

*République Algérienne Démocratique et Populaire*

*وزارة التعليم العالي والبحث العلمي*

*Ministère de l'Enseignement Supérieur et de la Recherche Scientifique*

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE d'ORAN Mohamed Boudiaf



*Faculté de Génie Electrique*

*Département d'Electronique*

*Spécialité : Electronique      Option : Techniques de communication moderne*

**MEMOIRE**

Présenté par

Mlle. OUESSAI ASMAA

Pour l'obtention du diplôme de Magister en Electronique

**Thème**

**Estimation des paramètres de trafic  
routier par vision artificielle**

Soutenu le 3-7-2012 devant la commission d'examen composée de :

<u>Qualité</u>	<u>Nom et Prénoms</u>	<u>Grade</u>	<u>Etb d'origine</u>
Président	Mr Ouamri Abdelaziz	Professeur	USTO
Rapporteur	Mr Keche Mokhtar	Professeur	USTO
Examineur	Mr Ould Mammam Madani	M.conf. A	Univ. Mostaganem
Examineur	Mr Loukil Abdelhamid	M.conf. A	USTO

Année universitaire : 2011-2012

# *Remerciements*

*Mes remerciements vont tout particulièrement à Dieu, le tout puissant pour la volonté, la santé et la patience qu'il m'a données durant toutes ces années d'études.*

*Je tiens à exprimer ma très profonde gratitude à mon encadreur Mr. Keche qui n'a ménagé aucun effort pour me prendre en charge pour la réalisation de ce travail.*

*Je le remercie pour son encadrement, son aide, ses conseils précieux et sa disponibilité.*

*Je remercie Monsieur Ouamri, directeur du laboratoire signaux et images LSI pour l'attention qu'il a accordé à mon travail, ainsi que pour son rôle en tant que président du jury de ce mémoire.*

*J'exprime ma reconnaissance à Monsieur Ould Mammar Maître de conférences à l'université d'Abdelhamid Ibn badis de Mostaganem, et Monsieur Loukil Maître de conférences à l'USTO pour leurs contributions à ce jury en tant qu'examineurs.*

*Mes remerciements vont ensuite, à Melle. Benoudnine, S.Hanafî, Mr Dahmani et Mr Meche. Je tiens ici à leur exprimer toute ma gratitude pour m'avoir aidé, conseillé, et encouragé.*

*Mes remerciements s'adressent également aux membres du LSI, et à toutes mes collègues.*

## Liste des abréviations

<b>AR</b>	Autoregressive model
<b>ARMA</b>	Autoregressive moving average model
<b>DCT</b>	Discrete cosine transform
<b>GPS</b>	Global positioning system
<b>HMM</b>	Hidden Markov model
<b>ITS</b>	Intelligent transport systems
<b>KKT</b>	Karush Kuhn Tucker
<b>KL</b>	Kullback-Leibler
<b>LDS</b>	Linear dynamic system
<b>LVQ</b>	Learning vector quantization
<b>ML</b>	Maximum likelihood
<b>MPEG</b>	Moving picture experts group
<b>NN</b>	Nearest neighbor
<b>RBF</b>	Radial basic function
<b>ROI</b>	Region of interest
<b>STAR</b>	Spatio-temporel autoregressive model
<b>SVD</b>	Singular value decomposition
<b>SVC</b>	Support vector classification
<b>SVM</b>	Support vector machines

# Table des matières

<b>Dédicace</b> .....	<b>i</b>
<b>Remerciements</b> .....	<b>ii</b>
<b>Liste des abréviations</b> .....	<b>iii</b>
<b>Table des figures</b> .....	<b>viii</b>
<b>Liste des tableaux</b> .....	<b>x</b>

## **Chapitre1: introduction générale**

1.1 Introduction .....	1
1.2 Problématique.....	1
1.3 Congestion routière .....	2
1.4 Solutions .....	2
1.5 Les différentes techniques de mesure des paramètres de trafic.....	3
1.5.1 Les capteurs intrusifs .....	3
1.5.2 Les capteurs non-intrusifs .....	4
1.6 Intérêt d'une approche vidéo.....	5
1.7 Objectif du travail.....	6
1.8 Plan du mémoire .....	6

## **Chapitre 2: Revue de littérature sur les systèmes d'estimation du trafic routier par vision**

2.1 Introduction.....	8
2.2 Méthodes basées sur la détection et la poursuite de mouvement des véhicules.....	8
2.3 Evaluation du trafic à partir des données microscopiques.....	12
2.4 Méthodes basées sur le traitement global de la scène (paramètres macroscopique) .....	13
2.5 Conclusion .....	14

## **Chapitre 3: Estimation des paramètres microscopiques**

3.1 Introduction.....	15
-----------------------	----

3.2 Variables microscopiques.....	15
3.2.1 Estimation de la vitesse.....	15
3.2.1.1 Détection des objets en mouvement.....	16
3.2.1.2 Poursuite de mouvement.....	20
3.2.1.3 Calcul de la vitesse .....	23
3.2.2 Estimation du taux d'occupation de la route .....	25
3.2.3 Distance inter -véhicules .....	29
3.2.4 Débit moyen du trafic routier.....	30
3.3 Résultats.....	31
3.3.1 Base de vidéos du trafic routier .....	31
3.3.2 Estimation de la vitesse .....	32
3.3.3 Estimation du taux d'occupation de la route.....	33
3.3.4 Estimation de la distance Inter- véhicules moyenne.....	34
3.3.5 Estimation du débit du trafic .....	35
3.4 Conclusion .....	36

#### **Chapitre4 : Estimation des paramètres macroscopique**

4.1 Introduction.....	37
4.2 Estimation du mouvement.....	37
4.3 Mise en correspondance de blocs d'images.....	38
4.3.1 Largeur de la fenêtre de recherche.....	39
4.3.2 Taille des blocs.....	40
4.3.3 Fonction de ressemblance.....	40
4.3.4 Complexité des algorithmes de mise en correspondance.....	41
4.4 Les textures dynamiques.....	43
4.4.1 Un domaine de recherche récent et dynamique.....	43
4.4.2 Discussion sur les textures dynamiques.....	45
4.4.3 Taxonomie de textures dynamiques.....	47
4.4.4 Modélisation de textures dynamiques .....	49
4.4.5 Modèle de texture dynamique.....	50
4.4.6 Synthèse de textures dynamiques .....	52

4.4.7 Comparaison entre deux textures dynamiques.....	53
4.4.7.1 Distribution des probabilités .....	53
4.4.7.2 Projection entre espaces d'états.....	54
4.4.7.3 Calcul de la distance KL entre espaces d'états.....	55
4.5 Résultats .....	56
4.5.1 Estimation de la vitesse par mise en correspondance.....	56
4.5.2 Densité des vecteurs de mouvement.....	57
4.6 Conclusion .....	59

## **Chapitre 5 : Méthodes de classification**

5.1 Introduction.....	60
5.2 Les k plus proches voisins: classifieur « paresseux ».....	60
5.2.1 Algorithme de classification par KNN.....	62
5.2.2 Critères de la méthode .....	62
5.2.3 Exemple de classification par KNN.....	63
5.3 Réseau de neurone LVQ .....	63
5.3.1 L'apprentissage compétitif.....	63
5.3.2 VQ Vector Quantization (quantification vectorielle).....	65
5.3.3 LVQ (Learning Vector Quantization).....	65
5.3.3.1 Structure de réseaux de neurones LVQ.....	66
5.3.3.2 Algorithme de classification par LVQ.....	67
5.4 Les Machines à Vecteurs de Supports (SVM).....	68
5.4.1 Principe de SVM.....	68
5.4.2 Cas simple : SVM linéaire à deux classes.....	70
5.4.3 Cas non linéairement séparable.....	73
5.4.4 Les SVM multi-classes.....	76
5.5 Validation croisée.....	77
5.6 Conclusion.....	79

## **Chapitre 6 : Résultats finals d'estimation du trafic routier**

6.1 Introduction.....	80
-----------------------	----

6.2 Classification du trafic routier .....	80
6.2.1 Classification du trafic routier par l'utilisation des paramètres moyens .....	82
6.2.1.1 Classification par le plus proche voisin (KPPV).....	82
6.2.1.2 Classification par le LVQ .....	83
6.2.1.3 Classification par SVM.....	85
6.2.2 Classification du trafic routier par l'utilisation du modèle de texture dynamique.....	94
6.2.2.1 Classification par KPPV.....	94
6.2.2.2 Classification par SVM.....	97
6.3 Conclusion .....	101
<b>Conclusion générale.....</b>	<b>102</b>

# Table des figures

Figure 1.1 Capteurs intrusifs.....	4
Figure 1.2 Capteurs non-intrusifs.....	5
Figure 2.1 Processus de traitement pour l'estimation des paramètres des véhicules.....	9
Figure 3.2 Détection de mouvement.....	20
Figure 3.3 Recherche de la trajectoire d'objet sur 3 images.....	22
Figure 3.4 Poursuite de mouvement (Scène 1).....	23
Figure 3.5 Poursuite de mouvement (Scène 2).....	23
Figure 3.6 Fonction d'échelle.....	24
Figure 3.7 Extrema d'une image numérique.....	26
Figure 3.8 Maxima d'une fonction.....	26
Figure 3.9 Extraction des maxima régionaux par une reconstruction géodésique.....	27
Figure 3.10 Dilatation et reconstruction géodésique.....	28
Figure 3.11 Extraction de la route par maxima régionaux.....	29
Figure 3.12 Processus d'estimation de la vitesse inter-véhicules.....	29
Figure 3.13 Vérité terrain de la base vidéo utilisée.....	31
Figure 3.14 Estimation de la vitesse moyenne.....	32
Figure 3.15 Estimation du taux d'occupation moyen .....	33
Figure 3.18 Estimation de la distance inter-véhicules moyenne.....	34
Figure 3.20 Estimation de la concentration du trafic.....	35
Figure 4.1 Notion de vecteurs de mouvement.....	37
Figure 4.2 Estimation du mouvement d'un bloc.....	38
Figure 4.3 Principe de la mise en correspondance par bloc.....	39
Figure 4.5 Estimation des vecteurs de déplacement par la mise en correspondance des blocs.....	42
Figure 4.6 Principales publications sur les textures dynamique depuis 1992 jusqu'au 2010	43
Figure 4.7 Exemple de textures dynamiques.....	44
Figure 4.8 Taxonomie de textures dynamiques.....	48



Figure 4.10 Synthèse de texture dynamique.....	53
Figure 4.12 Estimation de la vitesse de mouvement par mise en correspondance de blocs.	56
Figure 4.13 Estimation de la vitesse de mouvement par mise en correspondance de blocs.	57
Figure 4.14 Densité de vecteurs de mouvement.....	57
Figure 4.16 Estimation de la densité du trafic routier.....	58
Figure 5.1 Principe de KNN.....	61
Figure 5.2 Structure de réseaux de neurones LVQ.....	66
Figure 5.3 Principe de la technique SVM.....	69
Figure 5.4 Principe de la technique SVM.....	70
Figure 5.5 Exemple d'hyperplans séparateurs.....	71
Figure 5.6 Deux classes non linéairement séparables.....	73
Figure 5.7 Principe de "l'outlier".....	74
Figure 5.8 Schéma de validation croisée d'ordre r.....	78
Figure 6.2 Structure du réseau de neurones LVQ.....	83
Figure 6.3 Comparaison entre noyau linéaire et noyau RBF.....	87
Figure 6.4 Comparaison entre la stratégie un contre un et un contre tous pour la classification SVM.....	89
Figure 6.5 Comparaison entre un noyau RBF et linéaire pour la classification par SVM (un contre tous).....	89
Figure 6.6 Etats du trafic réel et estimé par SVM avec le schéma un contre tous.....	90
Figure 6.7 Exemples d'estimation du trafic routier.....	91
Figure 6.8 Estimation de l'état du trafic par SVM avec le schéma un contre tous pour les 4 tests.....	93
Figure 6.9 Exemples de patches utilisés.....	94
Figure 6.10 Taux de classification en fonction du nombre de composantes principales.....	95
Figure 6.11 Estimation de l'état du trafic routier par texture dynamique et KPPV.....	96
Figure 6.12 Estimation de l'état du trafic par texture dynamique et SVM.....	98
Figure 6.13 Comparaison entre la classification par textures dynamique et KPPV ou SVM.	99
Figure 6.14 Estimation de l'état du trafic par texture dynamique et SVM pour les 4 tests....	100

## Liste des tableaux

Tab 6.1	Classification de trafic routier par KPPV et les paramètres moyens.....	82
Tab 6.2	Classification de trafic routier par LVQ et les paramètres moyens.....	84
Tab 6.3	Classification de trafic routier par SVM et les paramètres moyens.....	86
Tab 6.4	Classification de trafic routier par SVM par la stratégie un contre tous.....	88
Tab 6.5	Matrice de confusion pour SVM avec le schéma un contre tous.....	91
Tab 6.6	Résultats de la classification par SVM pou les 4 tests.....	92
Tab 6.7	Table de confusion rassemblant les 4 tests.....	92
Tab 6.8	Matrice de confusion pour la classification par textures dynamiques et KPPV.....	96
Tab 6.9	Table de confusion pour la classification par texture dynamique et SVM.....	98
Tab 6.10	Comparaison entre les résultats de la classification par texture dynamique et KPPV ou SVM.....	99
Tab 6.11	Table de confusion rassemblant les 4 tests.....	100

## **1.1 Introduction**

Si de nombreux aspects de notre vie sont devenus plus agréables grâce à l'utilisation de technologies avancées, le secteur du transport par contre a mis un temps assez long pour pouvoir combler son retard. Aujourd'hui, ces progrès sont devenus incontournables. Les véhicules deviennent de plus en plus sûrs, mais en contrepartie l'environnement routier est devenu plus complexe, ceci est dû principalement à l'accroissement rapide du nombre de véhicules et les conséquences qui en résultent tel que les accidents de circulation, les embouteillages de routes,...etc.

Le développement d'un système capable de contrôler et de prévoir le trafic routier est l'une des priorités des systèmes de transport intelligents (ITS). Le but principal de ces systèmes est d'informer les conducteurs des dangers potentiels bien avant d'avoir un contact visuel.

Ces technologies, qui visent à accroître la sécurité, le rendement et la commodité de la conduite, sont actuellement en cours de conception et de développement.

## **1.2 Problématique**

La congestion routière est une réalité que chaque conducteur doit au moins affronter à un instant de sa vie. Cela affecte des millions d'individus à travers le monde et génère des effets indésirables à tous niveaux : personnels, professionnels et sociaux.

En effet, les congestions conduisent généralement à un ralentissement drastique de la circulation et engendrent des retards conséquents, ces retards ont un impact important sur la qualité de vie des usagers : les conducteurs soumis à des embouteillages fréquents sont sujets à des problèmes de stress, de bruits,... etc, avec un accroissement de risque d'accidents. Ce phénomène est largement identifié comme une source de coûts indirects (problèmes de santé). De plus, la congestion a des conséquences directes sur l'augmentation de la consommation énergétique, dans le contexte actuel d'envolée des prix du pétrole, ceci engendre une diminution drastique du pouvoir d'achat. Dans le secteur économique, ce

problème engendre des retards de livraison de marchandises, qui peuvent être dramatiques pour les entreprises. Enfin, du point de vue environnemental, les congestions ont un impact néfaste et négatif du fait de l'augmentation de la pollution due à une consommation supplémentaire de carburant qui entraîne des coûts écologiques en croissance exponentielle.

On peut donc conclure que le phénomène de congestion du trafic routier est un problème socio-économique crucial qui nécessite de rechercher des solutions pouvant être mises en œuvre rapidement.

### **1.3 Congestion routière**

L'apparition d'une congestion implique nécessairement l'existence d'un lieu de contrainte, c'est-à-dire d'un point où, au moins temporairement, la demande devient supérieure à l'offre (ex : une congestion dans le cas des heures de pointe), ou l'offre devient inférieure à la demande (ex : une congestion causée par un incident). Si la demande excède la capacité, alors des véhicules seront ralentis à l'entrée de l'infrastructure, formant ainsi un bouchon. Ces véhicules excédentaires seront à chaque instant plus nombreux qu'à l'instant précédent. Comme chaque véhicule occupe une certaine longueur de voie, la longueur de la file d'attente ne fera que croître en proportion du nombre de véhicules présents dans cette file d'attente.

La congestion peut être récurrente (se produire au même endroit, au même moment chaque jour de semaine, ou un jour de fin de semaine) ou non récurrente s'il s'agit de celle qui est occasionnée par des perturbations comme un accident ou des travaux.

### **1.4 Solutions**

Si la solution naturelle de ce problème consiste à construire de plus en plus d'infrastructures routières, elle ne peut être retenue aussi facilement du fait des prix exorbitants qu'elle exige et du manque d'espaces disponibles.

Une autre solution possible pour diminuer les congestions repose sur l'encouragement des usagers à utiliser les transports en commun et à assurer l'acheminement des biens par train ou par bateau. Cependant, afin de réaliser ce type d'opérations connues sous le nom de

« transport multimodal » et d'assurer le transfert des marchandises vers les trains et les navires, ces modes de transport doivent offrir une alternative viable qui exige souvent de gros investissements.

Il s'avère donc nécessaire de mettre en œuvre des actions pour que l'utilisation de la capacité des infrastructures existantes soit optimale. Dans ce contexte, la commande intelligente du trafic représente le seul moyen efficace pour éliminer ou du moins alléger les effets des congestions. Ce type d'action permet en effet de proposer des solutions fiables et efficaces pour un transport sûr, sans incidents avec une diminution de la pollution. Cependant, bien que les systèmes intelligents de transport (ITS) ont été considérés comme une solution possible pour optimiser l'usage des infrastructures, peu de solutions ont été proposées à ce jour, en raison de la complexité du problème à traiter.

Les technologies de mesure et d'acquisition d'informations sur le trafic routier sont variées, nous en introduisons dans ce qui suit les principales.

## **1.5 Les différentes techniques de mesure des paramètres de trafic**

On distingue généralement deux catégories de capteurs: les systèmes intrusifs (qui sont installés dans la chaussée) et les systèmes non-intrusifs (installés aux abords des routes, ou encore embarqués dans des véhicules).

### **1.5.1 Les capteurs intrusifs**

Les capteurs intrusifs sont la plupart du temps des boucles inductives (Fig 1.1), placées dans chaque voie de circulation, qui vont réagir au passage des véhicules. La mise en place d'une double boucle permet de mesurer également la vitesse et la longueur des véhicules. L'inconvénient majeur des boucles inductives est la nécessité d'intervenir dans la route (et donc de perturber le trafic) pour l'installation et la maintenance. De plus, les boucles sont limitées à des fonctions de comptage et de mesure et de classification individuelle par bande de circulation.

Un autre type des capteurs intrusifs est les tubes pneumatiques: ce sont des tubes creux en caoutchouc tendus au travers de la voie (Fig 1.1) permettant de mesurer la pression aux

passages des essieux des véhicules et d'en déduire la vitesse. Leur utilisation est temporaire puisque la précision de mesure se dégrade rapidement.

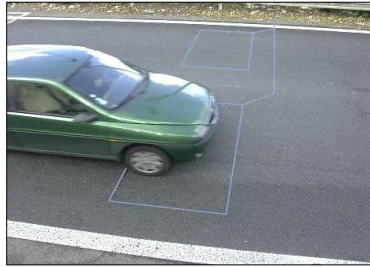
**Boucle inductive****Tubes pneumatiques**

Fig 1.1- Capteurs intrusifs

### 1.5.2 Les capteurs non-intrusifs

Les capteurs non-intrusifs sont des senseurs distants qui utilisent diverses technologies: radar, laser, ultrasons ou vidéo.

Le radar est capable de compter et de mesurer la vitesse des véhicules (par effet Doppler ou par modulation de fréquences) sur une bande de circulation quel que soit le climat. Les radars sont généralement des équipements assez onéreux.

Les lasers émettent un faisceau d'énergie qui balaie la surface de détection (une voie de circulation). Le temps de parcours est mesuré et sert à modéliser un profil du véhicule. En utilisant deux plans de détection successifs, un système laser est également capable de mesurer la vitesse du véhicule. Avec la capture du profil 3D du véhicule, les lasers permettent une classification détaillée et précise des véhicules. Ils opèrent généralement pour une seule bande de circulation et sont assez coûteux.

Les capteurs ultrasoniques travaillent de façon similaire au radar mais dans une gamme de fréquences plus faible (20000 Hz). Ils peuvent être perturbés par les émissions de certains véhicules et ont des difficultés à mesurer des déplacements rapides. Pourtant, les considérations économiques font de ces capteurs un choix privilégié des opérateurs.

Enfin, un capteur basé sur l'analyse vidéo va utiliser une caméra pour capturer des images de la route à analyser. Une caméra placée à une distance suffisante peut enregistrer jusqu'à

2x3 bandes de circulation. Une unité de traitement va ensuite réaliser l'interprétation de ces images pour extraire les véhicules et évaluer leur comportement.

Les caméras opérant dans le domaine du visible sont affectées par tous les phénomènes qui altèrent la visibilité (nuit, pluie, neige, brouillard...). Cependant, la caméra est devenue aujourd'hui un capteur fiable et très peu coûteux et d'autre part, elle fournit une représentation (les images) directement exploitable par l'être humain, car ce signal est proche de celui perçu par le cerveau et surtout par l'œil, qui est l'un de ses capteurs privilégiés. Elle a été largement installée sur les routes ces dernières années pour des besoins de vidéo surveillance passive.



**Capteur Radar**



**Capteur ultrasonique**



**Caméra**

Fig 1.2- Capteurs non-intrusifs

## 1.6 Intérêt d'une approche vidéo

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 de comptage de véhicules que de la détection d'incidents). 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, 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.

## 1.7 Objectif du travail

L'objectif principal de notre travail est d'estimer l'état du trafic routier par un système de vision, et ceci en comparant deux approches :

Une première approche basée sur l'estimation des paramètres individuels des véhicules (traitement microscopique), par une détection et une poursuite de mouvement, tels que la vitesse, le nombre de véhicules, la distance inter-véhiculaire, afin de déduire les paramètres moyens qui caractérisent le flux routier. Nous déterminons ensuite la combinaison de ces paramètres qui donne la meilleure estimation de l'état du trafic routier par l'utilisation de trois classifieurs : KNN, LVQ et SVM.

La seconde approche est basée sur une considération du trafic routier dans sa globalité (traitement macroscopique) sans passer par la détection et la poursuite des mouvements individuels des véhicules, qui est une méthode très sensible aux conditions variables du trafic routier. Deux paramètres globaux pour la classification du trafic routier ont été considérés, la vitesse moyenne estimée à partir du flux optique mesuré par la technique de mise en correspondance des blocs d'images et la texture dynamique. En se basant sur l'un de ces paramètres le trafic routier est classé en trois catégories : fluide, moyen, ou congestionné.

## 1.8 Plan du mémoire

Ce mémoire présente un système complet d'estimation de l'état du trafic routier, il est structuré comme suit :

Le chapitre 2 présente une revue de littérature des différents travaux qui traitent l'estimation du trafic routier par systèmes de vision.

Le chapitre 3 décrit les différentes méthodes adoptées pour l'estimation des principaux paramètres et variables du trafic routier à partir de la détection et la poursuite des mouvements individuels des véhicules (microscopique).

Le chapitre 4 est consacré à l'estimation du trafic routier avec des méthodes de traitement globale (macroscopique), deux méthodes y sont en particulier présentées.



Le chapitre 5 présente les différentes méthodes de classifications utilisées pour catégoriser le trafic routier.

Le chapitre 6 présente les résultats expérimentaux obtenus de la classification du trafic routier.

Nous terminons ce mémoire par une conclusion générale et des perspectives qui peuvent contribuer à ouvrir d'autres horizons aux chercheurs.

## 2.1 Introduction

Reconnaitre l'état du trafic routier quelque soit sa nature, les conditions climatiques, ou encore les conditions variables de la luminosité, est une faculté prodigieuse.

Beaucoup de travaux coexistent et tentent de surmonter les obstacles précités en proposant plusieurs solutions, des solutions basées sur la détection et la poursuite des véhicules permettant de donner des informations sur le comportement individuel de chaque véhicule et d'exprimer le comportement du conducteur; ainsi que des solutions basées sur la description globale du flux des véhicules. Celles ci bien qu'elles soient mieux adaptées pour résoudre les problèmes de la gestion du flux routier, néanmoins elles présentent encore quelques insuffisances: elles demeurent, en particulier, incapables de décrire les phénomènes locaux ou individuels.

Dans ce chapitre nous allons discuter des principaux travaux qui se sont intéressés aux systèmes d'estimation du trafic routier en adoptant les deux approches.

## 2.2 Méthodes basées sur la détection et la poursuite de mouvement des véhicules

Durant la dernière décennie, de nombreuses méthodes de détection et de poursuite de mouvement de véhicules ont été proposées dans la littérature scientifique.

Le but des méthodes de détection de mouvement est de séparer les véhicules de leurs environnements afin d'extraire des informations utiles pour le traitement de poursuite, telles que la position, la taille,...etc.

Le suivi des véhicules en mouvement dans une séquence d'images est actuellement un sujet de recherche très actif en vision par ordinateur, le but est de reconnaître les véhicules d'une image à une autre, et de reconstruire leur trajectoire afin d'extraire les informations qui caractérisent chacun d'entre eux tel que la distance parcourue, la vitesse,...etc.

Le schéma suivant résume le processus de traitement pour l'estimation des paramètres des véhicules:

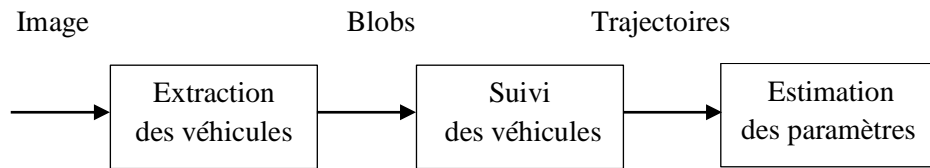


Fig 2.1- Processus de traitement pour l'estimation des paramètres des véhicules

Parmi les travaux qui ont traité ces problèmes, on peut citer :

Jin-Cyuan et Shih [JIS10] ont construit un système de surveillance de trafic routier en autoroute capable d'estimer des paramètres de trafic tel que le nombre et les classes des véhicules, le système proposé comprend trois étapes : extraction des régions de véhicules, poursuite de mouvement et extraction des paramètres de trafic.

La méthode de soustraction de fond est premièrement utilisée pour détecter les véhicules sur la route. Les deux auteurs ont utilisé quelques propriétés géométriques pour éliminer les fausses détections et un algorithme de soustraction d'ombre pour améliorer la précision de la détection. La méthode de poursuite utilisée consiste à associer les graphes (les contours) afin de trouver la correspondance entre les véhicules détectés à des instants différents. Dans ce travail les auteurs ont utilisé des vidéos avec différentes conditions d'éclairage pour prouver l'efficacité de leur système.

Ehsan Adeli Mosabbab et autres [MSF07] ont utilisé quelques paramètres statiques pour chaque trame de la vidéo tels que la moyenne et l'écart type afin de détecter et de segmenter les véhicules ; cette approche a été utilisée pour la détection de véhicules en mouvement dans une scène congestionnée, où il s'avère nécessaire de trouver des solutions pour surmonter le problème de fausses détections. Dans leur travail ils ont proposé de soustraire l'ombre afin de mieux séparer les véhicules très près les uns des autres. Pour montrer la robustesse et la précision de cette approche ils ont utilisé des scènes routières sous différentes conditions climatiques telles que le bruit, la pluie, la neige, et des conditions d'éclairage variables.

Zhe Liu et Yangzhou Chen [ZYZ09] ont proposé un système d'estimation des paramètres de trafic routier basé sur la détection de mouvement par l'algorithme de différence de

trames consécutives, et la poursuite par application de l'algorithme camshift qui s'est avéré efficace pour le suivi des véhicules de formes et de tailles variables et sous différentes conditions de luminosité. Ils ont ensuite reconstruit la trajectoire individuelle de chaque véhicule, ce qui donne la possibilité d'estimer les paramètres individuels de chaque véhicule, tels que la distance parcourue ainsi que la vitesse.

Xie Lei et Guangxi [XGZ05] ont utilisé le filtre de Kalman pour la poursuite des véhicules en mouvement en temps réel dans une scène routière, leur travail comprend deux étapes : détection de mouvement et poursuite. Ils ont testé leur algorithme sur différentes scènes et pour différentes conditions climatiques. Les résultats obtenus ont prouvé la robustesse de l'algorithme en temps réel et pour différentes conditions de luminosité.

François Bardet et Thierry Château [BAC08] se sont intéressés à la poursuite d'objets multiples dans un environnement routier en utilisant une ou plusieurs caméras, pour cela ils ont employé une méthode qui combine un filtre particulaire à une Chaîne de Markov.

Vamsi Krishna et Hanmandlu [VAH10] ont présenté une méthode d'estimation de la vitesse des véhicules, le mouvement est extrait par l'utilisation d'une équation basée sur la projection sphérique qui relie le mouvement de l'image au mouvement de l'objet. La tâche de poursuite est réalisée par l'algorithme de Kanade-Luca-Tomasi. L'équation de mouvement a été reformulée en un modèle d'espace d'état dynamique à laquelle ils ont appliqué le filtre de Kalman et sa version étendue pour l'estimation de la vitesse de l'objet et pour la prédiction de sa nouvelle position. Leur travail a été testé sur une scène en utilisant une caméra non calibrée, cet algorithme est simple et précis pour le calcul de la vitesse en temps réel.

Daniel J. Dailey et autres [DAP00] ont développé un algorithme pour l'estimation de la vitesse moyenne du trafic routier en utilisant une séquence d'images à partir d'une caméra non calibrée ils ont affirmé qu'une calibration de la caméra n'est pas nécessaire pour l'estimation de la vitesse, et pour cela ils ont utilisé :

- des relations géométriques intrinsèques disponibles dans l'image.
- certaines hypothèses qui permettent de réduire le problème à une géométrie d'une seule

dimension (1-D).

- Une différenciation des trames pour l'isolation des contours en mouvement et la poursuite des véhicules entre les trames.
- des paramètres à partir de la distribution de longueurs de véhicules pour l'estimation de la vitesse.

Leur méthode a présenté une alternative viable à la calibration de la caméra.

M.Burlin et autres [BHC09] ont proposé un système d'analyse de trafic routier capable de compter et de détecter les véhicules à l'arrêt ou en contre-sens. En définissant des pixels ambigus pour les objets détectés, le système utilise une modélisation de la scène pour améliorer la détection et le suivi des objets. La première étape de ce système consiste à extraire des informations sur la géométrie de la scène (position et caractérisation des voies). Les bordures des voies, l'estimation de la profondeur dans l'image, et les informations de mouvement sont ensuite utilisées pour aider à la segmentation des objets et à leur suivi image après image. Les résultats obtenus sont prometteurs et montrent la robustesse du système proposé pour le suivi de plusieurs objets simultanément et la correction des erreurs de segmentation.

Xinting Pan et autres [XYA10] ont proposé un système de surveillance de trafic pour la détection et le comptage des véhicules, tout d'abord, ils extraient la Région d'intérêt (ROI) de la vidéo en utilisant une combinaison de la soustraction de fond et la détection de contours. Après la détection ils ont utilisé une méthode automatique de division de voies. Enfin en se basant sur la largeur des voies, ils ont employé plusieurs fenêtres d'auto-adaptation pour le comptage des véhicules au lieu de la méthode de comptage traditionnelle à base de fenêtre fixe. Les résultats expérimentaux ont prouvé la robustesse de leur système.

Quelques travaux se sont intéressés à l'évaluation du trafic routier à partir de ces paramètres microscopiques. Dans ce qui suit, on en présentera les principaux.

## 2.3 Evaluation du trafic à partir des données microscopiques

Panita Pongpaibool et autres [PTN07] ont présenté une méthode d'évaluation du trafic routier à partir des données extraites des images, en utilisant la logique floue et des techniques de neurologiques adaptatives. Le système est conçu pour émuler l'expertise humaine en spécifiant trois niveaux de congestion. Les informations sur le trafic ont été extraites à partir de la détection et la poursuite des véhicules, elles comprennent: le volume du trafic et la vitesse moyenne, les résultats obtenus ont prouvé que la combinaison de ces deux informations donne plus de précision que la seule utilisation de l'information de vitesse moyenne.

Pattara-atikom et autres [PPT06] ont proposé une méthode pour l'estimation du degré de congestion routière basée sur des mesures de GPS prises d'une route urbaine. Trois niveaux de congestion ont été définis suivant les mesures de la vitesse moyenne par GPS, la perception humaine était utilisée pour définir les seuils de classification et l'évaluation des performances de leur méthode. Les auteurs ont essayé de limiter le nombre de paramètres utilisés pour définir l'état du trafic routier en utilisant seulement la vitesse moyenne.

Le travail de Xiaojun Shen et Chen [XIJ09] a fourni un modèle de prédiction de congestion routière basé sur les réseaux de neurones avec une quantification vectorielle (LVQ). Le modèle fournit une classification des situations fluides et congestionnées par l'apprentissage de LVQ en se basant sur des paramètres de trafic : volume, vitesse et taux d'occupation, fournis par deux détecteurs de véhicules placés sur deux voies d'une autoroute.

Pour surmonter les inconvénients de la poursuite des véhicules, qui dépend de la qualité de la détection de mouvement, et qui est sensible au nombre de véhicules qui se présentent dans la scène, les chercheurs se sont intéressés récemment à la prédiction de l'état du trafic routier sans passer par l'extraction des paramètres individuels (paramètres microscopiques) des véhicules, ce domaine est en cours de développement et peu d'articles y sont publiés. On présente dans ce qui suit les principaux travaux dans cet axe.

## **2.4 Méthodes basées sur le traitement global de la scène (paramètres macroscopiques)**

Fatih Porikli et Xiaokun Li [POL04] ont proposé un algorithme d'estimation de congestion routière qui opère sur des vidéos compressées en MPEG, ils ont extrait des caractéristiques de la congestion directement dans le domaine compressé sans avoir besoin de passer par la poursuite individuelle des véhicules. Ils ont employé le modèle de Markov caché avec la mixture gaussienne (GM-HMM) pour la détection de trafic. Premièrement ils ont construit un vecteur de caractéristiques multidimensionnelles formé par les coefficients DCT et les vecteurs de mouvement extraits à partir du décodage MPEG. Ils ont ensuite utilisé ce vecteur de caractéristiques dans une chaîne de Markov cachée à cinq états de trafic (vide, écoulement libre, congestion légère, forte congestion, arrêté), et le critère de maximum de vraisemblance (ML) pour déterminer l'état du trafic.

Antoni Chan et Vasconcelos [CHV05] ont proposé une modélisation du flux du trafic par un modèle holistique qui n'exige pas de phase de segmentation ou de poursuite. Ils ont adopté un modèle de texture dynamique et un processus autorégressif stochastique qui représente l'apparence et le mouvement séparément par deux distributions de probabilités. L'estimation des paramètres de ces deux densités de probabilités à partir des séquences vidéo permet la classification supervisée du trafic en se basant sur le calcul de la divergence de Kullback-Leibler. Ce système présente de bonnes performances pour la classification du trafic et une robustesse vis à vis des conditions environnementales variables telles que la luminosité et l'ombre.

Konstantinos G et Derpanis [KDW10] ont décrit un nouveau système de classification du trafic vidéo en se basant sur l'observation visuelle dynamique, dans leur travail les scènes de flux de trafic routier sont traitées comme des instances de textures dynamique, la classification est basée sur la mise en correspondance des distributions (ou histogrammes) de la structure d'orientation spatio-temporelle.

## **2.5 Conclusion**

Nous avons présenté dans ce chapitre les principaux travaux qui ont traité l'estimation du trafic routier par vision. La majorité de ces travaux utilisent des méthodes basées sur la détection et la poursuite du mouvement des véhicules, tandis que le traitement global du flux de trafic est un axe récent, en cours de développement.

Dans les chapitres suivants, nous allons décrire et implémenter quelques méthodes d'estimation de trafic routier appartenant aux deux approches.



### 3.1 Introduction

La manière la plus détaillée de considérer un flux routier est de considérer individuellement chaque véhicule comme un agent ayant ses propres caractéristiques : les modèles microscopiques associent à  $N$  mobiles se déplaçant sur la route  $N$  caractéristiques.

La détection et la poursuite du mouvement sont à la base de tout système d'estimation des paramètres microscopiques du trafic routier par vision, telle que la vitesse du véhicule, l'espace entre véhicules, ... etc.

Ce chapitre est consacré à l'estimation de quelques paramètres microscopiques dans le but d'estimer les paramètres moyens et de classifier le trafic correspondant.

### 3.2 Variables microscopiques

Pour la description du trafic au niveau granulaire, l'accent est mis sur le véhicule individuel et son interaction avec son environnement. Les variables sont donc exprimées pour chaque véhicule et sont étroitement liées à sa dynamique. L'estimation des paramètres moyens peut être ainsi obtenue par le calcul de la moyenne sur la totalité du trafic.

La vitesse est le paramètre le plus important pour la description du comportement d'un véhicule ainsi que l'état du trafic.

#### 3.2.1 Estimation de la vitesse

Généralement l'enregistrement dans le temps de la vitesse instantanée d'un véhicule le long d'un parcours sert à calculer les vitesses moyennes durant différents cycles de fonctionnement, cet enregistrement est capable d'être réalisé par le suivi du véhicule le long de sa trajectoire, le suivi requiert des informations synthétiques sur le véhicule tel que sa position, sa taille. Ce traitement intermédiaire s'appelle : « extraction des objets en mouvement » ou « détection des objets en mouvement ».

### 3.2.1.1 Détection des objets en mouvement

Cette phase consiste à détecter des objets en mouvement dans une scène filmée par une caméra fixe. L'algorithme de détection doit être robuste et capable de s'adapter aux conditions variables d'une scène routière à une autre. Il est important à cette étape d'avoir une bonne détection pour pouvoir faire le suivi et l'interprétation correctement.

Les problèmes confrontés lors du traitement des scènes routières sont liés à plusieurs facteurs :

- Le bruit dû à la variation climatique et environnementale.
- Le mouvement d'objets tel que les branches d'arbre est assez courant. Le but est de détecter ces objets comme appartenant au fond de l'image. Ce mouvement est de façon générale assez répétitif.
- Le changement de luminosité est assez courant quand on observe des véhicules sur la route, ceci peut être lié à un passage de nuage par exemple.
- Les scènes sombres ont des niveaux de couleur assez faibles, les différences entre les couleurs sont moins marquées. Il peut donc être plus difficile de détecter le fond de la forme dans ce type de scène.
- Les véhicules ayant une couleur proche du fond peuvent être plus difficilement détectés.
- Les ombres, ce sont des zones qui suivent les véhicules avec des changements de valeur de pixel par rapport au fond. Il est donc assez logique que les algorithmes puissent classer par erreur ces zones comme forme.
- La densité de circulation et le mouvement lent peuvent poser des problèmes. En effet les méthodes de détection doivent apprendre le fond, pour cela, elles considèrent que les valeurs des pixels les plus fréquentes sont des valeurs de fond. Ceci peut poser un problème dans des situations urbaines à circulation très dense où le fond n'est pas souvent visible en raison du flux dense et continu des véhicules.

#### ❖ Algorithmes de détection de mouvement

Il y a deux étapes importantes dans la détection d'objets en mouvement, l'ordre diffère d'un algorithme à un autre :

**a- La maintenance de l'arrière-plan**

Cette phase consiste à mettre à jour périodiquement une image modèle considérée comme l'arrière-plan, pour pouvoir détecter les changements qui se sont produits dans la scène par la suite. L'arrière-plan constitue la partie statique de la scène qui ne doit pas être détectée comme en mouvement par l'algorithme de détection. La partie qui doit être détectée constitue l'avant-plan.

**b- La détection des changements dans la scène**

Elle consiste à comparer l'image considérée comme arrière-plan avec une nouvelle image de la séquence pour voir si des changements se sont produits dans la scène.

La difficulté réside surtout dans la première étape, la deuxième étant généralement effectuée en faisant une soustraction entre deux images, l'image courante et celle représentant l'arrière-plan, puis en comparant la différence de chaque pixel avec un seuil.

Différentes approches ont été développées pour faire la détection des objets en mouvement, la plupart se basant sur l'étude de l'image pixel par pixel.

Une approche naïve serait de faire la différence entre l'image courante de la séquence  $F_i$ , et une image statique  $F_0$  qui représente l'arrière-plan et de vérifier si la différence est à l'intérieur d'un seuil  $Th$  dont la valeur est fixée expérimentalement :

$$\text{Si } |F_i(x,y) - F_0(x,y)| < Th$$

*Donc  $F_i(x,y)$  appartient à l'arrière plan.*

*Sinon*

*$F_i(x,y)$  appartient à l'avant plan.*

Mais cette méthode comporte de multiples problèmes liés aux :

- Changement dans l'intensité de la lumière, c'est-à-dire le changement de l'intensité des pixels de l'image de la séquence au cours du temps. Comme l'arrière-plan n'est pas mis à jour avec le changement graduel de la lumière, les résultats sont complètement erronés.
- Mouvements constants dans l'image telle que l'oscillation de feuilles d'arbres ou le mouvement de l'eau. Ces mouvements considérés comme inintéressants changent

constamment la valeur des pixels dans l'image à la même position. Donc avec cette méthode, ces mouvements sont détectés et faussent les résultats attendus.

Une autre méthode consiste à faire la soustraction de chaque paire d'images adjacentes ( $F_i, F_{i-1}$ ) de la séquence d'images :

$$\text{Si } |F_i(x,y) - F_{i-1}(x,y)| < Th$$

*Donc  $F_i(x,y)$  appartient à l'arrière-plan.*

*Sinon*

*$F_i(x,y)$  appartient à l'avant-plan.*

Encore là, cette méthode comporte des lacunes importantes. Elle ne résout pas le deuxième problème énoncé précédemment et dépend de plusieurs paramètres tels que la vitesse des objets en mouvement pour pouvoir reconnaître les pixels qui ont bougé dans l'image.

Pour résoudre les problèmes liés à ces méthodes de base concernant la maintenance de l'arrière-plan, plusieurs algorithmes ont été développés. Dans notre travail on s'intéresse à une méthode dite méthode de filtrage approximatif de médiane.

### ❖ Filtrage approximatif de médiane

N. Mcfarlane et C. Schofield ont proposé un filtre récursif simple pour l'estimation de la médiane. Cette méthode a été aussi employée pour la modélisation de l'image du fond pour des fins de surveillance du trafic urbain [CHK04].

Le filtre médian est l'un des techniques les plus couramment utilisées pour la modélisation du fond, dont le principe est le suivant :

Soit  $x_t$  l'intensité du pixel observée, et  $m_t$  l'approximation courante de la médiane,  $m_{t+1}$  sera donnée par :

$$m_{t+1} = \begin{cases} m_t + \alpha, & \text{si } x_t > m_t \\ m_t - \alpha, & \text{si } x_t < m_t \\ m_t, & x_t = m_t \end{cases} \quad (3.1)$$

Où  $\alpha$  représente la valeur du taux de convergence désirée.

Un pixel  $x_t$  est classé stationnaire s'il satisfait la condition suivante :

$$|x_t - m_t| < T \quad (3.2)$$

Où T représente le seuil de la classification des pixels.

L'avantage principal de cette méthode est qu'elle est simple à mettre en œuvre et peut traiter les changements dynamiques de la scène.

Par la suite on décrit le schéma d'implémentation. On peut initialiser la médiane avec la Première image de la séquence, ou simplement par une matrice nulle.

Les variables utilisées sont:

- $x_t(i)$  : la valeur du pixel i à l'instant t.
- $A_t$  : l'image à l'instant t.
- $m_t(i)$  : la médiane estimée du pixel i à l'instant t.
- $\alpha$  : taux de convergence.
- T : le seuil.
- B : image segmentée.

### Algorithme

```

Pour tout pixel i dans l'image At
  Si xt(i) > mt-1(i) alors mt(i) = mt-1(i) + α ;
  Si non
    Si xt(i) < mt-1(i) alors mt(i) = mt-1(i) - α ;
  Fin si ;
  Fin si ;
  Si xt(i) = mt-1(i) alors mt(i) = mt-1(i)
  Fin si ;
Fin pour ;
% Segmentation binaire
Pour tout pixel i dans l'image At
  Si |xt(i) - mt(i)| ≤ T alors B(i) = 0 ; % objet stationnaire
  Sinon B(i) = 1 ; % objet mobile
Fin pour ;

```

Les figures suivantes montrent la détection de mouvement pour des scènes de trafic fluide et dense, respectivement. Nous avons utilisé  $\alpha = 0.05$  ( $\alpha \in [0,1]$  est un niveau de gris) et  $T = 25$ , ces valeurs ont été choisies après plusieurs tests.



Fig 3.1- Images originales



Fig 3.2- Détection de mouvement

La détection de mouvement sera améliorée par l'utilisation de quelques filtres morphologiques tels que l'érosion, la dilatation, pour résoudre les problèmes liés souvent à des occlusions lorsque les véhicules sont trop proches.

### 3.2.1.2 Poursuite de mouvement

Le suivi des véhicules dans une scène routière est l'un des problèmes les plus difficiles de la vision par ordinateur (suivi d'objets multiple -SMO-). La partie la plus critique d'un système de SMO est l'appariement des objets (association de données) pendant les occlusions provoquées par des interactions entre les objets mobiles ou les structures fixes de

la scène surveillée. Dans la dernière décennie, beaucoup de recherches ont été faites pour développer des algorithmes robustes pour des scénarios de sécurité routière.

Dans notre travail, nous nous sommes intéressés par la mise au point d'un système de suivi approprié pour des applications de sécurité routière, capable de suivre plusieurs objets dans le but d'estimer leurs vitesse de mouvement et ceci sans avoir besoin de calibrer la caméra. Nous visons également un système qui soit robuste à certaines fausses détections provenant du module de détection d'objets.

Lorsque le système détecte des objets intéressants (véhicules), le système de suivi doit identifier et trouver les objets dans le champ visuel de la camera. Les deux composantes d'un système de suivi sont:

- La détection de l'objet.
- L'association des données.

Un objet peut être défini par sa forme, son apparence, ou tout ce qui est intéressant pour l'analyser d'avantage. Les objets sont suivis d'une trame à une autre par un procédé d'association de données pour cela une fonction de coût doit être définie pour associer les objets entre deux trames consécutives ou plus.

Dans ce travail, on utilise les centres de gravité comme une information représentative de chaque objet (véhicule), la méthode d'association des centres de gravités utilisée consiste à calculer un coefficient appelé coefficient de colinéarité et de corrélation [DAP00] entre trois points de l'image, ce coefficient est défini par la relation suivante :

$$R = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{(n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2)} \sqrt{(n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2)}} \quad (3.3)$$

Avec :

$n$  : nombre de trames utilisées pour la détection de linéarité.

$(x_i, y_i)$  : coordonnés de centres de gravités.

Ce coefficient est maximal lorsqu'il y a une linéarité entre les points. Dans la pratique on utilise un seuil égal à 0.95.

Puisque les véhicules circulent en une trajectoire linéaire donc ce critère est intéressant pour réaliser une poursuite au cours du temps. Nous avons amélioré cette méthode par le rajout d'un autre critère, vue les problèmes confrontés par la poursuite avec la seule condition de linéarité, ce critère consiste à estimer la distance entre deux points linéaires, cette dernière ne doit pas dépasser un certain seuil pour que les deux points peuvent être considérés comme appartenant à la même trajectoire. Ceci revient à déterminer la vitesse maximale qui peut être atteinte par un véhicule. Cette solution élimine les fausses linéarités entre les centres de gravités en excluant les variations de vitesses ou de directions trop brutales.

Donc cette phase consiste à trouver l'objet dans chaque image de la séquence et de là tracer sa trajectoire et calculer sa distance et sa vitesse.

La recherche de la meilleure trajectoire qui doit correspondre à un objet doit être effectuée sur trois images consécutives. Cette tâche doit répondre à une contrainte appelée contrainte d'unicité d'association qui provient du fait qu'un point de l'image ne peut représenter qu'un seul point physique, ce qui interdit d'associer un objet détecté à plusieurs trajectoires. Par ailleurs il n'y a pas toujours de correspondance puisque certains objets apparaissent ou disparaissent. Donc pour un point ou pour une trajectoire, on peut avoir soit une seule association ou aucune.

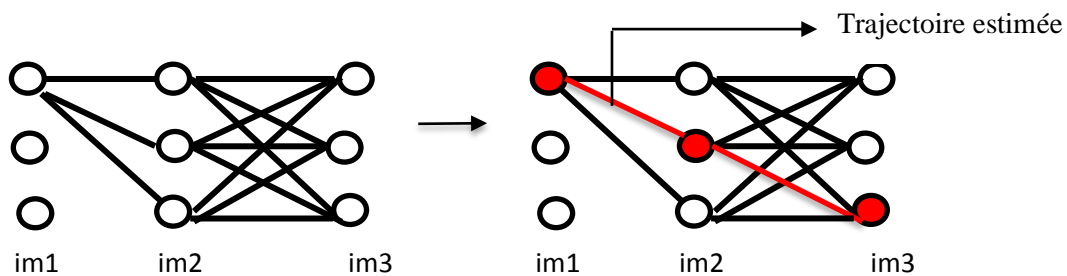


Fig 3.3- Recherche de la trajectoire d'objet sur 3 images

Les figures suivantes montrent la poursuite des mouvements de véhicules durant des instants différents et pour des scènes différentes :





Fig 3.4- Poursuite de mouvement (Scène 1)



Fig 3.5- Poursuite de mouvement (Scène 2)

Une fois la trajectoire d'un véhicule est trouvée, on peut donc calculer la distance parcourue et la vitesse de mouvement.

### 3.2.1.3 Calcul de la vitesse

La vitesse moyenne du trafic est obtenue par le moyennage des vitesses individuelles de tous les véhicules dans la scène. La vitesse individuelle d'un véhicule est calculée en divisant la distance qu'il a parcourue entre deux trames consécutives par le temps inter-trames (fréquence d'échantillonnage de la vidéo). Cette distance est estimée sans le recours à un calibrage préalable de la caméra.

Pour cela on réalise d’abord une estimation de la fonction d’échelle à partir des facteurs d’échelle individuels des véhicules présents dans une image (le facteur d’échelle d’un véhicule est estimé par le rapport entre la longueur moyenne réelle des véhicules et la longueur sur l’image en pixel du véhicule). La figure (3.6) illustre cette procédure.

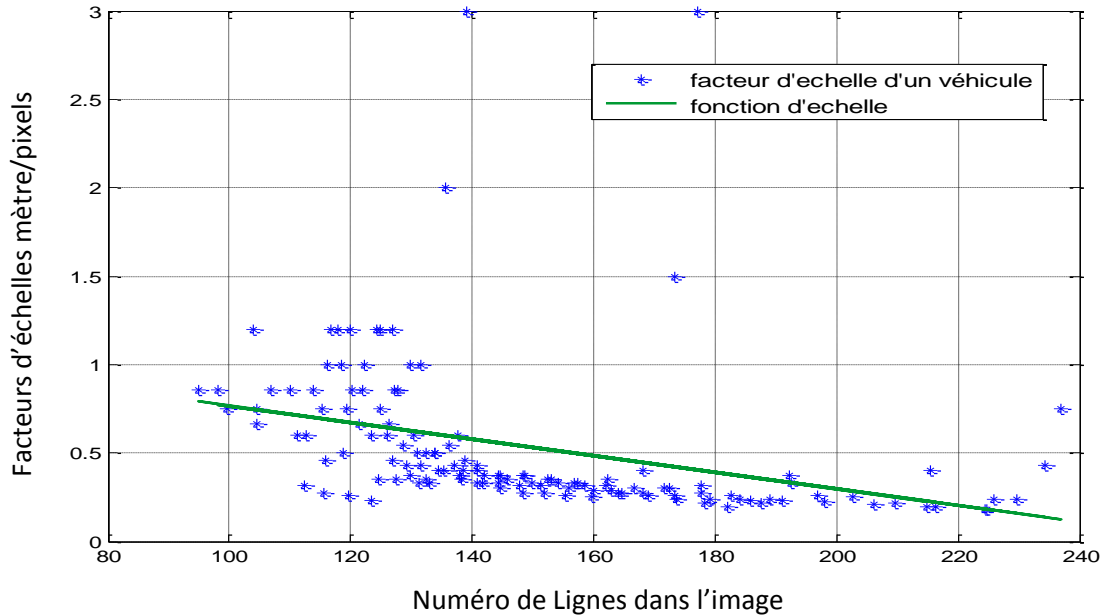


Fig 3.6- Fonction d’échelle

La ligne en vert sur la figure admet comme équation :

$$q(v)=m v + b \tag{3.4}$$

Celle-ci représente la fonction d’une droite dont les paramètres m et b sont estimés à partir du nuage de points par la méthode des moindres carrés.

La distance parcourue par le véhicule entre deux trames consécutives est alors donnée par le calcul de l’intégrale de la fonction q(v) entre les deux positions (v<sub>1</sub>, v<sub>2</sub>) du véhicule dans ces deux trames :

$$d = \int_{v_1}^{v_2} q(v)dv \tag{3.5}$$

Finalement la vitesse sera obtenue par la division de la distance parcourue par la période

d'échantillonnage de la vidéo ( $\Delta t$ ) :

$$\hat{S} = \frac{d}{\Delta t} \quad (3.6)$$

### 3.2.2 Estimation du taux d'occupation de la route

Le taux d'occupation de la route est défini par la fraction du temps durant laquelle le capteur détecte le véhicule, c'est un paramètre important pour la compréhension de l'état de la route.

Dans notre travail, nous introduisons une méthode de calcul du taux d'occupation basée sur le calcul du rapport entre la surface totale de la route et la surface des véhicules présents durant la période de l'observation de la scène.

Puisque le véhicule occupe une certaine surface de la route durant un laps de temps, une route congestionnée dont le nombre de véhicules est élevé aura un taux d'occupation plus élevé (la surface occupée par les véhicules est grande). Par contre une route fluide dont le nombre de véhicules est faible aura un taux d'occupation plus faible.

De cela apparait la nécessité d'extraire en premier lieu la région de la route, les véhicules en mouvement sont par la suite détectés par la méthode précédente.

#### 3.2.2.1 Segmentation de la région de la route

De nombreuses techniques de détection et segmentation de la route ont été proposées avec plus ou moins de succès. Elles reposent généralement sur la détection des marquages du sol en utilisant la transformée de Hough [YUJ97] par exemple, ou en effectuant une rétroprojection de l'image pour obtenir une vue d'altitude de la route et en extraire simplement la position des marquages, comme par exemple le projet de GOLD [VAC95].

Il existe de nombreuses autres approches que nous ne citerons pas ici, car nous ne prétendons pas faire une liste exhaustive des travaux de recherches menés dans ce domaine.

Pour notre travail, on emploie une méthode de détection de la route qui consiste à extraire l'extrémum régional.

### ❖ Extrémum régional

La notion d'extrémum régional [BEU90] joue un rôle important pour les images numériques. Il s'agit de «plateaux», au bord desquels on ne peut que descendre (pour les maxima régionaux), ou monter (pour les minima régionaux), strictement (Fig 3.7).

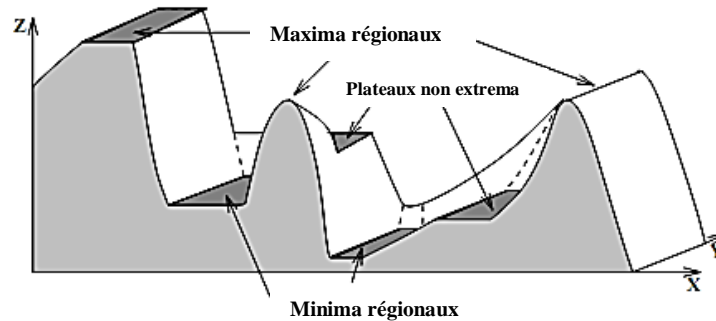


Fig 3.7- Extrema d'une image numérique

Soit  $f$  une fonction, imaginons son graphe comme une surface topographique. Un maximum de  $f$  (ou maximum régional) est un sommet de la surface topographique, c'est-à-dire une région connexe (mais pas forcément réduite à un point), d'où il n'est pas possible partant d'un point quelconque de cette région de rejoindre un point de la surface d'altitude supérieure par un chemin jamais descendant (Fig 3.8).

Ainsi le point  $x$  est un maximum, car partant de ce point, la seule façon de rejoindre un point d'altitude supérieure comme  $y$  est d'emprunter un chemin  $C_{xy}$  qui n'est pas toujours ascendant au sens large (c'est-à-dire un chemin formé par la concaténation de portions strictement ascendantes et de portions à niveau). Donc les points  $z$ ,  $y$ ,  $t$  n'appartiennent pas à des maxima.

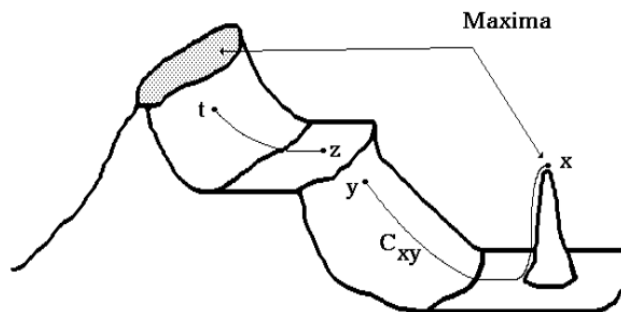


Fig 3.8- Maxima d'une fonction

### ❖ Extraction des extrema d'une image numérique

Les notions de maximum et de minimum régionaux ne sont pas locales : dans le cas général, on ne peut pas décider si un pixel  $p$  appartient à un extremum simplement en examinant les pixels voisins de  $p$ . Il faut parcourir l'ensemble du plateau contenant  $p$ . C'est pourquoi on parle généralement de maximum régional et de minimum régional.

Si l'on considère les seuils de  $f$ , un maximum d'altitude  $h$  de cette fonction sera une composante connexe du seuil  $X_h(f) = \{f(x) \geq h\}$  ne contenant aucune composante connexe de tout seuil  $X_s(f)$  où  $s > h$ . En fin un maximum régional  $M$  de  $f$  d'altitude  $h$ , satisfait :

$$\forall h' > h, X_{h'}(f) \cap M = \emptyset \quad (3.7)$$

Pour extraire les maxima régionaux de  $f$ , il suffit d'effectuer une reconstruction de  $f$  par dilatation géodésique de  $(f-1)$  sous  $f$ , et de soustraire le résultat de  $f$ .

$$\text{Max}(f) = X_1(f - \delta^\infty(f, f - 1)) \quad (3.8)$$

avec  $\delta^\infty$  représente l'opération de dilatation.

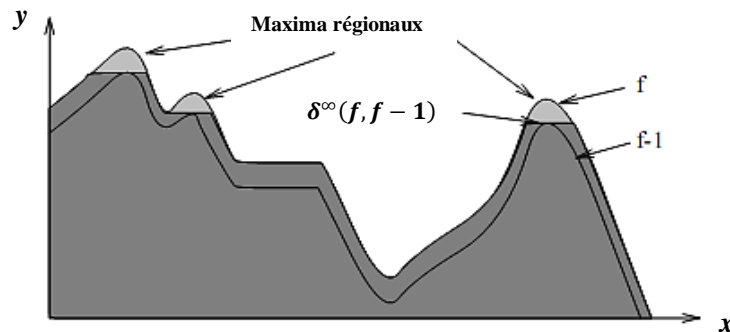


Fig 3.9- Extraction des maxima régionaux par une reconstruction géodésique

L'extraction des minima régionaux de  $f$  relève du même procédé; on peut également effectuer une reconstruction géodésique par érosion de  $(f+1)$  au dessus de  $f$ .

### ❖ Reconstruction géodésique

L'idée de reconstruction géodésique a fait son apparition en morphologie mathématique en 1967[BEU90] comme l'opération qui consiste à reconstituer les composantes connexes d'un ensemble A lorsque leur intersection avec les composantes connexes d'un second ensemble B est non vide. L'ensemble de références B est appelé généralement marqueur et l'ensemble A masque géodésique.

On définit les transformations géodésiques en considérant uniquement la partie de l'élément structurant à l'intérieur du masque.

La dilatation géodésique de taille n notée  $\delta^n(R, X)$  d'un ensemble X inclus dans un masque R est définie par :

$$\delta^1(R, X) = \delta_1(R, X) \cap R \quad (3.9)$$

$$\delta^n(R, X) = \delta_1(\underbrace{\delta_1(\dots \delta_1(X) \cap R) \cap R}_{n \text{ fois}}) \cap R \quad (3.10)$$

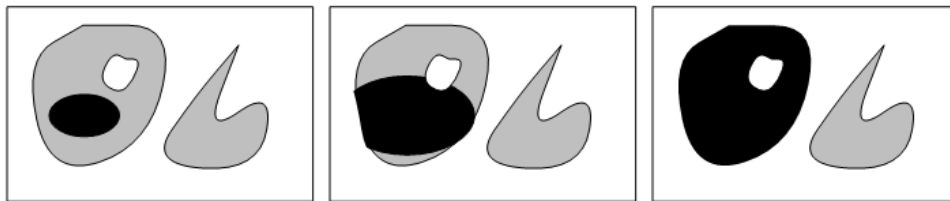


Fig 3.10- Dilatation et reconstruction géodésique

L'opération de reconstruction est réalisée par une dilatation de taille infinie (en pratique jusqu'à idempotence) à partir des marqueurs désignant en quelques sorte les objets que l'on désire préserver (Fig 3.10). Ceci permet de trouver les particules marquées dans leur intégralité.

Les figures suivantes représentent la segmentation de la route par extraction de maxima régionaux.



Image originale

Reconstruction

Maxima régionaux

Route segmentée

Fig 3.11- Extraction de la route par maxima régionaux

### 3.2.3 Distance inter -véhicules

L'espace inter véhicules est la distance séparant à un instant  $t$  donné, deux véhicules successifs circulant sur la même voie.

De nombreuses études ont tenté d'élaborer le lien entre ce paramètre et la vitesse instantanée afin de définir les distances nominales de sécurité entre des véhicules circulant en file. Certaines études ont supposé que devant un danger, le temps nécessaire à l'arrêt d'un véhicule peut être exprimé en fonction de la somme d'un temps de réaction et d'un temps de freinage. D'autres travaux exploitent cette variable pour l'estimation du temps de réaction des conducteurs.

Dans notre travail, nous exploitons cette variable pour l'estimation de l'état du trafic routier, et ceci on se basant sur l'hypothèse qu'en cas de congestion routière la distance moyenne entre les véhicules est réduite.

Pour ce faire nous opérons comme suit :

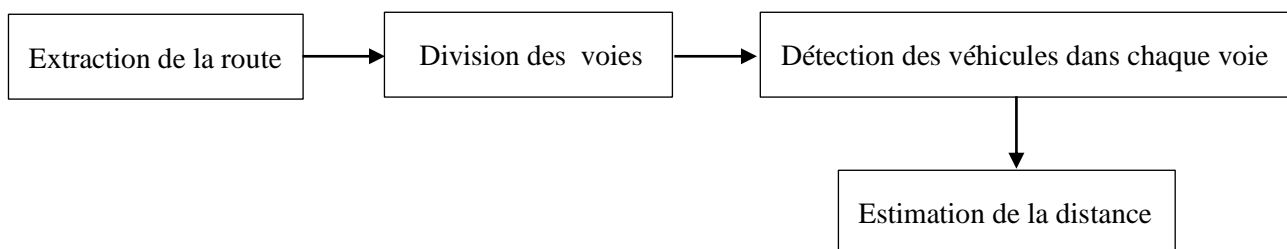


Fig 3.12- Processus d'estimation de la distance inter-véhicules

L'extraction de la région de la route est premièrement effectuée par la méthode présentée précédemment, ensuite nous divisons les voies de la route par une méthode manuelle (ce traitement se fait qu'une seule fois tant que la caméra est toujours fixée dans la même position.), les véhicules circulant en chaque voie sont par la suite détectés afin d'estimer la distance verticale entre deux véhicules successifs.

### 3.2.4 Débit moyen du trafic routier

Le débit du trafic sur une route, appelé aussi « intensité du trafic » représente la quantité moyenne du flux  $\bar{q}$  mesurée en nombre de véhicules  $N$  passés au point  $x$  de la section de route considérée, entre deux instants du temps  $t_1$  et  $t_2$ . Il permet de décrire la distribution des véhicules dans le temps.

$$\bar{q}(x, t_1, t_2) = \frac{N}{T}; T = t_2 - t_1 \quad (3.11)$$

En pratique, cette variable peut être calculée par un simple comptage des véhicules sur la section de route étudiée.

Dans ce travail on estime le débit du trafic routier par le calcul de la différence entre le débit entrant et le débit sortant (le nombre moyen des véhicules qui pénètrent dans le champ visuel de la caméra durant une période de temps définie et le nombre des véhicules qui en sortent).



### 3.3 Résultats

#### 3.3.1 Base de vidéos du trafic routier

La base de vidéos utilisée dans ce mémoire pour l'estimation des paramètres et la classification du trafic routier contient une séquence de 254 vidéos (une totalité de 20 minutes) prises par une seule caméra fixée sur une autoroute à Seattle (Washington) durant deux jours [CHV05]. La base contient une variété de modes de trafic routier avec différentes conditions climatiques (ex : Couvert, pluvieux, ensoleillé).

Chaque vidéo clip a été enregistré en couleur à 10 trames/seconde avec une résolution de 240×320 pixels et contient entre 42 et 52 trames.

La figure ci-dessous représente l'état réel du trafic routier aussi appelé vérité terrain (ground truth).

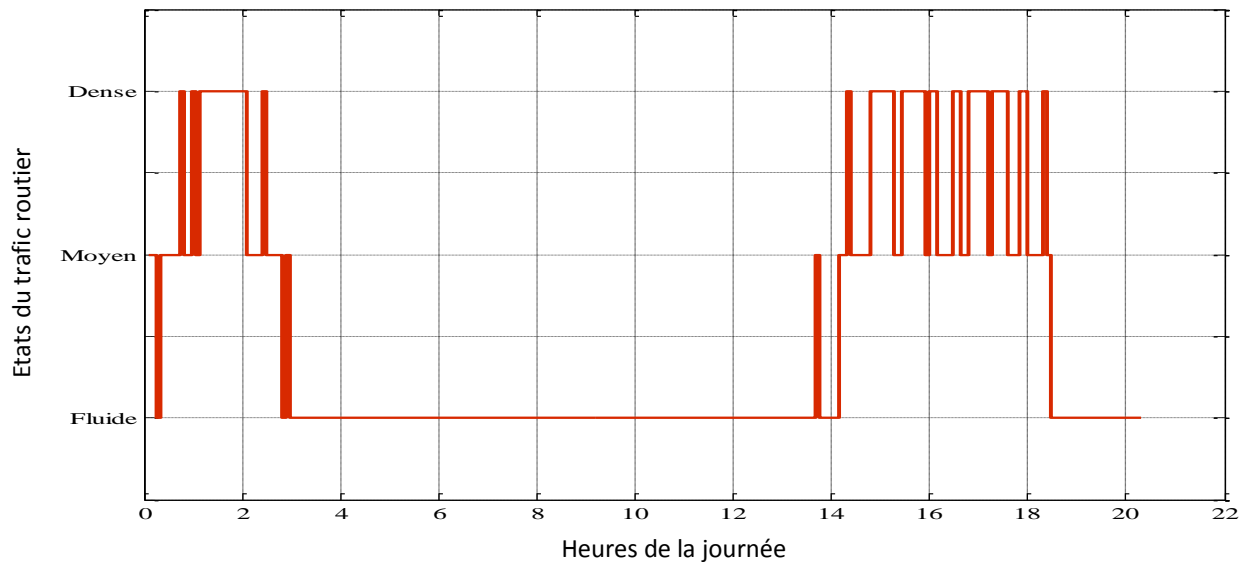


Fig 3.13- Vérité terrain de la base vidéo utilisée

On constate que les 254 vidéos correspondent à une observation de la route durant 21 heures.

De 1h jusqu'à 3h le trafic est élevé (moyen ou dense), ensuite le trafic devient fluide jusqu'à 14h. De 14h jusqu'à 18h le trafic augmente de nouveau avec plusieurs états de congestion, et ensuite il redevient fluide. On estimera les paramètres du trafic pour chaque

vidéo et on comparera les résultats obtenus avec l'état du trafic réel correspondant. Tous les traitements réalisés dans ce mémoire ont été effectués avec Matlab version 7.7.

### 3.3.2 Estimation de la vitesse

Les figures ci-dessous représentent les résultats de l'estimation de la vitesse moyenne du trafic routier pour trois types de trafic: fluide, moyen et dense.

On remarque la variation de la vitesse pour ces trois états de trafic routier.



Fig 3.14- Estimation de la vitesse moyenne

L'estimation de la vitesse moyenne pour toute la période d'observation est représentée par la figure suivante :

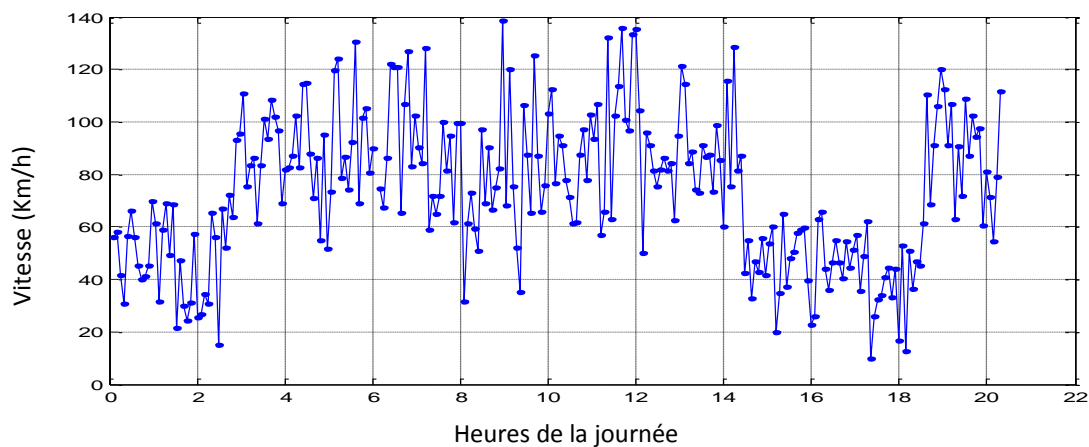


Fig 3.15- Estimation de la vitesse moyenne.

On remarque que la vitesse est faible entre 1h et 3h ce qui correspond à un trafic élevé, ensuite la vitesse augmente durant la période entre 3h et 14h étant donné que le trafic est

faible et diminue de nouveau lorsque le trafic devient dense.

On remarque qu'à 8h la vitesse est réduite pourtant le trafic est fluide ceci s'explique par le comportement des conducteurs ou par des erreurs d'estimation dues souvent à des conditions climatiques difficiles.

### 3.3.3 Estimation du taux d'occupation de la route

Les figures suivantes représentent une estimation du taux d'occupation de la route pour 3 types de trafic : fluide, moyen, dense.

Le taux d'occupation estimé dans la première image est faible (trafic fluide), tandis que dans la troisième image ce taux est élevé, correspondant à un trafic dense.



Fig 3.16- Estimation du taux d'occupation moyen

La figure suivante représente une estimation du taux d'occupation pour la totalité des scènes observées :

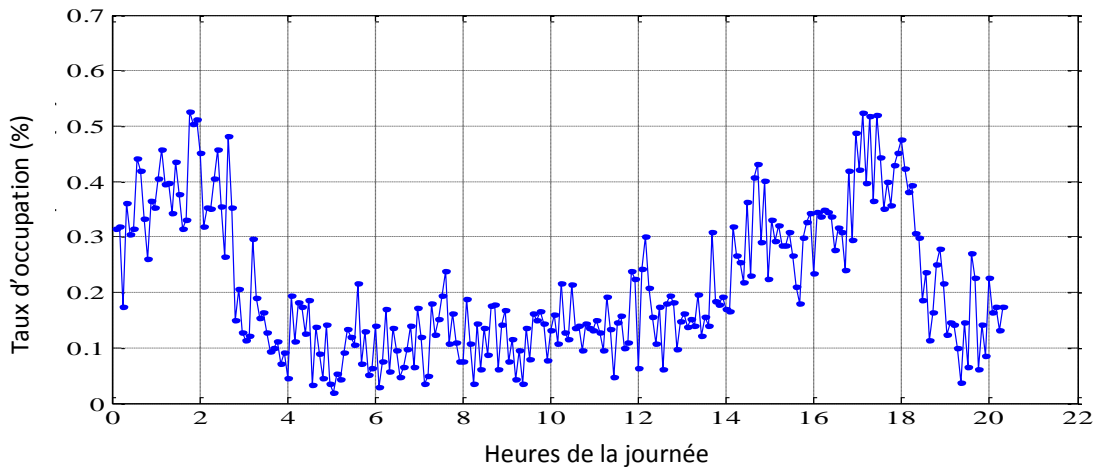


Fig 3.17- Estimation du taux d'occupation moyen

On remarque sur cette figure que le taux d’occupation estimé est élevé dans le cas d’un trafic dense, et est faible pour les situations où le trafic est fluide.

Dans certains cas le taux d’occupation estimé est élevé alors que le trafic est fluide, ceci peut correspondre à la circulation d’un nombre important de véhicules dans un régime fluide.

### 3.3.4 Estimation de la distance Inter- véhicules moyenne

Les résultats d’estimation de la distance inter-véhicules moyenne sont présentés aux figures suivantes pour trois modes de trafic routier.



Fig 3.18- Estimation de distance inter-véhicules moyenne

La figure suivante montre les résultats d’estimation de ce paramètre pour toute la période d’observation du trafic.

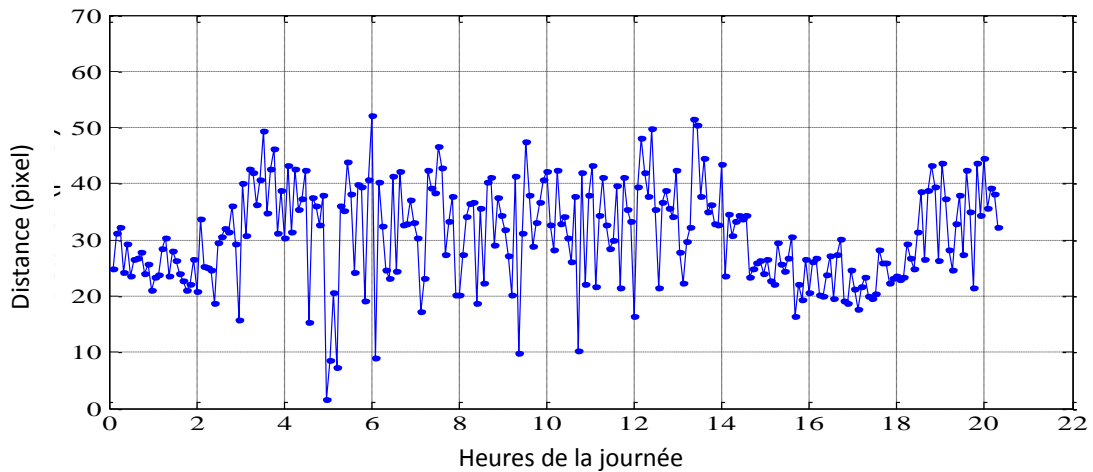


Fig 3.19- Estimation de la distance inter-véhicules moyenne

Une distance faible est estimée entre 1h et 3h ce qui correspond à la situation d'un trafic dense par contre la distance élevée correspond à un trafic fluide.

La distance inter-véhicules n'est pas toujours significative pour caractériser le trafic, car nous remarquons durant certaines périodes d'observation une diminution de cette dernière dans le cas d'un trafic fluide, ceci peut s'interpréter par la présence de quelques véhicules qui circulent trop proches les uns des autres même lorsque la route n'est pas congestionnée.

### 3.3.5 Estimation du débit du trafic

Nous appelons débit du trafic ou concentration du trafic le paramètre qui désigne la différence entre le débit entrant et le débit sortant de la route.

Les figures suivantes représentent les résultats pour les trois types de trafic.



Fig 3.20- Estimation de la concentration du trafic

La figure suivante montre les résultats pour la totalité des scènes :

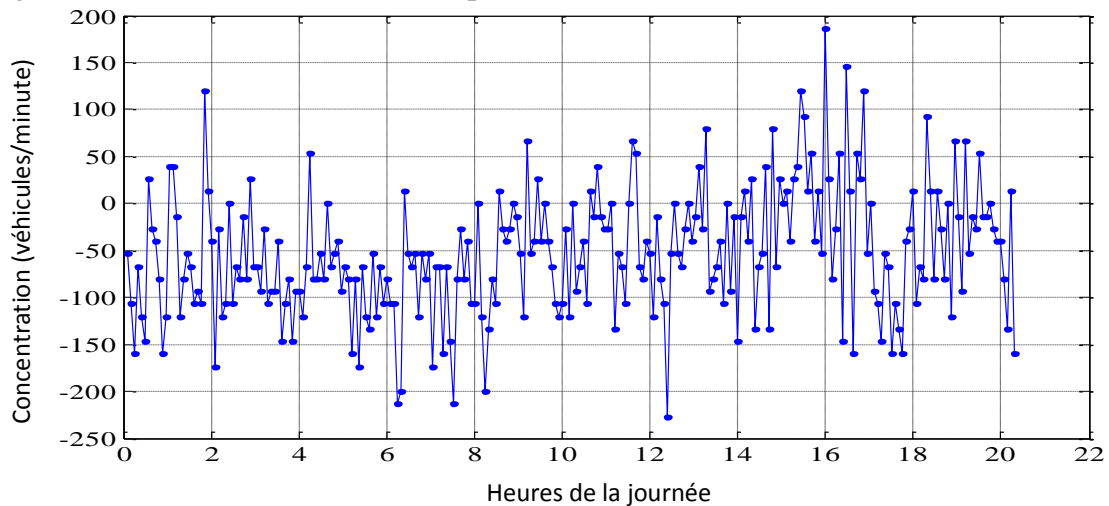


Fig 3.21- Estimation de la concentration du trafic

Dans le cas d'un trafic routier fluide le nombre de véhicules qui quittent le champ visuel de la caméra, est supérieur au nombre de véhicules qui pénètrent dans le champ visuel (débit entrant < débit sortant). L'inverse se produit dans le cas d'un trafic congestionné. Ceci est confirmé par les résultats présents aux figures ci-dessus.

La période d'enregistrement de chaque scène vidéo était très courte (5 secondes), ce qui nous a posé des difficultés lors de l'estimation du débit moyen du trafic, pour avoir une estimation plus précise on propose d'observer la route durant une période plus longue.

### **3.4 Conclusion**

Dans ce chapitre nous avons présenté quelques méthodes d'estimation des principaux paramètres du trafic routier par détection et poursuite du mouvement.

Les problèmes rencontrés lors du traitement individuel des objets imposent l'utilisation d'autres méthodes qui n'exigent aucune détection ou poursuite de mouvement et qu'on peut qualifier de globales. Le chapitre suivant représentera quelques unes de ces méthodes.

## 4.1 Introduction

A grande échelle, un flux routier peut être assimilé à un liquide s'écoulant dans un canal ou un tuyau. La représentation macroscopique du trafic emprunte donc les notions de densité et de vitesse propre à la mécanique des fluides. Dans ce chapitre nous allons estimer ces deux variables par l'utilisation d'une méthode basée sur l'estimation des vecteurs de vitesse du champ de flux optique par un algorithme de mise en correspondance des régions d'images.

Une autre méthode qui sera présentée dans ce chapitre consiste à modéliser le flux routier par un modèle appelé modèle de texture dynamique.

## 4.2 Estimation du mouvement

L'estimation de mouvement est un procédé qui consiste à étudier le déplacement des objets dans une séquence vidéo, en cherchant l'origine des objets qui se trouvent dans une image cible dans une image de référence (passée ou future), et d'extraire des vecteurs de mouvement représentant le déplacement de ces objets en se basant sur une mesure de ressemblance. On essaie donc de trouver un objet dans la référence tel que cette mesure soit minimale (Fig 4.1).

Plusieurs techniques basées sur l'estimation de mouvement sont utilisées, mais la plus efficace est celle de mise en correspondance des blocs (Block Matching) qui va être détaillée dans ce chapitre.

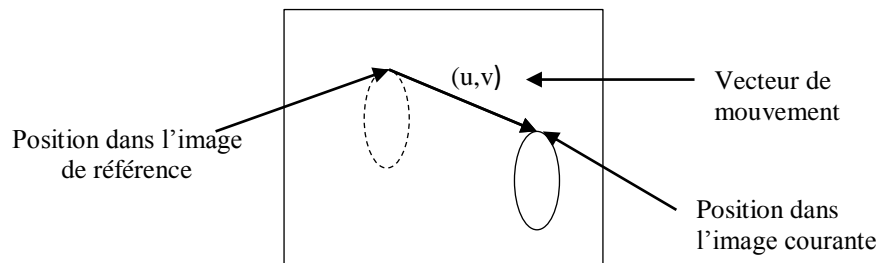


Fig 4.1- Notion de vecteurs de mouvement

L'objet primordial à utiliser est le pixel lui-même. La recherche d'un vecteur de mouvement pour chaque pixel, est une tâche épuisante et ne prend pas en compte les caractéristiques des images composées d'objets et de régions. La méthode optimale est de reconnaître les objets de la scène après une étape de reconnaissance d'objets; mais ceci nécessite un temps de calcul important et présente certains problèmes tels que l'ambiguïté, le chevauchement, la déformation... etc.

La méthode la plus efficace est de subdiviser l'image en blocs de taille égale (généralement de 8x8 ou 16x16 pixels, voir Figure 4.2) et de considérer chaque bloc comme étant un objet indépendant [LEJ08].

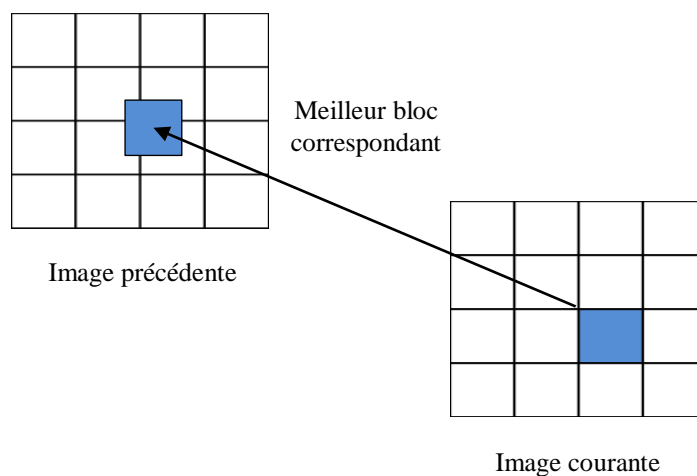


Fig 4.2- Estimation du mouvement d'un bloc

### 4.3 Mise en correspondance de blocs d'images

Les méthodes de mise en correspondance par blocs sont basées sur l'hypothèse selon laquelle l'intensité lumineuse de pixels est constante ou faiblement variable le long des trajectoires de mouvement [LEJ08]. Pour chaque point de l'image courante à l'instant  $t$  on cherche le point d'intensité la plus proche dans l'image cible (instant  $t-1$  ou  $t+1$ ). Pour éviter les correspondances incohérentes dans l'image cible, on limite la recherche à une fenêtre centrée sur le point courant  $(x,y)$ . Ainsi, le déplacement d'un pixel de coordonnées  $(x,y)$ , au centre d'un bloc  $B$  de dimension  $B_x \times B_y$ , de l'image courante est déterminé par la position du bloc de même dimension, le plus proche au sens d'un critère de similarité dans



l'image cible (*instants  $t-1$  ou  $t+1$* ). La recherche est limitée à une fenêtre de dimensions  $F=(2.d_{x\_max}+1)*(2.d_{y\_max}+1)$ , où  $d_{x\_max}$  (resp.  $d_{y\_max}$ ) est la valeur maximale du déplacement à estimer dans la direction  $x$  (resp.  $y$ ).

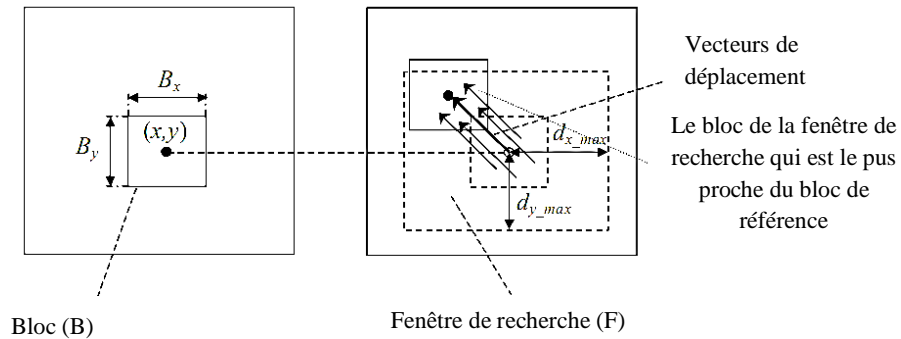


Fig 4.3- Principe de la mise en correspondance par bloc

Plusieurs paramètres influencent sur la mise en correspondance de blocs:

### 4.3.1 Largeur de la fenêtre de recherche

La largeur de la fenêtre de recherche affecte la vitesse et la précision du traitement ; dans la plupart des cas, c'est un paramètre déterminé à partir des expériences ou imposé par les contraintes des implémentations.

Cependant, les séquences de mouvement rapide, où le déplacement des objets d'une image à l'autre suivant est considérable, doivent faire l'objet d'une recherche dans une zone couvrant ce déplacement (donc une fenêtre un peu large), engendrant bien sûr un temps de calcul plus long. Par contre, les séquences de mouvement lent doivent réduire la zone de recherche pour réduire le temps de recherche. On peut également choisir une zone de taille fixe mais dont la position est déterminée en se basant sur estimation de la vitesse de déplacement des objets.

### 4.3.2 Taille des blocs

La taille du bloc choisie affecte clairement la recherche et la comparaison des blocs, et ainsi la performance de l'estimation de mouvement. En choisissant une grande taille des blocs (32,64,...), on réduit la précision des vecteurs du mouvement obtenus du fait que les grands blocs contiennent des objets de mouvements différents en vitesse et en direction.

Le choix d'une taille réduite (2,4,...) permet d'explicitier le mouvement des objets, ainsi les limites des objets seront bien identifiées et permet en outre de produire des vecteurs de mouvement très précis. D'un autre côté, les petits blocs souffrent du problème du temps de calcul important.

Les tailles de blocs, 8x8 et de 16x16 sont considérées généralement adéquates pour diverses applications.

### 4.3.3 Fonction de ressemblance

Pour trouver dans une image  $t$  le bloc correspondant à un bloc dans une image  $t-1$  ou  $t+1$  on utilise une mesure objective de ressemblance entre les valeurs des pixels contenus dans les deux blocs.

D'une manière générale, la plupart des articles traitant de l'estimation de mouvement considèrent que les blocs sont caractérisés par la luminance. Ceci est dû au fait que l'œil humain est plus sensible à l'intensité lumineuse qu'à la couleur. Cependant, tenir compte des chrominances (couleurs) peut s'avérer intéressant. Le critère le plus couramment utilisé est le SDA (somme des différences absolue).

#### ❖ SDA (Somme des différences absolue)

Le SDA est calculé par la somme des valeurs absolues des différences d'intensité des pixels de l'image originale et celle de référence. Le SDA peut être pris comme une fonction de coût à minimiser lors de l'estimation du mouvement, son expression est définie par la formule suivante :

$$SAD = \sum_{i=1}^N \sum_{j=1}^N |X(i, j) - X'(i, j)| \quad (4.1)$$

où  $X(i, j)$  est la luminance du pixel  $(i, j)$  du bloc courant, et  $X'(i, j)$  est la luminance du pixel  $(i, j)$  du bloc de référence.

#### 4.3.4 Complexité des algorithmes de mise en correspondance

L'optimisation des algorithmes de mise en correspondance a été largement étudiée au cours des deux dernières décennies. La complexité (le coût) d'un algorithme de mise en correspondance peut être mesurée grâce à la fonction :

$$Coût = N_{test} \times N_b \times C_p + C_{ctrl} \quad (4.2)$$

où:

- $N_{test}$  est le nombre de blocs testés ou vérifiés. Il est relatif à la largeur de la zone de recherche et à la stratégie de candidature des blocs utilisés dans cette zone.
- $N_b$  est le nombre de pixels du bloc comparés.
- $C_p$  est le coût de mesure de la correspondance entre deux pixels (pixel de référence et pixel cible) en nombre d'opérations de calcul (multiplications, additions, décalages, valeurs absolues,...). Il est relatif à la fonction de comparaison choisie.
- $C_{ctrl}$  est le coût de la décision en fonction des résultats obtenus de la comparaison.

Plusieurs algorithmes ont été développés visant la réduction des opérations de calcul en sacrifiant un peu de précision des vecteurs de mouvement. La plupart de ces algorithmes se basent sur la réduction de l'un (ou plus) des paramètres déjà cités.

La figure (4.5) présente une estimation des vecteurs de déplacement entre les blocs d'images pour des instants différents :



Fig 4.5- Estimation des vecteurs de déplacement par la mise en correspondance des blocs

Nous avons utilisé une fenêtre de recherche de taille  $(2R \times 2R)$  pour des blocs de taille  $(R \times R)$ , avec  $R=16$  pixels. Cette méthode permet d'obtenir un champ de vecteurs de vitesse de bonne cohérence temporelle : les mêmes configurations de vecteurs se retrouvent au long des trajectoires des objets.

La longueur des vecteurs de mouvement correspond à la distance de déplacement du bloc, les points rouge sur l'image représentent des vecteurs de mouvement de longueur nulle.

Une des limitations de la méthode de mise en correspondance de blocs est le temps de calcul, qui augmente avec le carré de l'amplitude maximale de mouvement. En effet la fenêtre de recherche peut être assimilée à un cercle centré sur le pixel central et de rayon égal à l'amplitude maximale du mouvement.

Pour surmonter cette limite nous proposons d'utiliser une nouvelle méthode qui ne requiert pas l'estimation du champ de vecteurs de déplacement et qui se base sur la modélisation du flux de trafic par un modèle de textures dynamiques.

## 4.4 Les textures dynamiques

### 4.4.1 Un domaine de recherche récent et dynamique

Les textures dynamiques représentent un sujet d'étude de plus en plus d'actualité, le nombre de publications principales sur ce thème est en forte augmentation depuis quelques années (par publications principales, nous entendons les revues ou les congrès de bonne notoriété, portant clairement sur les textures dynamiques et leur caractérisation).

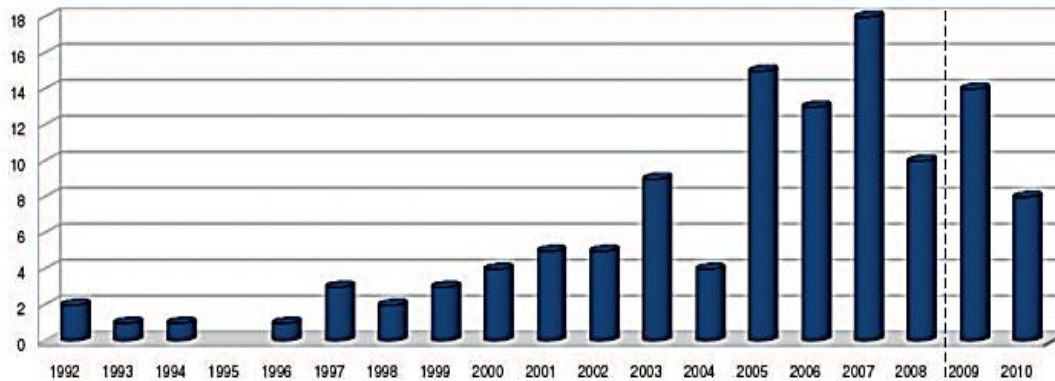


Fig.4.6- Principales publications sur les textures dynamiques depuis 1992 jusqu'au 2010[DUB11]

Les premières publications dans ce domaine sont apparues en 1992 avec les articles de Nelson Et Polana [NEP92], dans lesquelles les auteurs parlent de textures temporelles, et commencent à définir un premier cadre formel d'étude. Par la suite, jusqu'en 2001, quelques publications ont évoqué le sujet et proposé des méthodes pour les caractériser. C'est surtout à partir de l'année 2002 que le nombre de publications a augmenté. La baisse relative sur les dernières années du nombre de publications est due au délai de latence du référencement.

L'émergence de ce contexte de recherche peut s'expliquer, d'une part, par la démocratisation du support vidéo et des performances des ordinateurs pouvant les traiter, et d'autre part, par un vaste champ applicatif:

- **Indexation vidéo:** dans des bases de séquences d'images, l'objectif est de formuler des requêtes de plus en plus élaborées, associant des caractéristiques de nature sémantique, par exemple, rechercher des vidéos contenant une mer agitée (la notion "agitée" est une caractéristique temporelle), ou un feu, un lac calme, une forêt très peu agitée...

- **Vidéosurveillance** : dans certaines séquences, la texture dynamique est une caractéristique importante de la scène. Détecter un accident ou un comportement à risque dans un trafic routier (congestion routière), surveiller et caractériser le mouvement d'une foule, détecter des départs de feu de forêt ou de la fumée, sont des exemples où une description robuste de la texture dynamique est nécessaire.
- **Segmentation spatio-temporelle de séquences d'images**: dans le cadre de l'analyse vidéo, La segmentation spatio-temporelle de la séquence, relativement aux textures dynamiques, permettra d'enrichir notre compréhension de la scène et d'optimiser la segmentation en plages caractéristiques. Elle peut conduire à la détection de perturbation au sein d'une texture dynamique (présence d'un véhicule au milieu d'une forêt, d'un canard sur un plan d'eau), à aider à la construction de résumés vidéo (présence à un instant  $t$  d'une texture dynamique donnée).
- **soustraction de fonds dynamiques**: dans différentes séquences, le fond de la scène peut présenter une texture dynamique. Le soustraire permet d'améliorer l'efficacité des algorithmes.
- **suivi**: suivre des textures dynamiques dans une séquence d'images peut permettre par exemple d'analyser l'évolution de phénomènes comme le déplacement d'un feu, l'écoulement d'un fluide,...etc.

Les figures suivantes représentent quelques exemples de textures dynamiques :



Fig 4.7- Exemple de textures dynamiques

#### 4.4.2 Discussion sur les textures dynamiques

En parcourant la littérature, nous constatons que les phénomènes complexes que nous nommons textures dynamiques sont identifiés sous différents termes selon les auteurs: *temporal textures*, *spatio-temporal textures*, *dynamic textures*, *time-varying textures*, *moving textures*.

Ces différentes dénominations sont bien sûr dues à la diversité des définitions sous-jacentes proposées par les auteurs. Dans la littérature, quatre familles de définitions peuvent être identifiées.

La première définition rencontrée est apparue en 1992 avec l'article de Nelson et Polana [NEP92]. Dans celui-ci, les auteurs catégorisent les événements d'une séquence d'images en trois grandes classes: les activités, les évènements et les textures temporelles.

Les activités se définissent comme des évènements périodiques dans le temps et localisés spatialement (une personne qui marche ou saute sur place, un oiseau qui vole,...). Les évènements se décrivent comme une singularité spatio-temporelle ponctuelle ne présentant pas de périodicité temporelle ou spatiale (l'ouverture d'une porte, la chute d'un objet,...). Dès qu'un évènement présente une périodicité temporelle, celui-ci devient une activité. Enfin, les textures temporelles répondent à la définition suivante:

**Définition 1 :** *Les textures dynamiques sont des séquences d'images de scènes naturelles ayant des propriétés spatiales et temporelles stationnaires [NEP92].*

Cette définition met l'accent, d'une part, sur la notion de répétitivité de la texture dynamique, à la fois spatiale et temporelle, et d'autre part, sur la notion de scènes naturelles. Ces dernières regroupent l'ensemble des processus issus de la nature et présentant certaines caractéristiques stationnaires.

En 2001, une nouvelle définition de la texture dynamique est apparue :

**Définition 2 :** *Les textures dynamiques correspondent à des scènes naturelles en mouvement dans une séquence d'images, et qui présentent des propriétés stationnaires dans le temps [DUB11].*

Cette définition est similaire à la précédente, sauf qu'elle considère que l'essence même d'une texture dynamique doit posséder une caractéristique temporelle. Cette définition ne met pas l'accent sur les propriétés spatiales, il s'agit juste de scènes naturelles en mouvement.

La définition suivante, qui est apparue dans la littérature en 2002, lie les deux précédentes définitions puisqu'elle n'impose pas que la texture dynamique soit uniquement temporelle ou simultanément spatiale et temporelle:

**Définition 3 :** *Les textures dynamiques correspondent à des scènes de séquences d'images qui présentent certaines structures spatiales ou temporelles périodiques [DUB11].*

Cette définition est plus générale que les deux précédentes car elle ne se restreint plus aux scènes naturelles. En effet, elle met en avant les structures spatiales ou temporelles que l'on peut rencontrer aussi bien dans les scènes naturelles (processus issus de la nature: forêt, lac, mer,...), les scènes artificielles (fabriquées par l'être humain: escalator, moulin,...) et les scènes synthétiques (créées par l'ordinateur: jeux vidéos,...). Elle prend également en compte les scènes où des textures statiques (purement spatiales) peuvent induire une texture dynamique par le simple fait d'une acquisition effectuée en mouvement (travelling d'une caméra par exemple).

**Définition 4 :** *Les textures dynamiques correspondent à des scènes naturelles de séquences d'images en mouvement de nature stochastique [DUB11].*

Cette définition met l'accent sur les mouvements de la texture. Selon cette approche, une texture dynamique est une scène naturelle non statique, dans laquelle un mouvement caractéristique est présent (un mouvement d'ondulation par exemple) mais dont la localisation spatio-temporelle, dans la séquence d'images, est stochastique.

Cette définition permet d'identifier comme textures dynamiques les déplacements de fluides (vagues sur la mer,...) et celles présentant des caractéristiques proches (surface d'un lac, les drapeaux, les champs d'herbe ondulant au vent,...). Toutefois, les textures dynamiques comme celles induites par le mouvement d'une roue d'un moulin, d'un trafic routier,...ne correspondent pas à cette définition, puisque le mouvement est ici localisé dans le temps et



l'espace. Cependant cette définition a le mérite de classer un certain nombre de textures dynamiques que certains auteurs appellent textures dynamiques fortes [DUB11].

### 4.4.3 Taxonomie de textures dynamiques

Une séquence d'images peut contenir des composantes de textures statiques et/ou dynamiques. Par exemple, le pont, les cailloux et le lierre sont des éléments texturés statiques alors que l'eau et l'arbre sont en mouvement ce qui induit une texture dynamique.

- La composante texture dynamique contient au moins une texture dynamique. Par exemple, une composante texture vidéo est constituée de deux textures dynamiques distinctes: l'arbre et le ruisseau. Il serait préférable de parler d'arbres en mouvement et d'écoulement d'un ruisseau, mais pour alléger l'écriture, il sera fait référence uniquement aux éléments source de la texture dynamique. Ces différentes textures dynamiques peuvent bien sûr interférer. Dans le cadre d'une séquence naturelle d'un feu, les oscillations des flammes, les volutes de fumée en avant plan de l'oscillation d'un champ d'herbe interfèrent pour former une composante texture dynamique complexe.

- Quand on observe de nombreuses scènes, nous constatons que les textures dynamiques ne sont pas toutes de même nature: artificielles (créées par l'être humain), naturelles (issues de la nature), et synthétiques (produites par un ordinateur). Une texture dynamique se caractérise par trois facteurs:

- Une structure texturée (rigide ou déformable).
- Un mouvement induit par une force que subit la structure texturée, ou par le déplacement de la caméra: ce mouvement peut être d'apparence déterministe ou stochastique. Celui-ci peut être de plusieurs natures: force interne (moteur d'un escalator ou d'une voiture), force externe (le vent dans les ailes d'un moulin) ou induite par la caméra (translation de la caméra).
- Des changements de condition d'acquisition (éclairage, illumination,...). Ces modifications induisent une modification apparente de la texture et donc l'apparition d'une texture dynamique.

Les ailes d'un moulin présentant un mouvement de rotation, les marches d'un escalator défilant, le trafic routier, sont des exemples de textures dynamiques issues d'une structure de

texture rigide possédant un mouvement déterministe. Par contre, un banc de poisson, une colonie de fourmis, sont également des structures texturées rigides mais qui ont un mouvement stochastique.

Les textures dynamiques constituées de structures texturées déformables et présentant des mouvements stochastiques sont par exemple une cascade d'eau avec de multiples remous, par contre, un arbre ou des fleurs oscillants sous un vent constant, une cascade sans remous, sont des exemples de textures dynamiques issues de structures déformables et présentant un mouvement déterministe. Toutefois, selon l'échelle d'observation, certains de ces phénomènes peuvent être vus comme des structures rigides animées par un mouvement déterministe.

















		Objets texturés			
		Rigides		Déformables	
Mouvement	Déterministe				
	Stochastique				
					
					

Fig.4.8- Taxonomie de textures dynamiques

L'observation de certaines textures dynamiques permet de constater qu'elles peuvent être la composition de plusieurs phénomènes. Sur un exemple d'une vidéo de la surface de la mer, on observe l'écume (mouvement haute fréquence) portée par les vagues (mouvement basse fréquence). Ces composantes appartiennent à la même texture dynamique et peuvent être définies comme étant des modes. Ces modes peuvent se superposer.

L'ensemble de ces réflexions ont conduit Sloven Dubois [DUB11] à la plus récente définition:

**Définition 5 :** *Une texture dynamique est une structure texturée qui peut être rigide ou déformable. Cette structure possède un mouvement induit par une force qu'elle subit ou par le déplacement de la caméra. Ce mouvement, qu'il soit issu d'une force interne, d'une force externe ou induit par la caméra, peut être déterministe ou stochastique.*

*Les textures dynamiques sont constituées de modes, pouvant se superposer, caractérisés par des phénomènes spatiaux et temporels répétitifs.*

Cette définition possède l'avantage de décrire plus profondément la notion de texture dynamique.

Après ce tour d'horizon sur les textures dynamiques, on peut conclure qu'un phénomène de trafic routier peut être considéré comme une texture dynamique répondant bien à la dernière définition.

#### 4.4.4 Modélisation de textures dynamiques

Les textures dynamiques relèvent de phénomènes complexes. Ainsi, de nombreux auteurs ont cherché à les modéliser afin d'utiliser des paramètres du modèle comme descripteurs. Dans les approches par modèle, nous distinguons cinq grandes familles:

- Les modèles basés sur des lois physiques, on peut citer par exemple les travaux de Kung et Richards [DUB11]. Nous n'aborderons pas en détail ce type de modèle car il ne présente pas d'intérêt dans le contexte de notre travail.
- Les modèles auto-régressifs spatio-temporels;
- les modèles auto-régressifs à moyenne mobile et les approches dérivées;
- les modèles utilisant la réponse impulsionnelle des variables d'états cachés;
- les autres modèles.

Modéliser une séquence d'images de textures dynamiques en utilisant un modèle autorégressif spatio-temporel (STAR) a été proposé en premier temps par Szummer et Picard [DUB11]. Il s'agit d'une extension 3D des modèles autorégressifs (AR) utilisés pour la reconnaissance et la synthèse de textures 2D. La luminance de chaque voxel est estimée à

l'aide d'une combinaison linéaire de la luminance des voxels choisie dans un voisinage spatio-temporel, à laquelle se superpose un bruit. Les paramètres du modèle auto-régressif spatio-temporel (STAR) sont déterminés en minimisant l'erreur quadratique moyenne de prédiction.

Les modèles auto-régressifs spatio-temporels ont permis de modéliser efficacement les textures dynamiques, comme la fumée, les rivières,...etc, présentant des mouvements de translation. Toutefois, ils échouent dans la capture des mouvements rotationnels, d'accélération, et requièrent un temps de calcul long dû au nombre important de paramètres du modèle.

Afin de pallier aux difficultés rencontrées dans le modèle de Szummer et Picard, Doretto et autres [DOR05] ont proposé une série d'études utilisant un autre modèle appelé : Linear Dynamic System (LDS), celui-ci opère sur un modèle de type autorégressif à moyenne mobile (ARMA). L'apprentissage de ce modèle est réalisé selon deux méthodes: celle du maximum de vraisemblance et celle de la minimisation de l'erreur de la variance.

Un processus autorégressif stochastique (plus précisément un système linéaire dynamique) comprend une variable d'état cachée et une variable d'observation qui détermine la composante d'apparence.

#### **4.4.5 Modèle de texture dynamique**

Une vidéo numérique peut être interprétée comme un signal multidimensionnel. Le signal est une représentation visuelle d'un système durant une période de temps. Un modèle autorégressif d'un signal visuel consiste en un modèle d'apparence et un modèle dynamique. Les composantes d'apparence déterminent l'état du système, et les composantes dynamiques déterminent le changement des états durant une période de temps. Ces deux composantes permettent de caractériser le mouvement global d'une scène vidéo, c'est pour cette raison qu'on adopte un modèle de texture dynamique dans notre travail.

Doretto et autres [DOR05] ont incorporé le modèle d'apparence et le modèle dynamique pour représenter la texture dynamique comme la sortie d'un processus autorégressif de

premier ordre :

$$x_{t+1} = Ax_t + Bv_t \quad (4.3)$$

$$y_t = Cx_t + w_t \quad (4.4)$$

avec  $x_t \in R^n$  est une séquence de vecteur d'état de dimension n.

$y_t \in R^m$  est une séquence de m trames de la vidéo.

$$A \in R^{n \times n}, B \in R^{n \times n}, C \in R^{m \times n}$$

$Bv_t \sim N(0, Q)$  est un bruit de processus, et  $w_t \sim N(0, R)$  est un bruit d'observation.

Selon ce modèle, la séquence d'images,  $Y = [y_1, y_2, \dots, y_\tau]$  est définie en fonction des variables aléatoires d'états cachés,  $X = [x_1, x_2, \dots, x_\tau]$ .

Le modèle d'apparence est défini par l'équation (4.4). C représente la matrice de projection entre l'espace d'état (dynamique) et l'espace d'observation (d'apparence), les trames  $y_t$  sont une combinaison linéaire des composantes principales de la séquence vidéo entière, stockées dans les colonnes de C.

Le modèle dynamique est représenté par l'équation (4.3) ce modèle contient une composante déterministe et une composante stochastique. La composante déterministe est représentée par un processus de Markov de premier ordre où l'état actuel est une combinaison linéaire des éléments de l'état précédent. La matrice A définit la matrice de transition d'un état à un autre. La composante stochastique est définie par une distribution gaussienne avec une covariance donnée par :

$$Q = BB^T \text{ et } v_t \sim N(0, I) \quad (4.5)$$

La covariance du bruit d'observation est supposée être égale à  $R = \sigma^2 I$ .

#### 4.4.5.1 Estimation des paramètres

En général les paramètres d'un processus autorégressif peuvent être estimés par la méthode de maximum de vraisemblance [DOR05], mais cette solution n'est pas adéquate pour le modèle de texture dynamique, vu la grande dimension de l'espace d'image observé.

Une alternative sous-optimale, mais faisable consiste à estimer les paramètres spatiaux et temporels séparément [CHV05].

Soit  $Y_1^N = [y_1, \dots, y_N] \in \mathbb{R}^{m \times N}$  la matrice formée à partir de la séquence de trames de la vidéo observée, et soit sa décomposition en valeurs singuliers (SVD) donnée par :

$$Y_1^N = U \Sigma V^T \quad (4.6)$$

où les colonnes de  $U$  représentent une estimation des composantes principales et  $\Sigma V^T$  une estimation des vecteurs d'états. On a alors :

$$\hat{C} = U \quad (4.7)$$

$$\hat{X}_1^N = \Sigma V^T \quad (4.8)$$

où :  $\hat{X}_1^N = [\hat{x}_1, \dots, \hat{x}_N]$  est la matrice d'états estimés. Étant donnée l'estimation d'états, la matrice de transition est ainsi donnée par :

$$\hat{A} = \hat{X}_2^N (\hat{X}_1^{N-1})^\# \quad (4.9)$$

Où :  $M^\# = M^T (M M^T)^{-1}$  est la pseudo-inverse de  $M$ .

Finalement l'estimation de la matrice de covariance du bruit de processus  $Bv_t$  est donnée par:

$$\hat{Q} = \frac{1}{N-1} \sum_{i=1}^{N-1} \hat{v}_i \hat{v}_i^T, \quad (4.10)$$

avec :

$$\hat{v}_t = \hat{x}_{t+1} - \hat{A} \hat{x}_t \quad (4.11)$$

#### 4.4.6 Synthèse de textures dynamiques

Le problème de la synthèse de textures consiste à générer une image texturée se conformant à un modèle, à des mesures, ou bien à un exemple. Juger la qualité d'une texture synthétisée est essentiellement un problème subjectif qui touche à la perception humaine. Ce problème complexe est cependant simple à mettre en pratique. Dans le cas où l'on dispose d'un modèle, on peut par exemple superposer un échantillon de la texture

synthétisée sur la texture d'origine. La synthèse est un succès si l'œil humain n'arrive pas à retrouver la trace de la nouvelle texture (Fig 4.10).



Fig 4.9- Image originale



Fig 4.10- Synthèse de texture dynamique

#### 4.4.7 Comparaison entre deux textures dynamiques

Puisque une texture dynamique peut être modélisée dans un espace d'états et un espace d'images, deux textures dynamiques peuvent être comparées dans l'un de ces deux espaces. Si on opte par exemple pour l'espace d'états la distribution de probabilités dans le premier espace d'états est premièrement projetée dans le second espace d'état, ensuite on effectue une mesure de similarité en calculant une distance appelée, distance de Kullback-Leibler (KL) entre les deux distributions de probabilités [CHV05]. Dans ce travail nous nous intéressons à la comparaison entre textures dynamiques uniquement dans un espace d'états.

#### 4.4.7.1 Distribution des probabilités

Un état de texture dynamique peut être représenté par un processus de Markov-gaussien, où la probabilité conditionnelle de l'état  $x_t$  sachant l'état  $x_{t-1}$  est gaussienne :

$$p(x_t / x_{t-1}) = G(x_t, Ax_{t-1}, Q), \quad (4.12)$$

avec 
$$G(x, \mu, Q) = ((2\pi)^n |Q|)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \|x - \mu\|_Q^2\right) \quad (4.13)$$

et 
$$\|x\|_Q^2 = x^T Q^{-1} x$$

En substituant récursivement (4.12) dans (4.3), on obtient la densité de probabilité d'un état individuel :

$$p(x_t) = G(x_t, \mu_t, S_t), \quad (4.14)$$

où  $\mu_t$  est la moyenne de  $x_t$  donnée par :

$$\mu_t = A^t x_0 \quad (4.15)$$

et  $S_t$  est sa covariance donné par :

$$S_t = \sum_{i=0}^{t-1} A^i Q (A^i)^T \quad (4.16)$$

#### 4.4.7.2 Projection entre espaces d'états

La distance KL, dans l'espace d'état, entre deux textures dynamiques  $x_1$  et  $x_2$  ne peut pas être calculée en utilisant directement leurs paramètres, car chacune utilise un espace de composantes principales différent. Pour pouvoir donc comparer deux textures dynamiques en utilisant la distance KL, il faut au préalable projeter l'état  $x_1$  de la première dans l'espace d'état de la seconde en utilisant la transformation suivante :

$$\hat{x}_1 = C_2^T C_1 x_1 \quad (4.17)$$

Si l'état  $x_1$  a comme paramètres  $(A_1, B_1, x_{01})$ , alors sa projection  $\hat{x}_1$  aura les paramètres donnés par :

$$\hat{A}_1 = (C_2^T C_1) A_1 (C_2^T C_1)^{-1} \quad (4.18)$$



$$\hat{B}_1 = (C_2^T C_1) B_1 \quad (4.19)$$

$$\hat{x}_{01} = (C_2^T C_1) x_{01} \quad (4.20)$$

La distance KL peut être maintenant calculée entre  $\hat{x}_1$  et  $x_2$ .

#### 4.4.7.3 Calcul de la distance KL entre espaces d'états

La divergence KL entre processus aléatoire avec les distributions de probabilités,  $p(X)$  et  $q(X)$  où  $X=(x_1, x_2, \dots)$ , est définie par :

$$D(p(X) \parallel q(X)) = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} D(p(x_1^\tau) \parallel q(x_1^\tau)) \quad (4.21)$$

Où  $D(p \parallel q) = \int_x \log \frac{p(x)}{q(x)} dx$  représente la distance KL entre deux fonctions de densité de probabilité  $p(X)$  et  $q(X)$ .

Si  $p(x_1^\tau)$  et  $q(x_1^\tau)$  sont les densités de probabilité de deux textures dynamiques avec les paramètres  $(A_1, Q_1, x_{01})$  et  $(A_2, Q_2, x_{02})$ , la distance KL définie en (4.21) sera donnée par la forme suivante [CHV05] :

$$\begin{aligned} \frac{1}{\tau} D(p(x_1^\tau) \parallel q(x_1^\tau)) = & \frac{1}{2} \left[ \log \frac{|Q_2|}{|Q_1|} + \text{tr}(Q_2^{-1} Q_1) - n + \frac{1}{\tau} \|A_1 x_{01} - A_2 x_{02}\|_{Q_2}^2 \right. \\ & \left. + \frac{1}{\tau} \sum_{i=2}^{\tau} \text{tr}(\bar{A}^T Q_2^{-1} \bar{A} (S_{i-1} + \mu_{i-1} \mu_{i-1}^T)) \right], \quad (4.22) \end{aligned}$$

où  $\bar{A} = A_1 - A_2$ , et  $S_{i-1}$  et  $\mu_{i-1}$  sont respectivement la covariance et la moyenne associées à l'état  $x_{i-1}$  de la première texture dynamique.

## 4.5 Résultats

Nous considérons les mêmes scènes vidéo de trafic routier décrites dans le chapitre précédent, dont la vérité terrain est reproduite ci-après :

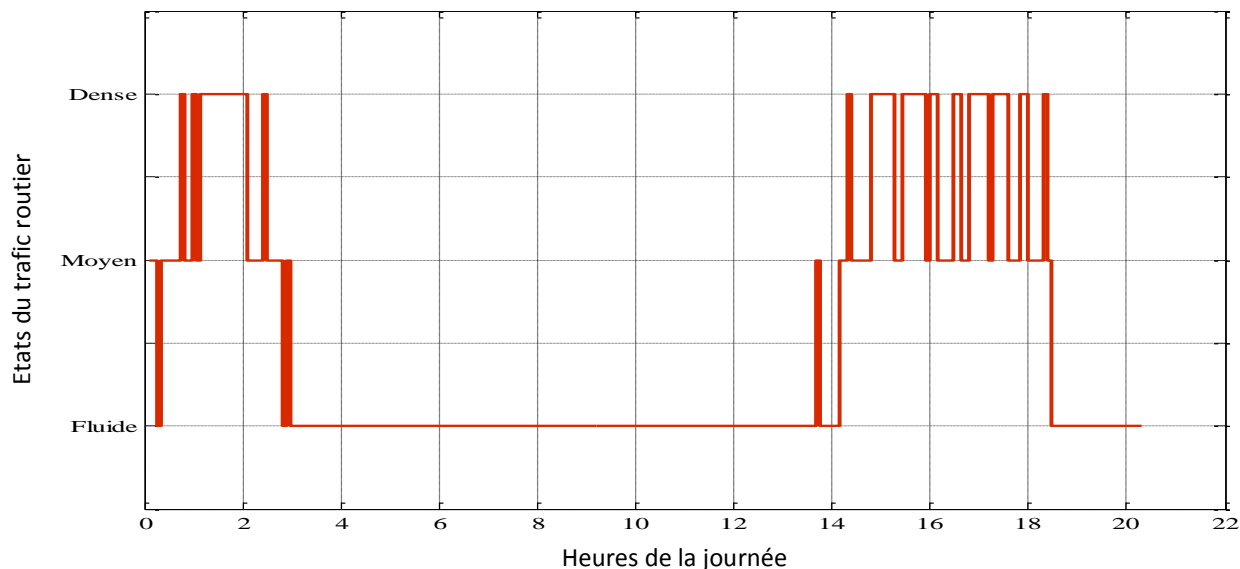


Fig 4.11 vérité terrain

### 4.5.1 Estimation de la vitesse par mise en correspondance

La vitesse moyenne du mouvement est calculée en fonction de la taille des vecteurs (distance de mouvement) et le temps inter-frames, la figure (4.12) montre les variations de cette vitesse pour trois scènes différentes: fluide, moyenne, dense.



Fig 4.12- Estimation de la vitesse de mouvement par mise en correspondance de blocs

La figure ci-dessous représente les résultats d'estimation de la vitesse moyenne du trafic routier pour la totalité des scènes observées.

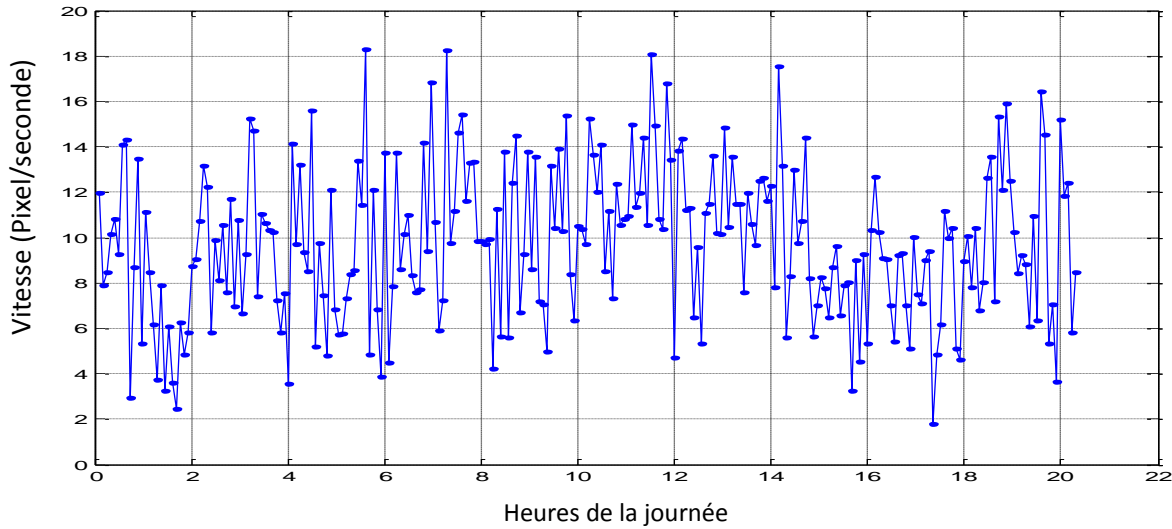


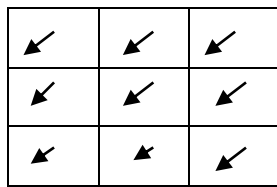
Fig 4.13- Estimation de la vitesse de mouvement par mise en correspondance de blocs

On remarque une diminution de la vitesse moyenne du trafic routier entre 1h et 3h, ce qui correspond à un trafic dense, de 3h à 15h on estime une vitesse élevée correspondant à un trafic fluide, la vitesse diminue ensuite entre 15h et 18h lorsque le trafic redevient dense.

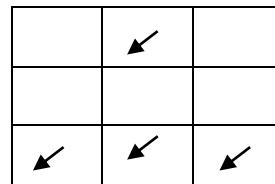
### 4.5.2 Densité des vecteurs de mouvement

Afin de profiter des informations fournies par l'estimation des vecteurs de mouvement, nous avons effectué une estimation de la densité du trafic routier par une estimation de la densité des vecteurs de mouvement non nuls.

Un trafic faible conduit à une densité estimée des vecteurs de mouvement faible, tandis que dans le cas d'un trafic élevé on estime une forte densité des vecteurs de mouvement non nuls (Fig 4.14).



(a) Forte densité des vecteurs de mouvement



(b) Faible densité des vecteurs de mouvement

Fig 4.14- Densité de vecteurs de mouvement

La figure ci-dessous présente une estimation de ce paramètre pour les trois modes de trafic.



Fig 4.15- Densité de vecteurs de mouvement

La figure suivante représente une estimation de la densité de mouvement pour la totalité de la période d'observation du trafic routier.

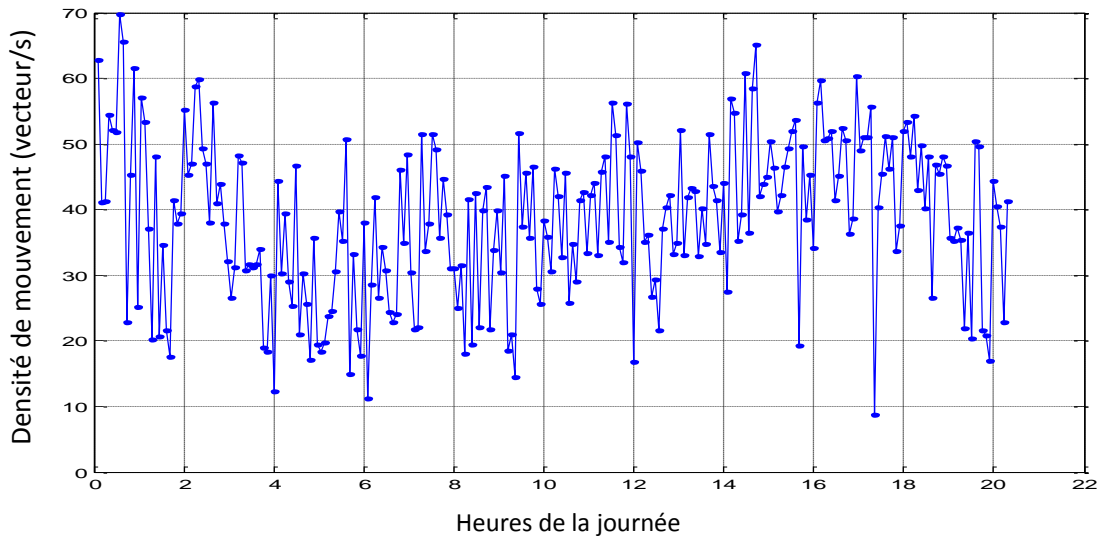


Fig 4.16- Estimation de la densité du trafic routier

L'augmentation de la densité des vecteurs de mouvement entre 1h et 3h se traduit par la présence d'un grand nombre de véhicules en mouvement ce qui peut correspondre à un trafic dense. La diminution de la densité des vecteurs de mouvement entre 3.30h et 14h peut être interprétée par la faible présence de véhicules en mouvement ce qui peut correspondre à un trafic fluide, entre 14h et 19h le trafic redevient dense et la densité des vecteurs de mouvement augmente.

Le paramètre de densité des vecteurs de mouvement doit être combiné avec d'autres paramètres tels que la vitesse pour mener à une meilleure estimation de la qualité du trafic routier, ceci représentera le but du chapitre suivant.

## **4.6 Conclusion**

Dans ce chapitre nous avons présenté deux méthodes d'estimation des paramètres de trafic routier directes sans avoir recours à la détection et la poursuite de mouvement.

L'estimation des vecteurs de déplacement par la méthode de mise en correspondance de blocs nous a permis d'extraire deux paramètres macroscopiques du trafic : la vitesse moyenne et la densité du trafic routier.

Dans le chapitre suivant nous allons présenter quelques méthodes de classification du trafic routier à base des paramètres précédents et à base de la mesure de similarité entre les textures dynamiques.

## 5.1 Introduction

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains traits descriptifs. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classification correspondante. Un système d'apprentissage doit alors, à partir de cet ensemble d'exemples extraire une procédure de classification, il s'agit en effet d'extraire une règle générale à partir des données observées. La procédure générée devra classer correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classer correctement de nouvelles descriptions.

Dans ce chapitre on présentera quelques méthodes de classification qu'on utilisera pour classer le trafic routier à savoir : le plus proche voisin (NN), réseaux de neurone à quantification vectorielle (LVQ), et le séparateur à vaste marge (SVM).

## 5.2 Les k plus proches voisins: classifieur « paresseux »

La méthode des plus proches voisins (PPV) (Nearest Neighbour (NN)) est une méthode dédiée à la classification, elle appartient à la famille des algorithmes dits « paresseux » : à l'inverse de beaucoup d'autres méthodes (telles que les réseaux de neurones, les méthodes à noyaux, ...etc) aucun apprentissage ne prend réellement place, c'est-à-dire qu'il n'y a pas de phase consistant en la construction d'un modèle à partir d'un échantillon d'apprentissage. Le principe est le suivant :

Etant donnée une base d'apprentissage d'images étiquetées correctement et un entier  $k$ , le classifieur K-PPV détermine la classe d'un nouvel objet en lui attribuant la classe majoritaire des  $k$  objets lui ressemblant le plus dans la base d'apprentissage. Les figures suivantes illustrent le principe de fonctionnement de ce classifieur [WIK12] .

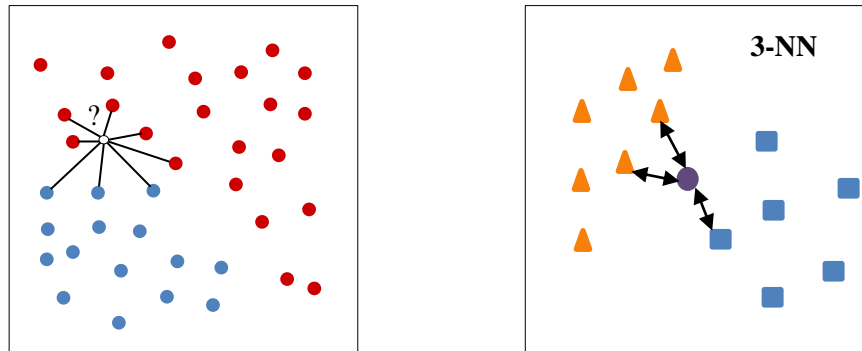


Fig 5.1- Principe de KNN

On note deux aspects importants de l'algorithme k-ppv : d'une part, à chaque nouvelle classification il est nécessaire de parcourir l'ensemble de la base d'apprentissage, ce qui on fait un algorithme qui n'est pas nécessairement très efficace (surtout qu'on vise à avoir la base d'apprentissage la plus grande possible afin d'avoir un meilleur classifieur). D'autre part, un point crucial de cet algorithme est la fonction distance utilisée pour mesurer l'approximité des objets. Il n'existe pas de distance (similarité) universellement optimale, et une bonne connaissance du problème à traiter guide généralement le choix de cette distance.

Considérant une base de données d'échantillons dont la classe est connue, la généralisation s'effectue en trouvant les  $k$  plus proches voisins de l'instance à classifier. Suivant l'identification des  $k$  voisins, un vote majoritaire sur la classe est tenu. Ainsi, si  $k=3$ , admettant que deux voisins appartiennent à la classe 1 et qu'un seul voisin appartient à la classe 2, l'échantillon sera classifié comme appartenant à la classe 1. Le nombre de voisins est donc un paramètre qui peut être optimisé. Lorsqu'un nombre de voisins égal au nombre de classes est sélectionné, il peut arriver que l'algorithme ne converge pas vers aucune des classes. Un choix basé sur la distance doit être fait dans ce cas, ce qui revient à reformuler le problème avec une valeur de  $k=1$ . Notons qu'il est aussi possible de considérer le type de distance comme un paramètre de l'algorithme. Toutefois, la distance euclidienne est généralement utilisée, dans notre travail nous avons utilisé la distance de Kulback-leibler entre deux modèles de textures dynamiques.

### 5.2.1 Algorithme de classification par KNN

L'algorithme générique de classification d'un nouvel exemple par la méthode PPV est le suivant :

**Paramètre** : le nombre  $k$  de voisins

**Donnée** : Un échantillon de  $m$  enregistrements classés  $(x, c(x))$

**Entrée** : un enregistrement  $y$

1. Déterminer les  $k$  plus proches enregistrements de  $y$  ;
2. Déterminer la classe  $c$  majoritaire dans ces  $k$  exemples ;

**Sortie** : la classe de  $y$  est  $c(y) = c$

### 5.2.2 Critères de la méthode

- **Pas d'apprentissage**

L'introduction de nouveaux exemples permet d'améliorer la qualité de la méthode sans avoir recours à la reconstruction d'un modèle. C'est une différence majeure avec des méthodes telles que les réseaux de neurones.

- **Clarté des résultats**

Bien que la méthode ne produise pas de règles explicites, la classe attribuée à un exemple peut être expliquée en exhibant les plus proches voisins qui ont amené à ce choix.

- **Temps de classification**

La méthode ne nécessite pas d'apprentissage, et donc tous les calculs doivent être effectués lors de la classification.

- **Stocker le modèle**

Le modèle est l'échantillon, il faut donc un espace mémoire important pour le stocker ainsi que des méthodes d'accès rapides pour accélérer les calculs.

- **Distance et nombre de voisins**

Les performances de la méthode dépendent du choix de la distance, du nombre de voisins et du mode de combinaison des réponses des voisins.



### 5.2.3 Exemple de classification par KNN

On considère que l'on dispose d'une base d'éléments dont on connaît la classe. On parle alors de base d'apprentissage. Dès que l'on reçoit un nouvel élément que l'on souhaite classifier, on calcule sa distance à tous les éléments de la base. Si cette base comporte 100 éléments, alors on calcule 100 distances et on obtient donc 100 nombres réels. Si  $K = 25$  par exemple, on cherche alors les 25 plus petits nombres parmi ces 100 nombres. Ces 25 nombres correspondent donc aux 25 éléments de la base qui sont les plus proches de l'élément que l'on souhaite classifier. On décide d'attribuer à l'élément à classifier la classe majoritaire parmi ces 25 éléments.

## 5.3 Réseau de neurones LVQ

Les réseaux de neurones artificiels sont devenus en quelques années des outils précieux dans des domaines très divers, notamment dans les domaines de reconnaissance de formes et de classification. Ces dernières années on a remarqué une tendance vers l'adoption de ces modèles pour des fins de prédiction de la congestion du trafic routier.

Les réseaux de neurones artificiels simulent certaines caractéristiques du cerveau humain, et ils sont constitués d'éléments simples qui sont inspirés des neurones biologiques. Grâce à leur traitement parallèle de l'information et à leurs mécanismes, ils infèrent des propriétés émergentes permettant de solutionner des problèmes jadis qualifiés comme complexes.

Le réseau avec apprentissage par quantification vectorielle (LVQ) est un modèle de réseaux de neurones qui permet d'entraîner les réseaux de neurones compétitifs de manière supervisée. Par rapport à de nombreux autres réseaux de neurones, ce modèle est le plus intéressant pour la classification du trafic routier [XIJ09].

Par la suite nous présentons l'apprentissage compétitif qui est à la base de réseau LVQ.

### 5.3.1 L'apprentissage compétitif

L'apprentissage compétitif, comme son nom l'indique, consiste à faire compétitionner les neurones d'un réseau [PAR06] pour déterminer lequel sera actif à un instant donné. Contrairement aux autres types d'apprentissage où généralement, tous les neurones peuvent

apprendre simultanément et de la même manière, l'apprentissage compétitif produit un « vainqueur ». Le neurone dont le vecteur poids est le plus proche (au terme d'une mesure de similarité comme la distance euclidienne ou le produit scalaire) d'un vecteur d'entrée est désigné comme vainqueur d'une compétition, et seul ce vainqueur bénéficie d'une adaptation de ses poids.

Une règle d'apprentissage compétitif comporte les éléments suivants :

- Un ensemble de neurones identiques (même types) sauf pour les valeurs de leurs poids synaptiques.
- Une limite imposée à la « force » d'un neurone.
- Un mécanisme permettant aux neurones de compétitionner pour le droit de répondre à un certain sous-ensemble des vecteurs d'entrée, de manière à ce qu'un seul neurone de sortie soit actif à la fois.

Ainsi, les neurones individuels peuvent apprendre à se spécialiser sur des sous-ensembles de vecteurs similaires pour devenir des détecteurs de caractéristiques.

Dans leur forme la plus simple, les réseaux de neurones qui utilisent l'apprentissage compétitif sont souvent constitués d'une seule couche de neurones de sortie, totalement connectée sur les entrées. Un neurone vainqueur modifiera ses poids synaptiques en les rapprochant (géométriquement) d'un vecteur  $P$  pour lequel il a battu tous les autres neurones lors de la compétition :

$$\Delta w = \begin{cases} \eta(p - w) & \text{si le neurone est vainqueur} \\ 0 & \text{autrement} \end{cases} \quad (5.1)$$

Où  $0 < \eta < 1$  correspond à un taux d'apprentissage.

Un neurone qui ne gagne pas la compétition ne modifiera aucunement ses poids. Il ne sera donc pas affecté par le vecteur en question.

L'apprentissage compétitif est surtout utilisé dans le contexte d'un apprentissage dit non-supervisé, c'est-à-dire lorsqu'on ne connaît pas les valeurs désirées pour les sorties du

réseau. On retrouve pour les réseaux compétitifs différentes méthodes d'apprentissage, la plus courante étant la méthode LVQ.

La méthode LVQ est une méthode d'apprentissage supervisé pour catégoriser les vecteurs. La méthode VQ utilise une méthode d'apprentissage non supervisé comme nous le verrons par la suite.

### **5.3.2 VQ Vector Quantization (quantification vectorielle)**

Introduite par Grossberg (1976) [PAR06], la quantification vectorielle est une méthode généralement qualifiée d'estimateur de densité non supervisé (apprentissage non supervisé). Elle désigne le fait de remplacer un vecteur dont les composantes peuvent être quelconques, par un vecteur d'un ensemble discret. Cette opération est analogue à la quantification d'un signal échantillonné: la valeur des échantillons constitue dans ce cas une variable mono-dimensionnelle.

La quantification vectorielle met en correspondance un ensemble de vecteurs d'entrée avec un ensemble de vecteurs de sortie. Tous les échantillons d'une même classe sont représentés par un seul code.

Cette méthode de classification a été très utilisée récemment, mais aujourd'hui on lui préfère la méthode LVQ car celle-ci emploie une technique d'apprentissage supervisé.

### **5.3.3 LVQ (Learning Vector Quantization)**

Les réseaux utilisant la méthode LVQ (Learning Vector Quantization) ont été proposés par Kohonen (1988) [PAR06]. C'est une version supervisée de l'algorithme de quantification vectorielle (VQ). Comme dans toutes les méthodes supervisées, pendant l'apprentissage, on utilise l'information d'appartenance des exemples d'apprentissage à une classe.

Cette méthode est réellement adaptée à la classification de données par "recherche du plus proche voisin".

### 5.3.3.1 Structure de réseaux de neurones LVQ

Un LVQ comprend généralement trois couches : Couche d'entrée, couche de compétition, couche de sortie. Le schéma d'un réseau de neurones LVQ est donné à la figure 5.2.

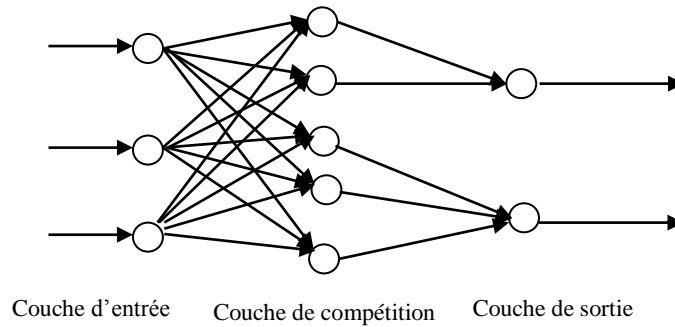


Fig.5.2- Structure de réseaux de neurone LVQ

La couche d'entrée et la couche de compétition sont complètement connectées, tandis que la couche de compétition et la couche de sortie sont partiellement connectées. Les éléments du vecteur résultant sont des valeurs binaires (0,1).

Les données sont appliquées à la couche d'entrée, dont les unités sont connectées à tous les neurones de la couche de compétition par des poids de connexion  $W_{ij}$  (poids de connexion de l'unité  $j$  de la couche d'entrée à l'unité  $i$  de la couche de compétition). Chaque réponse de neurone représente la distance entre le vecteur de poids et le vecteur d'entrée.

La couche de sortie activera un seul neurone, qui est appelé neurone gagnant. Le neurone gagnant est sélectionné comme le neurone ayant la plus petite distance entre un vecteur d'entrée et son vecteur de poids de connexion. Le neurone gagnant est autorisé à avoir une sortie égale à 1 et les autres neurones sont assignés à la sortie 0.

La modification des poids  $w_{ij}$  du neurone gagnant d'indice  $i$ , et de classe  $C_i$  est donnée par l'équation (5.2).

$$\Delta w_{ij} = lr. \delta. [A_j - w_{ij}] \quad \text{où } \delta = \begin{cases} +1, & \text{si la classe } C_i = T \\ -1, & \text{sinon} \end{cases} \quad (5.2)$$

où  $A_j$  est une composante  $j$  du vecteur d'entrée, et  $T$  sa classe et  $lr$  représente une constante d'apprentissage qui est inférieure à 1.

Les poids du neurone gagnant sont modifiés de façon à le rapprocher du vecteur d'entrée s'il appartient à sa classe désignée, où bien à l'en éloigner dans le cas contraire. Par conséquent la variable  $\delta$  prendra ici la valeur +1 ou -1 selon le cas.

### 5.3.3.2 Algorithme de classification par LVQ

De ce que nous avons vu précédemment, l'algorithme de LVQ peut être résumé comme suit :

**Initialisation** : initialiser les poids de connexion  $w$  de chaque neurone et le taux d'apprentissage de réseaux  $\eta$  ( $0 < \eta < 1$ ), et  $t=0$ . Les poids  $w$  sont initialisés aléatoirement.

Tant que les conditions d'arrêt ne sont pas remplies, exécuter :

- 1- Pour chaque vecteur  $x$  d'apprentissage : (les étapes de 2 à 5)
- 2- **Compétition** : recherche du neurone gagnant  $j^*$  dans le vecteur de sortie pour chaque vecteur d'entrée en utilisant l'équation suivante :

$$\|W_{j^*} - x\| = \min \|W_j - x\|$$

Avec,  $x$  vecteur d'entrée,  $W_j$  vecteur de poids du neurone  $j$ . La sortie du neurone gagnant  $j^*$  est mise à 1, celle des autres neurones est mise à 0.

- 3- On suppose que  $x$  appartient à la classe  $r$  et,  $j^*$  appartient à la classe  $s$ , alors :

$$\begin{cases} W_{j^*}(t+1) = W_{j^*}(t) + \eta(t)[x(t) - W_{j^*}(t)] & s = r \\ W_{j^*}(t+1) = W_{j^*}(t) - \eta(t)[x(t) - W_{j^*}(t)] & s \neq r \\ W_{j^*}(t+1) = W_{j^*}(t+1) & j \neq j^* \end{cases}$$

$$t = t+1$$

où  $t$  représente le numéro d'itération d'apprentissage.

- 4- Ajuster le taux d'apprentissage : si  $s=r$ ,  $\eta_{j^*} = \eta_{j^*} / (1 + \eta_{j^*})$ ;

$$\text{si } s \neq r, \eta_{j^*} = \eta_{j^*} / (1 - \eta_{j^*}) ;$$

- 5- Tester les conditions d'arrêt. Cela peut être l'atteinte d'un nombre d'itération donné ou le taux d'apprentissage atteignant une valeur suffisamment petite.

## 5.4 Les Machines à Vecteurs de Supports (SVM)

Les SVMs (Support Vector Machines) sont de nouvelles techniques d'apprentissage statistique, initiées par V. Vapnick en 1995. Elles permettent d'aborder des problèmes très divers, dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très hautes dimensions telles que les textes et les images. Depuis leur introduction dans le domaine de la reconnaissance de formes, plusieurs travaux ont pu montrer l'efficacité de ces techniques principalement en traitement d'images.

Cette technique est une méthode de classification à deux classes qui tente de séparer linéairement les exemples positifs des exemples négatifs dans l'ensemble des exemples [SCS01].

Chaque exemple doit être représenté par un vecteur de dimension  $n$ . La méthode cherche alors l'hyperplan qui sépare les deux groupes d'exemples, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Intuitivement, cela garantit un bon niveau de généralisation car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être tout de même situés franchement d'un côté ou l'autre de la frontière.

L'efficacité des SVM est très bonne pour la reconnaissance de formes. Un autre intérêt est la sélection de vecteurs de supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas. Cela en fait une méthode très rapide.

Dans cette partie nous présentons les aspects théoriques de la méthode SVM.

### 5.4.1 Principe de SVM

Le principe de SVM est de créer un hyperplan qui sépare les données en deux classes (classification binaire), en utilisant le principe de *marge maximale*: considérons des points d'apprentissage appartenant aux classes  $-1$  et  $+1$ , l'hyperplan maximise la marge qui sépare les deux classes des points d'apprentissage de telle façon que la distance entre les points les

plus proches de la marge de l'hyperplan soit maximisée. La première version de l'algorithme a été proposée par Vladimir Vapnik en 1963. Elle s'appliquait aux séparateurs linéaires (on parle de classes linéairement séparables). Puis, en 1992, Bernhard Boser, Isabelle Guyon et Vladimir Vapnik ont proposé la création d'un séparateur non linéaire en appliquant le « kernel trick » (méthode à noyau) au principe de la marge maximale [SCS01]. L'algorithme de base n'est pas modifié, seul le produit scalaire classique est remplacé par une fonction noyau non linéaire. Le principe consiste à projeter les données de l'espace d'entrée (appartenant à deux classes différentes) non linéairement séparables dans un espace de plus grande dimension appelé espace de caractéristiques (features space) de façon à ce que les données deviennent linéairement séparables. Dans cet espace, on construit un hyperplan optimal séparant les classes tel que :

- Les vecteurs appartenant aux différentes classes se trouvent dans différents côtés de l'hyperplan.
- La plus petite distance entre les vecteurs et l'hyperplan (la marge) soit maximale.

Le principe de la technique SVM est représenté dans la figure ci-dessous :

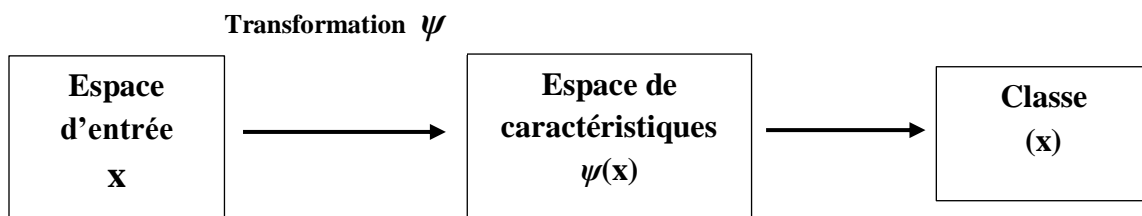


Fig 5.3 - Principe de la technique SVM.

Par la suite nous détaillons les deux cas : séparation linéaire et non linéaire.

### 5.4.2 Cas simple : SVM linéaire à deux classes

On considère le cas d'une base  $B$  de  $n$  objets à deux classes linéairement séparables. Chaque objet est alors représenté par le couple  $\{x_i, y_i\}$ ,  $x_i \in R^d$ ,  $y_i \in \{-1, 1\}$  où  $x_i$  est le vecteur décrivant chaque objet dans l'espace de description de dimension  $d$ , et  $y_i$  est l'étiquette de la classe de cet objet (Fig 5.4).

On suppose qu'il existe un hyperplan séparant les points de chaque classe, l'équation d'un tel hyperplan s'écrit alors comme suit :

$$\langle w, x \rangle + b = 0, \quad (5.3)$$

où :  $w$  est une normale à l'hyperplan,

$\langle w, x \rangle$  représente le produit scalaire entre  $w$  et  $x$ .

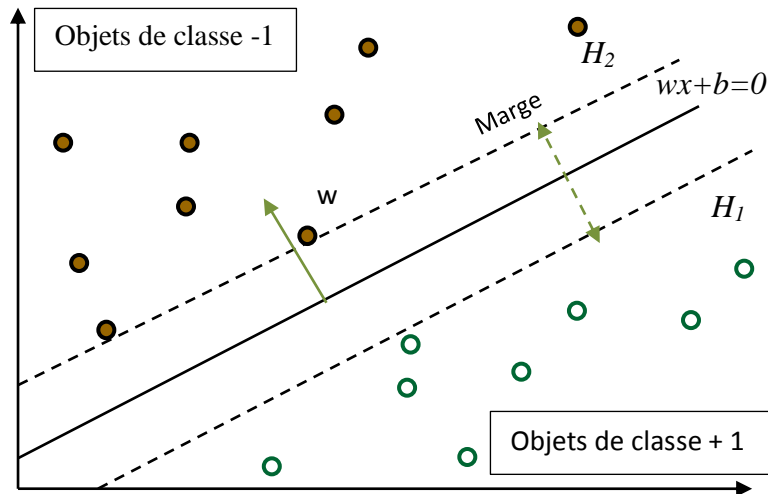


Fig 5.4- Principe des SVM

Il existe une infinité d'hyperplans respectant cette équation, il faut trouver le meilleur. La notion de meilleur hyperplan est introduite en considérant deux hyperplans parallèles ( $H_1$  et  $H_2$ ) passant chacun par les points les plus proches de la frontière des classes qu'ils limitent et n'ont aucun point de la base d'apprentissage  $B$  entre eux (Fig 5.5). Leurs équations respectives sont :

$$H_1: \langle w, x \rangle + b = 1 \quad (5.4)$$



$$H_2: \langle w, x \rangle + b = -1 \quad (5.5)$$

Pour garantir qu'il n'y ait aucun point de B entre eux, tout point  $i$  de B doit satisfaire l'inégalité suivante :

$$\forall i \in [1..n], \quad y_i(\langle w, x_i \rangle + b) \geq 1 \quad (5.6)$$

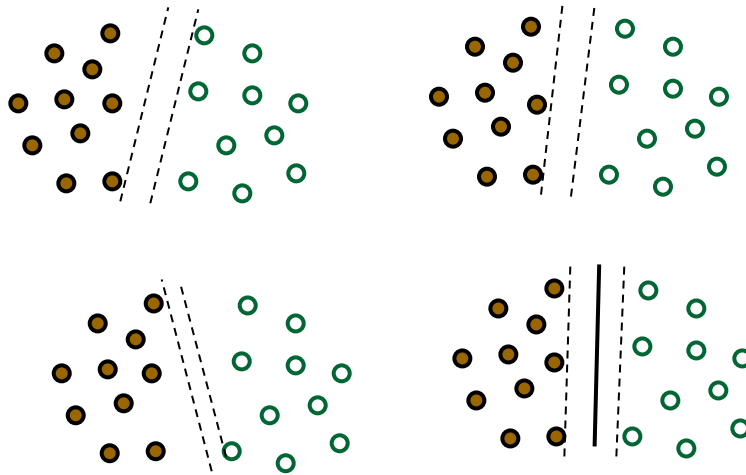


Fig 5.5- Exemple d'hyperplans séparateurs

La distance entre ces deux hyperplans (la marge) est  $\frac{2}{\|w\|}$  donc pour maximiser sa valeur, il faut minimiser  $\|w\|$ . Ceci est un problème d'optimisation quadratique. Le meilleur hyperplan séparant est alors celui qui est à égale distance de  $H_1$  et  $H_2$ . Pour résoudre ce problème, on utilise la théorie de Lagrange afin de reformuler le problème de minimisation sous la contrainte d'inégalité. On obtient alors la fonction de Lagrange définie par :

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i y_i (\langle w, x_i \rangle + b) + \sum_{i=1}^n \alpha_i, \quad \alpha_i \geq 0 \quad (5.7)$$

La fonction  $L$  est appelé le Lagrangien, cette dernière doit être minimisée par rapport aux variables dites primales  $w$  et  $b$ , et maximisée par rapport aux variables duales  $\alpha_i$  : ce sont les

conditions de Karush-Kuhn-Tucker (KKT) [SCS01]. Les  $\alpha_i$  sont aussi appelés les coefficients de Lagrange.

Notons que  $\frac{1}{2} \|w\|^2$  a été choisi au lieu de  $\frac{1}{2} \|w\|$  pour faciliter la résolution du système.

Les conditions KKT sont nécessaires et suffisantes, et la solution du problème est obtenue par le calcul de la dérivée partielle de  $L(w, b, \alpha)$  par rapport à  $w$  et  $b$ :

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i \quad (5.8)$$

et

$$\frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^n \alpha_i y_i \quad (5.9)$$

En annulant les deux dérivées partielles, on obtient le système suivant :

$$\sum_{i=1}^n \alpha_i y_i x_i = w \quad (5.10)$$

et

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (5.11)$$

On peut alors obtenir un problème dual en prenant en considération les éléments de la base d'apprentissage  $B$  par couple  $x_i$  et  $y_i$ , et en remplaçant  $w$  par sa valeur trouvée. Ce problème s'écrit alors :

$$L(w, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \quad (5.12)$$

$L(w, b, \alpha)$  est alors maximisée selon les  $\alpha_i$  en respectant les contraintes :

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad \text{et} \quad \alpha_i \geq 0 \quad (5.13)$$

Une fois que l'optimisation est effectuée, on peut aisément déduire  $b$ .

La décision de la classification est alors obtenue pour tout  $x$  n'appartenant pas à l'ensemble d'apprentissage par :

$$y = \text{signe}(\langle w, x \rangle + b). \quad (5.14)$$

En remplaçant  $w$  par sa valeur de l'équation (5.10) on obtient la fonction de décision suivante :

$$f(x) = \sum_{i=1}^n \alpha_i y_i \langle x_i, x \rangle + b \quad (5.15)$$

Chaque coefficient multiplicateur de Lagrange  $\alpha_i$  correspond à un élément  $x_i$  de la base d'apprentissage  $B$ . Les  $x_i$  qui ont contribué à la maximisation de la marge sont associés à des  $\alpha_i$  non nuls et sont donc appelés vecteurs de support (SV). Les autres vecteurs de la base d'apprentissage ne participent donc pas à la décision finale et l'équation (5.15) devient :

$$f(x) = \sum_{i \in SV} \alpha_i y_i \langle x_i, x \rangle + b \quad (5.16)$$

où  $SV$  est l'ensemble des indices des vecteurs de support (vecteurs pour lesquels les coefficients de Lagrange sont non nuls).

### 5.4.3 Cas non linéairement séparable

Le cas non linéairement séparable se présente lorsqu'il n'existe pas d'hyperplan d'équation linéaire séparant les données sans erreurs.

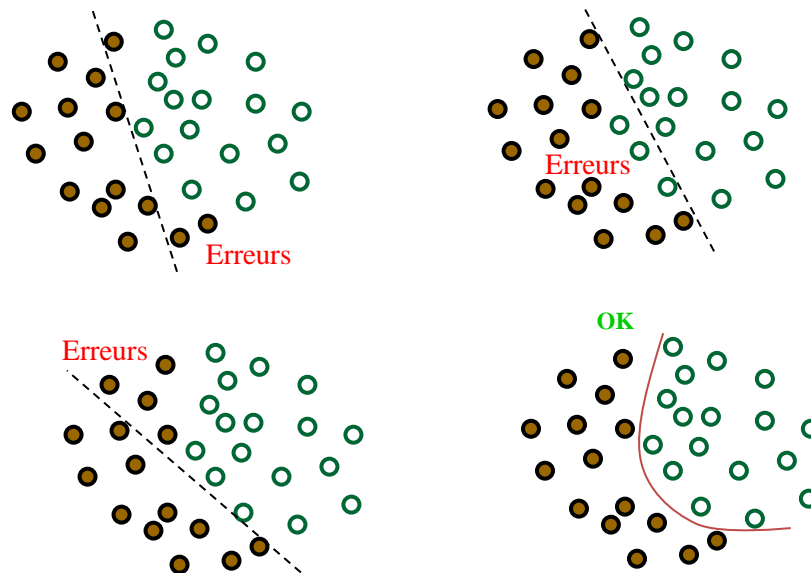


Fig 5.6- Deux classes non linéairement séparables

Pour apporter une solution à ce problème, on introduit la notion de marge souple (soft marge), qui autorise un certain nombre d'éléments d'une classe à se trouver du mauvais côté

de la frontière lors de l'apprentissage (Fig 5.7). Ces éléments sont appelés des « outliers » [SCS01].

On applique alors une marge d'erreur  $\varepsilon_i$  à l'équation (5.6) pour obtenir :

$$\forall i \in [1..n], \quad y_i(\langle w, x_i \rangle + b) \geq 1 - \varepsilon_i \quad (5.17)$$

Pour maximiser la marge il ne suffit plus de minimiser  $\|w\|$  mais la fonction :

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \varepsilon_i, \quad (5.18)$$

où  $C$  est le coût attribué à un échantillon mal classé, qui permet de contrôler le nombre d'erreurs tolérées.

De la même façon que dans le cas de données linéairement séparables, on détermine les vecteurs supports par résolution quadratique. Dans ce cas, la fonction de décision est identique à celle définie par l'équation (5.16).

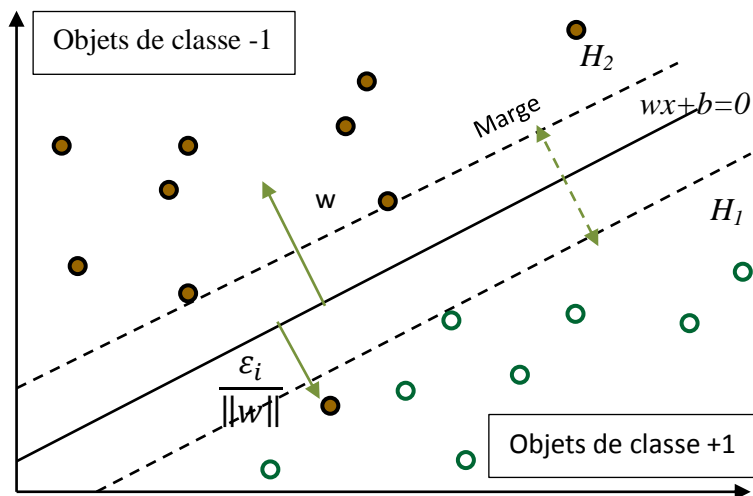


Fig 5.7- Principe de "l'outlier".

La frontière définie est là aussi linéaire mais décalée en fonction du coût fixé.

L'erreur due à ce type de séparation est souvent plus importante que l'erreur générée par une frontière non linéaire, c'est pourquoi une modélisation non linéaire a été introduite. Cette méthode dite « astuce du noyau » (kernel trick) a été proposée par Aizerman en 1964.

Les équations de base font intervenir les données d'apprentissage (les échantillons et leur classe) sous forme d'un produit scalaire.

### 5.4.3.1 Astuce de noyau

Il a été démontré qu'il est possible que les données initiales (de l'espace  $R^d$ ) soient linéairement séparables dans un espace euclidien  $F \subset R^e$  (avec  $e > d$ ), dans lequel on projette l'espace de description  $R^d$  à l'aide d'une fonction  $\emptyset$  telle que :

$$\emptyset: R^d \rightarrow R^e, \emptyset(x) = \tilde{x} \quad (5.19)$$

En effet, plus la dimension de l'espace de description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les classes est élevée. En transformant l'espace d'entrée en un espace de redescription  $F$  de très grande dimension, éventuellement infinie, il devient donc possible d'envisager d'utiliser la méthode de résolution quadratique précédente même dans le cas de classes non linéairement séparables.

Si ces espaces sont munis d'un produit scalaire, noté  $\langle \cdot, \cdot \rangle$ , on peut définir dans  $R^d$  une fonction noyau telle que ;

$$K: R^d \times R^d \rightarrow R^e \times R^e, K(x, y) = \langle \emptyset(x), \emptyset(y) \rangle = \langle \tilde{x}, \tilde{y} \rangle \quad (5.20)$$

L'algorithme des SVM est alors le même que dans le cas linéaire : il suffit de remplacer tous les produits scalaires par une fonction noyau non linéaire. Avec cette astuce, Les SVM peuvent définir le meilleur hyperplan séparateur linéaire dans le nouvel espace  $R^e$  en utilisant la fonction noyau appropriée.

En appliquant la fonction noyau à la fonction de Lagrange, on obtient la fonction suivante :

$$L(w, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (5.21)$$

Donc il s'agit de maximiser (5.21) sous la contrainte :

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad \text{et} \quad 0 \leq \alpha_i \leq C \quad (5.22)$$

Par rapport au cas séparable, une contrainte supplémentaire sur les  $\alpha_i$  a été introduite : ils

admettent à présent la borne supérieure C.

La fonction de décision devient donc :

$$y = \text{signe}(\sum_{i \in SV} \alpha_i y_i K(x_i, x) + b) \quad (5.23)$$

où le SV est l'ensemble des indices des vecteurs supports issus de la phase d'apprentissage.

Toute fonction satisfaisant le critère de Mercer peut être utilisée comme noyau [SCS01]. Ce critère est le suivant :

Soit  $k$  une fonction symétrique, on définit alors la matrice  $M$  dont l'élément  $(i,j)$  est  $k(x_i, x_j)$ . Si cette matrice est définie positive, c'est-à-dire que pour chaque vecteur colonne  $x$  de cette matrice  $x^T M x > 0$ , alors il existe un espace  $F$  et une fonction  $\phi$  tels que  $K(x, y) = \langle \phi(x), \phi(y) \rangle$ . On peut ainsi définir notre propre noyau, cependant il existe plusieurs noyaux prédéfinis dont les plus employés sont :

- Noyau linéaire :  $K(x, y) = x \cdot y$  (correspond au SVM linéaire)
- Noyau polynomial (d'ordre  $P$ ) :  $K(x, y) = (\gamma \langle x, y \rangle + c)^P$ , avec  $\gamma$  constante
- Noyau RBF (fonction à base radiale) :  $K(x, y) = \exp(-\gamma \|x - y\|^2)$ , avec  $\gamma$  une constante.
- Noyau RBF gaussien :  $K(x, y) = \exp(-\frac{\|x - y\|}{2\gamma^2})$ , avec  $\gamma$  constante.
- Noyau tangente hyperbolique :  $K(x, y) = \text{Tanh}(k \langle x, y \rangle - \theta)$ , où  $\theta$  est une constante et  $\text{Tanh}$  une fonction hyperbolique.

#### 5.4.4 Les SVM multi-classes

Les séparateurs à vaste marge ont été développés pour traiter des problèmes binaires. Toutefois, les bases de données réelles soulèvent des problèmes avec plusieurs classes.

Pour traiter ces problèmes, les deux stratégies les plus utilisées sont :

##### ❖ Un contre-tous

Ou encore appelé un contre le reste : dans cette stratégie, le problème multi-classes, avec  $k$  classes, est décomposé en  $k$  sous-problèmes binaires. Ainsi, un sous-problème pour la classe  $c$ , peut être ramené à un problème binaire avec les données de la classe  $c$  comme instance

positive et les autres données comme instances négatives. Pour chaque problème, un hyperplan séparateur  $(w^i, b^i)$  est « appris ». Le classement d'une nouvelle donnée  $x$  est donnée par :

$$y = \underset{i}{\operatorname{argmax}} (\langle w^i, x \rangle + b^i) \quad (5.24)$$

#### ❖ Un contre un

dans cette stratégie, le problème de  $k$  classes est décomposé en  $k(k-1)/2$  sous problèmes binaire. Chaque problème  $(i,j)$  consiste à déterminer un SVM séparant les instances de la classe  $i$  des instances de la classe  $j$ ; le classement s'effectue ensuite selon le schéma du vote majoritaire. Pour classer une donnée  $x$ , il faut ainsi traiter les  $k(k-1)/2$  sous-problèmes et affecter à  $x$  la classe qui aura été affectée le plus de fois lors du traitement de sous-problèmes.

## 5.5 Validation croisée

En classification supervisée, il faut utiliser une base d'apprentissage et une base de test pour éviter l'apprentissage "par cœur", c'est-à-dire d'utiliser en test des échantillons qui ont servi à l'apprentissage. Or les performances des méthodes de classification dépendent généralement du nombre d'échantillons d'apprentissage : plus ce nombre est élevé, plus fiables seront les règles de classification. En même temps, il est nécessaire de conserver un nombre significatif d'échantillons de test pour que l'évaluation de ces performances soit significative. La technique de la validation croisée est fréquemment utilisée pour répondre à ces deux besoins : elle consiste à diviser l'ensemble de départ en un certain nombre de sous ensembles de taille égale, chaque sous ensemble étant alors utilisé comme une base de test, alors que l'union de tous les autres sous ensembles est utilisée comme base d'apprentissage (Fig 5.8). Soit  $r$  ce nombre de sous ensembles, on parle alors de validation croisée d'ordre  $r$  (en anglais on parle de  $r$  folds cross validation –  $r$ -CV). Naturellement  $r$  est supérieur ou égal à 2.

Soit  $D$  notre ensemble de données de  $n$  échantillons (dont on connaît la classe), chacun comportant  $d$  attributs. L'ensemble  $D$  est alors découpé en  $r$  sous-ensembles disjoints de tailles identiques, la représentation statique de chaque classe étant préservée par rapport à

celle de l'ensemble D.

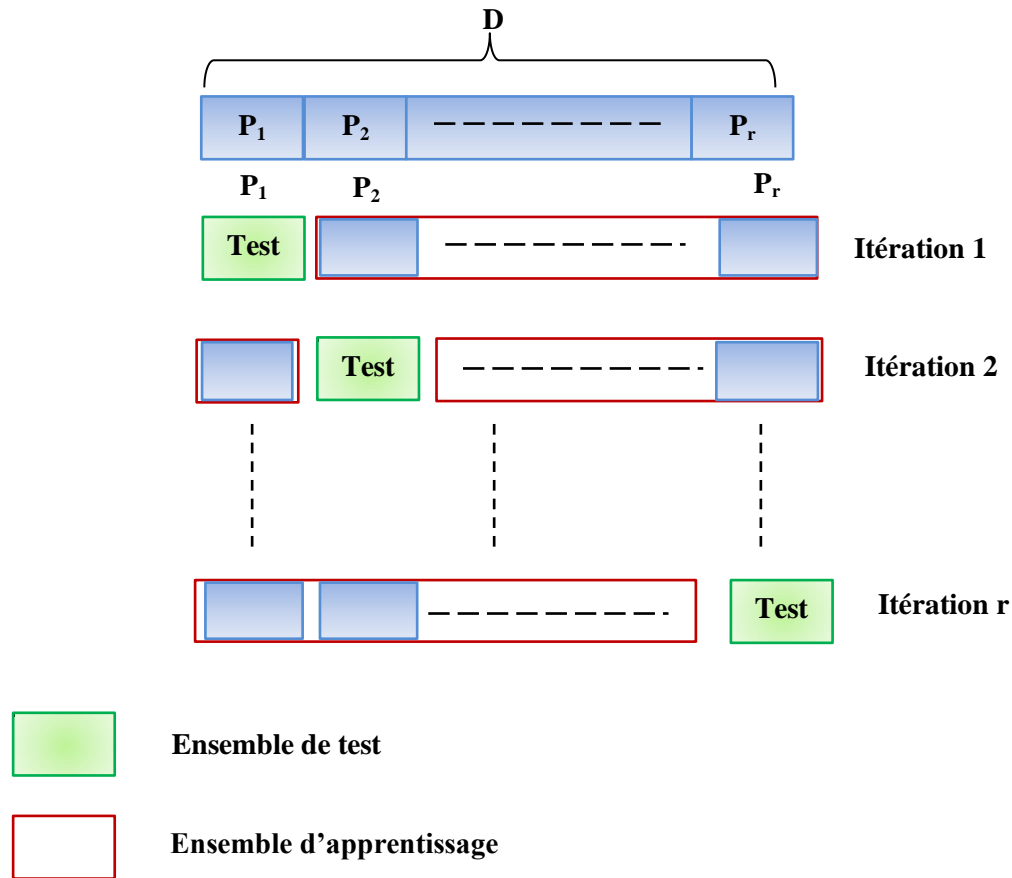


Fig 5.8- Schéma de validation croisée d'ordre  $r$

A chaque itération, le nombre d'erreurs de classification réalisé est comptabilisé et le taux d'erreur de classification total est obtenu en divisant ce nombre d'erreurs par le nombre d'échantillons  $n$ .

Etant donné que la classe de chaque échantillon est connue, nous pouvons également produire la matrice de confusion qui recense pour chaque classe réelle, le nombre d'échantillons que le classifieur a classé dans chaque classe, ce qui permet de détecter les éventuelles confusions que pourrait faire le système. Si  $C$  est le nombre de classe de  $D$ , la matrice de confusion est une matrice de taille  $C \times C$  où la cellule à l'intersection de la ligne  $i$  et de la colonne  $j$  est le nombre d'éléments de classe réelle  $i$  classés par le système dans la classe  $j$ .



On appelle « leave one out » [SCS01] le cas particulier pour lequel le nombre de sous ensemble de validation croisée est égal au nombre de mesures de l'ensemble ( $r=n$ ). L'apprentissage se fait alors sur la base complète sauf 1 échantillon et le test est effectué sur l'élément restant. Cela permet, dans le cas d'un faible ensemble de données de maximiser l'ensemble d'apprentissage, mais présente l'inconvénient de se rapprocher de l'apprentissage « par cœur ». La difficulté de cette méthode est de trouver la bonne valeur de  $n$  en fonction de la taille de l'ensemble de départ et de la complexité du problème : plus le problème est complexe, plus le système a besoin d'exemples pour apprendre.

- Si  $r$  est petit, l'ensemble d'apprentissage sera trop faible et le système de classification ne généralisera pas assez, entraînant une augmentation du taux d'erreurs. Par contre, il y a moins de phase d'apprentissage à réaliser et l'obtention des résultats est plus rapide. Cette solution est intéressante quand on dispose de beaucoup de mesures.
- Si  $r$  est grand, l'ensemble de tests sera plus faible par rapport à l'ensemble d'apprentissage et on risque le phénomène de sur-apprentissage (cas extrême : le « leave one out »).

## 5.6 Conclusion

Dans ce chapitre nous avons présenté les concepts de classification par trois méthodes, La méthode de voisin le plus proche est basée sur le calcul d'une distance, ce qui la rend sensible au choix de ce paramètre.

Le LVQ est une méthode prometteuse. Dans le SVM, la sélection des Vecteurs Supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan est une méthode efficace et très rapide puisque seuls ces vecteurs sont utilisés pour la discrimination

Dans le chapitre suivant nous allons présenter les résultats expérimentaux de la classification du trafic routier en utilisant ces trois classifieurs.

## 6.1 Introduction

Dans ce chapitre, nous nous intéresserons à la classification du trafic routier. Dans un premier temps nous allons appliquer divers combinaisons des paramètres de trafic routier estimés par les méthodes décrites dans les chapitres précédents à l'entrée d'un classifieur KPPV, ensuite LVQ et enfin un classifieur SVM, puis nous allons retenir la meilleure combinaison de paramètres qui produit l'estimation la plus précise de l'état du trafic avec l'un de ces classifieurs.

Ensuite nous estimerons l'état du trafic routier par l'utilisation du modèle de texture dynamique et la distance KL entre les vidéos clip, avec soit un classifieur KPPV, soit un classifieur SVM, que nous comparerons en terme de performance.

## 6.2 Classification du trafic routier

Il n'existe pas un standard unifié pour la définition de la situation du trafic congestionné, on considère la définition suivante [XIJ09]:

Les véhicules trouvent des difficultés en circulation, et la longueur de la file d'attente est supérieur à 250m dans les croisements, sans contrôle des feux de signalisation ou la signalisation de passage, les véhicules ne peuvent pas passer durant trois périodes de l'apparition du feu vert; de cela on définit ce passage comme passage congestionné, et la section de la congestion est définie lorsque les véhicules sont bloqués dans leurs voies et la longueur de la file atteinte plus d'un 1Km.

Grâce à cette définition on peut constater qu'un phénomène de congestion routière résulte de l'union de plusieurs paramètres tels que :

- Taux élevé de l'occupation de la route.
- Vitesse de circulation faible.
- File d'attente longue.

Il est nécessaire de noter qu'une route est considérée comme congestionnée lorsque les

véhicules ne peuvent pas passer durant trois périodes de feu vert. Généralement une observation de la route durant 3 minutes est exigée pour décider si une route est congestionnée ou non.

Dans cette partie nous allons estimer l'état du trafic routier par l'utilisation de multiples combinaisons de paramètres estimés par les méthodes décrites dans les chapitres précédents, le but est de trouver la combinaison la plus courte et la plus performante en termes de taux de classification. Nous comparerons les résultats obtenus en utilisant trois classifieurs : KPPV, LVQ et SVM qui ont été décrits dans le chapitre précédent.

Afin de pouvoir comparer nos résultats avec ceux publiés dans [CHV05], nous utilisons la même base vidéo. La vérité terrain correspondant à la totalité de cette base constituée de scènes vidéo, filmées pendant deux jours, est reproduite dans la figure suivante :

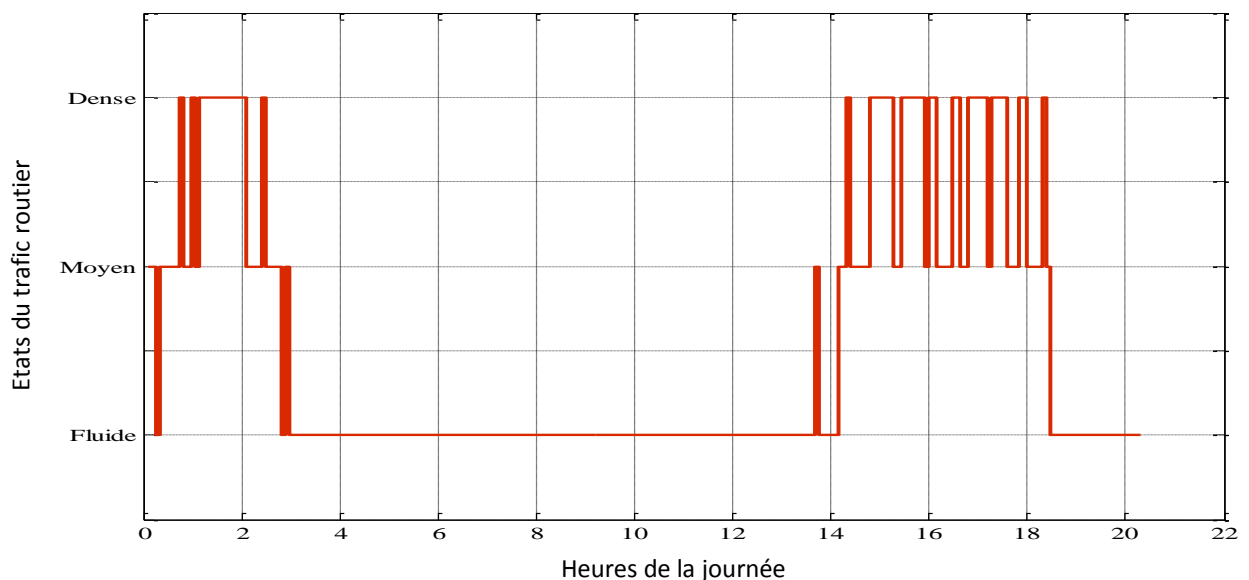


Fig 6.1- Vérité de terrain de la base vidéo utilisée pour le test et l'apprentissage.

Comme dans [CHV05] deux scénarios sont considérés. Dans le premier scénario l'apprentissage du classifieur est effectué en utilisant 4 heures d'observation de la route pendant le premier jour, ce qui correspond à 61 scènes vidéo de la base de données, les 193 scènes vidéo restantes sont réservées pour la phase de test; elles couvrent la période entre 6h et 21h pendant le jour suivant. Le deuxième scénario sera présenté dans les sections suivantes.

## 6.2.1 Classification du trafic routier par l'utilisation des paramètres moyens

### 6.2.1.1 Classification par le Plus Proche Voisin (KPPV)

Nous avons utilisé une série de combinaisons de paramètres de trafic routier afin de classer le trafic en trois classes : fluide, moyen, dense.

Nous avons utilisé premièrement le plus proche voisin dont les paramètres sont:

- K=1 voisin

- la distance euclidienne définie par :  $D(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$

où  $X=(x_1, \dots, x_n)$  et  $Y=(y_1, \dots, y_n)$  sont deux exemples.

Les résultats de classification obtenus sont donnés par le tableau suivant :

Tests	Paramètres de trafic utilisés	Taux de classification correcte
<b>Vitesse estimée par détection et poursuite de mouvement</b>		
1	Vitesse + Taux d'occupation	65.80%
2	Vitesse + Distance inter-véhicules	64.76%
3	Vitesse + Taux d'occupation+ Débit	62.69%
4	Vitesse + Distance inter-véhicules+ Débit	65.28%
<b>Vitesse estimée par mise en correspondance des blocs</b>		
5	Vitesse BM + Taux d'occupation	77.20%
6	Vitesse BM+ Distance inter-véhicules	51.29%
7	Vitesse BM+ Densité	89.11%
8	Vitesse BM+ Taux d'occupation+ Débit	50.25%
9	Vitesse BM + Distance inter-véhicules+ Débit	50.25%
10	Vitesse BM+ Densité+ Débit	51.29%
11	Vitesse BM+ Densité+ Taux d'occupation	89.11%

Tab 6.1- Classification de trafic routier par KPPV et les paramètres moyens

Le meilleur taux de classification obtenu est **89.11%** correspond à la classification par l'utilisation de la vitesse moyenne du trafic estimée par la mise en correspondance de blocs, et la densité moyenne du mouvement.

### 6.2.1.2 Classification par le LVQ

Afin d'améliorer la précision de la classification du trafic routier nous avons tenté d'utiliser le classifieur LVQ (présenté dans le chapitre précédent). La structure du réseau de neurones LVQ qu'on a adoptée dans ce travail est représentée à la figure suivante :

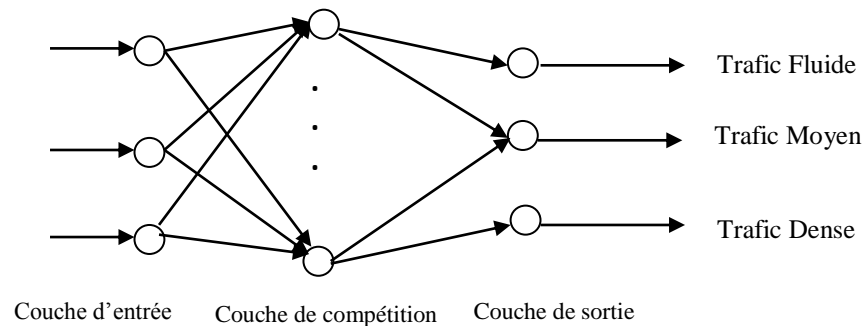


Fig 6.2-Structure du réseau de neurone LVQ

Le nombre de neurones dans la couche d'entrée varie entre deux et trois selon le nombre des paramètres utilisé. La couche de compétition contient entre 5 et 7 neurones, théoriquement le nombre des neurones dans cette couche doit être égal à deux fois le nombre des neurones dans la couche d'entrée plus un [XIJ09]. La couche de sortie comprend trois neurones, chacun correspond à un état du trafic: fluide, moyen, dense. Nous avons utilisé au maximum 700 itérations, et un taux d'apprentissage  $T=0.1$ .

Les résultats obtenus par ce classifieur sont donnés par le tableau suivant :

Tests	Paramètres de trafic utilisés	Taux de classification correcte
<b>Vitesse estimée par détection et poursuite de mouvement</b>		
<b>1</b>	Vitesse + Taux d'occupation	71.50%
<b>2</b>	Vitesse + Distance inter-véhicules	71.50%
<b>3</b>	Vitesse + Taux d'occupation+ Débit	72.02%
<b>4</b>	Vitesse + Distance inter-véhicules+ Débit	72.02%
<b>Vitesse estimée par mise en correspondance des blocs</b>		
<b>5</b>	Vitesse BM + Taux d'occupation	66.32%
<b>6</b>	Vitesse BM+ Distance inter-véhicules	58.54%
<b>7</b>	Vitesse BM+ Densité	86.01%
<b>8</b>	Vitesse BM+ Taux d'occupation+ Débit	72.53%
<b>9</b>	Vitesse BM + Distance inter-véhicules+ Débit	71.50%
<b>10</b>	Vitesse BM+ Densité+ Débit	67.35%
<b>11</b>	Vitesse BM+ Densité+ Taux d'occupation	86.01%

Tab 6.2 - Classification de trafic routier par LVQ et les paramètres moyens.

On remarque que le meilleur taux de classification obtenu est de **86.01%** et il correspond à la classification par l'utilisation de la vitesse macroscopique de trafic routier calculée par l'estimation des vecteurs de vitesse et la densité des vecteurs de mouvement (test 7). Ce résultat est le même que celui qui a été obtenu en ajoutant le taux d'occupation de la route moyen, ce qui signifie que c'est un paramètre redondant.

A l'exception des tests 5, 7 et 11, les résultats de tous les autres tests ont pu être améliorés grâce au LVQ.

Le LVQ n'ayant pas donné des résultats satisfaisants, nous avons pensé à l'utilisation du SVM, afin d'améliorer le taux de classification.

### 6.2.1.3 Classification par SVM

Pour l'implémentation de la SVM nous avons choisi la bibliothèque LIBSVM en matlab, fournie par Chang et Lin (2001) [CSI01], puisqu'elle est très bien documentée et très populaire dans le domaine. Nous allons dans ce qui suit la décrire brièvement et montrer comment nous l'avons utilisée.

#### ❖ La bibliothèque LIBSVM

LIBSVM est l'une des bibliothèques les plus utilisées. Elle implémente plusieurs versions de l'algorithme SVM. La version nommée C-SVC sera utilisée et correspond à la définition initiale du SVM telle qu'elle a été introduite par Cortes et Vapnik (1995). L'un des paramètres fondamentaux de LIBSVM est le choix du type de noyau. Pour arriver à séparer facilement les classes dans un espace donné, le SVM utilise une projection dans un espace augmenté en utilisant la notion de noyau (voir la section 5.4 pour plus de détails). Dans notre cas, le noyau RBF (radial basis function – fonction à base radiale) a été sélectionné. Ce noyau permet de gérer les cas où la relation entre les caractéristiques et les classes n'est pas linéaire.

Deux autres paramètres statiques (non-optimisés) sont suggérés également et sont relatifs uniquement à l'implémentation spécifique de LIBSVM. Premièrement, l'algorithme implémenté par LIBSVM inclut une heuristique d'arrêt hâtif de l'entraînement. Cette heuristique permet d'économiser les ressources en termes de temps de calcul. Enfin, le dernier critère suggéré est la taille du cache. Pour accélérer les calculs, l'implémentation du C-SVC conserve en mémoire (en cache) certaines données, la taille du cache d'apprentissage suggérée est de 100 Mo (méga-octets).

Enfin, les deux paramètres restants ont un impact majeur sur les performances et doivent être optimisés selon le contexte précis de l'apprentissage. Pour ce faire, le guide suggère une recherche de la configuration optimale en essayant de multiples combinaisons de paramètres.

Les deux paramètres à optimiser sont le coût de l'erreur  $C$  et la variance du noyau RBF  $\gamma$  (gamma). Étant donné le faible nombre de paramètres, et la nécessité de couvrir l'espace des possibilités, la recherche sera effectuée dans une grille de recherche (grid search). Pour ce

faire nous avons exploré l'espace des paramètres pour  $C$  et  $\gamma$  dans l'intervalle :  $[e^{-10} : e^4]$ , pour chaque couple  $(C, \gamma)$ , un SVM est entraîné sur la portion de validation croisée (1/3 de la base de données) telle qu'elle a été détaillée dans 5.5 ; le meilleur résultat en termes de taux de bonne classification est retenu.

La paramétrisation optimale pour le SVM est obtenue, et conservée pour la phase d'apprentissage réelle.

Le tableau suivant présente les résultats de classification du trafic par le SVM, où nous comparons entre un noyau linéaire et le noyau RBF :

Tests	Paramètres de trafic utilisés	Noyau RBF	Noyau linéaire
<b>Vitesse estimée par détection et poursuite de mouvement</b>			
<b>1</b>	Vitesse + Taux d'occupation	63.73%	84.97%
<b>2</b>	Vitesse + Distance inter-véhicules	67.87%	70.46%
<b>3</b>	Vitesse + Taux d'occupation+ Débit	67.35%	84.97%
<b>4</b>	Vitesse + Distance inter-véhicules+ Débit	67.35%	71.50%
<b>Vitesse estimée par mise en correspondance des blocs</b>			
<b>5</b>	Vitesse BM + Taux d'occupation	90.15%	92.22%
<b>6</b>	Vitesse BM+ Distance inter-véhicules	69.43%	62.17%
<b>7</b>	Vitesse BM+ Densité	90.15%	92.74%
<b>8</b>	Vitesse BM+ Taux d'occupation+ débit	71.50%	92.22%
<b>9</b>	Vitesse BM + Distance inter-véhicules+ Débit	59.06%	62.17%
<b>10</b>	Vitesse BM+ Densité+ Débit	78.23%	87.56%
<b>11</b>	Vitesse BM+ Densité+ Taux d'occupation	90.15%	92.74%

Tab 6.3- Classification de trafic routier par SVM et les paramètres moyens



La figure ci-dessous représente une comparaison entre les résultats obtenus par les deux noyaux :

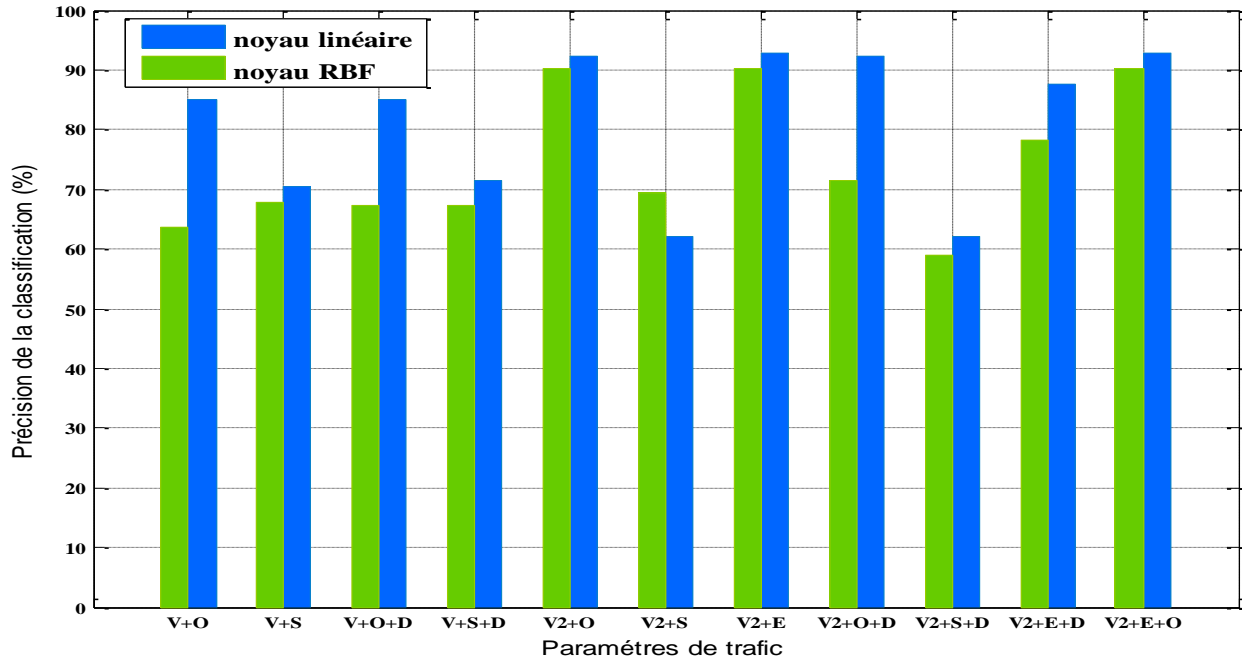


Fig 6.3- Comparaison entre noyau linéaire et noyau RBF

où :

V : vitesse moyenne par poursuite de mouvement

V2 : vitesse moyenne par mise en correspondance des blocs d'images.

O : taux d'occupation moyen

D : débit moyen de trafic routier

S : distance inter-véhicules moyenne

E : densité de mouvement moyenne

On remarque qu'en général les taux de classification ont augmenté par rapport aux résultats précédents. On remarque aussi qu'à l'exception du test 6 le noyau linéaire donne des résultats plus précis que ceux fournis par le noyau RBF.

Le meilleur taux de classification est **92.74%**, et a été obtenu en utilisant la vitesse BM et la densité avec un noyau linéaire. Ce résultat est encourageant. Le plus faible taux de classification est **62.17%**, et a été obtenu en ajoutant la distance inter-véhicules moyenne à

quelques autres paramètres, ceci nous amène à conclure que ce paramètre n'est pas précis et ne peut pas contribuer efficacement à la classification du trafic routier.

Les résultats présentés dans le tableau 6.3 ont été obtenus en utilisant la stratégie un contre un. Nous avons également utilisé le SVM avec la stratégie un contre tous; les résultats de la classification obtenus sont présentés par le tableau suivant :

Tests	Paramètres de trafic utilisés	Noyau RBF	Noyau linéaire
<b>Vitesse estimée par détection et poursuite de mouvement</b>			
<b>1</b>	Vitesse + Taux d'occupation	67.87%	82.38%
<b>2</b>	Vitesse + Distance inter-véhicules	69.43%	75.12%
<b>3</b>	Vitesse + Taux d'occupation+ Débit	74.61%	86.01%
<b>4</b>	Vitesse + Distance inter-véhicules+ Débit	74.61%	82.38%
<b>Vitesse estimée par mise en correspondance des blocs</b>			
<b>5</b>	Vitesse BM + Taux d'occupation	92.22%	91.70%
<b>6</b>	Vitesse BM+ Distance inter-véhicules	69.43%	65.80%
<b>7</b>	Vitesse BM+ Densité	90.67%	94.81%
<b>8</b>	Vitesse BM+ Taux d'occupation+ débit	65.80%	92.74%
<b>9</b>	Vitesse BM + Distance inter-véhicules+ Débit	70.98%	69.43%
<b>10</b>	Vitesse BM+ Densité+ Débit	84.45%	94.30%
<b>11</b>	Vitesse BM+ Densité+ Taux d'occupation	90.67%	94.30%

Tab 6.4- Classification de trafic routier par SVM par la stratégie un contre tous

Ces résultats de classification sont dans la plus part des cas (9/11) meilleurs que ceux obtenus par la stratégie un contre un (voir la figure 6.4). On remarque que pour le test 3 (vitesse, taux d'occupation, débit) le pourcentage de bonne classification a pu atteindre **86%**, cela nous rassure sur la précision des paramètres extraits par détection et poursuite de mouvement.

Comme dans le cas de la stratégie un contre un, la classification par l'utilisation du noyau RBF est généralement inférieure à la classification par le noyau linéaire, comme cela est illustré à la figure 6.5. Le meilleur taux de classification est toujours obtenu avec les paramètres vitesse BM et densité, il atteint **94.81%**. Ce résultat est très satisfaisant.

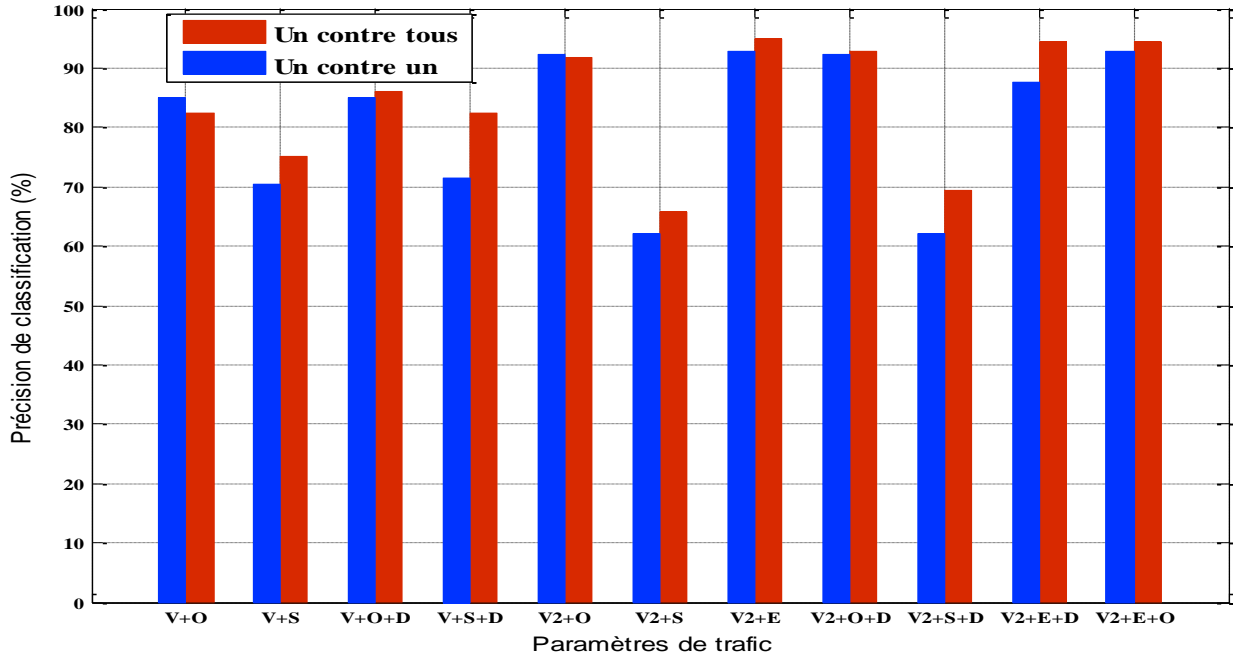


Fig 6.4- Comparaison entre la stratégie un contre un et un contre tous pour la classification SVM

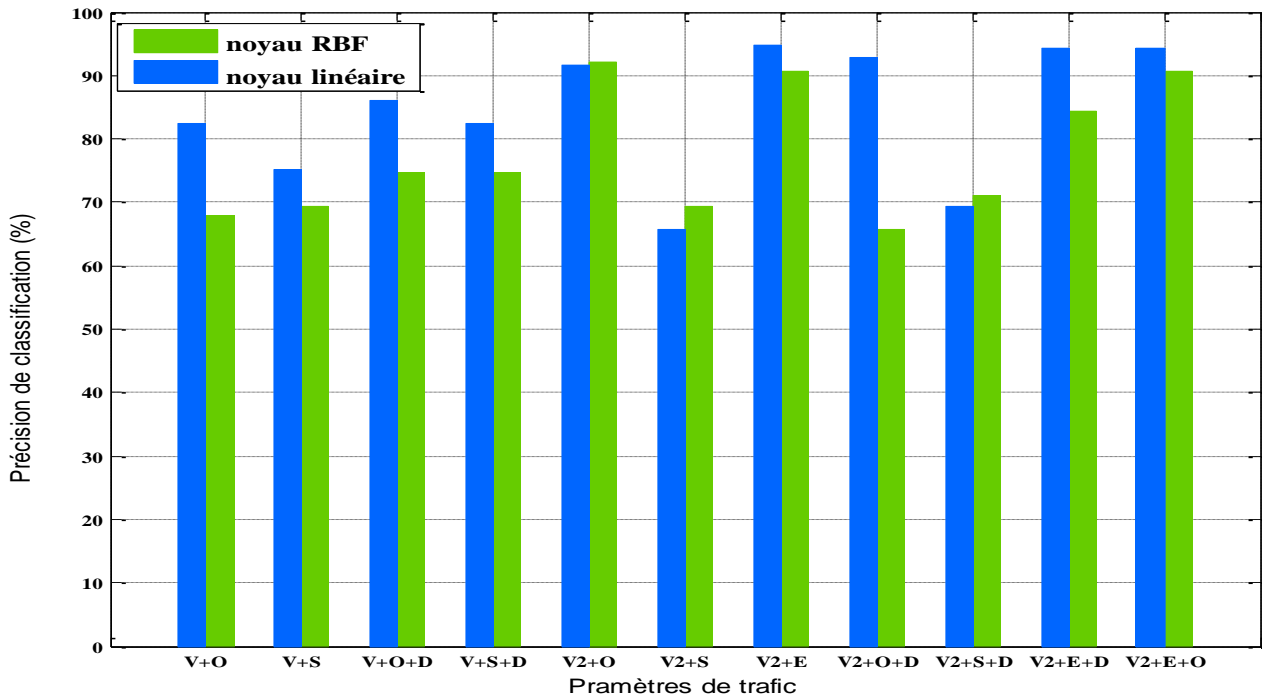


Fig 6.5- Comparaison entre un noyau RBF et linéaire pour la classification par SVM (un contre tous)

A la lumière des résultats de classification du trafic routier présentés on peut conclure:

- La classification du trafic routier par SVM donne des résultats plus précis que par le KPPV ou par le LVQ.
- Le problème de la classification du trafic par paramètres moyens est un problème linéaire.
- L'utilisation de la stratégie un contre tous est plus efficace que la stratégie un contre un pour la classification du trafic.
- La combinaison entre la vitesse et la densité des vecteurs de mouvement estimés par la mise en correspondance des blocs donne les meilleurs taux de classification.

Nous allons dans ce qui suit analyser plus en détails les résultats obtenus. Pour ne pas encombrer le mémoire, nous nous limiterons à ceux obtenus par le meilleur classifieur, à savoir le classifieur SVM avec la stratégie un contre tous et utilisant les paramètres vitesse BM + densité.

Le meilleur taux de validation croisée obtenu est 90.16% et il correspond à un coût  $C=1$ . La figure suivante représente l'évolution de l'état réel du trafic routier à laquelle est superposée celle du trafic estimé, les erreurs de classification sont indiquées par un cercle. Nous rappelons que la phase d'apprentissage a été effectuée en utilisant les scènes filmés durant le premier jour.

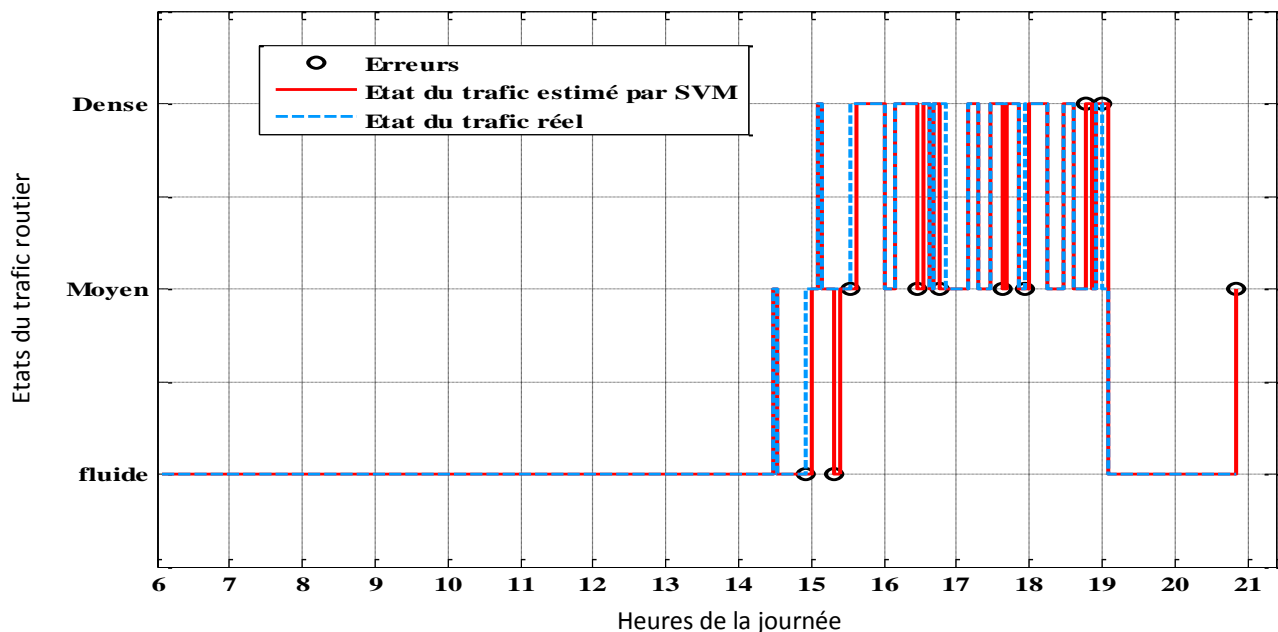


Fig 6.6- Etats du trafic réel et estimé par SVM avec le schéma un contre tous

Le tableau suivant représente la matrice de confusion qui correspond à ce test. Le taux de classification correcte est de 94.81%.

Prédites \ Réelles	Fluide	Moyen	Dense
Fluide	137	1	0
Moyen	2	22	2
Dense	0	5	24

Tab 6.5- Matrice de confusion pour SVM avec le schéma un contre tous

On remarque que la plus part des confusions ont lieu entre le trafic dense et moyen.

Les figures suivantes représentent quelques images appartenant à différents types de trafic et le résultat de l'estimation correspondant.



Trafic réel : **Moyen**  
Trafic estimé : **Fluide**



Trafic réel : **Fluide**  
Trafic estimé: **Fluide**



Trafic réel : **Dense**  
Trafic estimé: **Moyen**



Trafic réel : **Moyen**  
Trafic estimé: **Fluide**



Trafic réel : **Dense**  
Trafic estimé: **Dense**



Trafic réel : **Moyen**  
Trafic estimé: **Moyen**



Trafic réel : **Moyen**  
Trafic estimé: **Dense**



Trafic réel : **Fluide**  
Trafic estimé: **Fluide**



Trafic réel : **Dense**  
Trafic estimée: **Dense**

Fig 6.7- Exemples d'estimation du trafic routier

Les résultats précédents ont été obtenus sur une base contenant 254 vidéos dont 61 vidéos ont été réservées pour la phase d'apprentissage et 193 pour la phase de test. Pour une évaluation statistique plus poussée des performances de notre système, nous avons considéré un second scénario dans lequel la totalité de la base est divisée en 4 groupes. Chaque groupe représente 25% des vidéos de la base et contient 25% de la totalité des vidéos de chaque classe. La base contient une totalité de 44 exemples de trafic dense, 45 exemples de trafic moyen, et 165 pour le trafic fluide, nous prenons 25% de chaque classe pour former un groupe. On effectue donc 4 tests croisés, dans chaque test trois groupes sont utilisés pour l'apprentissage et le groupe restant est réservé pour le test. Nous représentons ci-dessous les résultats obtenus pour chaque test :

Tests	Résultats
Test1	90.47%
Test2	92.18%
Test3	93.75%
Test4	95.23%
<b>Moyenne</b>	<b>92.90%</b>

Tab 6.6- Résultats de la classification par SVM pour les 4 tests

Les résultats varient entre **90%** et **95%**, ceci prouve la robustesse de notre système. Le tableau suivant représente la table de confusion pour la totalité des tests.

<b>Prédites</b> <b>Réelles</b>	Fluide	Moyen	Dense
Fluide	165	0	0
Moyen	6	33	6
Dense	2	4	38

Tab 6.7- Table de confusion rassemblant les 4 tests

Les erreurs de classification apparaissent dans le cas du trafic dense ou moyen, tandis que pour le cas fluide les erreurs sont nulles. On conclut que ce système peut à 100% reconnaître un trafic fluide, à 74% un trafic moyen et à 86% un trafic dense. La prédiction du trafic moyen est importante pour prévenir les conducteurs de la qualité du trafic avant qu'il devienne dense. Si cet objectif n'a pas d'intérêt nous pouvons réduire le nombre de classes à deux classes seulement (le trafic fluide et dense) ; ceci améliorera, sans doute les résultats.

Les figures suivantes présentent les résultats de classification pour chaque test :

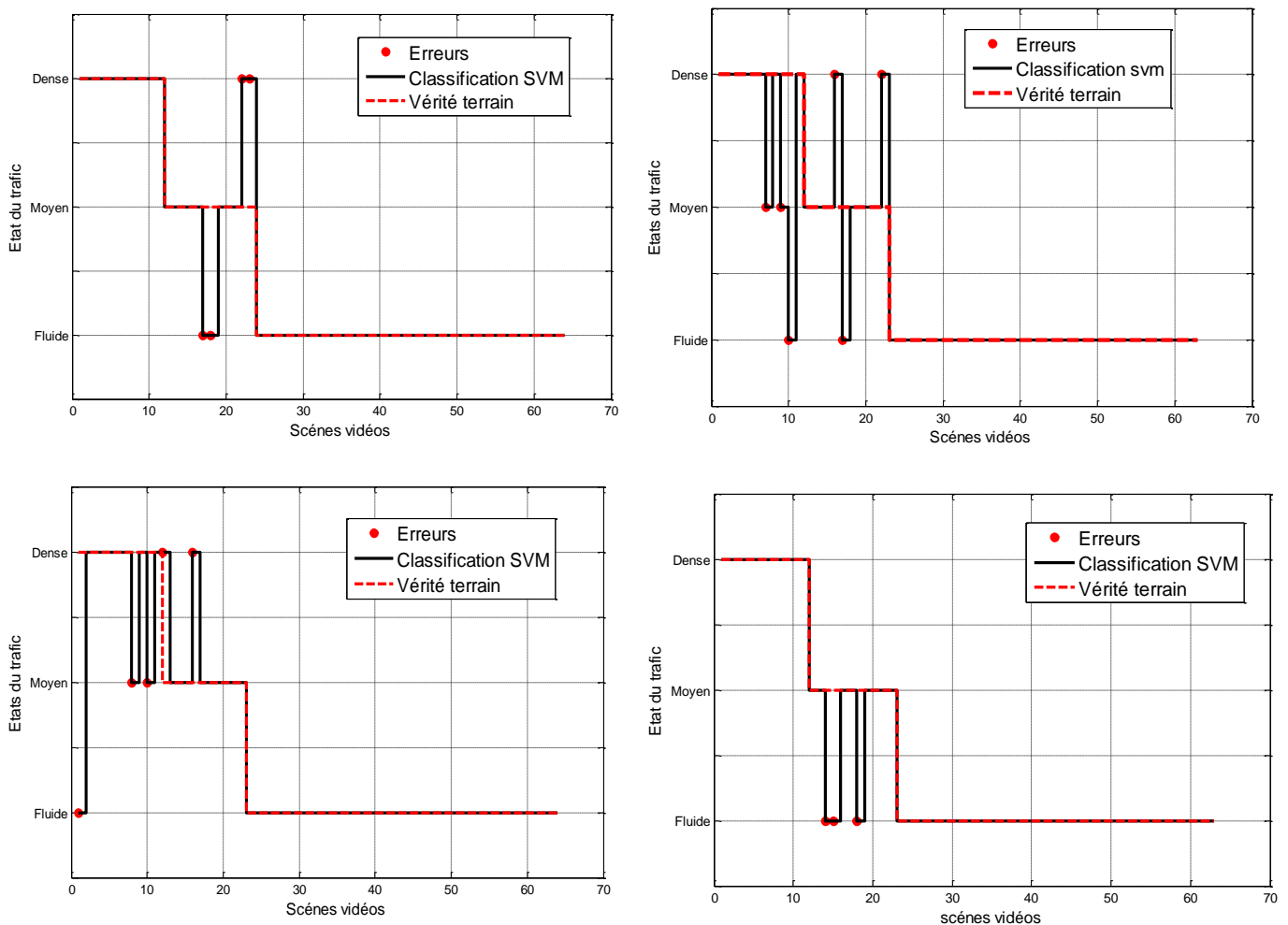


Fig 6.8- Estimation de l'état du trafic par SVM avec le schéma un contre tous pour les 4 tests.

La partie suivante sera consacrée à la présentation des résultats de la classification du trafic routier par l'utilisation du modèle de texture dynamique, décrit dans le chapitre 4.

## 6.2.2 Classification du trafic routier par l'utilisation du modèle de texture dynamique

Dans cette partie nous présentons les résultats de la classification du trafic routier par l'utilisation du modèle de texture dynamique. Nous avons vu comment calculer la distance de Kulback-Leibler entre deux textures dynamiques, cette distance est utilisée dans les classifieurs KPPV et SVM.

L'étape d'estimation des paramètres du modèle de texture dynamique pour chaque vidéo clip est initiée par un traitement des vidéos en niveaux de gris qui sont ensuite redimensionnées par la sélection d'une fenêtre de  $48 \times 48$  pixels qui représente la région où le mouvement est dominant. Ces régions sont appelées « patches ».

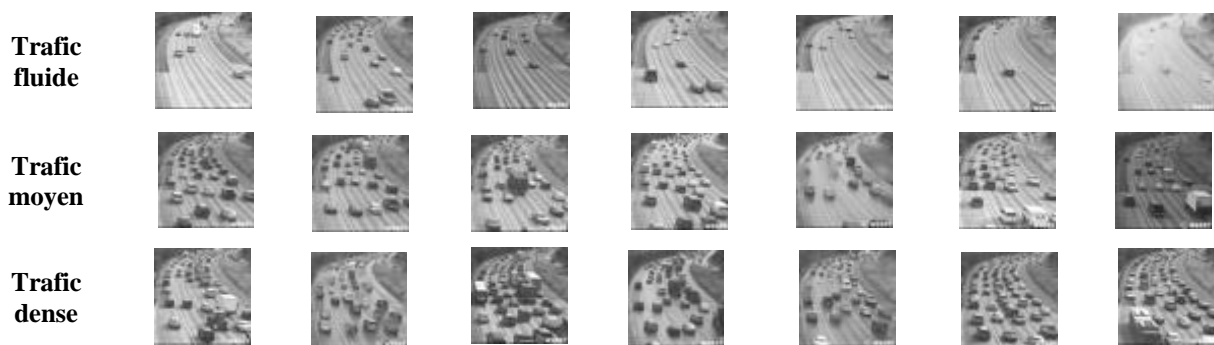


Fig 6.9- Exemples de patches utilisés

La figure (6.9) présente des exemples de patches utilisés, les trois lignes représentent respectivement des échantillons de trames pour le trafic fluide, moyen et dense.

### 6.2.2.1 Classification par KPPV

Dans un premier temps nous utilisons la méthode du plus proche voisin pour la classification du trafic basée sur le modèle de textures dynamiques.

Nous avons généré 250 images de synthèse de texture dynamique, et pour chaque image nous avons estimé la moyenne et la covariance d'états du modèle. Le nombre de composantes principales utilisé est  $n=15$ . L'algorithme utilisé est le suivant :



### 1- Extraction des paramètres du modèle de textures dynamiques

a- Création des patches ;

b- Estimation des matrices  $A$ ,  $B$ ,  $C$ ,  $X_0$  pour toutes les textures dynamiques

### 2- Classification

Pour  $i=1$  : nombre de scènes de test

Pour  $j=1$  : nombre de scènes d'apprentissage

Projeter l'espace d'état  $i$  dans l'espace d'état  $j$  ;

$\tau = 250$  ;

Estimer la moyenne et la covariance d'état  $x_1^T$  pour la texture dynamique  $i$  ;

Calculer la distance KL entre les espaces d'états  $i$  et  $j$  ;

Fin pour ;

Attribuer à la scène de test  $i$  la classe de la scène d'apprentissage  $j$  qui lui est la plus proche au sens de la distance KL ;

Fin pour.

Le choix du nombre de composantes principales est justifié par plusieurs tests qui ont montré que  $n=15$  est un choix judicieux, la figure suivante représente les résultats de la classification pour différentes valeurs de  $n$ .

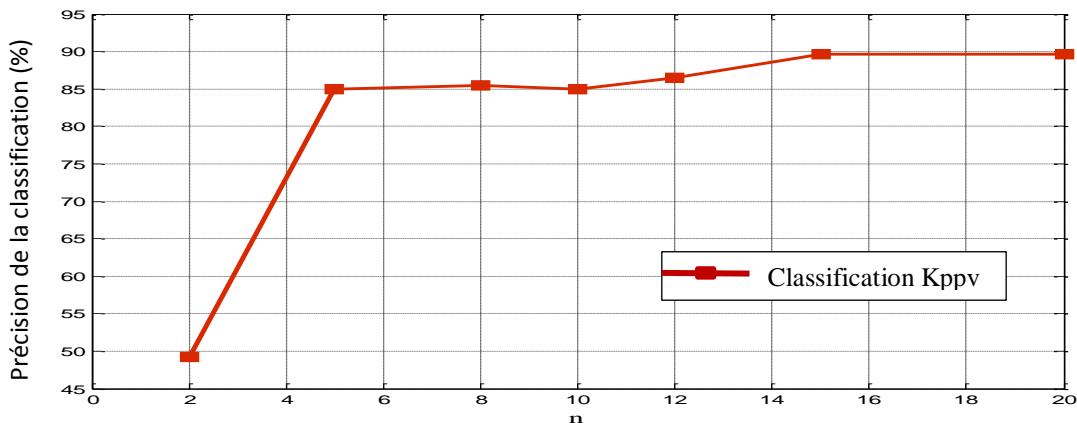


Fig 6.10- Taux de classification en fonction du nombre de composantes principales

Le taux de bonne classification augmente en fonction du nombre de composantes principales, et stagne à partir de  $n=15$ , où il atteint **89.63%**.

Nous présentons dans la figure suivante l'évolution du trafic routier estimé en utilisant la classification basée sur les textures dynamiques et le plus proche voisin. Sur la même figure est tracée l'évolution du trafic réel.

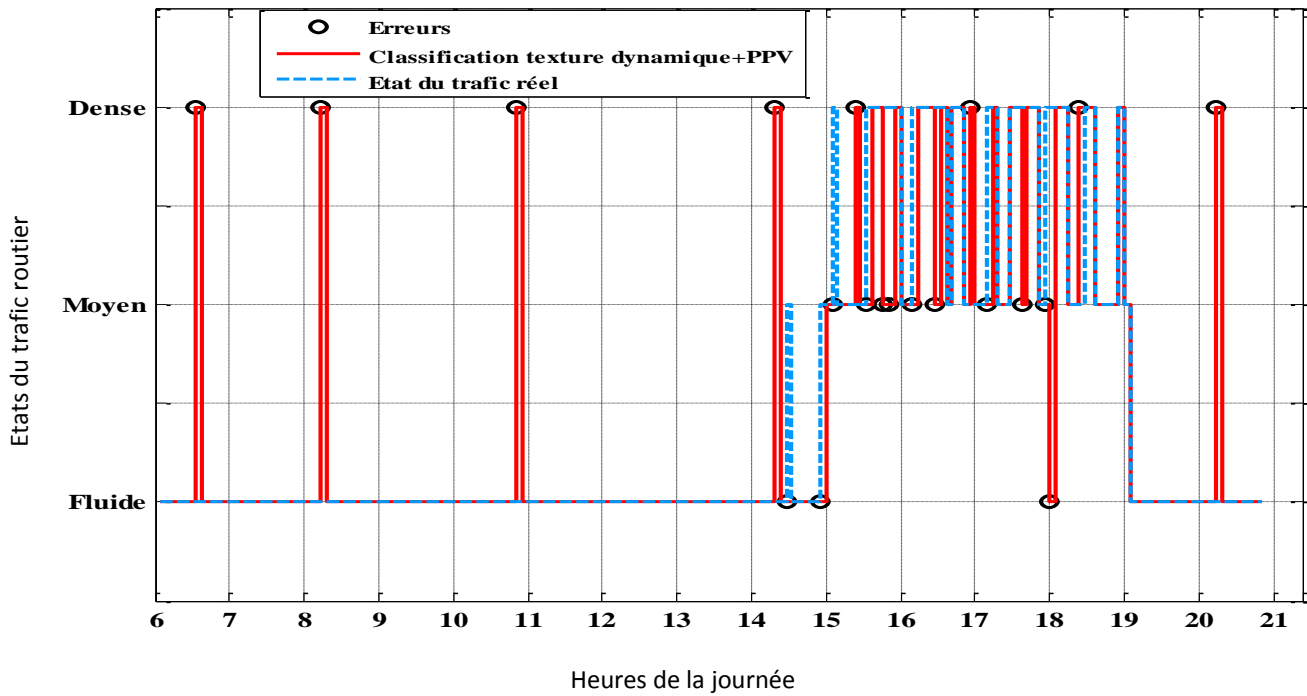


Fig 6.11- Estimation de l'état du trafic routier par textures dynamiques et KPPV

La matrice de confusion correspondante est donnée par le tableau suivant :

Prédites \ Réelles	Fluide	Moyen	Dense
Fluide	133	0	5
Moyen	2	21	3
Dense	1	9	19

Tab 6.8- Matrice de confusion pour la classification par textures dynamiques et KPPV

Le taux de bonne classification obtenu est acceptable. Nous appliquerons par la suite le SVM afin d'améliorer ce résultat.

### 6.2.2.2 Classification par SVM

Nous utilisons la même valeur pour le nombre de composantes principales  $n$  que précédemment. Le problème auquel nous avons été confrontés dans cette partie est le choix du noyau adéquat à nos données. Dans un premier temps nous avons testé les noyaux fournis par la bibliothèque LIBSVM, mais ceci n'a pas abouti à de bons résultats, finalement nous avons opté pour le noyau utilisé dans [CHV05], appelé noyau de Kulback-Leibler, qui a donné de meilleurs résultats.

#### ❖ Le noyau de Kulback-Leibler

Les résultats fournis par le SVM dépendent du noyau utilisé, comme nous l'avons vu dans le chapitre 5. L'utilisation d'un noyau consiste à projeter les données à l'aide d'une transformation non-linéaire de plus grande dimension. Une fonction noyau  $K(x_i, x_j)$  mesure la similarité entre deux points  $x_i$  et  $x_j$  dans un espace  $X$ . Le noyau le plus courant est le noyau gaussien défini par :

$$K_g(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (6.1)$$

Si les données d'apprentissage sont représentées par un modèle probabiliste (ex : texture dynamique), alors le noyau devient une mesure de similarité entre les distributions de probabilités. Un noyau probabiliste est alors défini par la projection :  $K: p \times p \rightarrow R$ , avec  $p$  un espace de distributions de probabilités. Un tel noyau est appelé noyau de Kullback-Leibler [CHV05], et est défini comme suit :

$$K_{KL}(p, q) = e^{-\gamma(D(p \setminus q) + D(q \setminus p))}, \quad (6.2)$$

où  $D(p \setminus q)$  est la distance de Kullback-Leibler, calculée entre les distributions de probabilités  $p(x)$  et  $q(x)$  :

$$D(p \setminus q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$$

Le noyau de Kullback-Leibler dans l'espace probabiliste est analogue au noyau gaussien dans un espace euclidien.

❖ Résultats de la classification

Nous avons testé le SVM avec le schéma de classification un contre tous. Le taux de bonne classification obtenu avec  $C=0.44$  et  $\gamma= 0.2$  est égale à **91.7%**.

La figure suivante présente l'évolution du trafic routier tel qu'estimé par le SVM :

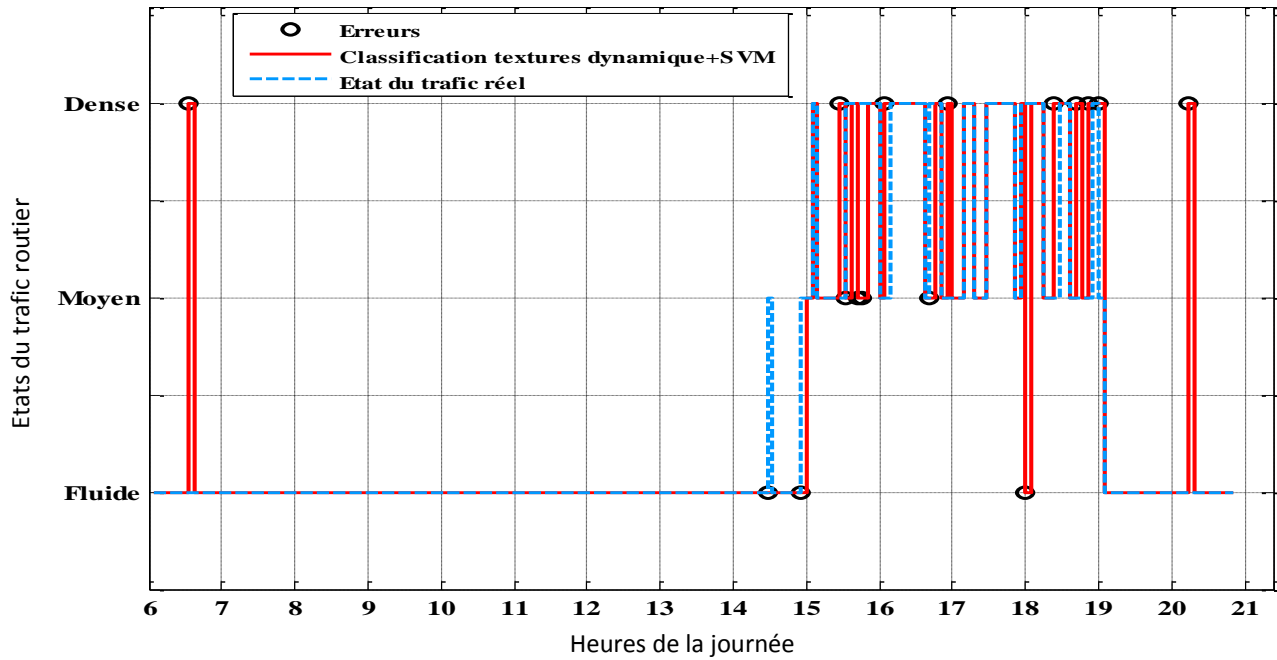


Fig 6.12- Estimation de l'état du trafic par textures dynamiques et SVM

La table de confusion est donnée ci-dessous :

Prédites Réelles \	Fluide	Moyen	Dense
Fluide	136	0	2
Moyen	2	17	7
Dense	1	4	24

Tab 6.9- Table de confusion pour la classification par textures dynamiques et SVM

On remarque que l'utilisation du SVM a permis d'améliorer les résultats de la classification, par rapport à l'utilisation du KPPV.

Comme nous l'avons fait précédemment, nous allons maintenant procéder à une évaluation statistique du système de classification par texture dynamique et SVM, en utilisant le second scénario.

Le tableau ci-dessous présente les résultats de la classification obtenus pour les 4 tests, en utilisant le KPPV et le SVM :

<b>Classifieur</b> <b>Test</b>	<b>KPPV</b>	<b>SVM</b>
<b>Test 1</b>	90.47%	93.65%
<b>Test 2</b>	87.50%	90.62%
<b>Test 3</b>	92.18%	92.18%
<b>Test 4</b>	93.65%	92.06%
<b>Moyenne</b>	90.95%	92.12%

Tab 6.10- Comparaison entre les résultats de la classification par textures dynamiques et KPPV ou SVM.

La figure suivante présente une comparaison entre ces deux classifieurs pour les 4 tests.

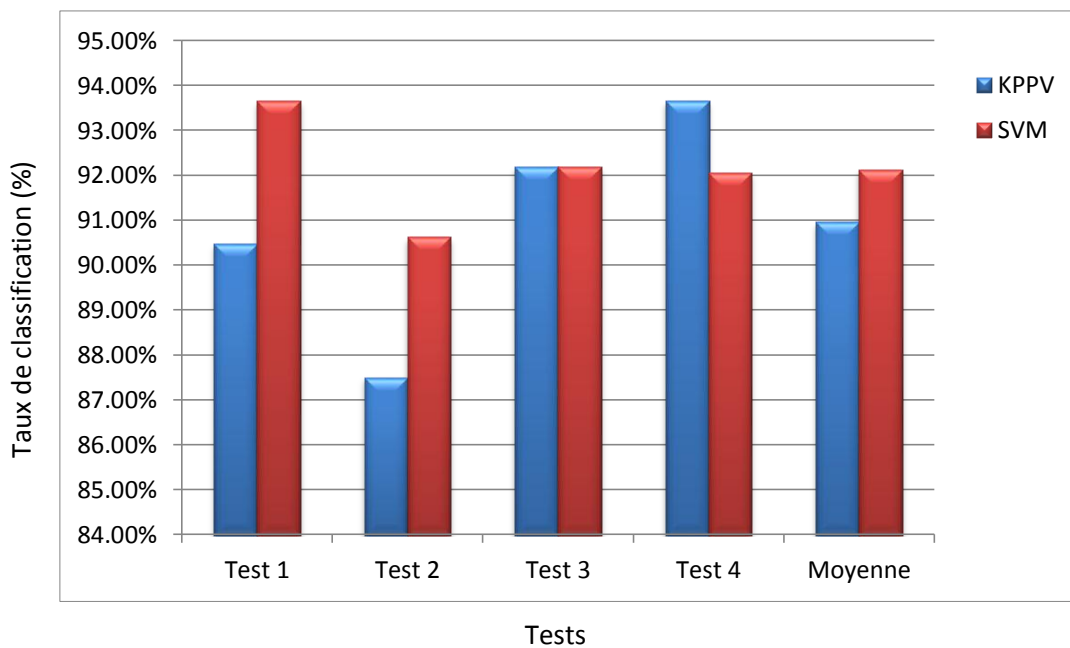


Fig 6.13- Comparaison entre la classification par textures dynamiques et KPPV ou SVM

Les figures suivantes présentent l'évolution du trafic estimé par texture dynamique et SVM, pour chaque test.

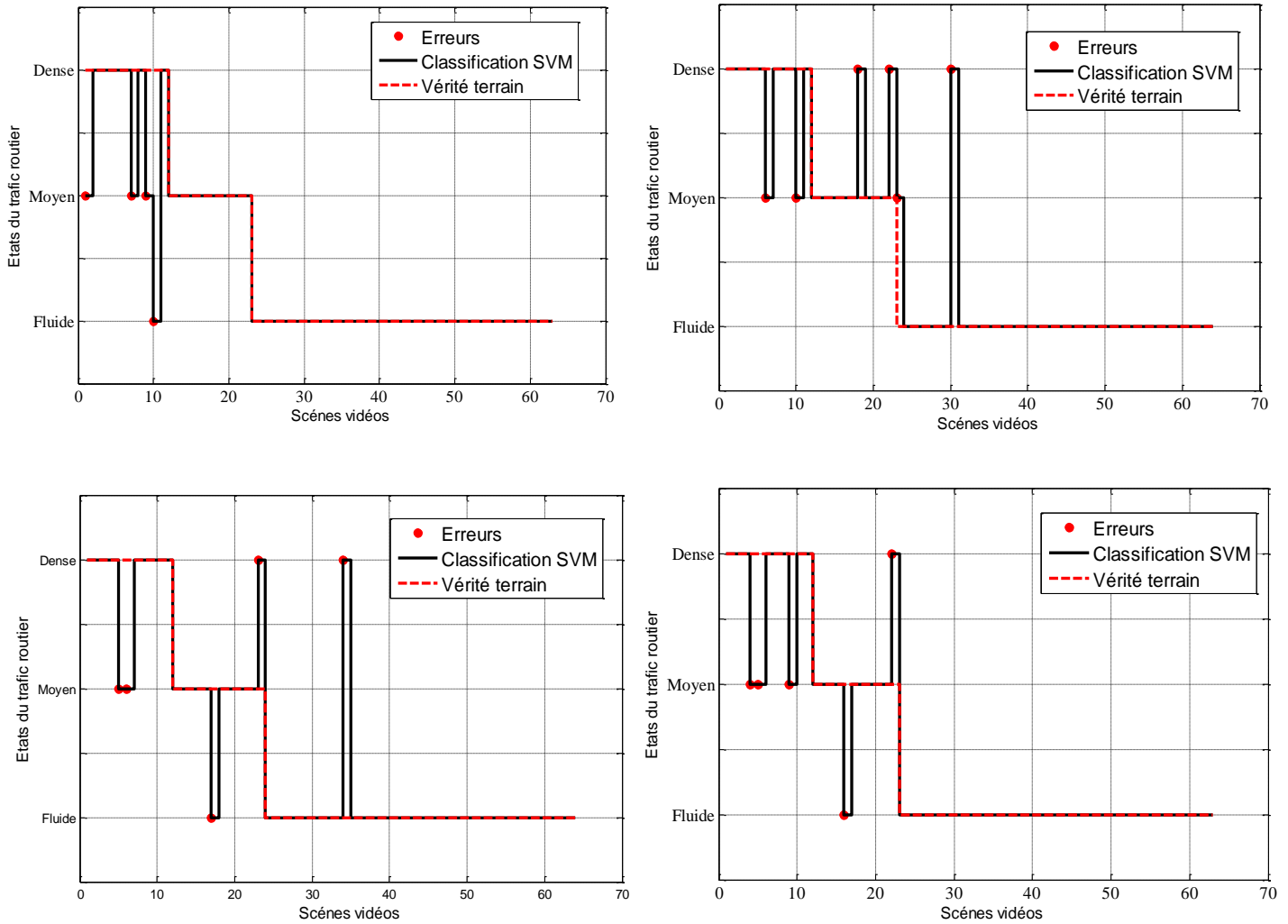


Fig 6.14- Estimation de l'état du trafic par textures dynamiques et SVM pour les 4 tests

La table de confusion globale pour les 4 tests est donnée dans le tableau suivant :

Prédites \ Réelles	Fluide	Moyen	Dense
Fluide	162	1	2
Moyen	2	39	4
Dense	1	10	33

Tab 6.11- Table de confusion rassemblant les 4 tests

Les résultats obtenus par la classification par texture dynamique et SVM pour les 4 tests varient entre **90%** et **93%**, la moyenne du taux de reconnaissance est de **92.12%**. Ce résultat est très proche du résultat obtenu par la classification par les vecteurs de mouvement et SVM.

Dans cette partie nous avons procédé à l'évaluation des performances d'une méthode d'estimation du trafic routier, basée sur un modèle de texture dynamique. Comparativement aux autres méthodes étudiées dans ce mémoire, cette méthode n'exige pas une détection et poursuite de mouvement ou une extraction des vecteurs de mouvement. Le SVM s'est avéré généralement plus performant que l'algorithme du KPPV puisqu'il permet de modéliser une relation complexe.

### **6.3 Conclusion**

Dans ce chapitre nous avons procédé à l'évaluation et la comparaison des performances de certaines méthodes de classification du trafic routier en trois classes : fluide, moyen, dense. Ces méthodes utilisent trois classifieurs, à savoir le KPPV, le LVQ et le SVM, alimentés par des paramètres caractérisant le trafic routier. Ces paramètres ont été extraits par différentes méthodes : méthode basée sur la détection et la poursuite de mouvement, méthode basée sur l'extraction des vecteurs de mouvement et enfin une méthode de modélisation du trafic routier par textures dynamiques.

Les meilleurs taux de classification correcte ont été obtenus par la méthode que nous avons proposée, qui utilise le classifieur SVM avec comme paramètres la vitesse par BM et la densité, et la méthode proposée dans [CHV05] qui utilise la modélisation par texture dynamique et le SVM. Comparativement à cette dernière, notre méthode donne de meilleurs résultats : 94.81% contre 91.7% dans le cas du premier scénario et 92.9% contre 92.12% dans le cas du second scénario.

Un taux de 94.81% est un taux très intéressant, et figure parmi les meilleurs taux publié dans la littérature à notre connaissance.

## **Conclusion générale**

Le travail présenté dans ce mémoire concerne l'étude du problème de l'automatisation de la classification du trafic routier à partir des vidéos numériques en utilisant des méthodes issues de la théorie de la vision et de l'intelligence artificielle. L'automatisation de cette tâche a été considérée dans le but de prévenir les conducteurs des situations difficiles du trafic routier.

Au cours des travaux menés sur cette problématique, plusieurs approches ont été testées pour l'extraction des paramètres de trafic routier ainsi que pour leurs classifications :

Les performances de la méthode d'extraction des paramètres de trafic routier par la détection et la poursuite du mouvement dépendent fortement de l'étape de détection de mouvement qui est sensible au bruit et à la luminosité.

L'utilisation des vecteurs de mouvement pour l'estimation des paramètres de trafic ne nécessite pas une segmentation et une poursuite de mouvement, mais cette méthode est sensible au mouvement de la caméra qui peut être causé par exemple par le vent.

L'analyse de textures est un problème délicat, dû essentiellement au fait qu'il n'existe pas de définition précise et rigoureuse permettant de caractériser complètement la notion de texture. La méthode de modélisation du trafic par textures dynamiques est robuste vis à vis des conditions climatiques et les changements de la luminosité.

Les SVM présentent de très bonnes performances en généralisation (c'est-à-dire la classification de nouveaux exemples de test). Intuitivement on sent que la marge joue en cela un rôle important. Cette zone correspond en fait à une zone d'incertitude. Pour améliorer ou affiner la marge de nouveaux points d'apprentissage sont à choisir.

Les performances de la méthode KPPV dépendent du choix de la distance utilisée, et du nombre de voisins. Contrairement aux autres méthodes, qui nécessitent un apprentissage



éventuellement long mais qui sont rapides en classification, cette méthode ne nécessite pas d'apprentissage et tous les traitements sont réalisés dans la phase de classification.

Pour la classification par le LVQ on mentionnera qu'il n'existe pas encore de théorie permettant de déterminer la structure optimale d'un réseau de neurones. En particulier la détermination du nombre de couches cachées et du nombre de neurones sur chacune d'entre elles relève encore largement de l'intuition de l'utilisateur et de sa capacité à expérimenter plusieurs architectures afin de retenir celle qui donne les meilleurs résultats. Plus le réseau est complexe, c'est-à-dire plus il comporte de neurones cachés et de liens synaptiques, plus il est capable de reconnaître les formes qui lui sont présentées à travers l'échantillon d'apprentissage. Mais malheureusement, l'accroissement de la complexité n'améliore pas nécessairement le taux de reconnaissance sur l'échantillon test.

En utilisant les paramètres vitesse par mise en correspondance (BM) et densité avec un classifieur SVM, nous avons obtenus des résultats très satisfaisants qui se comparent favorablement à ceux publiés dans la littérature.

A l'issue des travaux menés dans le cadre de cette thèse, nous citons quelques perspectives :

Comme nous l'avons constaté, l'estimation des paramètres du trafic routier est sujette à une certaine imprécision; nous pensons que le lissage de ces paramètres par un filtre, tel que le filtre de Kalman, en se basant sur un modèle de leur évolution, permettrait l'amélioration de cette estimation.

D'autre part les performances des méthodes que nous avons présentées ont été comparées en termes de bonne classification. Un autre critère de comparaison est la complexité. Une comparaison de ces méthodes en termes de complexité reste à faire.

La technique d'estimation des vecteurs de mouvement par la mise en correspondance des blocs est quelque peu lourde. Nous pensons que la mise en œuvre de cette technique peut être simplifiée pour la rendre plus rapide. Cependant, malgré cette lourdeur, il nous apparaît à première vue que la méthode que nous avons proposé, et qui est basée sur le classifieur SVM

avec comme paramètres les vecteurs de mouvement, est moins complexe que la méthode basée sur la modélisation par texture dynamique et le SVM, mais ceci reste à confirmer par une analyse quantitative de la complexité des deux méthodes.

Pour finir, nous pensons qu'il serait utile de confirmer les performances des méthodes d'estimation du trafic testées en utilisant d'autres bases de données.

## Références

- [BAC08] François Bardet et Thierry Château “ MCMC Particle Filter for Real-Time Visual Tracking of Vehicles” Proceedings of the 11th International IEEE, Conference on Intelligent Transportation Systems ,Beijing, China, pp 539 - 544 , 2008.
- [BEU90] Serge Beucher “ segmentation d’images et morphologie mathématique”, Thèse de doctorat en morphologie mathématique, Ecole national supérieure des mines de Paris.1990.
- [BHC09] M.Brulin, H.Nicolas, C.Maillet « Utilisation de la géométrie de la scène pour l’analyse du trafic routier » Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux, 2009.
- [CHK04] Sen-Ching Cheung, Chandrika Kamath “Robust technique for background subtraction in urban traffic video”, Center for Applied Scientific Computing Lawrence Livermore National Laboratory, Department of energy, University of California, 2004.
- [CHV05] Antoni B.Chan et Nuno Vasconcelos « Classification and retrieval of traffic video using auto-regressive stochastic process », IEEE Intelligent vehicles symposium, Las Vegas.2005.
- [CSI01] [www.csie.ntu.edu.tw/~cjlin/libsvm/](http://www.csie.ntu.edu.tw/~cjlin/libsvm/)
- [DAP00] Daniel Dailey,Suree Pumin member IEEE, F.w.cathey, and sure pumrin, ”an algorithm to estimate mean traffic speed using uncalibrated cameras” IEEE transactions on intelligent transporation systems ,vol.1,No.2, JUNE 2000.
- [DOR05] Gianfranco Doretto “Dynamic textures: Modeling, Learning, Synthesis, Animation, Segmentation, and Recognition” A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophyin Computer Science University of California Los Angeles. 2005.

- [DUB11] Sloven Duboi « Décompositions spatio-temporelles pour l'étude des textures dynamiques- Contribution à l'indexation vidéo » thèse de doctorat, Université de la Rochelle, 2011.
- [JIS10] Jin-Cyuan Lai, Shih-Shinh Huang “Image-Based Vehicle Tracking and Classification on the Highway”, National Kaohsiung First University of Science and Technology, IEEE 2010.
- [KDW10] Konstantinos G, Derpanis, Richard P.Wildes « Classification of traffic Video Based on a Spatiotemporal Orientation Analysis » Department of Computer Science and Engineering, York University, IEEE, PP 606-613, 2010.
- [LEJ08] Christine LEIGNEL , Jean-Michel JOLION « Étude bibliographique suivi de personnes et trajectoire dans un réseau de caméras », Projet ANR Canada , Laboratoire d'informatique en Image et Systèmes d'information. Université Claude Bernard Lyon 1.
- [MSF07] Ehsan Adeli Mosabbeeb , Maryam Sadeghi , and Mahmoud Fathy “ A New Approach for Vehicle Detection in Congested Traffic Scenes Based on Strong Shadow Segmentation” Springer-Verlag Berlin Heidelberg 2007.
- [NEP92] Randal C-Nelson et Ramprasad Polana “Recognition of Motion from Temporal Texture” IEEE, 1992.
- [PAR06] Marc Parizeau “ réseaux de neurones” rapport, université de LAVAL, 2006.
- [POL04] Fatih Porikli and Xiaokun Li “Traffic Congestion Estimation Using HMM Models Without Vehicle Tracking” Mitsubishi Electric Research Laboratories, Inc, 2004.
- [PPT06] W. Pattara-atikom, P Pongpaibool, and S. Thajchayapong “Estimating Road Traffic Congestion using Vehicle Velocity” 6th International Conference on ITS Telecommunications Proceedings,pp 1001-1004,2006.
- [PTN07] Panita Pongpaibool , Poj Tangamchit , Kanokchai Noodwong « Evaluation of Road Traffic Congestion Using Fuzzy Techniques » University of Technology Thonburi, THAILAND,IEEE,2007.

- [RAH10] H. A. Rahim, U. U. Sheikh, « Vehicle velocity estimation for traffic surveillance system », World Academy of science, Engineering and technology 69- 2010.
- [SCS01] Bernhard Scholkopf ,Alexander J. Smola « learning with kernel - Support Vector Machines, Regularization, Optimization, and Beyond » livre, Massachusetts Institute of Technology, The MIT Press .Décembre 2001.
- [VAC95] Corinne VACHIER « Extraction de caractéristiques, segmentation d’image et Morphologie mathématique », thèse de Doctorat, Ecole nationale supérieure des mines de paris. 1995.
- [VAH10] Vamsi Krishna Madasu and M. Hanmandlu “Estimation of Vehicle Speed by Motion Tracking on Image Sequences”, IEEE Intelligent Vehicles Symposium University of California, San Diego, CA, USA, pp 185-190, 2010.
- [WIK12] [www.wikipedia.org/wiki/ méthode\\_des\\_k\\_plus\\_proches\\_voisins](http://www.wikipedia.org/wiki/méthode_des_k_plus_proches_voisins)
- [XGZ05] Lei Xie, Guangxi Zhu, Zhenming Zhang “Real-time Vehicles Tracking Based on Kalman Filter in a Video-based ITS” Huazhong University of Science and Technology, IEEE, PP 883- 886, 2005.
- [XIJ09] Xiaojun Shen, Jun Chen “Study on Prediction of traffic congestion Based on LVQ Neural Network”, International Conferance on Measuring Technology and Mecatronivs Automation, IEEE, PP 318-321,2009.
- [XYA10] Xinting Pan,Yunlong Guo, Aidong Men “Traffic Surveillance System for Vehicle Flow Detection” , IEEE Second International Conference on Computer Modeling and Simulation, China. pp 314-318, 2010
- [YUJ97] B. Yu, A. Jain "Lane boundary detection using a multi resolution Hough transform". In Proceedings of ICIP 97, Vol. 2, pp. 748 – 751. 1997.
- [ZYZ09] Zhe Liu, Yangzhou Chen, Zhenlong Li “Camshift-based Real-time Multiple Vehicles Tracking for Visual Traffic Surveillance”, World Congress on Computer Science and Information Engineering, 2009.