Perception par caméra des bords de route

Jean-Philippe Tarel

Juillet 2002

Laboratoire central des Ponts et Chaussées
58, bd Lefebvre, F 75732 Paris Cedex 15

Jean-Philippe TAREL
Chargé de Recherche
Laboratoire Central des Ponts et Chaussées
Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs (LIVIC)

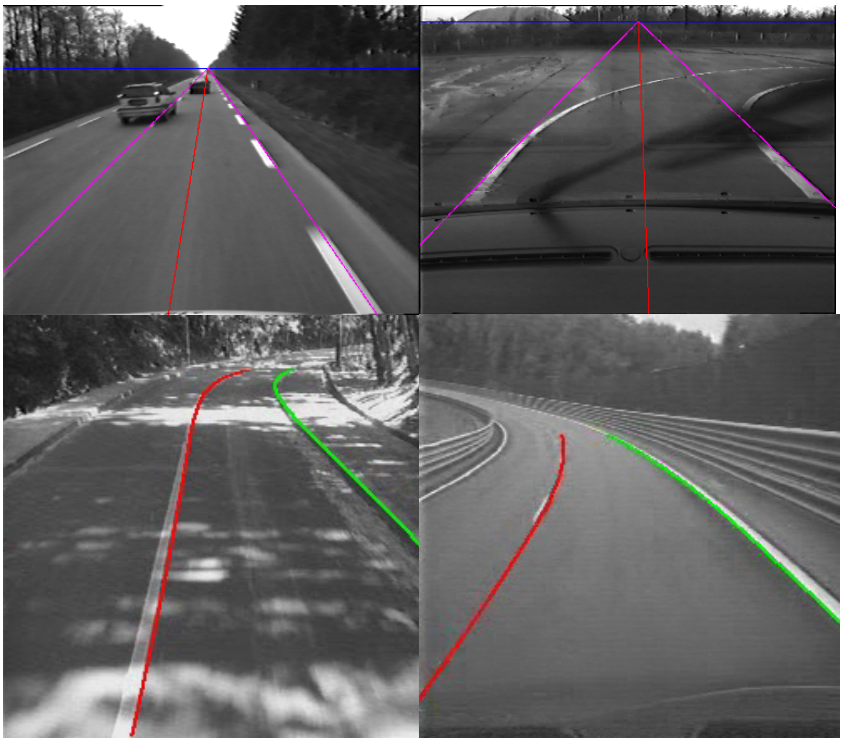


Illustration de couverture:
Exemples de résultats de détection des bords de voie sur différentes routes.

Table des matières

Résumé **5**

Introduction **7**

1 Conditions de perception **9**
1.1 Contrainte du temps réel 9
1.2 Hypothèse du monde plan 10
1.3 Variabilité de l'éclairage 10
1.4 Continuité temporelle 10

2 Résumé d'une image de route **13**
2.1 Invariance aux conditions d'éclairage 13
2.2 Extraction des segments 14

3 Transformation scène-image **19**
3.1 Image d'un point 19
3.2 Image d'une courbe explicite 22
3.3 Image d'une courbe algébrique 23

4 Positionnement latéral **25**
4.1 Approximation rectiligne de la route 25
4.2 Orientation du véhicule 25
4.3 Profil latéral de la route 27
4.4 Position latérale du véhicule 28
4.5 Continuité temporelle 29
4.6 Calibrage 29

5 Expérimentations et applications **33**
5.1 Validation sur le terrain 33
5.2 Validation avec caméra de côté 34
5.3 Validation de nuit 35
5.4 Application au contrôle latéral 38
5.5 Application à la sortie de voie 41

6 Tracé de la voie courbe **43**
6.1 Représentation d'un tracé courbe 43
6.2 Approximation récursive d'une courbe 44

7 Segmentation en courbes **47**
7.1 Principe "Best-First Segmentation" 47
7.2 "Maximum Length Criterion" 48
7.3 Détection globale 49
7.4 Régularisation a priori 51
7.5 Résultats 52

8 Détection de courbes par groupement local **57**
8.1 Critères locaux 57
8.2 Algorithme 57
8.3 Suivi temporel 58

9 Détection de la voie de circulation **61**
9.1 Combinaison de courbes 62
9.2 Régularisation et suivi 63
9.3 Résultats 65

Conclusion **65**

Annexes **69**

A Bien construire un histogramme **71**
A.1 Perturbations 71
A.2 Densité de probabilité 71
A.3 Vote discretisé 72

B Système de mesure de la position d'un véhicule par webcam de côté **75**
B.1 Description du système 75
B.2 Utilisation du système portable 75
B.3 Procédure de vérification du système portable 78

Résumé

Positionner au centre de la voie un véhicule est plus proche d'une tâche réflexe que d'une activité de planification. On peut donc espérer décharger avec intérêt le conducteur de cette tâche. Presque toute l'information utilisée actuellement par un conducteur pour se positionner latéralement est visuelle. En conséquence, pour concevoir un système de positionnement latéral automatique, sans modifier les routes actuelles, il faut commencer par l'étude des éventuels apports de la perception par une caméra embarquée sur un véhicule. Nous présentons, dans ce document, les techniques développées dans ce contexte et les résultats obtenus au sein du Livic de 1998 à 2001 au sujet de la perception des bords de la voie de circulation, avec une seule caméra.

Abstract

Lane keeping of a vehicle is closer to a reflex task rather than a higher level activity. Thus, we can expect an automatic system to substitute the driver for this task. Most of the information used presently by the driver for lane keeping comes from visual cues. Therefore without any modification of present lane markings on roads, the design of an automatic system goes by the study of vision system inboard of a vehicle. In this document, we explain the different techniques we developped for this application and describe the results obtained in Livic's team from 1998 to 2001 using a single camera.

Introduction

L'amélioration de la sécurité routière a conduit à deux types de développements : les systèmes passifs et actifs. Dans les premiers, on trouve notamment la ceinture de sécurité, la structure amortissante du véhicule, l'air-bag. Ces systèmes améliorent la protection des passagers et du conducteur en cas d'accident. Dans les seconds, on trouve l'ABS et l'ESP qui sont des systèmes actifs car ils assistent le conducteur dans des situations critiques. Cependant, le nombre d'accidents graves et mortels reste élevé.

Bien réagir à une situation critique nécessite souvent l'anticipation des problèmes. Pour cela, le conducteur utilise ses moyens de perception de l'environnement proche du véhicule. Comme presque toute l'information utilisée actuellement par le conducteur est visuelle, il faut avant tout étudier les apports de la perception par caméra.

Une caméra peut être utilisée pour plusieurs types de tâches : détection des bords de voie, reconnaissance des panneaux de signalisation, détection des obstacles, suivi des passants, etc. Nous nous sommes particulièrement intéressés à la détection des bords de voie avec une seule caméra. Notre objectif est de concevoir un système de perception des marquages routiers suffisamment fiable pour permettre le guidage automatique d'un véhicule, en latéral.

Le problème de la détection des bords de voie a déjà été abordée par différentes équipes de recherche à travers le monde [1, 2, 3, 4]. Contrairement aux approches précédentes qui ont montré la faisabilité d'une telle détection dans des conditions rarement précisées, nous avons, pour ce qui nous concerne, tenté de mieux caractériser les limites d'un système de perception avec une caméra, dans le contexte d'un guidage en sécurité sans rien changer des routes actuelles. Cela nous a conduit à développer de nouveaux algorithmes pour faire la détection des bords de voie.

Positionner au centre de la voie un véhicule est plus proche d'une tâche réflexe que d'une activité de planification. On peut donc décharger avec intérêt le conducteur de cette tâche. Etant donné que cette tâche doit être réalisée en continu, seul un positionnement par rapport à la route proche est utile, ce qui simplifie la difficulté du problème, si l'on compare avec le contrôle longitudinal.

Dans le premier chapitre, nous analyserons en détail ce problème afin de faire apparaître des hypothèses simplificatrices. Le chapitre suivant décrit le délicat problème des conditions d'éclairage et la façon dont il a été traité. Au chapitre 3, nous proposons un modèle pour la perception des bords de voie par caméra. Dans le chapitre 4, nous présentons un algorithme de détection des bords de voie rectilignes, puis, dans le chapitre suivant, sa validation expérimentale. Après avoir détaillé, dans le chapitre 6, notre modèle dans le cas des courbes, une approche globale de segmentation en courbes est décrite et illustrée dans le chapitre suivant. Une version locale, donc dégradée mais plus rapide, de l'algorithme de segmentation est indiquée dans le chapitre 8. Puis, en utilisant l'approche globale, un algorithme de détection des bords de voie,

unifiant le suivi et la reconnaissance, est présenté. Enfin, nous concluons notre étude sur la faisabilité d’un système de perception pour un guidage en sécurité.

Chapitre 1

Conditions de perception

Comme dans tous les problèmes de perception complexe, il est nécessaire d'analyser quelles sont les conditions de la perception, pour faire apparaître : les hypothèses simplificatrices qui serviront de fondement à la conception des traitements, et le type de perturbations auquel est sujet ce modèle.

Nous expliquerons d'abord la contrainte temps réel pour le guidage automatique d'un véhicule, puis décrirons les aspects géométriques de la modélisation de l'environnement routier. Ensuite, le problème de l'invariance aux conditions d'illumination sera soulevé et les hypothèses faites sur l'évolution temporelle de l'environnement routier seront introduites.

1.1 Contrainte du temps réel

Réaliser le positionnement automatique d'une voiture par rapport à sa voie nécessite des mesures à une fréquence minimale de dix images par seconde. Plus cette fréquence est grande, mieux la consigne de la commande sera suivie, et plus grande sera l'impression de confort du passager. Plus important encore, la sécurité des passagers ne peut être assurée que si des mesures fiables de positionnement sont obtenues avec une fréquence de rafraîchissement élevée, même si celles-ci ne sont pas extrêmement précises.

Afin d'aboutir à des traitements les plus efficaces possibles en terme de temps de calcul, nous avons cherché à développer une modélisation simple de l'environnement routier du véhicule. Caractéristique des problèmes de perception, les écarts aux modèles sont rarement très petits. Cela conduit souvent à des problèmes d'estimation robuste où l'on ne peut échapper à un compromis entre la précision des mesures estimées et la robustesse du traitement. Supposer un nombre minimal d'hypothèses pertinentes sur l'environnement tend à privilégier la robustesse des traitements par rapport à leur précision.

Une deuxième conséquence de la contrainte temps réel est que le traitement de chaque nouvelle image doit toujours être réalisé dans un temps limité. De fait, on ne peut utiliser que des algorithmes au nombre d'opérations borné.

1.2 Hypothèse du monde plan

Les caractéristiques intéressantes de l'environnement routier pour le guidage par vision sont principalement les lignes blanches continues et discontinues qui délimitent la voie, et éventuellement les trottoirs et autres structures. Ces structures sont pour l'essentiel des courbes approximativement parallèles au centre de la voie qui sont conçues pour permettre au conducteur de bien se positionner latéralement dans sa voie. Il est important de noter que ces courbes ont des courbures assez faibles, d'autant plus que la vitesse de circulation est importante.

Par contre, ces marquages ne contiennent que relativement peu d'indices visuels permettant d'estimer les variations de hauteur de la route. Puisqu'une seule caméra ne permet pas d'obtenir de façon directe beaucoup d'informations tridimensionnelles, nous avons fait l'hypothèse dite du *monde plan*. La position de ce plan peut être supposée approximativement fixe, ou d'inclinaison variable pour tenir compte des tangages du véhicule dus aux accélérations, aux décélérations, et aux variations de pente de la route.

Si l'on distingue deux catégories indépendantes de paramètres : la position et l'orientation du véhicule par rapport à la route et les caractéristiques locales du tracé de la route, l'estimation de ces paramètres est en pratique fortement couplée. Il faut donc porter une attention particulière au type d'informations contenues dans les marquages pour comprendre ce que celles-ci peuvent ou ne peuvent pas nous permettre d'estimer quant à la position du véhicule.

1.3 Variabilité de l'éclairage

La perception de la route est souvent perturbée par de grandes variations de contraste dues, par exemple, à l'ombre d'un pont, d'arbres, ou de parties mouillées de la chaussée. La grande variabilité des conditions d'éclairage rend l'estimation de la géométrie de la route plus délicate. Pour être robuste à ces variations d'illumination, les images doivent être traitées de façon invariante aux changements de contraste locaux, par exemple grâce à l'utilisation des courbes de niveau.

1.4 Continuité temporelle

Jusqu'à présent, seuls les aspects statiques du problème ont été décrits. Ajouter une hypothèse sur les variations des caractéristiques de la route comme sur la position du véhicule permet de : recouper les informations d'une image à l'autre pour détecter les estimations aberrantes, et limiter à un intervalle les zones de recherche des paramètres. Il est alors plus facile de faire face, de manière robuste, aux perturbations et aux écarts par rapport au modèle. Ces derniers peuvent être classés de la manière suivante :

- marquages de bord de voie discontinus, manquants, ou usés,
- marquages autres que ceux de bord de voie, comme les passages pour piétons, les flèches de déboîtement,
- traînées et salissures blanches sur la voie,

– occultations dues aux tâches lumineuses, disparitions des marquages dues aux ombres.

Ces différents types de perturbations consistent, pour l'essentiel, en la disparition d'une partie des marquages d'intérêt et l'apparition d'éléments pouvant être confondus avec les marquages d'intérêt. La première catégorie est un cas particulier de données manquantes, et la deuxième de données aberrantes.

L'hypothèse de continuité est valide la plupart du temps, mais elle n'est pas toujours vérifiée pour certaines configurations de routes, tels les croisements, les jonctions et disjonctions de voies. La validation a posteriori de cette hypothèse doit donc être réalisée entre chaque image.

Notre approche consiste donc à imposer les trois points principaux suivants :

– invariance à la variabilité de l'éclairage réalisé par exemple grâce à un résumé de chaque image sous la forme d'un ensemble de segments,

– estimation robuste aux données manquantes et aberrantes, des caractéristiques de la route, de la position et de l'orientation du véhicule,

– gestion temporelle de la continuité des paramètres estimés et validation de cette continuité.

Chapitre 2

Résumé d'une image de route

Face à la nécessité d'invariance aux conditions d'éclairage, nous avons choisi de réaliser un prétraitement de chaque image qui résume son contenu géométrique et qui est bien adapté aux images de route.

Cette étape peut être comparée à une extraction classique de contours bien qu'elle s'en différencie par plusieurs aspects importants au niveau de la gestion du bruit. En particulier, un de ses avantages est de permettre directement un filtrage géométrique du contenu de l'image, de façon très rapide. Les détails d'implantation de l'algorithme optimisé pour le traitement en séquence ont donné lieu à une demande de dépôt de brevet [5].

2.1 Invariance aux conditions d'éclairage

La plupart des algorithmes de détection de contours nécessitent au moins une étape de lissage et une de seuillage. La combinaison de ces deux étapes fait trop souvent, dans les algorithmes classiques d'extraction de contours [6, 7], une sélection sur les edgels, les éléments de base des contours, à partir d'un mélange de leurs caractéristiques géométriques et photométriques.

En effet, le lissage et le seuillage sont nécessaires pour retirer les petits détails créés par le bruit. Mais, comme le filtrage choisi est souvent linéaire en fonction des niveaux de gris, des pixels qui ont de petits gradients avec leurs voisins seront considérés a priori comme des détails lors du seuillage. Les contours avec un faible gradient seront donc perdus.

De plus, une perturbation de 10 niveaux de gris sur un pixel sera considérée comme équivalente à la perturbation d'un niveau sur 10 pixels. Les informations sur l'intensité et la taille spatiale sont donc mélangées lors du filtrage. L'extraction des contours ne peut donc pas être invariante à un changement du contraste de l'image. Par exemple, la Fig. 2.1 (b) et (c) montre les problèmes rencontrés par le détecteur de Canny-Deriche [6] appliqué sur une ligne blanche fortement perturbée par une ombre. L'importance du gradient le long de l'ombre est si forte que le lissage noie la ligne blanche que nous voulons détecter.

Au mieux, lisser devrait être suffisant (son échelle doit bien sûr être adaptée au bruit) et aucun seuillage ne devrait être appliqué. Dès lors, les contours dans les images deviennent extrêmement nombreux. Un autre critère de sélection est donc nécessaire.

Dans la Fig. 2.1 (d)(e)(f), nous en utilisons un autre qui n'est plus fondé sur des carac-

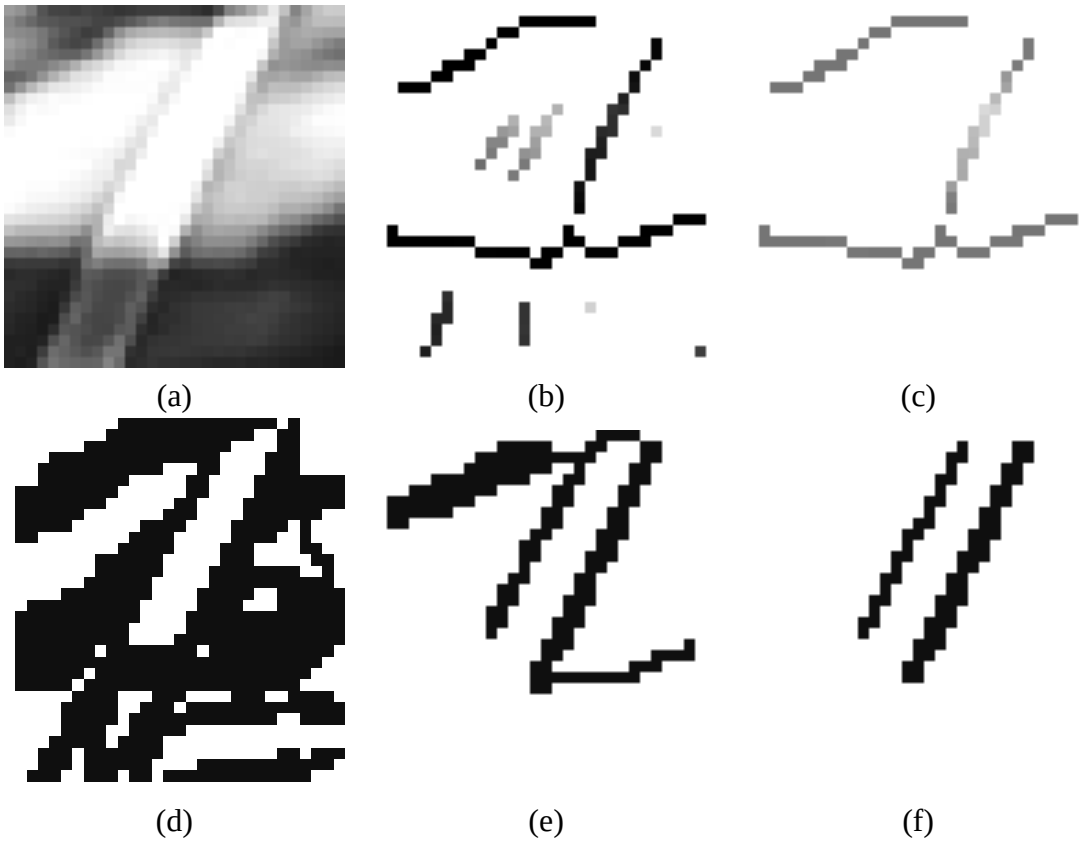


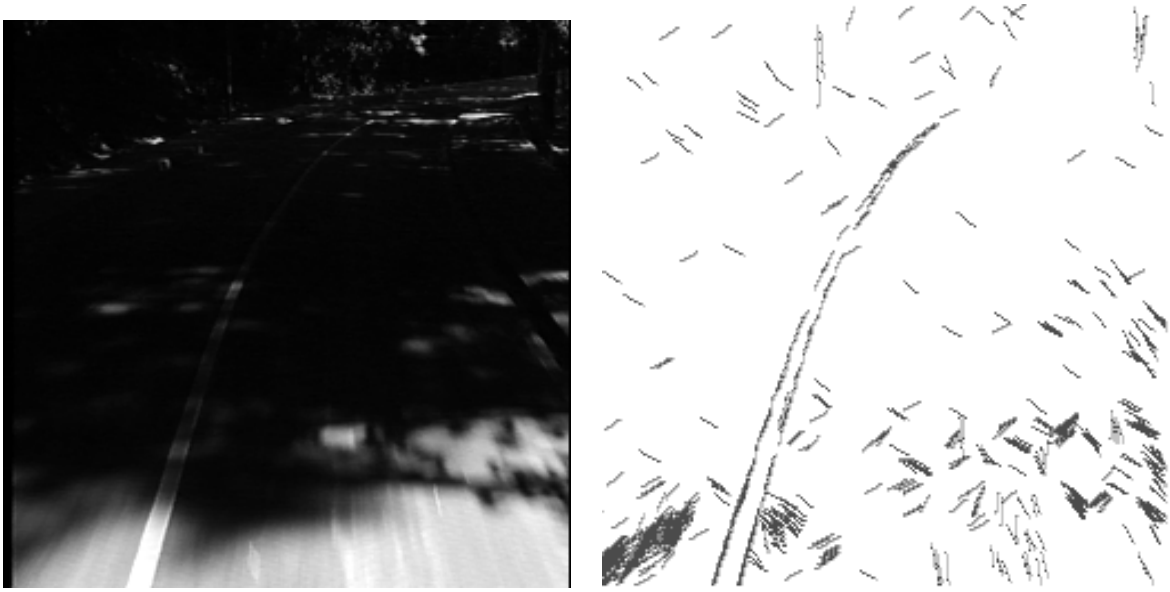
FIG. 2.1 – *(a) Image initiale perturbée par des ombres, (b) résultats du détecteur de Canny-Deriche avec un lissage de taille 1 pixel et aucun seuil sur la grandeur du gradient, (c) seuil sur les gradients inférieur à 40 niveaux de gris. Sur la deuxième ligne, résultat de notre détection des segments pour différentes valeurs de la longueur minimale: (d) 4 pixels, (e) 15 pixels et (f) 20 pixels.*

téristiques photométriques, mais sur des caractéristiques géométriques locales des images. En considérant ces caractéristiques locales, le prétraitement permet l'obtention d'un résumé invariant aux variations locales du contraste et donc des conditions d'éclairage.

2.2 Extraction des segments

Comme les marquages routiers forment des contours très réguliers et de faible courbure dans les images, nous avons choisi la caractéristique locale du contour d'ordre le moins élevé : l'équation de la tangente. Celle-ci est extraite, à cause de la digitalisation des images, comme un petit segment de droite le long du contour. Il reste alors à définir les contours de façon invariante au contraste. Nous proposons de les définir comme une ligne de niveau de l'image. Une ligne de niveau est le bord d'un ensemble de niveau. Un ensemble de niveau est l'ensemble des pixels de valeur supérieure à un certain seuil, comme cela est défini dans la thèse de Frédéric Guichard [8]. Bien sûr, d'autres types de définitions peuvent être envisagés telles les lignes où le Laplacien s'annule. La difficulté est alors de les extraire de façon invariante au changement

de contraste.



(a) (b)

FIG. 2.2 – *(a) Image originale avec un mauvais contraste et (b) résultat de l'extracteur avec une longueur minimale de 12 pixels, sans les segments proches de l'horizontale.*

Le principe d'extraction consiste à suivre, à partir de chaque pixel de l'image, les lignes de niveau qui en partent tant qu'elles restent rectilignes à la digitalisation près. Ceci est réalisé de façon rapide par l'utilisation d'un arbre qui contient toutes les chaînes codant des segments digitaux rectilignes jusqu'à une certaine longueur. Cet arbre est construit à partir de la construction récursive des chaînes de code de segments digitaux décrite dans [9]. D'après [10], on déduit qu'une estimée asymptotique du nombre de chaînes de codes de segments digitaux de longueur inférieure à N est $\frac{N^3}{\pi^2}$. Ceci induit une taille de l'arbre assez petite pour être mémorisée lors des traitements en séquence.

La Fig. 2.2 est un exemple d'images difficiles à traiter lorsque le traitement n'est pas invariant au changement de contraste. Malgré le manque de contraste, l'extracteur de segment est capable de retrouver les bords de la ligne blanche centrale.

Comme le montre la Fig. 2.3, l'extracteur de segment permet de filtrer de façon purement géométrique le contenu de l'image. Sont sélectionnés les segments dont la longueur est supérieure à un certain seuil qui spécifie l'importance du filtrage nécessaire. On peut montrer expérimentalement et théoriquement que le nombre de segments produits dans une image par un alignement fortuit de pixels bruités (voir Fig. 2.3 (b)) décroît exponentiellement avec la taille des segments.

Afin d'accélérer l'extraction, il est possible sans difficulté de restreindre l'extraction aux segments proches de la verticale de l'image. Les segments au-dessus de la ligne d'horizon sont aussi éliminés a priori. La réduction d'information engendrée par l'extraction des segments est importante et permet d'accélérer les traitements ultérieurs par rapport à une utilisation directe de l'image ou de ses contours.

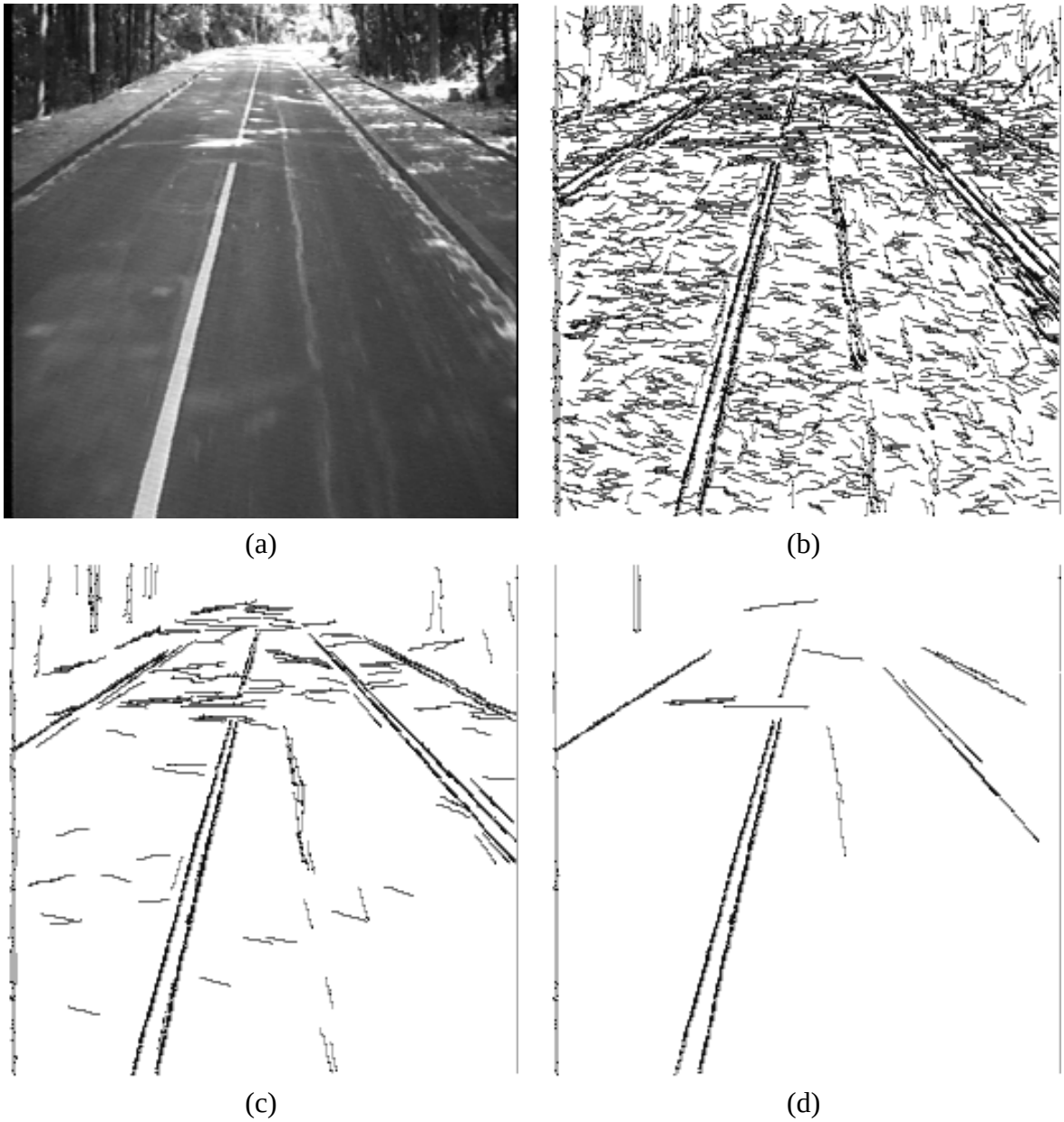


FIG. 2.3 – *(a) Image originale et résultat de l'extracteur pour différentes valeurs de la longueur minimale (b) 8 pixels, (c) 16 pixels et (d) 32 pixels.*

La sélectivité de la détection peut être forcée pour encore accélérer les traitements, mais cela se fait au détriment de l'exhaustivité du résumé du contenu intéressant de l'image. Il y a là un classique compromis entre la sensibilité et le bruit de détection. Nous avons choisi, pour faire face aux variations du contraste d'avoir une bonne sensibilité, ce qui implique l'utilisation de techniques robustes dans le traitement du résumé du contenu de l'image.

Chapitre 3

Transformation scène-image

Avant de décrire les algorithmes d'estimation de la position du véhicule par rapport à sa voie, il est maintenant nécessaire de formaliser notre modèle de la route et du véhicule. Nous commençons par décrire la relation géométrique qui existe entre la route et son image. Puis nous proposons une représentation explicite et implicite des courbes formées par les bords de voie et de leur image.

3.1 Image d'un point

Pour décrire le plus simplement possible la relation géométrique qui existe entre la route et son image, nous supposerons que la caméra peut être modélisée par une simple projection centrale sur le plan de la matrice CCD. C'est le modèle dit en trou d'épingle. Les distorsions géométriques dues à l'optique peuvent être corrigées numériquement juste après l'acquisition des images (voir par exemple [11] pour plus de détails sur ce point).

Nous avons besoin de quatre repères :

- **repère de la matrice CCD** : c'est un repère 2D orienté selon les axes de la matrice CCD avec, pour respecter les conventions habituelles en imagerie, x_m orienté vers la droite et y_m orienté vers le bas. Son origine est à l'intersection de l'axe optique de la caméra avec la matrice CCD.

- **repère de la caméra** : son origine est au foyer de la caméra, l'axe y_c est orienté selon l'axe optique, et les axes x_c et z_c selon x_m et $-y_m$, respectivement. Le centre de la caméra est placé sur l'axe longitudinal du véhicule à une certaine hauteur h. L'axe optique est supposé incliné seulement dans un plan vertical d'un angle ϕ par rapport à l'axe longitudinal du véhicule. L'angle ϕ est généralement négatif.

- **repère du véhicule** : son origine est au niveau du sol, à l'intersection de l'axe z_c avec le sol. En général, cela coïncide approximativement avec le milieu de l'essieu avant. L'axe x_v est parallèle à l'essieu, l'axe y_v à l'axe longitudinal, et z_v est selon la verticale au véhicule.

- **repère de la route** : son origine est au centre de la voie, avec l'axe z_r vertical et y_r selon la tangente au tracé du centre de la voie. La transformation rigide entre le repère route et véhicule est caractérisée par un angle θ et une translation t.

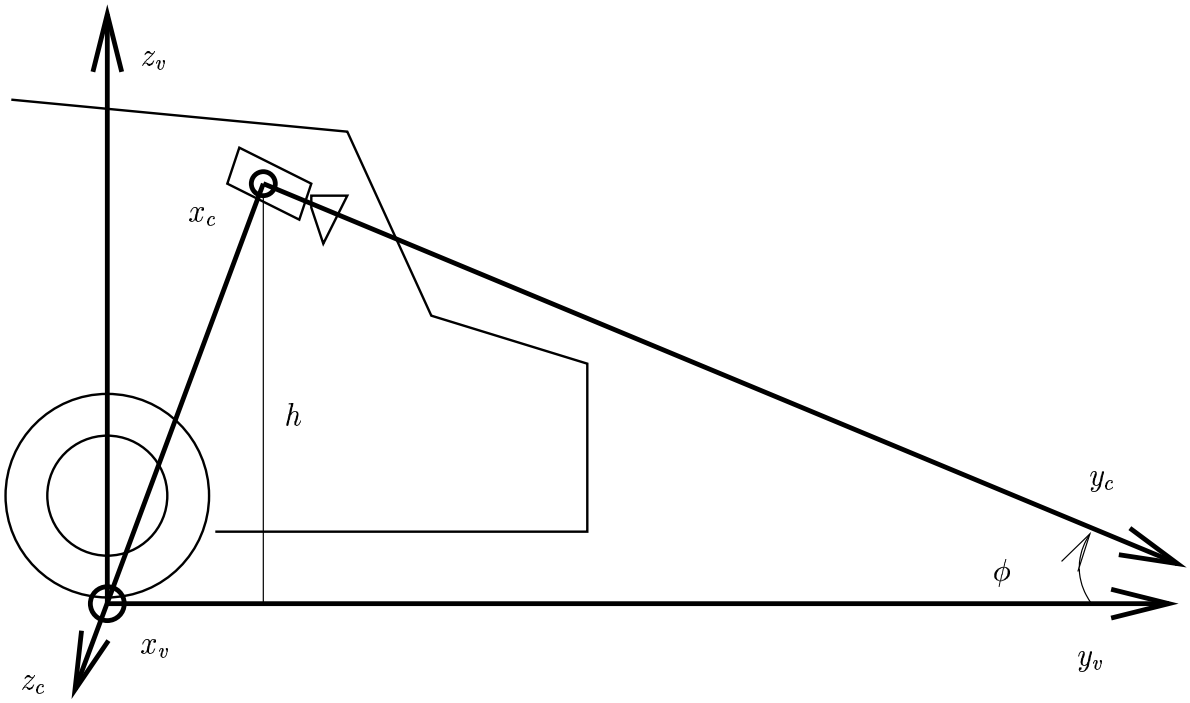


FIG. 3.1 – *Les repères de la caméra et du véhicule, vus de côté.*

Soit $(x_v, y_v, 0)$ un point au sol exprimé dans le repère du véhicule. Ses coordonnées dans le repère de la caméra, après rotation d'angle ϕ selon l'axe des x_v et translation de $-h/cos\phi$ le long de l'axe z_c, sont $(x_v, y_v cos\phi, -y_v sin\phi - \frac{h}{cos\phi})$. Les coordonnées de l'image de ce point sur la matrice CCD sont obtenues en divisant par la coordonnée selon l'axe optique puisque la caméra réalise une projection centrale, soit :

$$(x_m, y_m) = (\frac{x_v}{y_v cos(\phi)}, tan(\phi) + \frac{h}{y_v cos^2(\phi)}) \tag{3.1}$$

On remarque que notre choix d'origine du repère du véhicule a permis une simplification dans la formule précédente.

Enfin, pour obtenir des coordonnées en pixel (u, v) du point au sol, il faut introduire les paramètres intrinsèques de la caméra qui sont :

- (u_0, v_0) position, en pixels par rapport au coin supérieur gauche, du centre de l'image défini comme l'intersection de l'axe optique y_c avec la matrice CCD,

- α_u et α_v qui sont l'inverse de la taille horizontale et verticale, respectivement, d'un pixel, à la focale près.

Les coordonnées de l'image du point au sol sont donc $u = \alpha_u x_m + u_0$ et $v = \alpha_v y_m + v_0$, lorsque

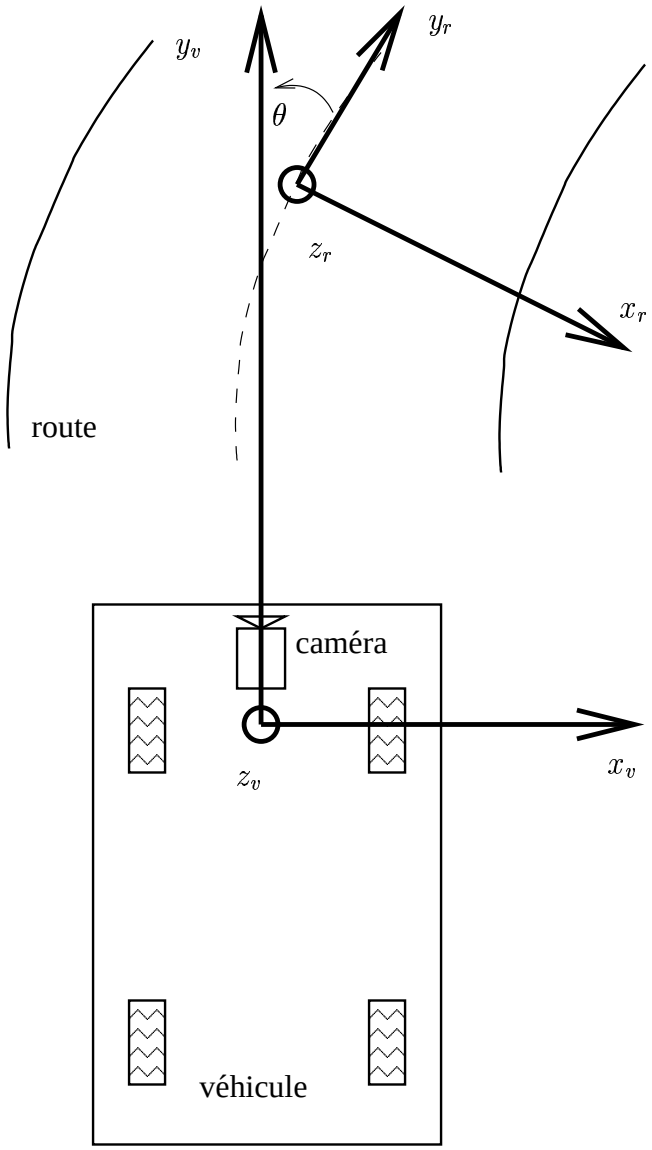


FIG. 3.2 – *Les repères du véhicule et de la route, vus de dessus.*

le coin supérieur gauche est pris comme origine. Après substitution de (3.1), on obtient :

$$\left\{ \begin{array}{lcl} u & = & \beta_u \frac{x_v}{y_v} + u_0 \\ v & = & \beta_v \frac{1}{y_v} + v_h \end{array} \right. \tag{3.2}$$

en posant $\beta_u = \frac{\alpha_u}{cos(\phi)}$, $\beta_v = \frac{\alpha_v h}{cos^2(\phi)}$, et $v_h = v_0 + \alpha_v tan(\phi)$. On remarque que lorsque le y_v du point au sol tend vers l'infini, son image converge vers un point de la ligne d'horizon d'ordonnée v_h dans l'image.

Il est important de remarquer que le système (3.2) est très simple. C'est une transformation homographique particulière (c'est-à-dire que chaque coordonnée est un rapport de transformations linéaires des coordonnées, de même dénominateur) entre les points de l'image sous la ligne d'horizon et les points du sol devant le véhicule. La transformation inverse qui fait passer d'un point de l'image à un point au sol est aussi homographique :

$$\left\{ \begin{array}{lcl} x_v & = & \frac{\beta_v}{\beta_u} \frac{u-u_0}{v-v_h} \\ y_v & = & \beta_v \frac{1}{v-v_h} \end{array} \right. \tag{3.3}$$

On note que dans cette configuration particulière, seulement quatre paramètres doivent être estimés pour complètement caractériser la transformation entre le sol et l'image : β_u, β_v, u_0, et v_h. Cette propriété peut être avantageusement exploitée lors du calibrage. Si l'on veut faire des mesures tridimensionnelles en dehors du plan de la route, il est nécessaire de faire le calibrage avec les six paramètres qui caractérisent complètement la géométrie du système : α_u, α_v, u_0, v_0, h et ϕ.

3.2 Image d'une courbe explicite

Dans la suite, nous allons supposer que u_0 et v_h sont connus. On peut donc les mettre à zéro dans (3.2). Le repère de l'image est donc positionné sur la ligne d'horizon à la verticale du centre de l'image, et non plus dans le coin supérieur gauche.

Intéressons-nous d'abord à l'image d'une droite dessinée au sol. Soit $x_v = a_0 + a_1 y_v$ l'équation des points au sol. En substituant (3.3) dans la précédente équation, on déduit que l'image d'une droite est une droite d'équation:

$$u = a_1 \beta_u + a_0 \frac{\beta_u}{\beta_v} v \tag{3.4}$$

On remarque qu'à un coefficient près, le rôle de a_0 et a_1 sont intervertis dans les équations de la droite sur l'image et au sol. Ceci implique qu'en estimant l'abscisse à l'origine de la droite sur l'image, on estime à un coefficient β_u près la pente a_1 de la droite au sol. Réciproquement, l'estimée de la pente de la droite sur l'image donne au facteur $\frac{\beta_u}{\beta_v}$ près l'abscisse à l'origine de la droite au sol a_0. Ces deux propriétés seront intensivement utilisées pour réaliser l'estimation du positionnement latéral. On obtient aussi qu'une erreur e_h d'estimation de la hauteur de l'horizon induit seulement une erreur de $-\frac{a_0}{\beta_v} e_h$ sur la valeur de a_1 de la pente. Contrairement, la valeur a_1 de l'abscisse n'est pas modifiée.

Si, au lieu d'une droite, on cherche l'image d'une courbe tracée au sol d'équation polynômiale de degré d :

$$x_v = a_0 + a_1 y_v + a_2 y_v^2 + \ldots + a_d y_v^d = \sum_{i=0}^{i=d} a_i y_v^i \tag{3.5}$$

On obtiendra son image de façon analogue en utilisant (3.3) :

$$u = b_0 v + b_1 + b_2 \frac{1}{v} + \ldots + b_d \frac{1}{v^{d-1}} = \sum_{i=0}^{i=d} b_i v^{1-i} \tag{3.6}$$

avec $b_i = a_i \beta_v^{i-1} \beta_u$. Nous nommons *polynôme hyperbolique* ce type de courbe explicite.

3.3 Image d'une courbe algébrique

Outre la droite, un autre cas particulier intéressant est celui de l'image d'un cercle au sol, vu les caractéristiques des routes. Une équation implicite du cercle est $x_v^2 + y_v^2 + a_{10} x_v + a_{01} y_v + a_{00} = 0$ où les paramètres a_{10}, a_{01} et a_{00}, sont reliés de façon bijective à la position du centre $(\frac{a_{10}}{2}, \frac{a_{01}}{2})$ du cercle et à son rayon $r^2 = \frac{1}{4}(a_{10}^2 + a_{01}^2) - a_{00}$. En utilisant (3.3), on obtient que l'image du cercle est une conique d'équation :

$$\frac{\beta_v^2}{\beta_u^2} u^2 + a_{10} \frac{\beta_v}{\beta_u} uv + a_{00} v^2 + a_{01} \beta_v v + \beta_v^2 = 0 \tag{3.7}$$

Le cercle est une courbe algébrique particulière. Ces dernières sont définies implicitement comme le lieu où un polynôme de degré d s'annule :

$$a_{00} + a_{10} x_v + a_{01} y_v + a_{20} x_v^2 + a_{11} x_v y_v + a_{02} y_v^2 + \ldots + a_{n0} x_v^d + \ldots + a_{0n} y_v^d = 0$$

ou

$$\sum_{0 \leq i,j,i+j \leq d} a_{ij} x_v^i y_v^j = 0 \tag{3.8}$$

L'image d'une courbe algébrique reste une courbe algébrique de même degré (voir [12, 13, 14] pour avoir des détails sur l'utilisation des courbes algébriques en recalage et en reconnaissance des formes). Notons que cette propriété reste vérifiée lorsque u_0 et v_h sont non nuls.

Grâce à l'équation (3.3) qui relie un point au sol et son image, nous avons pu dériver les équations des images de diverses courbes sur la route. Ces dérivations vont servir de base à divers systèmes de mesure de position du véhicule et de caractérisation du tracé de la voie que nous allons maintenant décrire.

Chapitre 4

Positionnement latéral

Nous présentons l'algorithme de positionnement du véhicule fondé sur une approximation rectiligne de la route. Après avoir détaillé les hypothèses supplémentaires que nous utilisons, le principe de l'estimation de l'orientation du véhicule et de la position latérale est expliqué. Ensuite, nous traitons de l'intégration temporelle, puis enfin du calibrage du système.

4.1 Approximation rectiligne de la route

Pour réaliser un positionnement latéral automatique du véhicule au centre de la voie, il faut que la commande dispose de mesures régulières de la position latérale du véhicule un peu en avant de celui-ci pour pouvoir anticiper les virages. Dans un premier temps, nous nous sommes intéressés à une anticipation sur une faible distance (5 mètres). Nous nous restreignons donc, dans cette partie, à la mesure de la position transversale et de l'orientation du véhicule par rapport aux marquages routiers vus entre 3 et 10 mètres. Connaître l'incertitude sur chaque donnée estimée est aussi une information importante à fournir à la commande pour qu'elle puisse être robuste.

Vu ces courtes distances, nous avons considéré que les marquages de la voie peuvent être correctement approximés par deux lignes droites parallèles. Nous avons donc conçu un premier système de perception qui analyse uniquement la portion de la route que l'on sait rester raisonnablement rectiligne, mais qui est robuste aux écarts à ce modèle, dans les virages en particulier. Contrairement aux autres algorithmes connus, il fonctionne aussi en l'absence de marquages si d'autres indices visuels sont disponibles tels les trottoirs, les bas-côtés, et les glissières de sécurité. Ceci est possible grâce à la construction et la remise à jour d'un profil latéral de la route qui contient les indications de position et d'épaisseur des différents indices au sol.

4.2 Orientation du véhicule

Nous décrivons maintenant la façon dont l'orientation du véhicule est estimée. Cette estimation est fondée sur les propriétés de l'image d'une ligne au sol décrites dans la partie 3.2. En effet, estimer la direction de la route consiste à évaluer la position sur la ligne d'horizon du point de convergence des lignes de marquages. Comme explicité en 3.2, cette position est directement proportionnelle à la tangente de l'angle entre le véhicule et la route.

FIG. 4.1 – *Histogrammes des accumulations de longueur des segments qui intersectent la ligne d'horizon. Le point de convergence des marquages est au croisement de la ligne d'horizon avec la ligne verticale qui passe par le pic de l'histogramme.*

Pour estimer la position de ce foyer avec un petit nombre de calculs, nous avons construit un histogramme sur la ligne d'horizon dont le pic marque la position du foyer. Chaque segment extrait de l'image participe à l'histogramme proportionnellement à sa longueur à l'endroit où il s'intersecte avec la ligne d'horizon. La construction de cet histogramme nécessite donc la connaissance de la hauteur d'horizon v_h. Selon la Fig. 4.1, cet histogramme consiste principalement en un seul pic, et peut être vu comme la densité de probabilité de présence du foyer sur la ligne d'horizon. En annexe A, nous expliquons comment utiliser cette interprétation pour construire l'histogramme en tenant compte des sources d'erreurs. Comme il est expliqué en annexe A, la prise en compte des erreurs des données permet d'améliorer grandement la qualité des histogrammes obtenus par rapport à une construction naïve. La médiane de l'histogramme est utilisée pour estimer de façon robuste la position du foyer, par exemple. L'étalement du pic donne une estimée de la précision de l'orientation.

FIG. 4.2 – *Histogrammes des accumulations de longueur des segments qui intersectent une ligne verticale passant par le foyer.*

On peut remarquer que, selon le même principe, l'on estime la hauteur d'horizon en mesurant la position du pic d'un histogramme construit selon une ligne verticale passant par le foyer. Un exemple de ce type d'histogramme est montré en Fig. 4.2. Comme précédemment, la médiane est utilisée pour estimer la hauteur de l'horizon. En général, la forme des pics suit la loi $l(t) = \lambda \exp(-\sqrt{1 + \frac{t^2}{\sigma^2}})$. La construction itérative de l'histogramme horizontal et vertical permet de converger vers la position du foyer.

Expérimentalement, l'orientation du véhicule par rapport à la voie est estimée de façon plus robuste lorsque la hauteur d'horizon est fixe plutôt que variable. Ceci s'explique facilement en regardant les images de route. En effet, les deux coordonnées de la position du foyer se déterminent sans ambiguïté seulement si les marquages gauche et droit sont bien visibles. Or, du fait des marquages pointillés, souvent un seul côté est observable. L'estimation des coordonnées horizontale et verticale du foyer est alors fortement corrélée.

Les variations de la hauteur d'horizon sont relativement faibles par rapport à celles de l'autre coordonnée. Nous avons donc préféré fixer la hauteur de l'horizon v_h à sa valeur moyenne pour avoir une évaluation moins sensible aux perturbations. Ceci est d'autant plus justifié qu'une erreur dans l'estimation de la ligne d'horizon n'induit théoriquement pas de biais sur la mesure de l'orientation du véhicule, comme cela a été remarqué dans la partie 3.2.

On remarque qu'il n'est nul besoin de distinguer les marquages gauche et droit pour calculer l'orientation du véhicule. En utilisant le foyer, nous pouvons tirer profit aussi bien des marquages que d'autres indices visuels comme les glissières de sécurité, les trottoirs, et les bas-côtés par exemple.

4.3 Profil latéral de la route

FIG. 4.3 – *Histogrammes des accumulations de longueur des segments en fonction de leur pente dans l'image. Les segments utilisés sont ceux qui convergent approximativement vers le foyer.*

Si, pour calculer l'orientation du véhicule, on n'a pas besoin d'identifier les marquages gauche et droite, il n'en va pas de même pour estimer sa position transversale. La difficulté réside alors dans la variabilité des caractéristiques des marquages routiers. Jusqu'ici, nous n'avons

utilisé aucune information sur la largeur de la voie, ou sur l'épaisseur des marquages. Ces paramètres dépendent du type de route et même du tronçon utilisé. L'usure et autres dégradations les modifient. Nous ne supposons pas de valeur a priori pour ces paramètres, mais nous allons les estimer à partir de l'image sous la forme d'un profil latéral de la route.

Nous nommons par profil latéral de la route la description géométrique des marquages et autres indices visuels selon une coupe transversale. Nous avons vu dans la partie 3.2 que la pente d'un segment dans l'image est proportionnelle à l'abscisse du segment au sol. Ainsi, l'histogramme des longueurs cumulées des segments en fonction de leur pente dans l'image est, à un coefficient $\frac{\beta_u}{\beta_v}$ près, le profil latéral de la route. Par exemple dans la Fig. 4.3, chaque pic correspond à un marquage. La Fig. 4.5 indique par des flèches la correspondance. Ces histogrammes ont été dédoublés pour distinguer plus facilement les contours plus sombres à droite des plus sombres à gauche. Il permet de calculer simplement le déplacement transversal du véhicule.

4.4 Position latérale du véhicule

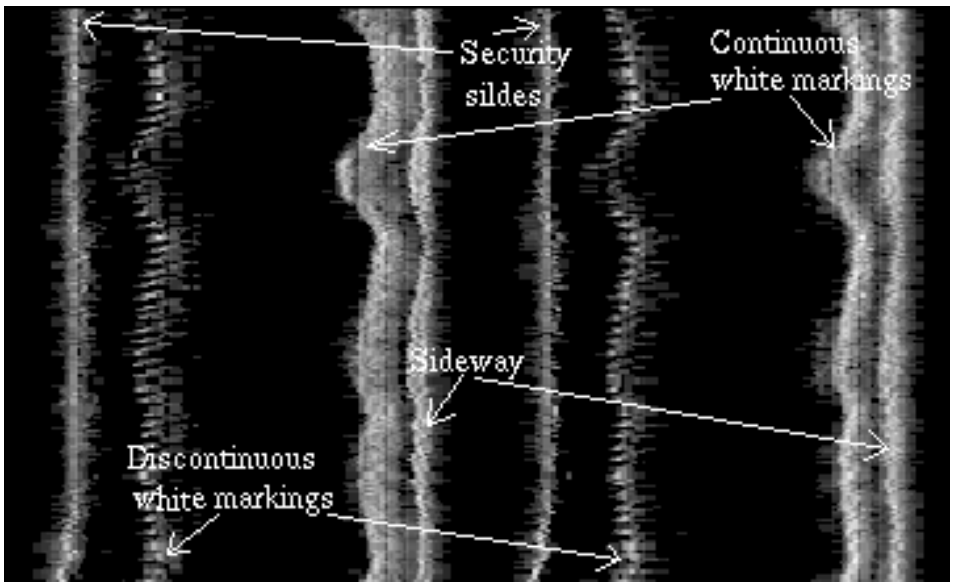


FIG. 4.4 – *Evolution au court du temps du profil de la route vue du véhicule.*

La Fig. 4.4 est un exemple d'évolution du profil de route vue du véhicule. Chaque ligne correspond à un instant différent et plus l'histogramme a des valeurs élevées, plus il est clair. Le décalage des pics entre deux lignes est proportionnel au changement de position du véhicule entre les deux instants correspondants. Par une simple corrélation des deux histogrammes consécutifs, la vitesse latérale est donc estimée.

Dès que l'on dispose d'un histogramme correspondant au véhicule centré sur sa voie, on peut estimer le positionnement latéral du véhicule par rapport au centre de la voie en utilisant une simple corrélation. Des techniques plus sophistiquées comme la programmation dynamique

peuvent être également employées pour permettre des petites variations dans les positions des pics dues à un changement de largeur de la voie par exemple.

Il reste alors à décider d'une méthode pour construire l'histogramme de référence. A l'initialisation, le véhicule est supposé bien positionné et le profil de route est calculé. Il devient la référence. Ensuite, l'algorithme estime à chaque nouvelle image les positions en calculant la translation relative entre l'histogramme de référence et le courant. L'ordre de grandeur de la précision de la mesure est obtenu en prenant l'inverse de la racine de la dérivée seconde de la fonction de corrélation au maximum.

4.5 Continuité temporelle

Comme le profil de route change plus ou moins rapidement, l'histogramme de référence doit être remis à jour dynamiquement. La remise à jour la plus simple consiste à faire une moyenne exponentielle avec le nouvel histogramme réaligné. On utilise alors l'hypothèse de variations lentes du profil de route. Pour éviter d'être trop perturbé par des contours aberrants ou trop de donnés manquantes, il est alors préférable, avant de faire cette moyenne, de vérifier que l'histogramme courant n'est pas trop différent de l'histogramme de référence. Ceci peut se faire facilement en considérant la valeur de corrélation entre les deux.

Lors du stage de Julien Estève [15], nous avons étudié de façon systématique les modes dégradés de ce système. Ils correspondent à des situations où l'image est pauvre en information et où certaines hypothèses que nous avons faites sur la présence d'un certain type d'informations ne sont plus valides. L'idée est de comparer la même variable mesurée par des moyens différents, pour voir si les valeurs obtenues sont cohérentes. L'écart entre les estimées doivent en effet être du même ordre de grandeur que la précision de la mesure. Par la même, nous validons a posteriori les hypothèses faites sur l'image. Quelques heuristiques ont été expérimentées, mais l'étude complète sur un plus grand nombre de tests reste à faire. Encore une fois, ce qui rend difficile cette étude est la relative diversité des images de route et de leur environnement.

Une approche fonctionnant bien consiste à comparer la valeur de la vitesse angulaire ou latérale obtenue en comparant le décalage estimé directement entre deux images et la différence entre les valeurs obtenues avec la référence pour les deux images. Nous avons pu observé que la valeur obtenue en comparant directement deux histogrammes consécutifs est beaucoup plus précise que l'autre. En effet, la dérivée est alors directement estimée sans passer par l'histogramme de référence.

4.6 Calibrage

Nous avons montré comment des mesures simples sur des histogrammes construits à partir des images permettent d'estimer, à un facteur $\frac{\beta_u}{\beta_v}$ près, l'abscisse à l'origine de l'axe de la route et, à un facteur β_u près, la pente de l'axe de la route. Pour pouvoir fournir des mesures dans les unités métriques, il est nécessaire de calculer les deux coefficients multiplicateurs. Le calcul de ces deux coefficients est nommé calibrage extrinsèque du système de mesure par caméra.

Plusieurs approches sont possibles. La première consiste à faire un calibrage seulement entre le plan de la route et l'image. Ceci permet de faire des mesures métriques au niveau du sol mais pas en dehors. Le calibrage consiste alors à estimer β_u, β_v, u_0, et v_h. Si le véhicule

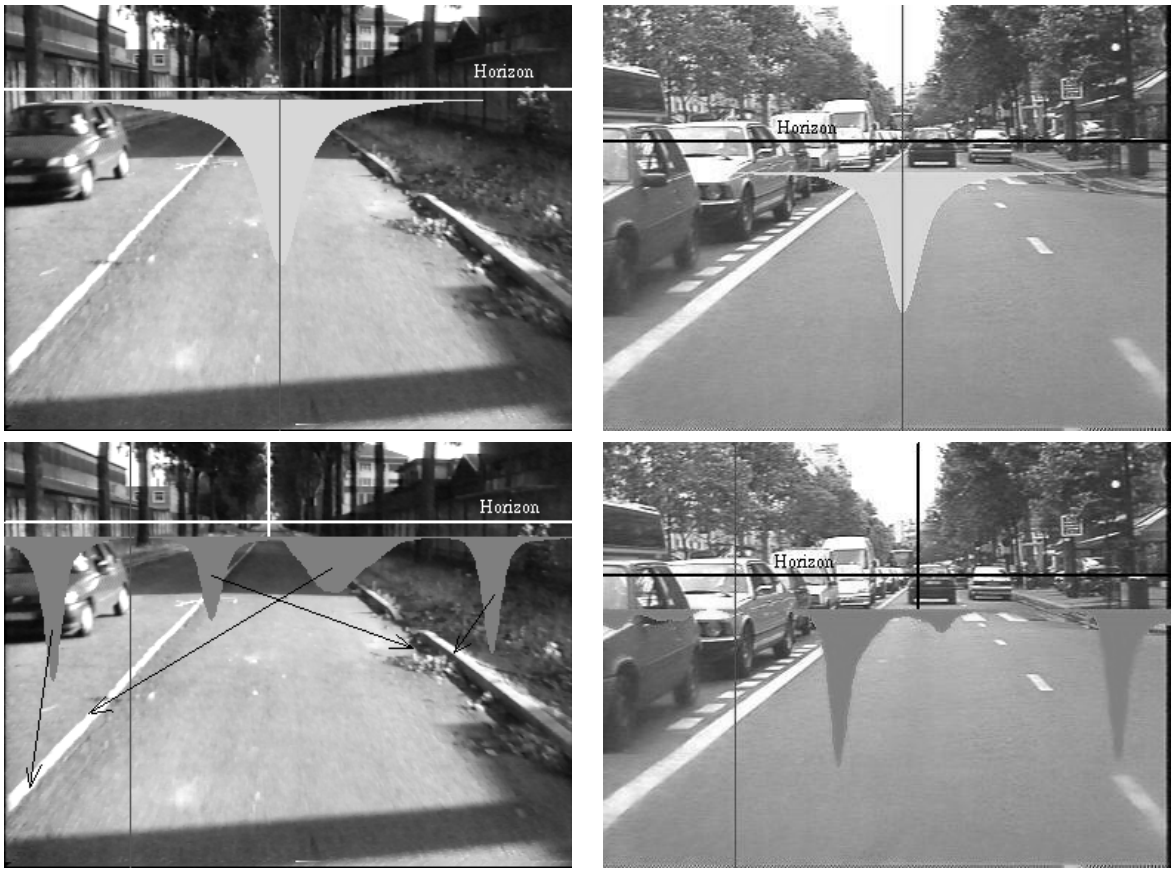


FIG. 4.5 – *Histogrammes de la position du point de fuite et du profil latéral, moyennés temporellement. Les flèches montrent l'association entre les pics et les contours. Ces histogrammes sont les histogrammes de référence utilisés dans les figures 4.1 et 4.3.*

est bien aligné sur sa voie, la deuxième coordonnée du foyer de convergence des marquages donne la hauteur de la ligne d'horizon v_h. La première coordonnée peut quant à elle être utilisée comme une bonne approximation de u_0. Cette astuce permet de compenser d'éventuelles erreurs d'alignement entre l'axe longitudinal du véhicule et l'axe optique de la caméra. Ensuite, le rapport de la largeur de la voie mesurée au sol et à partir de l'image donne directement le rapport de β_u et β_v. Enfin, la connaissance de la profondeur d'un point mesuré au sol et dans l'image permet de calculer β_v avec la deuxième équation du système (3.2).

L'inconvénient de l'approche précédente, malgré sa simplicité, est qu'elle nécessite au moins deux mesures dans la scène dont une, la mesure de profondeur, est délicate à automatiser sans l'utilisation d'une grille de calibrage au sol.

Nous avons donc cherché à automatiser au maximum le calibrage, sans avoir recourt à une grille au sol. La procédure est la suivante. Les paramètres intrinsèques de la caméra sont supposés déjà calibrés par une technique classique [11, 16]. Ensuite, dès que la caméra a été démontée ou a bougé, le conducteur doit aligner et centrer le véhicule sur une route droite et de largeur de voie connue. A partir de la seule mesure de la largeur de la voie, le calibrage peut s'effectuer.

Comme indiqué précédemment, on mesure u_0 et v_h en détectant dans l'image le foyer de

convergence des marquages. Ceci est réalisé en itérant la construction des histogrammes horizontaux et verticaux comme la partie 4.2 le décrit. La connaissance de v_h et du calibrage intrinsèque permet de calculer l'angle ϕ d'inclinaison de la caméra par rapport au sol en utilisant $tan(\phi) = \frac{v_h - v_0}{\alpha_v}$. On peut alors calculer $\beta_u = \frac{\alpha_u}{cos(\phi)}$. Le calcul de h est facile à partir du rapport $\frac{\beta_u}{\beta_v} = \frac{\alpha_u cos(\phi)}{\alpha_v h}$ et de la mesure de la largeur de la voie au sol. Les 6 paramètres qui caractérisent la géométrie du système : α_u, α_v, u_0, v_0, h et ϕ sont alors estimés.

L'avantage de cette technique de calibrage est qu'elle permet un recalibrage extrinsèque rapide, quelque soit le lieu. Inversement, comme elle repose sur un alignement manuel par le conducteur, des biais peuvent donc être introduits lors d'un calibrage peu soigneux.

Chapitre 5

Expérimentations et applications

Cette partie résume quelques expérimentations représentatives qui ont été réalisées avec le système de détection des bords de voie précédemment décrits. Ces expériences ont été menées dans différentes conditions atmosphériques et d'éclairage. Ensuite, des applications au contrôle latéral et à la détection de sortie de voie sont discutées.

5.1 Validation sur le terrain

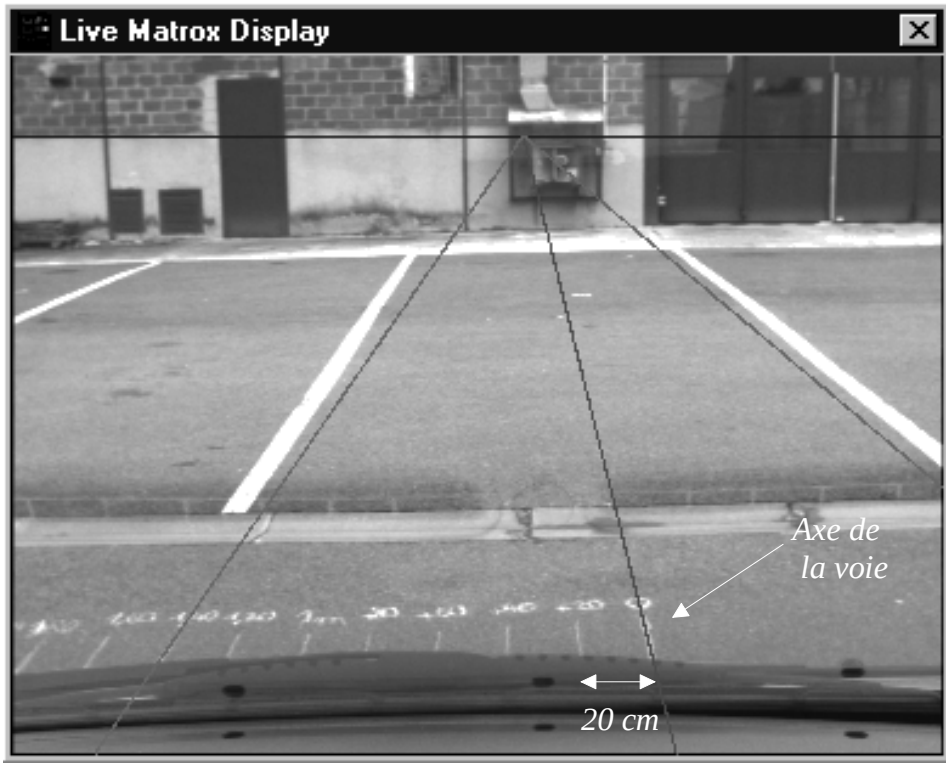


FIG. 5.1 – *Marquages sur le site de validation.*

La validation des mesures du déplacement latéral a été réalisée sur une série d'expériences

avec un véhicule équipé d'une caméra montée au niveau du rétroviseur. Les expériences décrites dans cette partie ont été réalisées par Georges Beurier et Vahé Badal Baghdassarians. Elles sont décrites avec plus de détails dans [17].

La première expérience a consisté à déplacer le véhicule transversalement au sol de $20cm$ environ, comme c'est visible dans la Fig. 5.1. Devant la difficulté d'aligner parfaitement le véhicule, et pour éviter d'ajouter des erreurs, le déplacement réel de la voiture a été mesuré à la main au niveau du sol. Le relevé de ces mesures permet la comparaison avec le déplacement estimé à partir de l'image, comme visible dans la Fig. 5.2. L'erreur commise est au maximum de $14cm$.

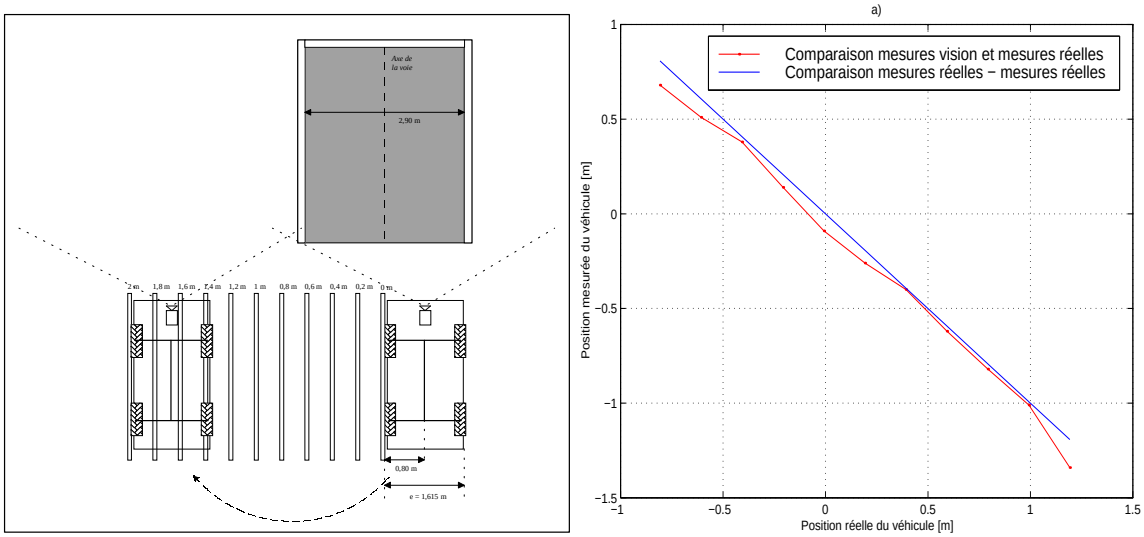


FIG. 5.2 – *Vue du site de validation et comparaison entre le déplacement latéral estimé et mesuré.*

Dans cette première série d'expériences, nous avons constaté qu'il existait un biais systématique augmentant de façon indésirable l'erreur de déplacement latéral. Celui-ci peut s'expliquer par le fait que le calibrage avait été réalisé sur une autre route. Celle du site était largement inclinée. Un recalibrage sur le site d'essai permet de diminuer la valeur maximale de l'erreur à $8.2cm$ sur une deuxième série d'expériences.

De façon similaire, les mesures d'orientation ont été effectuées en positionnant le véhicule de biais comme indiqué dans la Fig. 5.3. Le positionnement du véhicule n'étant pas exact, une erreur supplémentaire a été introduite que nous n'avons pas pu mesurer. Par exemple, une erreur de $5cm$ sur la position de la roue arrière gauche induit une erreur d'orientation de $1,1°$. De ce fait, il n'est pas possible de quantifier l'erreur commise sur l'orientation avec cette expérience. La Fig. 5.3 montre néanmoins que les orientations mesurées ne semblent pas avoir des valeurs aberrantes sur la plage entre $-15°$ et $35°$.

5.2 Validation avec caméra de côté

Pour estimer la précision du système de mesure de la position latérale et de l'orientation sur différents sites, un système portable qui permet de mesurer avec une grande précision la

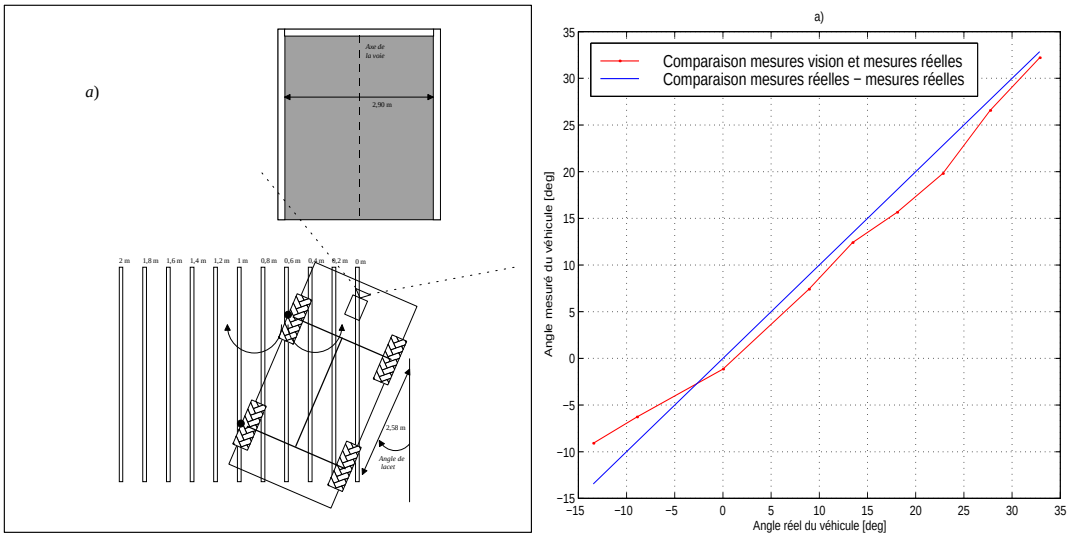


FIG. 5.3 – *Vue du site de validation et comparaison entre l'orientation estimée et imposée.*

distance du véhicule au bord de la voie a été développé (précision de l'ordre de $2cm$). Une webcam est placée sur le côté du véhicule, à la verticale de la bande blanche. La partie B détaille ce système. En comparant les données fournies par la webcam de côté avec celles de la caméra frontale, nous avons pu caractériser la précision de la mesure de position latérale en ligne droite. La comparaison dans les virages est moins facile car les deux caméras ne mesurent plus exactement la position par rapport au même point le long du tracé de la route.

Les écarts types observés (voir Fig. 5.4) en ligne droite sont de $10cm$ pour la position latérale et de $0.2°$ pour l'orientation du véhicule. Nous avons aussi pu estimer, en moyenne, les biais dus aux défauts d'alignement lors du calibrage par le conducteur : $10cm$ pour la position latérale et $0.5°$ pour l'orientation du véhicule. On note que le système fait aussi bien, si ce n'est mieux que le conducteur. Nous avons observé que le système avec une webcam de côté permet une mesure très précise, à $2cm$ près, de la position latérale du véhicule qui peut servir de référence. Par contre, l'orientation estimée est beaucoup plus précise quand elle est obtenue par la caméra frontale. Ceci s'explique simplement. En effet, les tracés de marquages ne sont pas très réguliers, l'erreur sur l'estimation de l'orientation va donc être directement liée à la longueur observée lors de la mesure. La caméra frontale fait une mesure moyenne sur environ $7m$ alors que la webcam de côté fait une mesure moyenne sur seulement $1m$. Le système avec webcam ne peut donc pas fournir de référence pour la mesure de l'orientation, à moins d'utiliser un champ beaucoup plus large et donc de dégrader fortement la précision de la position latérale.

5.3 Validation de nuit

La précision du positionnement dépend de la densité des marquages routiers et de l'éclairage de la scène. Pour quantifier l'influence de ces deux paramètres sur la précision des mesures de la position et de l'orientation du véhicule, Pierre Charbonnier du LRS a réalisé une série d'essais sur une route rectiligne de jour, au coucher de soleil, et la nuit.

La Fig. 5.5, montre des scènes routières proches, selon différentes conditions de luminosité

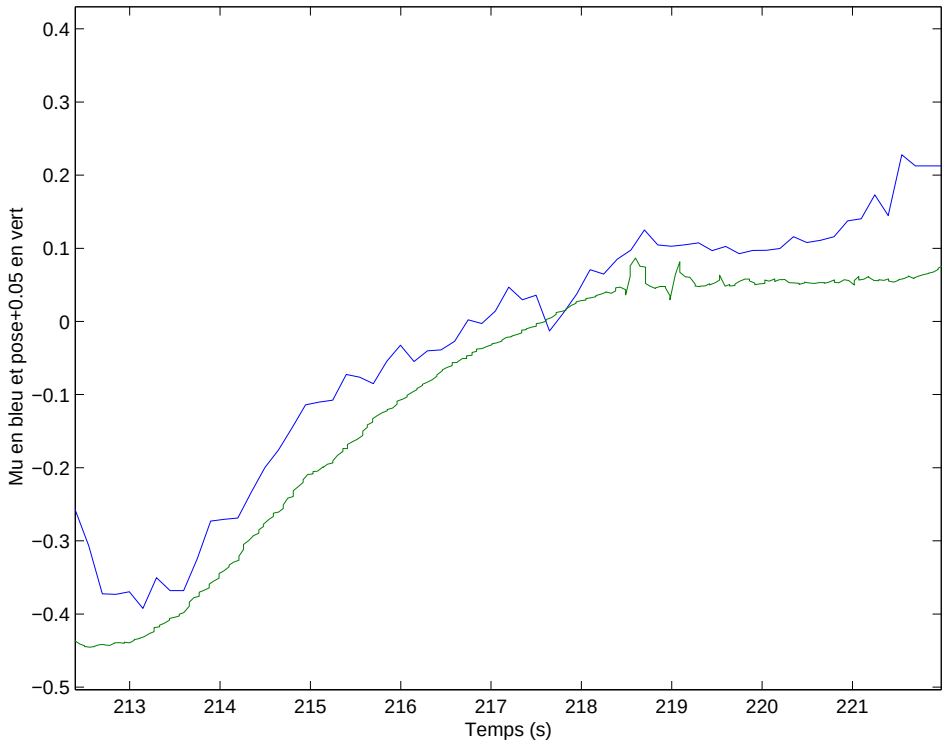


FIG. 5.4 – *Comparaison des mesures d'écart latéral obtenues par la caméra frontale (haut) et la webcam (bas).*

extérieure. Ces images illustrent bien le fait que, grâce aux phares du véhicule, la perception des marquages est plus facile la nuit que le jour. On remarque aussi que les images de jour sont beaucoup plus perturbées par le décor, alors que sur les images de nuit, seules les structures rectilignes sont des marquages. Des structures rectilignes parasites n'apparaissent sur les images de nuit que lorsqu'un véhicule plein phare sature la matrice CCD selon une droite verticale (effet de "smearing"). Expérimentalement, nous avons observé qu'effectivement, les mesures de position et d'orientation obtenues sur les séquences de nuit sont plus concentrées selon un nuage Gaussien, que dans le cas des séquences de jour.

erreur de jour	position	angle	erreur de nuit	position	angle
densité 34%	$7.1cm$	$0.28°$	densité 34%	$7.6cm$	$0.26°$
densité 60%	$4.6cm$	$0.25°$	densité 60%	$5.1cm$	$0.31°$
densité 73%	$5.6cm$	$0.16°$	densité 73%	$6.2cm$	$0.19°$

TAB. 5.1 – *Comparaison des erreurs de mesure de l'écart latéral et de l'orientation de jour et de nuit, en fonction de la densité des marquages gauches et droits.*

Pour qualifier la précision des mesures de jour et de nuit, leur écart type a été estimé. Il donne une information pertinente, car lors de l'acquisition de ces séquences sur ligne droite,

FIG. 5.5 – *Scènes routières similaires avec un éclairage de jour et de nuit.*

le conducteur pouvait s'aligner assez précisément par rapport aux marquages grâce au système Ecodyn. Comme le montre le Tab. 5.1, les précisions de jour et de nuit sont similaires, même si les mesures sont moins perturbées de nuit. Les variations entre les résultats de jour et de nuit peuvent être attribuées en partie à la présence de véhicule et de camions plus ou moins perturbants. Par contre, la précision varie de façon relativement importante suivant la densité des marquages pointillés, comme le présente le Tab. 5.1 où nous avons distingué trois types de marquages : T1 à droite et T2 à gauche qui correspondent à une densité moyenne des marquages de 34%, T3 et T2 de densité moyenne 60%, et Continu et T2 de densité moyenne 73%. L'augmentation de la précision est particulièrement significatif sur l'orientation du véhicule.

Enfin, une densité des marquages de 34% est suffisante pour avoir une relativement bonne précision sur les mesures, à une vitesse de conduite normale. Mais, il est important de remarquer que lorsque la vitesse est beaucoup plus basse que recommandée, les espaces entre les marquages sont vus de façon répétée un plus grand nombre de fois. Ceci peut finir par perturber gravement la construction des histogrammes de référence et conduire à des mesures éronnées.

5.4 Application au contrôle latéral

Le système de perception frontal est implanté depuis le début de l'année 2000 sur le premier véhicule expérimental du Livic. Ce dernier possède une colonne de direction commandable comme décrit dans la Fig. 5.6.

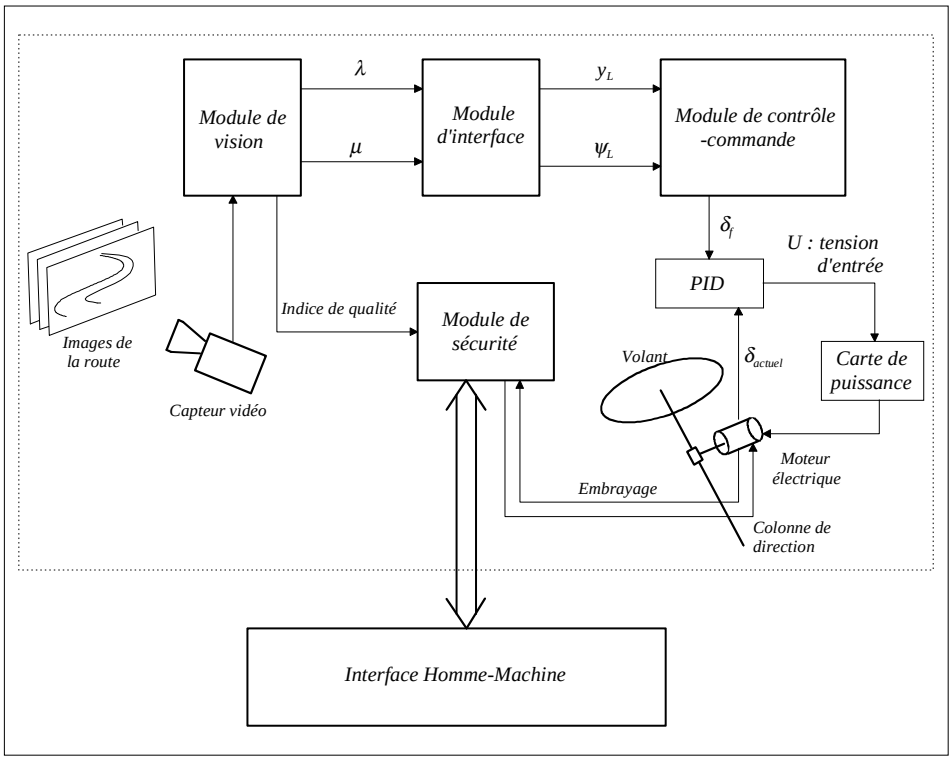


FIG. 5.6 – *Synoptique du système de contrôle latéral.*

Le véhicule est équipé d'une caméra frontale, placée à côte du rétroviseur, pointée vers l'avant, comme c'est visible sur la Fig. 5.7. La caméra fournit des images au format moitié du format vidéo. La caméra observe la route juste en avant du capot, soit à $l_{min} = 3m$, jusqu'à une distance de $l_{max} = 10m$. L'algorithme précédemment décrit fait une estimation moyenne de l'abscisse a_0 et de la pente a_1 de l'axe de la route. Cette moyenne est réalisée au niveau de l'image et non de la route. Ceci permet de mesurer l'abscisse et la pente de la tangente à l'axe de la voie à la distance de $l_v = 4.7m$ dans le repère véhicule. Cette distance $l_v = \frac{1}{2(\frac{1}{l_{min}}+\frac{1}{l_{max}})}$ est la moyenne harmonique, et non arithmétique, de l_{min} et l_{max} puisque dans (3.2) les points de l'image sont des fonctions inverses de la profondeur y_v. Une distance de $5m$ était requise pour une bonne anticipation de la commande du volant.

Un PC à 400 puis 800 Mhz est embarqué dans le véhicule (voir figure 5.7). Il réalise le traitement sur chaque nouvelle image à un rythme de 10 Hz environ en parallèle avec la commande. La digitalisation est réalisée par un carte Matrox II MC. Le logiciel qui traite les images est écrit en C.

Des détails sur les différentes commandes qui ont été testées et sur quelques-unes des nombreuses expérimentations réalisées par Georges Beurier sur les pistes du Giat sont disponibles

FIG. 5.7 – *Caméra, écran, et PC du véhicule expérimental.*

dans [17]. En particulier, le système n'est pas trop perturbé par temps de pluie, malgré le passage de l'essuie-glace.

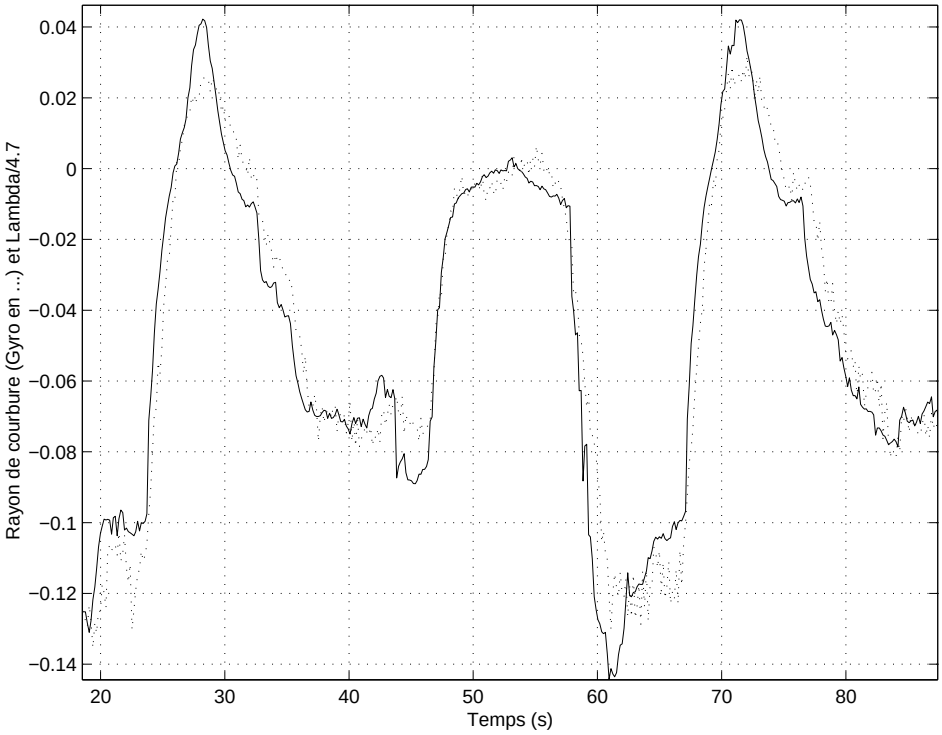


FIG. 5.8 – *Comparaison entre le rayon de courbure estimé avec le gyroscope (pointillés) et la pente de l'orientation du véhicule (plein).*

Nous avons pu aussi valider la mesure de l'orientation par un moyen détourné. Elle a été comparée au rayon de courbure estimé grâce aux données fournies par le gyroscope. En effet, il est aisé de montrer [17] que la pente a_1 de l'orientation du véhicule par rapport à la route et la courbure de la route c sont liées par $c = \frac{a_1}{l_v}$ lorsque le véhicule suit parfaitement le centre de sa voie. La Fig. 5.8 montre la comparaison entre les deux courbures estimées par le gyroscope et par l'image lors d'une expérience sur piste où le conducteur roulait bien au centre de la voie. Les deux courbes se superposent relativement bien malgré l'importance des courbures des virages.

FIG. 5.9 – *Images où les marquages sont difficilement visibles à cause : de l'effet d'éblouissement sur une chaussée bétonnée, du brouillard, de l'effet miroir après la pluie, ou d'un virage très serré.*

Ces expérimentations ont permis de cerner les limites intrinsèques d'un système de perception avec une seule caméra pour une utilisation en contrôle latéral. La Fig. 5.9 montre de telles situations. Le premier problème reste celui de l'éblouissement malgré le traitement invariant au changement de contraste. Nous avons en effet été surpris de constater la fréquence avec laquelle la qualité des images est dégradée par les conditions d'éclairage. Elles ne contiennent alors plus suffisamment d'informations pour faire le positionnement de façon fiable. L'utilisation de caméra CMOS de qualité, de part leur gain logarithmique, sera sûrement un progrès.

Une autre source importante de dégradation est le brouillard. Différents algorithmes pour estimer la distance de visibilité ont été conçu par Jean Lavenant lors de son stage DEA [18]. L'un de ces algorithmes a donné lieu au dépôt d'un brevet [19]. La pluie seule ne provoque pas trop de perturbations, si les essuies glaces ne sont pas trop usés. Mais l'effet miroir produit par la présence d'eau sur la route est lui très perturbant, dans le cas par exemple d'un soleil rasant, de face.

Une autre limitation est due au champ de vue limité des caméras. Un format plus allongé horizontalement que le format vidéo serait préférable. Ceci est nécessaire pour bien voir les marquages lors de virages très serrés (dernière image de la Fig. 5.9).

5.5 Application à la sortie de voie

Une application potentielle d'un système de mesure de la position d'un véhicule par rapport à sa voie est le déclenchement d'une alarme en cas de sortie de voie. Ce qui différencie cette application du contrôle latéral, c'est qu'il faut tenir compte du comportement du conducteur qui est difficile à connaître et à modéliser. Néanmoins, le temps de réaction d'un automatisme est beaucoup plus court que celui d'un être humain. Il est donc facile d'utiliser le système de vision décrit précédemment pour faire un système d'alerte. Il suffit d'y ajouter un seuil sur la distance latérale estimée. Celui-ci déclenche une alarme sonore dès que le véhicule roule trop près d'un marquage de bord de voie. L'alarme ne doit bien sûr être déclenchée que lorsque la distance latérale est estimée avec suffisamment de précision, pour éviter les fausses alarmes lorsque les marquages manquent.

Comme le temps de réaction du conducteur est assez long, habituellement 0.5 secondes, et que la mesure de l'écart latéral est estimée à partir des données à $5m$, on peut considérer que le conducteur est prévenu trop tardivement avec un tel système.

Lors du stage d'Alexandru Grosu, nous avons étudié la possibilité de pallier ce problème en utilisant les informations de vitesse et d'accélération longitudinales et latérales fournies par des capteurs proprioceptifs embarqués dans le véhicule [20]. A partir de ces expérimentations, nous avons pu montrer que pour prévenir le conducteur toujours suffisamment en avance, il faut pouvoir détecter le tracé de la route assez loin dans l'image, jusqu'à $20m$, distance parcourue en 0.5 seconde à $130km/h$. Nous avons observé qu'à $5m$ la précision est de $10cm$, et elle se dégrade proportionnellement à la distance. Il est donc beaucoup plus difficile de prédire à $20m$, sans fausse alarme, si le véhicule sera dans sa voie ou non vu l'imprécision sur les estimations. Cet inconvénient est accentué par le fait que plus la prédiction est éloignée dans le temps, plus les changements intermédiaires de consigne du conducteur produiront de fausses alarmes.

En conclusion, le premier système d'alarme, même s'il est relativement tardif, semble préférable pour minimiser les fausses alarmes. Cela nous semble être le meilleur compromis possible pour un système d'alarme. Enfin, il faut noter que les systèmes d'assistance sont plus intéressants de ce point de vue, car ils ne sont pas limités par le temps de réaction du conducteur.

Chapitre 6

Tracé de la voie courbe

Si le positionnement latéral automatique au centre de la voie nécessite au minimum l'écart latéral entre le véhicule et la route à $5m$, nous avons constaté que l'utilisation de l'orientation du véhicule permet d'avoir une commande plus douce et donc un sentiment de plus grand confort pour les passagers. De le même façon, la connaissance de la courbure de la route est utile. Il faut noter que l'ajout de ce troisième paramètre permet d'utiliser le système de perception avec la caméra frontale aussi pour le contrôle longitudinal du véhicule.

Pour pouvoir mieux traiter du cas des courbes, nous détaillons leur représentation et leur estimation. En particulier, l'estimation récursive d'une courbe est très utile.

6.1 Représentation d'un tracé courbe

Comme expliqué dans les parties 3.2 et 3.3, nous pouvons décrire le tracé de la route par une courbe explicite en y_v ou implicite en x_v et y_v. Les quatre avantages principaux du choix des courbes polynômiales implicites pour représenter le tracé de la route sont :

- la courbe qui approxime le tracé de la route est une fonction linéaire de ses paramètres,
- l'image d'une courbe de ce type vue au sol est aussi une courbe de ce type, même lorsque l'horizon est variable.
- si, sur une image, le tracé de la route est bien approximé par une courbe de ce type, il l'est aussi sur une autre image, quelque soit le mouvement intermédiaire de la voiture,
- les lignes et les cercles sont des cas particuliers simples de cette représentation.

Pour autant, nous avons préféré utiliser la représentation explicite pour avoir un traitement plus rapide. De plus, cette représentation empêche la détection des tracés de route selon des courbes qui font un demi-tour. La représentation explicite a le même premier avantage, et le même deuxième avantage mais seulement lorsque l'horizon est connu. Le troisième avantage reste vrai seulement si le véhicule se translate sans tourner. Enfin les lignes sont un cas particulier simple des courbes explicites mais pas les cercles. Néanmoins, sur une partie, il est possible d'approcher l'image d'un cercle par une courbe explicite. Pour ce faire, en négligeant le terme

en u^2 dans (3.7), on obtient l'équation explicite en v suivante :

$$u = -\frac{\beta_u a_{00}}{\beta_v a_{10}} v + \frac{\beta_u a_{01}}{a_{10}} + \frac{\beta_u \beta_v}{a_{10}} \frac{1}{v}$$

que l'on peut comparer à l'équation de l'image d'un polynôme explicite de degré deux :

$$u = b_0 v + b_1 + b_2 \frac{1}{v}$$

Ceci permet de relier de façon approchée la représentation implicite et explicite pour le cas du cercle et donc de calculer une valeur approchée du rayon à partir de la représentation explicite de degré deux soit :

$$r^2 = \frac{\beta_v^2}{4b_2^2}(\beta_u^2 + b_1^2 - 4\beta_v^2 b_0^2)$$

De même, à partir d'un polynôme hyperbolique explicite, on peut obtenir la position du centre du cercle approchant la route. Les termes b_0 et b_1 jouent le même rôle que $a_1\beta_u$ et $a_0\frac{\beta_u}{\beta_v}$ dans l'équation de l'image de la droite (3.4). Comme dans le cas de la droite, b_0 est, au facteur β_u près, la pente de la tangente au cercle au niveau du véhicule. Enfin, b_1 est, au facteur $\frac{\beta_u}{\beta_v}$ près, l'abscisse à l'origine de la tangente au cercle au niveau du véhicule. En conclusion, les paramètres b_0, b_1, et b_2 permettent de retrouver la courbure de la voie, la position et l'orientation du véhicule sur sa voie.

Pour ne pas avoir à distinguer à chaque fois entre la classe des polynômes et des polynômes hyperboliques, nous décrivons ces deux classes de manière unifiée en utilisant la notation vectorielle :

$$u = \sum_{i=0}^{i=d} f_i(v) a_i = F(v)^t A \tag{6.1}$$

ou (u, v) est un point dans l'image sur la courbe, $A = (a_i)_{0 \leq i \leq d}$ est le vecteur de paramètres de la courbe, et $F(v) = (f_i(v))_{0 \leq i \leq d}$ est le vecteur des fonctions de base de la classe choisie.

6.2 Approximation récursive d'une courbe

Nous allons, dans cette partie, montrer comment profiter de la linéarité des classes de fonctions que nous avons choisies (les polynômes et les polynômes hyperboliques) pour estimer récursivement les paramètres de la courbe qui approxime au mieux un contour. L'utilisation de la version récursive permet en effet des gains drastiques en temps de calculs lors de la détection d'une courbe dans une image. La difficulté de cette détection vient de la présence d'autres types de courbes non reliés aux marquages.

Comme décrit dans la partie 2, chaque image est résumée sous la forme d'une liste de segments décrits par la position de leurs deux points extrémités. Bien qu'il soit possible de directement utiliser les segments comme des tangentes lors de l'approximation d'un contour par une courbe (voir [21]), nous avons préféré réduire chaque segment à ses deux seuls sommets pour une présentation plus générique. Dans la suite, nous ne considérons que des éléments de contour qui sont, soit des points, soit des segments. Un segment est représenté par une paire de points et pour tenir compte de la longueur l_i du segment, chaque point est pondéré par l_i.

Lorsque les éléments de contours sont des points, on considère implicitement que les poids l_i sont tous égaux à 1.

La façon la plus simple d'approximer un ensemble de points $\{p_i\}_{1\leq i\leq m}$ par une courbe explicite est de minimiser l'écart au sens des moindres carrés entre la courbe et les points $p_i = (u_i, v_i)$:

$$e_m^{fit} = \sum_{1\leq i\leq m} (F(v_i)^t A_m - u_i)^2$$

où m est le nombre de points, et A_m est le vecteur des paramètres de la courbe qui approxime les m points au mieux. Si l'on veut tenir compte de la pondération, il faut introduire les l_i de la manière suivante :

$$e_m^{fit} = \sum_{1\leq i\leq m} l_i(F(v_i)^t A_m - u_i)^2 \tag{6.2}$$

La minimisation de l'erreur d'approximation précédente conduit aux équations normales bien connues, obtenues par dérivation de l'erreur :

$$M_m M_m^t A_m = M_m U_m \tag{6.3}$$

où $U_m = (l_i u_i)_{1\leq i\leq m}$ est le vecteur des premières coordonnées des m points pondérés par l_i, la matrice $M_m = (l_i F(v_i))_{1\leq i\leq m}$ est connue sous le nom de *design matrix*, la matrice $S_m = M_m M_m^t$ est quant à elle la *scatter matrix*. En introduisant $G_m = M_m U_m$, alors (6.3) s'écrit $S_m A_m = G_m$. Le calcul de la meilleure solution consiste donc simplement à résoudre le système linéaire (6.3).

Contrairement à [22], où la version récursive utilise une décomposition QR de M_m, qui est de taille $d+1 \times m$, nous nous sommes placés dans le contexte du filtrage de Kalman. Ce choix permet un gain en temps de calculs et de mémoire, car nous utilisons seulement S_m^{-1} qui est de taille $d+1 \times d+1$. Le filtrage de Kalman, dans sa plus simple expression, est fondé sur la propriété suivante qui permet la mise à jour de la solution :

$$(S + FF^t)^{-1} = S^{-1} - \gamma S^{-1} F F^t S^{-1} \tag{6.4}$$

avec $\gamma = (1 + F^t S^{-1} F)^{-1}$. Cette équation (6.4) suppose que la matrice S est inversible et découle du fait que F est un vecteur de taille $d+1$. A partir de (6.4), nous déduisons le calcul récursif de $K_{m+1} = S_{m+1}^{-1}$:

$$K_{m+1} = K_m - \gamma_{m+1} K_m F_{m+1} F_{m+1}^t K_m \tag{6.5}$$

avec $\gamma_{m+1} = l_{m+1}(1 + l_{m+1} F_{m+1}^t K_m F_{m+1})^{-1}$. Cette dernière équation donne la matrice de *covariance* K_{m+1} de A_{m+1} comme une fonction de la précédente matrice de covariance K_m et du vecteur F_{m+1} correspondant au nouveau point à approcher, pondéré par par l_{m+1}.

Ensuite, les nouveaux paramètres sont obtenus par :

$$A_{m+1} = A_m + l_{m+1} K_{m+1} F_{m+1}(u_{m+1} - A_m^t F_{m+1}) \tag{6.6}$$

En conséquence, la version récursive de l'estimation des paramètres de la courbe consiste à :

– Initialiser K_0 avec la matrice identité multipliée par un facteur r très petit, et initialiser A_0 à zéro.

– Pour chaque nouveau point (u_{m+1}, v_{m+1}), mettre à jour la matrice de covariance K_m avec (6.5) et mettre à jour les paramètres de la courbe A_m grâce à (6.6).

Le précédent choix de K_0 garanti que l'équation (6.5) peut être appliquée sans difficulté tant que r est strictement positif et ceci même si le nombre de points n'est pas suffisant pour contraindre complètement les paramètres de la courbe dans (6.2). Initialiser ainsi K_0 est équivalent à faire une régularisation nommée *Ridge Regression* que nous avons introduite dans le contexte de l'approximation non-récursive de courbes algébriques implicites [12]. Initialiser K_0 consiste à introduire un a priori sur la solution. On peut donc avantageusement utiliser d'autres façons d'initialiser K_0. Nous verrons plus loin comment cette dernière permet de faire la poursuite d'une courbe d'une image à l'autre.

On remarque aussi que l'erreur d'approximation e_{m+1} peut être récursivement calculée sans nécessiter le calcul du nouveau vecteur A_{m+1} ni même la nouvelle matrice de covariance K_{m+1}, en utilisant :

$$e_{m+1}^{fit} = e_m^{fit} + \gamma_{m+1}(u_{m+1} - A_m^t F(v_{m+1}))^2 \tag{6.7}$$

qui peut être obtenue en substituant (6.5) et (6.6) dans (6.2). Cette propriété s'avère d'une grande importance pour accélérer la détection des courbes, comme nous allons le voir maintenant.

Chapitre 7

Segmentation en courbes

Nous venons d'expliquer comment approximer au mieux un ensemble d'éléments de contours par une courbe explicite de manière directe et récursive. Ceci suppose que nous avons été capables de sélectionner l'ensemble de ces éléments de contours de façon à ce qu'ils soient sur une courbe, c'est-à-dire que la segmentation en courbe ait déjà été effectuée. Dans cette partie, nous expliquons une façon de faire cette segmentation fondée sur une utilisation astucieuse de l'estimation récursive d'une courbe et le principe que nous avons nommé "Best-First Segmentation" [23].

L'algorithme s'inspire des approches par regroupement perceptuel [24, 25, 26]. Mais, contrairement à ces techniques qui utilisent des indices locaux, nous imposons une contrainte globale sur chaque sous-ensemble d'éléments de contours : ils doivent former une courbe d'une classe donnée a priori.

Après avoir exposé le principe "Best-First Segmentation", nous approfondissons la question du choix de la fonction d'erreur à minimiser et introduisons le "Maximum Length Criterion" (MLC). Ensuite, nous présentons l'algorithme de segmentation, et détaillons plus particulièrement la phase d'initialisation qui permet de régulariser la solution. Enfin, le fonctionnement de l'algorithme est illustré par quelques résultats.

7.1 Principe "Best-First Segmentation"

La segmentation en courbe consiste à partitionner l'ensemble des contours en sous-ensembles, chacun pouvant bien être approximé par une courbe appartenant à une classe $\mathcal{S}$ donnée. Cette classe sera pour nous soit la classe des polynômes de degré d, soit la classe des polynômes hyperboliques de degré d.

Introduisons l'erreur d'approximation e^{approx} d'un ensemble d'éléments de contours $\{p_1, \ldots, p_m\}$ par rapport à la courbe de $\mathcal{S}$ qui les approxime le mieux. Cette erreur peut, par exemple, être une fonction croissante de e^{fit} :

$$e^{fit}(p_1, \ldots, p_m) = \min_{s \in \mathcal{S}} \sum_{i=1}^{m} d^2(p_i, s)$$

où $d(p_i, s)$ est la distance entre la courbe s et l'élément de contour p_i. Une erreur d'approximation importante indique que les données ne sont pas bien représentées par une courbe dans $\mathcal{S}$.

L'erreur e^{approx} est nécessairement une fonction croissante du nombre d'éléments de contours. On a :

$$e^{approx}(p_1, \ldots, p_m) \leq e^{approx}(p_1, \ldots, p_m, p_{m+1})$$

En conséquence, afin de regrouper les éléments de contours, il est nécessaire de compenser cet accroissement de e^{approx}. En effet, la minimisation de la seule erreur d'approximation conduit à une partition en singletons. Il faut donc introduire une mesure $e^{over}(p_1, \ldots, p_m)$ qui quantifie notre a priori sur le type de sous-ensembles désiré. Comme nous voulons partitionner les contours de l'image avec le nombre minimum de courbes, nous avons proposé dans [23] de prendre comme mesure une fonction croissante de $\sum_{i=1}^{m} l_i$, la longueur totale de $\{p_1, \ldots, p_m\}$. Des mesures plus compliquées, fondées sur la densité des contours sur la courbe optimale dans $\mathcal{S}$, peuvent aussi être utilisées.

Une entropie G, qui quantifie la consistance d'un ensemble d'éléments de contours par rapport à la courbe optimale dans $\mathcal{S}$, peut donc être définie comme la différence pondérée de e^{over} et de e^{approx}:

$$G(p_1, \ldots, p_m) = e^{approx}(p_1, \ldots, p_m) - \lambda e^{over}(p_1, \ldots, p_m)$$

où λ contrôle l'équilibre entre e^{approx} et e^{over}. Plus cette entropie G est faible, plus l'ensemble est supposé cohérent avec une courbe de $\mathcal{S}$, et plus il est une structure géométrique importante à détecter. Conserver les sous-ensembles qui ont une entropie faible est une alternative, à notre avis, plus intéressante que de faire une sélection sur des critères non géométriques comme la norme du gradient le long du contour.

Nous avons donc formulé la segmentation en courbes comme la recherche de la partition $\mathcal{P}$, de l'ensemble des contours de l'image, qui minimise une énergie de type "Mumford et Shah" [27]:

$$E(\mathcal{P}) = \sum_{P \in \mathcal{P}} \left(e^{approx}(\{p_i\} \in P) - \lambda e^{over}(\{p_i\} \in P) \right) \tag{7.1}$$

Mais comment minimiser cette énergie de façon efficace? Ce problème est de complexité algorithmique importante pour deux raisons. Premièrement, il est équivalent en terme de complexité au problème du voyageur de commerce (problème NP-complet). Deuxièmement, la minimisation de E nécessite a priori de calculer e^{approx} pour tous les sous-ensembles possibles. Cela conduit à des temps de calculs prohibitifs.

L'approche que nous avons proposée dans [23] pour réduire la complexité du problème, consiste à procéder de façon approchée et itérative. On trouve en premier lieu le sous-ensemble qui minimise l'entropie G, puis on retire les éléments sélectionnés de l'ensemble des contours. Ensuite, la recherche du sous-ensemble qui minimise l'entropie G est renouvelée. Nous avons nommé cette approche "Best-First Segmentation", car elle fournit les sous-ensembles dans l'ordre croissant de leur entropie.

7.2 "Maximum Length Criterion"

En pratique, comment choisir les mesures e^{over} et e^{approx}? De plus, comment interpréter λ et donc comment bien le régler, sachant qu'il dépend fortement de la convexité de e^{over} et

e^{approx}? Face à ces questions, il faut approfondir notre analyse pour réduire les choix possibles d'entropie G.

On remarque que la variation d'entropie relative au regroupement de l'élément de contour p_{m+1} avec l'ensemble $\{p_1, \ldots, p_m\}$ est :

$$\Delta G = G(p_1, \ldots, p_m, p_{m+1}) - G(p_1, \ldots, p_m) - G(p_{m+1})$$

Un ΔG négatif indique que le regroupement est potentiellement intéressant. En conséquence, pour compenser l'erreur d'approximation dans ΔG, e^{over} doit satisfaire :

$$e^{over}(p_1, \ldots, p_{m+1}) \geq e^{over}(p_1, \ldots, p_m) + e^{over}(p_{m+1})$$

et ne pas être une fonction linéaire de m. Un exemple simple d'une telle mesure est le carré des longueurs des éléments de contours $\{p_1, \ldots, p_m\}$. Mais, le choix d'une fonction de convexité particulière pour e^{over}, restreint le choix sur la fonction e^{approx}. En effet, les convexités doivent s'équilibrer dans G.

Au final, ce que nous désirons imposer, c'est qu'un ensemble d'éléments de contours soit considéré comme bien approximé par une courbe, lorsque la variance $\frac{1}{\sum_{i=1}^{m} l_i} e^{fit}$ des distances des p_i à la courbe optimale est inférieure à σ^2. La valeur σ est, par définition, la précision du tracé des courbes dans l'image. De plus, parmi tous les sous-ensembles qui respectent cette contrainte, nous sommes intéressés par ceux qui sont les plus longs. Nous pouvons donc formuler notre problème, toujours selon le principe "Best-First Segmentation", comme la recherche du sous-ensemble d'éléments de contour le plus long possible, avec une erreur d'approximation e^{fit} inférieure à $\sigma^2 \sum_{i=1}^{m} l_i$. Nous avons nommé ceci le "Maximum Length Criterion" (MLC) [28].

Si la formulation MLC paraît a priori très différente de la précédente, il n'en est rien. Nous pouvons la reformuler sous la forme d'une énergie de type "Mumford et Shah". En effet, le MLC peut être vu comme la minimisation de :

$$\mathcal{E}'(\mathcal{P}) = \sum_{P \in \mathcal{P}} g(e^{fit}(\{p_i\} \in P)) - g(\sigma^2 \sum_{\{p_i\} \in P} l(p_i)) \tag{7.2}$$

où $l(p_i) = l_i$ est la longueur de l'élément de contour p_i et $g(x)$ est une fonction strictement croissante. L'avantage de cette reformulation est de montrer que les convexités de $e^{approx} = g(e^{fit})$ et de $e^{over} = g(\sigma^2 \sum_{i=1}^{m} l_i)$ ne peuvent pas être choisies indépendamment. De plus, dans le cas particulier où $g(x) = x^\gamma$, l'entropie G d'un sous-ensemble d'éléments de contours est :

$$G = (e^{fit})^\gamma - \sigma^{2\gamma} (\sum_{i=1}^{m} l_i)^\gamma \tag{7.3}$$

Le paramètre λ s'interprète donc comme le terme $\sigma^{2\gamma}$. Ceci permet de choisir λ beaucoup plus facilement.

7.3 Détection globale

La recherche du sous-ensemble des éléments de contours qui minimise le critère MLC ou l'entropie G dans (7.3) reste de complexité équivalente à celle du problème de segmentation.

Mais, grâce à la version récursive de l'approximation par une courbe, e^{fit} peut être calculé simplement. Pour réduire la complexité de la recherche, nous supposons en plus que les éléments de contours sont reliés par un graphe connexe, acyclique, et orienté. Ce graphe indique, pour chaque élément de contour, avec quels autres éléments il peut être regroupé. Avec cette hypothèse, il est possible d'obtenir une solution approchée du minimum de G ou de MLC. Ensuite, en appliquant le principe "Best-First Segmentation", les contours sont partitionnés de façon efficace, en approchant la solution optimale de (7.2).

Pour constituer le graphe, nous allons considérer que chacun de ses noeuds est un élément de contours. Les arcs sont des liens entre les noeuds qui peuvent être potentiellement regroupés. Dans la partie 8.1, nous détaillons quels sont les critères locaux qui peuvent être utilisés pour créer ces arcs. Pour que le graphe soit orienté, il nous faut ordonner les éléments de contours selon leurs coordonnées verticale, puis horizontale. Les arcs sont orientés de façon à respecter l'ordre des noeuds des extrémités. Ceci garantit aussi que le graphe n'est pas cyclique. Soit p et p' deux éléments de contours, nous notons $p' \rightarrow p$ le fait qu'il existe un arc orienté partant de p' et allant vers p. A chaque arc, nous associons un coût qui correspond au regroupement avec p. Avec la formulation MLC, ce coût est l'opposé de la longueur de p si l'erreur moyenne $\frac{1}{\sum_{i=1}^{m} l_i} e^{fit}$ reste inférieure à σ^2 après regroupement. Sinon, le coût est infini et le regroupement impossible. Lorsque la formulation avec G est choisie, le coût de l'arc $p' \rightarrow p$ est la variation d'entropie ΔG liée au regroupement de p.

L'algorithme de Moore-Dijkstra [29], aussi connu sous le nom de l'algorithme de la programmation dynamique, effectue la recherche de la solution optimale du chemin le plus long dans un graphe orienté, lorsque des coûts associés à chaque arc sont fixes. Dans notre cas, il n'est pas possible d'utiliser directement ce type d'algorithme, car l'erreur e^{fit} n'est pas calculable sans connaître tous les éléments de contours précédemment sélectionnés. La difficulté du critère MLC, comme de la formulation avec l'entropie, est que le critère est global. C'est son aspect non local qui rend la complexité de ce problème équivalente à celle du voyageur de commerce.

Dans [23], nous avons proposé une modification du l'algorithme de Moore-Dijkstra qui permet de trouver une solution approchée à notre problème. L'idée consiste à conserver, à chaque noeud, non plus la meilleure hypothèse comme dans l'algorithme de Moore-Dijkstra, mais une liste ordonnée des b meilleures hypothèses. On note $\{p.s_i\}$, $1 \leq i \leq b$, les b hypothèses associées à p. Une hypothèse est un sous-ensemble des contours, caractérisé par : les coefficients $s_i.A$ de la courbe qui l'approche le mieux, la matrice $s_i.K$ de covariance associée, sa longueur totale $s_i.L$, et son entropie $s_i.G$. L'avantage de cette caractérisation est double : elle est complète, ce qui évite de conserver la description du sous-ensemble, et elle est récursive comme expliqué dans la partie 6.2, ce qui fait gagner beaucoup de temps de calculs.

Notons $p.l$ la longueur de l'élément de contour p. L'algorithme est en trois étapes pour chacun des noeuds p, pris dans l'ordre :

1. Pour chacun des b meilleures sous-ensembles s'_i de chaque noeud p' tel que $p' \rightarrow p$, calculer la nouvelle entropie G quand p est regroupé à s'_i. Cela nécessite de calculer la nouvelle longueur totale de $\{s'_i, p\}$ en ajoutant $p.l$ à $s'_i.L$ et la nouvelle erreur e^{fit} en utilisant (6.7).

2. Initialiser un sous-ensemble réduit à p et calculer e^{fit}, puis son entropie G.

3. Soit s_i les b sous-ensembles construit lors des deux étapes précédentes qui ont une entropie minimale. Calculer la matrice $s_i.K$, et les coefficients $s_i.A$, pour chacun des sous-ensembles sélectionnés en utilisant (6.5) et (6.6), respectivement. La liste s_i est associée au noeud p.

A la fin de cette boucle, le noeud, qui a dans sa liste d'hypothèses la plus grande valeur de l'entropie, est le dernier noeud du sous-ensemble recherché. On retrouve tous les noeuds en parcourant le graphe en sens inverse.

Cet algorithme nécessite deux boucles sur les noeuds. Sans l'estimation récursive des coefficients, le calcul des coefficients aurait dû être réalisé à l'intérieur des deux boucles. De plus, l'estimation récursive de e_{fit} évite de calculer des coefficients dans de nombreux cas et diminue d'autant la complexité de la détection. Au coeur des deux boucles, il ne reste que le calcul de e^{fit}, qui est simple, avec un nombre d'opérations constant.

Cet algorithme réalise un compromis entre optimalité et temps de calcul. En effet, lorsque b, le nombre maximum d'hypothèses conservées à chaque noeud, augmente jusqu'au nombre de noeud du sous-ensemble optimal, la détection devient optimale. Mais cela se fait au détriment de la vitesse de traitement.

7.4 Régularisation a priori

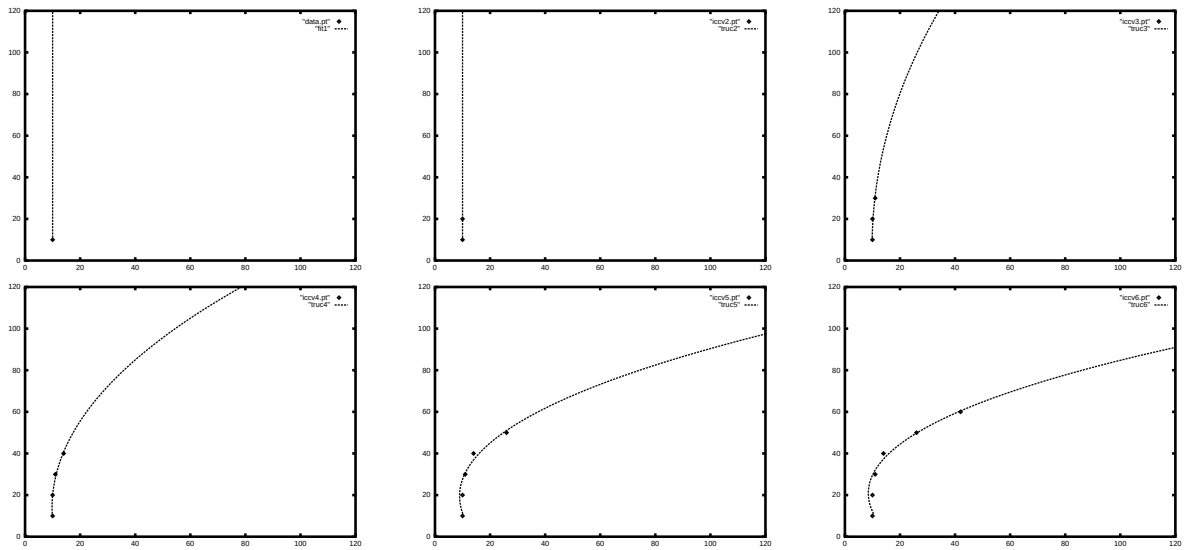


FIG. 7.1 – *Approximation incrémentale d'un ensemble de 6 points. La courbe utilisée est polynômiale de degré 4. Elle a 5 paramètres. L'approximation est biaisée vers des courbes de degré faible lorsque le nombre de points n'est pas suffisant pour déterminer tous les paramètres.*

Dans la partie 6.2, nous avons indiqué comment initialiser la matrice de covariance K_0 et les coefficients de la courbe A_0. De façon plus générale, cette initialisation permet de mettre un a priori sur la solution recherchée. Ceci est particulièrement intéressant lorsqu'il y a un problème de multicolinéarité, c'est-à-dire que les données ne suffisent pas pour obtenir une

solution unique. En particulier, il y a multicolinéarité lorsque le nombre de points est inférieur au nombre de paramètres de la courbe qui va approximer ces points.

Il est donc possible d'initialiser K_0 et A_0 de différentes façons selon l'objectif. Par exemple, on observe que les marquages routiers sont le plus souvent rectilignes et de temps en temps courbes. Cela indique qu'il faut privilégier, lors de la segmentation, les courbes de degré faible à celles de degré élevé. Cela conduit à initialiser K_0 avec une matrice diagonale de diagonale décroissante.

Une approche plus systématique consiste à faire l'apprentissage de K_0 et A_0 à partir d'images réelles. Pour ce faire, il suffit de segmenter en courbes des images de routes représentatives de leur diversité, et d'utiliser la moyenne et la matrice de covariance des coefficients des courbes obtenues comme valeurs de A_0 et K_0.

Comme montré dans la Fig. 7.1, la régularisation est avantageuse lorsque les données ne contiennent pas assez d'informations pour que tous les coefficients de la courbe de degré d soient estimés sans ambiguïté. Le résultat est une courbe biaisée vers un degré inférieur à d.

7.5 Résultats

La Fig. 7.2 montre quelques images de segmentation en courbes sur différents types de routes. Sur les deux premières images, on peut comprendre l'intérêt que peut avoir cette segmentation pour identifier et caractériser des situations complexes de marquages.

Les deux images suivantes illustrent le fait que la segmentation n'est pas trop perturbée par la présence d'obstacles sur la route. Néanmoins, plus on permet aux courbes d'avoir de larges discontinuités, et plus on a de chance d'obtenir des courbes correspondant à des alignements fortuits. Par là-même, il est souvent difficile de segmenter comme une même courbe les lignes pointillées lorsque les marquages font moins de 50% de sa longueur. Il est toujours possible d'avoir des alignements fortuits dans une image, mais ils se distinguent des autres par l'inconsistance de leur position d'image en image. Ce problème sera traité dans la suite grâce au suivi temporel des courbes.

L'algorithme de segmentation peut aussi être utilisé pour segmenter la partie chaussée du reste, comme le montre les deux dernières images de la Fig. 7.2. Contrairement aux images avec marquages, une image en niveaux de gris ne suffit plus. Comme cela a été proposé par divers auteurs [30, 31], il faut utiliser la combinaison des couleurs qui discrimine le plus la chaussée. Dans nos expériences, nous avons utilisé simplement le canal bleu de l'image Rouge-Vert-Bleu.

La figure 7.3 présente quelques résultats de segmentation en présence d'ombres et tâches lumineuses. Le degré des courbes utilisées est $d = 3$. Seules les courbes à l'entropie suffisamment basse sont affichées. Dans nos tests, nous n'avons, pour chaque noeud, conservé que les 6 meilleures hypothèses ($b = 6$). Nous avons essayé avec des valeurs plus importantes, sans noter de changement drastique des résultats obtenus.

Lorsque la voiture tangue, l'angle d'inclinaison ϕ de la caméra par rapport à la route change, et la position v_h de l'horizon varie. Lors de la segmentation, la valeur de v_h est fixée et ne peut pas être directement estimée par l'algorithme avec la représentation explicite. Nous avons observé qu'une forte erreur sur la position de l'horizon gêne la segmentation lorsque la courbe se rapproche de l'horizon. Cela a souvent pour effet de limiter la profondeur des courbes segmentées. En utilisant la représentation implicite plutôt qu'explicite, sans rien changer à l'algorithme

FIG. 7.2 – *Segmentation en courbes avec sélection des 10 plus longues courbes (MLC) pour les deux premières lignes, et sélection des deux plus longues (MLC) pour la dernière ligne. Les courbes ont 4 paramètres ($d = 3$) pour les deux premières lignes et 5 paramètres pour la dernière ($d = 4$).*

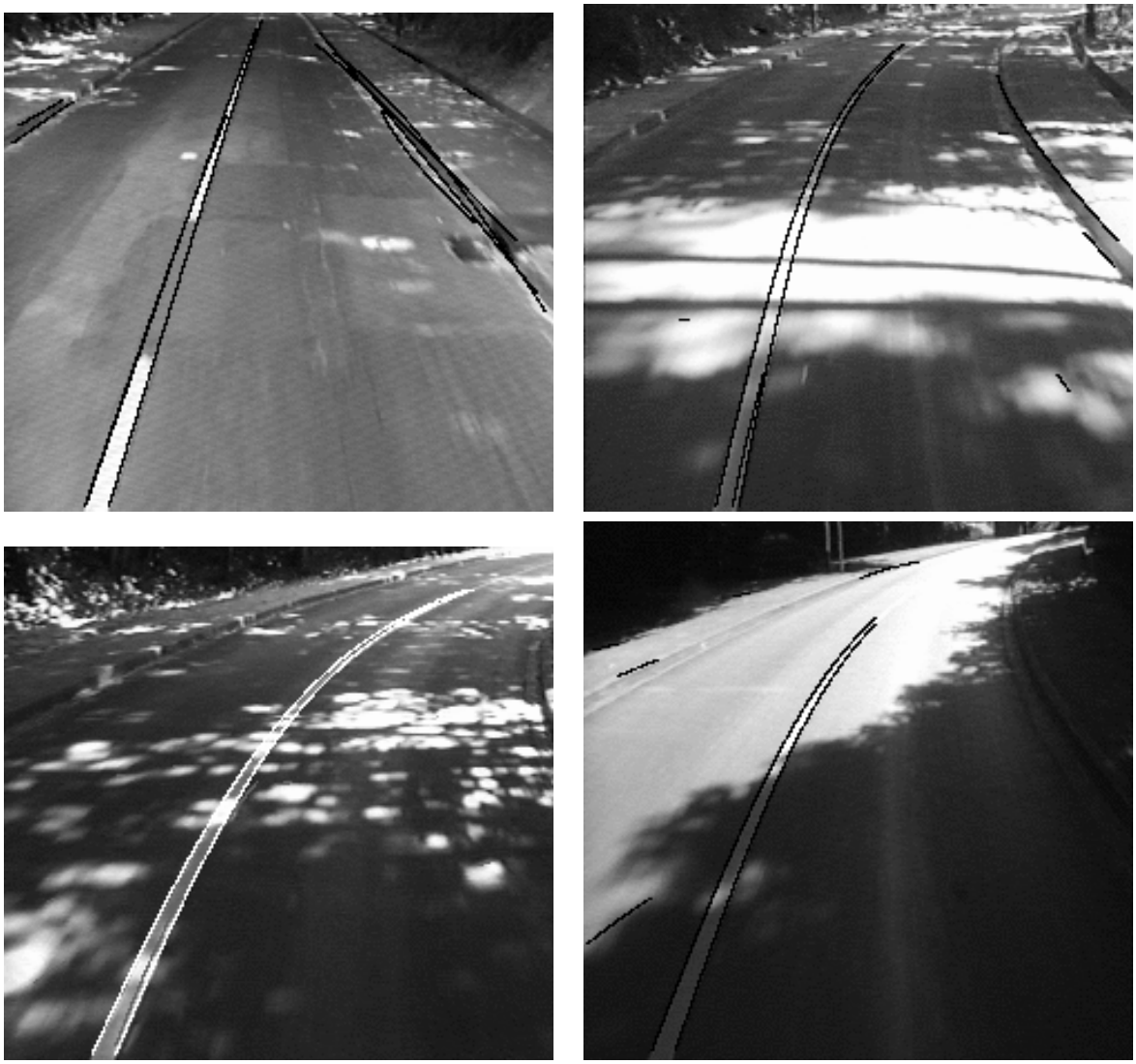


FIG. 7.3 – *Segmentation en courbes lorsque les conditions d'éclairage, des ombres, et des tâches de lumière produisent des perturbations importantes.*

de segmentation, ce problème disparaît. La contrepartie est une augmentation du temps de calcul pour un même degré. En effet, une courbe algébrique de degré d a $\frac{1}{2}d(d+1)$ coefficients alors que la courbe explicite de même degré a $d+1$ coefficients.

Chapitre 8

Détection de courbes par groupement local

La segmentation en courbe réalisée précédemment reste malgré tout trop exigeante en temps de calcul pour la puissance actuelle des ordinateurs, pour avoir un traitement à 25 images par seconde. Nous avons vu que la difficulté du problème vient du caractère global de l'énergie à minimiser. Nous avons donc été amenés à formuler une version dégradée de cette énergie qui conduit à une segmentation en courbes uniquement sur des critères locaux. Cela permet d'utiliser directement l'algorithme de Moore-Dijkstra pour minimiser cette énergie.

8.1 Critères locaux

L'erreur e^{fit} dépend des paramètres de la courbe qui approxime au mieux les données de contours. Remplaçons la par une erreur e^{loc} qui dépend seulement des données et qui quantifie si ces dernières ressemblent à une courbe lisse et continue. Cette erreur e^{loc} est construite comme la somme des erreurs entre points consécutifs sur la courbe, selon un ou plusieurs critères de continuité et de régularité. Il est possible de faire de nombreux choix de critères qui testent localement si une courbe est lisse ou pas. Voici les plus simples que nous avons utilisés :

- décalage selon la coordonnée u entre deux points consécutifs.
- angle entre les tangentes pour deux points consécutifs,

Des critères plus sophistiqués sur des triplés de points consécutifs et la courbure peuvent aussi être mis en oeuvre.

8.2 Algorithme

En substituant e^{fit} par e^{loc} dans (7.1), nous obtenons une entropie d'un type beaucoup plus simple à minimiser car elle a perdu son aspect global. Comme dans la partie 7.3, le graphe orienté des éléments de contours est construit, mais maintenant les coûts sur les arcs sont connus et fixes. Le coût sur un arc est une fonction croissante des critères locaux précédents appliquée au regroupement des deux éléments de contours liés par l'arc. Alors, directement, l'algorithme de Moore-Dijkstra permet d'obtenir le sous-ensemble des contours le plus long et le plus lisse.

Ensuite, on utilise le principe de ”Best-First Segmentation” pour partitionner l’ensemble des contours en courbes. Chaque sous-ensemble peut être approximé par un polynôme hyperbolique de degré deux ou trois. Cela permet par exemple de remonter à l’information de courbure, comme décrit dans la partie 6.1. Cela est aussi utile pour faire le suivi temporel des courbes.

8.3 Suivi temporel

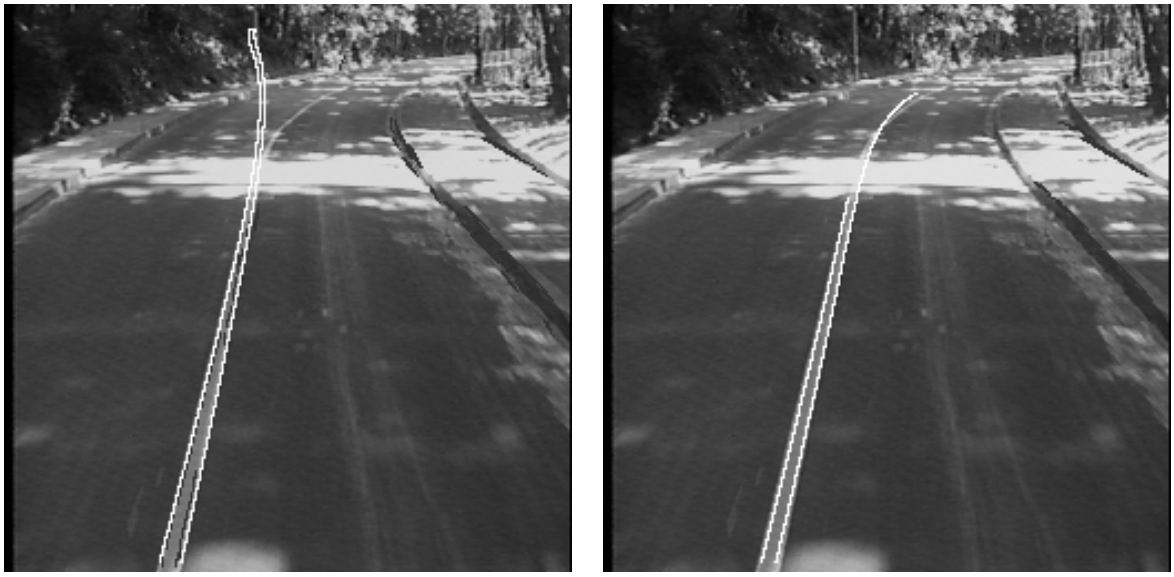


FIG. 8.1 – *A gauche, regroupement local des contours. On peut observer que le marquage est prolongé par le poteau. A droite, regroupement avec une contrainte globale.*

Avec une énergie locale, nous ne contraignons plus la courbe détectée à appartenir à une classe restreinte comme les polynômes d’un certain degré. C’est une courbe d’une classe beaucoup plus large : celles qui sont lisses et continues. Ceci rend la détection moins discriminante et l’apparition d’alignements fortuits plus probable. Un effet gênant est illustré sur la Fig. 8.1, où la courbe se prolonge trop loin à cause d’un panneau. Le regroupement avec une contrainte globale, plus discriminant, a moins ce problème. Par contre, la version locale fonctionne beaucoup plus rapidement que la globale.

Le problème des courbes fantômes dues à des alignements fortuits peut être en partie traité en utilisant la consistance temporelle. La façon la plus simple de faire le suivi d’une image à l’autre est d’approximer chaque sous-ensemble d’éléments de contours par un polynôme explicite ou implicite. Ensuite, les courbes de l’image courante sont appariées à celles de l’image précédente en cherchant le plus voisin dans l’espace des vecteurs coefficients selon la métrique donnée par la matrice de covariance de la courbe. Ce mécanisme de reconnaissance Bayesien permet d’éliminer les courbes fantômes qui apparaissent sur une image uniquement, mais introduit un retard d’une image lors de l’apparition d’une nouvelle courbe.

Malheureusement, ce mécanisme ne permet pas de traiter le cas où un alignement fortuit perturbe la courbe sur une partie seulement. Il faut alors avoir recours à une autre approche. Une solution est d’utiliser un système de vision stéréoscopique avec deux caméras. En effet,

avec un système stéréoscopique, il est possible de séparer ce qui est au niveau du sol de ce qui est au dessus. Cela permet d'éliminer dans chaque image tous les obstacles, et diminue donc la probabilité d'avoir des alignements fortuits, au sol seul (voir le travail de Raphaël Labayrade dans [32]).

Chapitre 9

Détection de la voie de circulation

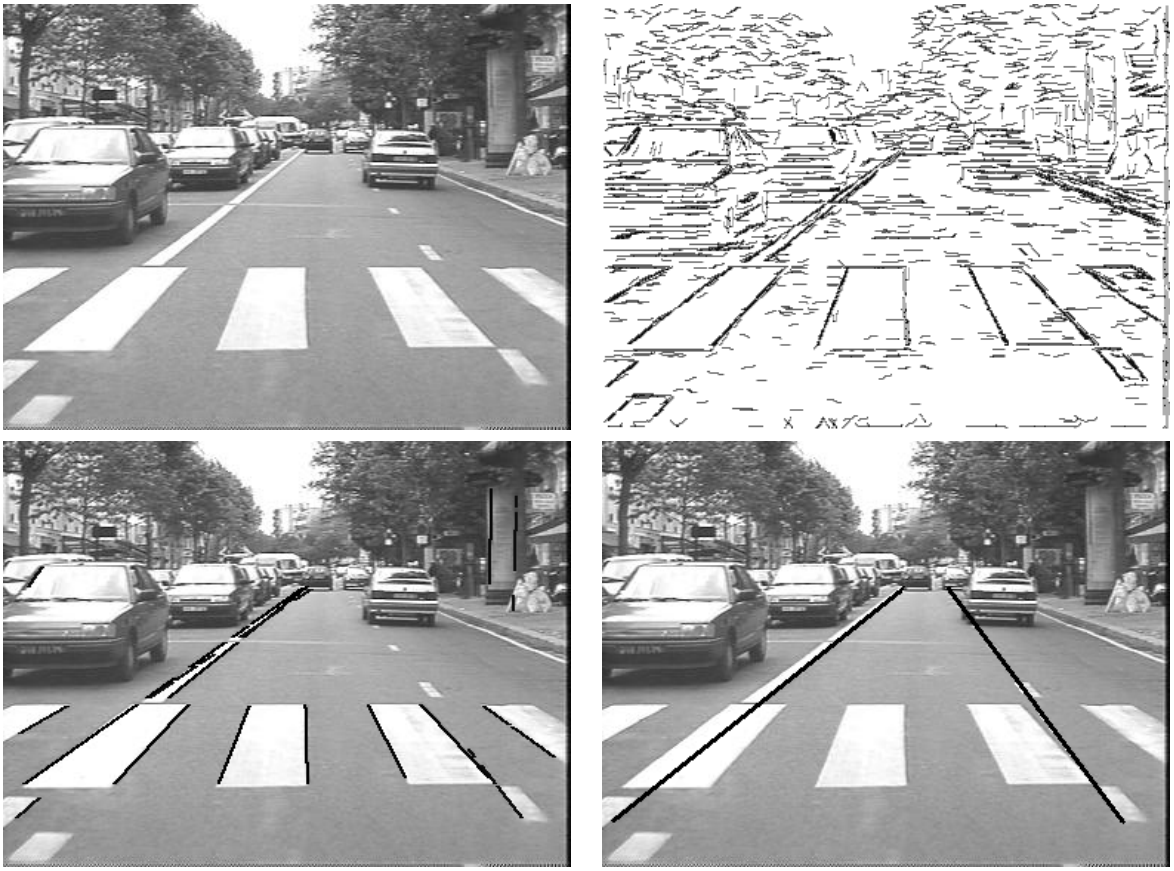


FIG. 9.1 – *Image originale, résultat de l'extraction des segments, segmentation en les 15 meilleurs courbes, reconnaissance et suivi de la voie.*

L'algorithme précédent segmente, à partir des segments extraits de l'image originale, des courbes d'un type donné a priori, dans l'ordre des longueurs décroissantes si MLC est utilisé, et dans l'ordre des entropies croissantes si G est utilisé. La Fig. 9.1 présente : une image originale, le résultat de l'extraction des segments, le résultat de la segmentation en les 15 meilleures courbes. Comme on le remarque sur cette segmentation, le marquage pointillé droit n'apparaît

pas parmi les 15 meilleures courbes, à cause du passage piétons. De plus, parmi les courbes segmentées, il est difficile, sans un traitement additif, de savoir celles qui correspondent au bord de la voie du véhicule. Le résultat que l'on attend est présenté dans la dernière image. Dans cette dernière, les marquages des bords de voie sont bien reconnus. Ce résultat n'a pu être obtenu qu'en intégrant le suivi dans la reconnaissance afin de lever les ambiguïtés lors du traitement d'images où les marquages sont nombreux.

Après avoir décrit notre modèle de bord de voie, nous expliquons comment faire le suivi de manière unifiée avec la reconnaissance.

9.1 Combinaison de courbes

Pour détecter la voie, il nous faut modéliser plus finement sa structure géométrique par rapport aux courbes représentant les marquages. Supposons que les deux bords sont des courbes polynômiales parallèles au niveau du sol. L'équation de l'image du bord gauche s_{gauche} est :

$$u = \sum_{i=0}^{i=d} a_i v^{1-i} \tag{9.1}$$

Le bord droit a la même forme, mais est juste décalé d'une distance w, la largeur inconnue de la route, l'équation de son image s_{droit} est :

$$u = \sum_{i=0}^{i=d} a_i v^{1-i} + wv \tag{9.2}$$

L'utilisation du parallélisme présente l'avantage de permettre la construction d'un modèle de voie au nombre de paramètres très réduit. Un seul paramètre, la largeur w de la voie, est à estimer en plus par rapport au cas de la courbe. Selon notre modèle, une voie est une forme, combinaison de deux courbes, caractérisée par le vecteur $(a_0, \cdots, a_d, w)$ de taille $d + 2$.

Le deuxième avantage de ce modèle de voie est que la paramétrisation reste linéaire, ce qui permet de rester dans le cadre précédemment exposé, et nous conduit à étendre le domaine d'utilisation de l'algorithme décrit dans la partie 7.3. En effet, la seule contrainte qui porte sur le choix de la classe $\mathcal{S}$ est qu'elle soit linéairement paramétrée. Cela inclut les classes de courbes précédemment décrites, mais aussi des formes plus compliquées combinaisons de courbes linéairement paramétrées. Seule contrainte, les liens entre les différentes courbes qui composent la forme doivent être eux-aussi linéaires. Comme nous venons de le voir, le parallélisme entre courbes est une contrainte linéaire. Mais de nombreuses autres contraintes peuvent être aussi utilisées, comme le passage par un même point particulier, des symétries, une tangente commune, des dérivées secondes égales sur une abscisse donnée... Ce type de formes peut être utilisé pour modéliser les marquages plus compliqués que les bords de voie, comme les disjonctions.

Comme dans le cas de la courbe, il est possible de définir une distance entre un élément de contours p et une forme comme le minimum des distances à chaque courbe qui la compose. Dans le cas de la voie, c'est donc $min(d(p, s_{droit}), d(p, s_{gauche}))$. Cette distance permet d'étendre facilement l'approximation récursive d'une courbe décrite dans 6.2 à une combinaison de courbes.

En plus de la modélisation géométrique, nous avons ajouté une contrainte sur le contraste des bords gauche et droit : le bord gauche doit être plus clair à gauche et le bord droit plus clair

à droite. Ceci permet de sélectionner le tracé intérieur de la voie et évite donc l'ambiguïté entre les bords extérieurs et intérieurs lors de la détection. Cette contrainte s'intègre facilement dans la distance entre un élément de contour p et la forme.

Nous avons donc toutes les conditions nécessaires pour que l'algorithme décrit dans la partie 7.3 s'applique. Il est toujours possible d'appliquer le principe "Best-First Segmentation", mais généralement c'est la première forme détectée qui est la plus intéressante. En effet, les contraintes sur la forme sont suffisamment grandes pour qu'il y ait très peu de chance de trouver une autre forme correspondant à la voie, de longueur significative. En complexifiant le type des formes recherchées, nous avons donc transformé l'algorithme de segmentation en un algorithme de détection, sans en changer le principe.

9.2 Régularisation et suivi

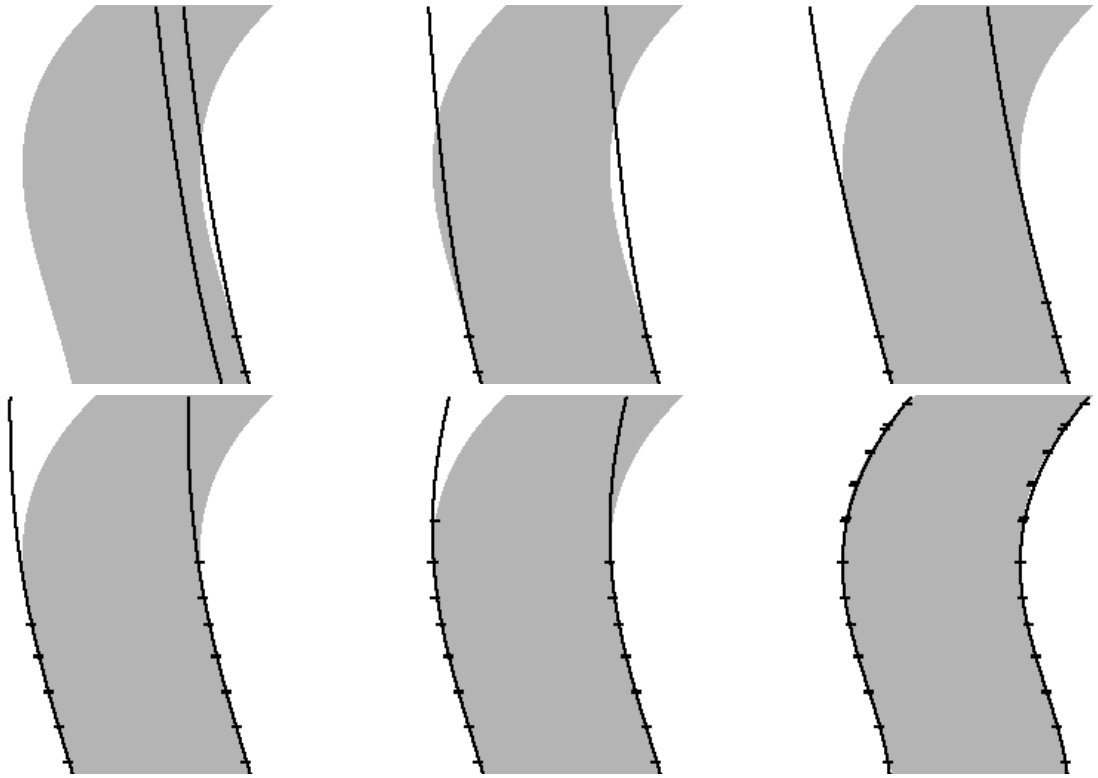


FIG. 9.2 – *Approximation récursive avec un nombre croissant de segments (1, 2, 3, 9, 11, et 20 respectivement) par une paire de courbes parallèles de degré 5. Ce type de forme a 7 paramètres. Lorsque le nombre de points est trop petit pour déterminer complètement les paramètres, le résultat est biaisé a priori vers des droites parallèles.*

Nous n'avons pas détaillé la façon d'initialiser l'approximation de la forme. En effet, l'initialisation qui sert aussi de régularisation permet d'intégrer de façon naturelle le suivi à l'algorithme de reconnaissance précédent. Il est important de préciser que, généralement, le suivi et la détection dans chaque image sont réalisés indépendamment. Il faut alors utiliser un algorithme

d'appariement du type décrit dans la partie 8.3. Ceci oblige donc de faire en séquence trois algorithmes (reconnaissance, suivi, et appariement), ce qui rend difficile l'utilisation de la totalité de l'information disponible. Faire le suivi lors de la phase d'initialisation ne souffre pas de cette difficulté.

Comme cela a été décrit dans la partie 7.4, la régularisation peut être réalisée en biaisant le résultat vers une solution a priori. Dans la Fig. 9.2, la régularisation biaise la solution vers deux droites parallèles, et cela d'autant plus que les données ne sont pas suffisamment nombreuses.

Des propriétés d'invariance attendues sur le résultat imposent aussi des contraintes sur le choix de la matrice de covariance K_0. Par exemple, si l'on veut une invariance par rapport aux translations horizontales de l'image, la première ligne et la première colonne de K_0 doivent être nulles.

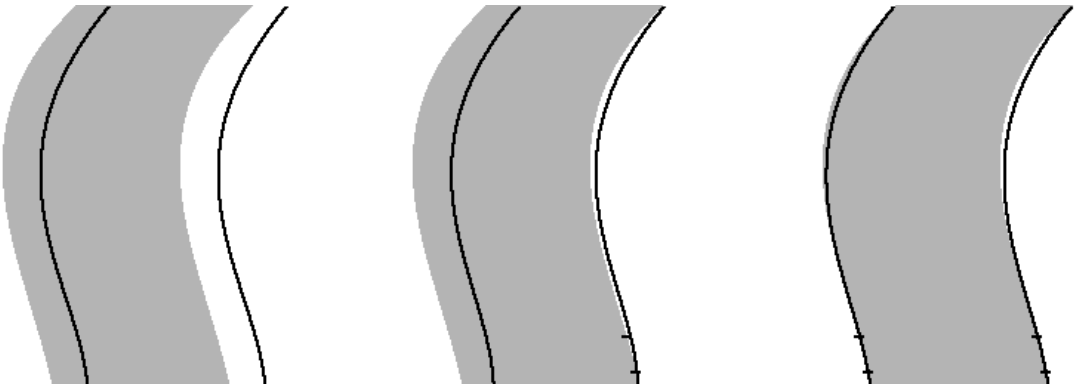


FIG. 9.3 – *Approximation récursive avec un nombre croissant de points par une paire de courbes parallèles de degré 5. L'information a priori utilisée est une version translatée de la courbe recherchée (à comparer avec la régularisation à priori utilisée en Fig. 9.2).*

La première image doit être régularisée a priori. Par contre, pour les autres images, il est beaucoup plus intéressant d'utiliser le résultat de l'image précédente pour biaiser la solution lorsqu'il n'y a pas assez de données dans l'image courante. Dans la Fig. 9.3 l'initialisation simule un suivi. On remarque que la convergence vers une forme proche de la solution est obtenue avec beaucoup moins d'itérations que dans le cas d'une initialisation a priori comme celle effectuée dans la Fig. 9.2.

La façon la plus simple de faire le suivi est d'initialiser A_0^t avec le vecteur obtenu dans l'image précédente A^{t-1}, et K_0^t avec μK^{t-1}. Un aspect intéressant est que le facteur μ contrôle la taille de la recherche autour de la solution A_0^t. Plus la valeur de μ est petite, plus l'espace de recherche est petit, et plus rapide sera la détection. D'un autre coté, plus μ est grand et plus on a de chance de bien détecter la courbe lors d'une variation brutale de sa forme ou de sa position. Le paramètre μ règle le poids relatif du suivi et de le reconnaissance.

Lorsque le modèle cinématique ou dynamique du véhicule est disponible, de meilleurs prédictions, que celles supposant des courbes immobiles, peuvent être obtenues pour faire l'initialisation.

9.3 Résultats

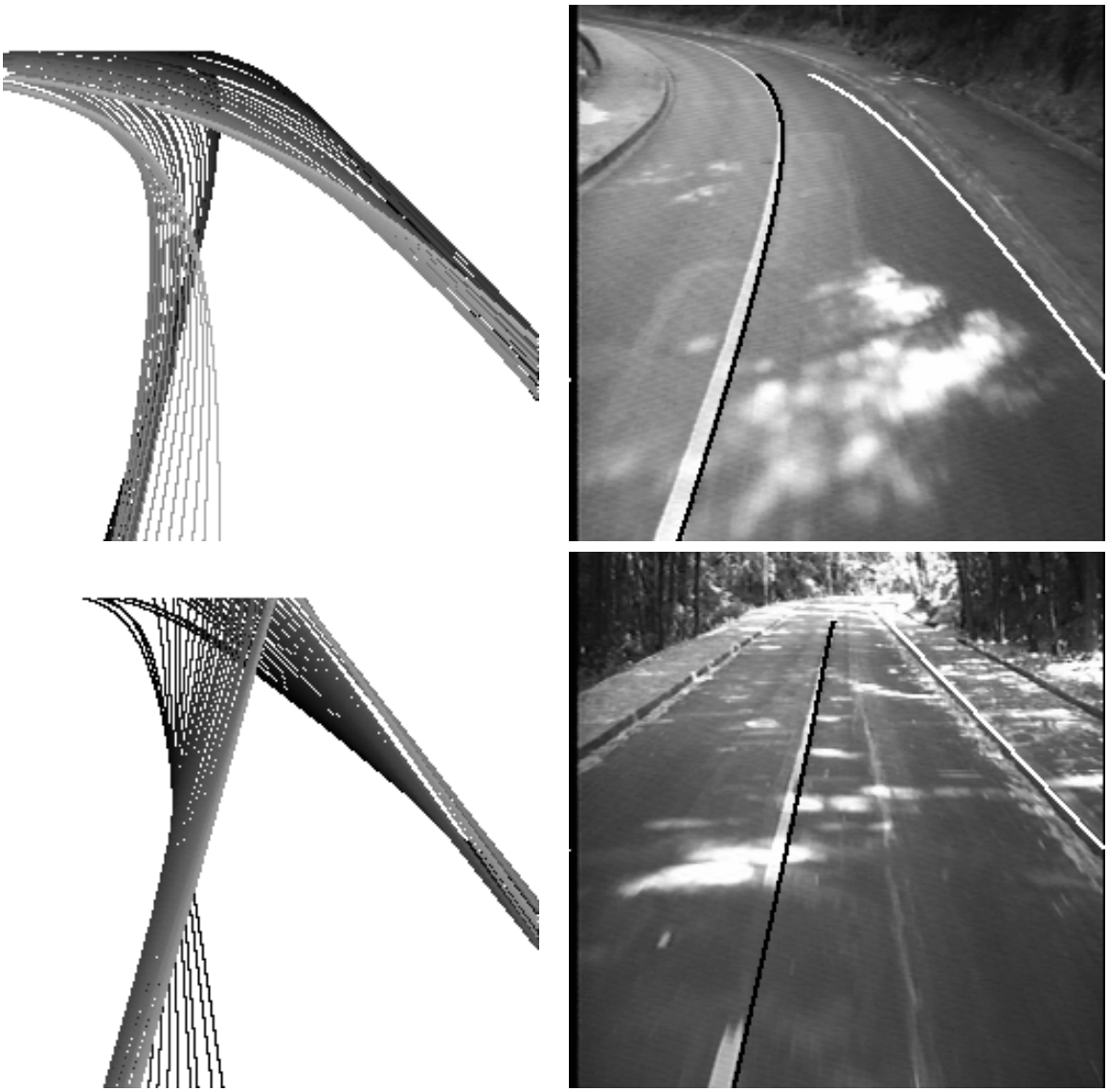


FIG. 9.4 – *Séquences de 40 images où la paire de bords de route la plus longue est détectée. Pour chaque séquence, une image présentant des tâches lumineuses est présentée.*

La Fig. 9.4 montre le résultat de la détection de la voie sur deux séquences de 40 images d'une route présentant des tâches lumineuses et quelques interruptions des marquages. Sur d'autres séquences, nous avons constaté que lors des virages et lorsque la route présente de forts dénivelés, la voie n'est pas détectée aussi loin qu'on peut l'espérer. En effet, l'hypothèse de parallélisme des bords droit et gauche, telle que nous l'avons posée, est moins bien vérifiée dans les virages très incurvés. Quant aux fortes dénivellations, elles peuvent mettre en défaut l'hypothèse du monde plan.

Le fait d'utiliser le suivi permet d'accélérer l'algorithme et d'atteindre des cadences de 3 à 4 images par secondes sur un PC à 700 Mhz.

Conclusion

Ce document présente les techniques développées et les résultats obtenus au sein du Livic de 1998 à 2001 au sujet de la perception des bords de voie, avec une seule caméra. Un algorithme de détection des segments dans les images [5] qui permet de résumer le contenu pertinent des images a été conçu et utilisé comme prétraitement de façon systématique. Un algorithme fondé sur l'hypothèse d'une route rectiligne a été développé [33] et testé relativement intensivement dans le cadre du contrôle latéral. Un algorithme qui réalise la reconnaissance et le suivi des bords de voie courbe a été présenté [28]. Enfin, deux algorithmes de segmentation en courbes, l'un local et l'autre global, ont été détaillés [23]. Ces derniers peuvent être utilisés pour traiter des passages piétons et des marquages autres que ceux des bords de voie.

Pour chacun, nous avons essayé de décrire et d'illustrer leurs points forts et faibles. Mais au delà de leurs défauts, qu'est-ce qui limite ce type de système de perception, indépendamment du traitement?

Les essais sur pistes que nous avons réalisés dans le cadre du contrôle latéral m'ont permis de mieux cerner ces limites. La perception de la route est fort différente suivant qu'elle est faite à travers le pare-brise ou la caméra.

J'ai été souvent surpris du fait que certaines séquences d'images soient impossibles à traiter car elles sont trop dégradées et donc trop pauvres en information, alors qu'en tant que conducteur, on ne ressent pas la moindre gêne lors de l'acquisition. La supériorité de l'oeil humain par rapport à une caméra n'est évidemment pas à démontrer.

Les deux problèmes essentiels que l'on rencontre avec une caméra d'aujourd'hui sont :

- **champ perçu limité** : l'apparition d'objets et de marquages dans l'image est soudaine du fait d'un champ perçu limité. Ceci empêche d'anticiper les perturbations, et rend la détection particulièrement délicate. Un progrès consisterait à utiliser un capteur panoramique [34, 35], avec un miroir adapté à la tâche de détection.

- **Dynamique insuffisante** : face à la grande variabilité des conditions d'éclairage, la dynamique et l'adaptation du gain des caméras sont loin d'être suffisantes. Les caméras CMOS permettent d'espérer une amélioration. Mais un progrès significatif ne sera réalisé que s'il est possible d'adapter, localement dans l'image, le gain et ceci très rapidement.

Mais en attendant les caméras du futur, est-ce qu'un système de perception des bords de route peut permettre de mesurer la position latérale de façon suffisamment robuste pour être utilisé dans un système sûr? Mon opinion actuelle est qu'une réponse positive passe par un contrôle fort sur les conditions d'observation et le marquage. Cela me conduit à préconiser les deux perspectives suivantes comme axes d'amélioration.

Environnement contrôlé

La sécurité d'un système de contrôle impose des contraintes très fortes sur ce système. En particulier, il doit toujours pouvoir caractériser son état, même lorsqu'il est en mode dégradé. Cela implique que le système soit capable de bien estimer la précision des mesure qu'il fournit et plus généralement qu'il doit être capable d'identifier avec certitude dans quel mode de fonctionnement il doit se trouver. Par exemple, il faut toujours pouvoir déterminer si le véhicule est dans ou hors de sa voie, auquel cas il doit ralentir. Avec les caméras et les scènes routières actuelles, il est très difficile de distinguer entre un véhicule hors voie et une absence de marquage due à une dégradation de ce dernier ou à des conditions d'éclairage rendant l'image inutilisable. Pour un système automatique, ceci est un exemple de situation accidentogène qui ne doit pas être permis.

Pour commencer, il faut que la qualité du marquage soit garantie dans le temps. De plus, il faut que le marquage de la voie soit continu pour qu'il soit détectable par le capteur embarqué à tout instant et ce quelque soit la vitesse du véhicule. Pour renforcer la certitude sur sa détection, il est avantageux de donner à ces marquages une structure très spécifique, comme une espèce de code barre.

Ensuite, il faut contrôler les conditions d'éclairage de ces marquages pour garantir que le capteur embarqué arrivera à toujours les observer, et ceci indépendamment des variations d'illumination. Les marquages n'étant plus forcément conçu pour être visibles par le conducteur, un capteur autre qu'une caméra peut être utilisé avec avantages. L'idée est de se ramener au même type de conditions d'observation des marquages que celles obtenues la nuit avec les phares. En effet, comme décrit dans la partie 5.3, les seules structures rectilignes perçues sont alors les marquages.

Suivi temporel

Avoir un environnement contrôlé pose le problème de la détection des marquages d'une façon plus simple en terme de traitement des images, par rapport à ce qui a été décrit en introduction. En effet, au lieu d'avoir un problème de détection fondé sur un modèle de ce que l'on doit détecter, on a seulement un problème de type suivi robuste d'une courbe d'une image à l'autre. Le suivi doit être robuste pour tenir compte des inévitables petites perturbations, mais maintenant la courbe à détecter n'est plus perdue parmi de nombreuses autres courbes. La diminution de complexité du problème permet de détecter des courbes en un temps de calcul constant. Ce type d'approche est actuellement en cours d'études au Livic dans le cadre de la thèse de Sio-Song Ieng [36].

Remerciements

Je tiens à remercier avant tout Frédéric Guichard avec qui j'ai eu le plaisir de travailler durant deux années. Ce rapport expose en grande partie notre travail commun. Je remercie aussi mes collègues : Didier Aubert, Vahé Bagdhassarians, Georges Beurier, Ryad Benosman, Jean-Marc Blosseville, Pierre Charbonnier, Sio-Song Ieng, Julien Estève, Jonathan Fabrizio, Sébastien Glaser, Alexandru Grosu, Damien Koenig, Raphaël Labayrade, Jean Lavenant, Said Mammar, et Jacques Sainte Marie, qui trouveront dans ce rapport de nombreux échos de nos discussions. Enfin, merci à Georges Coche et Bernard Georges pour leur aide technique lors des expérimentations.

Annexe A

Bien construire un histogramme

Dans le chapitre 4, nous avons construit deux types d'histogrammes pour traiter chaque image. Ces histogrammes sont construits par accumulation selon une caractéristique des segments présents dans l'image. Ainsi l'histogramme qui donne l'orientation du véhicule est une accumulation selon la position de l'intersection avec la ligne d'horizon. Le profil latéral est constitué par l'accumulation selon la pente des segments.

A.1 Perturbations

Trop souvent, l'accumulation est simplement réalisée par un incrément unitaire de l'ordonnée de l'histogramme. Il y a souvent mieux à faire. Dans notre contexte, il peut arriver qu'un même segment soit coupé en deux segments lors de la détection. Avec un vote unitaire, le segment coupé va avoir plus d'importance que s'il ne l'était pas. Ceci n'est pas satisfaisant. Le vote doit donc être proportionnel à la longueur pour éviter de privilégier les segments courts par rapport aux longs, ou inversement.

La position des segments utilisés lors de ces accumulations est bruitée. La digitalisation de l'image est la source principale de perturbations. Elle peut être assimilée à un bruit uniforme. Il est nécessaire de prendre en compte ces perturbations lors de la construction de l'histogrammme si l'on veut éviter l'apparition de pics fantômes. Par exemple, la pente d'un segment court est moins précisément déterminée que celle d'un segment long. Le vote des segments très courts doit donc être pondéré de façon adéquate pour éviter qu'ils ne contribuent trop fortement au profil latéral. Cette sur-représentation des segments courts peut conduire à l'apparition de pics sans existence réelle dans le profil latéral.

Si une pondération est nécessaire, est-ce qu'il y a une méthode systématique pour déterminer les poids adéquats en fonction du type d'histogramme construit? En se fondant sur l'interprétation statistique de ces histogrammes, il est possible de résoudre ce problème.

A.2 Densité de probabilité

Il est réaliste de considérer que les perturbations sur les pixels sont bien modélisées par un bruit uniforme indépendant et identiquement distribué, en position comme en intensité. La caractéristique utilisée en abscisse lors de la construction d'un histogramme est une fonction

des coordonnées géométriques et photométriques des pixels considérés. Si plusieurs caractéristiques sont utilisées, l'histogramme a un support à plusieurs dimensions.

Prenons l'exemple d'un histogramme selon la pente p et la longueur l des segments d'une image. Soit un segment de sommets (u_0, v_0) et (u_1, v_1). La position de chaque sommet est supposée perturbée par un bruit indépendant de variance s^2. Alors $u = u_0 - u_1$ et $v = v_0 - v_1$ sont des variables aléatoires de variance $2s^2$. La densité de probabilité du couple (u, v) est $f(u)f(v)dudv$. Maintenant que nous avons modélisé les données, nous pouvons calculer la densité de probabilités des variables (p, l) utilisées lors de la construction de l'histogramme. La pente et la longueur sont calculées par :

$$p = \frac{u}{v} l = \sqrt{u^2 + v^2}$$

La relation réciproque est alors :

$$\begin{aligned} u &= \frac{pl}{\sqrt{1+p^2}} \\ v &= \frac{l}{\sqrt{1+p^2}} \end{aligned} \tag{A.1}$$

La densité de probabilité du couple (l, p) est donc en utilisant (A.1) :

$$g(p, l) = f(\frac{pl}{\sqrt{1+p^2}}) f(\frac{l}{\sqrt{1+p^2}}) |J(p, l)| dpdl$$

où $|J(p, l)|$ est le déterminant du Jacobien du changement de variable (A.1). Cette densité de probabilité doit être utilisée pour chaque vote dans l'histogramme, si l'on veut tenir compte des perturbations. On remarque dans notre exemple que cette expression est assez simple puisque la densité de probabilité originale est juste multipliée par le déterminant du Jacobien de la transformation. Le déterminant du Jacobien a donc comme rôle de pondérer le vote. Après simplification, on obtient l'expression du Jacobien :

$$J(p, l) = \frac{1}{2} \frac{l}{1+p^2}$$

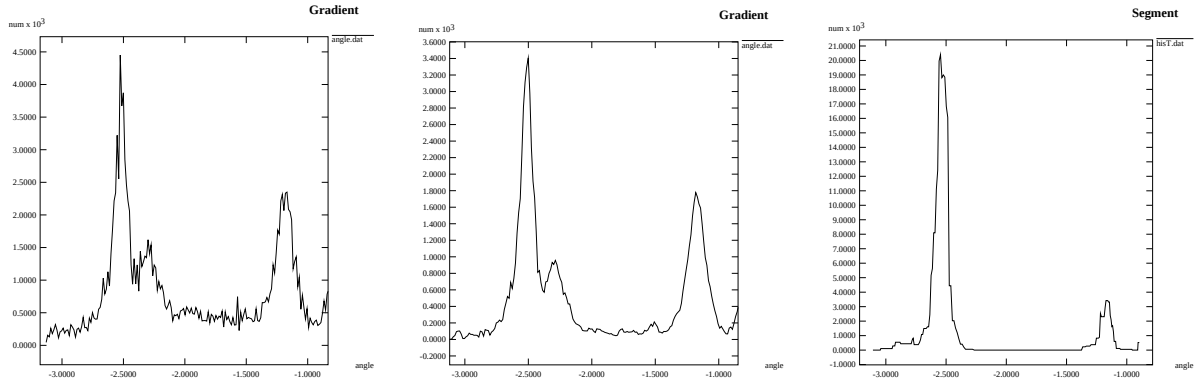
On remarque que J est proportionnel à la longueur l, ce qui rejoint l'intuition que plus la longueur est petite, moins la pente est fiable. On obtient aussi que plus la pente est grande, moins la pente est fiable. Le Jacobien va multiplier le poids du vote. Dans le cas d'un vote proportionnel à la longueur, on obtient comme pondération finale $\frac{1}{2} \frac{l^2}{1+p^2}$.

Plus généralement, le vote doit consister à ajouter la densité de probabilité des erreurs. Le Jacobien de la transformation entre les données dans l'image et les variables du support de l'histogramme permet de pondérer de façon adéquate chaque vote lors de la construction d'un histogramme.

A.3 Vote discretisé

Nous venons de décrire comment réaliser le vote dans le cas continu, mais il faut aussi tenir compte du fait que l'histogramme est représenté sous une forme discrétisée. La discrétisation oblige pour l'essentiel à accumuler dans chaque intervalle, l'intégrale de la densité de probabilité. La construction exacte de l'histogramme conduit donc à de nombreux calculs qui peuvent être superflus lorsque la densité de probabilité f est mal connue.

En pratique, comme nous avons considéré que f était uniforme, nous avons approché $g(p, l)$ par une densité uniforme pondéré par le Jacobien. Par exemple, la variance de u et v est $2s^2$, donc la pente p est approchée par une densité uniforme sur l'intervalle $[\frac{u-\sqrt{2}s}{v+\sqrt{2}s}, \frac{u+\sqrt{2}s}{v-\sqrt{2}s}]$ (si $u > \sqrt{2}s$ et $v > \sqrt{2}s$). Cette densité est normalisée pour avoir une aire égale à un. Ensuite, les pondérations avec le Jacobien et la longueur sont appliquées. Le fait d'utiliser une densité uniforme simplifie l'intégration selon les pas de l'histogramme. Cela permet d'obtenir des histogrammes relativement lisses sans lissage et sans pic fantôme.



(a) (b) (c)

FIG. A.1 – *Histogrammes obtenus : (a) avec les gradients locaux sans prise en compte de l'erreur de digitalisation, (b) avec prise en compte de la digitalisation, (c) avec l'extraction de segment et la prise en compte de la digitalisation.*

Ainsi, dans la Fig. A.1, il est clair que l'histogramme (b) construit avec la prise en compte des erreurs de digitalisation est beaucoup plus lisse que celui visible dans (a). Ces deux derniers sont des histogrammes d'angle du gradient obtenus sur une image de route. La transformation qui fait passer du vecteur gradient mesuré dans l'image à l'angle est polaire, d'où la nécessité de pondérer par la norme du vecteur gradient. A titre de comparaison, sur la Fig. A.1(c), l'histogramme des angles obtenus avec l'extracteur de segments est construit sur la même image. On note que les résultats sont similaires, mais l'histogramme obtenu avec les segments présente moins de bruit de fond et est un peu plus lisse. Un seuillage avec la norme du gradient permet de diminuer le bruit de fond sans compétement l'éliminer. Cela confirme la supériorité d'une sélection géométrique sur une sélection par la norme du gradient.

Annexe B

Système de mesure de la position d'un véhicule par webcam de côté

B.1 Description du système

Pour valider les performances de la voiture automatique, le besoin d'un système facilement montable et démontable de mesure de la position et de l'orientation du véhicule par rapport au bord de la voie, au niveau du véhicule, s'est fait sentir. La précision attendue est de l'ordre du centimètre en position et de l'ordre du degré en orientation.

Le système développé consiste en une webcam branchée sur un ordinateur portable.

La webcam est montée sur une barre de fixation sur le toit centrée de préférence sur le barycentre du véhicule pour éviter les erreurs introduites par le tangage du véhicule. La webcam observe la route et son marquage à la verticale, ce qui permet par un traitement simple de mesurer la pente des marquages par rapport à l'axe du véhicule $tan\beta$ ainsi que l'écart latéral relatif e des marquages par rapport à la verticale de la webcam. L'écart latéral du centre du véhicule par rapport au bord de la voie est donc $e' = e + \delta_0$ où δ_0 est la longueur de la barre de fixation entre la webcam et le centre du véhicule.

La mesure de la pente $tan\beta$ est obtenue par corrélation de deux lignes de l'image. L'écart e est calculé à partir du centre du créneau que la ligne blanche forme sur l'histogramme d'une moyenne de lignes de l'image.

B.2 Utilisation du système portable

1. Il faut disposer d'un ordinateur avec le système Windows 98 au minimum et une prise USB. Le driver de la webcam à utiliser doit être installé ainsi que DirectX Media. Les activeX *calib.ax* et *algo.ax* doivent être présents et avoir été déclarés dans la base de registre de l'ordinateur avec *regsrv32.exe*.

2. Lancer la commande *graphedit* du menu DirectX Media et charger le filtre *ecart.graph*. Si celui-ci n'existe pas le créer en enchaînant les 4 filtres suivants :

   (a) filtre de capture de l'image,

   (b) filtre intitulé *LIVIC calibration toolbox*,

FIG. B.1 – *Exemple d'installation du système.*

(c) filtre intitulé *LIVIC algo toolbox*,

(d) filtre de rendu de l'image résultat.

3. Lancer l'application, une fenêtre avec l'image de la scène observée apparaît.

4. En cliquant avec le bouton de droite, modifier les propriétés du premier filtre *LIVIC calibration toolbox*, et sélectionner *Radio Calib*. Si elle n'a jamais été effectuée, on passe en mode calibrage radiométrique. Durant le mode de calibrage radiométrique, la caméra doit observer une surface de couleur uniforme éclairée de façon uniforme comme l'alsphalte de la route. A la fin, l'image passe en niveaux de gris.

5. Modifier les propriétés du deuxième filtre *LIVIC algo toolbox* et sélectionner *Vel. Pose*, pour débuter la mesure de l'orientation et de la position du véhicule. Le centre de la ligne blanche détectée s'affiche sous la forme d'une ligne bleue.

6. Un fichier nommé *ecart.txt* contient les mesures. Sur chaque ligne, on y trouve dans l'ordre le temps en secondes, la pente, le coefficient de confiance sur la pente, l'ecart latéral en mètres, et enfin le coefficient de confiance sur l'écart latéral.

Voici un exemple de fichier de mesure :

```
%Start Time 6.047
%Time slope slope_conf pose pose_conf
```

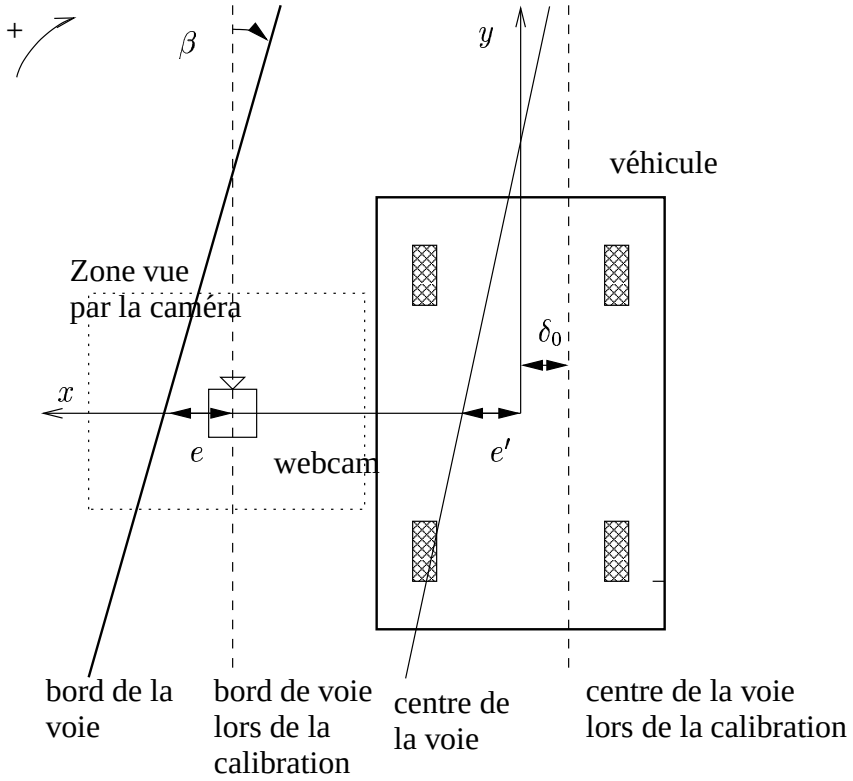


FIG. B.2 – *Repère de référence.*

```
6.047 -0.016020 2 +0.373837 7
6.110 -0.016891 2 +0.374163 7
6.141 -0.015472 4 +0.373747 7
6.219 -0.014632 6 +0.373637 7
```

Deux paramètres peuvent être modifiés lors des mesures (mode *Vel. Pose*):

– Le premier correspond à la taille en pixel de la partie de l'image à traiter. Cette taille est mise par défaut à 64 pixels pour pouvoir faire des mesures avec des rayons de courbure supérieurs à 10m. Si l'on sait que les rayons de courbures de la route sont supérieurs à 20m, on a intérêt à doubler ce paramètre pour que toute l'image soit traitée.

– Le deuxième paramètre correspond à la mesure en mètres de la distance entre le bord gauche et droit de l'image retroprojetée sur le plan de la route. Ce paramètre doit être remesuré dès que l'on change la hauteur de la webcam par rapport au sol.

B.3 Procédure de vérification du système portable

La procédure de vérification suivante doit être suivie après montage du système :

1. Vérifier par mesure que la barre de fixation de la webcam est bien perpendiculaire à l'axe du véhicule. Ceci assure le parallélisme entre l'axe du véhicule et l'axe vertical de la matrix CCD de la webcam matérialisé par des traits rouges sur l'image. Cette étape réalise le zéro mécanique en orientation.

2. Mettre la webcam en butée mécanique. Cette étape réalise le zéro mécanique en position latérale. L'axe optique de la caméra est maintenant aligné avec la verticale.

3. Mesurer la distance δ_0 entre le milieu du véhicule et la webcam le long de la barre de fixation.

Après cette procédure, l'écart latéral du centre du véhicule par rapport au bord de la voie est donné par $e + \delta_0$.

Bibliographie

[1] E. D. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proceedings of SPIE Conference on Mobile Robots*, volume 727, pages 161–168, October 1986.

[2] C. Thorpe, M. Herbert, S. Shafer, and T. Kanade. Vision an navigation for the canegie-mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):361–372, May 1988.

[3] A. Broggi. Parallel and local feature-extraction: A real-time approach to road boundary detection. *IEEE Transactions on Image Processing*, 4(2):217–223, February 1995.

[4] R. Chapuis, A. Potelle, J.L. Brame, and F. Chausse. Real-time vehicle trajectory supervision on the highway. *International Journal of Robotics Research*, 14(6):531–542, December 1995.

[5] J.-P. Tarel, F. Guichard, and D. Aubert. Procédé et programme informatique de traitement d'images numérisées fondé sur des critères uniquement géométriques, utilisation de ce procédé pour le contrôle d'un déplacement et dispositif pour sa mise en oeuvre. Demande de Brevet Européen par l'INRETS n° 01400726.4, Mars 2001.

[6] R. Deriche. Using canny's criteria to derive an optimal edge detector recursively implemented. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[7] B.K.P. Horn. *Robot Vision*. The MIT Press. Mc Graw-Hill Book Company, 1986.

[8] F. Guichard. *Axiomatization of images and movies scale-space*. PhD thesis, University of Paris IX-Dauphine, 1994.

[9] L.D. Wu. On the chain code of a line. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(3):347–353, May 1982.

[10] C.A. Berenstein and D. Lavine. On the number of digital straight line segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):880–887, November 1988.

[11] Jean-Philippe Tarel and André Gagalowicz. Calibration de caméra à base d'ellipses. *Traitement du Signal*, 12(2):177–187, 1995.

[12] T. Tasdizen, J.-P. Tarel, and D.B. Cooper. Improving the stability of algebraic curves for applications. *IEEE Transactions on Image Processing*, 9(3):405–416, March 2000. Also as LEMS Tech. Report 176, Brown University.

[13] J.-P. Tarel and D. B. Cooper. The complex representation of algebraic curves and its simple exploitation for pose estimation and invariant recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):663–674, July 2000.

[14] J.-P. Tarel, H. Civi, and D. B. Cooper. Pose estimation of free-form 3D objects without point matching using algebraic surface models. In *Proceedings of IEEE Workshop on Model-Based 3D Image Analysis*, pages 13–21, Mumbai, India, 1998.

[15] J. Estève. Développement en C d'un logiciel de positionnement latéral d'un véhicule pour l'assistance et l'automatisation de la conduite. Rapport de stage IUP Génie Mathématique et Informatique, Université de Parix IX Dauphine, 1999.

[16] P. Charbonnier and P. Nicolle. Vision 3D : compte-rendu des travaux 1999. Rapport technique, LCPC, 1999.

[17] V. Badal Baghdassarians. *Commandes Robustes pour le contrôle latéral d'un véhicule routier*. PhD thesis, Université d'Evry-Val d'Essone, 2001.

[18] J. Lavenant. Détection et caractérisation du brouillard. Rapport de stage DEA image, Université Jean Monnet, St Etienne, 2001.

[19] J. Lavenant, J.-P. Tarel, , and D. Aubert. Procédé de détermination de la distance de visibilité et/ou d'un brouillard. Demande de dépôt de Brevet Européen par l'INRETS/LCPC, Mars 2002.

[20] A. Grosu. Utilisation de la vision monoculaire pour la détection de situation non-nominale d'une voiture. Rapport de stage DEA SETI, Université d'Orsay, 2000.

[21] K. Kang, J.-P. Tarel, R. Fishman, and D.B. Cooper. A linear dual-space approach to 3d surface reconstruction from occluding contours using algebraic surfaces. In *IEEE International Conference on Computer Vision (ICCV'01)*, volume 1, pages 198 – 204, Vancouver, Canada, 2001.

[22] D.S. Chen. A data-driven intermediate level feature extraction algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):749–758, July 1989.

[23] F. Guichard and J.-P. Tarel. Curve extraction combining perceptual grouping and a kalman like fitting. In *IEEE International Conference on Computer Vision (ICCV'99)*, pages 1003–1008, Kerkyra, Greece, 1999.

[24] A. Shashua and S. Ullman. Grouping contours by iterated pairing network. *Advances in Neural Information Processing Systems*, 3:335–341, 1991.

[25] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):168–185, February 1998.

[26] T.D. Alter and R. Basri. Extracting salient curves from images: An analysis of the saliency network. *International Journal of Computer Vision*, 27(1):51–69, March 1998.

[27] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition (CVPR'85)*, pages 22–26, 1985.

[28] J.-P. Tarel and F. Guichard. Dynamic tracking of lane markings and boundaries in curved road. In *IEEE International Conference on Image Processing (ICIP'2000)*, volume I, pages 216–219, Vancouver, Canada, 2000.

[29] M. Gondran and M. Minoux. *Graphes et algorithmes*. 3 ed. Eyrolles, Paris, 1995.

[30] D. Aubert and C.E. Thorpe. Color image processing for navigation: Two road trackers. In *CMU Robotics Institute Technical Report 90-09*, April 1990.

[31] P. Charbonnier, P. Nicolle, Y. Guillard, and J. Charrier. Road boundaries detection using color saturation. In *Proceedings of European Signal Processing Conference (EUSIP-CO'98)*, pages 2553–2556, Rhodes, Greece, 1998.

[32] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection in stereo vision on non-flat road geometry through v-disparity representation. In *IEEE Intelligent Vehicle Symposium (IV'2002)*, Versailles, France, 2002. http://www-rocq.inria.fr/ tarel/iv02.html.

[33] J.-P. Tarel, F. Guichard, and D. Aubert. Tracking occluded lane-markings for lateral vehicle guidance. In *IEEE International Conference Circuits, Systems, Communications and Computers (CSCC'99)*, pages 154–159, Athens, Greece, 1999.

[34] J. Fabrizio. Copilote intelligent basé sur l'utilisation d'un capteur panoramique catadioptrique. Rapport de stage DEA RESIN, Université Pierre et Marie Curie, 2001.

[35] J. Fabrizio, J.-P. Tarel, and R. Benosman. Calibration of panoramic catadioptric sensors made easier. In *IEEE Workshop on Omnidirectional Vision (Omnivis'02)*, pages 45–52, Copenhagen, Denmark, 2002. http://www-rocq.inria.fr/ tarel/omnivis02.html.

[36] J.-P. Tarel, S.-S. Ieng, and P. Charbonnier. Using robust estimation algorithms for tracking explicit curves. In *European Conference on Computer Vision (ECCV'02)*, volume 1, pages 492–507, Copenhagen, Danmark, 2002. http:///www-rocq.inria.fr/ tarel/eccv02.html.