

Utilisation de la géométrie de la scène pour l'analyse du trafic routier

M. Brulin^{1,2}, H. Nicolas¹

C. Maillet^{1,2}

¹ LaBRI (Laboratoire Bordelais de Recherche en Informatique)

Université de Bordeaux I,
351, Cours de la Libération,
33405 Talence Cedex – France

{brulin, nicolas}@labri.fr

² Adacis Sarl,

5, Ferreau Sud,
33320 Bayas – France

maillet@adacis.net

Résumé

Cet article présente un système d'analyse de trafic routier dans un contexte de vidéo-surveillance. La première étape de notre 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. Nous nous plaçons dans le cas d'une seule caméra sans connaissance a priori sur les paramètres intrinsèques et extrinsèques la caractérisant.

Mots clefs

Analyse du trafic routier, modélisation de la scène, suivi d'objets.

1 Introduction

L'analyse du trafic routier par vision par ordinateur est un domaine de recherche en pleine expansion dont les applications sont essentielles pour la gestion, la sécurité et la compréhension du trafic. L'utilisation d'une ou plusieurs caméras est un choix économique qui offre les données nécessaires à de nombreuses fonctions, telles que le comptage et la détection d'anomalies ou d'accidents. Malgré les nombreuses recherches dans le domaine (ex. [1], [2]), l'élaboration d'un système robuste, efficace et automatique reste un challenge complet dû à la perte d'information causée par la projection perspective lors de la formation des images par le capteur. Plus récemment, des systèmes ont été proposés utilisant des caractéristiques de la route afin d'améliorer les résultats. Par exemple, Maduro et al. [3] utilisent les bordures des voies pour synthétiser une vue du dessus de la scène afin

de segmenter les objets. Nous proposons ici de récolter des informations sur la scène afin d'aider à la segmentation.

La première étape de notre système consiste à estimer des informations sur la géométrie de la scène ainsi que le mouvement global. Les bordures des voies sont déterminées et l'information de profondeur est utilisée pour diviser les voies en sous-régions. La seconde étape effectue l'analyse du trafic, qui consiste en quatre phases : la soustraction d'arrière-plan, l'extraction des objets, leur suivi puis l'analyse de comportement. Les informations sur la géométrie de la scène sont exploitées dans l'extraction des objets et leur suivi en définissant à l'intérieur des objets des pixels dits *ambigus* et en découpant les objets temporairement à l'aide des délimitations des voies. Le processus de suivi prend en compte cette ambiguïté en utilisant une information temporaire.

Le système proposé n'utilise qu'une seule caméra placée au dessus des voies et nous ne connaissons aucune information a priori sur les paramètres intrinsèques ou extrinsèques de la caméra.

2 Modélisation de la scène

La modélisation se déroule en 4 étapes qui sont combinées afin d'obtenir une estimation de la géométrie (Figure 1). Dans un premier temps, une image d'arrière-plan est estimée et utilisée afin d'obtenir une carte de contours et en extraire les bordures des voies. Les résultats obtenus, combinés avec une détection de mouvement, permet de délimiter les zones correspondants aux voies. Le point de fuite de la scène est ensuite estimé et utilisé pour extraire les lignes de profondeur. Enfin, la dernière étape combine les zones obtenues, le mouvement et sa direction, pour

obtenir le model final des voies : chaque voie est découpée en sous-région en fonction des lignes de profondeur. Toutes ces informations sont utilisées dans la section 3 pour améliorer la segmentation et le suivi des objets et permettre le comptage.

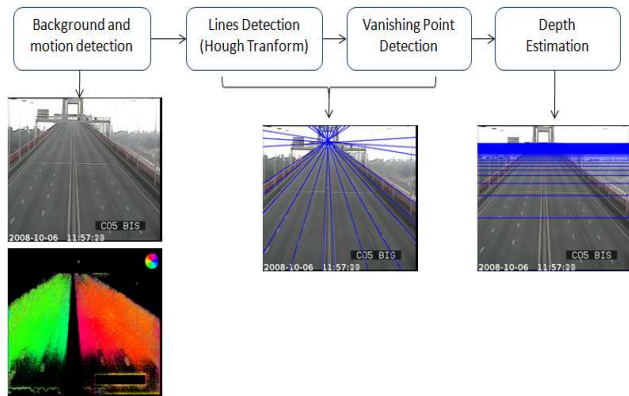


Figure. 1 – Les quatre étapes dans la modélisation de la scène.

Estimation de l'arrière-plan. L'information d'arrière-plan est obtenu en utilisant une modélisation par mixture de gaussiennes [4]. Cette modélisation consiste en une somme pondérée de lois gaussiennes, où les distributions couleur de chaque pixel peuvent être multi-modales (pour une étude approfondie, le lecteur intéressé pourra se référer à [5]). Le nombre de gaussiennes et leurs paramètres sont estimés et mis à jour par un algorithme Expectation-Maximization et seules les gaussiennes possédant un poids élevée et un faible écart-type sont considérées comme appartenant à l'arrière-plan (seuillage selon un critère sur le rapport poids - écart-type). Tous les pixels isolés sont supprimés et une détection d'ombre portée est appliquée en utilisant l'approche proposée par Horprasert et al. [6].

Détection de mouvement. Chaque nouvelle image de la vidéo est comparée à l'arrière-plan pour obtenir une première estimation des zones de l'avant-plan. Une étiquetage par analyse en composantes connexes [7] permet d'extraire les régions en mouvement. Pour chaque région, le flot optique est estimée en utilisant la méthode pyramidale proposée par Lucas Kanade [8]. L'ensemble des points d'intérêt [9] sont mis en correspondance dans l'image suivante, et les vecteurs de déplacement (norme et orientation) sont conservés.

Détection des voies. La détection de lignes appliquée ici est similaire à celle proposée dans [10]. Dans un premier temps et pour réduire le bruit, l'image d'arrière-plan est filtrée par un filtre médian (qui possède l'avantage de conserver les contours). L'information de contour est ensuite récupérée à l'aide du détecteur de Canny. Afin d'effectuer localement la détection, l'image contenant les contours est découpée en blocs de tailles égales. Pour

chaque bloc, les lignes sont extraites en utilisant une transformée de Hough [11].

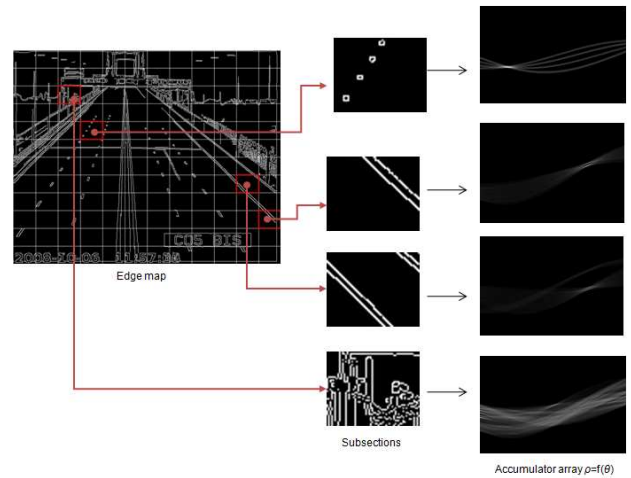


Figure 2 – Détection locale des bordures des voies. La carte des contours est découpée en blocs homogènes. Pour chaque bloc, la matrice d'accumulation (espace de Hough) est construite (un point de l'image contour correspond à une sinusoïde dans l'espace de Hough). Par extraction des maximum locaux de l'accumulateur, on obtient les paramètres des lignes correspondantes aux bordures des voies.

Estimation du point de fuite. En théorie, toutes les bordures de voies convergent vers un point de fuite. Il est donc intéressant d'estimer ce point pour valider et supprimer les lignes précédemment détectées. Le point de fuite est estimé en utilisant l'approche de Matessi et Lombardi [12] qui exploitent l'espace de Hough (représentation polaire). Leur méthode consiste à minimiser la fonction :

$$\min_{x_0, y_0} \sum_{i=1}^n W_i \cdot (\rho_i - x_0 \cdot \cos(\theta_i) - y_0 \cdot \sin(\theta_i))^2$$

Où $W_i = v_i/V$, v_i est le nombre de votes dans l'espace des paramètres (lignes paramétrée par (ρ_i, θ_i)) et V est le nombre total de votes. Une fois le point estimé, les lignes qui sont éloignées de ce point (en utilisant un seuil) sont supprimées.

Estimation des lignes de profondeur. Le but de cette étape est d'estimer les lignes de profondeur dans la scène. Théoriquement, la profondeur d'un point physique (relative à la caméra) dépend de sa position verticale sur l'image [13] tel que:

$$z = h_{cam} \cdot \tan \left(\tan^{-1} \left(\left(y_s - \frac{d}{2} \right) / f \right) + \alpha_{cam} \right)$$

où y_s représente la position du point sur le capteur, d est la longueur du capteur, f sa distance focale, z est la distance projetée du point sur le sol à la caméra, h_{cam} la hauteur de la caméra et α_{cam} son orientation. La première étape consiste à estimer α_{cam} en utilisant un algorithme des

moindres carrés. Puisque f et d sont inconnus, on suppose des valeurs standard et fixons $f=35\text{mm}$ et $d=24\text{mm}$. La position de la caméra h_{cam} est seulement un coefficient qui relie la position au sol z et la position sur le capteur y_s , nous le fixons à 5m. Un exemple de lignes obtenues est affiché sur la Figure 1.

Une fois l'ensemble de ces informations collectées, elles sont combinées afin d'obtenir le modèle final des voies. Chaque voie est délimitée (par les lignes détectées) et découpée en sous-bandes à l'aide des lignes de profondeur. Les informations de mouvement (norme et orientation des vecteurs de déplacement) sont conservées pour chaque sous-bande (Figure 3). Si très peu de mouvement a été observé dans une zone (caractérisé par de faibles normes des vecteurs de déplacement), alors cette zone est ignorée (considérée à l'extérieur de notre zone d'intérêt).

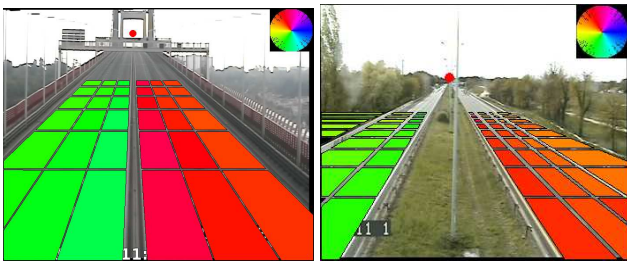


Figure 3 – Modélisation des voies de la scène. Les sous-régions, déterminées à l'aide des lignes de profondeur et du point de fuite possèdent en théorie la même surface dans le monde réel. Les vecteurs de mouvement estimés (norme et orientation) ont été conservés et constituent une information utile pour l'étape d'analyse de comportement. La palette de couleur utilisée pour l'orientation des vecteurs mouvement est représentée en haut à droite de l'image.

3 Analyse du trafic routier

L'analyse du trafic comprend les étapes suivantes (Figure 4) :

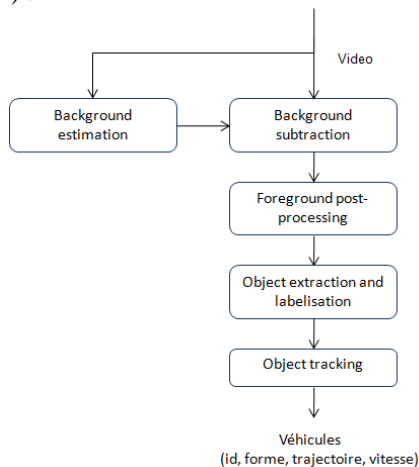


Figure 4 – Etapes du système pour l'analyse du trafic.

Tout d'abord, l'image d'avant-plan est obtenu par soustraction d'arrière-plan. Le résultat est ensuite filtré (suppression de l'ombre et des pixels isolés) et les objets en mouvement sont extraits en prenant en compte les bordures des voies. Le suivi des objets extraits permet ensuite de conserver l'identité des objets, ce qui permet d'en effectuer le comptage.

Extraction des objets. Une fois l'image d'avant-plan obtenue (cf. Section 2), les régions en mouvement sont extraits en utilisant un algorithme en composantes connexes [7]. La position au sol des objets est définie comme étant le centre du segment correspondant à la projection de la région sur la ligne perpendiculaire à la direction de la voie (sur laquelle se situe l'objet). Cette projection (le long de la ligne définie pour chaque point de la région et du point de fuite) nous donne un segment que nous appellerons PL (*projected line*). Pour des raisons de simplicité et pour atteindre un traitement en temps réel, chaque région est approximée par sa boîte englobante, et seule sa couleur moyenne et son écart-type sont conservées. Pour de petits objets (véhicules) se déplaçant dans une voie (c.a.d. sans dépassements), la majorité des pixels sont localisés à l'intérieur de cette même voie sur l'image. Tandis que pour des objets plus volumineux (camions par exemple), un nombre plus important peuvent être sur plusieurs voies, ce qui pose problème pour la localisation de l'objet.

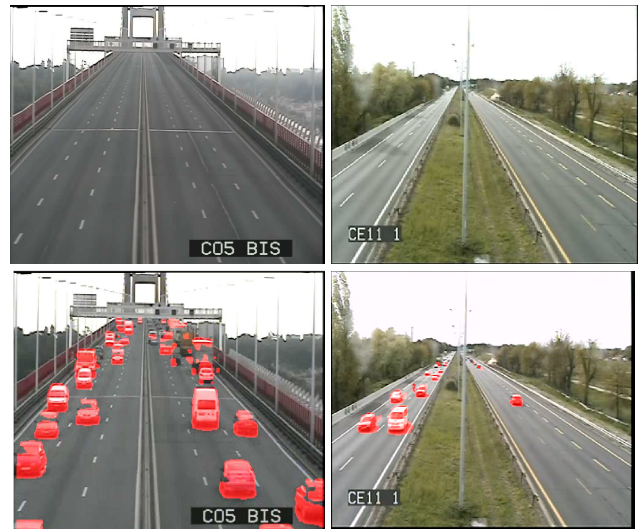


Figure 5 – (en haut) Exemples d'arrière-plan obtenus en modélisant l'évolution des couleurs de chaque pixel par une mixture de gaussiennes. (en bas) Les zones estimées en mouvement sont affichées en rouge.

Une première solution pour positionner l'objet correctement consiste à affecter l'objet à la voie dans laquelle la majorité des pixels se trouvent. Cependant, lorsque deux véhicules sont proches, ils sont souvent (par erreur) détectés comme un seul et même objet.

Pour contrer ce problème, chaque pixel des régions est classifié en deux classes : pixel *correct* ou *ambigu*, où les pixels *ambigus* sont ceux qui n'appartiennent pas au numéro de la voie qui a été associé à l'objet.

L'ambiguïté est levée en utilisant quatre critères :

- 1- Le rapport d'aire entre les pixels *ambigus* et la taille total de l'objet.
- 2- Le rapport entre la longueur du segment PL (*projected line*) et la largeur de la voie.
- 3- La différence euclidienne des distributions de couleur entre les pixels *ambigus* et *corrects* (un véhicule étant généralement globalement uniforme en couleur).
- 4- La différence entre les vitesses moyennes des pixels *ambigus* et *corrects*.

Pour chaque critère, les seuils ont été déterminés expérimentalement. Si au moins trois critères sont vérifiés, alors la région est découpée selon les bordures des voies (Figure 7).



Figure 6 – Exemple d'extraction d'objets sur l'image d'avant-plan.

Un véhicule qui change de voie est découpé en deux alors qu'il ne devrait pas l'être. On conserve donc l'état original du véhicule pendant le suivi afin d'être capable de revenir en arrière et corriger si nécessaire (durant l'étape de *suivi d'objets*).

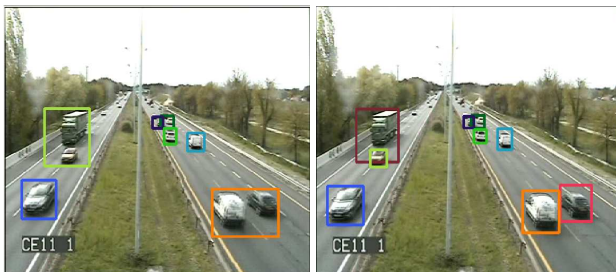


Figure 7 - (à gauche) Exemple d'erreur de segmentation rencontrée lorsque deux véhicules sont proches. Les régions orange et verte contiennent deux objets, mais seul un seul objet est détecté. (à droite) Correction de la segmentation en utilisant les bordures des voies.

Suivi d'objets. Le but de cette étape est de suivre un nombre variable de cibles et de maintenir leurs identités tout en prenant en compte les détections partielles et les occlusions. Le problème de suivi multi-cible est modélisé par un graphe pondéré dans lequel chaque nœud représente un objet détecté (Figure 8). Pour deux images successives, le graphe est réduit à un graphe biparti : Chaque nœud O_n^{t-1} du graphe G_{t-1} (à l'instant $t-1$) est relié à chaque nœud O_m^t du graphe G_t (à l'instant t). Un poids $w_{n,m}$ est associé à chaque arête du graphe et caractérise la similarité entre les objets.

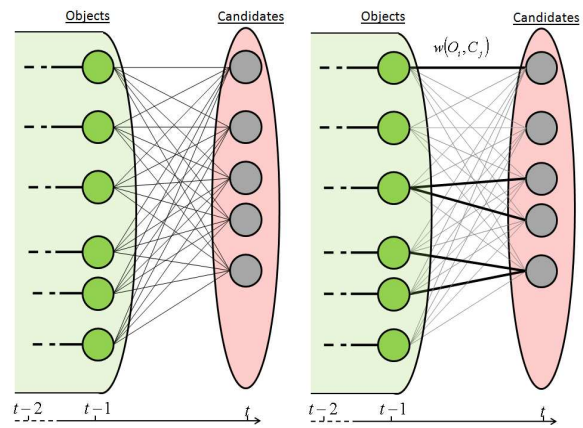


Figure 8 – (à gauche) Modélisation du suivi d'objets par un graphe pondéré. Chaque nœud à l'instant $t-1$ est relié à chaque candidat potentiel à l'instant t . (à droite) Seules les arêtes dont le poids est supérieur à un seuil sont conservées.

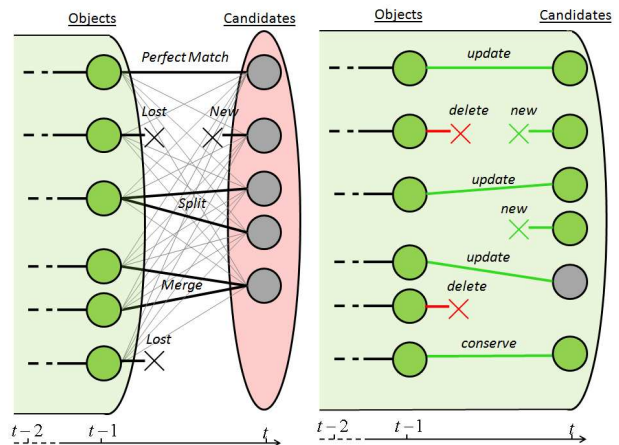


Figure 8 – (à gauche) Hypothèses d'association parmi les cinq cas possibles. (à droite) Décision et mise à jour des objets selon les hypothèses d'association.

Le processus d'association consiste à associer les objets suivis O_{t-1} (à l'instant $t-1$) avec les objets O_t détectés à l'instant t . Ce processus comporte quatre étapes :

1. La première étape est une étape de prédiction, dans laquelle la nouvelle position de l'objet est estimée à partir de sa vitesse.
2. La seconde étape est une mesure de similarité entre les objets permettant d'affecter un poids à chaque arête du graphe. Il s'agit d'une distance euclidienne entre les propriétés des objets (position, couleur, écart-type). Seules les arêtes dont le poids est supérieur à un seuil sont conservées.
3. La troisième étape concerne l'attribution d'hypothèses d'association pour chaque objet parmi les cinq cas possibles : *Correspondance Parfaite*, *Perdu*, *Nouveau*, *Split*, *Merge*.
4. La dernière étape met à jour les objets selon les hypothèses faites à l'étape 2.

Cette dernière étape (décision et mise à jour) se déroule de la façon suivante. Dans le cas d'une *Correspondance Parfaite*, l'objet est mis à jour avec le candidat correspondant. Un objet *Perdu* est supprimé s'il est *Perdu* suffisamment longtemps. Sinon, il est conservé et on continue de le suivre. Les *Nouveaux* objets sont directement ajoutés au graphe en temps que nouveaux nœud. Les cas de *Split* et de *Merge* sont résolus à l'aide d'informations temporelles, dans les deux cas, on conserve les objets originaux qui sont mis à jour avec les candidats ayant la meilleure similarité et les objets résultants sont créés et considérés comme *Nouveaux*, mais leur existence reste en suspend.

La décision finale est obtenue lorsque l'association de ces derniers ne change pas pendant un certain temps. Si deux objets sont séparés (*Split*) et se rejoignent de nouveau (*Merge*), seul le suivi de l'objet original est conservé. Deux objets qui ont subis un regroupement (*Merge*) ne sont validés que s'ils ont eu la même vitesse pendant la période de temps correspondant au déplacement d'un objet sur la moitié de la longueur de la voie (la partie supérieure de la route n'étant pas significative puisque les objets sont trop petits).

Comme mentionné dans la tâche d'*extraction d'objets*, lorsqu'un objet est découpé (par les bordures des voies), l'original est conservé et les objets résultants sont suivis indépendamment. Pour corriger les susceptibles erreurs de découpage, on utilise l'estimation de leur vitesse. S'ils ont conservés la même vitesse, alors ils sont de nouveaux regroupés. Le résultat de la détection d'objets et leurs trajectoires est montré sur la Figure 9.

Remarque. Les véhicules à l'arrêt ou en contre-sens sont détectés en utilisant les vitesses des objets qui sont comparées aux orientations globales des sous-voies. Si un véhicule est détecté comme étant à l'arrêt, la mise à jour de l'arrière plan est suspendue afin d'éviter l'incrustation de l'objet dans l'arrière-plan. Une classification grossière utilisant les tailles des objets (Figure 10) est faite pour chaque sous-voie en calculant quelques statistiques sur les tailles des objets qui ont été suivis. Ainsi chacune des voies construit son propre histogramme des tailles des objets. L'approche proposée est appliquée sur deux séquences de test dans lesquelles la vérité terrain a été obtenu manuellement. Le comptage de véhicules ainsi que leur classification sont effectués pour chaque voie. Les objets détectés et non-détectés sont comptés (un objet est considéré comme détecté s'il est correctement suivi durant la séquence, c.a.d durant la période de temps correspondant au déplacement d'un objet sur la moitié de la longueur de la voie), puis comparés à la vérité terrain, et on vérifie si les véhicules ont été détectés dans la bonne voie (Table I).

L'utilisation d'un découpage à l'aide des bordures des voies a permis d'améliorer la précision sur la détection (de 92% à 96% pour la première séquence et de 90 à 93% pour la seconde séquence), sur la localisation (de 90% à 93% pour la première séquence et de 93 à 96% pour la seconde séquence) et sur la classification (de 89% à 91% pour la première séquence et de 93 à 94% pour la seconde séquence).

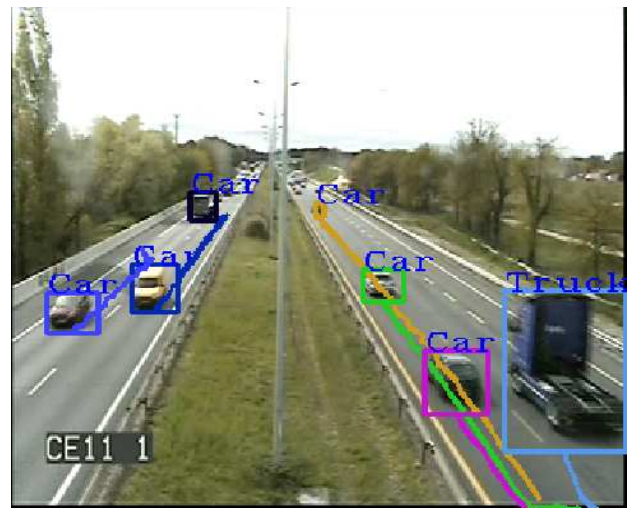


Figure 9 – Résultat de la détection des objets, leur suivi (trajectoires) et de leur classification en fonction des histogrammes de tailles de chaque sous-voie.

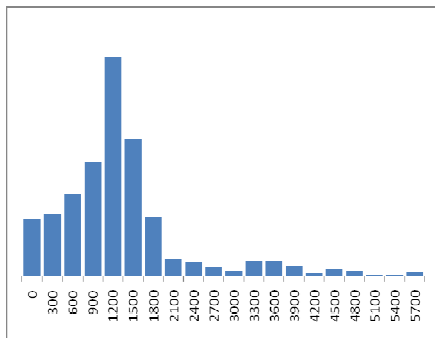


Figure 10 – Exemple d’histogramme des tailles des objets détectés pour une sous-région de la scène. Chacune des sous-régions construit son propre histogramme utile à la classification.

Séquence 1 (20 minutes)	Vérité Terrain	Précision Détection	Précision sur la position	Précision Classification
Sans découpage	224	92%	90%	89%
Approche proposée	224	96%	93%	91%

Séquence 2 (30 secondes)	Vérité Terrain	Précision Détection	Précision sur la position	Précision Classification
Sans découpage	30	90%	93%	93%
Approche proposée	30	93%	96%	94%

Table I – Résultat du comptage de véhicules pour les deux séquences de test.

4 Conclusion

Nous avons présenté un système d’analyse de trafic capable de compter et détecter les véhicules à l’arrêt ou en contre-sens. En définissant des pixels ambigus pour les objets détectés, le système utilise la modélisation de la scène pour améliorer la détection et le suivi des objets pendant le processus. Les résultats obtenus sont prometteurs et montrent la robustesse du système proposé à suivre plusieurs objets simultanément et à corriger les erreurs de segmentation. La simplicité de l’approche permet une implémentation facile et atteint un traitement en temps réel.

Références

[1] Z. Zhu and G. Xu, VISATRAM: A real-time vision system for automatic traffic monitoring. *Image Vis. Comput.*, vol. 18, no. 10, pp. 781-794, 2000.

[2] M. Yu and Y. D. Kim, Vision based vehicle detection and traffic parameter extraction. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E84A, no. 6, pp. 1461–1470, 2001.

[3] C. Maduro, K. Batista and J. Batista, Estimating Vehicle Velocity Using Image Profiles on Rectified Images. *IbPRIA*, pp. 64-71, 2009.

[4] C. Stauffer and W. Grimson, Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, 1999.

[5] T. Bouwmans, F. El Baf and B. Vachon, Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. *Recent Patents on Computer Science* 1, 3, pp. 219-237, 2008.

[6] T. Horprasert, D. Harwood, and L. S. Davis, A statistical approach for real-time robust background subtraction and shadow detection. in *ICCV Frame-Rate WS*, pp. 1–19, 1999.

[7] S. Suzuki and K. Abe, Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics and Image Processing*, 30: pp. 32-46, 1985.

[8] C. Tomasi and T. Kanade, Detection and tracking of features points. *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991.

[9] J. Shi and C. Tomasi, Good Features To Track. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, June 1994.

[10] Y. Wang, E. Teoh, and D. Shen, Lane detection and tracking using b-snake. *Image and Vision Computing*, 22(4):269–280, 2004.

[11] J. Illingworth, and J. Kittler, A survey of the Hough transform. *Comput. Vision Graph. Image Process.* 44, 1, 87-116, Aug. 1988.

[12] A. Matessi and L. Lombardi, Vanishing Point Detection in the Hough Transform Space. *Proceedings of the 5th international Euro-Par Conference on Parallel Processing*, vol. 1685, pp. 987-994, 1999.

[13] C. Käs and H. Nicolas, Rough compressed domain camera pose estimation through object motion, *ICIP'09*, Cairo, Egypt, 2009.