Spécification et validation à l'aide d'un langage synchrone d'un protocole d'appariement de données asynchrones

R REYNAUD°, Y. SOREL°°, C. LAVARENNE°°

°IEF Université Paris Sud - Orsay
°°INRIA Rocquencourt

RÉSUMÉ

On présente une méthode pour apparier les données fournies périodiquement par deux capteurs asynchrones. Ces capteurs font partie d'un système embarqué sur véhicule pour la détection d'obstacles par triangulation. Les deux capteurs infrarouges tournent approximativement à la même vitesse. Chaque capteur produit à chaque tour un tableau de bits ou chaque bit correspond a une direction donnée et indique si un obstacle a retourné un écho. Chaque tableau est envoyé a un calculateur par l'intermédiaire d'un réseau asynchrone local au véhicule. Le problème est d'apparier les données provenant des deux capteurs bien que ces données n'aient pas été acquises simultanément. Pour respecter les contraintes temps-réel, ce système réactif doit fournir un temps de réponse minimum. Le but de cet article est d'abord de choisir un protocole de synchronisation, puis de décrire les motivations qui nous ont conduit a utiliser un langage synchrone pour spécifier ce protocole, et enfin de spécifier et valider le protocole choisi dans le langage synchrone SIGNAL™.

ABSTRACT

We present a method to match data provided periodically by two asynchronous sensors. These sensors are part of an on-board vehicle system for obstacle detection by triangulation. The two infra-red sensors rotate approximately at the same speed. Each sensor produces a bit array at each revolution, each bit indicating whether an obstacle is echoed in a given direction. Each array is sent to a computer through an asynchronous vehicle area network. The problem is to match the data coming from the two sensors although these data are not simultaneously acquired. Due to real-time constableauts, this reactive system must provide a minimum response time. The goal of this paper is first to choose a synchronization protocol, then to describe the motivations that lead us to use a synchronous langage in order to specify the protocol and finally to specify and validate the chosen protocol in the SIGNAL synchronous language.

1 INTRODUCTION

On présente une méthode pour apparier les données fournies par deux capteurs tournants qui fonctionnent de façon totalement asynchrone. Ces capteurs font partie d'un système embarqué de détection d'obstacles utilisé pour l'aide à la conduite automobile [1,2]. Les deux capteurs actifs infrarouges tournent à des vitesses comparables mais peuvent dériver au cours du temps. Chacun renvoie à chaque tour un tableau de bits d'information, chaque bit indiquant la présence ou l'absence d'un écho dans une direction donnée. Les tableaux de bits sont envoyés à un calculateur par l'intermédiaire d'un réseau de communication.

La détection des obstacles utilise un principe de triangulation. Le problème est d'associer des informations provenant des deux capteurs alors que ces informations n'ont pas été acquises simultanément. L'aspect temps-réel de ce système réactif nous impose de plus la contrainte suivante : le temps de réponse de la chaîne réactive doit être minimal.

L'objectif de cet article est :
- de choisir un protocole de synchronisation des tableaux de bits provenant des capteurs droit et gauche ;
- de décrire les motivations qui nous poussent à utiliser un langage synchrone pour spécifier le protocole ;
- de spécifier et valider dans le langage synchrone SIGNAL™[3] plusieurs variantes de ce protocole.

Le protocole fonctionnel choisi consiste à associer les tableaux de bits les plus proches en temps en n'utilisant jamais deux fois le même tour, ce qui revient à dire que l'on accepte de "perdre" des tours lorsqu'un capteur tourne plus rapidement qu'un autre. A chaque fin de tableau de bit d'un capteur, nous mesurons l'écart de temps avec le dernier tableau de bits provenant de l'autre capteur. La contrainte de temps de réponse minimum nous oblige à prédire l'écart de temps avec le tableau de bits à venir. Nous savons alors si le tableau de bits actuel doit être associé avec le tableau de bits précédent ou suivant provenant de l'autre capteur.

2 APPLICATIONS

2.1 Détection d'obstacles

L'objectif de l'application que l'on vise est l'assistance à la conduite automobile par des moyens informatiques dans le cadre d'un projet européen PROMETHEUS PROCHIP.

La littérature scientifique est abondante d'approches pour résoudre ce difficile problème de la vision artificielle. La plupart de ces systèmes sont construits autour de caméras CCD. Le but recherché est la construction d'un modèle géométrique de la scène qui soit adapté à un traitement par ordinateur pour faire de la reconnaissance de formes, de la détection et de l'évitement d'obstacle, du guidage d'engins autonomes.

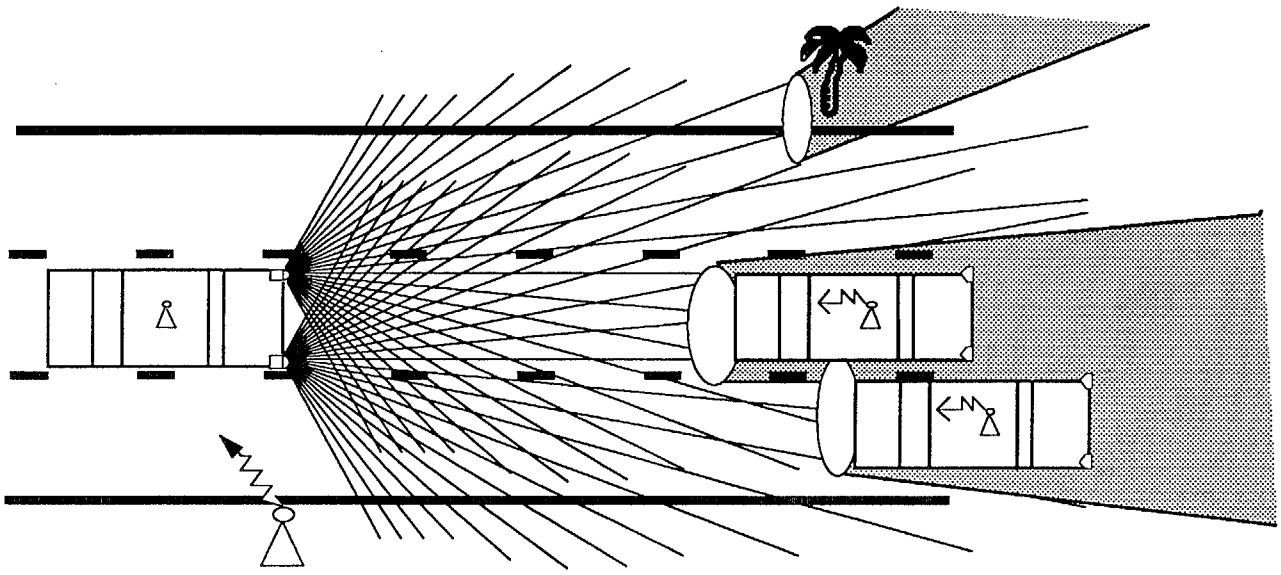

Figure 1 Détection des obstacles avec une représentation des tirs tous les 5 degrés sur cette figure.

Notre challenge est différent : il s'agit de faire un choix différent du classique système construit autour de caméras CCD, qui présente pour nous trop de défauts en particulier en ce qui concerne la saturation rapide des moyens de communication et de calcul embarqués sur le véhicule. Notre véhicule dit "intelligent" est équipé de capteurs frustes (par le débit d'information qu'ils injectent dans le système, les capteurs utilisés renvoient des images lignes binaires), de réseaux locaux et de moyens de calcul.

En supposant dans les deux cas que les traitements sont adaptés, on ne peut faire mieux que dans le cas d'un système à base de caméras CCD, l'abondance des données amenant une redondance d'information, et donc une robustesse supérieure si cette information est bien traitée.

Notre objectif est de montrer que l'on peut faire aussi bien dans le contexte des scénarii que l'on a choisi. On veut compenser l'aspect fruste des capteurs par l'introduction d'a priori et d'intelligence artificielle sous l'hypothèse que tous les acteurs de la scène sont coopératifs (utilisation de balises inter-véhicules) et que l'infrastructure a été étudiée pour répondre aux besoins du système embarqué (balises sur les éléments passifs, lignes réfléchissant correctement, ...).

Il est évident que la notion de coût est à l'origine de l'utilisation de capteurs frustes et que le coût d'un système est déterminant dans le milieu automobile. Mais notre objectif scientifique est de montrer qu'il est possible de choisir un autre compromis que celui choisi dans les systèmes à base de caméras : "observation abondante + traitement intensif bas-niveau". L'objectif correspondant à notre approche se définit par un compromis différent avec une information moins abondante mais, on espère, tout aussi pertinente + un traitement prenant en compte des modèles de plus haut niveau spatial et temporel.

La figure 1 montre les difficultés que l'on va rencontrer au cours de la reconstruction de la scène. La réponse du capteur est binaire et le réglage du seuil amène des non-détections. La triangulation avec un échantillonnage angulaire tous les degrés introduit des incertitudes sur la localisation des réflecteurs. Des ambiguïtés et masquages apparaissent en cas de plusieurs obstacles et d'objets non ponctuels introduisant des artefacts de reconstruction que l'on doit corriger.

Les outils et traitements utilisés pour reconstruire la scène sont pour la plupart classiques ou dérivés de traitements classiques et correspondent au suivi d'une trace temporelle avec suivi d'obstacles, génération d'obstacles potentiels, oubli d'obstacles, et évaluation d'une fonction de confiance de type probabiliste sur chaque obstacle.

Un autre rôle de la chaîne de traitement est d'adapter les formats pour passer d'une information physique qui est celle véritablement accessible par les capteurs et qui est composée de cônes de vision libres, cônes d'ombre masqués et de positions incertaines des réflecteurs à une information symbolique représentant la scène et qui est constituée d'une liste d'obstacles potentiels classés par urgence ou par ordre d'apparition. Chaque obstacle potentiel est constitué d'un identificateur, des positions en x & y, vitesses en x & y, vraisemblance, classe.

2.2 Asynchronisme des capteurs

La caractéristique qui nous intéresse dans ce papier concerne la spécification d'un protocole d'appariement de données asynchrones correspondant aux tableaux de bits représentant les échos binarisés reçus par les capteurs droit et gauche.

Le caractère asynchrone provient d'une nécessité économique pour la mise en œuvre des capteurs et du réseau. La synchronisation des capteurs imposerait le câblage d'un fil reliant les deux capteurs et une horloge de base. Or la tendance actuelle est de diminuer les coûts en prévoyant de commander l'ensemble des actionneurs et de récupérer l'ensemble des données à l'aide d'un réseau bus unique à 4 fils (Vehicle Area Network). Le réseau lui-même possède pour des raisons de fiabilité un fonctionnement autorisant la reprise de trames sur erreur de communication.

Le capteur lui-même réalise une scrutation continue dans un secteur angulaire fixe à l'avant de la voiture et récupère un écho tous les degrés. Lorsque l'on décide d'appairer un écho provenant

de chaque capteur pour réaliser une reconstruction de position par triangulation, les deux échos n'ont pas été acquis simultanément. C'est une cause d'imprécision notoire.

Mais il s'agit d'un problème classique d'échantillonnage et nous savons qu'il est toujours possible pour une horloge suffisamment fine de différencier deux événements émis par deux sources différentes. Ce qui nous intéresse est le corollaire, c'est-à-dire définir un grain de temps suffisamment grossier qui nous permettra de considérer que deux événements (émis par les capteurs droits et gauches) sont simultanés. Ce grain se définit en fonction des bases de temps naturelles de l'application.

Notre application possède 2 bases de temps. La première, la plus fine, correspond à la fréquence des échos. La deuxième correspond au nombre de tour par seconde. Comme les capteurs tournent à des vitesses non synchronisées et comme de plus il existe des fluctuations suffisantes dans les vitesses de rotation, il est illusoire de vouloir remonter à une précision correspondant à la base de temps la plus fine. Nous utilisons alors comme marquage temporel d'un tableau de bits l'occurrence du dernier bit du tableau comme indiqué sur la figure 2.

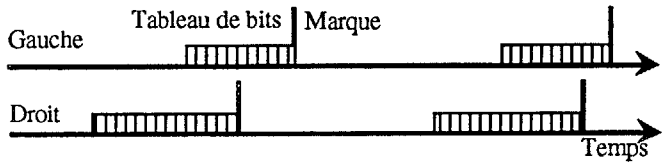

Figure 2 Définition du grain et de la marque temporelle associée.

On présente dans la suite différents protocoles d'appariement des tableaux de bits dont le rôle est de définir le grain de temps en fonction des bases de temps de l'application.

3 PROTOCOLES D'APPARIEMENT

Ce système réactif réagit à chaque appariement. Le temps de la réaction est borné (contrainte temps-réel) [4] par la durée minimum qui peut s'écouler entre deux appariements. On a intérêt à choisir le protocole qui maximise cette borne. Par ailleurs, pour obtenir la meilleure précision de triangulation, on a intérêt à minimiser le délai maximum qui peut exister entre les marques appairées. Nous allons comparer ces deux grandeurs sur 4 protocoles.

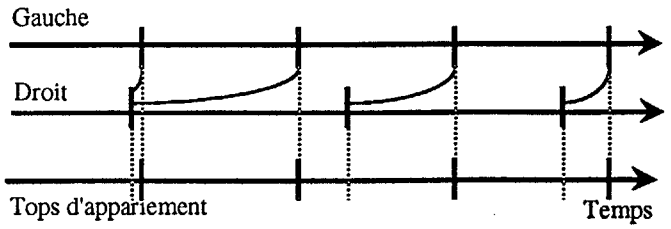

Figure 3 - Protocole Gauche

Protocole Gauche (Figure 3). La marque du top d'appariement est définie par la marque du tableau de bits du capteur gauche que l'on associe avec le tableau de bits du capteur droit précédent. L'association est symbolisée par l'existence d'un trait reliant les deux marques. On voit sur cet exemple que la borne temps-réel, correspondant au "grain temporel", est celui de la marque gauche et que le retard maximum entre 2 marques appairées peut être égal au grain.

Protocole Droit. Il s'agit du protocole symétrique. Pour que la borne temps-réel soit maximale, il faudrait choisir de se caler sur le capteur le plus lent. Mais comme ce n'est pas possible de connaître de quel capteur il s'agit, le pire cas correspond au capteur le plus rapide.

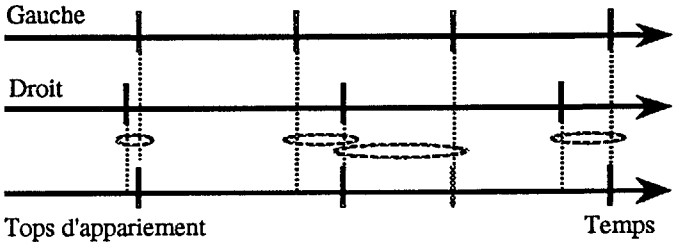

Figure 4 - Protocole au plus près

Protocole au plus près (figure 4) avec 2 variantes. Au plus près avec certains tableaux de bits utilisés deux fois. Au plus près avec non-prise en compte de tableaux de bits si nécessaire. Ici la marque en grisé correspond à un tour pour rien pour le capteur gauche. Dans les deux variantes, le retard maximum entre deux marques appairées est plus faible que pour les protocoles précédents, par contre le grain temporel est plus important pour le protocole avec suppression.

4 SPÉCIFICATION

4.1 Utilisation du langage SIGNAL™

La question est de savoir dans quel langage il est utile de spécifier les protocoles que l'on va vouloir tester. Le choix que nous avons fait ne porte pas sur les capacités de simulation et validation qui nous semblent équivalentes quel que soit le langage utilisé, mais bien sur la capacité de ce langage à spécifier de façon claire, simple et univoque les caractéristiques du protocole choisi.

Les langages synchrones et en particulier SIGNAL™ [3] ont justement été créés dans le but de spécifier de façon correcte des traitements sur des événements en temps logique avec au moment de la compilation une vérification sur la cohérence des horloges associées à ces événements en temps logique.

Notre problème était de définir la simultanéité des tableaux de bits pour autoriser leur appariement bien que ces données n'aient pas été acquises simultanément en temps physique. Et notre approche a consisté à définir un grain temporel correspondant à une certaine notion de simultanéité logique des événements à l'intérieur de ce grain sous deux contraintes : -maximiser l'intervalle de temps physique minimum sur lequel on peut tabler pour réaliser une reconstruction; - minimiser le retard physique maximum entre deux tableaux appairés.

Après avoir défini ce grain, nous sommes retombés au niveau de la spécification dans un monde synchrone et il est alors naturel d'utiliser un langage synchrone pour spécifier et valider le protocole.

4.2 Programme

Voici le source SIGNAL™ de la partie resynchronisation "au plus près" sous ces deux variantes, suivi de ce a quoi pourrait ressembler le programme "appariement" qui mémorise les valeurs des signaux et passe leurs horloges a "resynch". On peut voir au passage que le cas SUPpression d'un tableau de bits est un peu plus complexe à spécifier que le cas DUPlication d'un tableau de bits.

```
process appariement = (integer N)
{ ? [N]logical Ge, De; logical G, D
  ! [N]logical Gs, Ds
}
(| synchro{G, D, Gi, Di}
 | top := resynch(N){G, D}
 | synchro{when G, Ge}
 | Gi := Ge default zGi | zGi := Gi$1
 | Gs := Gi when top
 | synchro{when D, De}
 | Di := De default zDi | zDi := Di$1
 | Ds := Di when top
 |)
where
  [N]logical Gi, zGi init [{to N}:false];
  [N]logical Di, zDi init [{to N}:false];
  event top
  process resynch = (integer N) { ? logical G, D
                                  ! event sup }
(| synchro{G,D,dt}
 | synchro{when G default when D, un,dtG,dtD,dts}
 | un := 1
 | dt := un default zdt+1 | zdt := dt $ 1
 | zero := 0 when G when D
 | dtG := zero default zdt when G default zdtG
 | zdtG := dtG $ 1
 | dtD := zero default zdt when D default zdtD
 | zdtD := dtD $ 1
 | dup := when G when dtG<=dtD
          default when D when dtD<=dtG
 | dts := zero default zdt when sup default zdts
 | zdts := dts $ 1
 | sup := when G when (dtG<=dtD and zdts/=zdtD)
          default
          when D when (dtD<=dtG and zdts/=zdtG)
 |)
where
 event dup;
 integer dt, zdt init 0 ;
 integer dtG, zdtG init 0;
 integer dtD, zdtD init 0;
 integer un, zero, dts, zdts init -1
end
end
```

4.3 Validation

G et D sont deux signaux booléens synchrones dont l'horloge est périodique et dont chaque occurrence true représente l'instant de disponibilité de nouvelles données (la marque) des capteurs gauche et droit respectivement.

G	T	F	F	F	F	T	F	F	F	F	T	F	F	F	F	T	F	F	F	F	T
D	T	F	F	F	T	F	F	F	T	F	F	F	T	F	F	F	T	F	F	F	T
dup	T					T					T			T			T,				T
sup	T					T					T						T				T
dt	1	2	3	4	1	1	2	3	1	2	1	2	1	2	3	1	1	2	3	4	1
dtG	0				0	1			1		2		2			3	3				0
dtD	0				4	4			3		3		2			2	1				0
dts	0				4	1			3		2		2			3	1				0

Tableau 1 - Affichage des résultats de validation

Le tableau précédent est un exemple de résultat de simulation dans laquelle on a testé les 2 variantes du protocole au plus près. dup et sup donnent les instants d'appariement des données "au plus près", la différence entre dup et sup n'apparaissant qu'au moment où le capteur le plus rapide "dépasse" le capteur le plus lent, avec pour dup duplication (utilisation pendant deux appariements successifs) des données du capteur le plus lent, et avec pour sup suppression (perte d'un tour) des données du capteur le plus rapide. dt mesure le délai, compté relativement a l'horloge périodique de G et D, écoulé entre deux occurrences true de G ou D. dtG et dtD mémorisent dt à chaque occurrence true de G et D respectivement.

CONCLUSION

La spécification d'un protocole d'appariement est un des points de passage obligé de cette application à base de capteurs asynchrones. La notion de simultanéité des observations droite et gauche amène la définition d'un grain de temps, qui nous autorise à considérer que deux événements sont simultanés. Le rôle du protocole est alors la définition précise de ce grain que nous avons réalisée sous deux contraintes : maximiser la borne temps-réel et minimiser le délai entre deux événements appairés. La spécification en langage synchrone est alors la mieux adaptée à la prise en compte de cette notion de simultanéité. Les perspectives de ce travail vont porter sur l'intégration de ce protocole dans un exécutif distribué en utilisant l'atelier logiciel SynDEx™ [3]. Cet exécutif comprendra l'ensemble des traitements de l'application et sera réparti sur l'ensemble des moyens de calcule embarqué.

BIBLIOGRAPHIE

[1] A. Chebira, R. Reynaud & G. Demoment, "Fusion de données multi-capteurs : application à la détection d'obstacles en temps-réel", Colloque GRETSI, pp. 73-76, Juan-les-Pins, Sep 91.

[2] A. Chebira, R. Reynaud, T. Maurin, D. Berschandy, "on board data fusion and decision system used for obstacle detection : a network and a real-time approach", Euromicro'91 worhshop on real time systems, pp193-200, Paris, 1991.

[3] C. Lavarenne, O. Seghrouchni, Y. Sorel, M. Sorine. "SynDEx un environnement de programmation pour applications de traitement du signal distribuées". 13ème colloque GRETSI, 1991.

[4] A. Benveniste, M. Le Borgne, P. Le Guernic. "SIGNAL as a model for real-time and hybrid systems". rapport de recherche INRIA, 1988.