [27] W. ZHAO, R. CHELLAPPA, P. J. PHILLIPS, and A. ROSENFELD, Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399-458, 2003.



Xavier Clady

Ingénieur diplômé en Acquisition et Traitement d'Images à l'École Nationale Supérieure de Physique de Strasbourg (France) en 1998, Xavier Clady effectue son DEA Image à l'Université Jean Monet de Saint-Etienne avant de rejoindre le Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique (LASMEA) de l'Université Blaise Pascal à Clermont-Ferrant en 1999. Le 6 janvier 2003, il y obtient son doctorat en Vision pour la Robotique. En septembre 2003, il devient Maître de Conférences au sein de l'Université Pierre et Marie Curie (Paris VI). Il y effectue ses recherches au sein de l'Institut des Systèmes Intelligents et de Robotique (ISIR). Elles concernent le traitement d'images et la reconnaissance de formes, appliqués à la détection et au suivi d'objets, de gestes ou de personnes.



Pablo Negri

Ingénieur diplômé en Electronique à l'Université de La Plata (Argentine), Pablo Negri rejoint en 2003 l'Institut des Systèmes Intelligents et de Robotique, de l'Université Pierre et Marie Curie (Paris VI) où il réalise sa thèse de doctorat depuis décembre 2004. Ses sujets de recherche concernent le traitement d'images et la reconnaissance de formes.



M. Milgram

M. Milgram est né en 1948 à Paris, il a obtenu l'Agrégation de Mathématiques (1971) puis une Thèse de 3^{ème} cycle en Recherche Opérationnelle à l'UTC (1975) et une Thèse d'État à l'UTC sur les Réseaux d'Automates Probabilistes (1981). Il a été Assistant (1975) puis Maître-Assistant (1977) à l'UTC et Professeur à l'ENSEA (Cergy) (1983) puis Professeur à Paris VI (1986). Il a travaillé depuis 1983 en Vision pour la Robotique et en Reconnaissance de Formes (Écriture manuscrite) puis, depuis 2002, sur la localisation et l'analyse des visages. M. Milgram a encadré environ 30 thèses et publié un ouvrage (1993 : Reconnaissance des Formes, Armand Colin). Il a participé à plus de 20 contrats industriels.



Raphael Poulenard

Raphael Poulenard a fait des études de mathématiques jusqu'à l'obtention du diplôme de l'agrégation en 1996. Puis il s'oriente vers le traitement d'images et la reconnaissance des formes en autodidacte. Sensibilisé aux problèmes d'insécurité et de pollution posés par le trafic routier, il décide de se lancer dans les applications du traitement d'image au service de l'identification de véhicules. En 1999 il se tourne naturellement vers la lecture automatique des plaques d'immatriculation. En 2003, il est co-fondateur de la SARL LprEditor qui propose un moteur de lectures de plaques et quelques applications qui en découle. Aujourd'hui encore, Raphael Poulenard poursuit ses investigations dans ce domaine.