



HAL
open science

Mesure de Trajectoire par Stéréo-Vision pour des Applications de Suivi de Véhicules à Basse Vitesse

Julien Morat, Frédéric Devernay, Sébastien Cornou

► **To cite this version:**

Julien Morat, Frédéric Devernay, Sébastien Cornou. Mesure de Trajectoire par Stéréo-Vision pour des Applications de Suivi de Véhicules à Basse Vitesse. ORASIS 2007 - Journées du GDR-PRC Communication Homme-Machine, Jun 2007, Obernai, France. inria-00590240

HAL Id: inria-00590240

<https://inria.hal.science/inria-00590240v1>

Submitted on 9 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mesure de Trajectoire par Stéréo-Vision pour des Applications de Suivi de Véhicules à Basse Vitesse

Julien Morat^{†,*}, Frédéric Devernay^{*}, Sébastien Cornou[†]

INRIA Rhône-Alpes ^{*}
Renault [†]

Résumé

La recherche en matière de régulateurs de vitesse avec gestion des inter-distances est un sujet prometteur dans le domaine des transports intelligents. Un des principaux défis restant à relever dans ce domaine est la perception de l'environnement du véhicule, tout particulièrement à basse vitesse. Dans cet article, nous présentons une nouvelle approche basée sur un système de stéréovision permettant de suivre en 3-D la position et la vitesse du véhicule ciblé. Cette méthode est une extension de l'algorithme proposé par Lucas & Kanade [8, 1]. Nous avons intégré deux contraintes permettant d'exploiter au mieux la stéréovision : (1) la contrainte épipolaire, qui assure que les points suivis restent sur les lignes épipolaires, et (2) la contrainte de grandissement qui lie la profondeur du point suivi à sa taille apparente dans l'image. Nous présentons des résultats expérimentaux sur des données réelles et simulées prouvant l'apport de la méthode en terme de précision et de robustesse.

1 Introduction

La gestion automatique de vitesse des véhicules de séries est un des sujets de recherche les plus populaires auprès de l'industrie automobile [7]. Le régulateur vitesse (Cruise Control en anglais), capable de maintenir un véhicule à vitesse constante, constitue un premier pas dans cette direction. La prochaine étape est un régulateur intelligent capable de garder une distance de sécurité avec le véhicule précédent (Adaptive Cruise Control en anglais). D'ores et déjà, des ACC permettant de réguler des inter-distances à haute vitesse (sur autoroute) sont disponibles sur le marché. Des systèmes de régulation à basse vitesse (Low Speed Following) sont maintenant en cours d'introduction sur les véhicules haut de gamme.

Les systèmes actuels utilisent des LIDAR [10] ou des RADAR pour mesurer la distance et la vitesse relative

au véhicule ciblé. Ces capteurs actifs fournissent une mesure directe et fiable, mais souffrent d'un champ de vue limité ou d'un coût trop élevé.

Les systèmes basés sur la stéréovision peuvent fournir des images en haute résolution des champs de vue larges. De plus ces systèmes deviennent abordables et peuvent être utilisés pour de multiples applications (détection de lignes, détection d'obstacle pour pré crash, ...), créant l'attrait des constructeurs.

Néanmoins, suivre un objet en mouvement au cours d'une longue période de temps est une tâche difficile. La plupart des méthodes de vision, commencent par construire une image de profondeur de la scène, appelée image de disparité, puis en extraient les obstacles et finalement les identifient au cours du temps. Le problème de cette approche est son imprécision. En effet, le processus d'extraction d'obstacle est un problème très difficile qui donne une position relative. Pour obtenir la vitesse des cibles, il faut (1) identifier les obstacles au cours du temps et (2) déduire la vitesse de la succession des positions. Identifier un même obstacle au cours du temps repose souvent sur des hypothèses fortes, telles que des propriétés de mouvement, ce qui peut introduire une première source d'erreur. D'autre part, la vitesse est obtenue en dérivant la position accentuant alors l'imprécision de cette mesure.

Pour éviter cet enchaînement d'imprécisions, nous proposons de calculer une mesure de vitesse directement à partir des informations disponibles dans les images. En se basant sur l'étude comparative des performances des techniques de suivi en vision de Barron et al. [2], mettant en avant les précisions de mesures sur des séquences réalistes, nous reprenons la méthode de calcul du flux optique (basée sur le gradient) de Lukas et Kanade [8, 1]. Étant donné que cette méthode calcul un mouvement 2-D, nous la transposons au cas d'un système stéréo, prenant ainsi en compte l'information de distance. Nous obtenons ainsi un vecteur de mouvement 3-D à chaque pas de temps.

La suite de cet article s'organise en deux parties. Premièrement, la section 2 présente les différentes adaptations et améliorations de la méthode de Lucas

*38334 St Ismier Cedex, France
{julien.morat, frederic.devernay}@inrialpes.fr

[†]Environment Detection and Advanced Driving Assistance Systems, Electronic Department - Renault Research, France
{sebastien.cornou, javier.ibanez-guzman}@renault.com

Kanade au cas de la stéréovision. La formulation du problème est changée pour considérer un espace à 3 dimensions et tenir compte des changements de tailles en fonction de l'éloignement. Des résultats expérimentaux sont présentés pour comparer plusieurs techniques de suivi. Deuxièmement, la section 3 détaille une seconde adaptation de la méthode au cas particulier de suivi de véhicules. Des résultats expérimentaux sur des séquences réelles sont également présentés.

2 Extension de l'algorithme Lucas-Kanade au cas de la Stereo-vision pour le suivi d'objets 3-D

Rappelons que nous sommes intéressés par la mesure de position et de vitesse 3-D des véhicules. Les techniques de suivi 2-D d'éléments dans des images généralement dédiées au suivi d'objets observés par une caméra seule sont conçus pour prendre en compte la cohérence temporelle, mais ne permettent pas d'en déduire une mesure de distance fiable. Au contraire, les techniques de stéréovision fournissent une mesure de position 3-D, mais ne prennent pas immédiatement en compte les connaissances sur la cohérence temporelle (le suivi est souvent effectué dans une seconde phase). Par conséquent, notre approche combine de façon appropriée les deux techniques de manière à utiliser dès le début du processus de mesure, et de façon cohérente, toute l'information spatio-temporelle disponible.

2.1 L'algorithme de Lucas & Kanade

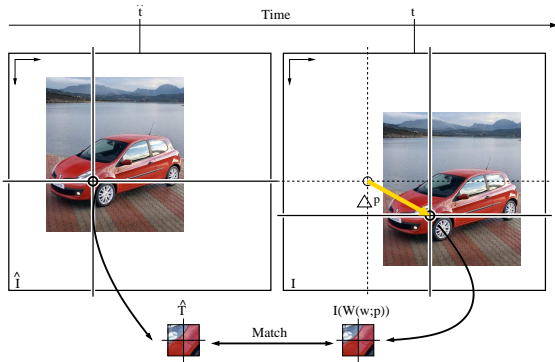


FIG. 1 – L'algorithme de Lucas & Kanade recherche les paramètres \mathbf{p} qui satisfassent la similarité entre le point estimé $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ et le point donné \hat{T} dans \hat{I} .

L'algorithme de Lucas & Kanade [8] est une méthode très utilisée connue pour calculer le flux optique ou suivre des points 2-D dans une séquence vidéo. Généralement, les points suivis sont des points pertinents, ou points d'intérêt ([9]). Le suivi donne une

mesure de déplacement entre une image \hat{I} au temps¹ \hat{t} et une image I au pas de temps suivant t . Le point suivi ce caractérise par une texture $\hat{T}(\mathbf{x})$, où \mathbf{x} est un point de la texture (une interpolation bilinéaire est utilisée pour re-échantillonner l'image). La texture est généralement une fenêtre carrée extraite de \hat{I} , centrée autour de la position sous-pixelique (cf. Fig 1). Etant donné les paramètres \mathbf{p} , la fonction de déformation $\mathbf{W}(\mathbf{x}; \mathbf{p})$ donne la correspondance entre chaque point \mathbf{x} de la texture T et ceux de l'image I . La méthode optimise les paramètres \mathbf{p} (c'est à dire la position du point) dans le but de minimiser la fonction de coût formée par la somme des différences au carré de la texture avec l'image :

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - \hat{T}(\mathbf{x}) \right]^2. \quad (1)$$

Une publication de référence sur le sujet est l'étude de Baker & Matthews [1], d'où proviennent les notations utilisées dans ce papier.

2.2 Suivi de mouvement avec deux points de vue

Dans notre cas, à chaque instant t deux images I_l (gauche) and I_r (droite) sont acquises, car il y a deux caméras. Ainsi, l'élément 3D à suivre est représenté par une paire de points m_l et m_r . Bien que les deux points suivis correspondent au même point 3-D, leurs apparences peuvent être différentes dans les images gauche et droite à cause de l'effet de parallaxe, surtout si l'objet observé est proche. Par conséquent, nous choisissons de caractériser l'élément suivi par deux textures différentes \hat{T}_l et \hat{T}_r de tailles \mathbf{x}_l et \mathbf{x}_r pour les vues gauche et droite. La fonction de coût devient :

$$\sum_{\mathbf{n} \in \{l, r\}} \sum_{\mathbf{x}_n} \left[I_n(\mathbf{W}_n(\mathbf{x}_n; \mathbf{p})) - \hat{T}_n(\mathbf{x}_n) \right]^2, \quad (2)$$

où $n \in \{l, r\}$ indique la vue gauche ou droite, et \mathbf{p} est constitué des coordonnées de la paire de points $m_l = (x_l, y_l)$ and $m_r = (x_r, y_r)$. L'algorithme de Lucas & Kanade est une méthode itérative optimisant $\Delta \mathbf{p}$ dans le but de minimiser :

$$\sum_{\mathbf{n} \in \{l, r\}} \sum_{\mathbf{x}_n} \left[I_n(\mathbf{W}_n(\mathbf{x}_n; \mathbf{p} + \Delta \mathbf{p})) - \hat{T}_n(\mathbf{x}_n) \right]^2. \quad (3)$$

Cette approche suppose qu'une estimation initiale \mathbf{p} doit être connue. Habituellement, \mathbf{p} est initialisé avec les précédentes valeurs observées, excepté pour la première observation où les points sont détectés par une méthode d'extraction de points d'intérêt (cf. [9]). En utilisant l'approximation au premier ordre d'un développement de Taylor de l'équation (2),

¹l'accent circonflexe dénote l'instant précédent

l'équation (3) donne :

$$\sum_{n \in \{l,r\}} \sum_{\mathbf{x}_n} \left[I_n(\mathbf{W}_n(\mathbf{x}_n; \mathbf{p})) + \nabla I_n \frac{\partial \mathbf{W}_n}{\partial \mathbf{p}} \Delta \mathbf{p} - \hat{T}_n(\mathbf{x}_n) \right]^2, \quad (4)$$

où ∇I_n est le gradient de l'image I_n . La dérivée partielle de l'équation (4) par rapport à $\Delta \mathbf{p}$:

$$2 \sum_{n \in \{l,r\}} \sum_{\mathbf{x}_n} \left[\nabla I_n \frac{\partial \mathbf{W}_n}{\partial \mathbf{p}} \right]^T \left[I_n(\mathbf{W}_n(\mathbf{x}_n; \mathbf{p})) + \nabla I_n \frac{\partial \mathbf{W}_n}{\partial \mathbf{p}} \Delta \mathbf{p} - \hat{T}_n(\mathbf{x}_n) \right].$$

Au minimum, la dérivée partielle doit être égale à zéro, ce qui conduit à l'expression suivante pour la mise à jour des paramètres :

$$\Delta \mathbf{p} = H^{-1} b, \quad (5)$$

où H est l'approximation par la méthode Gauss-Newton de la matrice Hessienne :

$$H = \sum_{n \in \{l,r\}} \sum_{\mathbf{x}} \left[\nabla I_n \frac{\partial \mathbf{W}_n}{\partial \mathbf{p}} \right]^T \left[\nabla I_n \frac{\partial \mathbf{W}_n}{\partial \mathbf{p}} \right], \quad (6)$$

et b est :

$$b = \sum_{n \in \{l,r\}} \sum_{\mathbf{x}_n} \left[\nabla I_n \frac{\partial \mathbf{W}_n}{\partial \mathbf{p}} \right]^T \left[\hat{T}_n(\mathbf{x}_n) - I_n(\mathbf{W}_n(\mathbf{x}_n; \mathbf{p})) \right]. \quad (7)$$

Les deux prochaines sections présentent l'extension de l'algorithme de Lucas & Kanade au cas stéréoscopique incorporant la contrainte 3-D et la contrainte de grandissement.

2.3 Suivre de points 3-D

Sans aucune considération géométrique, suivre une paire de points dans une séquence stéréo est équivalent à suivre les deux points de manière indépendante. Le vecteur de paramètres \mathbf{p} est ainsi composé d'une paire de coordonnées 2-D (i.e. 4 paramètres). Dans le reste de l'article, cette approche sera prise comme référence pour l'évaluation des différentes méthodes. Nous l'appelons *Lucas & Kanade non-contraint*.

Cependant, rappelons que les deux points \mathbf{m}_l et \mathbf{m}_r correspondent à la projection d'un unique point 3-D \mathbf{M} . La relation entre les deux vues d'une même scène induit une contrainte forte. Cette contrainte, appelée contrainte épipolaire, réduit les degrés de liberté du problème. L'algorithme ne prenant pas en compte cette contrainte épipolaire peut aboutir à une solution ne la satisfaisant pas, c'est à dire n'ayant pas d'existence dans l'espace 3-D. Pour prendre en compte cette contrainte, le vecteur de paramètres \mathbf{p} peut simplement se composer des coordonnées euclidiennes du point $\mathbf{M} = (X, Y, Z)$. Cela implique la définition de

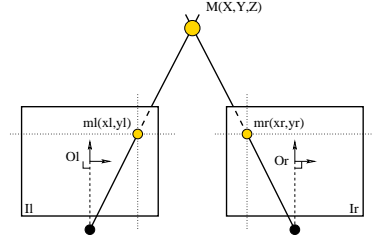


FIG. 2 – Le point 3-D M se projette en 2 points 2-D $m_l(x_l, y_l)$ et $m_r(x_r, y_r)$.

fonctions de projections \mathbf{P}_n transformant le point 3-D en coordonnées 2-D dans les images I_n (avec $n \in \{l, r\}$).

Cependant, nous préférons utiliser une représentation de l'espace 3-D basée sur l'espace image. D'un point de vue géométrique, la contrainte épipolaire s'exprime par le fait que pour un point donné m_l de l'image I_l , une ligne épipolaire associée peut être calculée dans l'autre image (I_r). Le point m_r , correspondant à la projection du même point 3-D que m_l , appartient forcément à cette droite épipolaire. Les coordonnées du point $m_l = (x_l, y_l)$ associé à la disparité d (l'ordonnée de m_r sur la droite épipolaire) constituent les trois dimensions du nouveau vecteur de paramètres \mathbf{p} . Dans la suite de cet article, nous appellerons ce système de coordonnées *espace basé image*.

Dans la configuration standard, la mesure de disparité d entre m_l et m_r est en fait un vecteur 2-D défini par [6] comme suit :

$$\vec{d} = O_l \vec{m}_l - O_r \vec{m}_r. \quad (8)$$

où O_l et O_r sont les points principaux dans les images I_l et I_r . Cependant, nous prenons pour hypothèse que la paire de caméras est dans une configuration spécifique (cf. Fig. 2), dans laquelle la ligne de base est alignée avec l'axe des abscisses et les axes optiques des deux caméras sont parallèles. Dans cette configuration, les droites épipolaires sont parallèles aux lignes de l'image et par conséquent la disparité peut être exprimée par un nombre réel $d = (x_l - O_x l) - (x_r - O_x r)$. Dans le cas où les caméras ne sont pas dans cette configuration fronto-parallèle, il existe une transformation image qui permet de transformer les images de sorte à retrouver cette configuration [5]. Pour calculer la mise à jour $\Delta \mathbf{p}$ des paramètres (Eq. 5), les éléments suivants doivent être définis : les textures $T_l(\mathbf{x})$ et $T_r(\mathbf{x})$, les déformations \mathbf{W}_l et \mathbf{W}_r , et finalement la Jacobienne de chaque déformation $\partial \mathbf{W}_l / \partial \mathbf{p}$ et $\partial \mathbf{W}_r / \partial \mathbf{p}$. Les textures $T_l(\mathbf{x})$ et $T_r(\mathbf{x})$ sont des fenêtres carrées extraites des images au temps \hat{t} , centrées autour des deux projections (dans les deux images) du point 3-D considéré. Rappelons que les déformations \mathbf{W}_n sont des fonctions 2-D qui mettent en correspondance un point de texture avec un point de l'image n (avec $n \in \{l, r\}$). Dans notre cas, chaque déformation \mathbf{W}_n est une translation

2-D dans l'image n définie par :

$$\mathbf{W}_n(\mathbf{x}_n; \mathbf{p}) = \mathbf{x}_n + \mathbf{P}_n(\mathbf{p}), \quad (9)$$

où $\mathbf{p} = (x, y, d)$, et sa projection $\mathbf{P}_n(\mathbf{p})$ s'écrit comme suit :

$$\mathbf{P}_l(\mathbf{p}) = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{P}_r(\mathbf{p}) = \begin{pmatrix} x + d \\ y \end{pmatrix}. \quad (10)$$

Les deux jacobiniennes $\partial \mathbf{W}_l / \partial \mathbf{p}$ et $\partial \mathbf{W}_r / \partial \mathbf{p}$ associées aux fonctions de déformations sont :

$$\frac{\partial \mathbf{W}_l}{\partial \mathbf{p}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \frac{\partial \mathbf{W}_r}{\partial \mathbf{p}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \quad (11)$$

Le terme de plus grande pente devient donc :

$$\begin{aligned} \nabla I_l \frac{\partial \mathbf{W}_l}{\partial \mathbf{p}} &= [I_{lx} \quad I_{ly} \quad 0], \\ \nabla I_r \frac{\partial \mathbf{W}_r}{\partial \mathbf{p}} &= [I_{rx} \quad I_{ry} \quad I_{rx}]. \end{aligned} \quad (12)$$

Etant donnée l' Eq. (12), nous développons l'approximation de la matrice Hessienne H de l'Eq. (6) :

$$H = \sum_{\mathbf{x}} H_l + \sum_{\mathbf{x}} H_r, \quad (13)$$

où H_l et H_r dependent des Eq. (12) et s'écrivent :

$$H_l = \sum_{\mathbf{x}} \begin{bmatrix} I_{lx}^2 & I_{lx}I_{ly} & 0 \\ I_{lx}I_{ly} & I_{ly}^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (14)$$

$$H_r = \sum_{\mathbf{x}} \begin{bmatrix} I_{rx}^2 & I_{rx}I_{ry} & I_{rx}^2 \\ I_{rx}I_{ry} & I_{ry}^2 & I_{rx}I_{ry} \\ I_{rx}^2 & I_{rx}I_{ry} & I_{rx}^2 \end{bmatrix}. \quad (15)$$

2.4 Tenir compte des variations de taille dans l'image

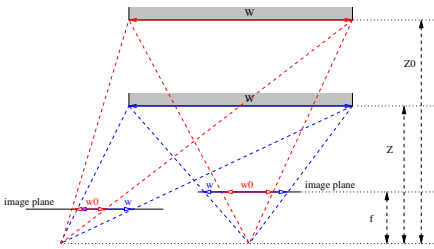


FIG. 3 – La taille w d'un objet dans l'image dépend de sa taille réelle W mais aussi de la focale f et de sa distance Z aux caméras.

Quand les coordonnées du point principal sont nulles, l'Eq (8) se simplifie pour obtenir :

$$d = Bf/Z. \quad (16)$$

B et f étant constants, un variation de d dépend directement d'une variation de Z . De façon similaire, il

existe une relation entre la taille apparente w d'un objet dans l'image et sa distance Z aux caméras (cf. Fig. 3) :

$$w = Wf/Z, \quad (17)$$

où W désigne la taille réelle de l'objet fronto-parallèle. Une grande variation de profondeur implique un changement significatif de la taille apparente du motif dans les images. Dans le cas d'un suivi utilisant une fenêtre de taille fixe, le processus peut être imprécis ou échouer (changement d'aspect trop important). Cette limitation est contradictoire avec l'application de suivi à basse vitesse qui implique des distances relativement faibles entre le véhicule équipé du système et le véhicule le précédant.

Par conséquent, nous proposons une méthode prenant en compte cette variation de taille apparente dans le cas d'objets fronto-parallèles. Eq. (16) et Eq. (17) montrent que les variations de taille w et de disparité d dépendent toutes les deux des variations de profondeurs Z , conduisant à :

$$\frac{d}{\hat{d}} = \frac{Bf/Z}{Bf/\hat{Z}} = \frac{\hat{Z}}{Z} = \frac{Wf/Z}{Wf/\hat{Z}} = \frac{w}{\hat{w}}, \quad (18)$$

où \hat{d} et \hat{w} désignent respectivement la disparité et la taille apparente précédemment mesurées. Ainsi $w = \hat{w} \times d/\hat{d}$, permet de connaître la variation de taille sans avoir à estimer un paramètre supplémentaire. Les fonctions de déformation \mathbf{W}_n deviennent alors (une interpolation bilinéaire est utilisée pour échantillonner l'image) :

$$\mathbf{W}_n(\mathbf{x}_n; \mathbf{p}) = \left(\frac{d}{\hat{d}} \right) \mathbf{x}_n + \mathbf{P}_n(\mathbf{p}), \quad (19)$$

par conséquent les Jacobiniennes deviennent :

$$\frac{\partial \mathbf{W}_l}{\partial \mathbf{p}} = \begin{pmatrix} 1 & 0 & \frac{i}{\hat{d}} \\ 0 & 1 & \frac{j}{\hat{d}} \end{pmatrix}, \quad \frac{\partial \mathbf{W}_r}{\partial \mathbf{p}} = \begin{pmatrix} 1 & 0 & \frac{i}{\hat{d}} + 1 \\ 0 & 1 & \frac{j}{\hat{d}} \end{pmatrix}, \quad (20)$$

et les termes de plus grande pente deviennent :

$$\nabla I_n \frac{\partial \mathbf{W}_n}{\partial \mathbf{p}} = [I_{nx} \quad I_{ny} \quad S_n], \quad (21)$$

avec $n \in \{l, r\}$ et

$$S_r = \frac{I_{rx}i + I_{ry}j}{\hat{d}} + I_{rx}, \quad S_l = \frac{I_{lx}i + I_{ly}j}{\hat{d}}. \quad (22)$$

Reprenant l'Eq. (21), les matrices H_n de l'Eq. (13) deviennent :

$$H_n = \sum_{\mathbf{x}} \begin{bmatrix} I_{nx}^2 & I_{nx}I_{ny} & I_{nx}S_n \\ I_{nx}I_{ny} & I_{ny}^2 & I_{ny}S_n \\ I_{nx}S_n & I_{ny}S_n & S_n^2 \end{bmatrix}. \quad (23)$$

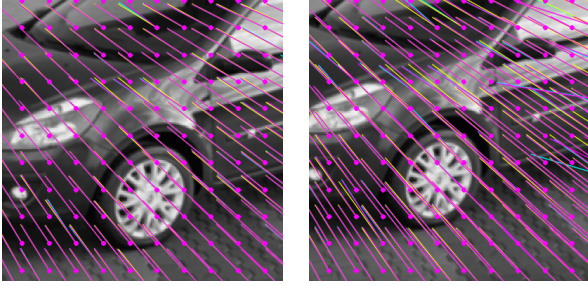


FIG. 4 – Agrandissement d’une portion de l’image montrant des points suivis avec la méthode non-contrainte (magenta), en utilisant des coordonnées 3-D (cyan) et en incorporant le grandissement (jaune). L’image du haut correspond à la vitesse de référence (cf. Fig. 7) et celle du bas à la vitesse double.

2.5 Résultats expérimentaux

Nous avons comparé l’algorithme de Lucas-Kanade non-contraint (décrit dans la Sec. 2.3) avec deux versions de notre algorithme : la première (1) utilisant des coordonnées 3-D (Sec. 2.3) basées image, et une seconde méthode (2) tenant compte des variations d’échelle dans l’image (Sec. 2.4).

Afin de quantifier les performances des différents algorithmes, nous avons généré un ensemble de séquences stéréo d’un plan texturé évoluant longitudinalement (cf. Fig. 6). Pour établir le comportement des algorithmes par rapport à différentes vitesses d’éloignement, la séquence est rejouée suivant 5 vitesses différentes. Pour établir le comportement des algorithmes par rapport à différents niveaux de bruits sur les intensités pixel, une séquence à vitesse de référence a été reprise et entachée avec plusieurs intensités de bruit blanc gaussien.

Les points suivis servant à évaluer les méthodes sont initialement répartis sur une grille régulière de 512×512 pixels. La Fig. 4 présente un agrandissement partiel du résultats des différentes méthodes sur une séquence à la vitesse de référence (en haut) et à la vitesse double (en bas).

Les résultats présentés ci-après ont été obtenus en utilisant des textures \hat{T} de 21×21 pixels sur des images de 1024×768 pixels.

Généralement, l’erreur *RMS* (Root Mean Square) est utilisée pour quantifier la précision de ce type d’algorithme. Mais la détection et l’élimination des outliers est néanmoins recommandée avant de calculer l’erreur *RMS*. La Fig. 5 montre clairement la présence d’outliers. Nous avons donc décidé de modéliser la précision par le mélange de deux gaussiennes : la première dédiée à la précision du suivi, et la seconde à la répartition des outliers. Les pourcentages de données aberrantes (outliers) et de points pour lesquels la méthode a échoué sont cumulés pour donner la proportion d’échecs. L’al-

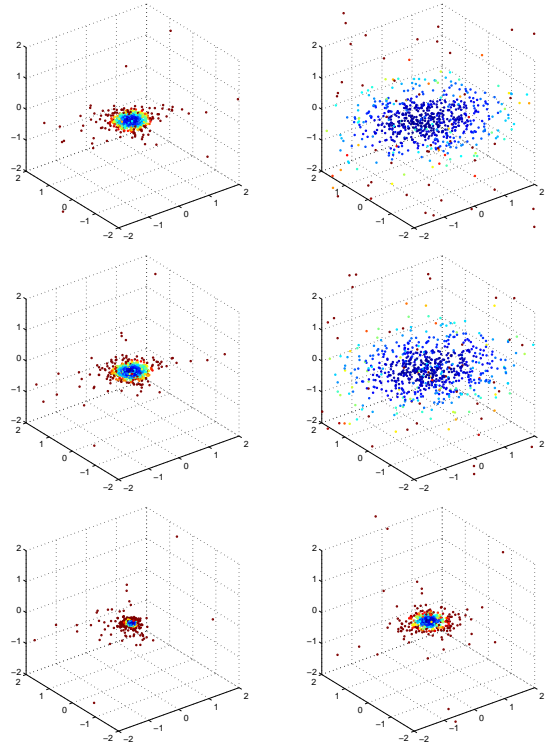


FIG. 5 – Répartition de l’erreur en (x, y, d) (pixel) pour l’algorithme *Lucas & Kanade non-contraint* (en haut), utilisant les coordonnées 3-D (centre) et incorporant les variations de taille (en bas). La colonne de gauche provient des mesures effectuées sur la séquence à vitesse $1\times$ et celle de droite à vitesse $5\times$ dans la Fig. 7.

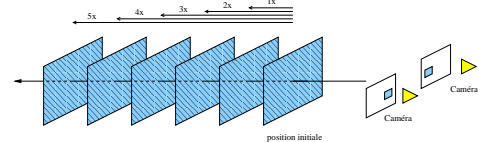


FIG. 6 – Génération d’un ensemble de séquences stéréo d’un plan texturé évoluant longitudinalement.

gorithme *Espérance Maximisation* [4] est utilisé pour retrouver les deux gaussiennes du modèle d’erreur.

L’erreur *RMS* présentée dans les Fig. 7 et Fig. 8 sont déduites des matrices de covariances des deux gaussiennes modélisant l’erreur.

Notre première approche utilisant des coordonnées 3-D basées image n’améliore pas la précision du suivi (*RMS inliers*) de manière significative (cf. Fig 7 et 8). Le pourcentage d’échecs est néanmoins réduit, ce qui s’explique par le fait que la contrainte 3-D est intrinsèquement prise en compte dans la formulation du problème. L’espace des solutions ainsi réduit élimine toutes les solutions mal conditionnées.

Comme attendu, la seconde approche améliore le pourcentage d’échecs, aussi bien que la précision de suivi. Le gain est encore significatif dans le cas d’objets proches et/ou s’éloignant rapidement. La méthode

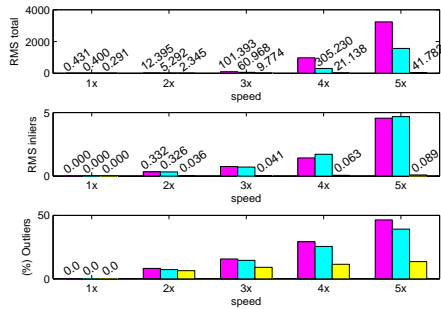


FIG. 7 – Comparaison des trois algorithmes en fonction des vitesses d'éloignement : (1) non-contraint (magenta), (2) utilisant les coordonnées 3-D (cyan) et (3) incorporant le grandissement (jaune).

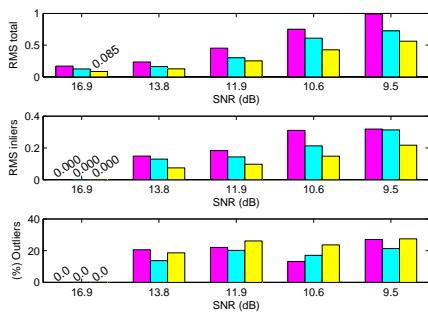


FIG. 8 – Comparaison des trois algorithmes en fonction du rapport signal bruit (dB) : (1) non-contraint (magenta), (2) utilisant les coordonnées 3-D (cyan) et (3) incorporant le grandissement (jaune).

s'applique à n'importe quel système de stéréovision, à condition que les images soient rectifiées avec des focales équivalentes et que la disparité soit nulle à l'infini.

3 Application au suivi de véhicules

La principale limitation de l'algorithme de Lucas-Kanade est que, suivant l'échelle de la texture \hat{T} , la méthode peut être capable de ne suivre que des déplacements d'environ un pixel. Pour suivre de plus amples mouvements, l'implantation de Bouguet [3], comme la nôtre, utilise une approche pyramidale (cf. Fig. 9), qui estime un mouvement à une résolution grossière (typiquement 1/8 de la taille d'image originale) avant de raffiner à des échelles plus fines (1/4, 1/2, et 1). En passant d'une échelle à l'autre, la taille de l'image change, et le vecteur de paramètres est mis à jour en conséquence. Par contre, la taille de la texture reste inchangée, ce qui a pour effet de considérer des régions de différentes proportions dans l'image. Un des effets indésirables est que la fenêtre de texture \hat{T} caractérisant un point peut recouvrir plusieurs objets. Quand ces objets ont des trajectoires différentes, le point suivi glisse entre les deux objets effectivement

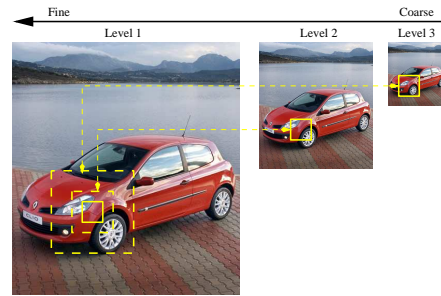


FIG. 9 – L'approche pyramidale de Bouguet [3] utilise toujours la même taille de fenêtre quelque soit le niveau de détail. La conséquence est que la fenêtre recouvre une plus grande région de l'image dans les niveaux les plus grossiers.

suivis.

3.1 Suivi de plans 3-D

Une solution pour éviter cet effet est de faire correspondre les textures 2-D \hat{T} à une surface réelle. Au lieu de suivre des points 3-D, nous proposons de suivre des plans 3-D. Pour ne pas ajouter de paramètres supplémentaires, nous ne considérons que des plans faisant face aux caméras. Ainsi, il n'est pas nécessaire d'ajouter des paramètres supplémentaires à estimer dans le vecteur \mathbf{p} pour quantifier l'orientation du plan. Cette limitation n'est pas gênante pour l'application de suivi à basse vitesse, car les véhicules suivis font face au véhicule équipé. Dans le cas de petites textures, il est toutefois difficile de suivre l'objet. En effet, une texture de 3×3 pixels peut être trop peu discriminante. Pour éviter de baser le suivi sur des informations non pertinentes, les niveaux de détail où la texture est trop petite ne sont pas considérés. A l'opposé, les textures de trop grande taille peuvent ralentir le calcul pour une précision pouvant être surdimensionnée. Dans le même esprit, les niveaux de détails où la texture considérée est trop grande seront ignorés.

3.2 Expérimentations sur séquences réelles

Nous présentons quelques résultats que nous avons obtenus avec un système monté sur véhicule d'essai (Fig. 10). Les caméras sont munies d'un capteur CCD de 640×480 pixels couplé avec un objectif de 6mm. Elles sont disposées derrière le pare-brise, espacées de 40cm de part et d'autre du rétroviseur. Le choix d'une position dominante correspond à une implantation réaliste sur véhicule de série. Le système a été calibré pour fournir des images rectifiées répondant aux exigences des algorithmes (lignes épipolaires parallèles et disparité nulle à l'infini).

La Fig. 11 présente le résultat des différents algorithmes sur une séquence réelle. Les coordonnées 3-D et dimensions initiales du véhicule cible ont été paramétrées manuellement sur la première image de la

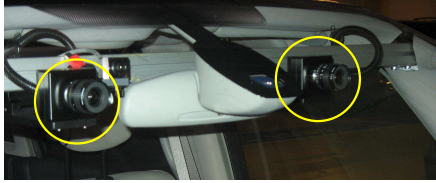


FIG. 10 – Paire de caméras montées derrière le pare-brise du véhicule d’essai.

séquence. La colonne de gauche présente les résultats obtenus par les méthodes décrites dans la Sec. 2, c’est à dire le suivi de point 3-D sans contrainte (magenta), incorporant la 3-D et le grandissement (jaune). La colonne de droite présente le gain obtenu avec l’adaptation de la méthode au cas spécifique de suivi de véhicules.

L’algorithme de suivi de point 3-D commence à perdre en précision à l’image N°150 pour perdre complètement le véhicule à l’image N°350. L’algorithme adapté au suivi de véhicule suit le véhicule durant toute la séquence (440 images), ce qui constitue un gain très significatif, non seulement en terme de robustesse mais aussi de précision.

4 Conclusions

La régulation de vitesse auto-adaptative nécessite une perception de l’environnement 3-D précise et fiable. En réponse à cette problématique, nous proposons une approche issue de celle de Lucas & Kanade [8] adaptée au cas de stéréovision. Sous la condition d’une rectification des images adaptée, cette méthode tire parti à la fois de la stéréovision qui apporte une information de profondeur et du flux optique qui apporte une information de vitesse. Les résultats obtenus sur des séquences virtuelles ou réelles prouvent l’apport de cette méthode en termes de robustesse et fiabilité tout en conservant des temps de calcul identiques au *Lucas & Kanade non-contraint*.

Références

- [1] S. Baker and I. Matthews. Lucas-Kanade 20 years on : A unifying framework. *IJCV*, 56(3) :221–255, February 2004.
- [2] J. L. Barron, D. J. Beauchemin Fleet, S.S., and T. A. Burkitt. Performance of optical flow techniques. In *Proceedings of Conference on Computer Vision and Pattern Recognition CVPR92*, pages 236–242, Urbana-Champaign, USA, 1992.
- [3] Jean-Yves Bouguet. Pyramidal implementation of the Lucas-Kanade feature tracker. Technical report, Intel Corp., Microprocessor Research Labs, 2000.
- [4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39 :1–38, 1977.

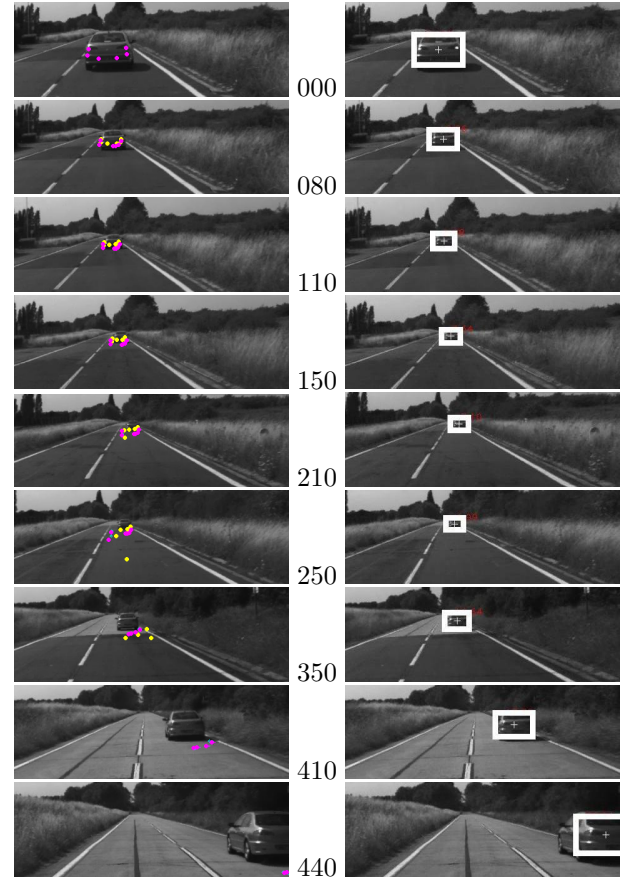


FIG. 11 – Séquence acquise à bord du véhicule d’essai alors qu’il en dépasse un autre. La colonne centrale indique le temps, la colonne de gauche montre le résultat des algorithmes présentés en Sec. 2 et la colonne de droite montre ceux obtenus avec la méthode adaptée au suivi de véhicule décrite dans cette section. Dans les deux cas, les cibles initiales ont été choisies manuellement sur la première image de la séquence.

- [5] Frédéric Devernay. *Vision stéréoscopique et propriétés différentielles des surfaces*. Thèse de doctorat, Institut National Polytechnique, 1997.
- [6] D. A. Forsyth and J. Ponce. *Computer Vision : a modern approach*. Prentice-Hall, 2002.
- [7] W.D. Jones. Keeping cars from crashing. *IEEE Spectrum*, pages 40–45, 2001.
- [8] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [9] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, pages 593–600, Seattle, WA, June 1994. IEEE Comp.Soc.
- [10] Y. Teguri. Laser sensor for low-speed cruise control. *Convergence Transportation Electronics Association*, 2004.