

Détection de route et suivi de véhicules par vision pour l'ACC

Road detection and vehicles tracking by vision for ACC

par **R. AUFRÈRE, F. MARMOITON, R. CHAPUIS, F. COLLANGE, J.P. DÉRUTIN**

Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique (LASMEA) UMR 6602 du CNRS Université Blaise-Pascal de Clermont-Ferrand
F-63177 AUBIERE Cedex
aufriere@lasmea.univ-bpclermont.fr

résumé et mots clés

Cet article présente, dans un premier temps, un procédé permettant de détecter la voie de circulation d'un véhicule par vision monoculaire embarquée. Ce processus de détection est basé sur une mise à jour récursive d'un modèle statistique de la voie obtenu par une phase d'apprentissage. Après avoir localisé la voie, un algorithme de reconstruction détermine la position du véhicule dans sa voie de circulation et le profil 3D de la route. Par la suite, nous nous intéressons à la détection, la localisation et surtout le suivi des véhicules situés à l'avant et équipés de marques visuelles afin de déterminer avec précision la position et la vitesse relative de ces véhicules. La combinaison de ces différentes informations permet de déterminer le véhicule le plus dangereux. La description détaillée de chacune des étapes de notre algorithme est suivie d'exemples significatifs.

Détection de route, reconstruction 3D, suivi de véhicules.

abstract and key words

This article deals with a process designed first to extract the lane of vehicle by on-board monocular vision. This detection process is based upon a recursive updating of a statistical model of the lane obtained by a training phase. Once the lane has been located, a reconstruction algorithm computes the vehicle location on its lane and the 3D shape of the road. Thereafter, we are focus at the detection, location and tracking of front vehicles equipped with specific visual markers in order to achieve an accurate determination of the location and speed of these vehicles. Merging these various informations allows to point out the most dangerous obstacle. Each of these three processes is detailed significant examples are provide.

Road detection, 3D reconstruction, vehicles tracking.

1. introduction

Depuis quelques années, de nombreux chercheurs travaillent sur des projets concernant l'aide à la conduite automobile par vision artificielle. Des projets comme par exemple « la route automatisée » ou « PATH »¹ ont pour but le développement de systèmes capables de réguler la vitesse de véhicules routiers (contrôle longitudinal) ou de contrôler la position des véhicules par rapport aux voies de circulation (contrôle latéral).

Le LASMEA est impliqué dans de tels projets et des travaux sont engagés afin de mettre en œuvre, sur notre véhicule expérimental (VELAC²), un dispositif de type régulateur de vitesse adaptatif (ou ACC : Adaptive Cruise Control [15]). Ce type de système a pour but de réguler la vitesse du véhicule afin de respecter les distances de sécurité avec les véhicules situés à l'avant.

Notre but est de concevoir d'une part un algorithme capable d'estimer la localisation d'un véhicule sur sa voie de circulation par vision (surveillance de trajectoire, guidage, ...). D'autre part, nous souhaitons identifier parmi l'ensemble des véhicules obstacles, ceux qui se trouvent dans la même voie de circulation que celle du véhicule porteur.

Notre approche repose ainsi sur deux modules : le premier permet la localisation des bords de la voie de circulation dans l'image et détermine l'attitude 3D du profil de la voie. Le deuxième module permet la localisation 3D, le suivi et la caractérisation des obstacles situés à l'avant du véhicule.

Cet article est organisé de la façon suivante : les deux prochaines parties détaillent les deux principaux modules de notre algorithme, la combinaison d'information entre ces différents modules est présentée dans la quatrième partie. Les principaux résultats obtenus sur chaque partie sont ensuite présentés et un bilan est ensuite tiré.

2. détection et reconstruction 3D de la route

2.1. détection de route

Les travaux existants à ce jour, pour la détection des bords de route, portent pour la plupart sur les routes marquées (autoroutes) et utilisent une caméra monochrome, suffisante pour

détecter le contraste entre les bandes et la route. De nombreux travaux utilisent un modèle pour caractériser la route dans l'image. Ce modèle permet, après une mise à jour des paramètres, de localiser le véhicule. Il peut également être utilisé pour tester la cohérence des primitives [6], [9], [16].

Les modèles considérant la courbure de la route comme constante [6], [11], sont simples à implanter mais introduisent des erreurs de modélisation de la route si celle-ci n'est pas plane ou si le véhicule présente un mouvement de tangage ou de roulis.

Les clothoïdes permettent également de modéliser les courbures verticale et horizontale de la route [9]. Néanmoins, cette modélisation est plus complexe et dans le cas de fausses détections, des problèmes apparaissent car le nombre de degrés de liberté du modèle est élevé. Pour cette méthode, certains paramètres sont de plus supposés connus et constants (largeur de la route ou angle d'inclinaison de la caméra). D'une manière générale, les principaux défauts de ces différentes approches sont la faible précision si le modèle est simple ou une grande sensibilité au bruit et une difficulté de mise à jour si le modèle est plus complexe.

Notre méthode (décrite plus précisément dans [1]) est basée sur une reconnaissance récurrente des bords de route guidée par un modèle probabiliste défini par un vecteur de paramètres \underline{X} et sa matrice de covariance \mathbf{C}_X . La reconnaissance se fait dans des zones d'intérêt optimales déduites de la mise à jour courante du modèle. La route est considérée reconnue lorsqu'un nombre minimum p_f de primitives est atteint.

modèle

Le modèle statistique est composé de $2n$ paramètres ($n = 10$ dans notre cas). Ce nombre peut varier suivant l'application (autoroutes, routes nationales, routes de campagnes, etc).

Le modèle est représenté par un vecteur \underline{X} et sa matrice de covariance \mathbf{C}_X (éq. (1)).

$$\underline{X} = (u_{1g}, \dots, u_{ng}, u_{1d}, \dots, u_{nd})^t \quad (1)$$

($u_{1g} \dots u_{ng}$) et ($u_{1d} \dots u_{nd}$) représentent les abscisses respectives des bords gauche et droit de la route dans l'image pour des ordonnées différentes v_i ($i \in [1, n]$) fixées une fois pour toutes.

La matrice \mathbf{C}_X permet de définir les variations des coordonnées u_{ig} et u_{id} . Le vecteur \underline{X} et la matrice \mathbf{C}_X définissant le modèle initial sont établis par une phase d'apprentissage.

apprentissage

Le but de cette procédure est de fournir les valeurs initiales $\underline{X}(0)$ et $\mathbf{C}_X(0)$ (connaissance *a priori* sur la position des bords de la route dans l'image). Ce calcul est réalisé au préalable et n'entre pas dans le processus de reconnaissance. Nous utilisons, pour cet apprentissage, un modèle de l'ensemble route/véhicule [4] tenant compte des hypothèses suivantes : route localement plane, courbure localement constante, ...

Ce modèle donne les abscisses u_{ig}, u_{id} des bords de la route dans l'image selon les ordonnées v_i , les paramètres de localisation du véhicule (x_0 : position latérale sur la voie, ψ : angle de cap), de

1. Partners for Advanced Transit and Highways.

2. Véhicule Expérimental du Lasmea pour l'Aide à la Conduite.

la caméra (α : angle d'inclinaison) et de la géométrie de la route (L : largeur, C_l : courbure latérale) en considérant l'angle de roulis comme négligeable car son effet se compense avec le terme x_0 :

$$u_{ig} = e_u \left[-\frac{e_v z_0}{2(v_i - e_v \alpha)} C_l + \frac{v_i - e_v \alpha}{e_v z_0} \left(x_0 + \frac{L}{2} \right) - \psi \right] \quad (2)$$

$$u_{id} = e_u \left[-\frac{e_v z_0}{2(v_i - e_v \alpha)} C_l + \frac{v_i - e_v \alpha}{e_v z_0} \left(x_0 - \frac{L}{2} \right) - \psi \right] \quad (3)$$

- $e_u = f/d_u$ et $e_v = f/d_v$,
- f : focale de la caméra,
- d_u, d_v : largeur et hauteur d'un pixel,
- z_0 : hauteur de la caméra.

Le vecteur moyen $\underline{X}(0)$ et sa covariance $C_X(0)$ sont calculés autour d'un très grand nombre N de réalisations de (2) et (3) avec les paramètres aléatoires : $C_l, x_0, \psi, \alpha, L$.

Nous considérons que ces paramètres suivent une loi gaussienne autour de valeurs fixées préalablement. Ces valeurs sont établies à partir de connaissances initiales moyennes sur l'état du véhicule, la route et la caméra.

A partir de toutes les coordonnées u_{ig} et u_{id} des N configurations, nous calculons un vecteur moyen \underline{X} et une matrice de covariance C_X qui définissent par la suite le modèle initial.

Ce modèle ($\underline{X}(0), C_X(0)$) est suffisant pour initialiser le processus de reconnaissance sur autoroute mais peut être affiné au fur et à mesure des images et s'auto-adapter à tout type de route.

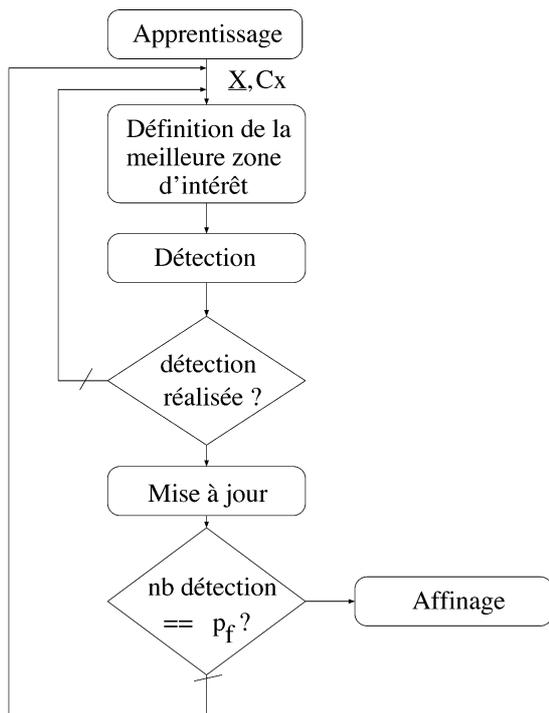


Figure 1. – Organigramme de la méthode de reconnaissance.

Les importantes variations autorisées autour des paramètres de localisation ($C_l, x_0, \psi, \alpha, L$) permettent de valider la phase d'initialisation dans des conditions diverses et variées (virage par exemple).

procédure de recherche

La figure 1 présente l'organigramme de la méthode de reconnaissance.

A la suite de la phase d'apprentissage, la meilleure zone d'intérêt est positionnée de façon optimale dans l'image. Dans cette zone, un segment est détecté pour une profondeur d'analyse $p = 0$. Cette procédure est itérative : le modèle est remis à jour, une nouvelle zone est définie et l'algorithme travaille maintenant à la profondeur $p + 1$. L'algorithme considère que les bords de la route sont trouvés dans l'image si p_f ($p_f = 10$) détections cohérentes au sens du modèle ont pu être isolées avec au moins 2 détections sur chaque bord afin d'assurer un maximum de cohérence entre les deux bords détectés. Une étape d'affinage permet, par la suite, d'améliorer le résultat de la recherche.

zones d'intérêt

Le plus petit intervalle de confiance I permet de caractériser la meilleure zone d'intérêt. Cet intervalle est défini en extrayant de la diagonale de C_X la variance σ_i^2 la plus faible.

$$I = [u_i - \sigma_i, u_i + \sigma_i]$$

La zone d'intérêt est alors définie par le trapèze de sommets (voir figure 2) :

$$P_1 : (u_i - \sigma_i, v_i), P_2 : (u_i + \sigma_i, v_i), P_3 : (u_{i+1} - \sigma_{i+1}, v_{i+1}), P_4 : (u_{i+1} + \sigma_{i+1}, v_{i+1}).$$

v_i : ordonnée du point d'abscisse u_i .

La figure 2 montre une image avec le modèle initial issu de la phase d'apprentissage ainsi que la meilleure zone d'intérêt.

détection

Cette phase consiste à détecter une primitive, dans la zone d'intérêt définie précédemment.



Figure 2. – Image avec le modèle initial et la première zone de recherche.

Pour chaque ligne, à l'intérieur de la zone d'intérêt, le point ayant le maximum positif ou négatif du gradient est localisé tout en sachant que dans une image donnée, le signe du gradient est conservé pour un même bord. Avec l'ensemble des points constitué, un segment de droite est ajusté par la méthode des moindres carrés médians qui prend en compte une contrainte de pente déduite de C_X . Ce détecteur essaye en fait de trouver un segment conforme au besoin du modèle. Cette détection fournit une mesure $\underline{y} = (u_i, u_{i+1})^t$.

Il se peut qu'aucun segment ne soit détecté dans la zone d'intérêt. Dans ce cas, la détection est tentée dans une nouvelle zone d'intérêt définie par la deuxième plus petite variance de la matrice de covariance.

Si aucun segment n'est détecté dans l'ensemble des zones non testées à la profondeur d'itération donnée p , l'algorithme abandonne cette branche et revient à la profondeur précédente : $p - 1$ et réitère le processus sur cette nouvelle branche de l'arbre de recherche.

mise à jour

La phase de détection donne pour une profondeur d'analyse p une mesure $\underline{y}(p) = (\hat{u}_i(p), \hat{u}_{i+1}(p))^t$ permettant d'affiner le modèle ($\underline{X}(p - 1), C_X(p - 1)$) pour la profondeur actuelle p .

Cette phase consiste à calculer un nouveau modèle ($\underline{X}(p), C_X(p)$) déduit de l'observation et de l'état précédent ($\underline{X}(p - 1)$ et $C_X(p - 1)$) de la manière suivante :

$$\begin{cases} \underline{X}(p) = \underline{X}(p - 1) + \mathbf{K}[\underline{y}(p) - \underline{x}(p - 1)] \\ C_X(p) = C_X(p - 1) - \mathbf{K}\mathbf{H}C_X(p - 1) \end{cases}$$

avec :

– $\mathbf{K} = C_X(p - 1)\mathbf{H}^t [\mathbf{H}C_X(p - 1)\mathbf{H}^t + \mathbf{R}]^{-1}$

– $\underline{x} = (u_i, u_{i+1})^t$,

– \mathbf{H} tel que $\underline{x} = \mathbf{H}\underline{X}$,

– \mathbf{R} : matrice de covariance sur l'erreur de détection \underline{u} , telle que $\underline{y} = (\hat{u}_i, \hat{u}_{i+1})^t = \underline{x} + \underline{u}$, fixée a priori à 5 pixels (ici $\mathbf{R} = 5^2 \cdot \mathbf{I}$ où \mathbf{I} représente la matrice identité de taille 2×2).

évolution de l'algorithme

La procédure de recherche est relancée de manière récursive pour chaque nouvelle itération et s'arrête lorsque la profondeur d'analyse $p = p_f$ est atteinte.

La figure 3 présente un exemple d'évolution de l'algorithme sur une image. La zone de recherche des bandes dans l'image, caractérisée par la matrice de covariance $C_X(0)$ issue de la phase d'apprentissage, est représentée en figure 3-a. La figure 3-b

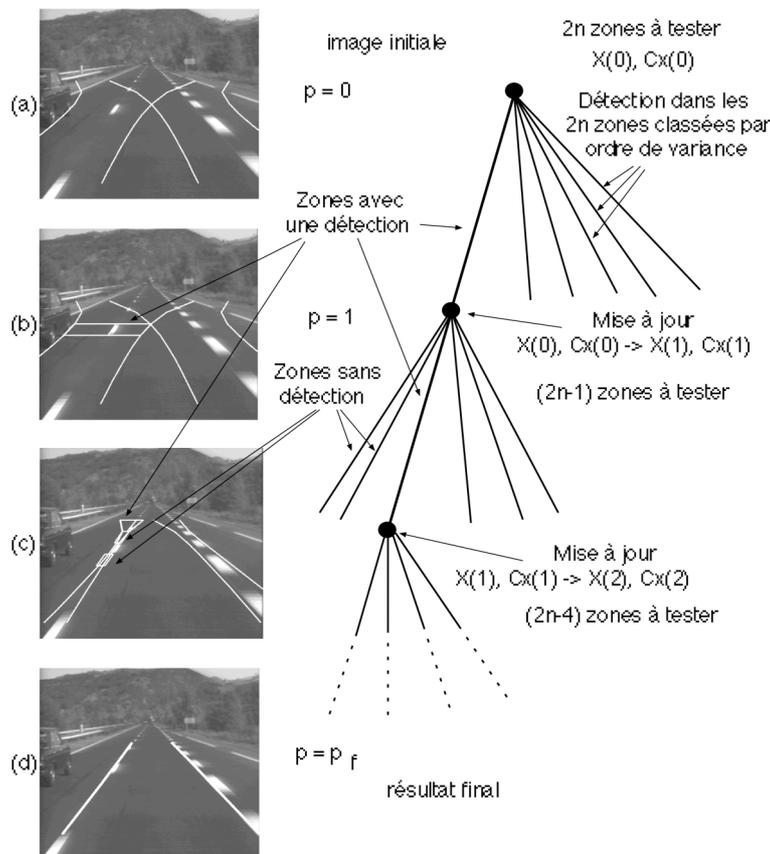


Figure 3. – Évolution de la méthode pour une image.

montre la première zone d'intérêt déduite de $C_X(0)$ et la détection faite à l'intérieur. Grâce à cette détection, $\underline{X}(0)$ et $C_X(0)$ seront mis à jour afin de limiter la zone de recherche dans l'itération suivante ($p = 1$). La figure 3-c montre la mise à jour $\underline{X}(1)$ et $C_X(1)$ de $\underline{X}(0)$ et $C_X(0)$. Dans cette configuration, la première et la deuxième zone d'analyse ne fournissent aucune détection. Une détection a cependant pu être réalisée dans la zone suivante et le vecteur $\underline{X}(1)$ et la matrice $C_X(1)$ peuvent être remis à jour. Le processus est réitéré jusqu'à $p = p_f$. La figure 3-d montre le résultat final (18 itérations de l'algorithme). A la fin de la procédure de recherche, p_f détections cohérentes ont été effectuées, ainsi les bords de la route sont supposés retrouvés. Néanmoins, afin d'obtenir une précision optimale nous relançons le processus dans les zones non testées à partir du vecteur $\underline{X}(p_f)$ et de la matrice $C_X(p_f)$.

Le résultat final de notre algorithme correspond au vecteur \underline{X} issu de la plus grande profondeur d'analyse.

Si p_f détections cohérentes ne sont pas retrouvées, l'algorithme a parcouru l'ensemble de l'arbre de recherche et considère la route comme non retrouvée dans l'image (pour un manque d'information par exemple). Le processus fournit tout de même le modèle ayant conduit au plus grand nombre de détections cohérentes. Le nombre d'itérations de l'algorithme peut généralement être limité à 200, principalement en phase de suivi (convergence plus rapide vers la meilleure solution).

résultats de détection

Les images utilisées ont une taille de 512×512 . Le vecteur modèle est composé de 20 paramètres ce qui correspond à 18 zones d'analyse (9 pour chacun des bords). La caméra utilisée présente une focale $f = 12$ mm et est placée à une hauteur $z_0 = 1.2$ m.

Les figures suivantes présentent des résultats relatifs à ce premier module à la fois sur routes marquées et non marquées. La première image montre l'image originale de la route et la seconde représente le résultat final de l'algorithme de détection des bords de route (correspondant au meilleur modèle trouvé).

La figure 4 présente les résultats sur routes marquées (cas avec occultations). La figure 5 montre des résultats dans des cas particuliers (ombres, surbrillance, imperfection du marquage). La figure 6 présente les résultats sur des routes sans marquage. Le détecteur bas niveau est le même que sur les images précédentes. Le nombre d'itérations varie de 18 à 1000 et dépend de la complexité de l'image. Le temps de calcul généré par le module de détection de route varie de 70 ms à 200 ms sur une station de travail HP J282.

2.2. reconstruction 3D de la route

La méthode développée dans ce paragraphe permet de localiser le véhicule sur sa voie de circulation et de calculer la géométrie 3D de la route afin de fournir une vue de dessus de la chaussée à l'algorithme de détection d'obstacles.

DeMenthon dans [7] et Kanatani dans [10] présentent des méthodes très bien adaptées à notre problème car les hypothèses qui sont faites sont réalistes quelle que soit la route utilisée. Cependant, ces procédés nécessitent la connaissance de 2 points homologues 3D de la route. Ces points homologues sont établis



Figure 4. – Résultats sur routes marquées.

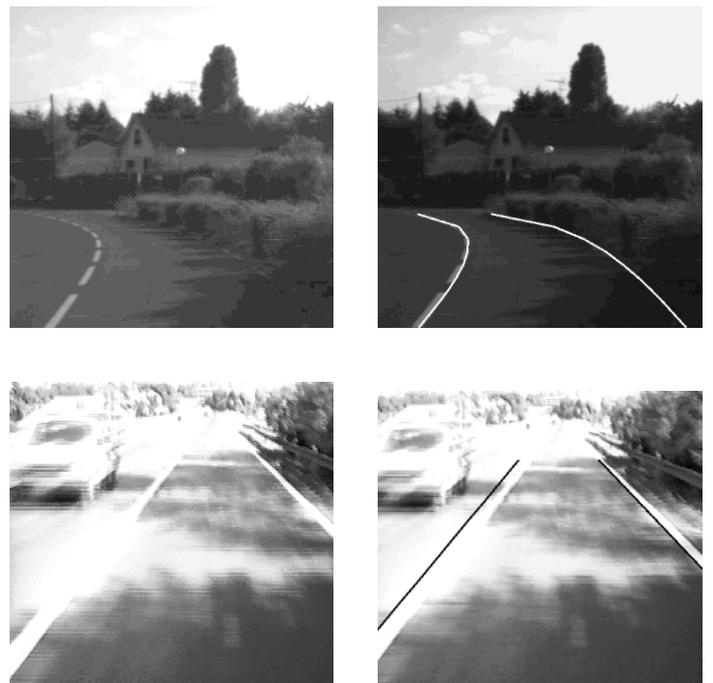


Figure 5. – Résultats sur routes urbaines.



Figure 6. – Résultat sur routes non marquées.

à partir d'un calcul d'angle qui est très sensible aux bruits et cou-teux en temps de calcul. Les désavantages de ces méthodes sont ainsi liés à l'instabilité introduite par l'intégration numérique et les temps de calcul. Notre algorithme fait appel à un précédent travail effectué au laboratoire [5] mais a été adapté à notre méthode de détection des bords de route. Dans ce module, nous avons adopté le modèle de DeMenthon [7], cependant, notre méthode de reconstruction est quelque peu différente afin d'éviter les instabilités. La reconstruction 3D est possible en utilisant certaines hypothèses qui évitent le calcul des points homologues et ainsi le problème de dérivation. La précision du processus de reconstruction est très acceptable dans des conditions normales.

reconstruction des points 3D de l'axe de la route

La figure 7 montre les repères 3D de référence que nous utilisons. Le repère $\mathcal{R} : (x, y, z)^t$ est lié à la route, le bord gauche est tangent à l'axe Oy , le repère $\mathcal{R}_c : (X, Y, Z)^t$ est lié à la caméra : le plan XZ est parallèle au plan image et l'axe Y est défini par l'axe optique.

Si $\underline{P}_g = (X_g, Y_g, Z_g)^t$ et $\underline{P}_d = (X_d, Y_d, Z_d)^t$ sont deux points respectivement situés sur le bord gauche et droit de la route, la distance qui les sépare pour $Y_g = Y_d$ est approximativement égale à la largeur de la route L . Ils sont exprimés dans le repère \mathcal{R}_c . Le point correspondant sur l'axe de la route est $\underline{P}_m = \frac{1}{2}(\underline{P}_g + \underline{P}_d)$. Les points \underline{P}_g et \underline{P}_d se projettent respectivement en coordonnées (u_g, v_g) et (u_d, v_d) dans le plan image. Nous faisons la supposition que le segment $[\underline{P}_g, \underline{P}_d]$ de

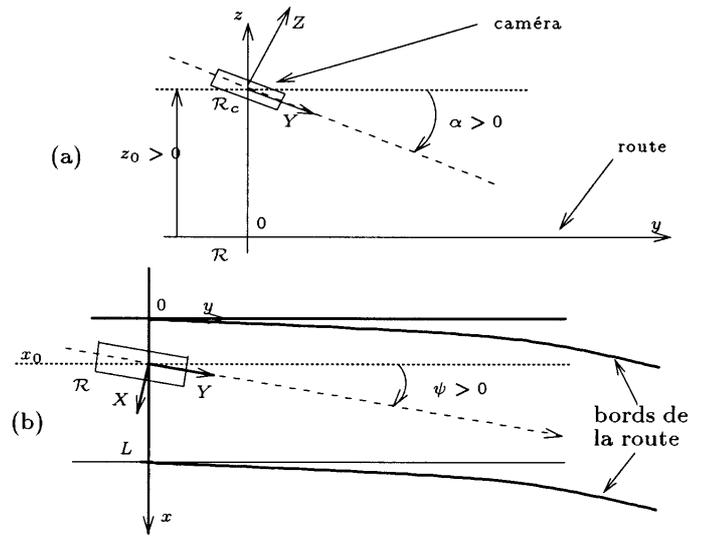


Figure 7. – Repères utilisés : vue de côté (a) et vue de dessus (b).

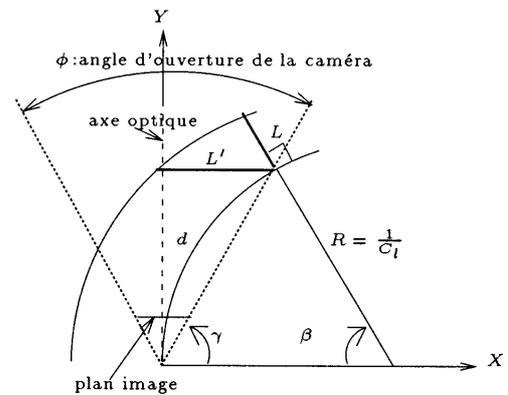


Figure 8. – Influence de la courbure de la route sur les segments formant la route.

longueur L peut être considéré comme parallèle à l'axe X donc $X_d - X_g = L$ et $Y_g = Y_d = Y_m$.

Cette hypothèse n'est plus réaliste si la route présente une courbure horizontale trop prononcée ou de façon similaire si elle est droite mais que la caméra a un angle d'orientation ψ non nul. Dans ce cas, la distance L' qui sépare les points \underline{P}_g et \underline{P}_d est différente de la largeur de la route L . La figure 8 représente ce cas de figure dans les plus mauvaises conditions : le véhicule est exactement placé sur le bord droit de la chaussée alors que la route tourne à droite.

Tableau 1. – Erreur relative entre L et L' pour deux rayons de courbure R et pour une longueur d'arc $d = 100$ m.

$R/(m)$	$100 \frac{L'}{L}$	Remarques
240	9 %	Limite de norme sur autoroute
500	2 %	Cas moyen

Dans ces conditions, le segment de la route fait un angle β avec l'axe des X en restant dans un plan défini par le centre de projection et une ligne de l'image. La valeur de β la plus élevée correspond à l'angle d'ouverture de la caméra ϕ et le point 3D correspondant sur le bord droit est éloigné d'une longueur d'arc $d = \frac{\phi}{C_l}$ du centre de projection. Pour un capteur de taille 8×8 mm avec une focale de 12 mm et une longueur d'arc fixée à 100 m, l'erreur relative entre L' et L correspondante dépend du rayon de courbure. Le tableau 1 donne des exemples de valeurs numériques. Nous supposons également que la route est sans torsion donc $Z_g = Z_d = Z_m$. Si l'angle de rotation de la caméra autour de son axe optique par rapport au plan de la route est non nul ou si la route présente une torsion alors les coordonnées extrêmes des segments sont différentes $Z_d \neq Z_g$ et $\Delta Z = \frac{Z_g - Z_d}{2}$ est non nul. Une autre conséquence est que la projection dans l'image du segment représentant la route n'est pas parallèle aux lignes de l'image $\Rightarrow v_d \neq v_g$.

Comme le montre la figure 9, la projection l du segment de longueur L n'est pas égale à l' correspondant à la projection du segment avec les points extrêmes situés sur la même ligne de l'image.

L'erreur due à la torsion de la route s'exprime de la façon suivante :

$$\frac{l'}{l} = \frac{1}{1 - \frac{\Delta Z^2}{Z^2}}$$

La figure 10 montre que pour une valeur de ΔZ de 20 cm, l'erreur relative de torsion $\frac{l'}{l}$ est supérieure à 10 % pour $-66 \text{ cm} \leq Z \leq +66 \text{ cm}$. Les valeurs de Z appartenant à cet intervalle critique ne sont tout simplement pas retenues. Notons qu'elles sont le plus souvent très peu nombreuses.

Compte tenu de ces hypothèses, les projections de \underline{P}_g et \underline{P}_d sont alors situées sur une même ligne de l'image : $v_g = v_d = v$ et les coordonnées image de la projection du point appartenant à l'axe de la route sont : $(u_m = \frac{u_g + u_d}{2}, v_m = v)$.

Selon le système de coordonnées choisi, les équations classiques de projection sont $u = e_u X/Y$ et $v = e_v Z/Y$. En appliquant

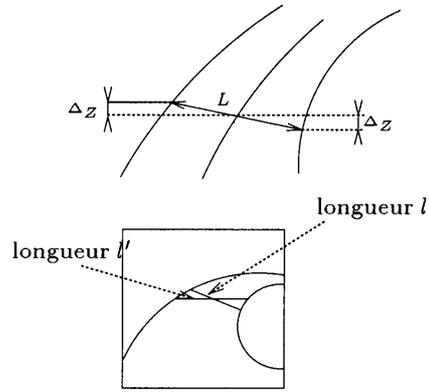


Figure 9. – Influence de la torsion de la route.

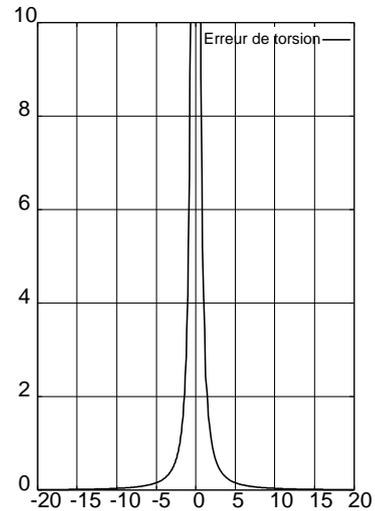


Figure 10. – Évolution de l'erreur relative de torsion $\frac{l'}{l}$ en fonction de Z en mètres pour un demi dévers $\Delta Z = 20$ cm.

ces équations à l'ensemble des points formant l'axe de la route, nous obtenons le système d'équations suivant :

$$\begin{cases} u_d + u_g = 2e_u \frac{X_m}{Y_m} \\ u_d - u_g = e_u \frac{L}{Y_m} \\ v = e_v \frac{Z_m}{Y_m} \end{cases} \Rightarrow \begin{cases} X_m = \frac{L(u_g + u_d)}{2(u_d - u_g)} \\ Y_m = e_u \frac{L}{u_d - u_g} \\ Z_m = \frac{e_u v L}{e_v u_d - u_g} \end{cases} \quad (4)$$

La reconstruction d'un point 3D le long de l'axe de la route nécessite donc la connaissance de la largeur de la route L et d'un couple de points (u_g, v) et (u_d, v) des bords de la route dans l'image.

Cette dernière information est disponible dans le vecteur modèle \underline{X} (éq. (1))

$$\underline{X} = (u_{1g}, \dots, u_{ng}, u_{1d}, \dots, u_{nd})^t$$

provenant du module de détection pour des coordonnées $v_i (i = 1 \dots n)$ fixées.

Le paramètre L peut être calculé en utilisant notre module de détection. Le modèle de route établi en (1) peut être complété par le paramètre L ce qui donne :

$$\underline{X}_L = \begin{pmatrix} \underline{X} \\ L \end{pmatrix} \text{ and } \mathbf{C}_{XL} = \begin{pmatrix} \mathbf{C}_X \\ \vdots \\ \dots \sigma_L^2 \end{pmatrix} \quad (5)$$

\underline{X}_L est lié à \underline{X} par : $\underline{X} = \mathbf{M}\underline{X}_L$ avec $\mathbf{M} = (\mathbf{I}_{n \times n} \mathbf{0}_{n \times 1})$.

Ce nouveau modèle peut être appris de la même façon que le précédent (voir § 2.1) d'où $\underline{X}_L(0)$ et $\mathbf{C}_{XL}(0)$.

Lorsque la route a été retrouvée, le vecteur \underline{X} et sa covariance \mathbf{C}_X sont connus. Un procédé pour extraire la valeur de L est de mettre à jour $\underline{X}_L(0)$ en utilisant le vecteur mesure \underline{X} et son erreur associée \mathbf{C}_X .

Les coordonnées 3D (X_i, Y_i, Z_i) de l'axe de la route sont ainsi calculées et regroupées dans un vecteur \underline{X}_{3D} :

$$\underline{X}_{3D} = (X_1, Y_1, Z_1, \dots, X_n, Y_n, Z_n)^t$$

La matrice de covariance \mathbf{C}_{X3D} de \underline{X}_{3D} est calculée en utilisant \mathbf{C}_{XL} et la matrice jacobienne du système d'équations (4) notée $\mathbf{J}_{2D,3D}$:

$$\mathbf{C}_{X3D} = \mathbf{J}_{2D,3D} \cdot \mathbf{C}_{XL} \cdot \mathbf{J}_{2D,3D}^t$$

L'ensemble de ces points 3D est approximé par la méthode des moindres carrés à un modèle polynômial tel que :

$$X = x_0 + x_1 Y + x_2 Y^2 + x_3 Y^3 \quad (6)$$

$$Z = z_0 + z_1 Y + z_2 Y^2$$

Les ordres des deux fonctions polynomiales ont été choisis selon des considérations expérimentales.

Les paramètres permettant de localiser le véhicule sur la chaussée sont simplement déterminés à partir du vecteur \underline{X}_{poly} :

- position latérale : x_0 ,
- angle de cap : $\psi = x_1$,
- angle d'inclinaison de la caméra : $\alpha = z_1$.

résultats de reconstruction

Le processus permet de reconstruire les profils 3D de l'axe de la route dans le repère caméra. Pour un soucis de clarté les profils reconstruits sont replacés dans le repère de la route en prenant en compte les valeurs de α et ψ établies précédemment.

La figure 11 représente les résultats de l'algorithme de reconstruction 3D obtenus sur une simulation de route complexe puisque la route comporte une importante courbure horizontale. Pour être plus réaliste, un bruit gaussien d'écart type 4 pixels a été ajouté aux points détectés. L'ensemble des points détectés est représenté sur la figure 11(a). Les profils horizontal et vertical sont représentés dans le repère de la route sur les figures 11(b) et 11(c). Les figures (b) et (c) permettent de comparer le profil idéal et le profil reconstruit. Les différences observées sur le profil vertical sont dues aux erreurs sur les points détectés et aux erreurs de modélisation 3D (eq. (6)).

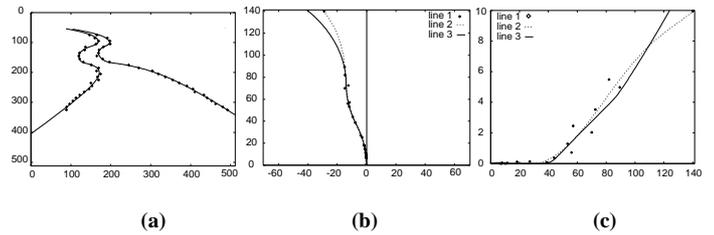


Figure 11. – Résultat de reconstructions dans un cas complexe. (a) : points détectés (avec bruit gaussien de $\sigma = 4$ pixels), (b) : profil horizontal, (c) : profil vertical (line1: points 3D extraits, line2: profil reconstruit, line3: profil réel).

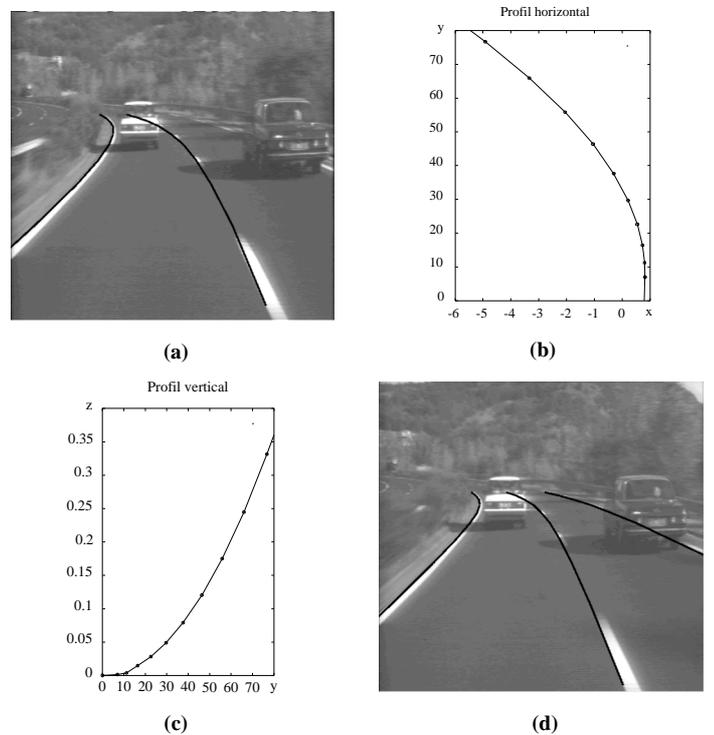


Figure 12. – (a) détection des bords, (b) et (c) profils horizontal et vertical reconstruits, (d) projection dans l'image des bords reconstruits.

La figure 12(a) présente le résultat de notre algorithme de détection sur une image réelle de route. Les profils de la route selon l'axe horizontal et l'axe vertical sont reconstruits et représentés sur les figures 12(b) et 12(c). La distance de reconstruction est de 80 m et les profils sont également représentés dans le repère de la route. La qualité de la reconstruction est difficile à estimer. Néanmoins, la figure 12(d) montre la bonne reprojection des trois bandes blanches dans l'image (alors que seulement les deux bandes de la voie de circulation du véhicule ont été prises en compte).

3. Localisation et suivi d'obstacles

Classiquement, les méthodes utilisées pour la localisation et le suivi d'obstacles reposent sur l'utilisation de capteurs **RADAR** ou **LIDAR** [3] [18]. Ces capteurs permettent une estimation relativement précise des distances inter-véhicules.

Afin d'envisager une solution purement vision, nous utilisons pour la localisation et le suivi d'obstacles la caméra déjà mise en place pour le suivi de route.

De nombreux travaux ont déjà été menés sur ce thème au laboratoire [12] [14]. Ce module de l'algorithme consiste à détecter, localiser et suivre les obstacles présents dans l'image de luminance. De nombreux travaux posent le problème de détection [13] [19] dans la littérature mais la localisation des véhicules est peu abordée, et seule une estimation grossière de la position des cibles est parfois réalisée [2]. Dans tous les cas, si l'on se limite à des techniques par vision monoculaire, il est nécessaire de connaître un modèle d'objet pour localiser. Nous avons équipé les véhicules cibles de marques visuelles appelés « amers » (les marques utilisées sont les 2 feux stop arrières et un feu additionnel sur le toit) (figure 13). Ainsi la localisation 3D des cibles est possible, moyennant la connaissance du positionnement des marques. Notre méthode de localisation fait appel à l'algorithme développé par DeMenthon et Davis [8] qui repose sur une approximation de la projection perspective, appelée orthoperspective. Pour effectuer le suivi des cibles détectées, nous proposons une approche de type « prédiction-vérification » reposant sur la théorie du filtrage de Kalman.

Dans un premier temps, nous présentons les différents modules permettant de localiser et de suivre les obstacles présents dans la scène. Dans un second temps, nous détaillons le processus de suivi et plus particulièrement l'estimation des positions et des vitesses des cibles. Nous terminerons en décrivant une méthode d'adaptation du filtrage au suivi de mobiles manœuvrants.



Figure 13. – Positionnement des amers.

3.1. le système de vision

Le suivi de plusieurs cibles en mouvement dans une scène routière impose le développement d'un système de vision capable de prendre en compte l'apparition (initialisation des paramètres cinématiques de la cible détectée) et la disparition (fin du suivi) des cibles. Ces difficultés nous ont conduits à construire 3 modules qui fonctionnent en parallèle (voir figure 14) :

- un module de détection de véhicules,
- un module de suivi (voir § 3.2),
- un module de supervision.

Nous décrivons ici de façon succincte, les modules de détection et de supervision. Il est évident que les marques visuelles constituent une aide précieuse pour chacun de ces modules.

module de détection

Ce module a pour but de détecter les véhicules qui entrent dans le champ de la caméra. La distance d'apparition des véhicules s'étend sur une plage variant de 10 à 100 mètres. Avec des méthodes de détection basées sur les niveaux de gris, les variations d'aspect des véhicules à détecter sont difficiles à gérer. Différents essais sur route montrent que la présence des marqueurs nous permet d'utiliser une même méthode de reconnaissance, que les véhicules soient éloignés ou proches.

La reconnaissance s'effectue en 2 étapes :

- Reconnaissance d'une voiture dans l'image de luminance. Elle repose sur la reconnaissance de la forme du triangle dans l'image, triangle créé par la projection des amers dans le plan image.
- Vérification de la cohérence temporelle de la forme. Est-ce que la forme identifiée lors de la première étape est viable ? Ne se déforme t-elle pas au cours du temps ? Cette étape utilise 3 images successives. Si elle est validée, les paramètres dynamiques de la cible sont calculés et le suivi de cette cible est lancé.

module de supervision

Ce module maintient une liste des voitures présentes dans la scène. Ce module gère efficacement l'apparition et la disparition

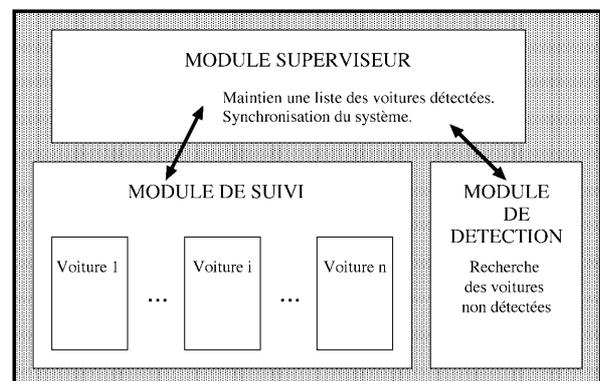


Figure 14. – Suivi multicible.

des véhicules en fonction des informations fournies par les modules de détection et de suivi. De plus, ce module permet une synchronisation de l'ensemble du système, et permet d'éviter que plusieurs pistes soient lancées sur un même véhicule.

3.2. suivi de cibles

Le module de suivi a pour but de résoudre le problème de l'appariement temporel entre les véhicules déjà détectés et les mesures réalisées dans l'image. Il permet d'associer 3 détections de l'image avec chacun des véhicules cibles suivis, problème délicat pour 2 raisons : (1) présence de fausses détections lors de la phase d'acquisition (bruit de mesure, réflexions, mauvaises conditions de luminosité) et (2) interférences entre les cibles.

Face à ces difficultés, nous proposons une approche de type « prédiction-vérification » reposant sur la théorie du filtrage de Kalman.

Le déroulement de l'algorithme se fait en 5 étapes. La première étape est la prédiction dans l'espace 3D de l'état de la cible qui impose la définition d'un modèle d'évolution dans le temps de l'état de la cible. Cette étape permet également de placer des zones de recherche des primitives dans l'espace image. La seconde étape permet de trier les primitives détectées. Seules les primitives compatibles avec la position de la cible sont ainsi prises en compte. La troisième étape a pour objectif d'apparier 3 primitives détectées dans l'image avec le véhicule suivi. Cet appariement est réalisé par un critère global de forme 2D. La localisation du véhicule, à partir des informations 2D, est faite dans la quatrième étape et la dernière étape permet de recalculer l'état de la cible en utilisant les résultats de localisation.

prédiction

La prédiction de l'état de la cible est obtenue par un modèle d'évolution à vitesse constante. Si, dans un repère associé au véhicule caméra, l'état estimé du système à l'instant k est décrit par le vecteur $\underline{S}_{k/k}$ tel que :

$$\underline{S}_{k/k} = (T_x, T_y, T_z, V_x, V_y, V_z, R_x, R_y, R_z, \omega_x, \omega_y, \omega_z)^t$$

où T_x, T_y, T_z sont les translations entre le véhicule caméra et le véhicule suivi, V_x, V_y, V_z les vitesses de translation, R_x, R_y, R_z les rotations et $\omega_x, \omega_y, \omega_z$ les vitesses de rotation. A $\underline{S}_{k/k}$, on associe une matrice de covariance appelée $\mathbf{P}_{k/k}$.

Selon l'axe (Ox), le modèle d'évolution donne :

$$\begin{cases} T_{x(k+1/k)} = V_{x(k/k)}T_e + T_{x(k/k)} \\ R_{x(k+1/k)} = \omega_{x(k/k)}T_e + R_{x(k/k)} \end{cases}$$

avec T_e la période d'échantillonnage du système. De façon symétrique, on peut déduire les équations d'évolution pour les axes (O_y) et (O_z).

Les commandes d'accélération, de freinage, de changement de direction, exercées sur les véhicules cibles ne peuvent actuellement être ni mesurées ni estimées et ne sont donc pas introduites dans le modèle d'évolution. Ces commandes sont prises en compte lors du calcul de la matrice de covariance associée au modèle d'évolution, sous forme d'un bruit décorréolé non fonction du temps.

Appariement local

Cette étape permet de trier les primitives détectées dans l'image et de ne conserver que celles compatibles avec la cible. La prédiction de l'état de la cible, avec sa matrice de covariance associée, permet de placer et de dimensionner 3 zones de recherche des primitives dans l'image 2D pour chacun des véhicules. Seules les détections appartenant aux fenêtres sont conservées, ce qui permet de supprimer les primitives éloignées de la cible.

appariement global

L'étape d'appariement global réalise l'appariement entre les primitives détectées et celles composant le véhicule suivi. Afin de sélectionner une primitive par fenêtre d'intérêt, nous calculons la distance de Mahalanobis entre la forme 2D prédite et celle composée par les primitives détectées.

Le vecteur, représentant la forme 2D recherchée, est défini comme suit :

$$\underline{x}_i = \begin{pmatrix} \frac{v_2 - v_0}{u_2 - u_0} \\ \frac{u_1 - u_0}{v_1 - v_2} \\ \frac{u_1 - u_0}{v_1 - v_0} \\ \sqrt{(u_2 - u_0)^2 + (v_2 - v_0)^2} \end{pmatrix}$$

où (u_0, v_0) , (u_1, v_1) et (u_2, v_2) sont les coordonnées 2D respectives du feu arrière gauche, du feu du toit et du feu arrière droit. La matrice de covariance associée au vecteur de forme moyen, nécessaire au calcul de la distance de Mahalanobis, est établie à partir de la matrice associée aux prédictions 3D.

La combinaison pour laquelle la distance est inférieure à un seuil est sélectionnée. Si aucune combinaison ne peut être triée, l'algorithme est dégradé de façon à fonctionner avec 2, voire 0 tache(s) (localisation avec prédictions).

localisation

La figure 15 montre le modèle associé à notre véhicule cible (P_0 feu arrière gauche, P_1 feu du toit, P_2 feu arrière droit). La méthode de localisation, développée par DeMenthon et Davis [8], utilise 3 amers.

Elle repose sur une approximation de la projection perspective, appelée orthoperspective par ses auteurs. Dans le cadre de l'application, cette approximation est réaliste car le rapport entre la profondeur du modèle et la distance caméra-cible reste faible.

La figure 16 montre la projection des 3 amers P_0 , P_1 et P_2 sous la projection orthoperspective. Nous en déduisons le système suivant :

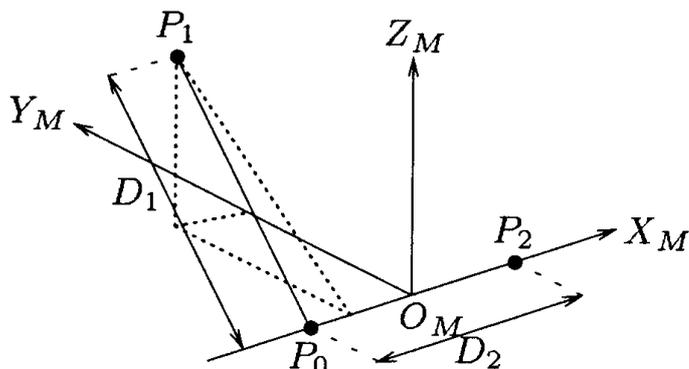


Figure 15. – Modèle des amers.

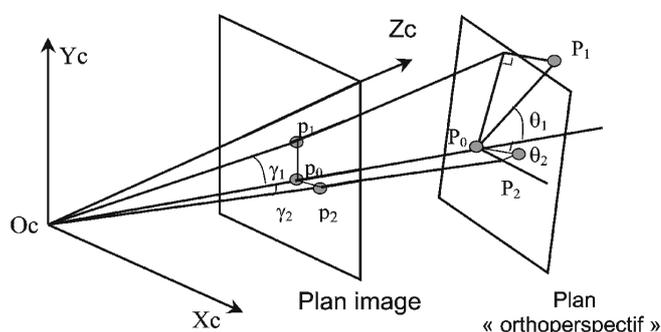


Figure 16. – Projection orthoperspective.

$$\begin{cases} \frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{P_0 P_2 \tan(\gamma_1)}{P_0 P_1 \tan(\gamma_2)} = K \\ \cos(\alpha_m) = \sin(\theta_1) \sin(\theta_2) \cos(\phi) + \cos(\theta_1) \cos(\theta_2) \end{cases} \quad (7)$$

où $\phi = p_1 \widehat{p_0 p_2}$ et $\alpha_m = P_1 \widehat{P_0 P_2}$.

On peut montrer que 2 couples (θ_1, θ_2) sont solutions de ce système. Vu la géométrie du dispositif (véhicule sur ses roues), un couple solution peut être facilement éliminé.

Au final, la méthode permet de calculer un vecteur de mesure, associé à chacun des véhicules. Ce vecteur de mesure a la forme suivante :

$$\underline{Y}_{k+1/k+1} = (T_{xm}, T_{ym}, T_{zm}, R_{xm}, R_{ym}, R_{zm})^t$$

où T_{xm}, T_{ym}, T_{zm} sont les translations mesurées et R_{xm}, R_{ym}, R_{zm} sont les rotations mesurées. A ce vecteur de mesure est associé la matrice de covariance $\mathbf{Q}_{mes_{k+1}}$.

mise à jour

L'étape de mise à jour consiste à utiliser le vecteur de mesure $\underline{Y}_{k+1/k+1}$ afin d'estimer le vecteur d'état $\underline{S}_{k+1/k+1}$. Cette mise à jour est réalisée suivant le formalisme du filtrage de Kalman :

$$\begin{cases} \underline{S}_{k+1/k+1} = \underline{S}_{k+1/k} + \mathbf{K}[\underline{Y}_{k+1} - \mathbf{H}\underline{S}_{k+1/k}] \\ \mathbf{P}_{k+1/k+1} = \mathbf{P}_{k+1/k} - \mathbf{K}\mathbf{H}\mathbf{P}_{k+1/k} \end{cases}$$

avec :

$$\mathbf{K} = \mathbf{P}_{k+1/k} \mathbf{H}^t [\mathbf{H}\mathbf{P}_{k+1/k} \mathbf{H}^t + \mathbf{Q}_{mes_k}]^{-1}$$

résultats

Les séquences présentées dans les prochains paragraphes sont composées d'images issues de la caméra embarquée dans notre véhicule au niveau du rétroviseur intérieur (cf. §5). Cette caméra est équipée d'un filtre interférentiel.

La séquence présentée figure 17 montre une voiture en déplacement devant notre véhicule. Cette séquence comprend 1400 images.

En début de séquence, la voiture à détecter se déplace à faible distance devant VELAC (notre véhicule expérimental) (environ 20 mètres). Elle dépasse 2 camions (images 0 à 425) puis se rabat (425 à 600). Le véhicule caméra (VELAC) suit la voiture équipée d'amers. De l'image 600 à la fin, le véhicule cible s'éloigne.

Nous montrons sur la figure 18 l'évolution de la distance (VELAC-véhicule cible) et de la vitesse relative estimées. On peut remarquer que le suivi se déroule sans perte (sinon distance égale à 0) malgré des mouvements latéraux et longitudinaux importants. La courbe de distance est relativement peu bruitée, profitant des avantages apportés par la présence d'amers. Par contre l'estimation de la vitesse est très bruitée, bruit devenant de plus en plus important au fur et à mesure que la distance augmente. Ce bruit est dû à l'introduction des commandes dans la matrice de bruit du modèle. Etant surestimé, il tend à augmenter la bande passante du filtre de Kalman. L'imprécision dans l'estimation de la vitesse de la cible rend moins précise la prédiction de l'état de la cible.



Image 200



Image 800

Figure 17. – Séquence monocible.

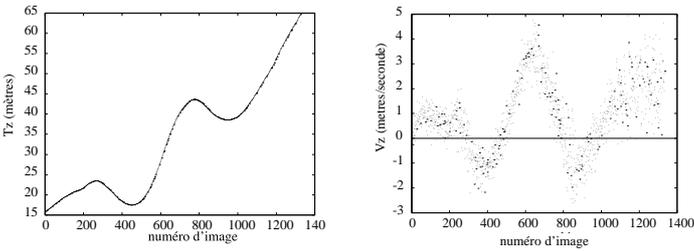


Figure 18. – Résultats du suivi monocible sans détection de manoeuvre.

3.3. suivi de mobiles manoeuvrants

Dans notre cas, la modélisation des mouvements de la cible est forcément imprécise car les commandes des véhicules ne peuvent pas être mesurées directement. L'erreur induite par ces commandes a été introduite dans le bruit de modélisation, bruit non dépendant du temps.

Nous proposons une méthode pour que le filtre puisse « s'auto-adapter » aux phases de manoeuvre. Si la cible n'est pas en manoeuvre, le filtre doit diminuer sa bande passante de façon à filtrer le bruit de mesure. Par contre, si la cible est en manoeuvre, le filtre doit être capable de, soit augmenter sa bande passante, soit modéliser correctement cette manoeuvre.

détection des phases de manoeuvres

Quand la cinématique de la cible ne correspond plus à celle décrite par le modèle de mouvement, la prédiction de l'état de la cible (vecteur $\underline{S}_{k+1/k}$) ne correspond plus à l'état réel de la cible. Le signal d'innovation, $\gamma_{k+1} = \underline{Y}_{k+1} - \underline{H}\underline{S}_{k+1/k}$, signal d'erreur entre les mesures et l'état prédit, n'est plus centré, à moyenne nulle. Dans le cas de manoeuvres de forte amplitude, l'erreur de prédiction peut causer des pertes de suivi.

Bar Shalom [17] propose de détecter ces phases de manoeuvres en vérifiant si le signal d'innovation est compatible avec sa matrice de covariance. S'il existe une compatibilité, le système est bien modélisé, dans le cas contraire le suivi est mauvais et les mobiles sont en phase de manoeuvre. Pour éviter des perturbations passagère, le calcul de la distance de Mahalanobis est moyennée sur une fenêtre glissante par :

$$\rho_k = \alpha\rho_{k-1} + (1 - \alpha)\varepsilon_k$$

où ε_k est la valeur de la distance de Mahalanobis sur les innovations à l'instant k . Si ρ_k est supérieure à un seuil, on estime que la modélisation ne coïncide plus avec la cinématique de la cible, et donc que la cible est en phase de manoeuvre.

adaptation du filtre

Bar Shalom [17] propose une approche par modification des modèles de bruit. Dès qu'une manoeuvre est détectée, les coefficients composant la matrice de covariance de bruit de modèle sont augmentés. Dans ce cas, le filtrage est conservé, seules les valeurs contenues dans les matrices de bruit sont adaptées.

Dès qu'une manoeuvre est détectée, nous augmentons les paramètres de \mathbf{Q}_k (matrice de covariance sur le modèle d'évolution), modification proportionnelle à l'amplitude de la manoeuvre. La correction des matrices de bruit est réalisée axe par axe, translation par translation, rotation par rotation. La réévaluation de la matrice \mathbf{Q}_K est réalisée à partir de la matrice de covariance sur l'innovation $\mathbf{N}_{k+1/k+1}$. Nous recherchons la matrice \mathbf{Q}_K qui permettrait de ne pas sous-évaluer la matrice $\mathbf{N}_{k+1/k+1}$.

En phase de manoeuvre, la matrice $\mathbf{N}_{k+1/k+1}$ est sous-évaluée. Si une manoeuvre est détectée pour la $i^{\text{ème}}$ composante du vecteur d'état, alors en notant $N_{k+1/k+1}[i, j]$, l'élément de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne de $\mathbf{N}_{k+1/k+1}$ et avec $\rho[i]$ représentant la distance de Mahalanobis par rapport à l'élément i du vecteur d'état (i et j compris entre 1 et 6) :

$$\tilde{N}_{k+1/k+1}[i, j] = \sqrt{(\rho[i] - (Seuil[i] - 1))} N_{k+1/k+1}[i, j]$$

où $\tilde{\mathbf{N}}$ est la matrice \mathbf{N} réévaluée vis-à-vis de la manoeuvre. $\rho[i]$ est la distance de Mahalanobis filtrée pour l'élément i du vecteur d'état. $Seuil[i]$ est le seuil sur la distance de Mahalanobis permettant de caractériser si la cible est en manoeuvre ou pas.

A partir de $\tilde{\mathbf{N}}$, nous calculons la matrice de covariance réévaluée $\tilde{\mathbf{L}}$ associé à $\underline{H}\underline{S}_{k+1/k}$:

$$\tilde{\mathbf{L}}_{k+1/k} = \tilde{\mathbf{N}}_{k+1/k+1} - \mathbf{Q}_{mes_{k+1}}$$

Connaissant $\tilde{\mathbf{L}}$ et la matrice d'évolution \mathbf{M} , nous pouvons calculer :

$$\underline{H}\tilde{\mathbf{Q}}_k\underline{H}^t = \tilde{\mathbf{L}} - \underline{H}\mathbf{M}\mathbf{P}_{k/k}\mathbf{M}^t\underline{H}^t$$

la matrice \mathbf{Q}_k représentent les covariances sur les erreurs de translations, de vitesses et de rotations. A partir de ces différentes connaissances, nous pouvons facilement calculer $\tilde{\mathbf{Q}}_k$, la matrice réévaluée qui sera utilisée lors de l'itération suivante.

résultats

Les courbes de la figure 19 présentent les résultats obtenus par le filtrage avec adaptation aux manoeuvres à partir de la séquence décrite dans § 3.2.

L'estimation de la vitesse est beaucoup moins bruitée que précédemment. Le tableau 2 récapitule les résultats obtenus pour l'estimée de la vitesse relative de la cible pour une distance entre la caméra et l'obstacle fixée à 25 mètres.

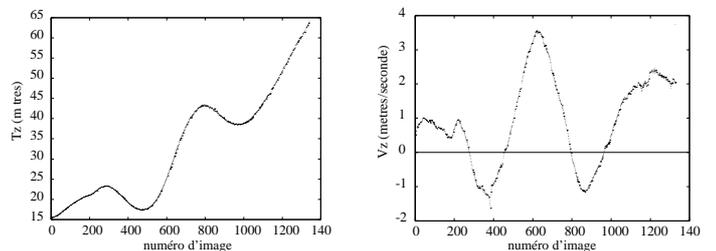


Figure 19. – Résultats du suivi monocible avec adaptation aux phases de manoeuvres.

Tableau 2. – Les vitesses estimées à une distance de 25 mètres.

Vitesse « réelle » km/h	Vitesse estimée m/s (km/h)	Erreur
40	10.4 (37.4)	2.6 (6.5 \%)
50	14.4 (51.8)	1.8 (3.6 \%)
60	15.2 (54.8)	5.2 (8.7 \%)
70	20.3 (73.1)	3.1 (4.4 \%)
80	24.4 (87.8)	7.8 (9.8 \%)

L'état de la cible est alors prédit avec plus de précision, avec des covariances plus faibles, ce qui réduit la taille des fenêtres d'intérêt. En terme de temps de calculs, la surcharge provoquée par l'adaptation du filtre aux manœuvres est compensée par la diminution de la taille des fenêtres d'intérêt.

Au final, on obtient donc un filtre plus précis sans globalement une surcharge de calcul. Le filtre est ainsi capable de s'adapter à tout mouvements relatifs caméra / obstacles (freinage, accélérations, virages, changements de voie de circulation, . . .).

4. coopération

Les deux parties précédentes de cet article ont été consacrées à deux modules permettant l'assistance à la conduite : détection de bords de route et reconstruction des profils 3D de la route d'une part, localisation et suivi de véhicules obstacles d'autre part. La combinaison de ces différentes informations va permettre d'identifier le véhicule le plus dangereux dans la voie de circulation de notre véhicule. Contrairement aux méthodes utilisant des capteurs de type RADAR et LIDAR, qui permettent une bonne estimation de la distance inter-véhicule, nous pouvons fournir une information concernant la géométrie 3D de la voie de circulation du véhicule porteur de la caméra et ainsi ne pas considérer des véhicules extérieurs à la voie comme dangereux (cas fréquent en situation de virages).

4.1. approche

Le procédé que nous avons mis en œuvre pour combiner les informations des différents modules est assez simple. A partir du module de reconstruction de la route (voir § 2.2), nous déterminons le profil horizontal 3D de la voie de circulation du véhicule porteur. Ce module nous permet également d'estimer la position relative du véhicule dans sa voie de circulation x_0 ainsi que son angle de cap ψ .

L'algorithme de localisation et de suivi de véhicules obstacles est capable de fournir la position des véhicules suivis par rapport au véhicule porteur. Nous avons également une information concernant la vitesse des véhicules cibles.

Notre but est d'identifier le véhicule le plus dangereux appartenant à la voie de circulation du véhicule porteur.

Dans notre cas, la localisation 3D des bords de la voie est approximativement donnée par (voir [5] pour les suppositions utilisées) :

$$X_g = x_0 - L/2 + x_1Y + x_2Y^2 + x_3Y^3$$

$$X_d = x_0 + L/2 + x_1Y + x_2Y^2 + x_3Y^3$$

Les coordonnées d'un véhicule cible i sont données, dans le repère lié au véhicule, par :

$$\underline{V}_i = (T_{xi}, T_{yi}, T_{zi})^t$$

Un simple vecteur de translation \underline{T} permet d'obtenir les coordonnées \underline{V}_{ci} du véhicule cible dans le repère caméra :

$$\underline{V}_{ci} = (X_{vi}, Y_{vi}, Z_{vi})^t = \underline{V}_i + \underline{T}$$

Un véhicule i est considéré comme appartenant à la voie de circulation du véhicule porteur si :

$$X_{vi} \in [X_g, X_d] \text{ pour } Y = Y_{vi}$$

Par la suite, les données concernant la position et l'orientation du véhicule porteur (VELAC)(cf. § 5) et la vitesse des véhicules cibles nous permettront d'intervenir sur les commandes du véhicules (latérale ou longitudinale) dans le cas de situations jugées dangereuses.

4.2. résultats

Les résultats, figure (20), sont relatifs au module de coopération de notre algorithme.

Les images de gauche présentent les résultats des phases de détections des bords de voie et d'obstacles. Les distances indiquées dans le bas de l'image représentent les distances entre chacun des véhicules cibles et VELAC.

Pour une image donnée, la figure de droite montre la vue de dessus de la scène routière reconstruite par l'algorithme. Le rectangle sur l'axe x symbolise le véhicule caméra VELAC. Les bandes parallèles représentent les bandes blanches extraites par l'algorithme alors que les traits horizontaux montrent les véhicules détectés dans l'image.

A partir de ces résultats, nous pouvons identifier le véhicule le plus dangereux pour notre véhicule expérimental et ainsi envisager un contrôle longitudinal.

Sur la figure 20-a, les deux véhicules cibles se trouvent dans la même voie de circulation que le véhicule porteur. Le véhicule le plus dangereux est donc le plus proche. La figure 20-b présente un cas où les 2 véhicules cibles ne sont pas sur la même voie de

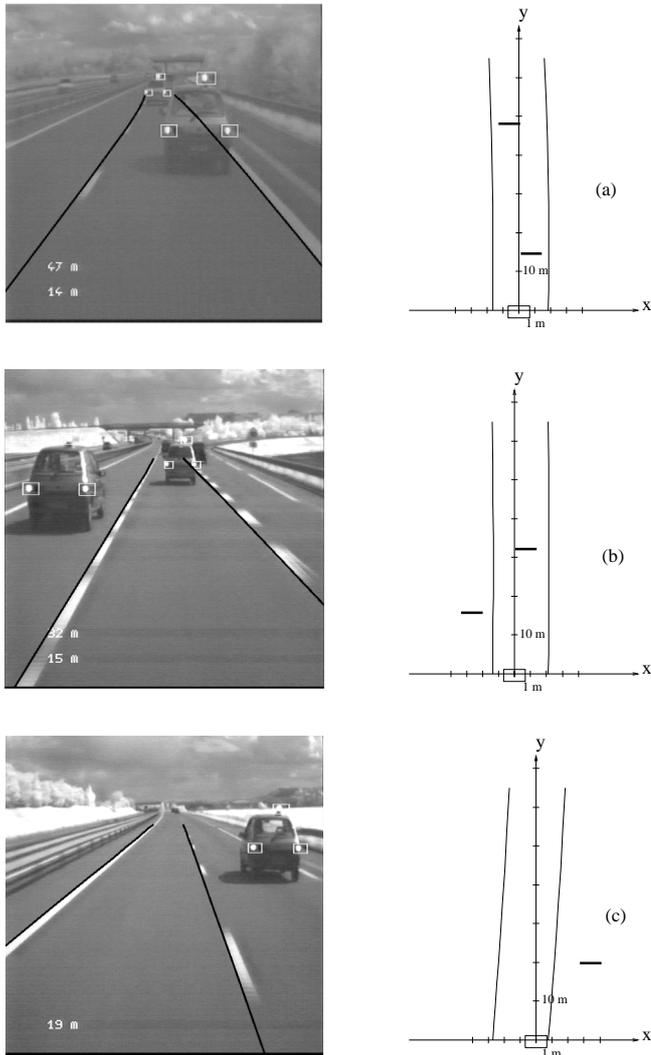


Figure 20. – Résultats de l'algorithme de coopération.

circulation. Dans ce cas, le véhicule identifié comme « le plus dangereux » n'est pas le plus proche (pas la même voie de circulation) mais celui situé sur notre voie. Aucun véhicule n'est jugé dangereux sur la figure 20-c car le seul véhicule détecté n'est pas sur la même voie de circulation que notre véhicule. Sur cet exemple, le véhicule **VELAC** se déplace de la file de droite à la file de gauche ce qui explique un angle de cap important. Ces résultats peuvent être améliorés par une collaboration plus complète entre ces 2 algorithmes. Par exemple, la connaissance de la position des obstacles peut permettre d'identifier les zones où les bords de la route sont masqués. Par conséquent, la recherche des bords de la route est inutile dans ces zones. Réciproquement, le module de détection de route pourrait aider à la recherche des véhicules entrant dans le champ de la caméra appartenant à la voie de circulation.

5. implantation

L'ensemble de l'algorithme a été implanté dans le véhicule expérimental du laboratoire **VELAC** (figure 21).

Ce véhicule est équipé d'une architecture permettant les traitements temps réel (voir figure 22). Elle est constituée d'une architecture MIMD-DM basé sur des processeurs de type DEC alpha. La détection de véhicules dans la scène a été implantée sur un processeur DEC. Un second processeur gère le suivi des véhicules déjà détectés. Le temps de calcul pour le suivi d'un véhicule cible est inférieur à 20 ms.

L'implantation de l'algorithme de suivi de lignes blanche a été faite sur un troisième processeur. Les temps de calcul engendrés par ce processus varient de 80 à 200 ms pour une image donnée. Les temps de communication et les temps de calcul des moindres



Figure 21. – VELAC : Véhicule Expérimental du Lasmea pour l'Aide à la Conduite.

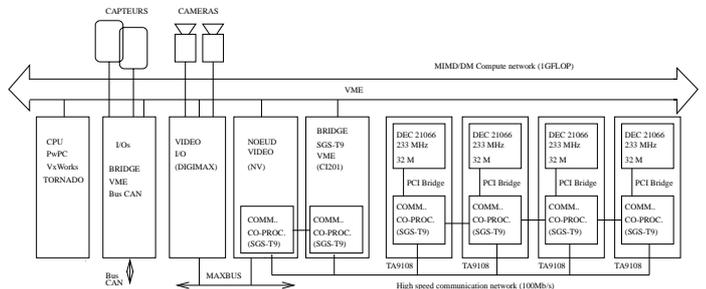


Figure 22. – Architecture informatique de vision.

carrés médians représentent en moyenne 60 % du temps total sur une image. Cette première implantation montre la faisabilité de l'approche proposée.

6. conclusion et perspectives

Nous avons décrit dans cet article une méthode permettant d'identifier le véhicule le plus dangereux appartenant à la voie de circulation d'un véhicule à partir d'images fournies par une caméra monochrome embarquée dans celui-ci. L'algorithme est principalement basé sur deux modules. Le premier permet de détecter les bords de route dans l'image et d'en déduire l'attitude 3D de l'axe de la voie de circulation. Un modèle d'ordre élevé des bords de la voie est utilisé. Il est initialisé par une phase d'apprentissage et mis à jour de façon récursive mais optimale après chaque détection. Les principaux avantages de ce processus sont la précision et la gestion des occultations ou de l'imperfection du marquage. De plus, cette étape permet de localiser le véhicule dans sa voie de circulation.

Le deuxième module est capable de détecter, localiser et suivre des véhicules équipés d'amers. Ce processus permet de suivre les véhicules cibles dans l'espace 3D et d'estimer leur vitesse relative par rapport au véhicule caméra. La coopération de ces deux modules permet de détecter les obstacles les plus dangereux. Les résultats sur chacun de ces modules sont très satisfaisants. Les perspectives à court terme liées à cette méthode sont une coopération plus complète entre les deux algorithmes, une implantation plus précise (parallélisation) du processus de suivi de lignes blanches dans le véhicule expérimental du laboratoire (VELAC) afin de limiter les temps de calcul et de développer une stratégie de contrôle de l'accélérateur et du freinage de VELAC. Ce contrôle compléterait alors la mise en œuvre du système ACC.

BIBLIOGRAPHIE

- [1] R. Aufrère, R. Chapuis, and F. Chausse. Détection précise d bords de route par vision monoculaire embarquée. In *12ème congrès Reconnaissance des Formes et Intelligence Artificielle*, Paris, France, 1-3 Février 2000.
- [2] M. Betke and H. Nguyen. Highway scene analysis from a moving vehicle under reduced visibility conditions. In *Processing of the IEEE International Conference on Intelligent Vehicles*, volume 1, pages 131–136, Stuttgart, Germany, October 1988.
- [3] M. Camiade, D.Domnesque, P.F. Alleaume, A. Mallet, and H. Daemkes. Gaas mmics for automotive radar applications. In *Proceeding of the IEEE International Conference on Intelligent Vehicles*, volume 2, pages 371–375, Stuttgart, Germany, October 1998.
- [4] R. Chapuis. *Suivi de primitives image, application à la conduite automatique sur route*. PhD thesis, Univ. B. Pascal, Clermont-Ferrand, France, Janvier 1991.
- [5] R. Chapuis, R. Aufrère, and F. Chausse. Recovering a 3d shape of road by vision. In *7th International Conference on Image Processing and its Applications*, Manchester (U.K.), July 12-15 1999.
- [6] R. Chapuis, A. Potelle, J.L. Brame, and F. Chausse. Real time vehicle trajectory supervision on highway. *International Journal of Robotics Research*, 14(6) : 531–542, December 1995.
- [7] D. DeMenthon. Reconstruction of a road by matching edge points in the road image. In *Technical Report CAR-TR-368, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park MD20742*, June 1998.
- [8] D.F. DeMenthon and L.S Davis. New exact and approximate solutions of the three point perspective problem. *IEEE Transactions on Pattern analysis and Machine Intelligence*, pages 1100–1105, November 1992.
- [9] E.D. Dickmanns and V. Graefe. Dynamic monocular machine vision and applications of dynamic monocular vision. *International Journal of Machine Vision and Application*, 1 : 223–240 and 241–261, 1988.
- [10] K. Kanatani and K. Watanabe. Reconstruction of 3d road geometry from images for autonomous land vehicles. *IEEE Trans. On Robotics and Automation*, 6 : 127–132, February 1990.
- [11] C Kreucher, S. Lakshmanan, and K. Kluge. A driver warning system based on th lois lane detection algorithm. In *Proceeding of the IEEE International Conference on Intelligent Vehicles*, volume 1, pages 17–22, Stuttgart, Germany, October 1998.
- [12] F. Marmoiton, F. Collange, J. Alizon, and J.P. Dérutin. 3d localization of a car observed through a monocular video camera. In *Proceedings of the IEEE International Conference on Intelligent Vehicles*, volume 1, pages 189–194, Stuttgart, Germany, October 1998..
- [13] N.D. Matthews, P.E.An, D. Charnley, and C.J. Harris. Vehicle detection and recognition in greyscale imagery. In *2d symposium on Intelligent Autonomous Vehicle*, pages 31–36, Espoo, Finland, June 1995.
- [14] E. Montagne, J. Alizon, P. Martinet, and J. Gallice. Real time 3D location of a car from three characteristic points observed in a video image sequence. In *7th Symposium on Transportation Systems*, volume 2, pages 385–390, Tianjin, China, August.
- [15] S. Schultz. *A modeled based codesign applications : the design of an autonomous intelligent cruise controller*. PhD thesis, Department of Electrical and Computer Engineering, University of Arizona, USA. 1997.
- [16] L.T. Shaaser and B.T Thomas. Finding road lane boundaries for vision guided navigation. *Proceedings of the Roundtable discussion on Vision Vehicle Guidance*, pages 10.1–10.9, 1990.
- [17] Y. Bar Shalom and T.E. Fortman. *Tracking and data association*, volume 179. Academic Press, INC., 1988.
- [18] L. Trassoudaine, J. Alizon, J. Gallice, and F. Collange. Visual tracking by a multisensorial approach. In *1st Workshop on Intelligent Autonomous Vehicles*, pages 113–118, Southampton, April 1993.
- [19] T. Zielke, M. Braukmann, and W. Von Steelen. Intensity and edge based symmetry detection with an application to car following. In *Computer Vision Graphics and Image Processing, Image Understanding*, volume 58(2), pages 177–190, September 1993.

Manuscrit reçu le 25 février 2000

LES AUTEURS

Romuald AUFRERE



Titulaire d'un DEA Electronique et Systèmes, option Vision pour la Robotique, obtenu en 1998 à l'université Blaise Pascal de Clermont-Ferrand. Il prépare actuellement une thèse de doctorat sur le développement et l'implantation d'algorithmes de suivi de route au sein du Groupe Automatique : Vision et Robotique (GRAVIR) du Laboratoire des sciences et Matériaux pour l'Electronique, et d'Automatique (LASMEA UMR 6602 du CNRS).

Jean-Pierre DERUTIN



Il a obtenu l'habilitation à diriger des recherches en 1993 à l'université Blaise Pascal à Clermont-Ferrand. Professeur depuis 1994, il est actuellement responsable des activités « Architectures et programmation des machines de perception visuelle » du LASMEA (UMR 6602 CNRS). Il s'intéresse plus particulièrement au modèle MIMD-DM pour les machines dédiées à la vision, à leur programmation ainsi qu'aux méthodologies de prototypages rapide d'application sur ce type de machine dans une optique Adéquation Algorithmique Architecture A³.

François MARMOITON



Ingénieur Génie Electrique CUST en 1995, il est titulaire en 2000 d'un doctorat de l'université Blaise Pascal à Clermont-Ferrand (vision pour la robotique). Son thème de recherche, au Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique (LASMEA) est la vision pour la robotique, notamment les problèmes de reconstruction tridimensionnelle et de suivi de primitives dans l'espace image.

François COLLANGE



Maître de conférences à l'IUT de Clermont-Ferrand. Après une thèse de doctorat en Electronique en 1986, il a rejoint en 1987 le Centre de Morphologie Mathématique de Fontainebleau. Depuis 1990, il travaille au Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique (LASMEA). Ses recherches portent sur les méthodes de fusion de données multicapteurs.

Roland CHAPUIS



Il est titulaire en 1991 d'un doctorat de l'université Blaise Pascal à Clermont-Ferrand (spécialité électronique). Il est actuellement Maître de conférences au Centre Universitaire des Sciences et techniques (CUST) à Clermont-Ferrand. Son thème de recherche, au Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique (LASMEA), est la reconnaissance et le suivi dynamique de primitives dans les images en environnement extérieur.