MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et de l'Informatique
Département de Mathématiques et d'Informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : **Ingénierie des Systèmes d'Information**

THEME :

Détection de la voie de roulement

Etudiant : Hamdi cherif Fethi

Encadrant : Boumedienne Mohammed

Année Universitaire 2015/2016

Résumé

Actuellement la nécessité des systèmes informatiques dans le domaine de sécurité routière s'agrandissent d'une manière exponentielle, leurs utilitésde détection et d'intervention contre les accidents routières reposent sur plusieurs techniques ; dans un premier temps il s'agit de capturer la route par une caméra embarquée sur le véhicule, dans un deuxième temps on applique les méthodes de traitement d'image afin de localiser les bords de la voie dans l'image capturée, et enfin on déduisant la position de la voiture et on lance un avertissement s'il y a un risque de sortie de voie.

Mots clés : Détection de voie de roulemen, transformée de hough, tranformée IPM, point de fuite.

REMERCIMENTS

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Jetender à exprimer toute ma reconnaissance à mon Directeur de mémoire Monsieur ***Boumedienne Mohammed****. Je le remercie de m'avoir encadré, orienté et aidé. J'ai profité pendant longtemps du savoir et du savoir-faire dont j'ai pu bénéficier au cours de nombreuses discussions. J'aimerais aussi leremercier pour l'autonomie qu'il m'a accordé, et ses précieux conseils qui m'ont permis de mener à bien ce travail.*

J'exprime toute ma reconnaissance à mes jurys de la faculté des sciences exactes et de l'informatique de l'université d'Abd El Hamid Ibn Badis, trouvent ici l'expression de mes vifs remerciements pour avoir bien voulu juger ce travail.

J'adresse mes sincères remerciements à tous les intervenants et toutes les personnes qui par leurs paroles, leurs contributions dans ce travail, leurs écrits, leurs conseils et leurs critiques ont guidé mesréflexions et ont accepté de répondre à mes questions durant mes recherches. Je ne peux achever ce projet, sans exprimer mes sincères gratitudes à tous les professeurs de notre faculté, pour leur dévouement et leur assistance tout au long de ma formation.

Je remercie mes très chers parents, qui ont toujours été là pour moi,
« Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous m'avez données un magnifique modèle de labeur et de persévérance. Je suis redevables d'une éducation dont me suis fières ».

Enfin, Je remercie très spécialement ma famille pour leur encouragement.

À tous ces intervenants, Je présentemes remerciements, mes respects et ma gratitude.

DEDICACE

Au nom du dieu le clément et le miséricordieux louange à **ALLAH** *le tout puissant.*

Je dédie ce modeste travail en signe de respect, reconnaissance et de remerciement :

A mes parents,

Sans leurs soutiens et leurs conseils, mes accomplissements n'auraient pas eu lieu, ils ont été derrière moi dans chacun de mes pas tout au long de ma vie, ma plus profonde gratitude leurs ai exprimé, aucun mot ne pourrait qualifier l'estime que je leur porte ni le bien qu'ils m'ont fait, apporter et donner.

A tous mes ami(e)s nacer, saidahmed, messoud, abderrahmen, et tous ceux qui me sont chers.

Fethi,

Sommaire

Chapitre II : Etat de l'art de détection de la voie de roulement

Chapitre III : Conception et mise en œuvre

Chapitre VI : Application

Liste des figures :

Liste des abréviations

ADAS: *Addvanced Driving Assistant System.*

IPM: *Inverse Perspective Mappin.*

CHEVP: *Canny/Hough Estimation of Vanishing Points.*

MMSE: *Minimum Mean Square Error.*

MAAGS: *MAximum Gradient Square Sum.*

RANSAC: *RANdom SAmple Consensus.*

EDF: *Edge Distribution Function.*

LBROI: *Lane Boundary Region Of Interest.*

POO: *Programmation Orientée Objet.*

OpenCV: *Open source Computer Vision.*

ANSI: *American National Standards Institute.*

RAD: *Rapid Application Development.*

OpenCV: *Open source Computer Vision.*

2D : *2 Dimensions (dans l'espace).*

3D *: 3 Dimensions (dans l'espace).*

Introduction générale :

La sécurité routière demeure une question primordiale pour les pouvoirs publics. En effet, les accidents de la route tue chaque année dans le monde plus 1,24 million de personne et ce chiffre ne cesse d'augmenter. Face à ce constat alarmant, même les industriels et les équipementiers automobiles prennent en considération cet aspect sécuritaire lors de la conception de nouveaux véhicules en intégrant des dispositifs d'aide à la conduite. L'objectif de ces derniers est l'évaluation du risque d'accident afin de l'éviter ou de réduire ses conséquences. Pour cela, ces systèmes d'aide à la conduite perçoivent l'environnement routier et l'interprète ce qui permet d'offrir une assistance au conducteur dans sa tâche de conduite.

Parmi les systèmes développés d'aide à la conduite il y a l'avertisseur de sortie de voie. Ce système contrôle en continu la position latérale du véhicule et détecte toute sortie de voie involontaire. L'avertisseur est basé sur une caméra embarquée sur l'avant du véhicule ce qui assure une perception de l'environnement routier notamment le marque au sol. Par la suite les limites ou les bords de la voie de roulement sont détectés dans les images fournies par la caméra embarquée. Cette détection permet d'estimer la position latérale du véhicule dans la voie. Si le véhicule dévie trop de sa trajectoire idéale, le système informe le conducteur d'une éventuelle sortie de voie.

Dans ce projet, nous nous intéressons à la détection de la voie de roulement. Le mémoire est organisé comme suit :

Chapitre 1 : nous présentons le principe d'un avertisseur de sortie de voie ainsi que ses étapes.

Chapitre 2 : nous présentons un état de l'art de la détection de la voie.

Chapitre 3 : nous discutons notre approche adoptée pour la détection de la voie.

Chapitre 4 : nous présentons notre application développée qui détecte la voie de roulement.

CHAPITRE I :

Introduction aux avertisseurs

De sortie de voie.

I.1 Introduction :

De notre jour, la sécurité routière reste l'un des enjeux majeurs des constructeurs automobiles. En effet, l'objectif est de réduit le risque d'accident tout en garantissant un confort de conduite aux conducteurs. Cependant, les collisions de véhicules demeurent la principale cause de décès et de blessures accidentelles dans la plupart des pays ayant une circulation congestionnée comme le Royaume-Uni, les États-Unis, et les pays asiatiques [1]. C'est pourquoi, les industriels automobiles intègrent de plus en plus de dispositifs d'aide à la conduite lors de la conception de nouveaux véhicules. Ces dispositifs, les systèmes avancés d'aide à la conduite communément appelés *ADAS (Advenced Driving Aide System)*, assiste le conducteur dans sa tâche de conduite.

Un ADAS est un système qui détecte les situations potentiellement dangereuses et avertit le conducteur afin de pouvoir éviter l'accident. L'une des principales technologies impliquées dans ce système est la vision par ordinateur, qui est un outil puissant pour percevoir l'environnement routier.

La vision par ordinateur a été largement utilisée dans les applications de transport intelligent comme le détecteur d'obstacles, et l'avertisseur de sortie de voie [2].La figure 1 illustre ces deux ADAS. Le premier dispositif détecte les véhicules présents dans l'environnement du conducteur et tient ce dernier informé afin de pouvoir éviter toute collision. Quant au deuxième dispositif, il détecte la voie de roulement afin de maintenir une trajectoire latérale idéale pour le véhicule et signale, au conducteur, toute sortie de voie involontaire.

Ce deuxième type d'ADAS perçoivent la scène routière via une caméra embarquée sur l'avant du véhicule. Par la suite, les données de la caméra sont traitées afin de détecter les limites de la voie de roulement. Dans le cas où le véhicule dévie de sa trajectoire idéale, le système avertir le conducteur [3].

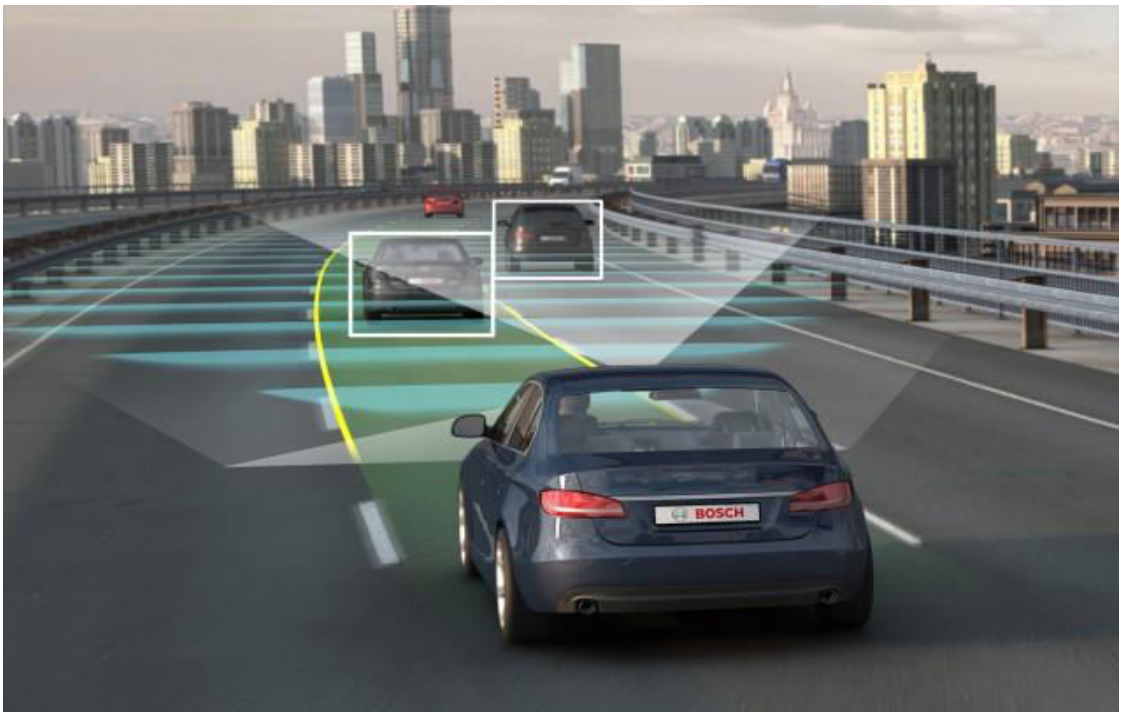


Figure I.1 : Systèmes de détection de véhicules et de détection de la voie de roulement.

I.2 Avertisseur de sortie de voie:

Les lignes de marquage au sol constituent un dispositif sécuritaire mis en place par les gestionnaires de voiries pour éviter les sorties de routes et les collisions frontales. Le système détecte par traitement d'images les lignes de marquage au sol qui délimitent la voie de roulement (les bords de voie), ensuite détermine la position latérale du véhicule par rapport à ces lignes. Si le véhicule dévie trop fortement de sa trajectoire idéale, le système envoie au conducteur une mise en garde, sous la forme d'un signal sonore ou tactile, avant son éventuelle sortie de voie [1].

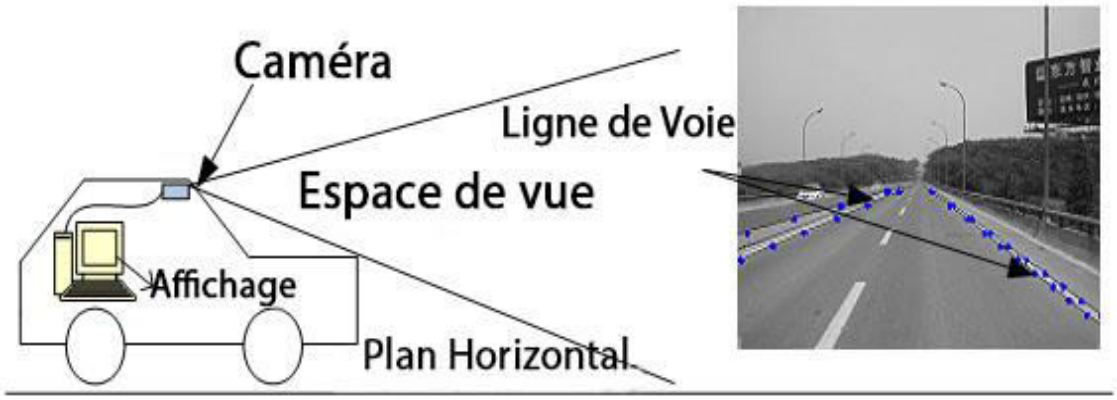


Figure I.2 : Avertisseur de sortie de voie [4].

Un avertisseur de sortie de voie nécessite l'exécution de trois processus :acquisition, détection de la voie de roulement, et prise de décision.

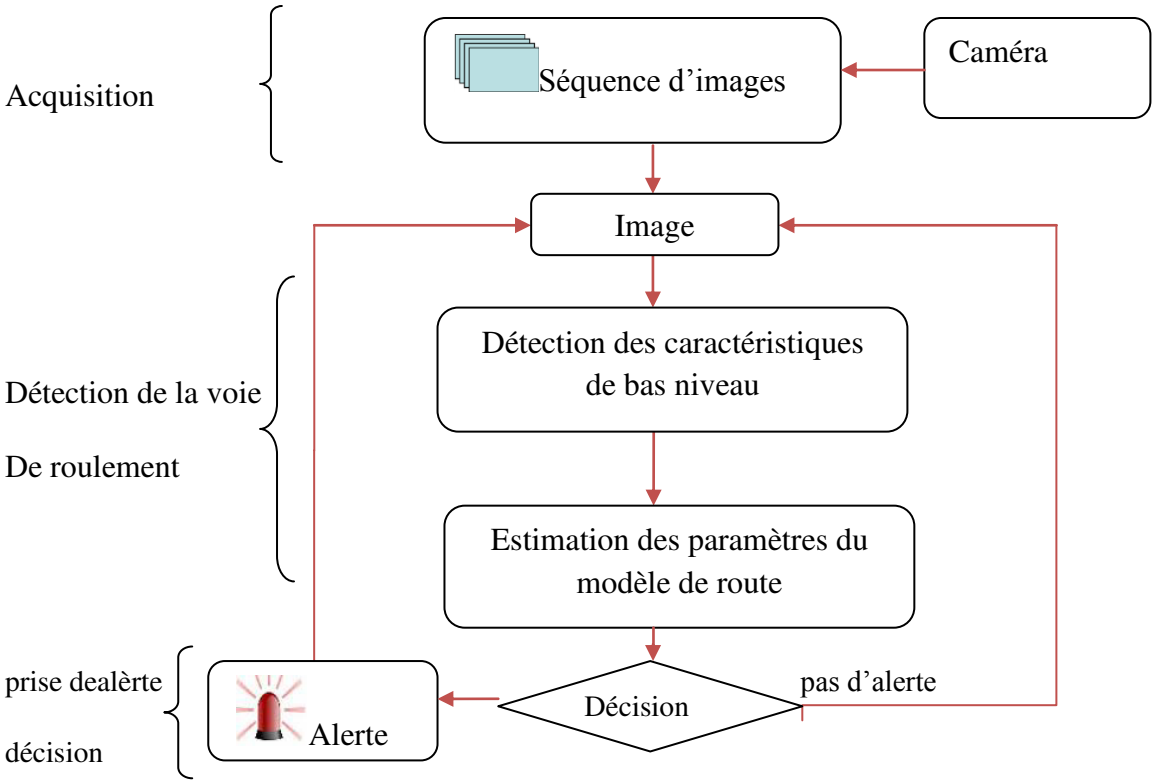


Figure I.3 : Architecture d'un avertisseur de sortie de voie [3].

I.2.1 Acquisition :

Pour la perception de l'environnement routier, une caméra est embarquée sur l'avant du véhicule. Ce processus d'acquisition fournit un flux d'images qui seront traitées par le processus de détection de la voie de roulement.

Figure I.4: Caméra frontale embarquée sur un véhicule [3].

I.2.2 Détection de la voie de roulement:

La détection de la voie de roulement se base sur deux étapes fondamentales : extraction des caractéristiques de bas niveau(les lignes blanches, les contours, couleurs…), et l'estimation des paramètres du modèle des bords de la route.

Différents algorithmes ont été proposés pour la détection de la voie et chacun d'entre eux utilise des paramètres spécifiques [3]:

- Type des lignes de marquage (continu, discontinu).
- Modèle de la voie (linéaire, parabolique).
- Type de caméra (couleur, monochrome).
- Différentes méthodes (transformée de Hough, réseau de neurones, détection de contour, segmentation).

I.2.3 Prise de décision :

L'utilité de ce processus est de décide si la véhicule risque de sortir de la voie de roulement ou pas. Le système calcule en permanent un temps noté TLC (Time to Line Crossing ou temps avant coupure de la ligne) [5] puis le compare à seuil prédéfini. Le système alerte le conducteur si le TLC est en dessous du seuil [3].

I.3 Les contraintes des algorithmes de détection de la voie [3]:

Les contraintes sont l'ensemble des limites qui empêche la bonne performance d'un système détection de la voie :

I.3.1Variabilité de l'éclairage : La perception de la route est souvent perturbée par des grandes variations de contraste dues, par exemple, à l'ombre d'un pont, des arbres ou des parties mouillées de la chaussée. La grande variabilité des conditions d'éclairage rend la détermination de la géométrie de la voie de roulement plus délicate. Pour être robuste à ces variations d'illumination, les images doivent être traitées de façon invariante aux changements de contraste locaux.

Figure I.5: Variabilité de l'éclairage dans les images.

I.3.2Type de marquage : Le type de marquage peut influencer sur la méthode de détection. Par exemple un marquage discontinu ou usé rend la tâche plus délicate qu'un marquage continu.

Aussi les marquages autres que ceux des bords de la voie, comme les flèches de rabattement, peuvent engendrer des fausses détections.

I.3.3 Présence d'autres véhicules *:* lors des dépassements sur la scène, le véhicule masque une partie du marquage, cela représente un manque d'information que l'algorithme doit surmonter.

Figure I.6 : Image en présence d'un autre véhicule.

I.3.4 Contrainte du temps réel : La procédure de détection et du suivi de la voie de roulement par vision nécessite des mesures à une fréquence minimale de dix images par seconde pour obtenir des résultats satisfaisants. Donc le traitement de chaque image doit toujours être effectué avant la capture de la prochaine image, cela oblige l'utilisation d'algorithmes dont le nombre d'opérations n'est pas élevé.

I.4 Principe de la détection de la voie de roulement :

La détection de la voie est la deuxième phase d'un avertisseur de sortie de voie, on a déjà dit que cette tâche se fait en deux étapes : détection des paramètres de bas niveau (extraction du marquage) et l'estimation des paramètres du modèle de la voie (modélisation des bords de la voie) :

I.4.1 Extraction du marquage:

Il y 'a plusieurs algorithmes bien connus dans le domaine du traitement d'image qui permettent l'extraction des contours comme Canny, Sobel, Laplacien, Prewitt…etc. Le plus utilisé est le filtre de Canny(1986) [6] et cela pour ces bonnes performances:

- bonne détection : faible taux d'erreur.
- bonne localisation : minimum distances entre les contours détectés et les contours réels.

- clarté de la réponse : une seule réponse par contour et pas de faux positifs.

La mise en œuvre de filtre de Canny est comme suivant [6]:

- Réduction de bruit : par le filtrage de gaussien on réduit le bruit de l'image avant d'en détecter les contours.
- Gradient d'intensité : après le filtrage de bruit, l'étape suivante est d'appliquer un gradient *ΔI* qui retourne l'intensité des contours. l'équation utilisée est :

$$\mathrm{Gx} = \frac{\partial I}{\partial x} = (-1\ 0\ 1) \qquad (01)$$

- Direction des contours : les orientations des contours sont déterminées par la formule :

$$\theta = \arctan\left(\frac{Gx}{Gy}\right) \qquad (02)$$

Nous obtenons finalement une carte des gradients d'intensité en chaque point de l'image accompagnée des directions des contours.

- Suppression des non-maximas : La carte des gradients obtenue précédemment fournit une intensité en chaque point de l'image. Une forte intensité indique une forte probabilité de présence d'un contour. Toutefois, cette intensité ne suffit pas à décider si un point correspond à un contour ou non. Seuls les points correspondant à des maxima locaux sont considérés comme correspondant à des contours, et sont conservés pour la prochaine étape de la détection.
- Seuillage des contours : La différenciation des contours sur la carte générée se fait par seuillage à hystérésis. Cela nécessite deux seuils, un haut et un bas; qui seront comparés à l'intensité du gradient de chaque point. Le critère de décision est le suivant. Pour chaque point, si l'intensité de gradient de chaque point est inférieure au seuil bas, le point est rejeté, si elle est supérieure au seuil haut, le point est accepté comme formant un contour, si elle est entre le seuil bas et le seuil haut le point est accepté s'il est connecté à un point déjà accepté.

La figure II.3du titre «Méthode basée sur les B-snakes » représente un exemple d'application de filtre de canny.

I.4.2 Modélisation des bords de la voie [3]:

Dans le but de crée un système de détection des lignes de la voie plus efficace, on modélise la forme des bords de la voie par un modèle géométrique. Donc le problème de détection de voie devient un problème d'estimation où il faudra estimer les paramètres d'un modèle donné qui impose des contraintes globales sur la forme des bords, il existe trois modèles de la route adoptés par les algorithmes de détection: 2D linéaire, 2D parabolique non linéaire, et tridimensionnel.

I.4.2.1 Le modèle 2D linéaire: Ce type de modèle considère que la route est plane et les bords sont parallèles et droits. Chaque bord est représenté par l'équation suivante :

$$y = ax + b \qquad (3)$$

Avec *a* : la pente de la droite.

b : la valeur de y lorsque x=0.

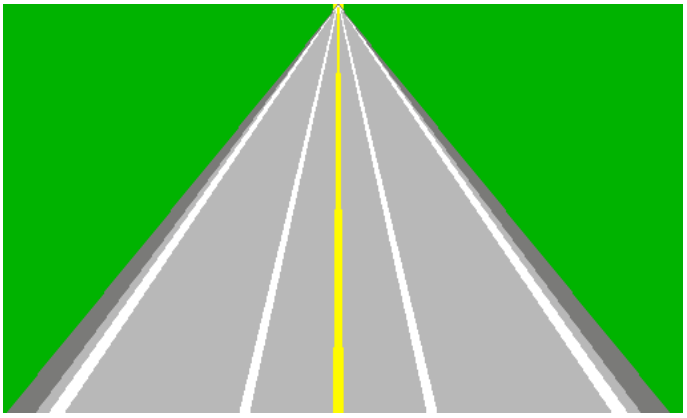


Figure I.7 : Modélisation 2D linéaire de la route.

L'inconvénient de ce modèle est que ne peut pas modéliser les routes courbées, pour résolu ce problème on cumulant des segments de droite pour obtenir un marquage continu.

I .4.2.2 Le modèle 2D parabolique non linéaire *:* c'est le modèle le plus utilisé, parce qu'on peut modéliser la route courbée par une parabole, splines ou par d'autres approximations. L'équation générale qui exprime ce modèle de la route est :

$$y = ax^2 + bx + c \qquad (4)$$

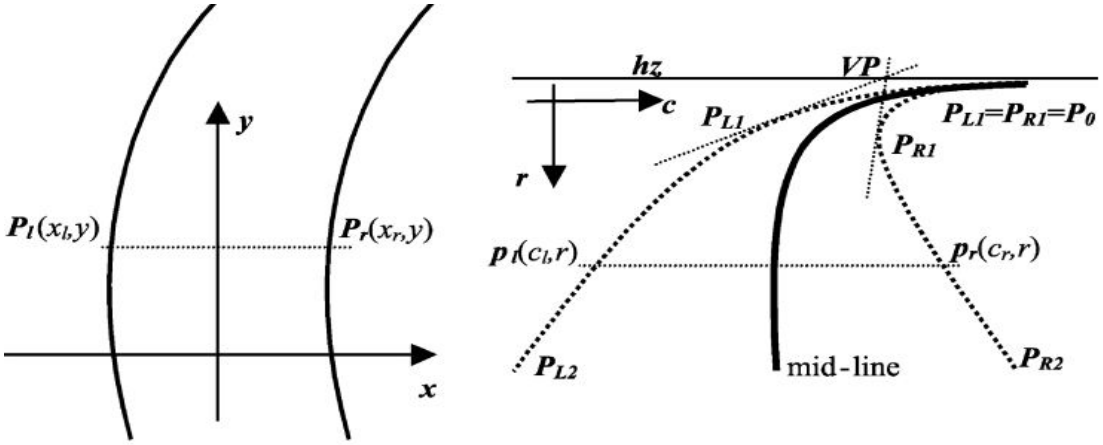


Figure I.8: Les lignes parallèles de la route et sa modélisation 2D parabolique non linéaire.

I.4.2.3 Le modèle 3D :

Ce modèle permet de modéliser les courbures verticale et horizontale de la route, certains paramètres sont de plus supposés connus et constants (largeur de la route ou angle d'inclinaison de la caméra), pourtant ce modèle est plus récent, mais il reste le plus difficile à la détection, si la route n'est pas plane, ou un véhicule présente sur la route ou un bruit, la modélisation serait impossible [7].

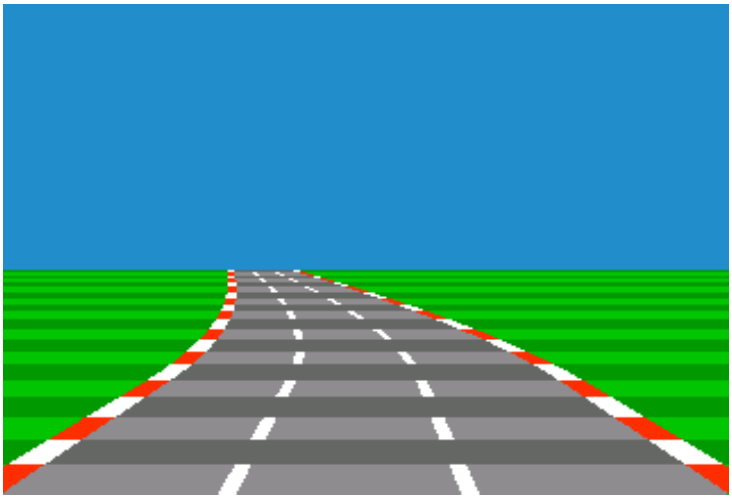


Figure I.9 : Modélisation 3D de la route.

I.4.2.4 Avantages de la modélisation des bords de la route : La modélisation de la route a plusieurs avantages :

- Une réduction importante de la zone de recherche, qui se résume au voisinage du modèle appliqué.
- Robustesse aux perturbations de l'environnement.
- Séparation de la détection des bords gauche et droit de la voie, et donc il y'aurait une possibilité de distribuer les tâches sur plusieurs unités de calcul ce qui diminuera le temps de traitement.

I.5 L'objectif du projet :

Notre objectif dans ce mémoire est d'effectuer une étude sur les algorithmes proposés pour la détection de la voie de roulement par vision. Par la suite nous implémentons un des algorithmes et nous le testons sur une séquence d'images.

I.6 Conclusion :

Dans ce chapitre nous avons présenté l'architecture générale d'un avertisseur de sortie de voie. Ce dernier est basé sur trois processus : acquisition, détection de la voie, et prise de décision. Le principal processus est la détection de la voie qui nécessite une modélisation de cette dernière afin de mieux localiser le véhicule par rapport aux marquages au sol. Plusieurs modèles ont été utilisés pour la modélisation de la voie de roulement tel que le modèle linéaire, parabolique et le modèle 3D.

Le chapitre suivant présentera un état de l'art des algorithmes de détection de la voie de roulement.

CHAPITRE II :

Etat de l'art de détection

De la voie de roulement.

II.1 Introduction :

Aucun principe unique ne pouvait répondre efficacement aux besoins de toutes les applications de détection. Un certain nombre de techniques de traitement d'images ont été proposées, sémantiques et morphologiques. Chaque algorithme à ses forces et ses limites, et en conséquence, chaque technique de traitement d'images est utilisée dans une application particulière. Pour les caractéristiques Morphologiques (physiques), nous décrirons la reconnaissance la localisation des bords de la route, les contrastes, la géométrie des lignes et le positionnement géométrique pour les caractéristiques comportementales.

II.2.Quelques algorithmes proposés dans la littérature :

II.2.1 Algorithme basée sur la méthode IPM (Inverse Perspective Mapping) [3]:

La méthode IPM permet de reproduit une image représentant la route vue d'en haut, c-a-d vue de oiseau (figure II.1).IPM est une transformation qui supprime les effets de perspective ce qui permet de visualiser les bords de la route comme étant des lignes droites parallèles. La figure suivante représente le résultat de cette transformation.

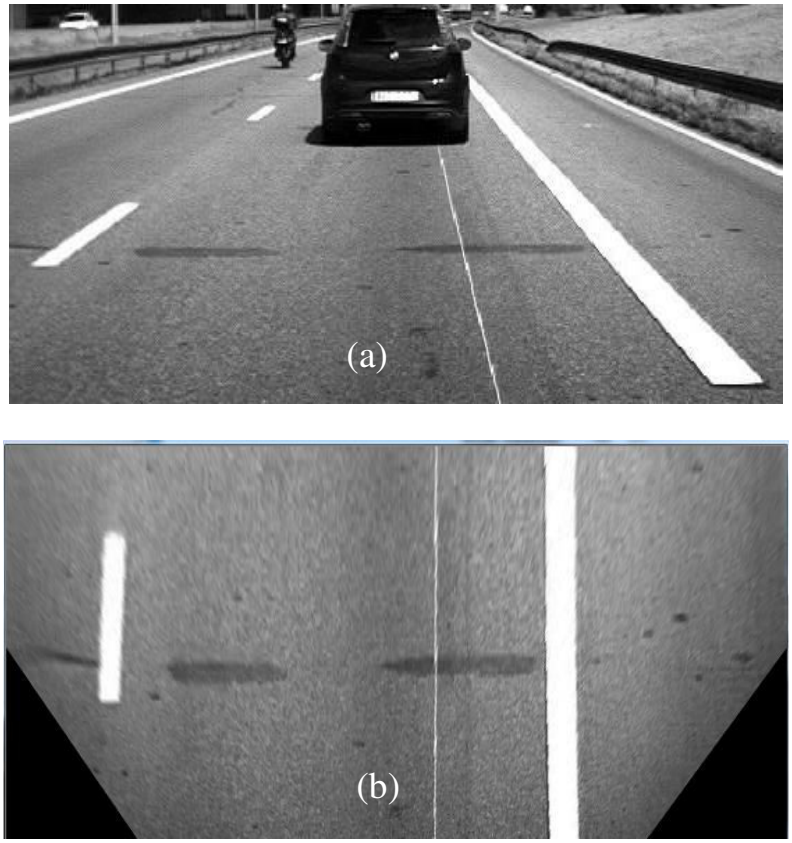


Figure II.1 : La transformée inversée par la méthode IPM, (a) Image vue de scène, (b) Image vue d'oiseau.

Cette approche supprime le point de fuite ou de poursuite, représentant le croisement des bords dans l'image de la scène routière, et donc la représentation proposée considère les

lignes de marquage comme portion droite ou circulaire, de cette manière on peut modéliser la voie par un vecteur de trois paramètres :

$$X = \begin{bmatrix} \frac{1}{R} \\ \theta \\ d \end{bmatrix} \quad (5)$$

Où : R : la courbure de la voie, qui peut être infinie si la voie est droite.

θ : l'angle entre la voie de roulement et la direction du véhicules.

d : la distance entre le bord droit de la voie et la position de la caméra.

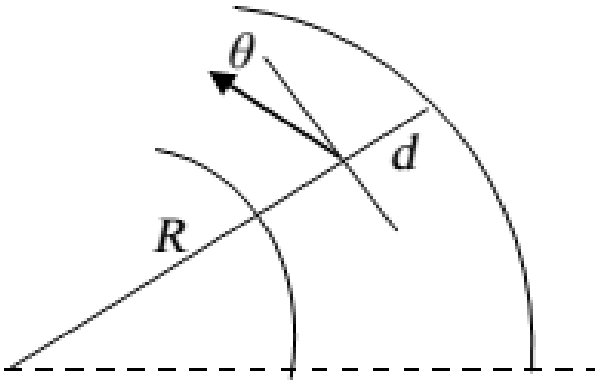


Figure II.2 : Modélisation de la voie par *R, θ, d.*

Les valeurs des paramètres *R, θ, d son*t estimé à pâtir d'une fonction de vraisemblance entre une image binaire représentant le modèle décrit et une image de contour obtenue par filtre de sobel appliqué sur l'image vue d'oiseau.

II.2.2 Méthode basé sur les B-Snakes [8]:

Cette méthode est basée sur les B-Splines cubiques et des contours actifs (Snakes), et adopte le modèle 2D parabolique non linéaire [7]. Son application s'effectue en deux étapes : initialisation du modèle et sa mise à jour.

II.2.2.1 Initialisation du modèle :

L'initialisation s'effectue grâce à l'algorithme CHEVP (Canny/Hough Estimation of Vanishing-points), qui permet la détection des bords de la route ainsi que le point de poursuite. Les étapes de l'algorithme sont :

1. **l'extraction des contours en utilisant le filtre de Canny.**

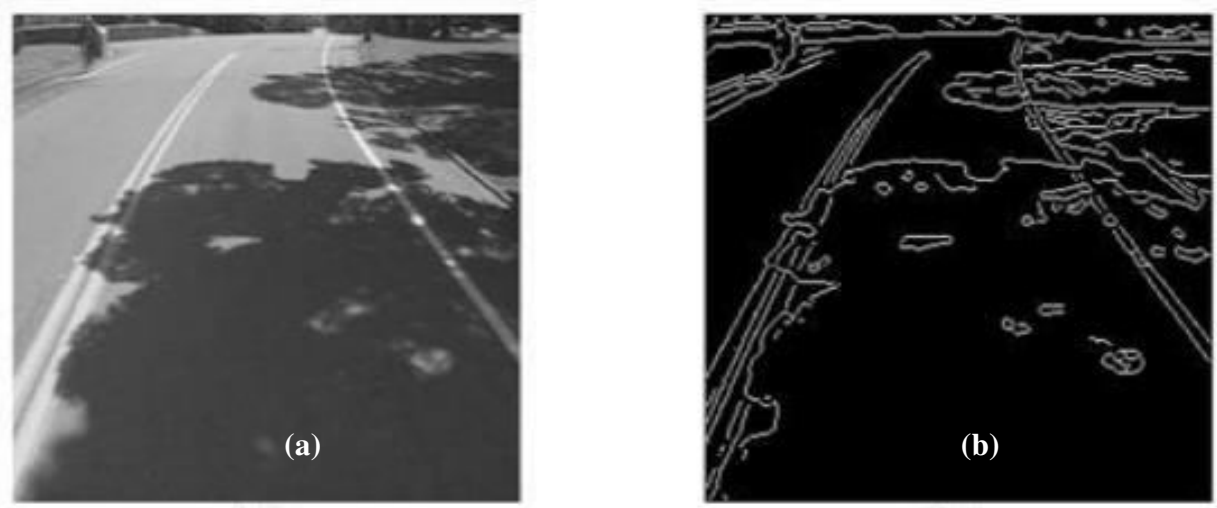


Figure II.3 : Détection de contour par le filtre de Canny, (a) Image au niveau de gris, (b) L'image en filtre de Canny.

2. La détection des lignes droites par la transformée de Hough.
3. Détection de l'horizon et du point de poursuite.
4. L'estimation du centre de la route et du paramètre k.
5. L'initialisation des points de contrôle Qi.

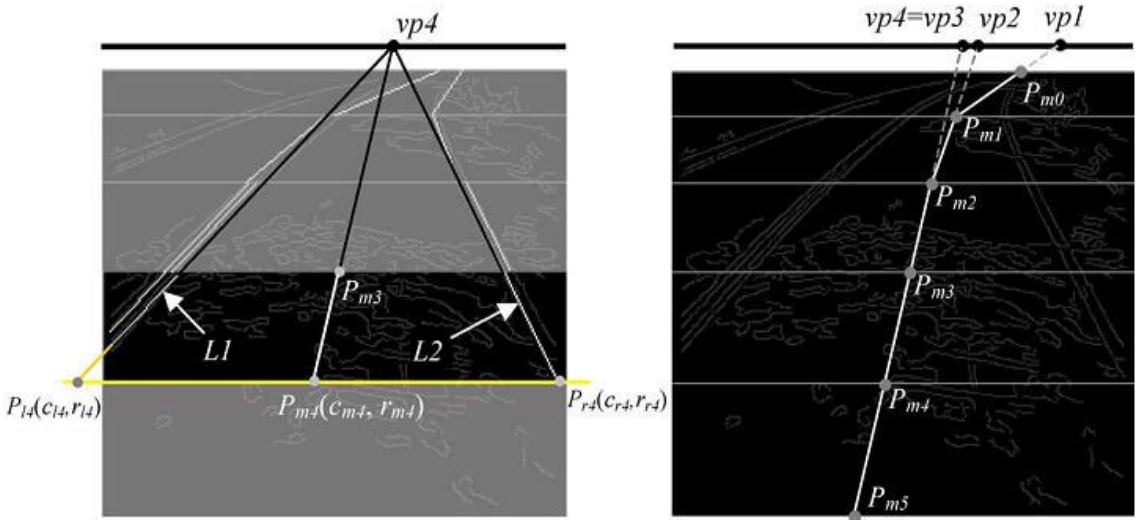


Figure II.4 : L'estimation des points de contrôle.

II.2.2.2. Mise à jour du modèle:

Consiste à modifier les points de contrôle afin de poursuivre la route à travers la séquence d'images, pour cela l'algorithme utilise la méthode MMSE (Minimum Mean Square Error) qui consiste à minimiser les forces externes appliquées sur le modèle de la route précédemment déterminé dans l'image en cours de traitement.

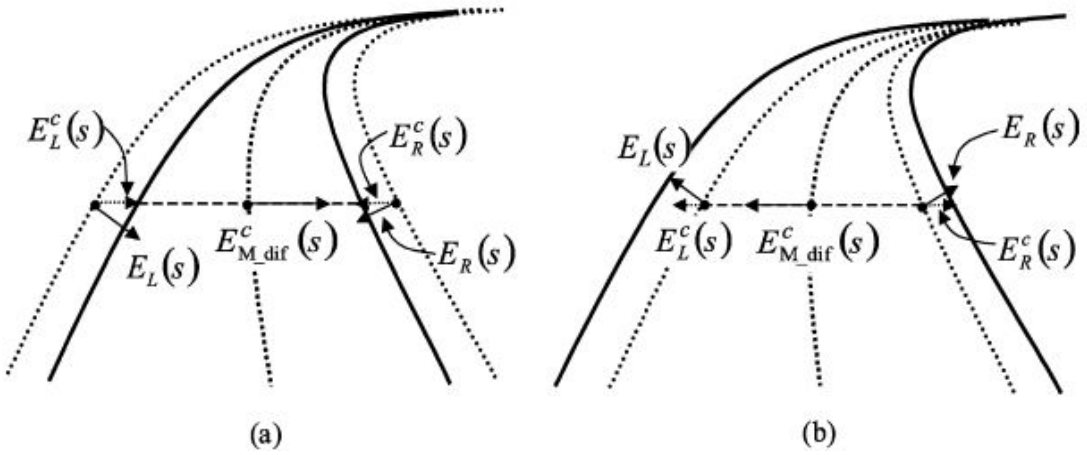


Figure II.5 : (a) le modèle de la route estimé est à l'intérieur de route réel, (b) le modèle de la route estimé est à l'extérieur de la route.

Figure II.6 : Résultat de détection des lignes, résultat final de l'algorithme.

II.2.3.Détection de plusieurs voies de circulation (Multi-Lane detection) [9]:

Cet algorithme permet de détecter trois voies de circulation sur les autoroutes (figure VII.3.3).Le principe de cet algorithme est basé sur trois principales phases : la détection des lignes de la route (trois lignes au minimum), estimation de trois lignes (génération d'hypothèses, *hypothesis generation HG*) à partir des trois lignes détectées lors de la phase précédente, puis la vérification des lignes proposées.

- Détection des lignes : à l'aide de l'algorithme de RANSAC (RANdom SAmple Consensus) Qui extrait les lignes présentes dans l'image (au moins trois lignes).
- Estimation des lignes adjacents par hypothèse (**HG**): on suppose que toutes les lignes de la route sont parallèles et ont le même largeur, après, on transforme l'image en vue

d'oiseaux à l'aide d'une matrice homographie dynamique, et enfin on déduit les lignes adjacentes par la méthode de cross-ratio [19].

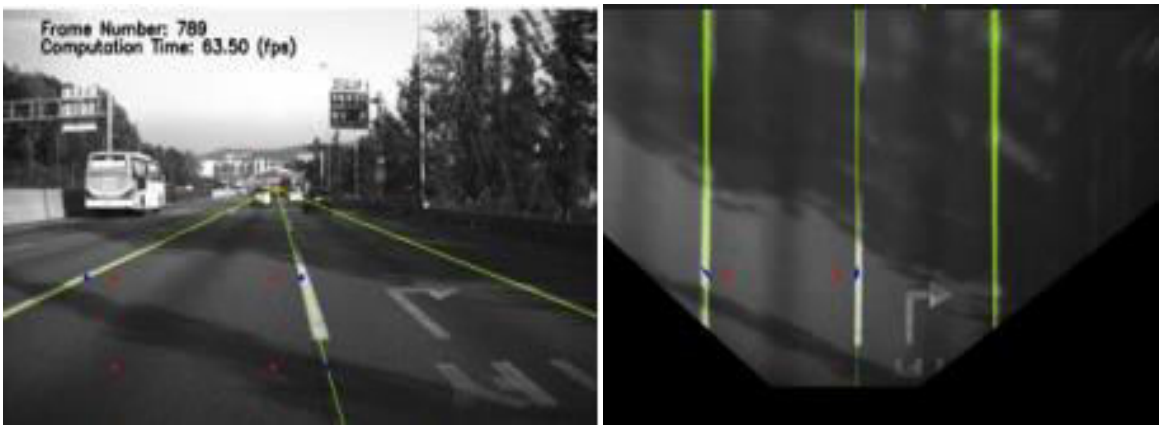


Figure II.7 : Image normale, Image en vue d'oiseaux.

- Vérification des lignes estimées (*hypothesis verification* ***HV***) : se fait à l'aide de l'algorithme de MAGSS (MAximum Gradient Square Sum) [9].

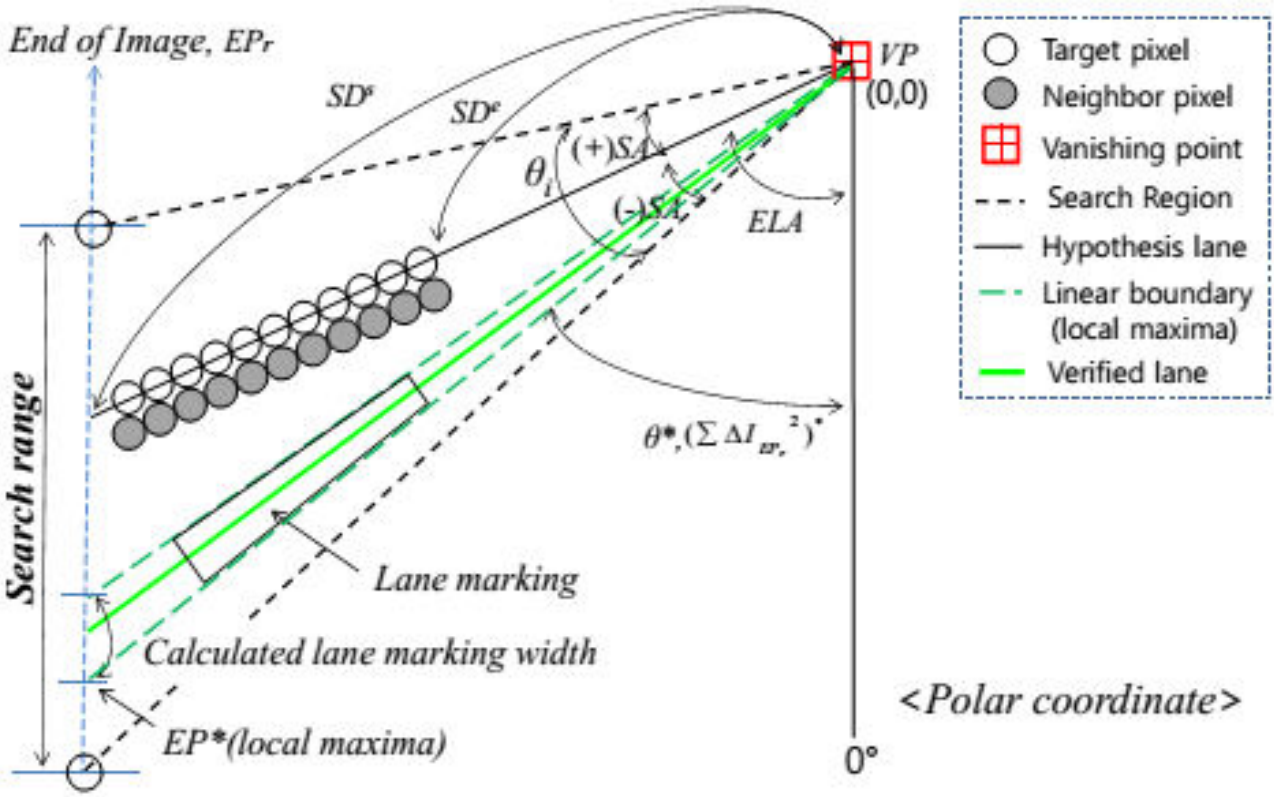


Figure II.8: Vérification des lignes de l'hypothèse par l'algorithme de MAGSS.

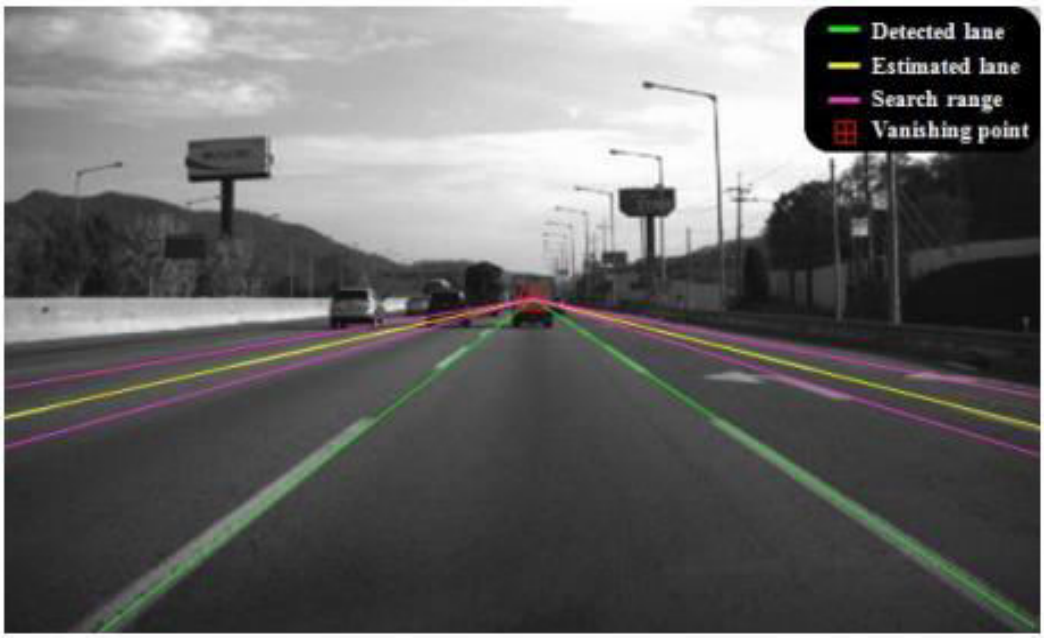


Figure II.9 : Résultat obtenu par l'algorithme de Multi-Lane detection.

II.2.4. Algorithme à base du modèle linéaire [3]:

Cet algorithme permet la détection des voies droites uniquement. Chaque bord étant délimité par une ligne droite de chaque coté. Donc pour les deux bords de la voie il va y'avoir 4 lignes droites (figure II.10) : LL (ligne de gauche pour le bord gauche), LR (ligne de droite pour le bord gauche), RL (ligne de gauche pour le bord droit), et RR (ligne de droite pour le bord droit). Chacune de ces droites peut être décrite par l'équation suivante :

$$(y_i - y_0)cos(\alpha_i) = (x_i - x_0)sin(\alpha_i) \qquad (6)$$

Où (x_i, y_i): Un point contour appartenant à la droite (LL, LR, RL, ou RR).

(x_0, y_0): Le point de poursuite (point ou les deux bords se croisent dans l'image).

α_i : La pente de la droite.

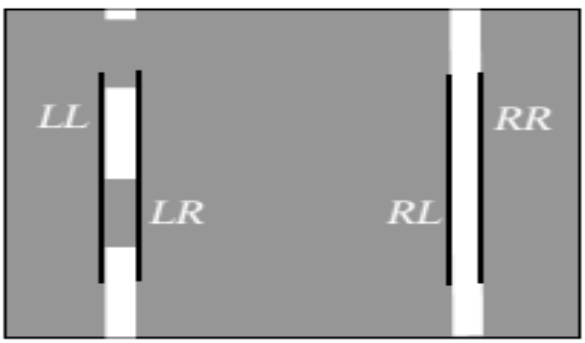


Figure II.10 : Limites des bords de la voie.

Le fonctionnement de cet algorithme se passe par :

- le transformée de Hough 2D : pour déterminer le point de poursuite.
- filtre de canny : détection des points de contours.
- le transformée de Hough 2D : déterminer les droites LL, LR, RL, et RR où chaque point de contour détecté vote pour une pente de droite donnée selon l'équation suivante :

$$tan\ (\alpha_i) = (y_i - y_0) / (x_i - x_0) \qquad (7)$$

Cela permettra d'avoir 4 pics dans l'espace de Hough qu'il faudra associer aux lignes LL, LR, RL, et RR. Cette association est possible après avoir définie l'angle de partition β qui représente la pente de la droite reliant le point de poursuite et le centre de la première ligne à partir du bas de l'image ainsi que les relations suivantes :

$$LL= \{\alpha_i | (\alpha_i \leq \beta+d\beta) \wedge (PerpGrad_i > 0)\}, \qquad (8)$$

$$LR= \{\alpha_i \mid (\alpha_i \leq \beta + d\beta) \wedge (PerpGrad_i < 0)\}, \qquad (9)$$

$$RL= \{\alpha_i \mid (\alpha_i \geq \beta - d\beta) \wedge (PerpGrad_i > 0)\}, \qquad (10)$$

$$RR= \{\alpha_i \mid (\alpha_i \geq \beta - d\beta) \wedge (PerpGrad_i < 0)\} \qquad (11)$$

Avec $d\beta$ =1° *et* $PerpGrad_i$ indique l'orientation du gradient par rapport à la droite ayant la pente α_i. Le résultat final est représenté dans la figure suivante :

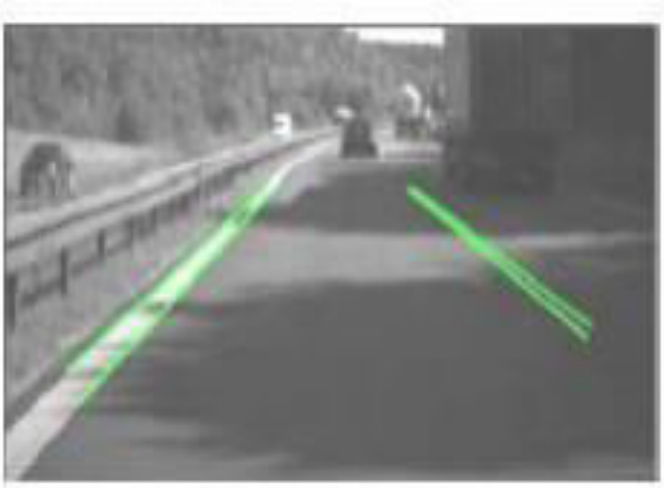


Figure II.11 : Les lignes LL, LR, RL, et RR détectées.

II.2.5. Algorithme à base de modèle linéaire-parabolique[3] :

Cet algorithme est basé sur la division horizontale de l'image de la scène routière en deux parties, la partie linéaire représente la zone proche du véhicule où les bords de la voie de roulement sont toujours droits, et la partie parabolique représente la zone plus éloignée du véhicule (figure II.12) où les bords de la voie peuvent être paraboliques.

Figure II.12 : La zone proche et la zone éloigner.

La détection des bords se fait à l'aide de la fonction de distribution de contour EDF (*Edge Distribution Function*) afin de représenter l'histogramme de l'amplitude de gradient selon l'orientation, l'angle θ doit être entre -90 et 90 (figure II.12), et transformée de Hough pondérée pour localiser les lignes(figure II.13).

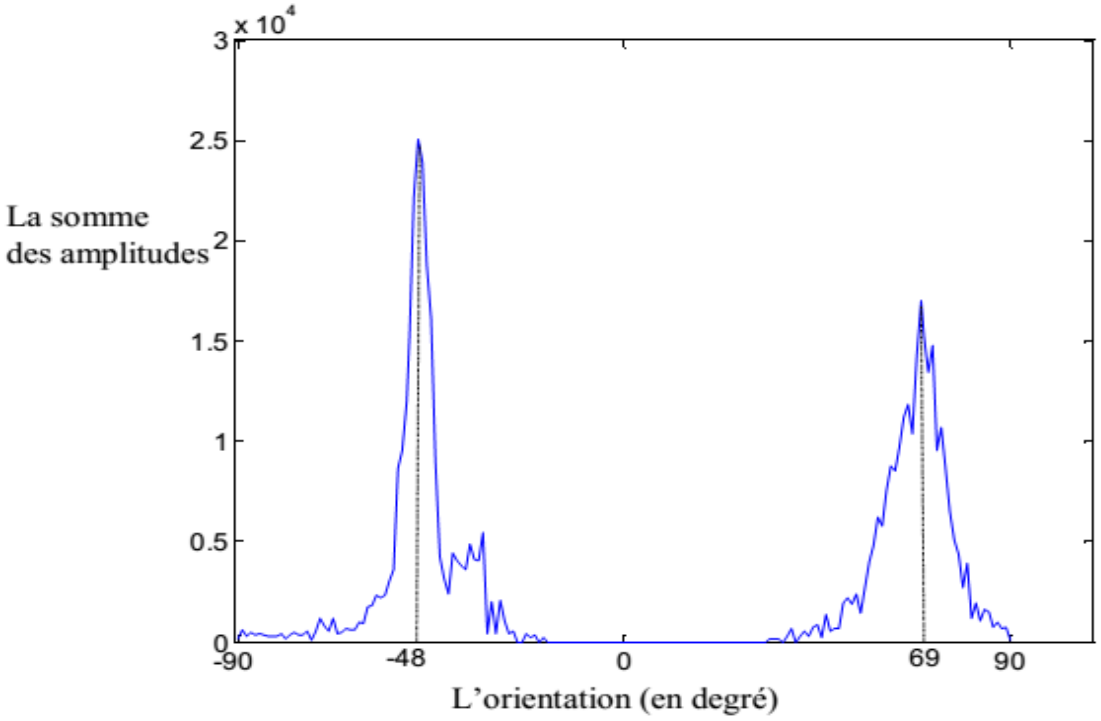


Figure II.13 : La fonction EDF.

Le pic (θ= -48°) se trouvant sur la portion gauche de l'histogramme correspond à l'orientation du bord droit (la ligne qui limite la voie de roulement par la droite).
θ Le pic (θ= 69°) se trouvant sur la portion droite de l'histogramme correspond à l'orientation du bord gauche (la ligne qui limite la voie de roulement par la gauche).

Si α correspond à l'orientation du bord qu'on désire détecter (droit ou gauche), alors on peut définir une fonction g :

$$g(x,y) = \begin{cases} | \Delta I(x,y) |, \text{Si } | \theta(x,y) - \alpha | < T\alpha \\ \\ 0 \text{ sinon} \end{cases} \qquad (12)$$

Où T_α est un seuil d'erreur.

La localisation de la ligne serait possible par l'utilisation de la transformée de Hough où C(p ,θ) représente le nombre de contour satisfaisant l'équation :

$$p = x.\cos\theta + y.\sin\theta \qquad (13)$$

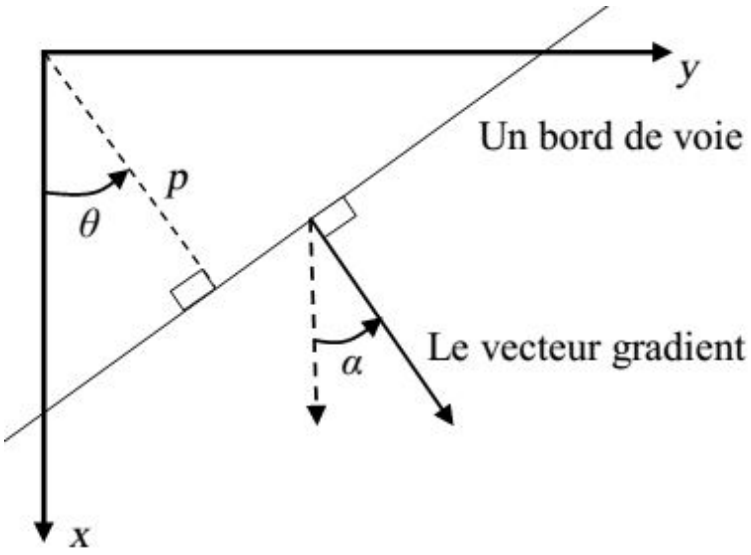


Figure II.14 : Transformer de Hough.

Dans le but d'assurer une bonne détection, cet algorithme utilise la modélisation linéaire des bords (figure II.15) et l'ajoute une zone d'intérêt appelée LBROI (*Lane Boundary Region Of Interest*), autour de chaque bord (figure II .16).

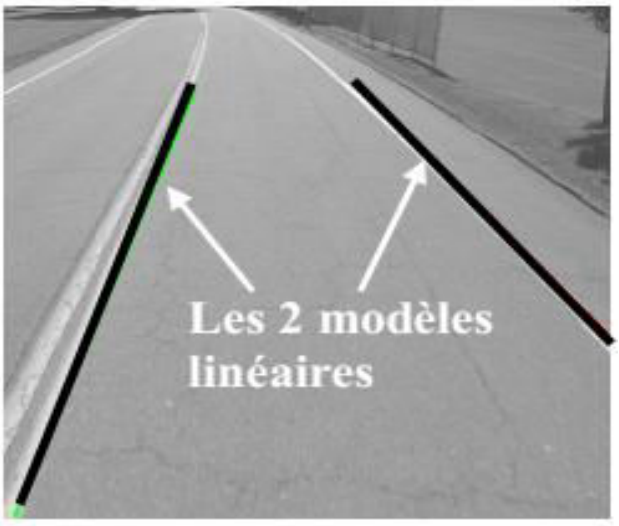


Figure II.15 : les deux modèles obtenus.

Figure II.16 : La zone LBROI.

II.3. Conclusion :

Dans ce chapitre nous avons présenté quelques méthodes déjà proposées, et proposées pour la détection de la voie de roulement par vision, Ces algorithmes sont toujours en développement car les techniques mathématique et informatique dans ce domaine évoluent tout le temps.

Dans le chapitre suivant nous discutons notre implémentation de la méthode IPM pour la détection de la voie de roulement.

CHAPITRE III :

Conception et mise en œuvre.

III.1. Introduction :

Dans ce chapitre, nous allons présenter la partie mise en œuvre. On va décrire l'algorithme adoptée, son architecture générale, les étapes de développement employées pour réaliser notre approche.

III.2. La méthodologie de l'algorithme :

Un tel système intelligent de détection de la voie de roulement, exige un algorithme qui répond à des conditions de la bonne performance, dans ce but on va discuter notre approche agréé, au premier, nous parlons l'acquisition de l'image, image niveau de gris, zone d'intérêt, puis nous détaillons la méthode d'IPM (Inverse Perspective Mapping) pour obtenir une vue d'en haut(vue d'oiseau), la binarisation, détection des contours, puis l'application de transformée de hough, et enfin l'élimination des lignes indésirables selon des critères prédéfinis .

III.3. Architecture générale de l'algorithme :

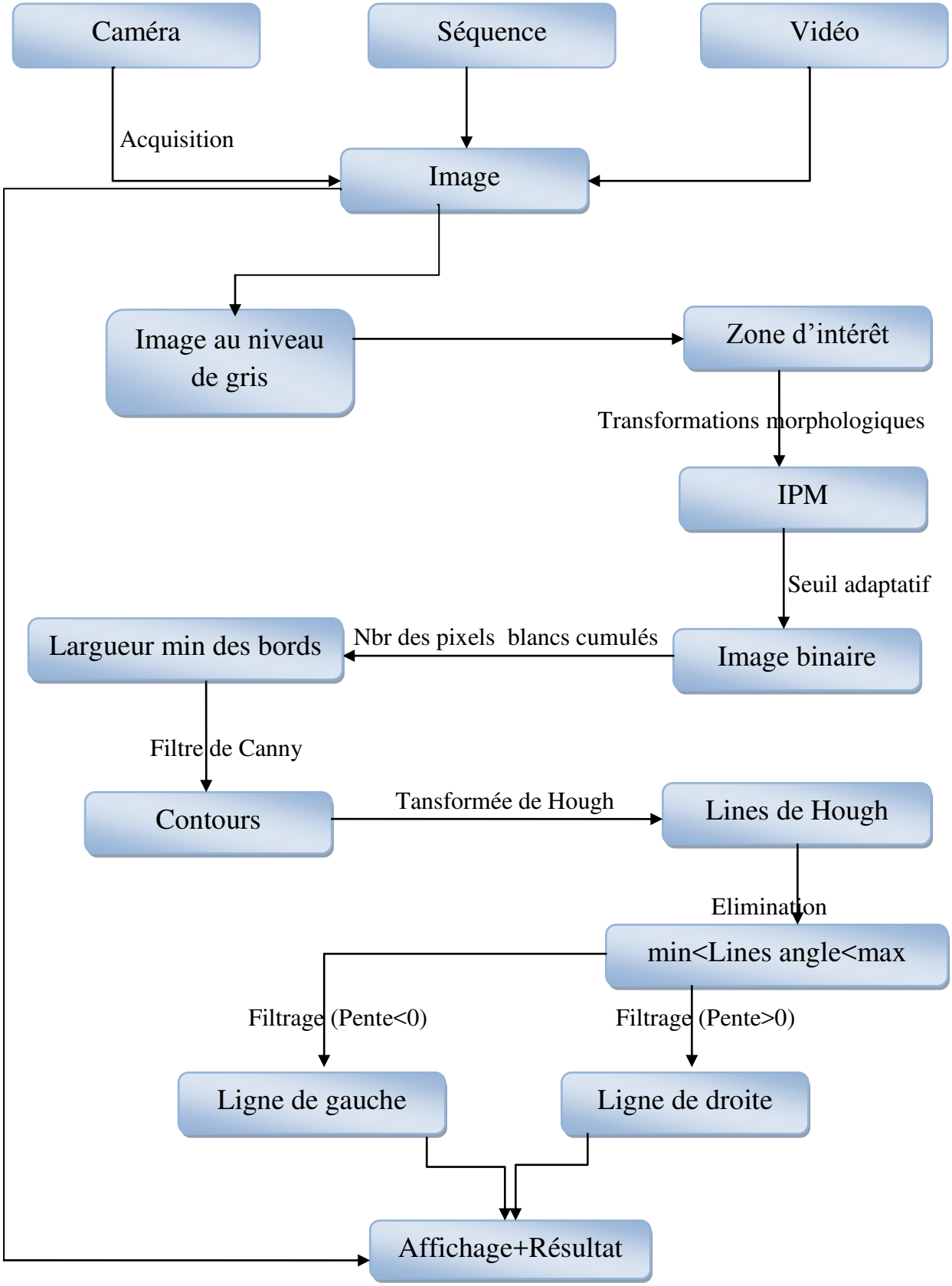


Figure III.1: Architecture générale de l'algorithme.

III.4. Acquisition :

La capture est la première étape dans le processus. Il faut réussir à enregistrer la scène routière sans trop de bruit afin de faciliter la détection de la voie de roulement. Pour cela on peut fixer la caméra sur le rétroviseur. Le codage consiste en l'acquisition d'image et sa digitalisation, ce qui donne lieu à une représentation bidimensionnelle de la route. L'image dans cette étape est dans un état brut ce qui engendre un risque de bruit qui peut dégrader les performances du système et donne lieu à une représentation 2D (la matrice des niveaux de gris) pour un objet 3D (la voie) **[2]**.

III.5. Image niveau de gris :

Dans cette étape, on transforme l'image couleur capturée en une image niveau de gris car l'information couleur n'est pas pertinente pour la détection de la voie **[11]**.

III.6. Image à demi :

Cherchons la zone d'intérêt dans l'image qui contient les bords de la voie, Dans les meilleures considérations cette zone est La moitié inférieure de l'image capturée **[12].**

Figure III.2 : La zone d'intérêt.

III.7. IPM (Inverse Perspective Mapping) [9]:

Avec IPM (Inverse Perspective Mapping) on peut transformer une image de scène en une image vue d'oiseau, Pour appliquer l'IPM on a besoin de Paramètres intrinsèques et extrinsèques de la caméra (distance focale, haut par rapport au sol, angle de vue, etc…).

Durant notre projet, nous avons utilisé une séquence d'images dont nous ne disposons pas des paramètres de la caméra. Ces derniers sont nécessaires pour le calcul de la transformé IMP. Par conséquent, il a fallu calculer la transformé IMP à partir des effets de perceptions.

Pour estimer la matrice IPM on doit sélectionner quarte points source dans l'image de la scène, là on perçoit les effets de perspectives, et quatre autre points destinatairesdans l'image vue d'oiseau où ces effets seront supprimés.

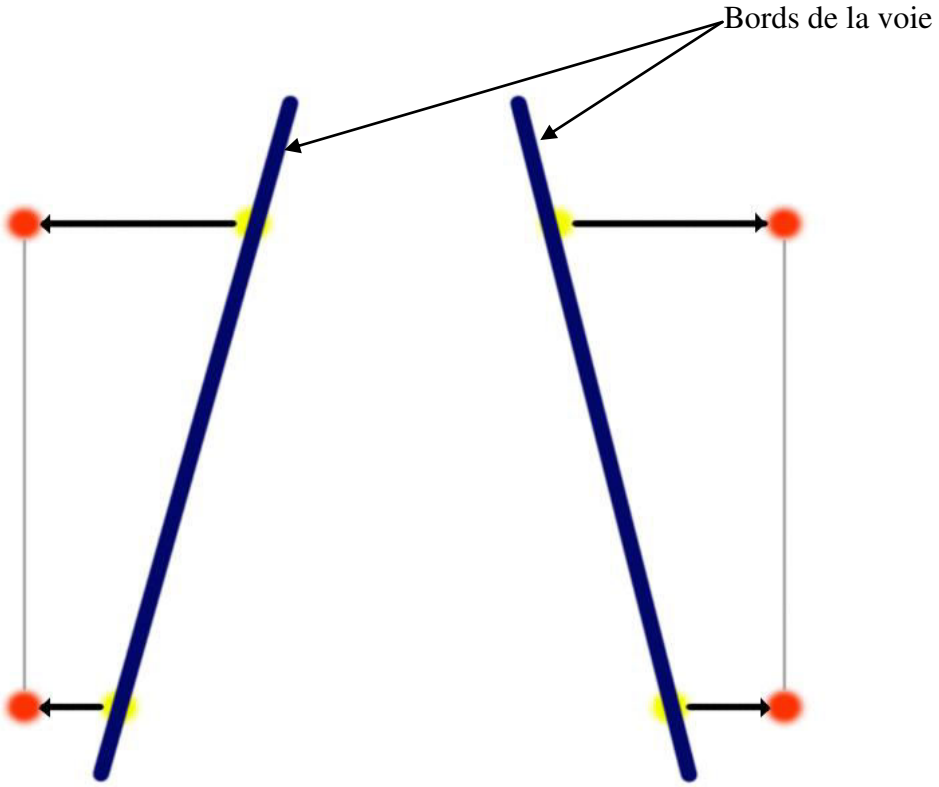


Figure III.3 : Points sources (en jaune) et points destinataires (en rouge).

Les points sources : Sont sélectionné manuellement sur les bords de la voie, les points jaunes de la figure.

Les points destinataires : Sont générer automatiquement (les points rouges qui de la figure).

Matrice homographie : se calcule par l'équation suivante :

$$X = H * x \qquad (14)$$

Où X, x et H représentent le vecteur des cordonnées des points dans l'image vue d'oiseau, vecteur des cordonnées des points dans l'image de scène, et la matrice homographie respectivement. L'équation peut être redéfinit comme suit :

$$\begin{bmatrix} Xt \\ Yt \\ t \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (15)$$

Avec $t = gx + hy + 1$.

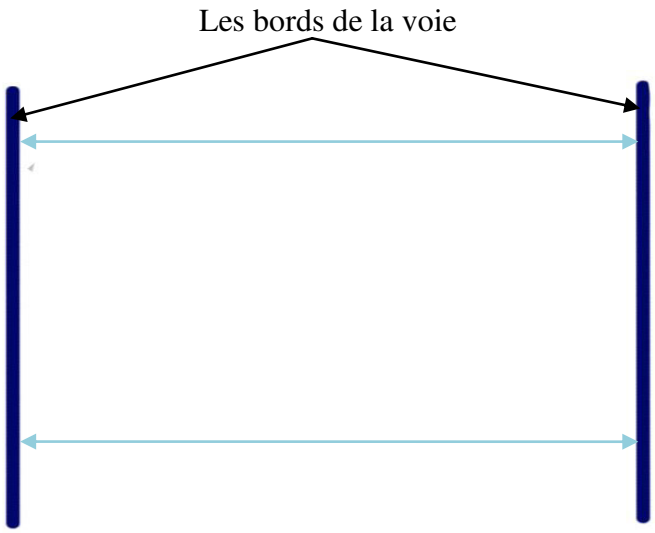


Figure III.4 : Les bords de la voie après la transformation IPM.

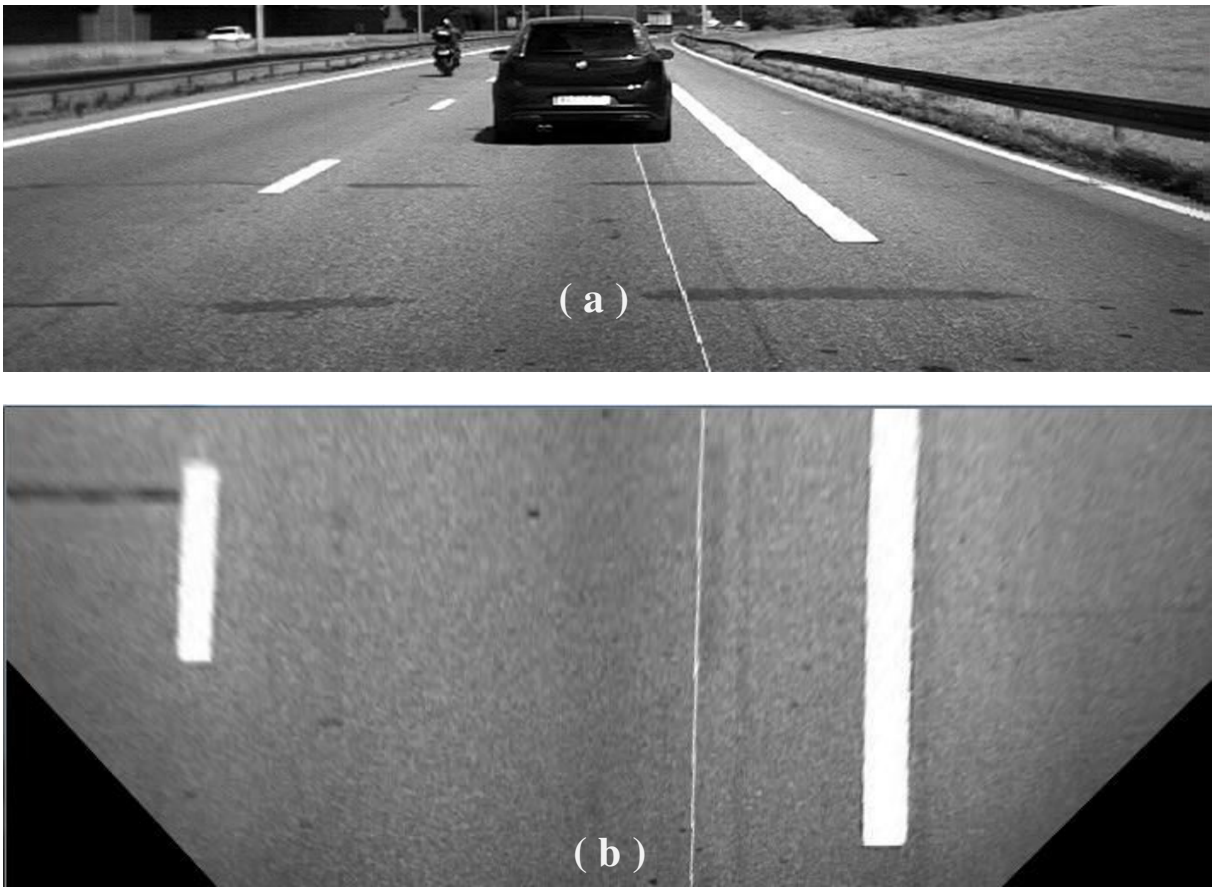


Figure III.5 : Exemple réel de résultat après application de la transformée IPM, (a) image de la scène routière, (b) résultat d'IPM (vue d'oiseaux).

III.8. Binarisation :

La binarisation consiste à transformer une image en niveaux de gris (matrice de valeurs comprises entre 0et 255 inclus) en une image binaire (matrice de 0 ou 1). De nombreuses méthodes sont proposées pour obtenir une image binaire (seuillage adaptatif, méthode d'Otsu,... etc), Dans notre cas, nous avons opté pour un seuillage adaptatif **[13]**.

Seuillage adaptatif :

Cette méthode est une approche locale : pour déterminer la valeur binaire d'un pixel donné, la méthode compare la valeur du pixel à la valeur moyenne des pixels dans une zone de taille prédéfinie, centrée sur le pixel. Ainsi, si le pixel est plus clair que son environnement, il lui sera affecté 1, et inversement. On peut également choisir d'ajouter une constante positive ou négative à la valeur moyenne **[13]**.

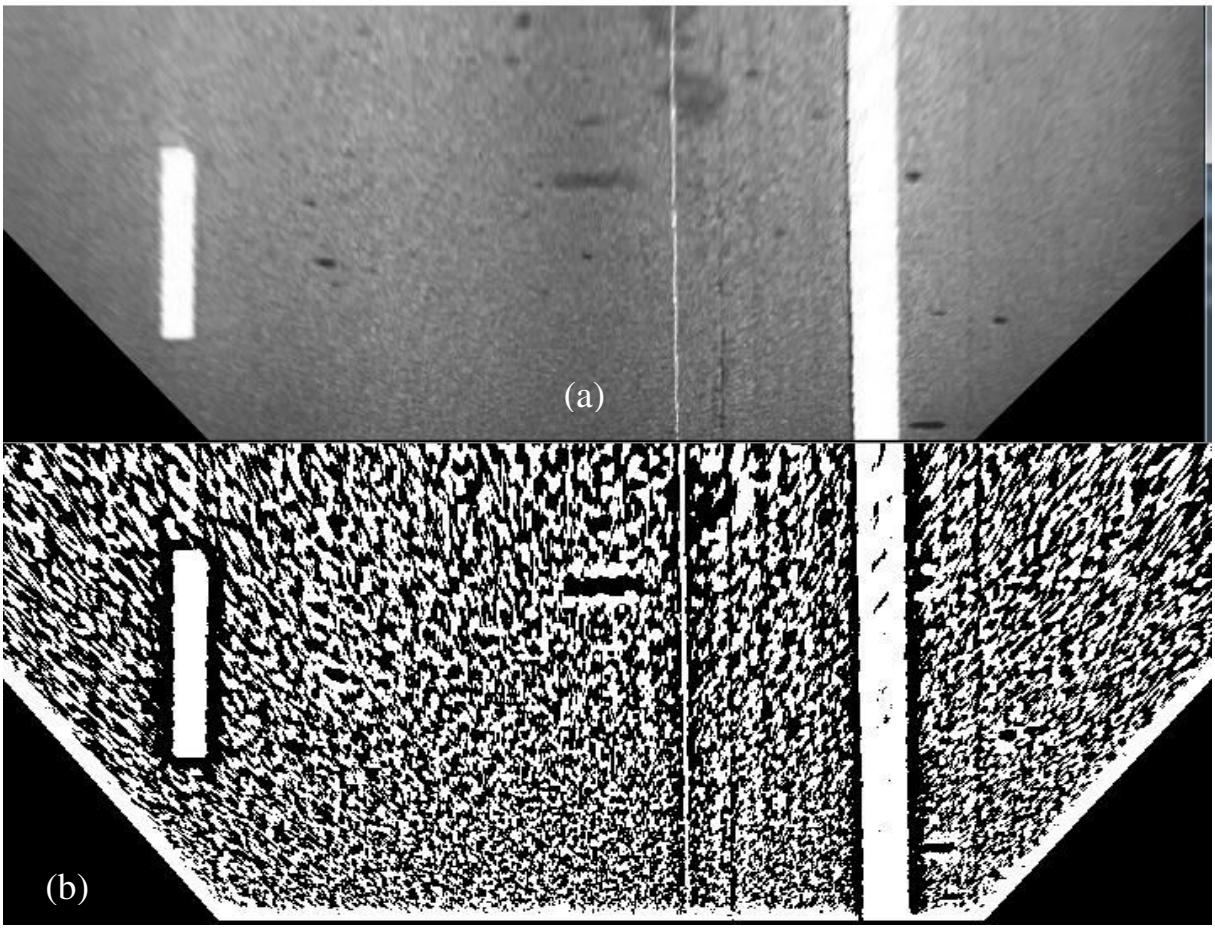


Figure III.6 : Binarisation par la méthode de seuil adaptatif, (a) Image source, (b) L'image binaire correspond à l'mage source.

III.9. Largeur minimale des bords :

Afin d'éliminer le bruit, on supprime tout ensemble de pixels blancs dont la largeur horizontale est inférieure à un seuil prédéfini, En effet, on estime que les pixels appartenant aux bords de la voie constituent des ensembles contigus de pixels ayant une certaines intensité par rapport au voisinage. La figure suivante illustre l'intérêt de cette étape où on constate clairement la suppression de plusieurs pixels ne représentant pas les bords.

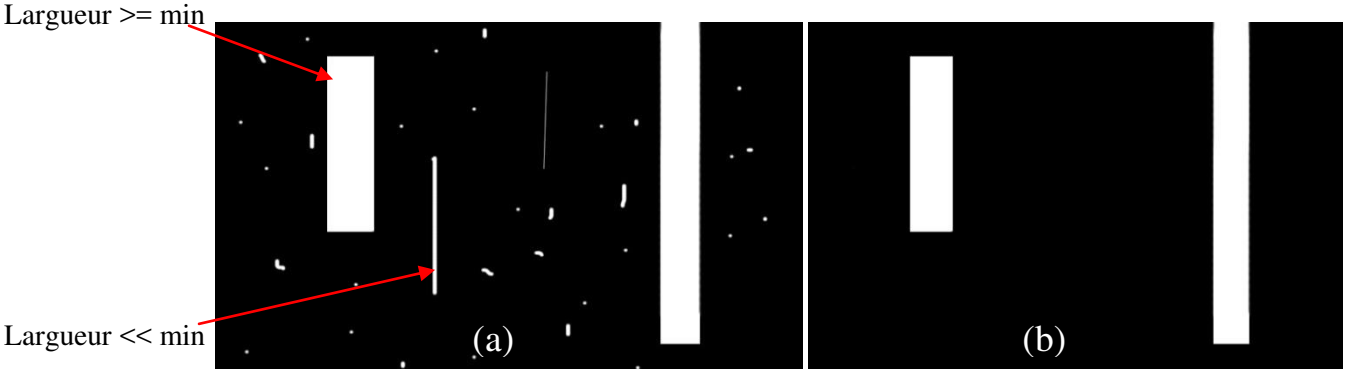


Figure III.7 : Elimination selon le largeur minimal, (a) Image binaire source, (b) L'image résultat d'élimination selon le largeur minimal.

III.10. Détection des contours :

Se fait par le filtre de canny (pour plus des détails sur ce filtre, retournez à l'élément I.4.1 de cette mémoire) **[6].**

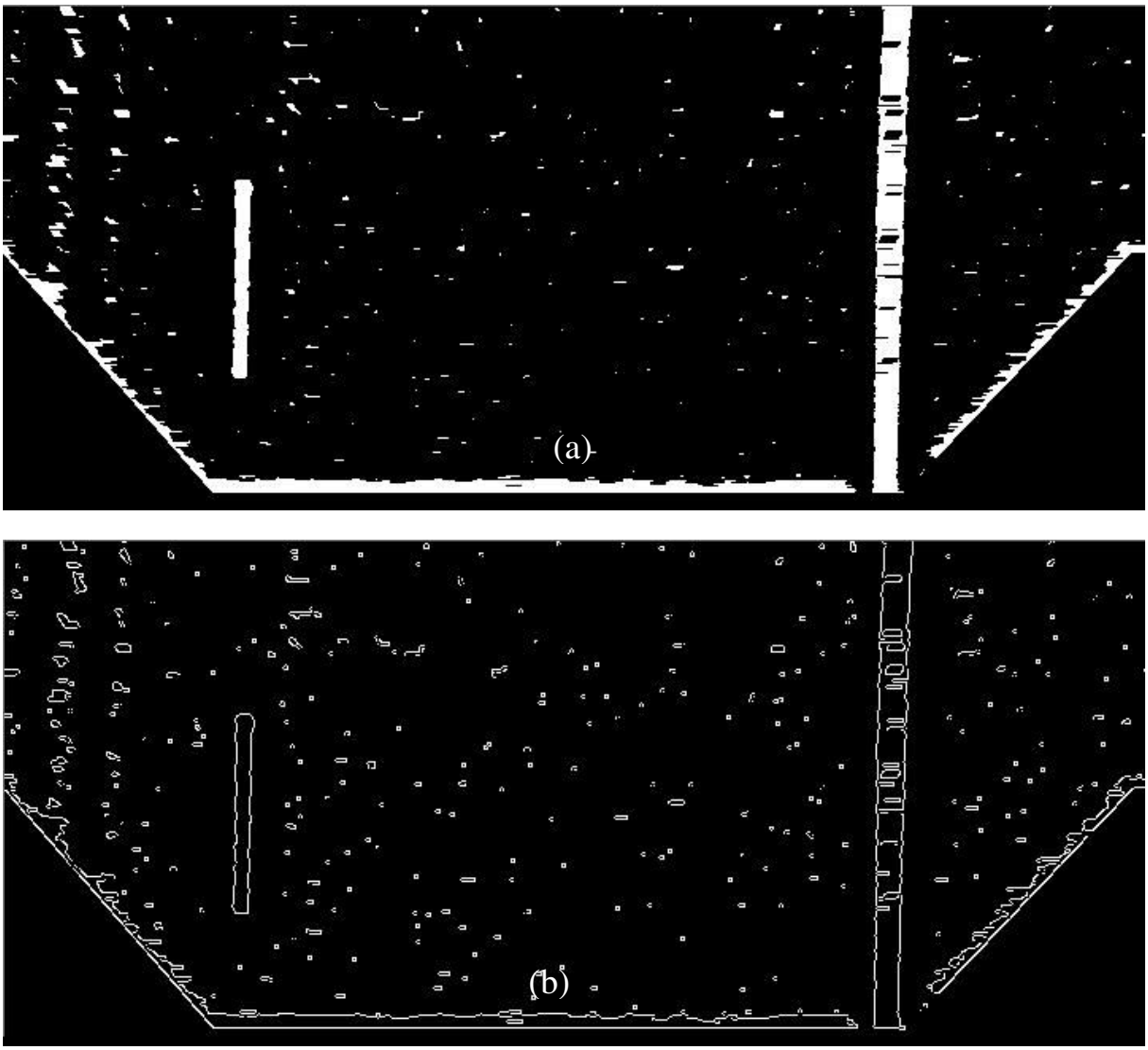


Figure III.8 : Détection des contours par le filtre de canny, (a) Image au nivaux de gris, (b) Les contours par le filtre de canny.

III.11. Transformée de Hough :

La transformée de hough est une technique générale pour identifier la location et l'orientation de certaines forme géométriques dans une image numérique, développé par Paul Hough en 1962 et présentée par IBM **[14]**.

Le principe de la transformée de Hough est qu'il existe un nombre infini de lignes qui passent par un point, dont la seule différence est l'orientation (l'angle). Le but de la transformée est de déterminer lesquelles de ces lignes sont les plus fréquentes dans l'image analysée.

On suppose tout d'abord que si des lignes ou des segments de droites sont présents dans une image, ils feront partie des contours présents dans l'image. On commence donc dans un premier temps par identifier tous les points de contours de cette image, par exemple à l'aide de techniques de mesures de gradients locaux entre les valeurs des pixels autour de chaque point de l'image. Les points de l'image présentant les gradients les plus élevés dans leur voisinage, soit globalement pour l'image (seuillage fixe), soit par rapport aux gradients généralement présents dans un voisinage plus large autour du point (seuillage dynamique), sont les plus susceptibles d'appartenir à des contours de cette image.

Chacun des points des contours ainsi identifiés (x, y) va alors permettre une projection dans un plan (le plan transformé) des coordonnées polaires de toutes les droites passant par ce point. Les équations des droites passant en chacun de ces points (x, y) sont représentées par l'équation "normalisée"

$$\rho = x * cos(\theta) + y * sin(\theta, \qquad (16)$$

où θ est l'angle de la droite et ρ la distance de la droite à l'origine (au lieu de y = a *x + b). Au point (x, y) du contour, on fait donc correspondre une courbe [θ, ρ], où θ prend toutes les valeurs possibles de 0 à 2π, ρ prenant la valeur ρ = x * cos(θ) + y * sin(θ). Une fois appliquée à tous les points des contours, et, pour chacun, à tous les angles θ possibles, les points (θ,ρ) de l'espace transformé les plus souvent adressés sont les coordonnées des droites ou des segments de droite les plus représentés dans l'image de départ.

En pratique, l'espace transformé de Hough sera représenté par une image, dont les abscisses seront les angles θ, dont les ordonnées les valeurs de ρ, et dont l'intensité au point quelconque (θ, ρ) est le nombre d'occurrences de (θ, ρ) provenant de l'image d'origine. Aucune hypothèse de continuité des droites ou segments de droite de l'image de départ n'est faite ici, ce qui rend la transformée robuste à l'absence de points : masquage partiel des droites, et au bruit dans l'image.

Les valeurs de θ peuvent être discrétisées par exemple en degrés (selon la précision souhaitée), et les valeurs de ρ en pixels représentant la distance (toujours inférieure en valeur absolue au diamètre de l'image de départ), la fréquence étant bornée par le nombre de points sélectionnés dans les contours de l'image de départ.

Dans la transformée de Hough, dite aussi transformée standard de Hough ou SHT, chaque ligne est un vecteur de coordonnées paramétriques :

- θ: l'angle
- ρ: la norme du vecteur (la longueur du segment perpendiculaire à la droite d'angle θ et passant par l'origine).

En transformant toutes les lignes possibles qui passent par un point, c'est-à-dire en calculant la valeur de ρ pour chaque θ, on obtient une sinusoïde unique appelée espace de Hough. Si les courbes associées à deux points se coupent, l'endroit où elles se coupent dans l'espace de Hough correspond aux paramètres d'une droite qui relie ces deux points **[15].**

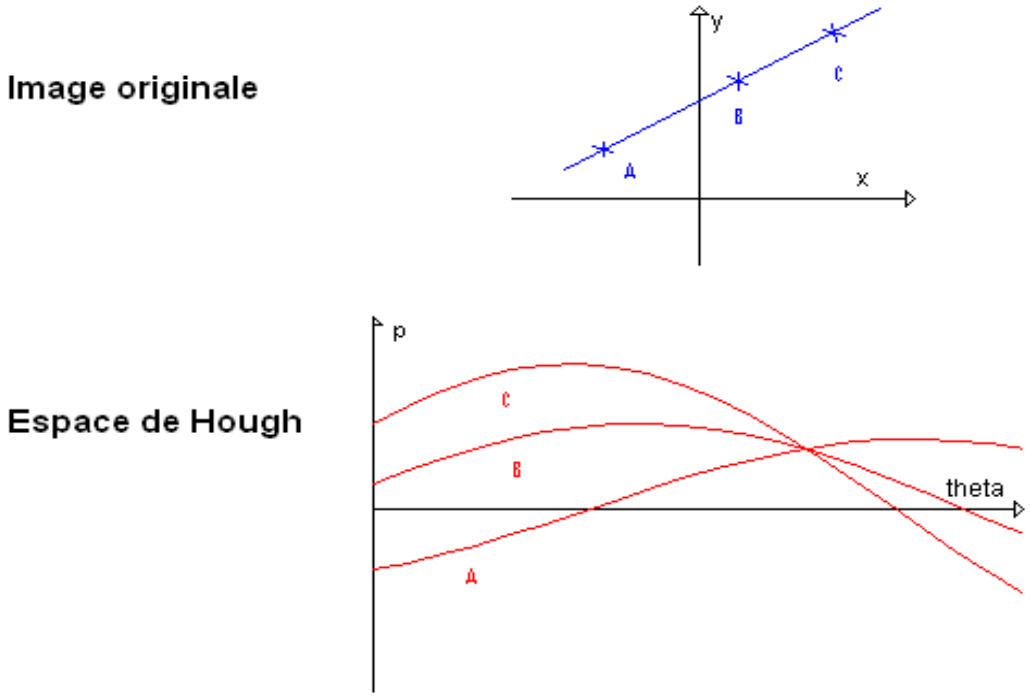


Figure III.9 : Espace de Hough.

L'image suivante représente les lignes de Hough détectées dans une image vue d'oiseau.

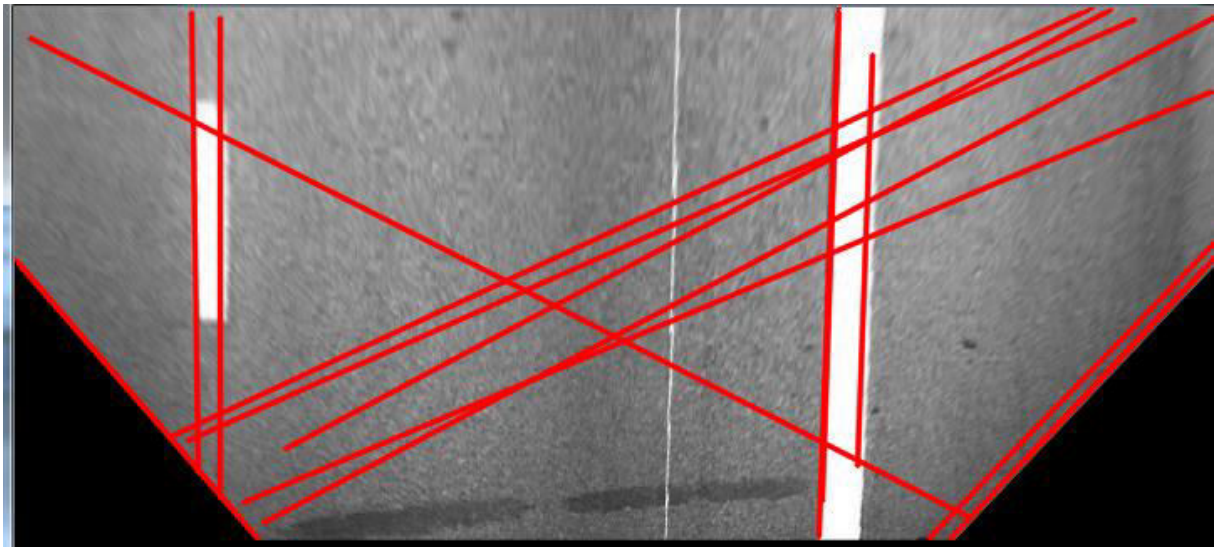


Figure III.10 : Lignes de transformée de Hough.

III.12. Elimination selon l'orientation :

Après avoir extraire les lignes candidates par la transformée de Hough, on prend quelques propriétaires afin de filtrer ces lignes et d'en éliminer les lignes indésirables c.-à-d. celles ne représentant pas les limites de la voie de roulement.

Les bords de la voie dans Une image vue d'oiseau se caractérisent par une orientation autour de 90 degré, Par conséquent, on exploite cette propriété a fin de filtrer les lignes détectées.

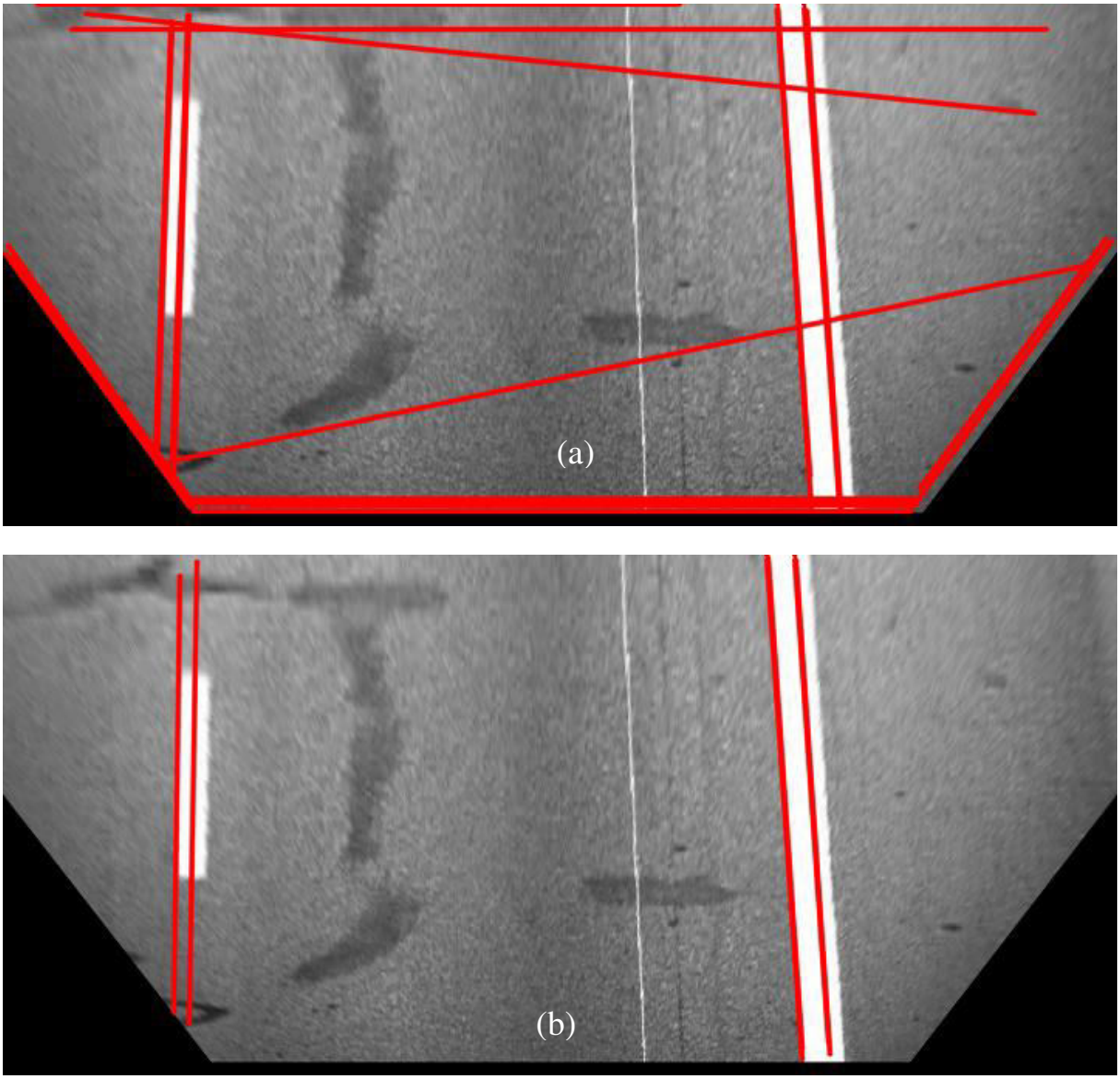


Figure III.11 : Elimination selon orientation (angle<110 & angle>70), (a) Les lignes de hough, (b) Les lignes ayant l'orientation des bords.

II.13. Point de fuite :

Point de fuite est le point de croisement des bords de la voie de roulement [9], pour notre étude ce point se calculé après la détection des bords de la voie, nous permettra d'estimer le déplacement latérale du véhicule sur la voie de roulement.

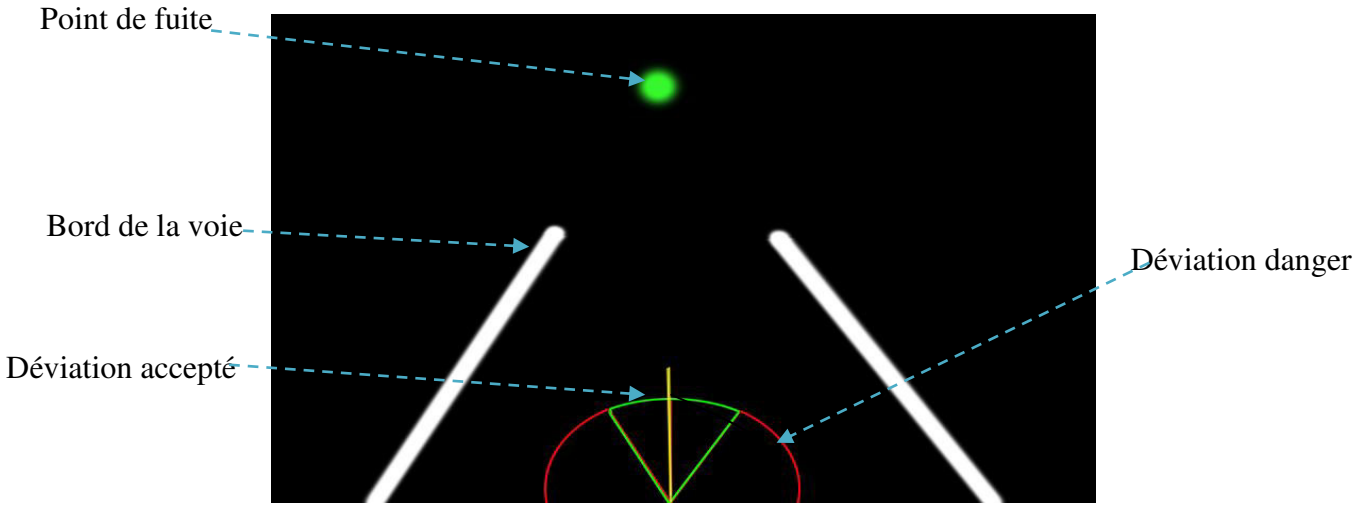


Figure III.12 : Point de fuite.

III.14 Conclusion :

Dans ce chapitre, nous avons donné une importance aux méthodes de traitement d'images nécessaires pour notre algorithme puisqu'elles seront notre sujet de travail, on a décrit l'architecture générale de l'algorithme adopté, la zone d'intérêt, nous avons détaillé bien la méthode d'IPM (Inverse Perspective Mapping) comme étant un outil efficace de transformation d'image de scène en une image vue d'en haut, les détails de l'opération de binarisation, puis on a vu la mise en œuvre de la transformée de Hough, et comment on peut choisir des lignes de bords possibles.

CHAPITRE IV :

Application

IV.1. Introduction :

Après avoir présenté un état de l'art des méthodes de détection de voie et détaillé notre approche basée sur la transformée IPM, il s'agit maintenant de mettre en œuvre les étapes explicitées dans les chapitres précédents pour obtenir une application de « détection de la voie de roulement». Ce chapitre porte sur la construction d'une application qui permet de reconnaitre des bords de la route en utilisant les différentes approches et technologies et de algorithmes de traitement d'image, en exploitant des outils qui ont permis d'accomplir ce dernier, par la suite, on présente les résultats correspondants à des exemples des tests effectués sur un séquence d'image données.

IV.2. Environnement matériel et logiciel:

Dans cette section, Nous présenterons l'environnement de travail pour réaliser notre système.

IV.2.1. Ressources utilisées

Les ressources physiques utilisées sont :

- Processeur Intel® Core™ i3-4210U d'une fréquence de 2.40 GHz.
- Une mémoire vive d'une capacité de 4 GO.
- Une carte graphique NVIDIA GeForce de 820 MB.

Et pour ce qui est côté logiciel (Soft) :

- Système d'exploitation: Windows 7 Professionnel.
- Microsoft Visual Studio 2012 Professional.
- Open source Computer Vision: OpenCV version: 2 .4.9.

IV.2.2. Langage de programmation :

Le langage de programmation choisi dans notre travail est le **C++** qui reste un des langages les plus utilisés dans le domaine de la reconnaissance des formes. Cela est dû au fait que le langage C possède un compilateur très performant permettant de générer un code très rapide et qu'il comporte des structures et des instructions de haut niveau.

Le **C++** est le descendant du langage C. Ces deux langages, bien que semblables au premier abord, sont néanmoins différents. Le **C++** propose de nouvelles fonctionnalités comme la **P**rogrammation **O**rientée **O**bjet (**POO**). Elles en font un langage très puissant qui permet de programmer avec une approche différente du langage C. Ainsi, un programme écrit en C en respectant la norme **ANSI** est portable (sans changements) sur n'importe quel système d'exploitation disposant d'un compilateur C : Windows, Unix, MAC OS, etc.

Le **C++** permet la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation générique et la POO **[16]**.

IV.2.3. Microsoft Visual Studio 2012 Professional:

Visual Studio offre un moyen pratique et facile de créer des applications informatiques pour les systèmes d'exploitation Microsoft Windows. Il utilise plusieurs langages de programmations C++, C#, F#, VB.net,…etc.

C++ de l'ordinateur comme sa syntaxe de base et de la logique de programmation, en adhérant à la norme ANSI avec beaucoup d'améliorations d'éléments personnalisés de ses bibliothèques de composants.

Tout d'abord C++ est un outil RAD (Rapid Application Development), c'est à dire tourné vers le développement rapide d'applications sous Windows. En un mot, Visual C++ permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événement système, etc.). Pour ce faire, Visual C++repose sur un ensemble très complet de composants visuels prêts à l'emploi. La quasi-totalité des contrôles de Windows (boutons, boîtes de saisies, listes déroulantes, menus et autres barres d'outils) y sont représentés, regroupés par famille.

Leurs caractéristiques sont éditables directement dans une fenêtre spéciale intitulée éditeur d'objets. L'autre volet de cette même fenêtre permet d'associer du code au contrôle sélectionné. Il est possible d'ajouter à l'environnement de base des composants fournis par des sociétés tierces et même d'en créer soit même **[17]**.

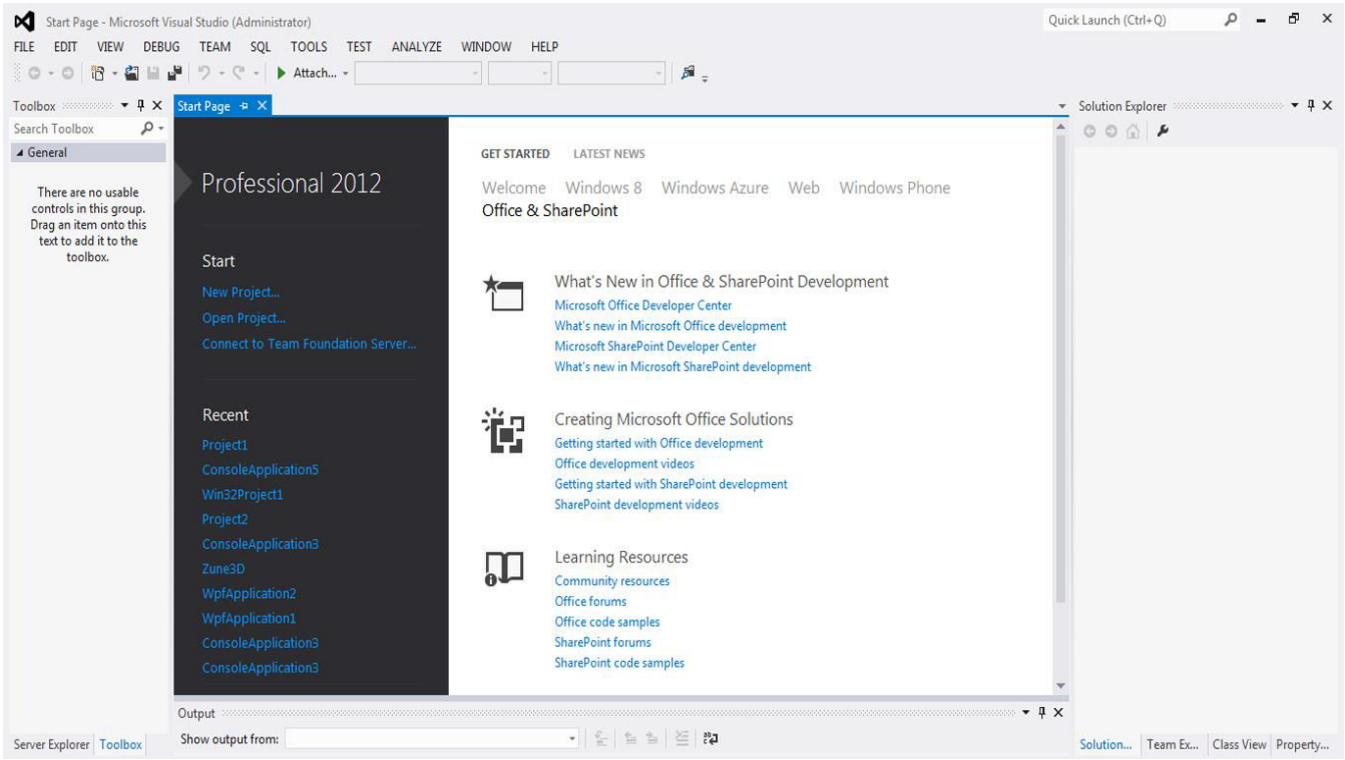


Figure IV.1: Interface Visual Studio 2012 Professional.

IV.2.4. OpenCV 2.4.9 :

IV.2.4.1 Définition:(Open source Computer **V**ision), est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage assure le support de cette bibliothèque depuis 2008 **[18].**

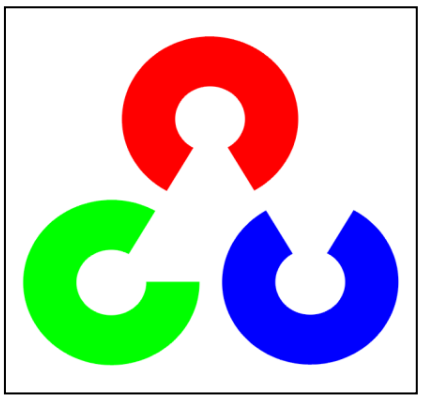


Figure IV.2 : Logo d'OpenCV.

IV.2.4.2 Configurations de l'installation d'OpenCV 2.4.9 avec Visual studio 2012 Professional :

1. Ajouter un variable de système (PATH) correspondante au dossier bin d'opencv.
2. Créer un nouveau projet c++.
3. Ouvrir les paramètres du projet.
4. Pour les systèmes 64bits : configuration manager ->add x64 platform
5. On choisit dans le haut de la fenêtre de propriétés : all configurations, puis

*C/C++ ->Additional Include Directories->ajoutons OpenCV include

dossiers.

*Linker ->Additional Librairy Directories -> ajoutons le dossier des librairies

d'opencv.

6. On choisit dans le haut : Debug, puis en passant à :

*Linker -> Additional Dependencies -> Add Dependences librairies,

on charge les noms des librairies .lib

7. Maintenant on choisit : Release,

* Linker -> Additional Dependencies -> Add Dependences librairies,

on charge les noms des librairies .lib

IV.3. L'interface d'accueil de l'application :

L'interface d'accueil s'affiche lors de double clicks sur l'exécutable du projet, dans cette fenêtre l'utilisateur doit choisi la source de données : WebCam, séquence d'images, ou vidéo.

Figure IV.3: L'interface d'accueil de l'application.

1 : Ouvrir la source en Webcam.

2 : Ouvrir la source en dossier des séquences ordré.

3 : Ouvrir la source à partie d'une vidéo sur la mémoire.

4 : Ouvrir l'aide.

IV.4. L'interface de la scène:

Après le choix de source de scène, une première image se déclenche demande de sélectionner quatre points dans un ordre de lettre Z permettent d'estimer les paramètres intrinsèques et extrinsèques de la caméra, la ligne rouge Délimite la zone d'intérêt représentant la partie inférieure de l'image, les Points sélectionnés doivent être dans cette zone d'intérêt doivent en bas de cette ligne :

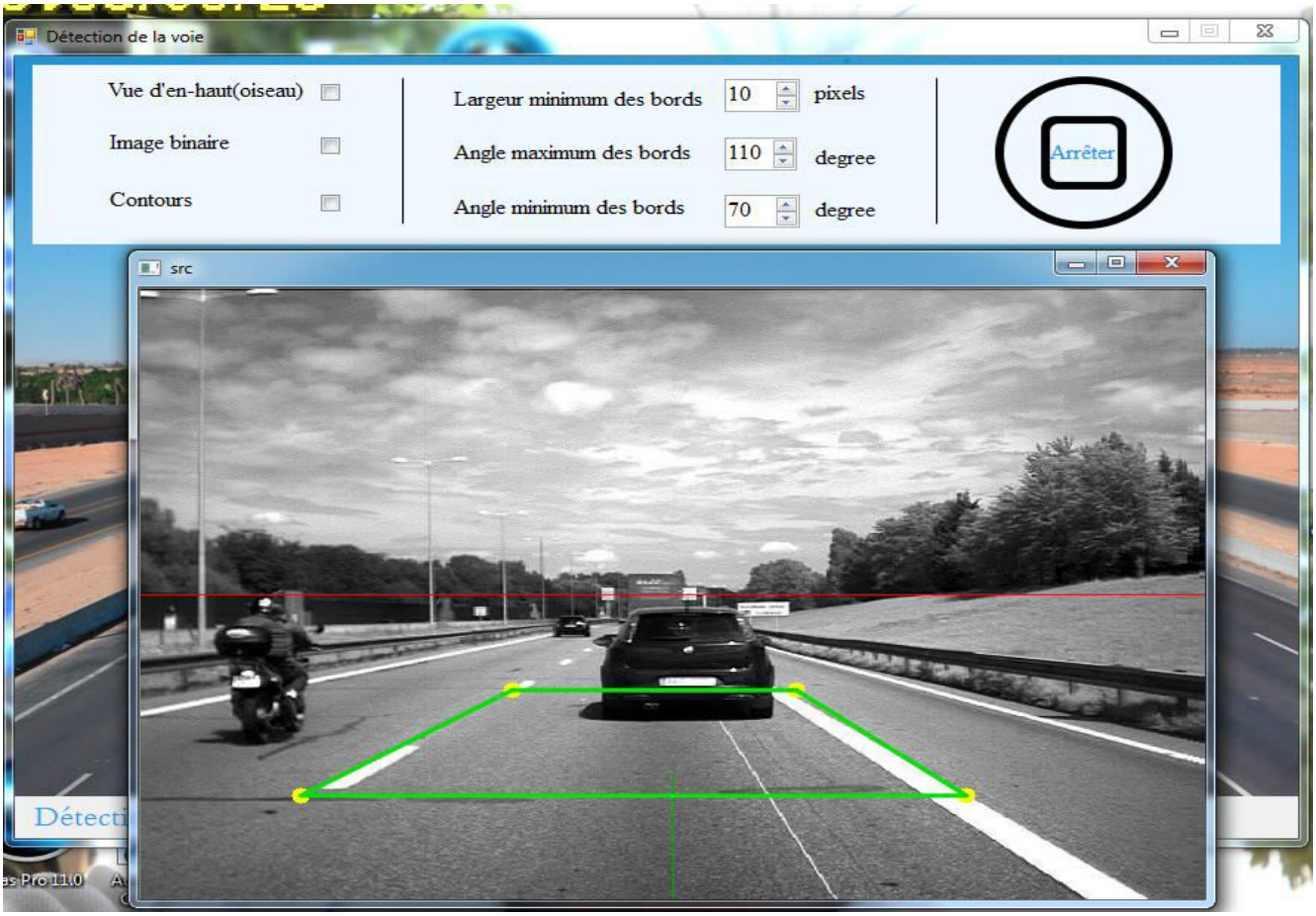


Figure IV.4 : L'interface de la scène.

Pour passer à l'étape suivante (détection) on cliquant sur un tel clé du clavier.

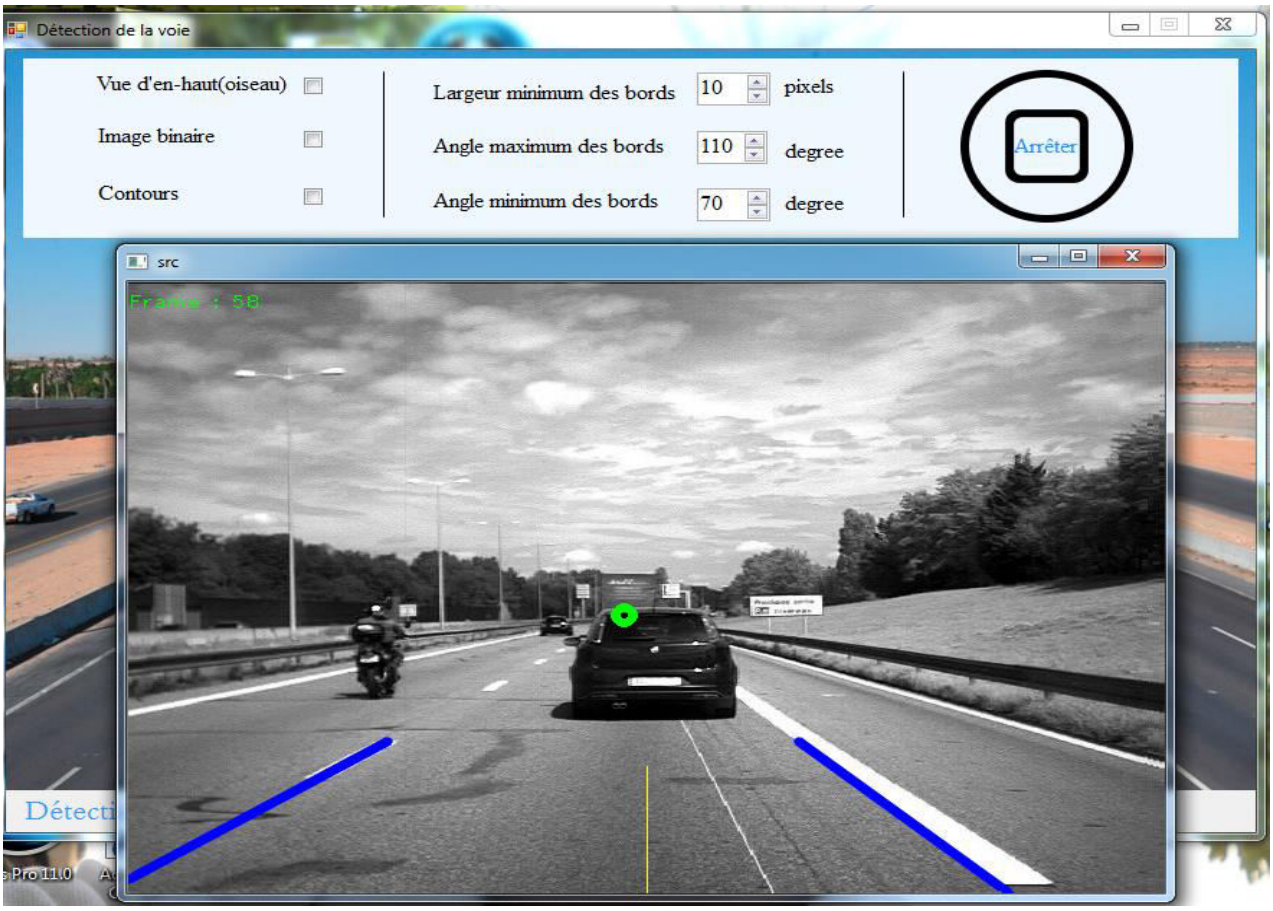


Figure IV.5 : L'interface de la détection.

Les lignes bleues : représentent les bords de la voie de roulement détectés.

Le point verte : le point de fuite (ou le point de croisement des lignes de la voie).

Arrêter : terminer la détection et retour à la page d'accueil.

Vue d'en haut (vue d'oiseaux) : afficher la fenêtre correspondante à la vue d'oiseau de la route.

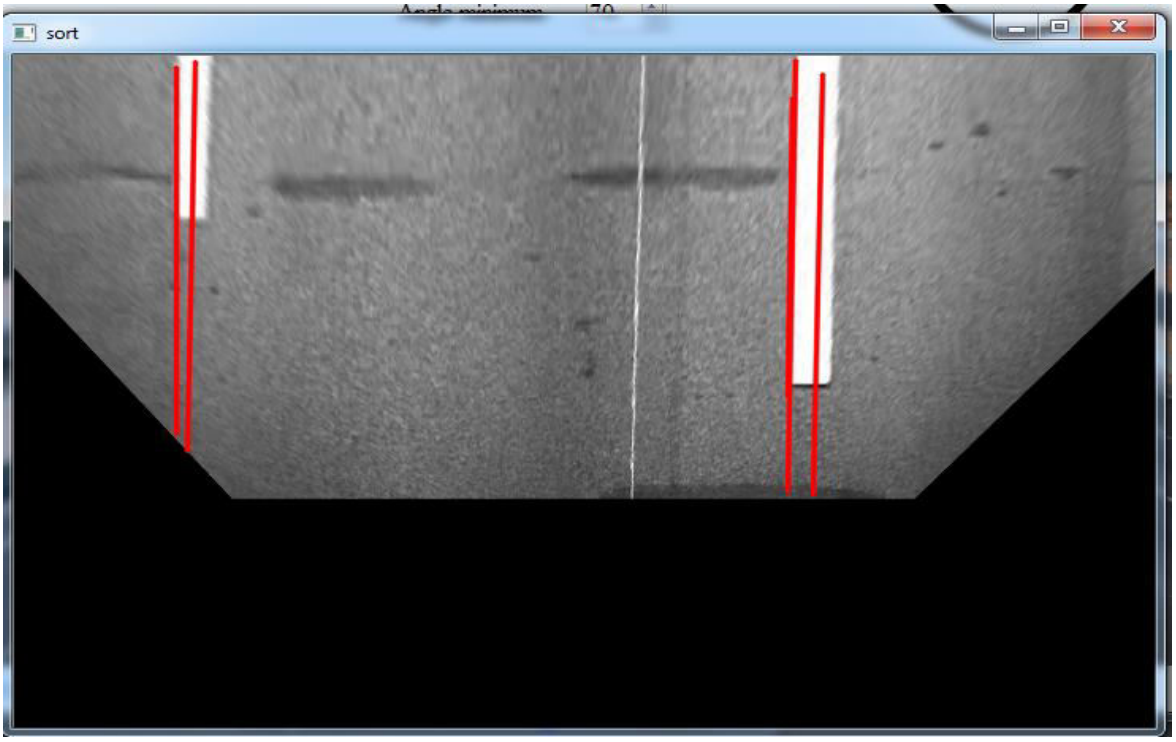
Image binaire : afficher la fenêtre correspondante à l'image binaire de l'image bird-eye.

Contours : afficher la fenêtre correspondante à l'application de filtre de canny.

Largeur minimum : option plus utile, permet de spécifier le largeur minimal des bords de la route pour améliorer bien la qualité de détection.

Angle maximum, Angle maximum : l'algorithme de l'application doit spécifier l'angle (maximum et minimum) des lignes de bords pour éliminer les lignes intrus.

IV.5. Fenêtre vue d'oiseau: s'affiche la vue d'en haut (vue d'oiseau), et les lignes détectées par la transformée de Hough :



:

Figure IV.6 : Fenêtre de vue d'oiseau avec lignes détectées par la de Hough.

IV.6. Fenêtre d'image binaire : s'affiche l'image binaire correspondante à la scène:

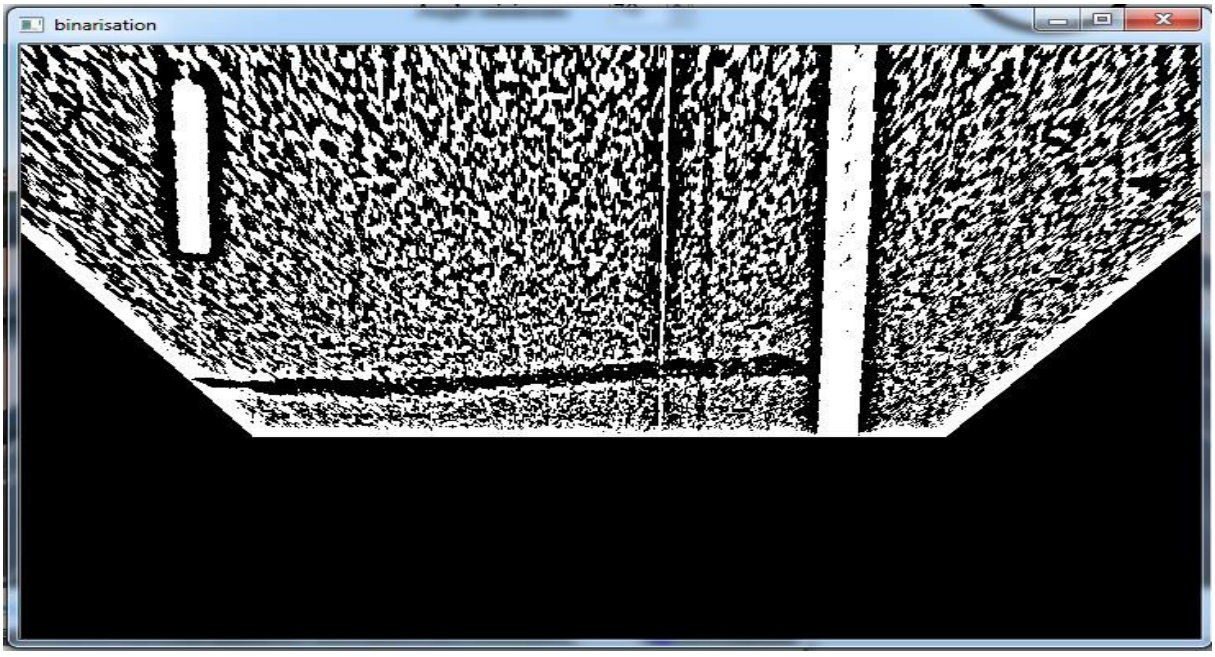


Figure IV.7 : Fenêtre d'image binaire.

IV.7. Fenêtre de contours :

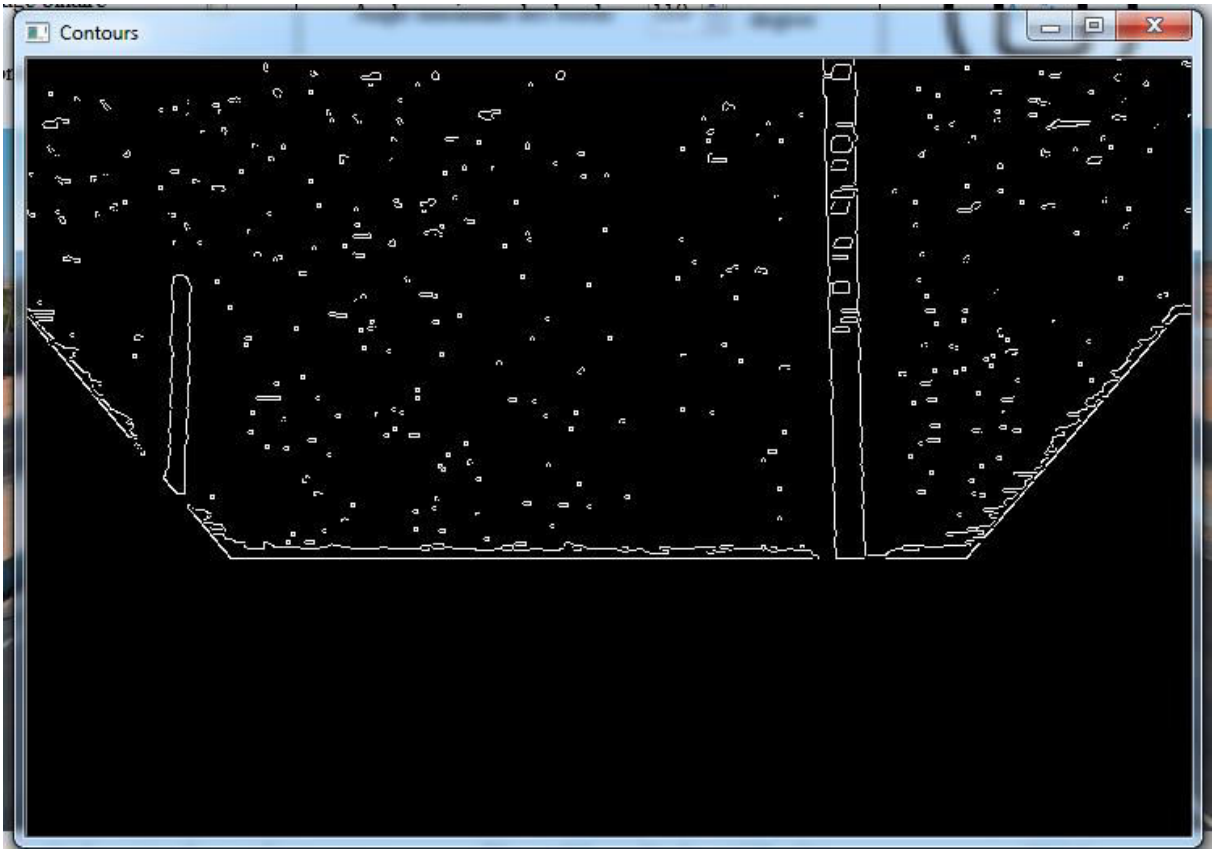


Figure IV.8 : Fenêtre des contours de canny.

IV.8. Conclusion :

Dans ce chapitre, nous avons décrit les étapes de la réalisation de notre système de détection de la voie de roulement, l'environnement matériel et logiciel, ainsi les détails des interfaces développées. En vue de ce qui a été fait dans l'application, on peut dire que notre application répondre certaines caractéristiques suivantes:

- Une interface intuitive et simple à utiliser.
- Rapidité d'exécution.
- Une bonne performance des résultats.
- Un minimum d'erreurs.

CONCLUSION GENARALE :

Le projet de détection de la voie de roulement nous a permis d'approcher les différentes méthodes de traitement d'images et de comprendre les possibilités et les limites de la détection de la voie. De nos jours, le système de détection de la voie est considéré comme étant un des principaux dispositifs d'aide à la conduite. Ses applications sont diverses : à la sécurité routière, aux véhicules intelligents dits autonomes, dans le domaine de la robotique, dans l'aéronautique pour le guidage des avions au sol, etc.

Dans ce projet nous nous sommes intéressés à la problématique de la détection de la voie de roulement. C'est pourquoi un état de l'art des algorithmes de détection a été présenté et ce qui nous a permis d'implémenter notre détecteur de voie.

Lors de notre étude sur les algorithmes de détection, nous avons constaté que les recherches dans le domaine du véhicule intelligent sont nombreuses. En effet, elles ont donné lieu au développement d'une multitude de techniques allant de la simple détection des marquages de la voie et la localisation précise de l'automobile à la reconnaissance des panneaux routiers, détection de piétons, détection de véhicules, etc. Cependant, les solutions proposées jusqu'à maintenant ne sont pas toujours applicables dans des environnements non contrôlés ou non structurés et donc ce domaine connaitra encore d'autres avancés.

BIBLIOGRAPHIE

[1]Alexander BARGETON, "Fusion multi sources pour l'interprétation de l'environnement routière", thèse de doctorat, école national supérieure des mines de Paris (ParisTech), 2009.

[2] shabana HABIB,Mhammad A HANNAN, "Lane departure detection and transmission using Hough Transform Method",PRZEGLAD ELECTROTECNICZNY, vol: 89,N° 5,2013.

[3] Mohamed BOUMEDIENE, " Détection et suivi de voie de roulement par vision ", mémoire demagister, Université des Sciences et de la Technologie d'Oran, 2006.

[4] Xiaodong,Miao,Shunming,Li,Huan Shen, "On Board Lane Detection For Intelligent Vehicule Based On Monocular Vision", International Journal on smart sensing and intelligent systems, vol.5, N°4, december2012.

[5] Sachin Sharma,D.J Shah, "A Mush Advanced And Efficient Lane Detection Algorithm for Intelligent HighWay Safety", University Ahmedabad, India.

[6] Yoan sculo, « Introduction au traitement d'images Détection de contours et segmentation », exposé, université de Troyers, 2009.

[7] R.Aufrère,F.Mormoiton,R.Chapuis,F.Collange,J.P Dérutin, "Détection de route et suivi de vehicules par vision pour l'ACC", Traitement du signal, vol.17, N°3, 2000.

[8]M.Boumediene, A.Ouamri, "Détection de voie basée sur un modèle B-Snake", Conférence, Université Mohamed BOUDIAF d'ORAN (USTO).

[9] Seung-Nam Kang,Soomok Lee,JunhwaHur,Seung-Woo Seo, "Multi-lane Detection based on Accurate Geometric Lane Estimation in Highway Scenarios",IEEE Intelligent Vehicles Symposium (IV) , Michigan, USA, 2014.

[10]http://www.cut-the-knot.org/pythagoras/Cross-Ratio.shtml, consulté le 12/02/2016.

[11]Thittaporn Ganokratanaa, Mahasak Ketcham et Sasipa Sathienpong,"Real-timelanedetectionfordrivingsystemusing imageprocessingbasedonedgedetection and hough transform",éxposé, University of Technology North Bangkok, 2013.

[12]Shu-Chung Yi, Yeong-Chin Chen, Ching-Haur Chang," A lane detection approach based on intelligent vision" , Article, Department of Computer Science and Information Engineering, National Changhua University of Education,Taiwan,(42) 201523-29.

[13]Loïc SABAU, "Pointé automatique sur cible à l'aide de la camera coaxiale de la Leica MS50", Mémoire de soutenance de Diplôme d'Ingénieur INSA Spécialité TOPOGRAPHIE, entreprise EDF-DTG Service Ingénierie Topographie, Lyon, 2014.

[14] tsi.telecom-paristech.fr/pages/enseignement/ressources/beti/ellipses/Hough.html, consulté le 01/05/2016.

[15] "Hough Transform for Lines and Curves", Article, University of California at Santa Cruz.

[16] A. Beloudane et S. Benkerdagh, « Indexation d'une Base de Données MultiMédia Spécialisée (Visages) », Mémoire de Master, Option : Ingénierie des Systèmes d'Information, Faculté des Sciences Exactes et de l'Informatique,Département de Mathématiques et d'Informatique, Université Abdelhamid Ibn Badis, Mostaganem, 2012.

[17] A. Beloudane et S. Benkerdagh, « Indexation d'une Base de Données MultiMédia Spécialisée (Visages) », Mémoire de Master, Option : Ingénierie des Systèmes d'Information, Faculté des Sciences Exactes et de l'Informatique,Département de Mathématiques et d'Informatique, Université Abdelhamid Ibn Badis, Mostaganem, 2012.

[18] http://encyclopedia2.thefreedictionary.com/OpenCV, consulté le 15/05/2016