

# Commande de pelotons de véhicules avec détection de plaques d'immatriculation

## Vehicle platoon control with license plate detection

Roméo MUSELEFU Mbula<sup>1</sup>, Jean-Marie MOANDA Ndeko<sup>2</sup>, Angelo KUTI Lusala<sup>2</sup>

<sup>1</sup> faculté Polytechnique, département d'électromécanique de l'université de Lubumbashi, Lubumbashi, République Démocratique du Congo, romeomuselefum@gmail.com

<sup>2</sup> faculté Polytechnique de l'université de Kinshasa, Kinshasa, République Démocratique du Congo.

**RÉSUMÉ.** Cet article est consacré d'une part au développement de lois de commandes d'un peloton de véhicules en longitudinal et latéral et d'autre part à l'élaboration d'un système de gestion de capture et de traitement d'image de plaques d'immatriculation de véhicules. Ce système est capable de mesurer le volume et les caractéristiques du trafic en temps réel, de jour comme de nuit. De par ses besoins limités en ressource, ce système peut être directement intégré dans une unité embarquée de traitement. En outre, ce système permet de lire automatiquement le numéro des plaques d'immatriculation, en utilisant un ensemble complémentaire de techniques pour extraire du flux d'images le numéro de ces dernières dans le trafic routier.

**ABSTRACT.** This paper is devoted, on the one hand, to the development of longitudinal and lateral vehicle platoon control laws and, on the other hand, to the development of a management system for the capture and processing of vehicle license plate images. This system is capable of measuring the volume and characteristics of traffic in real time, day and night. Due to its limited resource requirements, this system can be directly integrated into an onboard processing unit. In addition, the system can automatically read license plate numbers, using a complementary set of techniques to extract the license plate number from the image stream in the traffic.

**MOTS-CLÉS.** Commande de peloton de véhicules, système de capture vidéo, traitement de plaques d'immatriculation.

**KEYWORDS.** Vehicle platoon control, video capture system, license plate processing.

### Introduction

Les technologies ont considérablement progressé au cours des dernières années dans les domaines de la télécommunication, de l'informatique, de l'électronique et de l'internet, élargissant les perspectives pour les systèmes de transport intelligents. Avec ces progrès technologiques, les outils d'aide à la conduite se sont progressivement diversifiés et multipliés : information trafic, temps de parcours, détection automatique de bouchon (DAB) et d'incident (DAI), lecture automatique de plaque d'immatriculation de véhicule (LAPIV), observatoire de l'évolution du trafic, etc.

Dans le domaine du recueil des données du trafic routier, qui joue un rôle important pour l'exploitation de la route, de nouveaux besoins apparaissent en temps réel, en temps différé, en données individuelles ou des besoins plus spécifiques comme la détection de certaines catégories de véhicules (poids lourds, deux-roues motorisées ou non motorisées, piétons...), la sécurité routière, etc.

Le but général de cette contribution est de concevoir un système utilisable dans l'amélioration de la gestion du trafic routier (but logistique) dans l'augmentation de la rentabilité (but économique), et dans l'optimisation de l'espace de circulation routière afin de diminuer les risques d'accident (but social). Cet article est ainsi divisé en deux parties :

- Premièrement, nous allons utiliser un modèle très simple de peloton de véhicules longitudinal et latéral, de façon à permettre d'éventuelles manœuvres de dépassement d'un peloton par un autre. Nous développons ensuite des loi de commande de suivi de trajectoires pour le véhicule leader ainsi que pour le reste du peloton.

- Deuxièmement, nous élaborons un système de gestion de capture et de traitement d'image des plaques d'immatriculation de véhicules dans le trafic routier.

## 1. Motivation de la partie conduite autonome

Les objectifs de systèmes de type CACC (Cooperative Adaptive Cruise Control) ou systèmes de régulation de vitesse adaptatifs coopératifs sont les suivants (BU et CHAN 2012; MILANÉS et SHLADOVER 2014; WANG, WU et BARTH 2018) :

- Augmenter le débit de la circulation en permettant des inter-distances plus courtes entre les véhicules sur une même voie, sans perdre la stabilité de suivi des voitures
- Réduire le risque de collision par l'arrière, en augmentant la vitesse de réaction aux changements des véhicules précédents et la capacité de communication avec les véhicules précédents
- Améliorer le confort du conducteur en servant non seulement d'aide à la conduite automobile, mais aussi en amortissant les perturbations le long de la chaîne de véhicules
- Réduire la consommation de carburant en améliorant la fluidité du trafic et la résistance aérodynamique si les distances sont suffisamment proches.

Nous élaborons ici un modèle très simple pour un peloton de véhicules en longitudinal et en latéral, de façon à permettre d'éventuelles manœuvres de dépassement d'un peloton par un autre. Ceci est une innovation par rapport à l'ensemble des publications connues des auteurs sur le sujet du CACC qui ne considèrent que la dynamique longitudinale, sous une forme souvent très simplifiée. Une loi de commande pour le véhicule leader est proposée, ainsi qu'une loi pour le reste du peloton. Ces deux lois utilisent la propriété de platitude différentielle des modèles mis en jeu.

## 2. Modèles de véhicules, de pelotons de véhicules

### 2.1. Équations du modèle d'un véhicule

Le modèle  $\mathcal{V}$  d'un véhicule est le suivant

$$\dot{x} = v \sin \psi \quad (1a)$$

$$\dot{y} = v \cos \psi \quad (1b)$$

$$\dot{\psi} = \frac{v}{l} \tan \delta \quad (1c)$$

$$m\dot{v} = \gamma \quad (1d)$$

où  $x$  et  $y$  sont les coordonnées du milieu de l'essieu moteur,  $v$  est la vitesse longitudinale du véhicule,  $\psi$  l'angle de lacet,  $\gamma$  est la force de propulsion et  $\delta$  l'angle de braquage, prises comme commandes.

Les seules constantes de ce modèle sont  $m$ , la masse du véhicule et  $l$ , l'empattement, c.à.d. la distance inter-essieu.

## 2.2. Équations du modèle d'un peloton de véhicules

Un peloton de véhicule  $\mathcal{P}$  est la réunion de modèles  $\mathcal{V}_i$  :

$$\dot{x}_i = v_i \sin \psi_i \quad (2a)$$

$$\dot{y}_i = v_i \cos \psi_i \quad (2b)$$

$$\dot{\psi}_i = \frac{v_i}{l_i} \tan \delta_i \quad (2c)$$

$$m_i \ddot{v}_i = \gamma_i \quad (2d)$$

où  $i \in \{0, \dots, N\}$ ,  $N$  étant le nombre de véhicules du peloton.  $\mathcal{V}_0$  est le modèle du véhicule de tête. Les distances inter-véhicules  $d_i$  sont données par :

$$d_i = x_{i-1} - x_i \quad (3)$$

où  $i \in \{1, \dots, N\}$ . La distance de référence est :

$$d_{i,r} = r + h v_i \quad (4)$$

où  $r$  est une distance inter-véhicule à l'arrêt. Cette distance de référence est l'une de celles permettant d'obtenir une meilleure stabilité du peloton.

## 3. Platitudo différentielle

### 3.1. Platitudo différentielle du modèle d'un véhicule

Le modèle (1) est différentiellement plat (FLIESS et al. 1995), de sorties plates  $x$  et  $y$ . En effet, (1a)<sup>2</sup> + (1b)<sup>2</sup> donne :

$$v = \sqrt{\dot{x}^2 + \dot{y}^2}$$

Ensuite,  $\gamma$  est donnée par :

$$\gamma = \frac{\dot{v}}{m} = \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{m\sqrt{\dot{x}^2 + \dot{y}^2}}$$

Puis, (1b)/(1a) donne :

$$\psi = \text{Atan} \left( \frac{\dot{y}}{\dot{x}} \right)$$

Et, finalement, (1c) fournit :

$$\dot{\psi} = \frac{d}{dt} \left[ \text{Atan} \left( \frac{\dot{y}}{\dot{x}} \right) \right] = \frac{\dot{y}\dot{x} - \dot{y}\dot{x}}{\dot{x}^2 + \dot{y}^2} = \frac{v}{l} \tan \delta$$

Nommons  $P$  le milieu de l'essieu arrière du véhicule (dont les coordonnées sont  $x$  et  $y$ ) et soit  $\mathcal{C}$  la courbe décrite par le point  $P$ . La courbure  $\kappa$  de  $\mathcal{C}$  en  $P$  est donnée par :

$$\kappa = \frac{\dot{y}\dot{x} - \dot{y}\dot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}$$

Et l'on a donc :

$$\dot{\psi} = v\kappa$$

L'angle de braquage  $\delta$  est alors :

$$\delta = \text{Atan}(l\kappa) = \text{Atan}\left(l \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}\right)$$

Finalement, la paramétrisation du système (1) est la suivante :

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \quad (1a)$$

$$\gamma = \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{m\sqrt{\dot{x}^2 + \dot{y}^2}} \quad (1b)$$

$$\psi = \text{Atan}\left(\frac{\dot{y}}{\dot{x}}\right) \quad (1c)$$

$$\delta = \text{Atan}\left(l \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}\right) \quad (1d)$$

### 3.2. Platitude différentielle du modèle d'un peloton de véhicules

Le modèle (2) est différentiellement plat (FLIESS et al. 1995), de sorties plates  $(x_i, y_i)_{i=0,\dots,N}$ . Les paramétrisations associées sont alors :

$$v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2} \quad (2a)$$

$$\gamma_i = \frac{\dot{x}_i\ddot{x}_i + \dot{y}_i\ddot{y}_i}{m_i\sqrt{\dot{x}_i^2 + \dot{y}_i^2}} \quad (2b)$$

$$\psi_i = \text{Atan}\left(\frac{\dot{y}_i}{\dot{x}_i}\right) \quad (2c)$$

$$\delta_i = \text{Atan}\left(l_i \frac{\ddot{y}_i\dot{x}_i - \dot{y}_i\ddot{x}_i}{(\dot{x}_i^2 + \dot{y}_i^2)^{3/2}}\right) \quad (2d)$$

**Remarque 3..1.** Notons qu'il est alternativement possible de prendre  $(d_i, y_i)_{i=0,\dots,N}$  comme sorties plates, en posant  $d_0 = x_0$ . En effet, il suffit de remplacer  $x_i$  par :

$$x_i = d_0 - \sum_{j=1}^{i-1} d_j \quad (3)$$

□

## 4. Suivi de trajectoires

### 4.1. Suivi de trajectoires du leader

Le véhicule leader, correspondant à  $i = 0$ , doit suivre une trajectoire  $(x_{0,r}(t), y_{0,r}(t))$  prescrite à l'avance. La dynamique en  $x_0, y_0$  est obtenue en différentiant les deux premières équations (1a) et (1b) :

$$\ddot{x}_0 = \dot{v}_0 \cos \psi_0 - v_0 \dot{\psi}_0 \sin \psi_0 = \frac{\cos \psi_0}{m_0} \gamma_0 - \frac{v_0^2 \sin \psi_0}{l_0} \tan \delta_0$$

$$\ddot{y}_0 = \dot{v}_0 \sin \psi_0 + v_0 \dot{\psi} \cos \psi_0 = \frac{\sin \psi_0}{m_0} \gamma_0 + \frac{v_0^2 \cos \psi_0}{l_0} \tan \delta_0$$

ou encore, sous forme matricielle :

$$\begin{pmatrix} \ddot{x}_0 \\ \ddot{y}_0 \end{pmatrix} = \begin{pmatrix} \cos \psi_0 & -\frac{v_0^2 \sin \psi_0}{l_0} \\ \frac{m_0}{\sin \psi_0} & \frac{v_0^2 \cos \psi_0}{l_0} \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \tan \delta_0 \end{pmatrix} \quad (1)$$

Pour assurer le suivi, on doit avoir :

$$\begin{pmatrix} \ddot{x}_0 \\ \ddot{y}_0 \end{pmatrix} = \begin{pmatrix} \ddot{x}_{0,r} - \lambda_{x,0} e_{x,0} - \mu_{x,0} \dot{e}_{x,0} \\ \ddot{y}_{0,r} - \lambda_{y,0} e_{y,0} - \mu_{y,0} \dot{e}_{y,0} \end{pmatrix} \quad (2)$$

avec  $e_{x,0} = x_0 - x_{0,r}$ ,  $e_{y,0} = y_0 - y_{0,r}$  et des gains  $\lambda_{d,0}$ ,  $\mu_{d,0}$ ,  $\lambda_{y,0}$ ,  $\mu_{y,0}$  tous positifs. Ceci fournit le bouclage de suivi de trajectoire avec stabilité suivant :

$$\begin{pmatrix} \gamma_0 \\ \tan \delta_0 \end{pmatrix} = \frac{m_0 l_0}{v_0^2} \begin{pmatrix} \frac{v_0^2 \cos \psi}{l_0} & \frac{v_0^2 \sin \psi_0}{l_0} \\ -\frac{\sin \psi_0}{m_0} & \frac{\cos \psi_0}{m_0} \end{pmatrix} \begin{pmatrix} \ddot{x}_{0,r} - \lambda_{d,0} e_{d,0} - \mu_{d,0} \dot{e}_{d,0} \\ \ddot{y}_{0,r} - \lambda_{y,0} e_{y,0} - \mu_{y,0} \dot{e}_{y,0} \end{pmatrix} \quad (3)$$

ce qui se réécrit finalement :

$$\begin{pmatrix} \gamma_0 \\ \tan \delta_0 \end{pmatrix} = \begin{pmatrix} m_0 \cos \psi_0 & m_0 \sin \psi_0 \\ -\frac{l_0 \sin \psi_0}{m_0 \left( \int_0^t \gamma_0(\theta) d\theta \right)^2} & \frac{l_0 \cos \psi_0}{m_0 \left( \int_0^t \gamma_0(\theta) d\theta \right)^2} \end{pmatrix} \begin{pmatrix} \ddot{x}_{0,r} - \lambda_{d,0} e_{d,0} - \mu_{d,0} \dot{e}_{d,0} \\ \ddot{y}_{0,r} - \lambda_{y,0} e_{y,0} - \mu_{y,0} \dot{e}_{y,0} \end{pmatrix} \quad (4)$$

**Remarque 4..1.** Notons que, pour l'implantation numérique, les termes

$$\int_0^t \gamma_0(\theta) d\theta \quad (5)$$

apparaissant dans (4) seront approximés par une discréttisation (par une méthode des trapèzes, de Romberg ou autre) de

$$\int_0^{t-h} \gamma_0(\theta) d\theta \quad (6)$$

où  $h$  est la période d'échantillonnage. Ceci permet d'éviter l'apparition de boucles algébriques □

## 4.2. Suivi de trajectoires local au sein d'un peloton de véhicules

Afin de simplifier l'étude, nous supposerons dans ce qui suit que la voie de circulation est parallèle à l'axe des abscisses  $Ox$ . Si tel n'est pas le cas, des rotations appropriées devront être prises en compte.

Chaque véhicule  $i$  devra alors suivre une trajectoire longitudinale induite par  $d_i(t) \mapsto d_{i,r}(t)$  et une trajectoire latérale  $y_{i,r}(t)$  qui incluera le cas échéant des signmoïdes permettant des manœuvres de dé-passement. La dynamique en  $d_i, y_i$  est alors obtenue par :

$$\begin{aligned} \ddot{d}_i &= \ddot{x}_i - \ddot{x}_{i-1} = \dot{v}_i \cos \psi_i - v_i \dot{\psi}_i \sin \psi_i - \dot{v}_{i-1} \cos \psi_{i-1} - v_{i-1} \dot{\psi}_{i-1} \sin \psi_{i-1} \\ &= \frac{\cos \psi_i}{m_i} \gamma_i - \frac{v_i^2 \sin \psi_i}{l_i} \tan \delta_i - \frac{\cos \psi_{i-1}}{m_{i-1}} \gamma_{i-1} + \frac{v_{i-1}^2 \sin \psi_{i-1}}{l_{i-1}} \tan \delta_{i-1} \end{aligned}$$

$$\begin{aligned}\ddot{y}_i &= \dot{v}_i \sin \psi_i + v_i \dot{\psi} \cos \psi_i \\ &= \frac{\sin \psi_i}{m_i} \gamma_i + \frac{v_i^2 \cos \psi}{l_i} \tan \delta_i\end{aligned}$$

Posant

$$\phi_i = -\frac{\cos \psi_i}{m_i} \gamma_i + \frac{v_i^2 \sin \psi_i}{l_i} \tan \delta_i$$

on obtient la dynamique suivante en  $d_i, y_i$

$$\ddot{d}_i = \phi_{i-1} + \frac{\cos \psi_i}{m_i} \gamma_i - \frac{v_i^2 \sin \psi_i}{l_i} \tan \delta_i \quad (7a)$$

$$\ddot{y}_i = \frac{\sin \psi_i}{m_i} \gamma_i + \frac{v_i^2 \cos \psi}{l_i} \tan \delta_i \quad (7b)$$

ou encore, sous forme matricielle :

$$\begin{pmatrix} \ddot{d}_i - \phi_{i-1} \\ \ddot{y}_i \end{pmatrix} = \begin{pmatrix} \frac{\cos \psi_i}{m_i} & -\frac{v_i^2 \sin \psi_i}{l_i} \\ \frac{\sin \psi_i}{m_i} & \frac{v_i^2 \cos \psi}{l_i} \end{pmatrix} \begin{pmatrix} \gamma_i \\ \tan \delta_i \end{pmatrix} \quad (8)$$

Pour assurer le suivi, on doit avoir :

$$\begin{pmatrix} \ddot{d}_i - \phi_{i-1} \\ \ddot{y}_i \end{pmatrix} = \begin{pmatrix} \ddot{d}_{i,r} - \phi_{i-1} - \lambda_{d,i} e_{d,i} - \mu_{d,i} \dot{e}_{d,i} \\ \ddot{y}_{i,r} - \lambda_{y,i} e_{y,i} - \mu_{y,i} \dot{e}_{y,i} \end{pmatrix} \quad (9)$$

avec  $e_{d,i} = d_i - d_{i,r}$ ,  $e_{y,i} = y_i - y_{i,r}$  et des gains  $\lambda_{d,i}, \mu_{d,i}, \lambda_{y,i}, \mu_{y,i}$ . Ceci fournit le bouclage de suivi de trajectoire avec stabilité suivant :

$$\begin{pmatrix} \gamma_i \\ \tan \delta_i \end{pmatrix} = \frac{m_i l_i}{v_i^2} \begin{pmatrix} \frac{v_i^2 \cos \psi}{l_i} & \frac{v_i^2 \sin \psi_i}{l_i} \\ -\frac{\sin \psi_i}{m_i} & \frac{\cos \psi_i}{m_i} \end{pmatrix} \begin{pmatrix} \ddot{d}_{i,r} - \phi_{i-1} - \lambda_{d,i} e_{d,i} - \mu_{d,i} \dot{e}_{d,i} \\ \ddot{y}_{i,r} - \lambda_{y,i} e_{y,i} - \mu_{y,i} \dot{e}_{y,i} \end{pmatrix} \quad (10)$$

ce qui se réécrit finalement :

$$\begin{pmatrix} \gamma_i \\ \tan \delta_i \end{pmatrix} = \begin{pmatrix} m_i \cos \psi_i & m_i \sin \psi_i \\ -\frac{l_i \sin \psi_i}{m_i \left( \int_0^t \gamma_i(\theta) d\theta \right)^2} & \frac{l_i \cos \psi_i}{m_i \left( \int_0^t \gamma_i(\theta) d\theta \right)^2} \end{pmatrix} \begin{pmatrix} \ddot{d}_{i,r} - \phi_{i-1} - \lambda_{d,i} e_{d,i} - \mu_{d,i} \dot{e}_{d,i} \\ \ddot{y}_{i,r} - \lambda_{y,i} e_{y,i} - \mu_{y,i} \dot{e}_{y,i} \end{pmatrix} \quad (11)$$

## 5. Types de récolte des données du trafic

### 5.1. Motivation

Le système de gestion de capture et de traitement d'image de plaques d'immatriculation de véhicule permet de sécuriser l'exploitation du trafic routier. Les exploitants et les gestionnaires de réseaux routiers ont en charge la gestion du trafic, la sécurité des déplacements avec notamment le maintien de la viabilité et l'information de l'usager. On distingue principalement 4 types de récolte des données : la récolte des données en temps différé, en temps réel, celle des données individuelles et spécifiques (CEREMA 2012).

## **5.2. Récolte des données en temps différé**

Le recueil de données trafic en mode temps différé est la première source d'information de la connaissance du réseau routier. Il concerne de nombreux domaines, outre l'exploitation, comme la conception des infrastructures, l'entretien des chaussées, la sécurité routière, l'environnement, etc.

Il s'agit de l'application la plus courante ; les données sont agrégées en séquences horaires ou journalières pour des besoins statistiques. La plupart des gestionnaires ne s'intéressent, dans ce cas, qu'aux débits et à la composition du trafic, débits des véhicules légers (VL) et des poids lourds (PL), le plus souvent pour la planification de l'entretien des routes.

## **5.3. Récolte des données en temps réel**

Les données sont utilisées au fil de l'eau par les gestionnaires et sont généralement fournies par séquences de 6 minutes ou d'une minute (voire toutes les 20s ou 6s, pour la gestion des trafics importants). La nature des mesures est généralement le débit, la vitesse, la longueur des véhicules et le taux d'occupation de la chaussée (BACELAR 2014).

Elles sont destinées aux besoins d'exploitation pour la gestion du trafic routier comme l'affichage des temps de parcours, la détection automatique des congestions ou simplement le renseignement de l'état du trafic.

## **5.4. Récolte des données individuelles**

Les données individuelles (dites aussi microscopiques) de trafic sont des mesures fines enregistrées au fil de l'eau et stockées véhicule par véhicule. Les paramètres à recueillir sont constitués de la voie de circulation, de l'horodate de passage du véhicule avec une résolution au centième de seconde, de la vitesse, de la longueur, du temps inter véhiculaire et suivant le besoin, de la catégorie et du poids du véhicule (poids par essieux et/ou poids total) (JÜRGGEN 2012).

## **5.5. Récolte des données spécifiques**

Il s'agit principalement de la détection de véhicules particuliers. Les objectifs sont d'étudier, d'alerter, ou de renseigner la présence sur la chaussée de véhicules particuliers comme les deux roues motorisées (2RM), les véhicules de transports en commun (TC), les véhicules de transports de matières dangereuses (TMD), les vélos etc. Les applications sont multiples en matière de statistiques, de sûreté, de stratégie d'itinéraire ou de péage (ROSEY et SUBIRAT 2010).

## 6. Capteurs vidéo

### 6.1. Capteurs vidéo visibles et infrarouges

#### Principe de fonctionnement

Le principe de base est de paramétriser sur l'image de la route des boucles virtuelles dont le comportement sera analogue aux boucles inductives (électromagnétiques) standards (MUSELEFU 2016). Afin d'améliorer les performances globales du système, d'autres techniques de traitement d'image sont appliquées afin d'éliminer le bruit dans l'image, mais surtout de déterminer les mouvements de groupes de pixels, d'extraire les contours des objets, les reconnaître et les classifier via des processus de reconnaissance des formes.

#### Mesures disponibles

Selon les matériels : débit tous véhicules, débits VL/PL, taux d'occupation, temps inter-véhiculaire, distance inter-véhiculaire et vitesse, par sens et par voie. Le séquencement des mesures recueillies dépend du module de traitement électronique couple au compteur. Il y a possibilité, selon les constructeurs, de distinguer les véhicules en quatre classes, de distinguer les 2RM et de donner des niveaux de qualification de trafic. Possibilité également de mesurer les longueurs de queues, de repérer des trajectoires particulières (mouvements tournants) d'intégrer les fonctions de DAI (DAB, véhicule arrêté, contre-sens, etc.) et de détecter les piétons.

#### Limites de fonctionnement

Sensibilité aux conditions météorologiques (pluie, neige), aux illuminations fortes (soleil rasant) ou changeantes, aux masquages et occlusions (végétation, véhicules de grands gabarits), aux flous de bougé (occasionné par le support).

#### Technologies connexes

Algorithmes de traitement d'image et vision par ordinateur sont les technologies connexes des capteurs vidéo visibles et infrarouges.

### 6.2. Capteurs spécialisés dans la lecture automatique de plaques d'immatriculation (LAPI) visible et infrarouge

#### Principe de fonctionnement

En règle générale, les systèmes de Lecture Automatique de Plaque d'Immatriculation (LAPI) utilisent, en cascade, deux techniques de traitements d'image : La détection d'objet pour repérer les plaques d'immatriculation potentielles, suivi de la reconnaissance optique de caractère afin d'identifier les caractères alphanumériques de la plaque (MUSELEFU 2016).

Dans quelques cas, la LAPI est couplée à une phase de détection/reconnaissance de véhicule afin de renforcer les performances de la détection de plaques. Afin d'accélérer et d'améliorer les performances de la phase de détection des plaques, une technique courante est d'utiliser un a priori sur les zones poten-

---

tielles de détection et de réduire la recherche, par l’analyse des localisations des détections précédentes, au cadre des caméras ayant une prise de vue large (plusieurs voies).

Les cameras utilisées pour l’acquisition des séquences d’images sont toutes matricielles. Dans le domaine visible, deux types de capteurs sont utilisés, les CCD et les CMOS. Dans le cas où les caméras sont utilisées dans un environnement sombre, il est généralement choisi de travailler dans le domaine infrarouge. Les capteurs utilisés sont également de type CCD (mais sans filtre infra-rouge) et un illuminateur permet d’avoir une source de lumière infrarouge (non visible par l’œil humain) qui est diffusée sur la scène visée.

C’est la réflexion des sources infrarouge qui permet de constituer une image en niveau de gris sur la camera. Cette technique ne permet pas d’obtenir d’image couleur, mais permet de s’affranchir des conditions de luminosité par des réglages auto-adaptatifs. Dans la plupart des cas, ce sont des caméras en niveaux de gris couplées avec un illuminateur infrarouge qui sont utilisées.

Toutefois, dans le cas de prises de vue orientées pour l’analyse de trafic, on commence à voir apparaître l’utilisation de cameras HD couleurs (voire jour/nuit) utilisées sans illuminateurs infrarouges.

Dans le cadre des dispositifs de LAPI, plusieurs solutions matérielles sont proposées :

- L’utilisation d’une caméra en champ proche, qui ne filme alors qu’une seule voie de circulation. Ceci permet d’avoir un capteur abordable et un traitement d’image facilité, mais impose d’avoir autant de capteurs que de voies de circulation,
- L’utilisation d’une caméra Haute Définition (HD) qui permet de filmer l’ensemble des voies. Ceci, même si le capteur utilisé est plus couteux, permet de réduire les couts globaux de maintenance et de mise en place. Toutefois, les performances de ce type de systèmes sont dépendantes de la puissance du calculateur.

Dans le cadre du Contrôle Automatisé, il est à noter que les cameras utilisées sont des appareils photos numériques (APN) de type CCD de bonne qualité et prenant l’ensemble des voies. Ceux-ci sont couples à des flashes et la prise de vue est déclenchée par la mesure du dépassement de la vitesse limite autorisée à l’aide d’un cinémomètre. Le traitement des images peut se faire en temps réel ou en post-traitement. Les dispositifs LAPI fonctionnent aussi bien en latéral (bord de voie ou terre pleine centrale) qu’en surplomb.

Ce dernier positionnement est généralement préféré par les industriels car il limite les effets des déformations de l’image et d’éloignements des voies, tout en évitant les masquages occasionnés par les véhicules de grands gabarits. Ces dispositifs sont conçus pour fonctionner aussi bien en rapprochement qu’en éloignement des véhicules. Certains sont capables de gérer les deux sens sur plusieurs voies en même temps.

## Mesures disponibles

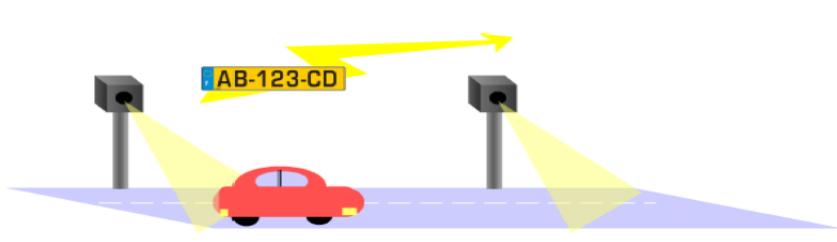
Le séquencement des mesures recueillies dépend de la logique développée par l’industriel. On dispose généralement de la vitesse moyenne sur une section de route. Les applications visées sont multiples : péages automatiques, contrôle d’accès et gestion des parkings, identification des véhicules volés et d’usurpation de plaques (dispositifs intégrés dans les voitures de police et associés à des algorithmes de reconnaissance des véhicules : marque, modèle, couleur), vérification des temps de pause des PL, suivi du transport des matières dangereuses, etc.

## Limites de fonctionnement

Les performances sont liées à la qualité du couple capteur/traitements mais aussi aux paramétrages physiques du capteur (position de prise de vue, éloignement, etc.) et aux paramétrages des algorithmes de détection (zone de recherche, ...) et de reconnaissance (seuils de contraste, etc.).

## Technologies connexes

Les algorithmes de traitement d'image et de vision numérique sont les technologies connexes des capteurs de Lecture Automatique de Plaques d'Immatriculation (LAPI). La figure 1 représente les capteurs de Lecture Automatique de Plaques d'Immatriculation (LAPI) :



**Figure 1.:** Capteurs de Lecture Automatique de Plaques d'Immatriculation (LAPI).

## 7. Traitement et analyse d'images

### 7.1. Motivation

Le traitement et l'analyse des images regroupent plusieurs disciplines que l'on classe généralement en deux catégories (BOLON et CHEHIKIAN 1999) :

- Les processus de bas niveau, qui nécessitent très peu d'informations sur le contenu des images. Il s'agit des processus de filtrage, d'amélioration et de restauration d'images, ainsi que l'extraction d'indices.
- Les processus de haut-niveau, qui fonctionnent en aval de ceux de bas niveau, peuvent nécessiter des informations sur le contenu des images. Il s'agit de la reconnaissance de formes, de la reconstruction 3-D, de processus cognitifs de façon générale.

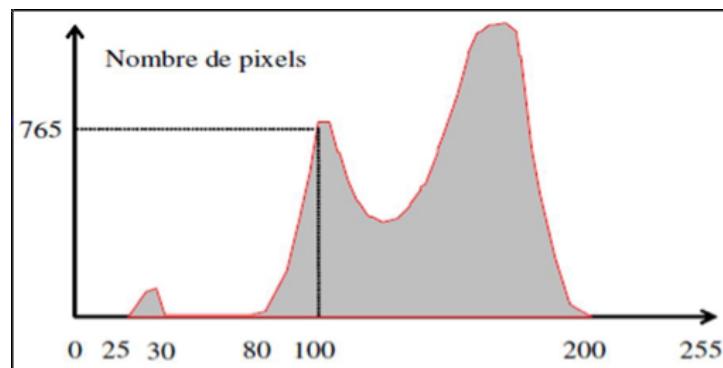
### 7.2. Histogramme d'une image

L'histogramme d'une image  $h(i)$  est la fonction qui associe à une valeur d'intensité  $i$  (ou de couleur), le nombre de pixels de l'image ayant cette valeur. C'est un graphique qui donne la distribution des comptes numériques de cette image, c'est-à-dire le nombre ou pourcentage de pixels ayant un compte numérique donné. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus faible (à gauche) au plus élevé (à droite). Ainsi, l'histogramme d'une image dont les comptes numériques vont de 0 à 255 est un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées (LINGRAND 2004).

Un histogramme qui contient uniquement un objet (i.e. une population de pixels connexes avec des comptes numériques "similaires") de teinte assez uniforme sur un fond d'image uniforme a en général un

aspect uni-modal de type gaussien. Un histogramme avec plusieurs modes (tel celui représenté en figure 2) peut être dû à la présence de différents types de population dans l'image ou à différentes configurations d'éclairement ou d'observation d'une même population.

Toute image multicanaux est caractérisée par plusieurs histogrammes. Ainsi, une image codée en RGB a 3 histogrammes qui donnent la distribution des comptes numériques de ses composantes rouge, verte et bleue. L'histogramme cumulé représente la distribution cumulée des intensités des pixels d'une image, c'est-à-dire le nombre de pixels ayant une intensité lumineuse supérieure.

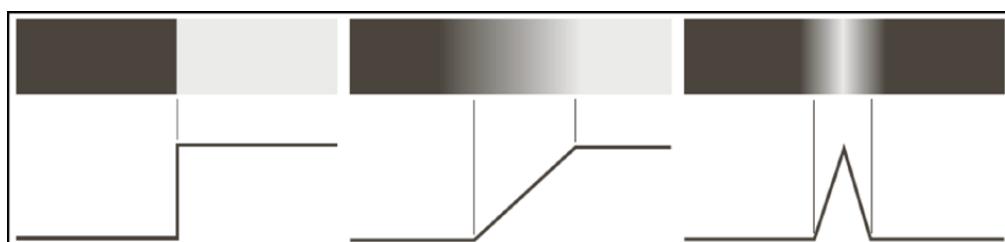


**Figure 2.:** Histogramme d'image codée sur un octet.

Le codage sur un octet signifie que les comptes numériques peuvent prendre 256 valeurs. L'aire grisée est égale au nombre total de pixels de l'image.

### 7.3. Détection de contours

Les variations d'intensité de lumière et de couleur sont très bien perçues par le système visuel humain. En effet, une frontière est définie comme un endroit de l'image où la transition en niveaux de gris est la plus importante entre 2 régions. Ces frontières constituent le contour des objets. Le contour s'appuie alors sur les transitions plus ou moins prononcées (de type marche, rampe ou trait) ou encore sur les frontières entre deux zones homogènes distinctes ou non, représentées en figure 3 suivante (AHMED 1998) : En général, un élément de contour est un point de l'image appartenant à la frontière entre objets



**Figure 3.:** Types de contour (ALJAHDALI 2013; GILL 2019). À gauche : marche ; au milieu : rampe ; à droite : trait.

ou régions de niveaux de gris différents.

Ainsi, la détection de contours est souvent le premier problème qu'on rencontre en traitant une image et elle constitue un problème classique du traitement d'images. Mais les difficultés de la détection des contours proviennent du bruit important présent dans les images (bruit du capteur, bruit d'échantillonnage, irrégularité des surfaces des objets, etc.).

Il existe actuellement deux catégories de méthodes de détection de contours :

- Les méthodes simples basées sur des filtres fixes (Sobel, Prewitt, etc.)
- Les méthodes complexes, basées sur des filtres adaptatifs. Les plus connues sont ceux de Canny et Deriche.

Nous présentons ici simplement une méthode basée sur des filtres adaptatifs.

#### 7.4. Détection de contours par seuillage adaptatif

Le seuillage est l'une des approches les plus basiques pour déterminer une région d'intensité relativement uniforme. C'est une approche qui fonctionne bien sur un certain nombre d'images bien particulières ou d'images qui ont subi des pré-traitements, par exemple un lissage (LINGRAND 2004).

Il existe deux grandes familles de seuillage : Le seuillage fondé sur l'analyse de l'histogramme, et le seuillage adaptatif, par information de contours. Le seuillage adaptatif est un type de seuillage paramétrique, c'est-à-dire qu'il tient compte des variations d'éclairements présents dans l'image. En fait, ce type d'opération fait intervenir les plus proches voisins d'un pixel. L'une des variantes du seuillage adaptatif consiste à déterminer le seuil en calculant une moyenne pondérée autour du pixel considéré à une constante près. Pour une moyenne simple, tous les voisins du pixel sont pondérés par un coefficient constant. Pour une moyenne pondérée à proprement parler, les voisins du pixel considéré sont pondérés par une fonction gaussienne de leur distance au pixel considéré comme centre (KAEHLER 2014).

Si  $g$  est la valeur du pixel en nuance de gris et  $s$  la valeur de la moyenne calculée, cette moyenne sera appelée seuil et les contours seront déterminés de la manière suivante :

- Si  $g > s$ , le pixel est à l'intérieur d'une région,
- Si  $g \leq s$ , le pixel appartient à un contour (une frontière).

#### 7.5. Détection des points d'intérêt

##### Motivation

Les points d'intérêt sont des points possédant des caractéristiques permettant de les distinguer des autres points de l'image, comme par exemple un fort contraste. Les points d'intérêt sont également utilisés dans différentes applications : la robotique, l'indexation ou la reconnaissance d'objets (on peut extraire d'une image une suite de points caractéristiques d'un objet afin de pouvoir l'indexer dans une base de données), le suivi (ou tracking des objets dans une séquence d'images), ou encore l'imagerie médicale (les points d'intérêt correspondent à des points anatomiques particuliers). On distingue trois catégories de détecteurs (GALES, CROUZIL et CHAMBON 2010) :

- les détecteurs s'appuyant sur une détection de contours ;
- les détecteurs exploitant les niveaux de gris ;
- les détecteurs utilisant des modèles paramétriques.

##### Méthodes fondées sur les contours

On commence par appliquer un détecteur de contours dans l'image puis on cherche sur ces contours les points d'inflexion, les points où la courbure est localement maximale ou encore les points d'intersections de plusieurs contours (les jonctions). Ces méthodes sont parmi les plus anciennes.

## Méthodes fondées sur l'intensité

Ces méthodes sont basées sur la dérivée des niveaux de gris pour repérer les points où l'intensité varie fortement dans une ou plusieurs directions. On trouve également dans cette catégorie le célèbre détecteur de Harris que nous détaillerons plus particulièrement.

## Méthodes fondées sur des modèles paramétriques

Ces méthodes s'appuient sur la déformation d'un modèle paramétrique de coin pour qu'il se rapproche des niveaux de gris au voisinage d'un coin. Ce détecteur est précis à condition d'avoir de bonnes valeurs initiales pour les paramètres du modèle.

## Les détecteurs

Il existe une quantité importante de détecteurs de points d'intérêt, les plus utilisés étant le détecteur de Harris, le détecteur SIFT de Lowe, le détecteur de Kadir, le détecteur de Moravec, le détecteur de Beaudet et le détecteur SUSAN de Smith. Nous ne décrivons que le détecteur de Harris. De façon générale, un détecteur de point d'intérêt consiste à calculer une valeur de réponse représentative de l'intérêt pour chaque pixel de l'image et ensuite à sélectionner les meilleurs.

**Le détecteur de Harris** Il se base sur une fonction d'autocorrélation du signal c'est-à-dire sur les changements du signal dans plusieurs directions. On commence par calculer en chaque pixel  $P(i, j)$  de l'image la matrice d'autocorrélation (ou des moments du second ordre ou tenseur de structure) :

$$M(i, j) = \begin{pmatrix} \mu_{11}(i, j) & \mu_{12}(i, j) \\ \mu_{21}(i, j) & \mu_{22}(i, j) \end{pmatrix} \quad (1)$$

Avec :

$$\begin{aligned} \mu_{11}(i, j) &= \sum_{p=-n}^n \sum_{q=-n}^n \omega(p, q) I_i^2(i + p, j + q) \\ \mu_{22}(i, j) &= \sum_{p=-n}^n \sum_{q=-n}^n \omega(p, q) I_j^2(i + p, j + q) \\ \mu_{12}(i, j) &= \mu_{21}(i, j) = \sum_{p=-n}^n \sum_{q=-n}^n \omega(p, q) I_i(i + p, j + q) I_j(i + p, j + q) \end{aligned}$$

Où  $I_i$  et  $I_j$  sont les dérivées premières des niveaux de gris de l'image obtenue par convolution avec les masques de dérivation  $\omega()$  issus du filtre gaussien et  $(p, q)$  sont des points de lissage gaussiens tels que :

$$\sum_{p=-n}^n \sum_{q=-n}^n \omega(p, q) = 1 \quad (2)$$

On calcule ensuite les valeurs propres de chaque matrice  $M(i, j)$  :

$$\lambda_1(i, j) = \frac{1}{2} \left( \mu_{11}(i, j) + \mu_{22}(i, j) + \sqrt{(\mu_{11}(i, j) - \mu_{22}(i, j))^2 + 4\mu_{12}(i, j)^2} \right) \quad (3)$$

$$\lambda_2(i, j) = \frac{1}{2} \left( \mu_{11}(i, j) + \mu_{22}(i, j) - \sqrt{(\mu_{11}(i, j) - \mu_{22}(i, j))^2 + 4\mu_{12}(i, j)^2} \right) \quad (4)$$

On peut alors caractériser le pixel  $P(i, j)$  de la manière suivante :

- 
- Si les deux valeurs propres sont grandes, alors on est en présence d'un point d'intérêt ;
  - Si les deux valeurs propres sont petites, alors le pixel étudié est dans une zone homogène,
  - Si les deux valeurs propres sont très différentes, alors le motif de texture au voisinage du pixel  $P(i, j)$  est unidirectionnel.

## Calcul du flux optique

Le calcul du flux optique consiste à déterminer le mouvement apparent à partir d'un couple ou d'une séquence d'images (BARATE 2008). De nombreuses méthodes ont été proposées depuis une trentaine d'années pour effectuer ce calcul. Nous allons présenter rapidement les principales.

### Méthode différentielle de Horn et Schunck

Les méthodes différentielles utilisent les dérivées spatio-temporelles calculées sur l'image pour déterminer les vecteurs de déplacement correspondant au flux optique. Cela suppose que l'intensité des pixels reste constante au cours du temps, c'est-à-dire

$$I_t(x, y) = I_t(x + v_x, y + v_y) \quad (5)$$

où  $I_t(x, y)$  est l'intensité au temps du pixel de coordonnées  $x, y$  et où  $(v_x, v_y)$  est le vecteur vitesse correspondant au flux optique. En utilisant les dérivées spatio-temporelles de l'image, cette équation peut s'écrire de la manière suivante, en négligeant les dérivées d'ordre supérieur :

$$\frac{\partial I_t}{\partial t} + \frac{\partial I_t}{\partial x} v_x + \frac{\partial I_t}{\partial y} v_y = 0 \quad (6)$$

Cette équation de contrainte du gradient est la base de toutes les méthodes de calcul différentielles. Le problème vient du fait qu'elle est essentiellement mal posée, car elle compte deux inconnues  $v_x$  et  $v_y$  pour une seule équation linéaire. Il est donc nécessaire de poser d'autres contraintes sur la forme du flux optique afin de la résoudre. La différence entre toutes les méthodes différentielles vient du choix de ces contraintes et dans une moindre mesure de la méthode utilisée pour le calcul des dérivées spatio-temporelles.

Horn et Schunck supposent que le mouvement apparent provient du déplacement d'objets d'une certaine taille dans le champ de vision, et par conséquent que le flux optique résultant doit être lisse et régulier. Pour modéliser cela, ils utilisent une contrainte de régularité globale sur le flux optique. Cette contrainte cherche à minimiser le gradient des vecteurs vitesses du flux optique. Formellement, ils cherchent à minimiser l'erreur  $E_G$ , ainsi que le gradient de vitesse  $E_{V^2}$ , soit :

$$E_G = \frac{\partial I_t}{\partial t} + \frac{\partial I_t}{\partial x} v_x + \frac{\partial I_t}{\partial y} v_y \quad (7)$$

$$E_{V^2} = \left( \frac{\partial v_x}{\partial x} \right)^2 + \left( \frac{\partial v_x}{\partial y} \right)^2 + \left( \frac{\partial v_y}{\partial x} \right)^2 + \left( \frac{\partial v_y}{\partial y} \right)^2 \quad (8)$$

## 8. Système de capture et de traitement d'image des plaques d'immatriculation de véhicules

### 8.1. Motivation

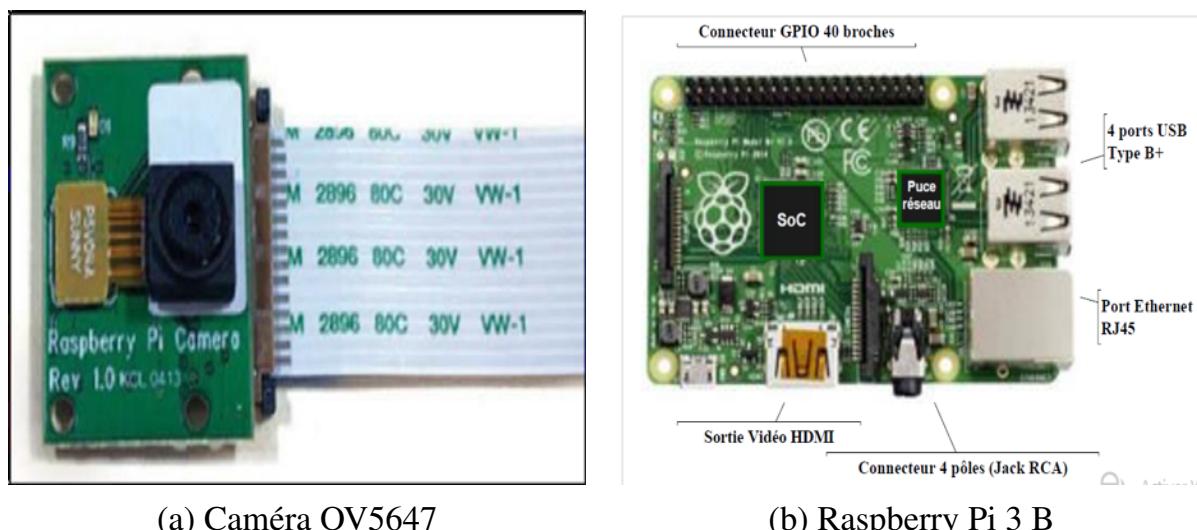
L'application de vision que nous devons réaliser devrait avoir des fonctionnalités simples, mais efficaces pour répondre à la reconnaissance de formes en temps réel. Nous avons choisi d'utiliser la bibliothèque OpenCV, nous paraissant la mieux adaptée pour les raisons qui suivent :

- il s'agit d'une bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie ;
- elle propose un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API pour les langages C, C++, Ruby, JAVA, Matlab, Python et bien d'autres ;
- elle est fortement orientée en temps réel ;
- elle est distribuée sous une licence libre pour les plateformes Windows, GNU/Linux, Android et MacOS ;
- elle est de plus en plus utilisée, et sa documentation est plus accessible. De plus, de nombreux forums d'utilisateurs de cette bibliothèque existent, d'où un apprentissage rapide. Utilisée en C/C++, elle permet de gagner en rapidité de l'exécution.

### 8.2. Fonctionnement du système

#### Camera

Pour la capture d'image, nous avons utilisé une caméra Toshiba Camileo OV5647, numérique Full HD, munie d'un zoom de 10x 5 Mpix, présentée en figure 4 (a), fonctionnant sur un mini-ordinateur Raspberry Pi présenté en figure 4 (b).

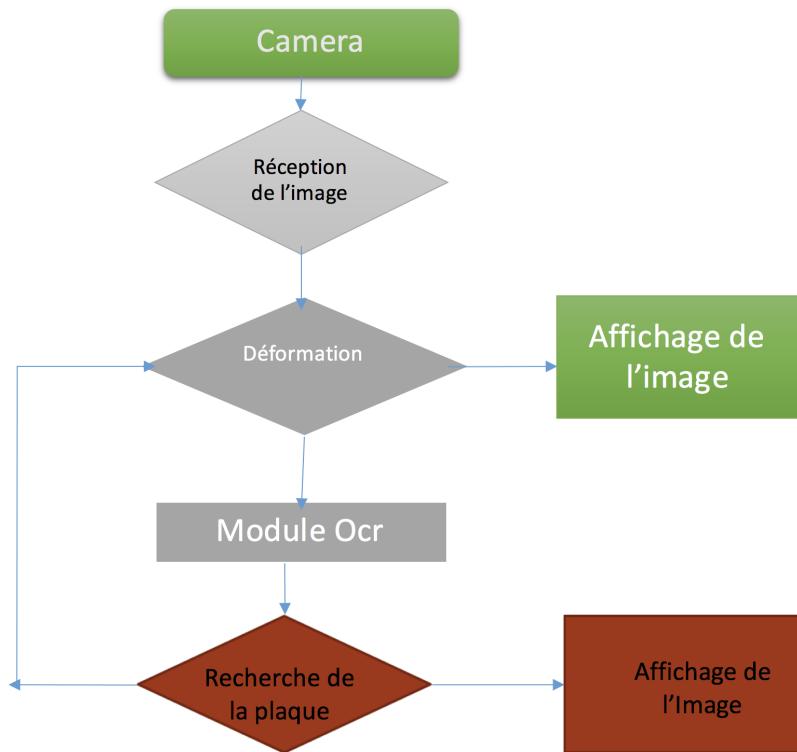


**Figure 4.:** Caméra et Raspberry PI.

#### Logiciel du système

Nous avons utilisé trois bibliothèques de programmation à savoir éclipse Luna, Open CV, module OCR et trois langages de programmation à savoir Java, C++ et C.

Par ailleurs, nous représentons à la figure 5, la fonctionnalité de notre système de capture et de traitement d'image des plaques d'immatriculation de véhicules dans le trafic routier.



**Figure 5.:** Schéma de fonctionnement du système de capture et de traitement d'image des plaques d'immatriculation de véhicules.

Programmes de capture vidéo et de traitement des plaques d'immatriculation

Code pour la capture d'image

Le code ci-dessous effectue la capture vidéo :

```
import io
import socket
import struct
import time
import picamera
import numpy as np

client_socket = socket.socket()
client_socket.connect(('192.168.0.3', 21))
frames=1
FPS=1
connection = client_socket.makefile('wb')
try:
    with picamera.PiCamera() as camera:
        camera.resolution=(640,480)
        camera framerate=FPS
        camera.start_preview()
        time.sleep(2)
        start=time.time()
        camera.capture_sequence(['image%2d.jpg' % i
                               for i in range(frames)], use_video_port=True)
```

---

```

finish = time.time()
print(' %d frames capturees a %.2ffps' % (frames,frames/(finish-start)))
my_stream_out=io.BytesIO()
for i in range(frames):
    if i < 10:
        my_stream_out=open('image '+str(i)+'.jpg','rb')
    else:
        my_stream_out=open('image '+str(i)+'.jpg','rb')
        connection.write(my_stream_out.read())
        connection.write('\xFF\xD8\x6E\xFF\xD9')
finally:
    connection.close()
    client_socket.close()

```

## Codes de traitement d'image

**Première phase** Cette phase permet de déformer l'image reçue de la caméra de manière à rendre lisible l'image. Le code fourni est en C, utilisant OpenCv.

```

void superpose (const IplImage *src1, IplImage* dst, int n)
{
    int x, y;
    uchar *p_dst, *p_src1, *p_tmp;

    for (y = 0; y < dst->height; ; ++y) {
        for (x = 0; x < dst->width; ; ++x) {
            p_src1 = cvPtr2D (src1, y, x, NULL);
            p_dst = cvPtr2D (dst , y, x, NULL);
            p_tmp = cvPtr2D (dst , y, x, NULL);
            *p_dst = *p_src1;

            if (*p_dst >n) {
                *p_dst=255;
            } else {
                if (*p_dst <=30) {
                    *p_dst=255;
                }
                else {
                    *p_dst=0;
                }
            }
        }
    }
}

```

**Deuxième phase** Cette phase consiste à rechercher dans l'image déjà déformée les caractères sur l'image. Le langage utilisé est java.

```

void fix_origin (IplImage* img)
{
    if (img->origin == IPL_ORIGIN_BL)
    {
        cvConvertImage(img, img, CV_CVTIMG_FLIP);
        img->origin = IPL_ORIGIN_TL;
    }
}

```

---

```

try {
    String chemin="E:/Image/image_dst11.png";
    String chemin_txt="G:/Fichier.txt";

    Tesseract instance = Tesseract.getInstance();
    String result = instance.doOCR(new File(chemin));
    PrintWriter fluxSortie= null;
    try {
        fluxSortie = new PrintWriter(new FileOutputStream(chemin_txt));
    } catch(FileNotFoundException e) {
        System.out.println("Erreur d'ouverture Fichier.txt");
        System.exit(0);
    }
    fluxSortie.println(result);
    fluxSortie.close();
    System.out.println("La valeur recuperée est :" + result);
} catch (TesseractException e) {
    System.err.println(e.getMessage());
}
}
}

```

**Troisième phase** Dans cette phase, on recherche parmi les caractères relevés le format des caractères correspondant au format de plaque d'immatriculation. Le langage utilisé est le C.

```

char * resolver_numero(int numero,char * re)
{
    FILE * fp;
    char buffer [1024];
    size_t bytes_read ;
    char* match;
    char match2[8];
    char result[11]="";
    char* test;
    char test3[6];

    fp = fopen ("G:\Base_de_donnee.txt", "r");
    bytes_read = fread (buffer, 1, sizeof(buffer), fp);
    fclose (fp);
    buffer[bytes_read] = '\0';

    sprintf(match2,"%d%s",numero," :%s\n");
    sprintf(test3,"%d %s",numero,":");
    test=test3;
    match=strstr(buffer,test);
    if (match==NULL)
        return re;

    if (match!=NULL)
    {
        sscanf(match,match2,result);
        re=result;
        return re ;
    }
    memset (buffer, '\0', sizeof(buffer));
}

```

**Quatrième phase** Cette phase nous permet de rechercher le fichier correspondant au nom de la plaque retrouvée précédemment, puis d'afficher le résultat. Le langage utilisé est le C.

```
int resolver_plaque(int nn)
{
    FILE * fp;
    FILE * fpf;
    char buffer [1024];
    char buffer_match [1024];
    size_t bytes_read ;
    char* match ;
    char* buf_match ;
    char* match_test;
    char match_resultat[100]="";
    char match_chemin[100]="G:/Base_General/";
    int testeur=2;
    int n;
    char * re=NULL;

fp = fopen ("G:/Doc.txt", "r");//A remplacer par Fichier.txt
bytes_read = fread (buffer, 1, sizeof(buffer), fp);
fclose (fp);
buffer[bytes_read] = '\0';
match=buffer;
for (n=1;n<10;n++) { //Debut for
    while ( match < buffer + bytes_read ) {//Debut while
        match_test = strstr(match , resolver_numero(n,re));
        if ( match_test != NULL ) {
            strncpy(match_resultat,match_test,8);
            match_resultat[8] = '\0';
            testeur=0;
            n=11;//pour quitter la boucle for si il y a correspondance
            match =buffer + bytes_read + 11;//pour quitter la boucle while si il y
        }
        match += strlen(match) + 1;
    }//Fin while
}//Fin for
memset(buffer,'0',sizeof(buffer));
if (( testeur==2 )&(nn==255))
{
    // afficher l'image
    return 1;
}
if ( testeur == 0 )
{
    memset(buffer,'0',sizeof(buffer));
    strncat(match_chemin,match_resultat,8);
    strncat(match_chemin,".txt", 4);
    fpf = fopen(match_chemin,"r");
    bytes_read = fread (buffer, 1, sizeof(buffer), fpf);
    fclose (fpf);
    buffer[bytes_read] = '\0';
    printf ("Plaque Identifiée\n");
    printf ("%s\n",&buffer);
    memset(buffer,'0',sizeof(buffer));
}
```

```
    return 2;
}
return 0;
}
```

## 9. Conclusion

Cet article a proposé d'une part des lois de commande pour un peloton de véhicules, avec prise en charge possible des dépassements et d'autre part un système de gestion de capture et de traitement d'image des plaques d'immatriculation de véhicules dans le trafic routier.

La propriété de platitude différentielle des modèles adoptés pour les pelotons permet d'obtenir des lois de commande de suivi de trajectoires de manière aisée

Malgré les nombreuses recherches dans le domaine de la vision par ordinateur, l'analyse et la compréhension automatique du trafic reste un challenge. Le capteur vidéo semble être le choix d'avenir parce qu'il offre toutes les données nécessaires à la réalisation de nombreuses fonctions (aussi bien du comptage de véhicule que de la détection d'incident). C'est un capteur très flexible et économiquement intéressant.

Cependant, les processus de traitement chargés d'analyser ce flux vidéo sont des systèmes complexes et difficiles à développer. Ils doivent être efficaces pour tous types de trafic (et jusqu'à 2x3 bandes de circulation) et dans toutes les conditions possibles d'observation. Ils doivent réaliser plusieurs fonctions et générer des résultats en temps réel. Plusieurs produits existent aujourd'hui qui répondent partiellement à ces critères, mais les nouveaux développements en matière d'algorithme et les puissances de calcul offertes par les ordinateurs modernes ouvrent de nouvelles perspectives dans l'analyse de trafic.

La qualité des images vidéo est un facteur important pour la robustesse et la précision de l'analyse. Les images doivent être nettes et contrastées, de jour comme de nuit, quelles que soient les conditions météorologiques. La configuration minimale suffisante est un flux d'image au format PAL (résolution de 720x576 pixels entrelacés à une fréquence de 25 Hz) issus d'une caméra standard. C'est le type de caméra le plus utilisé et le plus économique. Les services de surveillance des routes disposent déjà aujourd'hui d'un nombre important de ces caméras en fonction. Cependant, si le système d'analyse doit fournir des mesures très précises (en vitesse et caractéristiques des véhicules) alors il est possible d'utiliser des caméras plus performantes, capables de fréquences et/ou de résolutions plus élevées.

## Bibliographie

- AHMED, Sidi Mohamed (1998). « Segmentation d'images par détection de région, de contour, ou par morphologie mathématique. Étude comparative et évaluation des performances ». Thèse de doctorat. Université cheikh Anta DIOP.
- ALJAHDALI, Sultan (2013). *Advanced Techniques for Image Segmentation Image Processing*. Saarbrücken : LAP LAMBERT Academic Publishing. ISBN : 9783659322617.

- BACELAR, Alexis (2014). *Vers un contrôle automatique du nombre d'occupants des véhicules*. Rapport d'étude. Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement. URL : [https://www.cerema.fr/system/files/documents/2017/12/rapport\\_technos\\_nb\\_doccupants\\_vfinale\\_cle211dd4.pdf](https://www.cerema.fr/system/files/documents/2017/12/rapport_technos_nb_doccupants_vfinale_cle211dd4.pdf).
- BARATE, Renaud (nov. 2008). « Apprentissage de fonctions visuelles pour un robot mobile par programmation génétique ». Theses. Université Pierre et Marie Curie - Paris VI. URL : <https://tel.archives-ouvertes.fr/tel-00811614>.
- BOLON, Philippe et Alain CHEHIKIAN (1999). « Analyse d'image Traitements de bas niveau ». In : H3608 v1. URL : <https://www.techniques-ingenieur.fr/base-documentaire/archives-th12/archives-technologies-logicielles-et-architecture-des-systemes-tiahb/archive-1/analyse-d-image-traitements-de-bas-niveau-h3608/>.
- BU, Fanping et Ching-Yao CHAN (2012). « Adaptive and Cooperative Cruise Control ». In : *Handbook of Intelligent Vehicles*. Springer London, p. 191-207. URL : [https://doi.org/10.1007/978-0-85729-085-4\\_9](https://doi.org/10.1007/978-0-85729-085-4_9).
- CEREMA (2012). *Panorama des systèmes de recueil de données de trafic routier*. Rapport d'étude. Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement. URL : <https://www.cerema.fr/fr/centre-ressources/boutique/panorama-systemes-recueil-donnees-trafic-routier>.
- FLEISS, Michel et al. (juin 1995). « Flatness and defect of non-linear systems : introductory theory and examples ». In : *International Journal of Control* 61.6, p. 1327-1361. URL : <https://doi.org/10.1080/00207179508921959>.
- GALES, Guillaume, Alain CROZIL et Sylvie CHAMBON (2010). « Détection de points d'intérêt pour la mise en correspondance par propagation (regular paper) ». In : *Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2010), Caen, 19/01/2010-22/01/2010*. <http://www.afrif.asso.fr> : AFRIF-AFIA.
- GILL, Rana (2019). *Comparative Analysis of Edge Detection Techniques*. Saarbrücken : LAP LAMBERT Academic Publishing. ISBN : 6200095655.
- JÜRGGEN, Abs (2012). *Intelligent and secure truck parking*. Deployment Guideline F&L-DG01. <https://www.transport.gov.mt/IntelligentAndSecureTruckParking.pdf-f1739>. Easy Way ([www.easyway-its.eu](http://www.easyway-its.eu)).
- KAehler, Adrian (2014). *Learning OpenCV, 2nd Edition*. City : O'Reilly Media, Incorporated. ISBN : 9781449314651.
- LINGRAND, Diane (2004). *Introduction au traitement d'images*. Paris : Vuibert informatique. ISBN : 978-2-7117-4841-9.
- MILANÉS, Vicente et Steven E. SHLADOVER (2014). « Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data ». In : *Transportation Research Part C : Emerging Technologies* 48, p. 285-300. URL : <https://doi.org/10.1016/j.trc.2014.09.001>.
- MUSELEFU, Roméo (2016). « conception et réalisation d'un Radar feu Rouge ». Mémoire de master 2. Polytechnique Unilu.
- ROSEY, F. et P. SUBIRAT (2010). *La détection des deux-roues motorisés :quels systèmes, quels outils ?* Rapport d'étude du Certu. Centre d'Études sur les réseaux, les transports, l'urbanisme et les construc-

---

tions publiques. URL : <http://www.bv.transports.gouv.qc.ca/mono/1069450.pdf>.

WANG, Ziran, Guoyuan WU et Matthew J. BARTH (2018). « A Review on Cooperative Adaptive Cruise Control (CACC) Systems : Architectures, Controls, and Applications ». In : *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. URL : <https://doi.org/10.1109/itsc.2018.8569947>.