



HAL
open science

Stratégie de perception active pour l'interprétation de scènes : Application à une scène routière

Coralie Bernay-Angeletti, Claude Aynaud, Romuald Aufrère, Roland Chapuis

► To cite this version:

Coralie Bernay-Angeletti, Claude Aynaud, Romuald Aufrère, Roland Chapuis. Stratégie de perception active pour l'interprétation de scènes : Application à une scène routière. Orasis, Congrès des jeunes chercheurs en vision par ordinateur, Jun 2013, Cluny, France. hal-00829440

HAL Id: hal-00829440

<https://hal.science/hal-00829440v1>

Submitted on 5 Jun 2013

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.

Stratégie de perception active pour l'interprétation de scènes : Application à une scène routière

C. Bernay-Angeletti¹

C. Aynaud¹

R. Aufrère^{1,2}

R. Chapuis¹

¹Institut Pascal
UMR 6602 UBP / CNRS
Campus des Cézeaux
63177 AUBIERE Cedex

² LIMOS
UMR 6158 UBP / CNRS
Campus des Cézeaux
63177 AUBIERE Cedex

Coralie.Bernay_Angeletti@etudiant.univ-bpclermont.fr,
(claude.aynaud, romual.aufrere, roland.chapuis,)@univ-bpclermont.fr

Résumé

Cet article décrit une méthode générique pour reconnaître des objets donnés en cherchant à utiliser au mieux toutes les connaissances a priori disponibles de la scène. Chaque objet est composé d'un ensemble de parties. A chacune de ces parties sont associés une primitive et un détecteur pour la trouver. Les différentes étapes de l'approche seront alors : la focalisation des parties (c'est à dire qu'on détermine la zone de recherche des primitives associées), la sélection de la "meilleure partie" (celle qui a priori doit apporter le plus pour la reconnaissance de l'objet), la détection des primitives dans la zone associée à cette partie et la sélection de la meilleure primitive (celle qui correspond le plus à nos attentes) et enfin la mise à jour de l'objet compte tenu de la réussite (ou de l'échec) de la détection précédente. Ce papier décrit cette approche avec une application dédiée à une scène routière comprenant une route et un panneau de limitation de vitesse.

Mots Clef

reconnaissance d'objets, recherche focalisante, reconnaissance guidée par le modèle

Abstract

This article describes a general method for objects recognition by trying to make the best use of our knowledge about the world and the objects. Every object consists of a set of parts. At each of these parts are associated a primitive and a detector to find it. The various stages of this approach are : the focus of the parts (we determine the area of research of the associated primitive), the selection of the " best part " (the one which should give the most for the object recognition), the detection of the primitive in the search area of this part, the selection of the best feature (the one which is the most to our expectations) and finally the update of the object considering the success (or the failure) of the previous detection. In this article, this approach

will be applied to a road scene wich is composed of a road and a panel of speed limits.

Keywords

objects recognition, search tree, model driven recognition

1 Introduction

La reconnaissance d'objets est une étape primordiale pour l'interprétation de scènes mais reste néanmoins une opération très difficile du fait de la grande variabilité possible des objets d'une même classe. On peut classer les méthodes en trois grandes catégories : les approches top-down [2], les approches bottom-up [9] et celles combinant les deux [8, 10, 11, 13].

La première catégorie cherche à utiliser les connaissances sur l'apparence ou la forme de l'objet. Ces connaissances sont le plus souvent obtenues par un apprentissage d'un modèle de classe. On compare ensuite les caractéristiques de l'image au modèle de l'objet pour en trouver les instances, on exploite dans ce cas des techniques de classifications [3, 5] telles que SVM ou AdaBoost. Ces approches permettent d'aller rapidement vers les hypothèses prometteuses mais l'apprentissage peut nécessiter de très grandes bases de données et ne pas avoir une flexibilité suffisante pour représenter toutes les variations intraclasse.

Les approches de types bottom-up [7, 6] au contraire extraient des primitives de bas ou de moyen niveaux comme des coins ou des segments, les rassemblent selon des règles de construction et génèrent ainsi leurs hypothèses qui sont ensuite évaluées à l'aide d'une fonction de coût. Elles assurent une consistance importante avec les caractéristiques de l'image ce qui est essentiel pour la segmentation par exemple. Néanmoins, le groupement demandera, pour ces approches, beaucoup d'effort. C'est pourquoi de plus en plus, un mixte des deux méthodes permettant de cumuler les avantages des deux est utilisé.

Notre méthode s'inscrit parmi ces dernières. Un modèle de

notre objet nous permet de *focaliser* notre attention sur les zones intéressantes de l'image. Dans ces zones seront recherchées des primitives (c'est à dire des caractéristiques de notre objet). Chaque fois qu'une nouvelle primitive sera trouvée, l'état de reconnaissance de l'objet sera mis à jour afin de restreindre encore davantage les zones de recherche restantes. Ce faisant, nous assurons une cohérence au niveau du groupement des primitives, et nous limitons les risques de mauvaises détections en restreignant l'espace de recherche.

Notre approche permet de prendre en compte de manière explicite les constituants de l'objet et de ne pas le considérer comme une entité fermée apprise sous différents points de vues. En particulier, la variabilité des positions des parties de l'objet recherché est gérée explicitement. En outre, la connaissance explicite de tous les objets constituant une scène devrait nous permettre de gérer les interactions entre les objets à la fois d'un point de vue du positionnement possible et des occultations qui peuvent survenir (même si dans un premier temps, elles ne seront pas gérées). Tout ceci devrait aider à l'interprétation complète d'une scène. Par exemple, dans le cas d'une scène routière, une fois la route détectée, les positions possibles pour un panneau sont limitées.

Les différentes étapes de l'approche présentée seront décrites dans une première partie. Une deuxième partie sera consacrée aux résultats obtenus. Enfin, la dernière partie présentera les conclusions et les perspectives de ce travail.

2 Description de l'approche

Dans cette partie, nous décrivons l'approche générique de la reconnaissance d'objet. L'idée est d'utiliser une méthode top-down guidée par le modèle qui lancera des phases de détection bottom-up. On suppose disposer d'un modèle de l'objet à reconnaître sous forme d'un vecteur d'état et de sa covariance. A chaque partie de l'objet à reconnaître correspond une ou plusieurs composantes du vecteur d'état. L'objectif sera donc de trouver à chaque instant quelle est la partie qui améliorera le mieux l'estimation de l'état de l'objet.

Par souci de clarté, l'exemple du panneau de limitation de vitesse sera donné par la suite mais l'approche s'applique aisément à d'autres types d'objets.

2.1 Principe général

La méthode proposée repose sur deux piliers fondamentaux. Le premier est la focalisation : contrairement aux autres approches, tout n'est pas cherché dans toute l'image. Au contraire dès qu'une information est connue (provenant du modèle ou d'une détection antérieure), la recherche de données supplémentaires se fera uniquement sur une zone restreinte de l'image assurant ainsi à chaque instant la cohérence des détections entre elles. Cette étape est mieux précisée dans 2.4. Le second est l'estimation de la confiance : non seulement l'estimation des paramètres de l'objet est recherchée mais en plus il s'agit de savoir quelle

est la fiabilité de cette estimation. L'objectif est d'être précis tout en restant intègre ou du moins en caractérisant l'intégrité. La principale préoccupation de cette méthode est donc en résumé d'assurer un maximum de cohérence, que ce soit entre les détections ou entre l'estimation et la réalité.

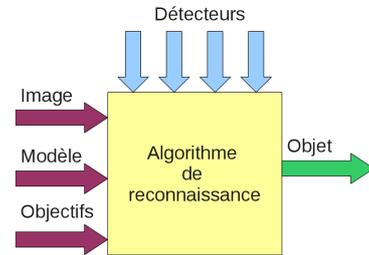


FIGURE 1 – entrées et sorties de l'algorithme

Elle prendra donc en entrée un modèle de l'objet, une image et des objectifs à atteindre : un objectif de précision et un objectif de confiance. Elle utilisera pour parvenir à sa reconnaissance des détecteurs dont les caractéristiques (fiabilité, temps) seront connues. En sortie, elle renverra l'estimation des paramètres de l'objet avec la confiance.

2.2 Objet, Partie et Primitive

Objet. Un objet sera caractérisé par trois paramètres : un vecteur d'état E , sa covariance $covE$ et une probabilité pE . Le vecteur d'état E et sa covariance $covE$ permettent de connaître la valeur moyenne de la position de l'objet avec son incertitude. La probabilité pE représente la confiance en l'estimation donnée par E et $covE$ de l'objet.

Le panneau de limitation de vitesse pourra, par exemple, être défini par : $(X_p, Y_p, Z_p, R_{ext}, r)^t$ (s'il est vu de face) avec :

- $(X_p, Y_p, Z_p)^t$: la position du panneau dans le monde
- R_{ext} : le rayon du cercle externe du panneau
- $r = \frac{R_{int}}{R_{ext}}$ avec R_{int} : le rayon du cercle intérieur.

Reconnaître un objet va alors consister à estimer son vecteur d'état E avec une certaine précision (donnée par $covE$) tout en conservant une confiance pE élevée dans cette estimation.

L'objectif de détection est alors double : il s'agit à la fois de parvenir à une certaine précision et d'avoir une grande confiance dans cette estimation.

Partie. Pour réaliser cette reconnaissance, l'objet sera découpé en Parties. Compte tenu des liens entre l'objet et ses parties, chacune d'elles pourra utiliser sa connaissance de l'état de l'objet afin de déduire dans quelle zone elle devra effectuer sa recherche. A l'inverse, lorsque sa localisation sera connue, elle pourra mettre à jour l'objet afin de lui faire prendre en compte cette nouvelle information. Les parties d'un objet sont donc propres à cet objet et ne peuvent en aucun cas être partagées par un autre objet. L'objet *panneau* disposera des parties *blobRouge*, *grandCercle* et *petitCercle*. On retrouve un découpage similaire dans [4] où le rouge est extrait de l'image est

ensuite un cercle est cherché. Cependant, dans notre approche, les liens entre la position du rouge et celles des cercles sont explicites.

Primitive. Reconnaître une Partie d'un objet va alors consister à trouver dans l'image une Primitive associée à cette partie. On va donc associer à chacune des parties, une Primitive et un détecteur. Ce dernier permettra de trouver dans une zone donnée toutes les primitives du même type que celle nécessaire à la reconnaissance de la partie.

Si les parties sont propres à un objet, les primitives au contraire peuvent être nécessaires à différentes parties d'un même objet et même à plusieurs objets différents.

Reconnaître l'objet va donc consister à reconnaître les différentes parties qui le composent, c'est-à-dire à trouver pour chacune d'elles une primitive adéquate.

2.3 Algorithme de reconnaissance

L'objet à détecter est modélisé à l'état initial par $E_0, covE_0, pE_0$. Supposons cet objet constitué de M parties. L'objet sera à l'étape $k \in [1..M]$ de sa reconnaissance lorsque k parties auront lancé leur détecteur (que cette détection ait réussi ou non). Sur la figure 2, le déroulement des différentes étapes de la méthode est montré.

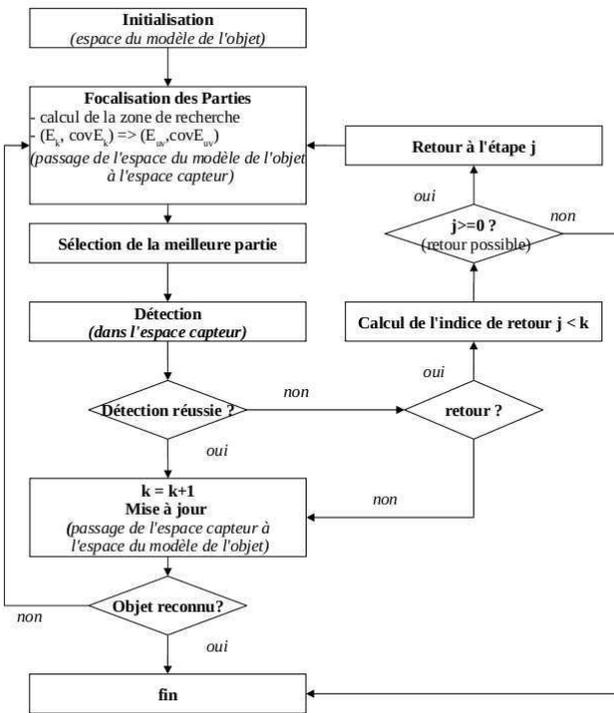


FIGURE 2 – le schéma de fonctionnement global de notre méthode

Nous allons traiter dans les sections suivantes les différentes étapes de cet algorithme de manière plus détaillée.

2.4 Focalisation

Principe. La focalisation consiste à extraire de l'état de l'objet les données intéressantes pour chacune des parties et doit permettre le passage de l'espace monde à l'espace capteur (une image issue d'une caméra dans notre cas). Ce principe de focalisation a déjà été utilisé dans [1, 12]

Par exemple, pour focaliser la partie *grandCercle* du panneau, les informations sur (X_p, Y_p, Z_p) ainsi que sur R_{ext} seront nécessaires, par contre r ne nous intéressera pas pour cette partie (utile pour *petitCercle* par contre). Il s'agira alors de transformer les informations sur la position et la covariance associée de ces données, en une zone de recherche dans l'image et en un ensemble de valeurs possibles pour la primitive à trouver (soit une valeur moyenne et une covariance associée). Pour le *grandCercle*, il nous faut une fonction f telle que $(u_0, v_0, r_{ext}) = f(X_p, Y_p, Z_p, R_{ext})$ où

- (X_p, Y_p, Z_p) : la position du panneau en 3D
- R_{ext} : le rayon du cercle extérieur en 3D
- (u_0, v_0) : le centre du cercle attendu dans l'image
- r_{ext} : le rayon du cercle dans l'image

Avantages. Ce principe de focalisation présente plusieurs intérêts :

- la cohérence des détections : les parties n'étant recherchées que dans des zones plausibles compte tenu de l'état de l'objet, elles ne pourront pas être trouvées dans un endroit en contre indication avec les connaissances sur l'objet
- la qualité des détections : plus la zone de recherche sera petite, plus il sera probable que la détection soit valable (optimisation du rapport signal sur bruit).
- une diminution des temps de calculs : plus la taille de la zone sera faible, moins les temps de traitements pour extraire les primitives seront élevés

La figure 3 permet d'illustrer cette *focalisation*. Sur la première image, la *zoneRouge* a été trouvée, en jaune apparaissent les zones de recherche possibles pour le *petitCercle* et le *grandCercle*. Elles débordent de la *zoneRouge* du fait de la grande imprécision du capteur de rouge. Sur l'image suivante, le *grandCercle* a été trouvé et la zone de recherche pour le *petitCercle* est diminué en conséquence.

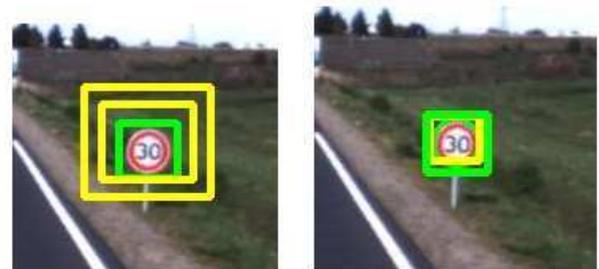


FIGURE 3 – focalisation des parties : exemple pour la détection de panneaux

2.5 Choix de la meilleure partie à rechercher

Il s'agit de définir ici un critère qui permettra de comparer les différentes parties afin de sélectionner celle qui semble la meilleure soit parce qu'elle apportera de la précision sur la localisation de l'objet, soit parce qu'elle confortera l'hypothèse de localisation. Ce critère doit prendre en compte plusieurs facteurs.

Apport en précision. Une partie sera sélectionnée si elle permet d'améliorer davantage la précision de la reconnaissance globale par rapport aux autres dans la limite des objectifs de précision demandés. Par exemple, si une précision à dix centimètres sur la position était recherchée, même si détecter la partie pourrait permettre d'être précis au millimètre près, ce n'est pas le plus intéressant : dans ce cas, on cherchera plutôt à garantir une meilleure confiance.

Probabilité de faire une bonne détection. La précision ne suffit pas à elle seule à établir un critère valable, il faut aussi tenir compte de la confiance. Celle-ci va évoluer en fonction de la probabilité de faire une bonne détection qui recoupe deux éléments :

- la probabilité de détecter une primitive compatible avec celle recherchée. Si un objet *pieton* recherché dispose d'une partie *lunettes* mais que seulement un pour cent des piétons possèdent des lunettes, cette partie ne sera probablement pas choisie du fait de sa faible probabilité d'existence même si réussir cette détection pourrait apporter beaucoup en précision. Si pour deux parties équivalentes en terme de temps de traitement et d'apport en précision, l'une possède un détecteur très fiable (s'il y a X primitives dans la zone, il renverra très exactement les X primitives) alors que l'autre possède un détecteur moins fiable car moins robuste ou plus sensible au bruit (il ne renverra que $0.8X$ primitives), la première sera évidemment à privilégier.
- la probabilité de faire le bon choix parmi les primitives détectées : imaginons un objet composé de dix parties dont neuf nécessitant la primitive *cercle* et une nécessitant la primitive *carré*. Supposons en outre qu'au départ, les zones de recherches de toutes ces parties soient confondues. Lorsqu'un *cercle* sera trouvé il pourrait appartenir à n'importe laquelle des neuf parties nécessitant un *cercle*, les chances de faire une mauvaise association sont donc élevées ce qui entraînera une chute de la confiance. Au contraire, une seule partie réclame un *carré*, il n'y a donc pas de risque de confusion.

Le gain des autres parties. Lorsque plusieurs parties nécessitent la même primitive et ont des zones qui se chevauchent, il y a un risque de confusion. Mais ce chevauchement peut aussi présenter des avantages : puisque les détecteurs renvoient toutes les primitives de la zone qu'ils ont détectées, il est possible qu'en plus d'avoir trouvé la primitive correspondant à leur partie, ils aient aussi trouvé la primitive d'une autre partie. Ainsi, pour cette partie au lieu de lancer son détecteur sur toute une zone, il lui suffira d'aller la chercher parmi la liste des primitives déjà trouvées (as-

pect bottom-up). Dans l'exemple du panneau de limitation, le *grandCercle* pourra être privilégié car il permettra très probablement de trouver le *petitCercle* en même temps.

Temps de calcul. Les parties seront également sélectionnées en fonction des temps de calculs nécessaires à leurs détecteurs pour trouver des primitives. Si une partie possède un détecteur extrêmement précis mais qui nécessite des temps de calculs très élevés alors qu'une autre partie apportera moins de précision mais pourra le faire rapidement, cette seconde semble, dans un premier temps, plus intéressante. L'autre partie sera traitée ultérieurement, lorsque, par focalisation, sa zone de recherche sera plus restreinte et donc moins coûteuse en temps.

L'apport informationnel d'une partie. Enfin, un dernier critère réside dans l'apport d'information que peut faire une partie : si une détection a déjà été tentée dans l'image pour une partie, elle ne pourra pas nous donner d'autres informations : soit elle a été détectée et l'objet a déjà profité de la connaissance de sa position, soit la détection a échoué et elle n'est donc pas détectable.

Une grande partie de ces sous-critères provient directement des objectifs de précision (estimer l'apport en précision de la partie) et de confiance (limiter les risques de confusion, choisir un détecteur fiable). Le reste provient d'un souci d'efficacité de l'algorithme (choisir les parties les plus probables, tenir compte des temps de détections...).

2.6 Détecteur et Détection

Caractéristiques du détecteur. A la fois pour pouvoir calculer le critère pour le choix de la meilleure partie (voir 2.5) et surtout pour la mise à jour de la confiance après l'échec ou la réussite d'une détection (2.8), il est nécessaire d'avoir les caractéristiques techniques des détecteurs.

- α : la probabilité qu'a le détecteur de ne rien renvoyer malgré la présence d'une primitive (qu'il sait normalement détecter) dans sa zone de recherche. Elle pourra éventuellement dépendre de la taille de la zone de recherche ou encore de la distance par rapport à l'objet. Dans une première approximation, on la considérera comme fixe.
- β : la probabilité qu'a le détecteur de renvoyer une détection sur une primitive qui n'existe pas réellement dans sa zone de recherche (il a confondu du bruit avec une primitive). Comme précédemment, on la considérera fixe bien que des critères identiques pourraient entrer en compte.
- t_{Calcul} : le temps de traitements nécessaire au détecteur pour extraire une primitive dans une zone de taille donnée.
- *précision* : la précision que garantit le détecteur sur sa mesure. En effet, la précision du détecteur influencera évidemment l'apport de précision pour l'état de l'objet si la détection réussie.

Détection.

1. Première phase : recherche dans l'existant

On dispose d'une liste $L_{primitives}$ de primitives déjà trouvées. Cette liste sera vide à l'étape 0 puis elle se remplira au fur et à mesure que des détections seront lancées par les différentes parties.

La détection d'une primitive débutera donc par une recherche dans cette liste. Seules les primitives du type adéquat seront examinées plus en détail (si un *cercle* est cherché, les *lignes* déjà détectées ne présentent aucun intérêt). Parmi celles-ci seront éliminées celles qui ne se trouvent pas dans la bonne zone et celles qui sont déjà utilisées (si un *cercle* a été associé à la partie *grandCerle* il ne pourra évidemment pas servir à la partie *petitCerle*). Parmi les primitives restantes, la meilleure sera sélectionnée. Si cette primitive convient (voir 2.7), la phase de détection se termine et indique qu'elle a réussi sinon, elle passe à la seconde phase.

2. Seconde phase : recherche de nouvelles primitives

Rien de probant n'ayant été trouvé dans la liste $L_{primitives}$ de primitives déjà détectées, le détecteur va donc chercher de nouvelles primitives dans la zone de recherche associée à sa partie. Il renvoie la liste complètes des primitives qu'il trouve. Ces primitives seront en partie ajoutées à $L_{primitives}$. En effet, certaines primitives avaient pu déjà être trouvées. Ensuite, la meilleure primitive parmi les nouvelles sera cherchée.

2.7 Choix de la meilleure primitive

Lors de la focalisation, en plus de déterminer la zone de recherche pour le détecteur, une valeur moyenne avec sa covariance a été calculée qui détermine les caractéristiques estimées de la primitive recherchée dans l'image. Néanmoins, l'intervalle de valeurs admissibles doit être augmenté en fonction de la précision du détecteur utilisé. En effet, le détecteur peut *mal* trouver une primitive : il estime sa position avec un léger décalage par rapport à la réalité du fait de son manque de précision or ce décalage peut suffire à la faire rejeter. Pour choisir la meilleure primitive parmi les primitives candidates, la distance de Mahalanobis entre la primitive trouvée et la primitive attendue est calculée : si E_{uv} est l'état attendu dans l'image, $covE_{uv}$ sa covariance associée (incluant l'erreur possible du détecteur) et \widetilde{E}_{uv} la mesure réalisée (correspondant à la primitive trouvée) alors on a :

$$d_{Mahalanobis} = \sqrt{(E_{uv} - \widetilde{E}_{uv})^t covE_{uv}^{-1} (E_{uv} - \widetilde{E}_{uv})}$$

La meilleure primitive est donc celle dont la distance est la plus faible. Elle sera considérée comme acceptable si : $d_{Mahalanobis} \leq D$ où D est le seuil d'acceptabilité. Si la meilleure distance est supérieure à ce seuil fixé, la détection aura échoué.

2.8 Mise à jour

La partie vient de finir sa détection. La réussite ou l'échec de celle-ci va nécessairement avoir un impact sur l'état de l'objet. Celui-ci va donc être mis à jour afin de tenir compte

des informations fournies par cette détection. Les deux cas vont être traités séparément.

Les événements suivants serviront pour le calcul de la confiance dans les deux cas :

– P_f^z : présence d'une primitive de type f dans une zone de taille z .

En effet, avoir détecté une primitive ne nous apportera que peu de garanties sur la localisation si cette primitive apparaît très fréquemment dans l'image. Plus la zone sera faible, moins il sera probable d'avoir réussie la détection par chance.

– d^z : faire une détection dans une zone de taille z .

La probabilité associée dépendra évidemment des caractéristiques du détecteur et de la confiance à l'état précédent. Si le détecteur est quasiment parfait pour une partie existente, ne pas la trouver entraînera une chute brusque de la confiance.

– \bar{d}^z : avoir échoué la détection dans une zone de taille z .

L'impact de l'échec d'une détection sur la confiance dépendra de la qualité du détecteur (est il possible qu'il ait manqué la primitive en dépit de sa présence) et de la probabilité d'existence de la partie associée (plus celle-ci est faible, moins un échec sera grave).

– E_k : être dans la zone définie par le vecteur d'état de l'objet et sa covariance. Par définition, $p(E_k) = pE_k$, il s'agit de la valeur de la confiance.

– v_k : la partie k est présente dans l'objet.

Rappel :

– α : la probabilité qu'a le détecteur de ne rien renvoyer malgré la présence d'une primitive (qu'il sait normalement détecter) dans sa zone de recherche.

$$D'où : \alpha = p(\bar{d}^z | P_f^z)$$

– β : la probabilité qu'a le détecteur de renvoyer une détection sur une primitive qui n'existe pas réellement dans sa zone de recherche.

$$D'où : \beta = p(d^z | \bar{P}_f^z)$$

Echec de la détection.

1. Mise à jour de l'état.

Sans primitive trouvée, aucune information supplémentaire sur la position (et l'incertitude liée) de l'objet ne peut être fournie. L'état mis à jour est donc strictement le même que le précédent ; il en va de même pour la covariance.

$$- E_{k+1} = E_k$$

$$- covE_{k+1} = covE_k$$

2. Mise à jour de la confiance.

Par contre, ne pas avoir détecté nous donne de l'information sur notre confiance.

$$\text{On cherche à calculer : } pE_{k+1} = p(E_k | \bar{d}^z)$$

$$\text{On obtient : } p(E_k | \bar{d}^z) = \frac{p(\bar{d}^z | E_k) p(E_k)}{p(\bar{d}^z)}$$

avec la formule de Bayes,

$$- p(E_k) = pE_k : \text{ connue}$$

$$- p(\bar{d}^z | E_k) = p(\bar{d}^z | E_k, v_k) p(v_k) + p(\bar{d}^z | E_k, \bar{v}_k) p(\bar{v}_k)$$

avec la formule des probabilités complètes

$$- p(\bar{d}^z | E_k, v_k) = p(\bar{d}^z | E_k, v_k, P_f^z) = \alpha.$$

En effet, on ne gère pas les occultations donc si la partie est présente, alors sa primitive doit se trouver dans la zone.

$$- p(\bar{d}^z | E_k, \bar{v}_k) = p(\bar{d}^z | E_k, \bar{v}_k, P_f^z) p(P_f^z) + p(\bar{d}^z | E_k, \bar{v}_k, \bar{P}_f^z) p(\bar{P}_f^z)$$

$$- p(\bar{d}^z | E_k, \bar{v}_k, P_f^z) = p(\bar{d}^z | P_f^z).$$

La réussite ou non de la détection ne dépend pas de l'état de l'objet mais juste de la présence ou de l'absence de primitives dans la zone (par contre la probabilité d'avoir une ou plusieurs primitives dans la zone observée dépend en partie de E_k). On a alors :

$$p(\bar{d}^z | E_k, \bar{v}_k, P_f^z) = \alpha$$

$$- p(\bar{d}^z | E_k, \bar{v}_k, \bar{P}_f^z) = 1 - \beta$$

$$- p(P_f^z) = 1 - p(\bar{P}_f^z) = 1 - (1 - p(P_f^{z_{ref}})) \frac{z}{z_{ref}}$$

$$- p(\bar{d}^z) = p(\bar{d}^z | P_f^z) p(P_f^z) + p(\bar{d}^z | \bar{P}_f^z) p(\bar{P}_f^z)$$

Réussite de la détection.

1. Mise à jour de l'état.

Un filtre de Kalman est utilisé pour mettre à jour notre état :

Prédiction

A priori, l'objet n'a pas bougé puisqu'on est toujours en train de chercher les différents éléments qui le composent dans une même image.

$$- E_{k+1|k} = E_k$$

$$- cov E_{k+1|k} = cov E_k$$

Mise à Jour

$$y_k = z_k - H_k E_{k+1|k}$$

$$S_k = H_k cov E_{k+1|k} H_k^T + cov Y_k$$

$$K_k = cov E_{k+1|k} H_k^T S_k^{-1}$$

$$E_{k+1} = E_{k+1|k} + K_k y_k$$

$$cov E_{k+1} = (I - K_k H_k) cov E_{k+1|k}$$

avec

z_k : observation ou mesure du process à l'instant k

H_k : matrice qui relie l'état E_k à la mesure z_k

$cov Y_k$: matrice de covariance du bruit de mesure

I : matrice identité aux dimensions adéquates

2. Mise à jour de la confiance.

On cherche à calculer : $p E_{k+1} = p(E_k | d^z)$ Avec la formules de Bayes, on obtient :

$$p(E_k | d^z) = \frac{p(\bar{d}^z | E_k) p(E_k)}{p(d^z)}$$

$$- p(d^z | E_k) = 1 - p(\bar{d}^z | E_k)$$

$$- p(d^z) = 1 - p(\bar{d}^z)$$

Pour plus de détails, se référer au cas de l'échec de détection, les calculs sont très similaires.

2.9 Retour

Il s'agit ici de prendre en compte les erreurs qui ont pu avoir lieu lors des étapes précédentes, en particulier les mauvaises associations qui ont pu être réalisées. Par exemple, la primitive qui aurait dû être associée au *petitCercle* a été associée au *grandCercle*. Il est évident

qu'avec cette mauvaise hypothèse, la partie *petitCercle* ne pourra pas être trouvée. Il faut donc revenir en arrière.

Critère de retour. Il faut définir un critère pour savoir à quel moment revenir en arrière et faire de nouvelles hypothèses.

– une première idée pourrait être de remonter dès que la confiance devient trop faible. Mais une faible confiance peut avoir plusieurs explications :

– plusieurs candidats possibles : il y avait plusieurs primitives qui auraient pu être sélectionnées pour la partie et donc une mauvaise association a pu survenir, ce qui diminue grandement la confiance. Pour autant, tant que la partie suivante n'a pas réalisé sa détection, il n'est pas possible de savoir si le choix précédent était le bon ou non.

– une ou plusieurs détections précédentes ont échouées : dans ce cas effectivement, sans indice pour corroborer les hypothèses faites précédemment, un retour en arrière semble nécessaire.

– une deuxième idée serait de revenir en arrière dès qu'une détection échoue mais ce n'est pas non plus un bon critère. En effet, cet echec peut survenir à cause d'une erreur du détecteur, d'un problème d'occultation ou encore à cause de l'inexistence d'une partie et non pas à cause d'une mauvaise hypothèse faite auparavant.

– il faut donc tenir compte des deux critères précédents : il faudra à la fois un échec de détection ET une confiance faible pour revenir en arrière. Ainsi, si la confiance est faible du fait d'une mauvaise association, la détection ayant réussie (sinon il n'y a pas d'association possible), cela reporte le retour éventuel à l'étape suivante et il sera donc fonction de la prochaine détection. Pour la détection, si elle a échoué en raison de la faible probabilité d'existence de sa partie, lors de la mise à jour, la confiance diminuera peu et donc le processus se poursuivra également.

Indice de retour. Une fois les conditions de retour remplies, le retour se fera jusqu'à l'état juste avant la dernière détection réussie. En effet, s'il y a un retour en arrière, cela doit venir d'une mauvaise association lors d'une des étapes précédentes, or pour avoir fait une mauvaise association, il faut nécessairement avoir réussi la détection. Pour éviter de refaire le même choix, la primitive qui semble avoir été mal associée sera marquée. Elle ne pourra donc plus être choisie à cette étape là, le choix se fera alors parmi les primitives adéquates restantes. S'il y avait une seule et unique primitive correspondant, le retour débouchera à l'itération suivante sur une non-détection qui entrainera un retour encore antérieur.

3 Resultats

Dans cette partie, nous présenterons les premiers résultats de cette approche dans le cadre d'une scène routière avec une première application qui cherchera les panneaux de limitations de vitesse et une seconde application pour les bords de route. (le modèle pris est celui de [1]).

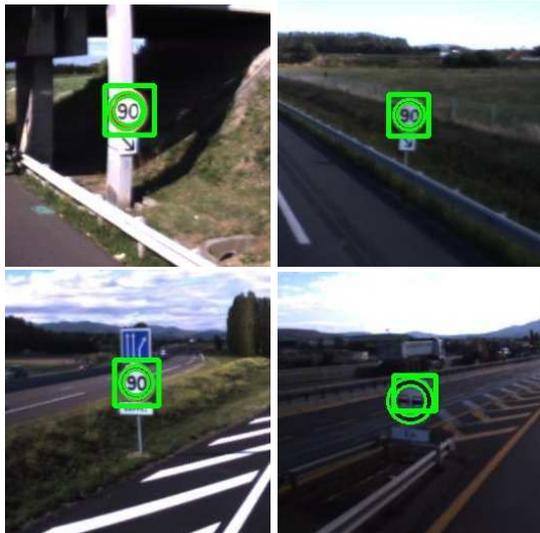


FIGURE 4 – quelques résultats pour la détection de panneaux

Sur la figure 4, les panneaux sont bien détectés. On remarque, sur la dernière image que tout le pourtour du panneau n'est pas considéré comme rouge. L'incertitude liée à cette détection étant assez élevée, les cercles sont malgré tout trouvés par la suite mais dans d'autres images, l'échec à trouver des zones rouges convenables a empêché la poursuite de la reconnaissance. Souvent, ce sont les résultats du détecteur de cercles qui sont insuffisants pour assurer la reconnaissance de l'objet.

Le seuil pour la confiance a été fixé à 0.85. Les valeurs de r et R_{ext} ont été fixés. La précision des détecteurs de cercles a été fixée à 4 pixels, celle pour le détecteur de rouge à 15 pixels. Avec ces valeurs, la précision obtenue est aux alentours d'une vingtaine de centimètres pour la position latérale, d'une quinzaine de centimètres pour la hauteur du panneau et d'environ deux mètres pour la profondeur. Cette dernière varie beaucoup en fonction de la distance du panneau : plus le panneau est estimé loin, plus la précision sur la profondeur sera faible. En effet, le panneau sera plus petit dans l'image et l'erreur de quatre pixels du détecteur représentera proportionnellement une part plus importante du rayon du panneau dans l'image.

Si l'algorithme est lancé sur plusieurs images successives (sans suivi), la position latérale et la hauteur va peu varier d'une image à l'autre alors que la profondeur va nettement diminuer au fur et à mesure (le véhicule se rapproche du panneau).

La méthode a ensuite été appliquée à la détection de bords de route grâce à son marquage. Pour ce faire, l'objet a été décomposé en dix-huit parties : neuf pour le bord gauche et neuf pour le bord droit. L'image est découpée en dix zones sur sa hauteur (les ordonnées du découpage sont figées). Les primitives seront alors les couples d'abscisses (u_b^i, u_h^i) tels que (u_h^i, v_h^i) et (u_b^i, v_b^i) soient les deux

extrémités d'une droite $((v_b^i, v_h^i))$ sont respectivement les ordonnées basses et hautes d'une zone par rapport au découpage de l'image). Le détecteur utilisé est ici un détecteur de lignes selon le principe de la transformée de Hough et le segment trouvé est prolongé si nécessaire pour prendre fin aux ordonnées données (la route est face au véhicule donc les lignes horizontales ne sont pas intéressantes et ne posent donc pas de soucis).

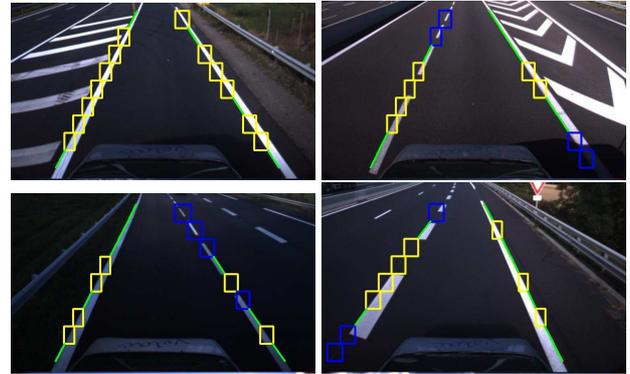


FIGURE 5 – quelques résultats pour la détection de bords de route

Pour la route, les lignes vertes représentent le résultat des détections, les zones jaunes représentent les parties pour lesquelles rien n'a été tenté et les zones bleues correspondent aux échecs de détection. Sur la première ligne, les bords de route détectés de chaque côté sont cohérents. Par contre, sur les autres images, le bord gauche de la ligne blanche de gauche a été associé avec le bord droit de la ligne de droite au lieu du bord gauche. En effet, une certaine latitude est autorisée sur la valeur de la largeur de la route de départ. Une fois le choix initial fait, les détections suivantes n'étant pas gênées par ce choix, l'objet est estimé détecté. On notera que l'échec de détection d'une ou plusieurs parties de la route n'empêche pas la détection de cette dernière même si trop d'échecs conduisent inévitablement au rejet de la solution. Par contre, la détection de toutes les parties n'est pas nécessaire pour remplir les objectifs fixés, par conséquent, le programme s'arrête dès que ceux ci sont remplis. C'est pourquoi, il reste des zones jaunes où aucune détection n'a été lancée.

La figure 6 montre l'évolution de la confiance au cours des différentes itérations. La première image indique l'ordre de détection des différentes parties et le graphique suivant montre les variations de la confiance en fonction de la réussite ou non des détections. Les flèches rouges correspondent aux échecs de détection. Le critère pour la confiance a été fixé à 0.95. Pour la position, une précision de cinq centimètres est demandée pour la position latérale et la largeur de la route. On remarque que le critère de confiance est rempli une première fois mais le programme ne s'arrête pas : en effet, le critère de position n'est pas encore atteint à l'itération correspondante. Avec les critères choisis, la détection d'une route nécessite environ 300 ms.



FIGURE 6 – évolution de la confiance au cours des itérations

4 Remerciements

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'avenir dans le cadre des projets EquipEx Robotex (ANR-10-EQPX-44) et LabEx IMobS3 (ANR-10-LABX-16-01), d'une aide de l'Union Européenne au titre du Programme Compétitivité Régionale et Emploi 2007-2013 (FEDER – Région Auvergne), d'une aide de la Région Auvergne et de l'Institut Français de Mécanique Avancée

5 Conclusion et perspectives

Cette méthode de reconnaissance permet de prendre en compte les connaissances a priori que nous avons sur la scène et les objets qui la composent. Elle s'adapte à différents objets tant qu'un découpage en partie avec primitive et détecteur associés est possible. Le processus est en outre capable de choisir de manière automatique selon ses connaissances actuelles dans quel ordre il doit lancer les détections. L'étape de focalisation permet de s'assurer de la cohérence des positions des différentes parties trouvées, elle diminue en outre les risques de mauvaises détections et le temps de traitement nécessaire pour chaque partie. Le travail sur la confiance permet de remettre en cause les hypothèses d'associations des différentes primitives.

L'approche est récente, en particulier toute la gestion de la confiance. Les premiers résultats sont encourageants. Les perspectives de ce travail portent sur plusieurs axes. En premier lieu, une optimisation des détecteurs utilisés devrait améliorer la fiabilité du processus et permettre de travailler

en temps réel. Ensuite, un suivi des objets reconnus sur une suite d'images prises par un véhicule serait intéressant. Enfin, le plus important sera de complexifier la scène afin de pouvoir reconnaître plusieurs objets en tenant comptes de leurs interdépendances non seulement en terme de position mais aussi de visibilité (ceci afin de permettre de gérer les occultations). Par exemple, il s'agira de gérer les interactions entre l'objet *route* et l'objet *panneau*

Références

- [1] R. Aufrère, R. Chapuis, and F. Chausse. Détection et suivi de bords de route en temps réel par vision artificielle. In *ORASIS 2001, Congrès francophone de vision, Cahors, 5-8 juin 2001*, pages 183–192.
- [2] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, pages 109–124, 2002.
- [3] Y.-T. Chen and C.-S. Chen. Fast human detection using a novel boosted cascading structure with meta stages. *Image Processing, IEEE Transactions on*, 17(8):1452–1464, aug. 2008.
- [4] M.L. Eichner and T.P. Breckon. Integrated speed limit detection and recognition from real-time video. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 626–631, 2008.
- [5] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, sept. 2010.
- [6] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.
- [7] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506, 2000.
- [8] I. Kokkinos, P. Maragos, and A. Yuille. Bottom-up and top-down object detection using primal sketch features and graphical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1893–1900, 2006.
- [9] G. Kootstra, N. Bergström, and D. Kragic. Gestalt principles for attention and segmentation in natural and artificial vision systems. In *Semantic Perception, Mapping and Exploration (SPME), ICRA 2011 Workshop*. eSMCs, 2011.
- [10] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 878–885, June 2005.
- [11] V. Navalpakkam. An integrated model of top-down and bottom-up attention for optimal object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2049–2056, 2006.
- [12] N. Trujillo-Morales. *Stratégie de perception pour la compréhension de scènes par une approche focalisante, application à la reconnaissance d'objets*. PhD thesis, Université Blaise Pascal - Clermont II, 2007.
- [13] L. Wang, J. Shi, G. Song, and I. F. Shen. Object detection combining recognition and segmentation. In *Computer Vision – ACCV 2007*, volume 4843 of *Lecture Notes in Computer Science*, 2007.