

# Gestion entropique du flot d'informations et minimisation des ressources matérielles en fusion d'informations

## Entropy data flow management and minimization of the hardware resources in information fusion

par **Bienvenu FASSINUT-MOMBOT, Jean-Bernard CHOQUEL**

Laboratoire d'Analyse des Systèmes du Littoral (LASL-UPRES EA 2600), Université du Littoral Côte d'Opale (ULCO) - BP 699, 62228 Calais Cedex, France  
Tel : +33(0)3.21.46.06.87 Fax : +33(0)3.21.46.06.86  
Bienvenu.Fassinut-Mombot@lasl-gw.univ-littoral.fr  
Jean-Bernard.Choquel@lasl-gw.univ-littoral.fr

### *résumé et mots clés*

Nous proposons dans cet article une approche de fusion probabiliste reposant sur des critères entropiques dont le but est de réduire l'espace de combinaison en représentant explicitement les notions de redondance et de complémentarité des sources d'information. Ce type de modélisation est en particulier intéressant pour optimiser le choix des mesures, issues des sources d'information, à combiner dans un système de fusion. Il est en accord avec le souci de rapidité de traitement et de minimisation des ressources matérielles qui se pose en fusion d'informations. Pour répondre à cela, nous avons réalisé une étude de la parallélisation de l'algorithme de fusion entropique développé en vue de son implantation parallèle dans le cadre d'une application en robotique mobile. La spécification de l'algorithme faisant apparaître du parallélisme potentiel est ensuite implantée sur un réseau de stations de travail fonctionnant en mode MIMD-NORMA à l'aide des environnements de programmation SynDEx, qui supporte la méthodologie AA-A, et PVM, qui est de type CSP de Hoare.

**Fusion d'informations multi-sources, gestion du flot d'informations, modèle de fusion entropique, implantation parallèle/distribuée, SynDEx, PVM.**

### *abstract and key words*

In this paper we describe a probabilistic fusion approach based on entropic criteria, which aims at reducing the combination space by explicitly representing the notions of source redundancy and source complementarity. This modelling is particularly interesting to optimize the choice of measurements provided by sources in order to combine in multi-sources fusion system. It is in agreement with the preoccupation to perform efficiently fusion processing and to minimize the hardware resources in information fusion problem. To answer that, we made a study

of the parallelization of the entropy fusion algorithm developed for its parallel implementation in the framework of an application to mobile robotics. The algorithm specification exhibiting potential parallelism is then implemented on a network of workstations running in mode MIMD-NORMA using the parallel/distributed programming environments SynDEx which support AA-A methodology and PVM which support Hoare's CSP concept.

Multi-sources information fusion, data flow management, entropy model fusion, parallel/distributed implementation, SynDEx, PVM.

# 1. introduction

La fusion d'informations propose de fournir un outil d'aide à la décision en présence de sources d'information multiples. Pour cela, elle s'appuie généralement sur la théorie des probabilités associée à la théorie bayésienne de la décision [Duda73] et les théories de l'incertain non probabilistes (la théorie des possibilités associée à la théorie des ensembles flous [Zadeh78] [Dubois88], la théorie des fonctions de croyance [Demp68] [Shafer76]) pour évaluer la *quantité*<sup>1</sup> et la *qualité*<sup>2</sup> de l'information apportée par chacune des sources et pour combiner ces informations afin de déterminer une décision qui soit la plus précise et la plus robuste possible. Une source d'information peut représenter un capteur, un expert humain ou une base de données.

D'une façon générale, l'utilisation de plusieurs sources d'information offre de nombreux avantages parmi lesquels :

- la redondance des sources d'information qui offre une grande robustesse de l'information en permettant de faire face à la défaillance de l'une des sources d'information (tolérance aux pannes) et permet ainsi d'augmenter la précision du résultat de la décision ;
- la complémentarité des sources d'information qui permet de déduire une information globale jugée la plus fiable et la plus complète concernant certains aspects du problème traité qu'une source d'information, opérant individuellement, serait incapable de saisir.

Ces deux notions de complémentarité et de redondance forment une dualité : on ne peut en privilégier une qu'au détriment de l'autre.

Si la théorie des probabilités associée à la théorie bayésienne de la décision et les théories de l'incertain non probabilistes permettent de prendre en compte, chacune à leur façon, dans le pro-

cessus de fusion, la modélisation des informations imparfaites (incertaines, imprécises et incomplètes) et la combinaison des informations hétérogènes (numériques et/ou symboliques), elles ne permettent pas de prendre en compte la modélisation de la redondance et/ou de la complémentarité des informations qui sont nécessaires à la gestion du flot d'informations. En effet, lorsqu'on dispose de plusieurs sources d'information, l'ensemble de ces informations peut être trop important et la combinaison de telles données est en général complexe. Cette complexité s'exprime à travers la complémentarité qui élargit l'espace de combinaison, et des redondances qui n'améliorent pas souvent les performances du système de fusion, ou noie celui-ci sous un flot d'informations trop coûteux à gérer.

Bien que les théories de l'incertain non probabilistes ont été introduites pour faire face aux inconvénients rencontrés en théorie des probabilités [Bloch94], leur principal inconvénient se situe au niveau de la phase de combinaison. La théorie des possibilités offre pourtant une multitude d'opérateurs de combinaison mais elle ne donne aucune règle quant au choix de l'opérateur approprié. Dans la théorie des fonctions de croyance, la combinaison orthogonale de  $S$  sources d'information entraîne une explosion combinatoire pour constituer la *distribution de masse* unique. Pour ces raisons, nous proposons de développer en fusion d'informations une approche probabiliste originale s'appuyant sur la théorie de l'information [Fass02] qui tient compte des remarques précédentes afin d'optimiser le choix des mesures à combiner (*cf.* tableau 1).

Par ailleurs, la gestion du flot d'informations peut influencer fortement sur la complexité algorithmique de la méthode de fusion, sur sa mise en œuvre, et sur la durée de traitements (calculs et communications) de l'algorithme par une architecture informatique (ordinateurs, systèmes embarqués ou autres). Pour cela, nous avons étudié la parallélisation et l'implantation, sur un réseau de stations de travail<sup>3</sup> fonctionnant en mode MIMD-

<sup>1</sup> Elle est liée à la multiplicité des sources d'information.

<sup>2</sup> Elle est liée aux notions de redondance et de complémentarité des sources d'information.

<sup>3</sup> Les réseaux de stations de travail individuelles représentent l'architecture parallèle/distribuée la plus familière et la plus répandue actuellement [Cappel98] [Gatin00].

**Tableau 1. – Gestion du flot d'informations par les différents formalismes et l'approche probabiliste et entropique.**

Approche	Redondance	Complémentarité	Espace de combinaison
Probabiliste	implicite	implicite	important
Possibiliste	implicite	implicite	important
Crédibiliste	implicite	implicite	important
Probabiliste et Entropique	explicite	explicite	réduit

NORMA<sup>4</sup>, de l'algorithme de fusion entropique développé à l'aide des environnements de programmation SynDEx (*SYNchronized Distributed EXecutive*) [Lava91] [Sorel94], qui supporte la méthodologie AA-A (*Adéquation Algorithmique-Architecture*) [Lava93] [Sorel94], et PVM (*Parallel Virtual Machine*) [Gesit94], qui est de type CSP (*Communicating Sequential Processus*) de Hoare [Hoare85].

La méthodologie AA-A consiste à étudier simultanément les aspects algorithmiques et architecturaux ainsi que leurs interactions dans le but d'effectuer une implantation efficace et optimisée de l'algorithme sur l'architecture, tout en satisfaisant certaines contraintes exécutives et temporelles. En revanche, le modèle CSP (en français, *Processus Séquentiels Communicants*) proposé par Hoare définit des processus asynchrones les uns par rapport aux autres, qui ne partagent pas de mémoire et ne communiquent que par *rendez-vous* (échange de données et synchronisation). Dans ce modèle, le programmeur n'a pas le contrôle complet de l'ordonnement des tâches, ce qui peut rendre, par exemple, l'évaluation des durées d'exécution plus délicat qu'avec un programme séquentiel.

Notre but à travers cette expérience est d'illustrer comment optimiser l'implantation parallèle, en prenant en compte la spécification de l'algorithme et de l'architecture, la granularité<sup>5</sup> du parallélisme, les contraintes temps-réel et en minimisant les ressources matérielles (nombre de processeurs et de liaisons de communication).

<sup>4</sup> Les réseaux de stations de travail font partie des machines MIMD-NORMA (*Multiple Instruction Multiple Data stream-NO Remote Memory Access*) à mémoire distribuée car ils sont composés de stations de travail possédant un processeur, une mémoire locale (privée) physiquement distribuée mais non partagée et une interface réseau leur permettant d'échanger des messages entre elles [Blaze00].

<sup>5</sup> Le terme *granularité* est employé pour désigner la taille des tâches ou la longueur des séquences de calcul effectuées en parallèle (volume de traitement effectué par le processeur). Il est également employé pour décrire la taille des ressources matérielles disponibles sur une architecture et sur lesquelles les tâches vont s'exécuter.

L'article est organisé autour de 7 sections principales. Après une introduction sur la problématique abordée en section 1, la section 2 propose une méthode de fusion probabiliste et entropique conduisant à un algorithme séquentiel d'Agrégation Adaptative Entropique (A2E) présenté en section 3. Dans la section 4, nous présentons une application de l'algorithme A2E dans le cadre de la robotique mobile. La section 5 expose une description succincte des deux environnements de programmation parallèle, SynDEx et PVM, utilisés par la suite. La section 6 présente, dans le cadre de l'application de robotique, l'étude de la parallélisation de l'algorithme A2E en vue de son implantation en parallèle sur une architecture parallèle formée de plusieurs stations de travail. Enfin, nous concluons sur le sujet en section 7.

## 2. approche probabiliste et entropique

### 2.1. évaluation de la redondance et de la complémentarité

Considérons la variable vectorielle  $X$  (l'entrée) et la variable vectorielle  $Y$  (la sortie) d'un système de fusion. Les entropies « simples »  $H(X)$  et  $H(Y)$  mesurent l'information délivrée par chaque variable considérée isolée et l'entropie mutuelle  $H(X, Y)$  mesure l'information délivrée par la réunion des deux variables. La *propriété de sous-additivité* s'écrit :

$$H(X, Y) \leq H(X) + H(Y) \quad (1)$$

avec égalité *Ssi* (si et seulement si)  $X$  et  $Y$  sont des variables statistiquement indépendantes. Cette propriété indique que l'information apportée par la réunion de plusieurs groupes de variables est inférieure ou égale à la somme des informations apportées par chacun d'eux. De plus, dans l'expression  $H(X) + H(Y)$ , on compte deux fois une partie de l'information qui se trouve aussi bien dans  $X$  que dans  $Y$ . Une mesure de cet échange d'information, appelée *information mutuelle* ou encore *transinformation*, notée  $I(X, Y)$ , est définie par (cf. figure 1) :

$$I(X, Y) \triangleq H(X) + H(Y) - H(X, Y) \quad (2)$$

$$\geq 0 \quad (3)$$

avec égalité *Ssi*  $X$  et  $Y$  sont des variables statistiquement indépendantes. La quantité  $I(X, Y)$  est une mesure de la dépendance statistique entre les variables  $X$  et  $Y$ . L'information mutuelle mesure donc la quantité d'information qui se trouve aussi bien en  $X$  qu'en  $Y$  et que l'on suppose transmise entre ces deux variables. **Elle permet de mesurer la redondance de l'information transmise.**

On peut aussi s'interroger sur la quantité d'information dont on aurait besoin pour connaître la sortie  $Y$  une fois connue l'entrée  $X$  du système. L'entropie conditionnelle de Shannon répond à ce besoin. Elle est obtenue en soustrayant à l'entropie simple  $H(Y)$  de la sortie l'information mutuelle  $I(X, Y)$  transmise entre l'entrée  $X$  et la sortie  $Y$  du système. Elle est donnée par l'expression suivante (cf. figure 1) :

$$H(Y/X) \triangleq H(Y) - I(X, Y) \quad (4)$$

$$= H(X, Y) - H(X) \quad (5)$$

$$\geq 0 \quad (6)$$

La quantité  $H(Y/X)$  s'interprète comme l'incertitude sur la variable  $Y$  lorsque l'on connaît l'information fournie par la variable  $X$ . Elle s'obtient également en soustrayant à l'entropie mutuelle  $H(X, Y)$  le surplus d'information apporté par l'entropie simple  $H(X)$ . L'entropie conditionnelle  $H(Y/X)$  mesure également la quantité d'information que fournit encore la sortie  $Y$  lorsque l'on fixe l'entrée  $X$  concernée du système. **Elle permet de mesurer la complémentarité de l'information fournie.** Shannon a appelé l'entropie conditionnelle l'équivoque.

Le caractère dual de la redondance et de la complémentarité se retrouve avec la théorie de l'information. En effet, d'après les équations 2 et 4 on a la relation suivante :

$$I(X, Y) + H(Y/X) = H(Y) = \text{constante} \quad (7)$$

Cette formule s'interprète comme le fait que l'information délivrée par la sortie  $Y$  du système est séparable en deux parties : l'information commune avec l'entrée  $X$  du système (l'information de redondance) représentée par l'information mutuelle, et l'information spécifique de  $Y$  (l'information de complémentarité) représentée par l'entropie conditionnelle de Shannon.

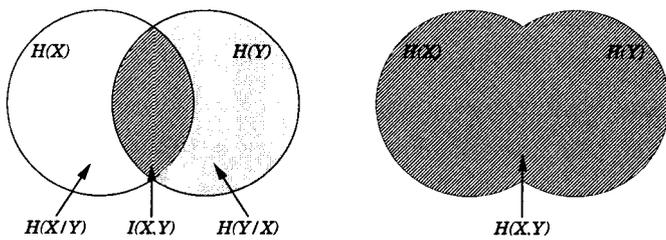


Figure 1. – Diagramme de Venn : relation entre entropie simple, entropie mutuelle et information mutuelle.

## 2.2. processus de fusion d'informations

La méthode de fusion par l'approche entropique, que nous désignons par *Modèle de Fusion Entropique* en abrégé MFE [Fass00] [Fass02], est une méthode probabiliste s'appuyant sur

la notion d'information mutuelle de Shannon, ou par équivalence sur celle d'entropie conditionnelle, issue de la théorie de la communication [Shan48] et de la théorie de l'information [Shan49]. Cette approche de la fusion d'informations fournit une représentation de l'incertitude relative à la vérité d'une information par une probabilité  $P$  et l'imprécision sur cette incertitude (contenu de l'information) par l'entropie de la probabilité  $H(P)$ .

L'objectif du MFE est de rechercher le sous-ensemble de variables  $\tilde{X}^*$  qui maximise l'information mutuelle  $I(X, Y)$  entre l'entrée  $X$  et la sortie  $Y$  du système de fusion et qui par conséquent réduit l'espace de combinaison du système de fusion. D'après l'équation 7, maximiser l'information mutuelle  $I(X, Y)$  (resp. minimiser les redondances des sources d'information) revient par équivalence à minimiser l'entropie conditionnelle  $H(Y/X)$  (resp. maximiser la complémentarité entre les sources d'information), c.-à-d. on recherchera le sous-ensemble de variables  $\tilde{X}^*$  du système de fusion tel que :

$$\tilde{X}^* = \underset{\tilde{X} \geq X}{\text{arg min}} \left\{ H(Y/\tilde{X}) \right\} \quad (8)$$

où  $\tilde{X} \geq X \implies H(Y/\tilde{X}) \leq H(Y/X)$  indique que le sous-ensemble de variables le plus gros  $\tilde{X}$  apporte plus d'information que l'ensemble de variables le plus fin  $X$ . La *relation de finesse*  $\geq$  décrit un *ordre partiel* [Toro82] et a les propriétés de réflexivité, de transitivité et d'anti-symétrie, c.-à-d.,  $X_1 \geq X_1$  ; si  $X_1 \geq X_2$  et  $X_2 \geq X_3$  alors  $X_1 \geq X_3$  ; si  $X_1 \geq X_2$  et  $X_2 \geq X_1$  alors  $X_1 = X_2$ .

Le sous-ensemble de variables **réduit**  $\tilde{X}$  décrit aussi bien que l'ensemble de variables  $X$  la connaissance extraite de l'observation du phénomène considéré. Ainsi le sous-ensemble de variables  $\tilde{X}^*$  du système recherché représente le « bon » compromis que l'on fait sur la dualité des notions de redondance et de complémentarité des sources d'information.

### 2.2.1. étape de modélisation et apprentissage

#### Formulation

Le mécanisme d'inférence bayésien peut être décrit comme suit :

- on définit l'ensemble  $\Omega$  d'événements possibles  $\{H_k\}_{k=1}^K$ . Cet ensemble  $\Omega$  désigne le *cadre de discernement*, composé de  $K$  hypothèses  $H_k$ , pouvant être conditionnées par l'observation de  $S$  sources d'information  $x_s$ , noté  $x$  ;
- chaque source d'information  $x_s$  fournit un vecteur de mesure  $d^s = \{d_k^s\}_{k=1}^K$  reliant les caractéristiques de l'événement, de la situation ou du phénomène observé aux hypothèses possibles  $H_k$  ;

- chaque hypothèse  $H_k$  de  $\Omega$  est définie par un ensemble de  $M$  paramètres  $z_m$ , noté  $z$ . Par exemple, si on a deux paramètres caractéristiques  $z = \{z_1, z_2\}$  de la discrimination d'un objet observé, où le paramètre  $z_1 =$  forme peut prendre les modalités {carrée, rond, rectangulaire}, et le paramètre  $z_2 =$  taille peut prendre les modalités {petit, grand}, un objet balle de tennis peut être défini comme « rond et petit », et un objet fenêtre peut être défini comme « rectangulaire et grand » ;
- on suppose que, parmi toutes les hypothèses recensées, il y en ait une et une seule qui soit solution du problème à traiter (*propriété d'exclusivité et d'exhaustivité*).

On peut alors écrire :

$$\left\{ \begin{array}{l} \forall \kappa \neq k, \quad P(H_k \cap H_\kappa) = 0 \quad \text{car} \quad H_k \cap H_\kappa = \emptyset \\ \text{et} \quad \sum_{k=1}^K P(H_k) = 1 \end{array} \right. \quad (9)$$

où, la première ligne de l'équation 9 indique le caractère exclusif de l'ensemble des hypothèses, et signifie que la solution est unique. La deuxième ligne, quant à elle, indique le caractère exhaustif de la solution, elle est obligatoirement l'une des hypothèses du cadre de discernement. Cela correspond à l'hypothèse du monde fermé ou *closed-world* (on suppose tout connaître du problème). L'existence d'éventuelles hypothèses non modélisées (monde ouvert ou *open-world*) est souvent introduite explicitement dans  $\Omega$  par une *hypothèse d'ignorance* ou une *hypothèse de rejet*, ou encore une *hypothèse de conflit* [Smets90].

La fonction  $P$ , appelée *mesure de probabilité*, est une application de l'ensemble des  $2^K$  parties  $A$  ou sous-ensembles du cadre de discernement  $\Omega$ , notée  $2^\Omega$ , dans  $[0, 1]$ , définie par :

$$P : \left[ \begin{array}{ll} 2^\Omega & \longrightarrow [0, 1] \\ A & \longmapsto P(A) \end{array} \right] \quad (10)$$

où  $A$  est appelé un *événement aléatoire*. L'ensemble des  $2^K$  parties  $A$  de  $\Omega$ , dont l'expression est la suivante :

$$\begin{aligned} 2^\Omega &= \{A / A \subseteq \Omega\} \\ &= \left\{ \underbrace{\emptyset, H_1, H_2, \dots, H_K}_{\text{hypothèses simples}}, \underbrace{H_1 \cup H_2, H_1 \cup H_3, \dots, \Omega}_{\text{hypothèses composées}} \right\} \end{aligned} \quad (11)$$

est composé non seulement d'hypothèses simples, mais également de l'ensemble de toutes les disjonctions possibles de ces hypothèses (hypothèses composées).

Par analogie, la variable vectorielle  $X$  ou entrée du système de fusion correspond aux vecteurs de mesure  $d^s$  des  $S$  sources d'information  $x_s$ . La variable vectorielle  $Y$  ou sortie du système de fusion correspond à l'une des hypothèses  $H_k$  de  $\Omega$ .

### Modélisation de la confiance

La représentation de l'information concerne le choix du meilleur format et du meilleur niveau de représentation de l'information pertinente. Dans la théorie des probabilités, associée à la théorie bayésienne de la décision [Duda73], l'information est caractérisée par une distribution de probabilité conditionnelle. Dans notre cas, il s'agit de la probabilité qu'une hypothèse  $H_k$  de  $\Omega$  soit l'hypothèse recherchée, étant données les observations issues des sources d'information. D'après la règle de Bayes, on a :

$$\forall H_k \in \Omega, \quad P(z_m^k / d_k^s) \triangleq \frac{P(d_k^s, z_m^k)}{\sum_{m=1}^M P(d_k^s, z_m^k)} \quad (12)$$

où  $P(d_k^s, z_m^k)$  désigne la probabilité conjointe de réalisation simultanée d'une observation  $d_k^s$  par la source d'information  $x_s$  et d'une caractéristique  $z_m^k$  de l'hypothèse  $H_k$  de  $\Omega$ .

### Modélisation de la méconnaissance

L'imprécision sur le contenu de l'information modélisée  $P(z_m^k / d_k^s)$  est donnée par :

$$\forall H_k \in \Omega, \quad H(H_k / d_k^s) = \sum_{m=1}^M P(z_m^k / d_k^s) \log P(z_m^k / d_k^s) \quad (13)$$

où  $H(H_k / d_k^s)$  représente l'entropie conditionnelle de  $H_k$  lorsque les observations sont disponibles successivement. Cette mesure permet de modéliser la méconnaissance sur  $H_k$  une fois connu l'observation  $d_k^s$  de la source d'information  $x_s$ .

### 2.2.2. règle de combinaison

La règle de combinaison entropique 8 consiste à résoudre un problème d'optimisation, c.-à-d. à trouver le sous-ensemble des mesures optimal  $\tilde{d}_k^*$  tel que l'information mutuelle  $I(\tilde{d}_k^*, H_k)$  soit la plus grande possible (*resp.* l'entropie conditionnelle  $H(H_k / \tilde{d}_k^*)$  soit la plus petite possible). Cette règle peut être réécrite comme suit :

$$\forall H_k \in \Omega, \quad \tilde{d}_k^* = \arg \min_{\tilde{d}_k \geq d_k} \left\{ H(H_k / \tilde{d}_k) \right\} \quad (14)$$

où

$$H(H_k / \tilde{d}_k) \triangleq H(\tilde{d}_k, H_k) - H(\tilde{d}_k) \quad (15)$$

$$= \sum_{s=1}^S P(d_k^s) \times H(H_k / \tilde{d}_k = d_k^s) \quad (16)$$

$$= - \sum_{s=1}^S \sum_{m=1}^M P(d_k^s, z_m^k) \log P(z_m^k / d_k^s) \quad (17)$$

avec  $P(d_k^s, z_m^k) = P(d_k^s) \times P(z_m^k/d_k^s)$ . Le sous-ensemble des mesures optimal  $\tilde{d}_k^*$  exprime la pertinence des informations (incertitude et imprécision) fournies par les différentes sources d'information, relativement à une hypothèse particulière  $H_k$  de  $\Omega$ .

Le but visé par le choix d'un tel critère est de pouvoir minimiser la complexité de la combinaison des informations (ou mesures). En effet, certaines informations fournies peuvent être redondantes et/ou complémentaires pour le problème de fusion, ou peuvent noyer le système de fusion sous un flot d'informations trop coûteux à gérer. La minimisation de l'entropie conditionnelle  $H(H_k/\tilde{d}_k^*)$  permet alors d'aboutir à une décision sur le phénomène observé aussi pertinente qu'en utilisant l'ensemble des vecteurs de mesure  $d_k = \{d_k^s\}_{s=1}^S$ .

2.2.3. prise de décision

Cette dernière étape du processus de fusion concerne le choix d'une des hypothèses  $H_k$  de  $\Omega$  qui soit la plus précise et la plus robuste possible (compromis précision-confiance). Partant du principe que la minimisation de l'entropie conditionnelle équivaut à une minimisation des erreurs de classement, la règle de décision entropique s'écrit :

$$\tilde{H}_k^* = \arg \min_k \left\{ H(H_k/\tilde{d}_k^*) \right\} \quad (18)$$

Cette règle permet donc de caractériser la pertinence de la décision  $u = (\tilde{H}_k^*, \tilde{d}_k^*)$ , sachant que les informations disponibles sont fiables.

Pour lever toute ambiguïté sur cette décision, nous définissons une mesure de la qualité de la décision par un indice informationnel de qualité qui s'écrit comme suit :

$$q(H_k/\tilde{d}_k^*) = \frac{I(\tilde{d}_k^*, H_k)}{\sum_{\kappa=1}^K I(\tilde{d}_k^*, H_\kappa)} \quad (19)$$

$$= \frac{H(H_k) - H(H_k/\tilde{d}_k^*)}{\sum_{\kappa=1}^K H(H_\kappa) - H(H_\kappa/\tilde{d}_k^*)} \quad (20)$$

où le terme  $\sum_{\kappa=1}^K H(H_\kappa) - H(H_\kappa/\tilde{d}_k^*)$  est interprété comme une mesure de conflit entre l'hypothèse  $H_k$  et le sous-ensemble  $\tilde{d}_k^*$ . L'indice  $q(H_k/\tilde{d}_k^*)$  prend des valeurs comprises entre 0 (resp. pour  $H(H_k/\tilde{d}_k^*) = H(H_k)$ , il y a pas de possibilité d'améliorer la décision  $u = (\tilde{H}_k^*, \tilde{d}_k^*)$ ) et 1 (resp. pour

$\sum_{\kappa \neq k}^K H(H_\kappa) - H(H_\kappa/\tilde{d}_k^*) = 0$ , il y a plus de chance d'avoir  $u = (\tilde{H}_k^*, \tilde{d}_k^*)$ ). Cet indice de qualité indique donc la possibilité d'améliorer la qualité de la décision en mettant en cause tout le processus de fusion.

En résumé, la règle de décision entropique que nous retenons est la suivante :

$$u = (\tilde{H}_k^*, \tilde{d}_k^*) \quad \text{Ssi} \quad \begin{cases} H(\tilde{H}_k^*/\tilde{d}_k^*) = \min_k \{ H(H_k/\tilde{d}_k^*) \} \\ q(\tilde{H}_k^*/\tilde{d}_k^*) = \max_k \{ q(H_k/\tilde{d}_k^*) \} \end{cases} \quad (21)$$

### 3. l'algorithme de fusion entropique développé

Pour mettre en œuvre cette méthodologie de fusion par l'approche entropique, nous avons développé l'algorithme A2E (Agrégation Adaptative Entropique). C'est un algorithme itératif qui utilise des démarches heuristiques dont le fonctionnement est donné ci-après.

#### 3.1. présentation

La figure 2 montre le synoptique général de l'algorithme A2E. Le fonctionnement de l'algorithme A2E se déroule en trois phases [Fass02]. La phase d'agrégation est une étape d'initiali-

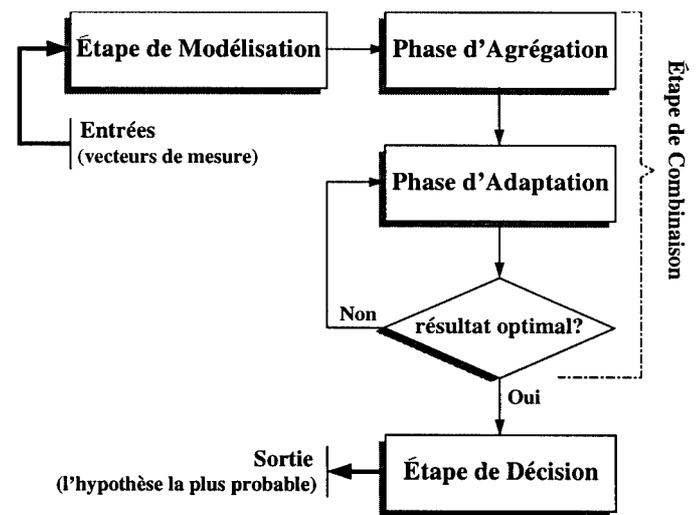


Figure 2. – Synoptique de l'algorithme A2E.

sation qui permet une première approche de la solution recherchée dans l'étape de la combinaison des sources d'information. Les paramètres d'entrée de cette phase, qui sont les sorties de l'étape de modélisation, sont connus ou ont été préalablement estimés au moyen d'algorithmes d'estimation de lois de probabilités [Fass99] [Fass02]. La phase d'*adaptation* permet de faire converger le sous-ensemble des mesures agrégé issue de la phase d'agrégation vers le « bon » minimum global. C'est un processus d'approximation successive, qui teste l'optimalité d'une solution possible en composant deux types d'heuristiques : l'*heuristique d'agrégation* et l'*heuristique de désagrégation*. Ces heuristiques sont inspirées des algorithmes de classification développés dans le contexte des arbres de décision [Quin83], [Quin86], [Perche95]. La phase de *décision* permet de faire le choix de l'hypothèse la plus probable au sens de l'entropie parmi l'ensemble des hypothèses  $H_k$  du cadre de discernement  $\Omega$  définissant le problème traité.

### 3.2. heuristiques d'agrégation et de désagrégation

L'approche agrégative est la démarche la plus naturelle pour effectuer la combinaison des sources d'information. En effet, intuitivement on commencera par rechercher le vecteur de mesure  $d_k^s$ , fourni par la source d'information  $x_s$ , relativement à une hypothèse  $H_k$ , qui contient l'information la plus pertinente sur le problème traité. Puis on combine ce vecteur de mesure avec un autre, qui augmente le plus la connaissance du système de fusion, tout en tenant compte du résultat précédent. Et on itère l'opération jusqu'à satisfaction du compromis redondance-complémentarité, c.-à-d., jusqu'à ce qu'il n'y ait plus de vecteur de mesure pertinent. Il s'agit donc ici d'agréger itérativement les vecteurs de mesure  $\{d_k^s\}_{s=1}^S$ , fournis par chacune des sources d'information  $x_s$ , contenant de l'information relativement à l'hypothèse  $H_k$ , de façon à déterminer le sous-ensemble des mesures  $d_k^{\text{HAG}}$  satisfaisant la règle de combinaison agrégative entropique suivante :

$$\forall H_k \in \Omega, \quad \forall s \in [1, \dots, S],$$

$$d_k^{\text{HAG}} = \arg \min_{d_k^s \notin d_k^{(\rho-1)}} \left\{ H \left( H_k / \left( d_k^{(\rho-1)}, d_k^s \right) \right) \right\} \quad (22)$$

où  $\rho$  est l'indice d'itération de l'agrégation ( $\rho \leq S$ ) et HAG désigne l'opération d'agrégation. Le symbole  $\setminus$  est une opération qui exprime le retrait d'un (des) élément(s) dans un ensemble. On remarquera que  $d_k^{(0)} = \emptyset$  (à l'initialisation  $\rho = 0$ ) et  $d_k^{(\rho)} = \left( d_k^{(\rho-1)}, d_k^s \right)$  (à l'itération  $\rho$ ). Le terme  $d_k^{(\rho)}$  désigne l'ensemble des vecteurs de mesure agrégés de cardinal  $\rho$ .

La convergence de cette heuristique est assurée puisque d'une part, l'entropie de Shannon est une fonction convexe de sa probabilité  $P$ , et d'autre part, l'entropie conditionnelle est une fonction concave de sa probabilité d'après la propriété de sous-additivité ( $0 \leq H(H_k/d_k^s) \leq H(H_k)$ ). Cependant, il peut arriver qu'une (ou plusieurs) source(s) d'information combinée(s) avec d'autres apporte(nt) plus d'information au sens de l'entropie conditionnelle sur le problème à traiter, alors que cette (ou ces) même(s) source(s) d'information prise(s) séparément n'apporte(nt) en réalité aucune information (leur entropie conditionnelle est grande). Dans ce cas, l'heuristique d'agrégation combinera à tort le ou les vecteur(s) de mesure de cette ou ces source(s) d'information qui ne permettra(ont) pas de minimiser le critère entropique.

L'heuristique de désagrégation pallie cette difficulté, car dès la première itération, elle prend en compte l'ensemble des mesures  $d_k^{(S)}$ , contenant tous les vecteurs de mesure  $\{d_k^s\}_{s=1}^S$  relativement à l'hypothèse  $H_k$ , pour en écarter un à un le vecteur de mesure apportant le plus d'information au système de fusion jusqu'à ce que l'on obtienne le sous-ensemble des mesures « optimal ».

La règle de combinaison désagrégative entropique associée s'écrit alors :

$$\forall H_k \in \Omega, \quad \forall s \in [1, \dots, S]$$

$$d_k^{\text{HDG}} = \arg \min_{d_k^s \in d_k^{(\rho+1)}} \left\{ H \left( H_k / \left( d_k^{(\rho+1)} \setminus d_k^s \right) \right) \right\} \quad (23)$$

avec  $d_k^{(0)} = d_k^{(S)}$  (à l'initialisation de l'itération) et  $d_k^{(\rho)} = \left( d_k^{(\rho+1)} \setminus d_k^s \right)$  (à l'itération  $\rho$ ). Le sigle HDG désigne l'opération de désagrégation.

Là encore, cette démarche ne s'effectue pas sans inconvénients. En effet, elle est d'une part, plus coûteuse en termes de calculs puisqu'elle nécessite de connaître, dès la première itération, toutes les entropies conditionnelles et en particulier le calcul exhaustif de l'entropie conditionnelle  $H(H/d_k^{(S)})$  est impossible lorsque le nombre  $S$  des sources d'information est grand. Et d'autre part, l'inverse du problème présenté dans la démarche agrégative pourrait se poser, à savoir : l'heuristique de désagrégation peut retirer de la combinaison des sources d'information qui sont complémentaires et ne conserver que les sources d'information qui sont redondantes ou peu informatives.

### 3.3. description des phases de l'algorithme

Dans cette section nous allons décrire les deux premières phases de l'algorithme A2E : la phase d'agrégation et la phase d'adaptation. La phase de décision repose sur la règle de décision

entropique dont l'expression est donnée par l'équation 21. Elle se résume à faire la recherche de l'hypothèse  $H_k$  la plus probable (au sens de l'entropie) à l'intérieur du cadre de discernement  $\Omega$ .

### 3.3.1. phase d'agrégation

La phase d'agrégation utilise l'heuristique d'agrégation pour approcher par itérations successives le sous-ensemble des mesures à optimiser. Ce sous-ensemble des mesures intermédiaire constituera le sous-ensemble des mesures initial pour la phase d'adaptation. L'arrêt de la phase d'agrégation peut se faire de trois façons : en bornant le nombre d'itérations à exécuter (le nombre  $p$  de vecteurs de mesure est alors connu d'avance), ou en fixant la valeur minimale de l'entropie conditionnelle  $H_{\min}$  ou en attendant que le critère d'erreur passe en deçà d'un seuil fixé.

Fixer le nombre d'itérations  $p$  pour effectuer la phase d'agrégation revient à donner la dimension du sous-ensemble des vecteurs de mesure agrégés. Sachant que la connaissance de  $p$  ou de  $H_{\min}$  n'est pas évidente *a priori*, l'utilisation de la troisième condition d'arrêt est donc plus appropriée. Elle s'écrit ainsi :

$$\left| H\left(H_k/d_k^{(p+1)}\right) - H\left(H_k/d_k^{(p)}\right) \right| < \text{SEUIL} \quad (24)$$

où  $d_k^{(p+1)}$  et  $d_k^{(p)}$  sont des sous-ensembles des mesures de dimension respectivement  $(p+1)$  et  $p$ . Le paramètre SEUIL représente la quantité d'information à prendre en compte pour passer d'une agrégation de  $p$  vecteurs de mesure à  $(p+1)$  vecteurs de mesure. Lorsque le critère d'erreur passe au-dessus de ce seuil, le  $(p+1)^{\text{ème}}$  vecteur de mesure ne sera pas pris en compte dans le processus de combinaison puisqu'il est considéré comme apportant peu d'information au système de fusion. La dimension du sous-ensemble des mesures résultant sera alors fixée à  $p$  vecteurs de mesure (où  $p \leq S$ ).

Dans le but de garantir l'optimalité du sous-ensemble des mesures et pour profiter des avantages des deux heuristiques d'agrégation et de désagrégation, nous proposons de tester l'optimalité de la combinaison agrégative suivant trois niveaux :  $(p-1)$ ,  $p$  et  $(p+1)$ , où  $p$  désigne le nombre de vecteurs de mesure agrégés. Pour cela, on lui applique successivement l'une et l'autre des deux heuristiques et vice-versa jusqu'à ce qu'il y ait une stabilisation, c.-à-d. les différentes compositions de ces deux heuristiques donnent le même résultat. Nous décrivons cette procédure ci-après.

### 3.3.2. phase d'adaptation

Soient HAG et HDG les opérations d'agrégation et de désagrégation. Soient  $\text{HAG} \circ \text{HDG}$  et  $\text{HDG} \circ \text{HAG}$  les fonctions de com-

position des opérations d'agrégation et de désagrégation. La condition d'arrêt de la phase d'adaptation est donnée par la propriété suivante :

$$\begin{aligned} \forall H_k \in \Omega, \quad & H\left(H_k/d_k^{\text{A2E}}\right) \\ &= H\left(H_k/\text{HAG} \circ \text{HDG}\left(d_k^{\text{A2E}}\right)\right) \\ &= H\left(H_k/\text{HDG} \circ \text{HAG}\left(d_k^{\text{A2E}}\right)\right) \end{aligned} \quad (25)$$

où  $d_k^{\text{A2E}}$  désigne le sous-ensemble des mesures optimal ou le résultat de la combinaison.

Cette propriété est une conséquence du *caractère contractant* [Toro82] de la composition des deux heuristiques de combinaison HAG et HDG. Elle indique que le sous-ensemble des mesures optimal  $d_k^{\text{A2E}}$  est aussi un sous-ensemble stable puisqu'il constitue un *point fixe* à la fois pour les fonctions  $\text{HAG} \circ \text{HDG}$  et  $\text{HDG} \circ \text{HAG}$ .

Finalement, nous proposons la règle de combinaison suivante, que nous appelons règle d'agrégation adaptative entropique :

$$\forall H_k \in \Omega, \quad d_k^{\text{A2E}} = \arg \min_{d_k^{(p)} \text{ où } p \text{ est fixé}} \left\{ H\left(H_k/\text{HAG} \circ \text{HDG}\left(d_k^{(p)}\right)\right) \odot H\left(H_k/\text{HDG} \circ \text{HAG}\left(d_k^{(p)}\right)\right) \right\} \quad (26)$$

où  $d_k^{(p)}$  est initialement donnée par l'équation 22. Le symbole  $\odot$  représente une opération de type « *divided and aggregated* », c.-à-d. le processus d'approximation successive.

## 4. application à la robotique

À titre d'exemple, nous avons appliqué le modèle de fusion entropique à une application de la robotique mobile adaptée de [Lars98]. L'application est un robot mobile navigant dans un milieu dont la perception de celui-ci, au travers des objets qui l'entourent, n'est pas forcément correcte. Afin d'améliorer la connaissance sur l'objet à identifier, le robot mobile devra combiner ces informations, aussi bien au point de vue quantitatif, c.-à-d. plus d'informations, par l'utilisation de la complémentarité des sources d'information, qu'au point de vue qualitatif, c.-à-d. donner une information plus fiable, en exploitant la redondance des sources d'information [Luo89].

Nous montrons à travers cette application que la méthodologie de fusion entropique est intéressante dans le problème de fusion d'informations imparfaites et permet d'améliorer la fiabilité de

la prise de décision en exploitant simultanément des informations complémentaires et redondantes lors de l'étape de combinaison, tout en réduisant la dimension de l'espace de combinaison [Fass01a, Fass01b].

### 4.1. description de l'application

Le problème consiste à faire de la discrimination entre plusieurs hypothèses représentatives de l'identité d'un objet dans un contexte de perception multi-sources. Le cadre de discernement  $\Omega$  est composé de 5 hypothèses  $H_k$  :

- $H_1$  : l'objet est une porte fermée
- $H_2$  : l'objet est une porte ouverte
- $H_3$  : l'objet est une fenêtre
- $H_4$  : l'objet est une personne
- $H_5$  : l'objet est inconnu

à chaque hypothèse  $H_k$  est associé un ensemble de 6 paramètres  $z_m$  :

- $z_1$  : de forme carrée
- $z_2$  : encastré dans un mur
- $z_3$  : avec une profondeur (ouverture)
- $z_4$  : avec une certaine dimension
- $z_5$  : au ras du sol
- $z_6$  : inanimé

dont les valeurs particulières permettent d'identifier le type de l'objet.

Pour cela, le robot est équipé d'un ensemble de 3 sources d'information : un capteur IR (*Infra Rouge*) et 2 caméras CCD (*Charge Coupled Device*) dont l'une est supposée apporter peu d'information au système de fusion dans la suite du problème. Les sources d'information ne communiquent pas entre elles. Chacune d'elle est reliée directement au système de fusion.

### 4.2. données de l'application

Le robot mobile est capable d'identifier 5 objets, du milieu où il évolue, à partir de leurs différentes caractéristiques, au nombre de 6. Un objet inconnu de type  $H_k$  doit être identifié (cf. figure 3). Le robot mobile doit être capable de déterminer en particulier si l'objet devant lui est une porte fermée ou ouverte. L'objet génère donc des valeurs spécifiques pour chacun des 6 paramètres  $z_m$  de  $z$ . Ces paramètres sont observés (imparfaitement) par les 3 sources d'information qui équipent le robot mobile.

Le tableau 2 montre les données observées et fournies par les différentes sources d'information. Ces données sont modélisées par des distributions de probabilité conditionnelle. Il faut noter que ces distributions de probabilité ont été estimées et non

trouvées expérimentalement. De plus, elles ne sont pas normalisées :  $\sum_{m=1}^6 P(z_m^k/d_k^s) \neq 1$ .

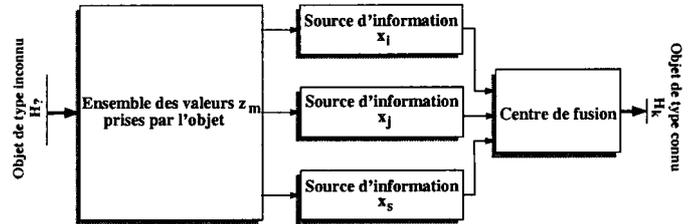


Figure 3. – Système de fusion multi-sources.

Tableau 2. – Données des observations des sources d'information.

Vecteurs de mesure						
$d_k^1$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$
$d_1^1$	0.95	0.95	0.10	0.95	0.90	0.10
$d_2^1$	0.95	0.70	0.98	0.95	0.90	0.10
$d_3^1$	0.95	0.60	0.30	0.10	0.10	0.10
$d_4^1$	0.20	0.70	0.20	0.10	0.90	0.80
$d_5^1$	0.30	0.50	0.20	0.10	0.90	0.20

caméra CCD N° 1

Vecteurs de mesure						
$d_k^2$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$
$d_1^2$	0.98	0.95	0.10	1.00	0.85	0.08
$d_2^2$	0.98	0.75	1.00	0.92	0.95	0.12
$d_3^2$	0.89	0.60	0.30	0.05	0.10	0.10
$d_4^2$	0.20	0.65	0.25	0.05	0.88	0.80
$d_5^2$	0.30	0.50	0.15	0.20	0.92	0.20

IR sensor

Vecteurs de mesure						
$d_k^3$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$
$d_1^3$	0.50	0.50	0.50	0.50	0.50	0.50
$d_2^3$	0.50	0.50	0.50	0.50	0.50	0.50
$d_3^3$	0.50	0.50	0.50	0.50	0.50	0.50
$d_4^3$	0.50	0.50	0.50	0.50	0.50	0.50
$d_5^3$	0.50	0.50	0.50	0.50	0.50	0.50

caméra CCD N° 2

L'utilisation de l'approche entropique nécessite de connaître, pour chaque hypothèse  $H_k$ , les fréquences d'occurrence,  $P(z_m^k)$ , des paramètres  $z_m$  pour évaluer les mesures de qualité de la décision (cf. tableau 3). Le système de fusion doit disposer également des connaissances sur les distributions de probabilité *a priori*,  $P(d_k^s)$ , des observations des sources d'information. Ces distributions de probabilité *a priori* sont supposées « **non-informatives** », c.-à-d. les distributions de probabilité sont uniformes. On suppose donc que toutes les 3 sources d'information ont la même fiabilité (cf. tableau 4).

Tableau 3. – Les probabilités d'occurrences des paramètres sont connues. Notons que  $\sum_m P(z_m^k) = 1$ .

Distribution de probabilités <i>a priori</i>						
$P(z_m^k)$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$
$H_1$	0.24	0.20	0.06	0.24	0.21	0.05
$H_2$	0.20	0.16	0.22	0.19	0.21	0.02
$H_3$	0.44	0.29	0.15	0.02	0.05	0.05
$H_4$	0.05	0.10	0.09	0.02	0.50	0.24
$H_5$	0.08	0.10	0.07	0.09	0.60	0.06

Tableau 4. – Chaque source d'information est initialisée avec une information *a priori* équiprobable (elles sont supposées toutes fiables). Notons que  $\sum_k P(d_k^s) = 1$ .

Connaissances <i>a priori</i>			
$P(d_k^s)$	$d_k^1$	$d_k^2$	$d_k^3$
$H_1$	0.20	0.20	0.20
$H_2$	0.20	0.20	0.20
$H_3$	0.20	0.20	0.20
$H_4$	0.20	0.20	0.20
$H_5$	0.20	0.20	0.20

### 4.3. résultats de l'application

Sur la figure 4, on peut visualiser la convergence de l'algorithme A2E. Parmi les 3 sources d'information, l'une d'elles n'apporte qu'une information redondante ou pas d'information du tout. Il n'est par conséquent pas utile de la prendre en compte dans le processus de combinaison. Effectivement, on peut voir sur les différentes courbes que l'entropie conditionnelle diminue jusqu'à 2 sources d'information et augmente à nouveau pour 3 sources d'information.

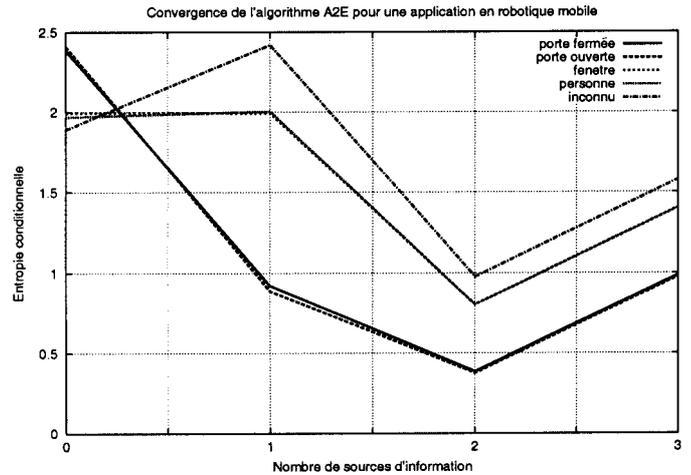


Figure 4. – Convergence du critère de combinaison entropique en fonction du nombre de sources d'information.

On observe également une réduction de l'espace de combinaison comme illustré au tableau 5. Ces résultats sont tout à fait conformes à ce que l'on pouvait s'attendre, compte tenu du fait d'avoir introduit explicitement une source d'information redondante ou peu informative. Ainsi, s'il fallait donner l'identité de l'objet se trouvant dans le champ d'observation du robot mobile, on dira que l'objet est une porte ouverte (hypothèse  $H_2$ ). On obtient le sous-ensemble des mesures suivant, composé dans l'ordre de combinaison des sources d'information, {capteur IR, caméra CCD  $N^o1$ } avec une entropie conditionnelle de 0,377 et un indice de qualité de 0,278.

On remarquera que la combinaison de la totalité des informations fournies par les 3 sources d'information aurait également privilégié l'objet porte ouverte. On voit bien là l'intérêt de notre approche entropique. La réduction de l'espace de combinaison a été aussi vérifiée pour les autres hypothèses, compte tenu du choix des valeurs des distributions de probabilité de la source d'information mise en défaut.

Tableau 5. – Entropies conditionnelles et indices de qualité. Notons que  $\sum_k q(H_k/d_k^{A2E}) = 1$ .

	Objets observés				
	$H_1$	$H_2$	$H_3$	$H_4$	$H_5$
$H(H_k/d_k^{(3)})$	0.987	0.977	1.405	1.404	1.578
$H(H_k/d_k^{A2E})$	0.387	0.377	0.805	0.804	0.978
$q(H_k/d_k^{A2E})$	0.274	0.278	0.163	0.160	0.125
$p$	2	2	2	2	2

## 5. environnements de programmation parallèle ou distribuée

La programmation parallèle consiste à découper le traitement global d'une application en tâches indépendantes qui se synchronisent par divers mécanismes afin de collaborer à la réalisation d'une tâche plus importante en taille et en durée de calcul que ne le permettrait une programmation séquentielle. La *parallélisation* est liée au *problème d'optimisation d'allocation de ressources*, où les ressources sont les séquenceurs d'instructions et les médias de communication inter-séquenceurs, et où les tâches à allouer à ces ressources sont les opérations de l'algorithme ainsi que celles de synchronisation et de communication [Grand98]. La solution peut être envisagée de deux façons : soit par allocation statique de ressources en utilisant, par exemple, l'environnement SynDEx ; soit par allocation dynamique de ressources en utilisant, par exemple, l'environnement PVM.

### 5.1. l'environnement SynDEx

SynDEx, acronyme de *SYN*chronized *D*istributed *EX*ecutive, est un environnement graphique interactif de développement, supportant la méthodologie AA-A, pour la programmation parallèle d'applications de traitement du signal et des images devant garantir des contraintes temps-réel et d'embarquabilité [Sorel96], [Aiglon96]. Il est conçu, développé et distribué (gratuitement<sup>6</sup>) par l'INRIA (*Institut National de Recherche en Informatique et en Automatique*) de Rocquencourt, en France.

L'environnement SynDEx permet à partir de la spécification de l'algorithme mise sous la forme d'un *graphe flot de données* (graphe d'algorithme) décrivant l'application, et à partir de la spécification de l'architecture cible représentée par un *graphe matériel* (graphe d'architecture), de réaliser la phase de placement (utilisation du *parallélisme disponible*<sup>7</sup> de l'architecture) et d'ordonnement (réduction du *parallélisme potentiel*<sup>8</sup> de l'algorithme) grâce à une *heuristique d'adéquation* [Vic99] (cf. figure 5).

Après exécution de l'heuristique, SynDEx produit un *diagramme ou graphe temporel* représentant la répartition (distribution) des opérations sur les différents processeurs spécifiés (opérateurs), l'ordonnement de ces opérations et les communications inter-processeurs. Ce point est important car il permet de

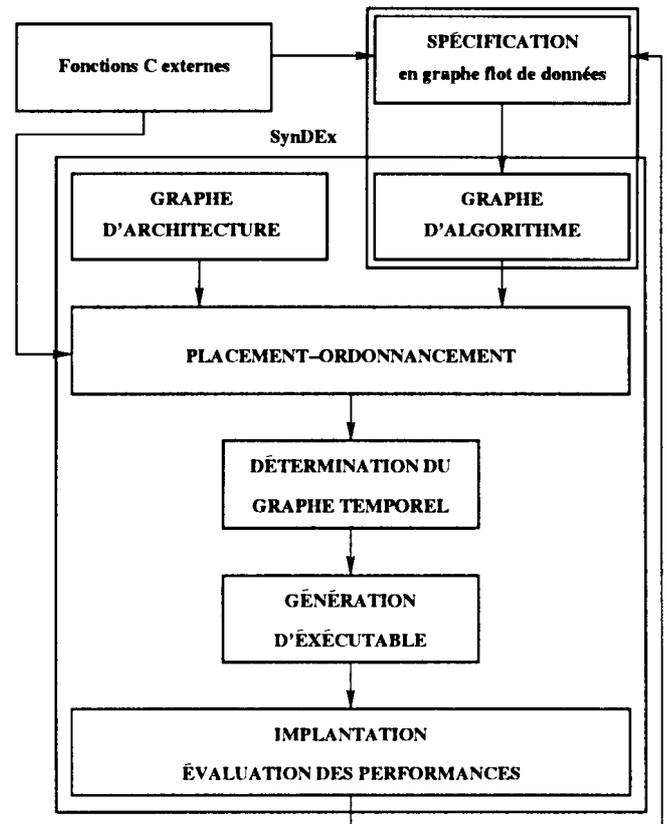


Figure 5. – Résumé de l'approche SynDEx appelant des fonctions C externes.

faire l'évaluation des performances de l'application sans réaliser réellement son implantation. Pour cela, lors d'un premier passage sous SynDEx, il suffit d'effectuer une exécution séquentielle du graphe flot de données de l'algorithme sur une machine mono-processeur après compilation avec les instructions de chronométrage. De cette exécution sont ainsi déterminés les durées de traitement de chaque sommet du graphe de l'algorithme. Ces estimations temporelles<sup>9</sup> des tâches et des communications sont ensuite spécifiées au niveau du graphe de l'algorithme, lors du dernier passage sous SynDEx, pour obtenir une répartition optimisée de l'application (cf. figure 6).

Enfin, SynDEx génère automatiquement un exécutif statique, distribué, optimisé, synchronisé, implantable sur une plateforme matérielle quelconque. Ceci permet d'une part, de décharger l'utilisateur des tâches fastidieuses de programmation bas niveau (écriture du code système) et de la phase de mise au point de l'algorithme sur l'architecture cible, et d'autre part, d'autoriser du même coup une meilleure concentration sur les problèmes

<sup>6</sup> Site officiel SynDEx : [www-rocq.inria.fr/syndex](http://www-rocq.inria.fr/syndex)

<sup>7</sup> Le parallélisme disponible de l'architecture représente les différentes ressources matérielles disponibles.

<sup>8</sup> Certaines opérations du graphe d'algorithme qui ne sont pas en dépendance mutuelle de données sont potentiellement exécutables en parallèle.

<sup>9</sup> Le chronométrage est effectué à l'aide de l'horloge interne du processeur.

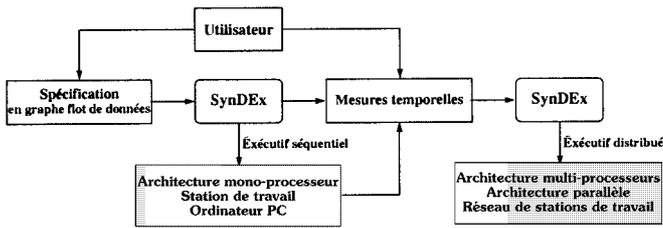


Figure 6. – Principe du chronométrage sous SynDEX.

directement liés au programme applicatif. Ainsi, l'utilisateur n'a pas d'autre code à écrire que celui de ses opérations de calcul et ceux d'entrée et de sortie. Tout le reste (ordonnancement et appel des opérations de calcul, d'entrée et de sortie, allocation du tampon mémoire nécessaire aux opérations de communication inter-opérations, intra-processeur ou inter-processeurs et code des opérations de communication) est généré automatiquement par SynDEX à partir, des graphes d'algorithme et d'architecture spécifiés par l'utilisateur, des résultats de placement-ordonnancement et du noyau générique de l'exécutif SynDEX.

## 5.2. l'environnement PVM

PVM, acronyme de *Parallel Virtual Machine*, est un standard de programmation parallèle orienté vers la programmation d'applications parallèles généralistes écrites en langages C/C++ ou Fortran sans contraintes temps-réel. Développé par des chercheurs de l'Université du Tennessee et du ORNL (*Oak Ridge National Laboratory*) [Geist94], PVM est aussi accessible gratuitement à partir du WEB<sup>10</sup>.

PVM est à l'origine une *bibliothèque de routines et de primitives de communication par passage de messages*, destinée à contrôler l'exécution du programme sur une *machine virtuelle*<sup>11</sup> et la configuration de celle-ci. L'intérêt réside dans le fait que la machine résultante est extensible (par adjonction de machines physiques), modifiable (par remplacement standard des composants périmés), stable (car elle s'appuie sur une technologie de masse qui est au point) et peut comporter des machines spécialisées (stations de travail, compatibles PC, processeurs vectoriels).

La machine virtuelle est constituée d'un certain nombre de *nœuds*, où les nœuds peuvent être des machines mono-proces-

seurs, des machines multi-processeurs ou des machines parallèles complètes, permettant ainsi à l'application parallèle de s'exécuter sur l'architecture qui est la plus adaptée. Elle est associée à l'environnement d'exécution, appelé démon (ou daemon) pvmd, qui collecte les messages et qui les transmet du processus émetteur vers le processus destinataire. Le démon pvmd doit être présent obligatoirement sur chaque machine ou nœud susceptible de faire partie de la machine virtuelle (cf. figure 7). Il sert de routeur pour les messages et de contrôleur dans la communication entre les différentes tâches qui coopèrent, en étant globalement connu par les tâches de la machine virtuelle.

Les communications entre les différentes tâches PVM se font par échange de messages. En général, les tâches ne communiquent pas directement entre elles. Elles passent par le démon pvmd local (par l'intermédiaire de sockets<sup>12</sup> Unix avec en en-tête le tid<sup>13</sup> de la tâche distante), qui à son tour va contacter le démon pvmd du nœud distant (par l'intermédiaire du réseau de communication et des protocoles de communication UDP/TCP-IP<sup>14</sup>), qui lui-même contactera la tâche destinataire. Lorsque les deux tâches qui coopèrent, se trouvent dans le même nœud, le message passe par le démon pvmd local qui le stocke jusqu'à ce que la tâche destinataire le lise par un appel à la routine de réception (cf. figure 7).

L'utilisation de PVM se fait en mode commande (console pvm) ou par l'entremise d'une interface graphique, XPVM (version X-Window de PVM), qui permet d'effectuer la prise et l'analyse des traces de programmes PVM.

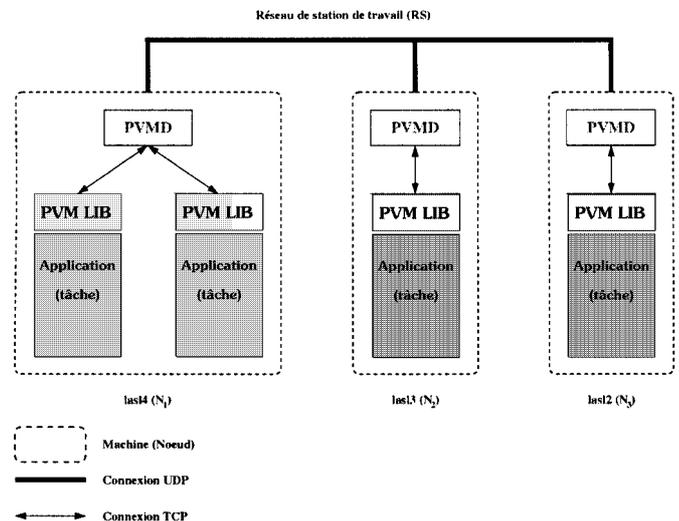


Figure 7. – Mécanisme de communication de PVM.

<sup>10</sup> Site officiel PVM : [www.epm.ornl.gov/pvm/pvm\\_home.html](http://www.epm.ornl.gov/pvm/pvm_home.html)

<sup>11</sup> Le principe est remarquable et se limite à faire croire à l'application qu'elle s'exécute sur une machine multi-processeurs à mémoire distribuée. En fait, cette machine n'existe pas, ce qui explique que l'on emploie l'adjectif « virtuelle » pour la qualifier. Il s'agit juste de l'amalgame de plusieurs ordinateurs connectés entre eux, fonctionnant de concert grâce à PVM.

<sup>12</sup> Les sockets sont des points de connexion qui permettent des communications au sein du même système ou vers l'extérieur.

<sup>13</sup> Pour *task identifier* ou Task ID par analogie au PID attribué par le système Unix à tout processus en exécution.

<sup>14</sup> *User Datagram Protocol/Transmission Control Protocol-Internet Protocol*

## 6. étude de la parallélisation de l'algorithme A2E

L'algorithme A2E que nous avons développé est un processus itératif par convergence qui combine une à une les informations les plus pertinentes, appartenant à l'ensemble des informations disponibles, jusqu'à satisfaction du critère de convergence. Le fonctionnement de cet algorithme se déroule en trois phases :

- une phase d'agrégation permettant d'obtenir un ensemble de sources d'information agrégées constituant l'ensemble initial de la phase d'adaptation ;
- une phase d'adaptation qui effectue l'optimisation de l'ensemble des sources d'information agrégées. Elle a pour but de vérifier si aucune information pertinente n'a été omise et/ou si une information non pertinente a été considérée ;
- une phase de décision permettant de choisir l'ensemble des sources d'information agrégées le plus optimal parmi plusieurs ensembles solutions.

### 6.1. algorithme séquentiel

La version séquentielle de l'algorithme A2E est décrite de la façon suivante :

- ① Initialisation des paramètres  
SEUIL paramètre à fixer

- ② Phase d'agrégation

**Pour**  $k = 1$  à  $K$  **Faire**

**Pour**  $s = 1$  à  $S$  **Faire**

calculer les  $H(H_k/d_k^s)$

**Fin Pour**

Déterminer  $d_k^s$  avec  $H(H_k/d_k^s)$  minimale

$H_{\min} = \min(H(H_k/d_k^s))$

$d_k^{(1)} = d_k^s$

**Répéter**

OLD =  $H_{\min}$

**Pour**  $s = 1$  à  $S$  **Faire**

calculer les  $H(H_k/(d_k^{(\rho-1)}, d_k^s))$  avec

$d_k^s \notin d_k^{(\rho-1)}$

**Fin Pour**

Déterminer  $d_k^s$  avec  $H(H_k/(d_k^{(\rho-1)}, d_k^s))$  minimale

$$H_{\min} = \min(H(H_k/(d_k^{(\rho-1)}, d_k^s)))$$

$$d_k^{(\rho)} = (d_k^{(\rho-1)}, d_k^s)$$

**Jusqu'à**  $|H_{\min} - \text{OLD}| < \text{SEUIL}$  ou  $\rho \leq S$

**Fin Pour**

- ③ Phase d'adaptation

**Pour**  $k = 1$  à  $K$  **Faire**

STABLE = FAUX

**Tant que**  $\neg \text{STABLE}$  **Faire**

**Tant que**  $\text{HAG} \circ \text{HDG}(d_k^{(\rho)}) \neq d_k^{(\rho)}$  **Faire**

$$d_k^{(\rho)} = \text{HAG} \circ \text{HDG}(d_k^{(\rho)})$$

**Fin Tant que**

**Tant que**  $\text{HDG} \circ \text{HAG}(d_k^{(\rho)}) \neq d_k^{(\rho)}$  **Faire**

$$d_k^{(\rho)} = \text{HDG} \circ \text{HAG}(d_k^{(\rho)})$$

**Fin Tant que**

**Si**  $\text{HAG} \circ \text{HDG}(d_k^{(\rho)}) = \text{HDG} \circ \text{HAG}(d_k^{(\rho)}) =$

$d_k^{(\rho)}$  **Alors**

STABLE = VRAI

**Fin Si**

**Fin Tant que**

**Fin Pour**

- ④ Phase de décision

retenir  $H_k$  dont  $H(H_k/d_k^{(p)})$  est la plus petite

et  $q(H_k/d_k^{(p)})$  est le plus grand.

La convergence de l'algorithme A2E dépend fortement du bon choix du paramètre SEUIL, et faiblement de la condition nécessaire d'optimalité. En effet, plus l'entropie conditionnelle du sous-ensemble des mesures initial (résultat de la phase d'agrégation) sera proche de la valeur minimale, plus il y aura de « chance » de trouver le sous-ensemble des mesures stable.

### 6.2. durée d'exécution de l'algorithme en séquentiel

Nous distinguons trois parties dans l'algorithme A2E, à savoir : la phase d'agrégation, la phase d'adaptation et la phase de décision.

La durée nécessaire à l'initialisation de l'algorithme dépend fortement du temps passé dans la boucle principale de la phase

d'agrégation. Le temps passé dans la phase d'adaptation correspond à celui passé pour exécuter la boucle de test de stabilité. La durée d'exécution de la phase de décision est égale à celle de  $K$  instructions de comparaison effectuées sur les valeurs des entropies conditionnelles des  $K$  sous-ensembles des mesures stables entre elles et sur leurs indices de qualité entre eux.

Les calculs effectués dans les boucles principales (boucle principale de la phase d'agrégation et boucle de test de stabilité) reviennent à examiner consécutivement un certain nombre de « solutions candidates », en vue de trouver la plus favorable. Ce nombre, appelé *ordre de la boucle principale*, est un paramètre important des boucles, puisque la durée d'exécution d'une boucle lui sera proportionnelle :

$$\left[ \begin{array}{l} \text{durée requise} \\ \text{pour examiner} \\ \text{une solution} \\ \text{candidate} \end{array} \right] \times \left[ \begin{array}{l} \text{ordre de la boucle} \\ \text{principale} \end{array} \right] \quad (27)$$

### 6.2.1. phase d'agrégation

Pour chacune des  $K$  hypothèses  $H_k$  de  $\Omega$ , on peut dire :

- déterminer le sous-ensemble des mesures  $d_k^{(1)}$  nécessite d'examiner  $(S - 1)$  alternatives ;
- $d_k^{(1)}$  étant connu, déterminer  $d_k^{(2)}$  nécessite d'examiner  $(S - 2)$  alternatives ;
- en généralisant,  $d_k^{(p-1)}$  étant connu, déterminer le sous-ensemble des mesures  $d_k^{(p)}$  nécessite d'examiner  $(S - p)$  alternatives ;

Ainsi, l'ordre de la boucle principale s'écrit :

$$\text{Ordre(agrégation)} = K \times \sum_{p=1}^{S-1} (S - p) \quad (28)$$

$$= K \times \frac{S(S - 1)}{2} \quad (29)$$

### 6.2.2. phase d'adaptation

La phase d'adaptation correspond à l'application des fonctions de composition  $\text{HAG} \circ \text{HDG}$  et/ou  $\text{HDG} \circ \text{HAG}$  autant de fois que cela est possible jusqu'à stabilisation. L'ordre de la phase d'adaptation dépend, d'une part, du nombre d'alternatives examinées lors de l'application de chacune des fonctions de composition.

**Fonction**  $\text{HAG} \circ \text{HDG}$  – le calcul de  $\text{HDG}(d_k^{(p)})$  nécessite d'examiner  $p$  alternatives. On obtient comme résultat un sous-ensemble de  $(p - 1)$  vecteurs de mesure. Puis, l'application de l'opération  $\text{HAG}$  à  $\text{HDG}(d_k^{(p)})$  nécessite quant à elle d'examiner

$(S - p)$  alternatives (les  $(S - 1)$  vecteurs de mesure restant à tester auxquels on retire les  $(p - 1)$  déjà testés). Ainsi, il faudrait examiner  $(p + (S - p) = S)$  alternatives lors de l'application de la fonction de composition  $\text{HAG} \circ \text{HDG}$ .

**Fonction**  $\text{HDG} \circ \text{HAG}$  – De même, il faudrait examiner  $((S - p) + p = S)$  alternatives lors de l'application de la fonction de composition  $\text{HDG} \circ \text{HAG}$  ( $(S - p)$  alternatives pour le calcul de  $\text{HAG}(d_k^{(p)})$ ,  $p$  pour l'application de l'opération  $\text{HDG}$  à  $\text{HAG}(d_k^{(p)})$ ).

D'autre part, si on suppose que les fonctions de composition  $\text{HAG} \circ \text{HDG}$  et  $\text{HDG} \circ \text{HAG}$  sont en moyenne appliquées  $\beta$  fois, c.-à-d.  $\frac{\beta}{2}$  applications de  $\text{HAG} \circ \text{HDG}$  et  $\frac{\beta}{2}$  de  $\text{HDG} \circ \text{HAG}$ , on obtient :

$$\text{Ordre(adaptation)} = \beta \times K \times \sum_{p=1}^{S-1} S \quad (30)$$

$$= \beta \times K \times S(S - 1) \quad (31)$$

### 6.2.3. bilan

En combinant les équations 29 et 31, on peut écrire :

$$\left[ \begin{array}{l} \text{durée exécution} \\ \text{algorithme A2E} \end{array} \right] = K \left( \frac{S(S - 1)}{2} \right) T_{\text{sol\_cand}}^{\text{agrégation}} + K \left( \beta S(S - 1) \right) T_{\text{sol\_cand}}^{\text{adaptation}} + K \times T_{\text{init\_agrég}} + T_{\text{décision}} \quad (32)$$

où  $T_{\text{sol\_cand}}^{\text{agrégation}}$  et  $T_{\text{sol\_cand}}^{\text{adaptation}}$  représentent les durées de traitement d'une solution candidate respectivement dans la phase d'agrégation et dans la phase d'adaptation. Le terme  $T_{\text{init\_agrég}}$  représente la durée de l'initialisation de la phase d'agrégation. Cette dernière inclut la durée des  $S$  calculs de l'entropie conditionnelle. Et, le terme  $T_{\text{décision}}$  représente la durée de la phase de décision.

## 6.3. étude du parallélisme potentiel

Dans cette section nous présentons l'étude de la parallélisation de l'algorithme A2E pour une implantation sur un réseau de stations de travail fonctionnant en mode MIMD-NORMA avec une communication par échange de messages.

Pour explorer au maximum la puissance de ce réseau de stations de travail, nous étudions une implantation de l'algorithme à gros grain. Les calculs de haut niveau, tels que le calcul des entropies conditionnelles  $H(H_k/d_k^s)$  et les boucles principales de l'algo-

rithme, seront considérés comme des opérations « atomiques ». La parallélisation des opérations élémentaires est plutôt intéressante pour des machines de type SIMD<sup>15</sup> pour lesquelles les communications se font en mémoire partagée. La parallélisation des boucles principales des phases d'agrégation et d'adaptation de l'algorithme A2E n'est pas adaptée au réseau de stations de travail. De plus, ces boucles principales sont des processus séquentiels et dépendants. Il faut donc faire un compromis entre le grain du parallélisme et le surcoût de la gestion du parallélisme induit. Seules les déterminations des  $K$  sous-ensembles des mesures optimaux relatifs aux hypothèses solutions possibles sont indépendantes les unes des autres, et peuvent donc être effectuées en parallèle. La répartition équilibrée de ces déterminations sur le réseau de stations de travail est la seule contrainte d'efficacité.

### 6.3.1. spécification de l'algorithme

Les performances de l'implantation de l'algorithme A2E dépendent donc de la distribution de la phase d'agrégation avec celle d'adaptation sur le réseau de stations de travail. Le niveau de granularité choisi ne fait apparaître que l'enchaînement des trois phases principales : agrégation, adaptation et décision. On exhibe ainsi, pour chacune des  $K$  hypothèses, uniquement le *parallélisme pipe-line* potentiel. La phase de décision nécessite des communications des processeurs esclaves vers le processeur maître afin de déterminer le sous-ensemble des mesures optimal et l'hypothèse « vainqueur ». Ainsi, la spécification de l'algorithme fera apparaître à la fois le parallélisme des tâches et le parallélisme pipe-line.

La représentation flot de données de notre algorithme présente une partie régulière, c.-à-d. des *motifs* ou sous-graphes répétitifs représentant des traitements identiques (sur des données différentes). La distribution de ces motifs sera faite de telle sorte que chaque station de travail du réseau exécute au moins une partie de l'itération. Par exemple, une spécification sous forme factorisée de  $K$  répétitions de motif de l'algorithme peut se décomposer en  $PE$  factorisations de  $\frac{K}{PE}$  répétitions du motif.

### 6.3.2. spécification de l'architecture

Le niveau de granularité choisi permet d'exploiter de façon optimale le parallélisme disponible d'un réseau de  $K$  stations de travail à partir du parallélisme potentiel de l'algorithme. En effet, avec un nombre  $PE$  de stations de travail, tel que  $K = (2PE + 1)$ , il n'y aura que  $PE$  motifs qui pourront être exécutés en parallèle, pendant qu'une station de travail traitera le dernier motif, les  $(PE - 1)$  autres seront inactives, d'où une chute importante de l'efficacité de l'implantation.

<sup>15</sup> Les machines de type SIMD (*Single Instruction Multiple Data stream*) sont des machines synchrones à mémoire distribuée.

Comme nous allons le constater lors de l'implantation de l'application de robotique, il existe une dissymétrie de répartition des motifs sur un réseau de  $(K - 1)$  stations de travail du graphe de l'algorithme optimum pour  $K$  stations de travail. Cette dissymétrie réduit les performances de l'implantation de l'algorithme A2E. Le choix de la distribution de  $\frac{K}{PE}$  tâches sur chaque station de travail est réellement optimal pour un nombre entier  $\frac{K}{PE}$ .

## 6.4. implantation de l'application

Une première implantation sur une unique station de travail du réseau s'avère nécessaire, afin de déterminer les durées de traitements des différentes tâches de l'application. Le point de départ est la description flot de données de l'application.

### 6.4.1. avec l'environnement SynDEx

Le graphe flot de données de l'application (sous SynDEx) est constitué de 5 motifs répétitifs, correspondant aux différents hypothèses  $H_k$  de  $\Omega$  que le système de fusion peut discriminer (cf. figure 8).

Voici les rôles de chacun des sommets du graphe :

- AcqDonnees, VISU1 et VISU2 sont des fonctions d'entrée-sortie, correspondant respectivement à l'acquisition des données de l'application dans des fichiers et d'écriture des résultats dans plusieurs fichiers. Elles sont contraintes à s'exécuter sur le processeur maître. La lecture et l'écriture des données dans les fichiers ont été optimisées de manière à concentrer les accès disques au début et à la fin de la boucle principale du programme ;
- Les sommets Fork permettent de diffuser les données d'acquisition vers les différents niveaux du graphe ;
- Les sommets PhaseAgregation effectuent chacun la phase d'agrégation de l'algorithme A2E ;
- Les sommets PhaseAdaptation effectuent chacun la phase d'adaptation de l'algorithme A2E à partir du résultat de la phase d'agrégation ;
- Le sommet PriseDecision effectue la recherche de la solution globale ;
- Le sommet CritereQualite calcule l'indice de qualité de chacune des solutions possibles ;
- Le sommet CalculEntGen donne les valeurs des entropies conditionnelles des ensembles des mesures  $d_k^{(S)}$  ;
- Le sommet JoinEnt constitue un flot vecteur des entropies conditionnelles des sous-ensembles des mesures  $d_k^{(P)}$ .

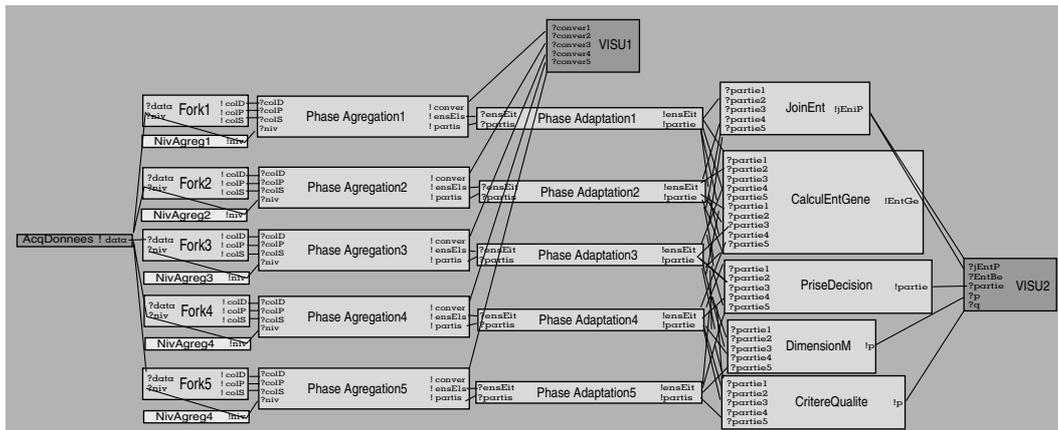


Figure 8. – Graphe flot de données de l'application.

### 6.4.2. avec l'environnement PVM

La parallélisation sous PVM du code du programme d'une application s'effectue selon le schéma maître-esclave. Elle se décompose en deux types de tâches :

- la tâche maître, dans notre exemple applicatif, sera chargée de distribuer le travail aux différentes stations de travail du réseau, de lancer un calcul (un ou plusieurs motifs du graphe), de récupérer les résultats intermédiaires (provenant des autres motifs du graphe) et de traiter la phase de décision ;
- les tâches esclaves seront chargées de communiquer avec la tâche maître et de lancer le traitement de  $\frac{K}{PE}$  motifs du graphe flot de données de l'application.

Le programme PVM de l'application repose sur une répartition dynamique de tâches sur les stations de travail du réseau. La charge de travail pour l'application doit être adaptée à chaque station de travail. Le choix de la distribution de  $\frac{K}{PE}$  motifs sur chaque station de travail est donc optimal. Le programme est tolérant aux pannes, afin de permettre au programme maître de redistribuer le travail non effectué par les stations de travail en causes à d'autres stations de travail du réseau.

### 6.5. évaluation de performance

Nous avons mesuré la durée d'exécution globale de l'application pour des configurations matérielles allant de 2 à 5 stations de travail. Le résultat du chronométrage d'exécution « mono-station » de travail, ainsi que la durée de chaque phase de l'algorithme A2E pour l'application concernée sont données dans le tableau 6. Ces mesures ont été effectuées sur une station de travail Sun Ultra 10 à 440 MHz. Les résultats sont exprimés en micro-seconde ( $\mu s$ ). Nous obtenons ainsi une durée globale d'exécution de l'algorithme séquentiel de 451  $\mu s$ .

Tableau 6. – Chronométrage des sommets du graphe flot de données de l'application.

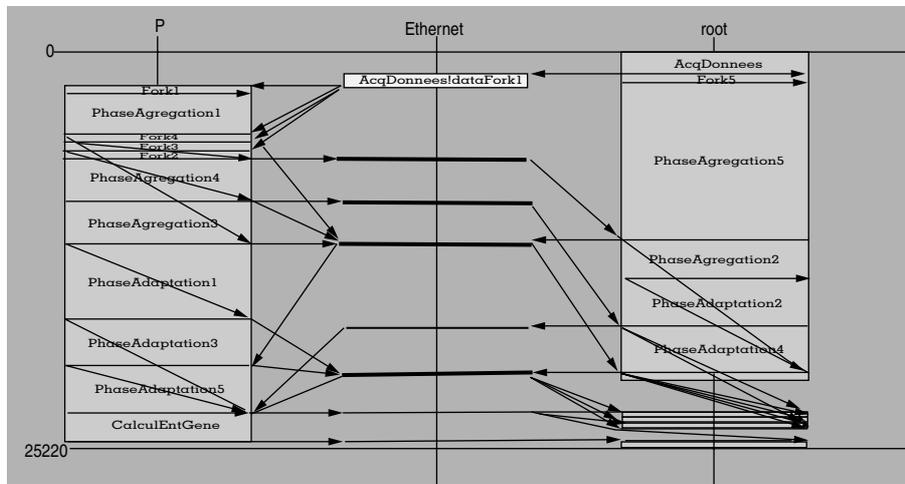
Sommets	Durée d'exécution ( $\mu s$ )
	Prédiction de SynDdex
AcqDonnees	15
Fork (1 à 5)	$5 \times 5$
PhaseAgréation1	26
PhaseAgréation2	25
PhaseAgréation3	27
PhaseAgréation4	28
PhaseAgréation5	100
PhaseAdaptation1	48
PhaseAdaptation2	31
PhaseAdaptation3	30
PhaseAdaptation4	30
PhaseAdaptation5	30
CritèreQualité	3
PriseDécision	4
JoinEnt	21
DimensionM	1
CalculEntGene	18
VISU (1 et 2)	$4 \times 2$
Total	451

D'après le résultat du chronométrage, la durée requise pour examiner une solution candidate, dans la phase d'agrégation varie entre  $8,33 \mu s$  et  $33,33 \mu s$ . Cette durée se situe entre  $5\beta \mu s$  et  $8\beta \mu s$  pour la phase d'adaptation, où la variable  $\beta$  est le nombre moyen d'utilisation des fonctions de composition  $HAG \circ HDG$  et  $HDG \circ HAG$ .

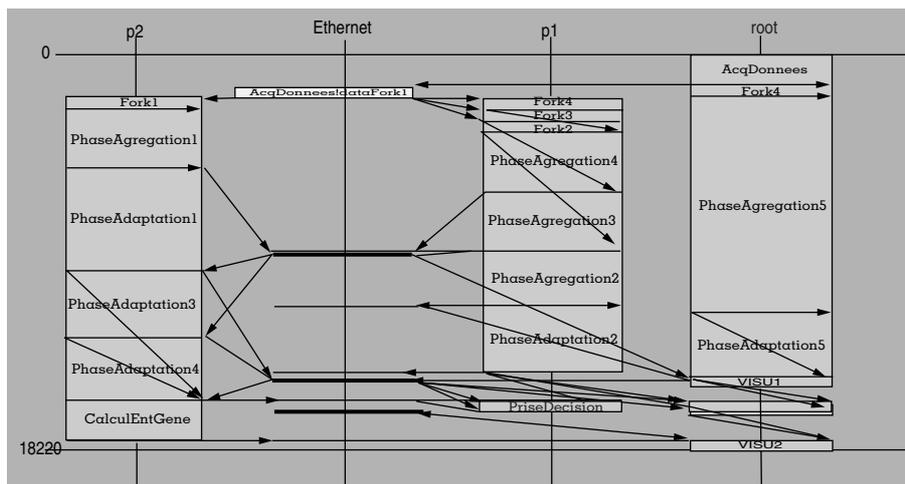
Dans l'échange des messages entre les processus communicants de l'application, le flot de données le plus important correspond à la transmission des données d'acquisition, c.-à-d. un tableau de 150 éléments de type réel double précision (« double ») correspondant aux données des tableaux 2, 3 et 4. La taille du message est de 1,2 Koctets, soit  $150 \times 8$  octets (où les 8 octets correspondent à la taille d'un flottant double précision sur un sys-

tème Sun4). D'après les résultats du chronométrage de la durée mise par un message (de taille nulle<sup>16</sup>) pour effectuer un aller-retour (en anglais, *Round Trip Time* ou RTT) entre deux processeurs pour des communications intra-processeur et pour des communications inter-processeurs (cf. tableau 7), il faut à SynDEX un débit d'environ  $0,75 \text{ byte}/\mu s$  pour transmettre ce message. Par conséquent, il faudra  $0,04 \mu s$  pour transmettre un flottant double précision.

Les prédictions de SynDEX de l'implantation parallèle, sur 2 à 5 stations de travail du réseau, sont représentées sur les figures 9 et 10. Les temps de prédiction sur les graphes temporels sont tels que 100 unités de temps sur le diagramme représente  $1 \mu s$  à l'échelle réelle. À partir de ces figures, on constate que l'aug-



(a) 2 stations de travail



(b) 3 stations de travail

Figure 9. – Diagramme temporel SynDEX : équilibrage des charges sur 2 et 3 stations de travail.

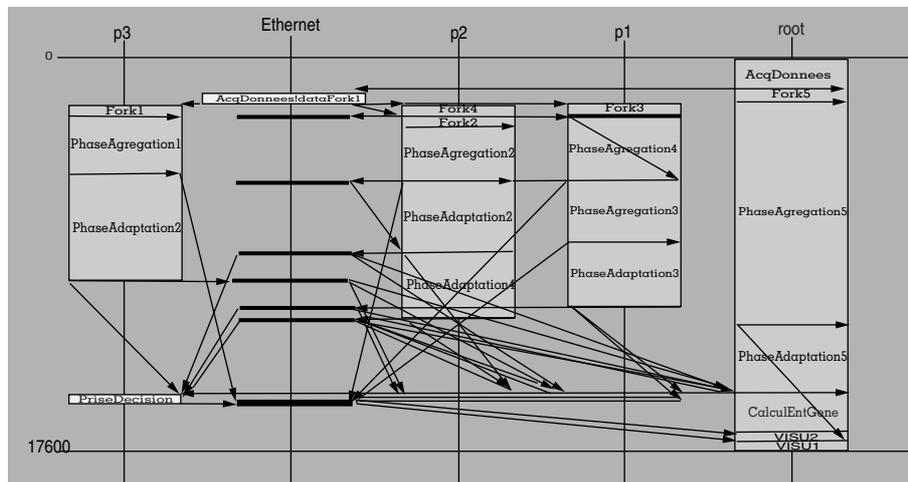
<sup>16</sup> Le message est de petite taille de sorte que le débit a une influence minimale sur la mesure.

Tableau 7. – Comparaison de la RTT SynDEx et PVM sur un réseau Ethernet 100Mbps.

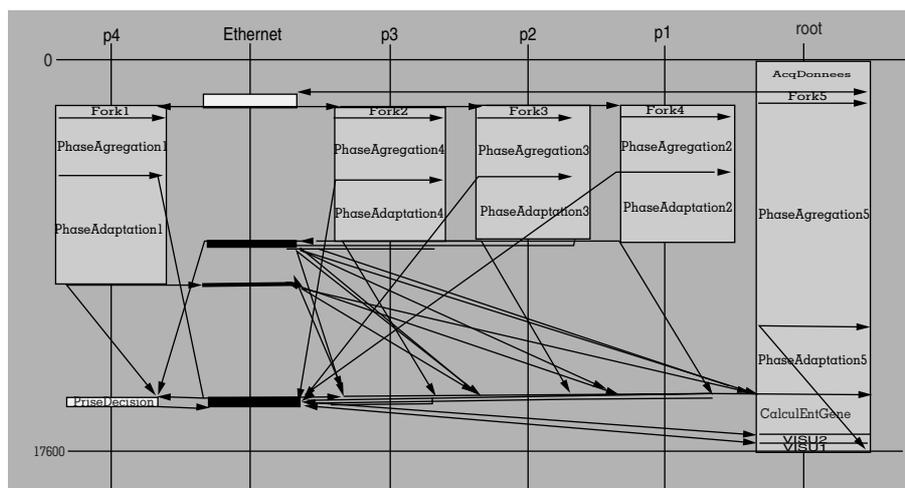
Communications	RTT ( $\mu$ s)	
	Syndex	PVM
intra-processeur	203	805
inter-processeurs	286	893

mentation du nombre de stations de travail entraîne l'augmentation de l'accélération du réseau de stations de travail. Cependant, c'est l'équilibrage des charges (placement-ordonancement des tâches de calcul) qui joue le rôle le plus important. Elle influe donc sur l'efficacité de l'implantation.

Les figures 10(a) et 10(b) montrent que l'efficacité du réseau de stations de travail diminue à partir de 4 processeurs de 30%. Elle passe de  $\frac{451}{182,2} \times \frac{100}{3} = 81\%$  à  $\frac{451}{176} \times \frac{100}{5} = 51\%$  (on a une accélération ou *speed up* sub-linéaire qui reste constant à 2,56). Pourtant, on n'observe aucune dissymétrie de répartition des tâches sur les stations de travail et le coût des communications est complètement masqué par le calcul. Cette perte d'efficacité s'explique par l'existence de périodes d'inactivité de certaines stations de travail. En effet, le coût du calcul des sommets PhaseAgrégation5 et PhaseAdaptation5 sur la station de travail root est plus élevé que celui des autres sommets. Ces temps d'attente sont donc admissibles compte tenu de l'optimisation du graphe flot de données en fonction du nombre d'hypothèses  $H_k$  possibles que le système de fusion doit traiter.



(a) 4 stations de travail



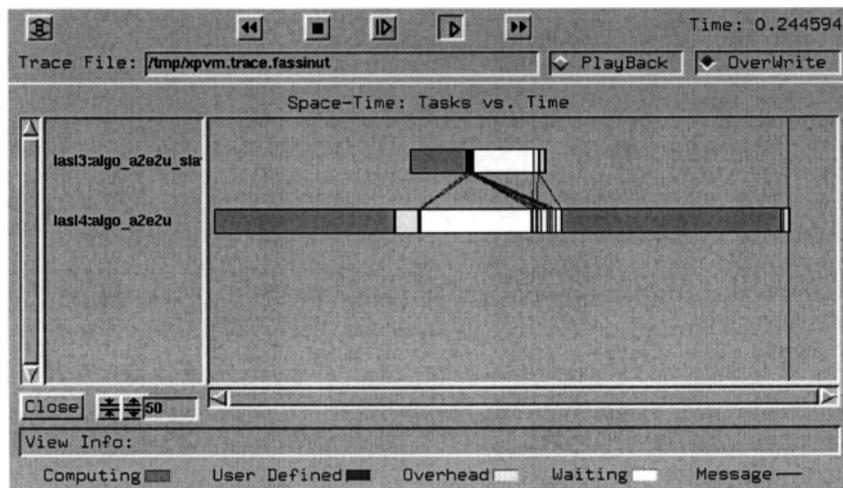
(b) 5 stations de travail

Figure 10. – Diagramme temporel SynDEx : équilibrage des charges sur 4 et 5 stations de travail.

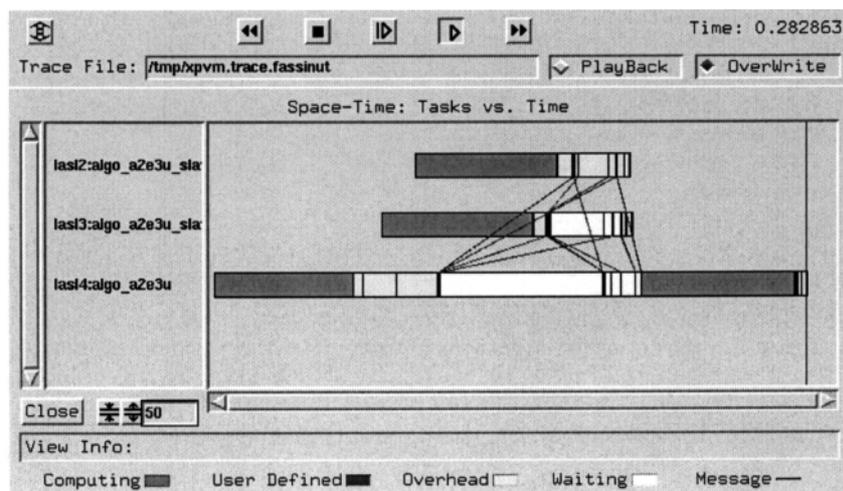
Ces mêmes constatations apparaissent sur les graphes de l'activité des tâches PVM (cf. figures 11 et 12). En effet, on peut observer que les durées d'attente *Waiting* augmentent avec le nombre de tâches PVM. Il faut rappeler que la répartition des charges sur la machine PVM a été effectuée à partir du placement-ordonnancement déterminé pour l'implantation *SynDEX*. Cela nous a permis de simplifier fortement la détermination des emplacements des séquences de communication de PVM. Les primitives de communication appelées par *SynDEX* ont été simplement remplacées par celles de la bibliothèque de communication PVM.

À partir de la figure 11(b) on peut constater encore que :

- la diffusion des données d'acquisition (de taille égale à 1,2 Koctets) compactées dans le tampon d'envoi monopolise de la mémoire ;
- le nombre de communications nécessaire à l'échange de messages entre les tâches PVM augmentent avec celles-ci ;
- les pertes de temps dues aux communications (*Overhead*) augmentent. Il faut dire que les messages échangés sont de petites tailles (moins de 2 Koctets). Il y a beaucoup trop d'initialisations de communications, ce qui nuit aux performances de l'implantation. En effet, nous remarquons sur la figure 13 qu'il y a un surcoût important de l'ordre de la dizaine de milli-secondes auquel il faut ajouter celui de l'empaquetage et du dépaquetage du message. Le premier surcoût se manifeste surtout sur la latence d'amorçage du fait de la prise en charge des communications par le démon PVM ;
- le surcoût dû aux communications allonge la durée d'exécution du programme.

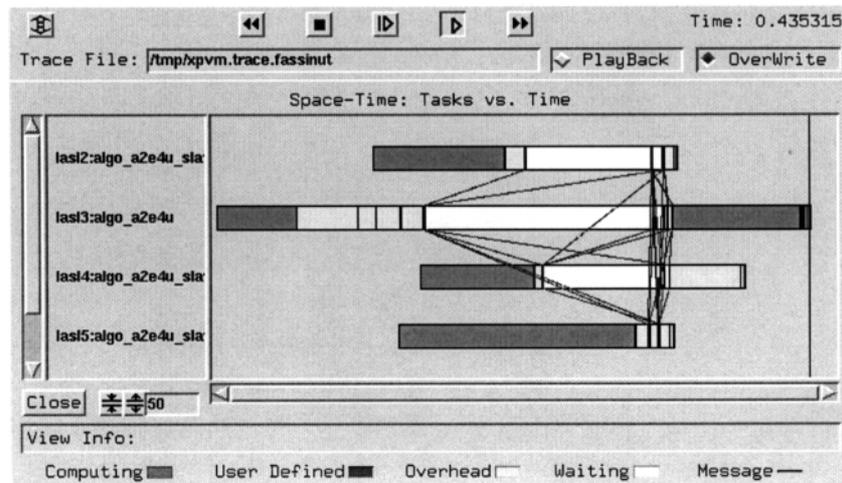


(a) 2 stations de travail

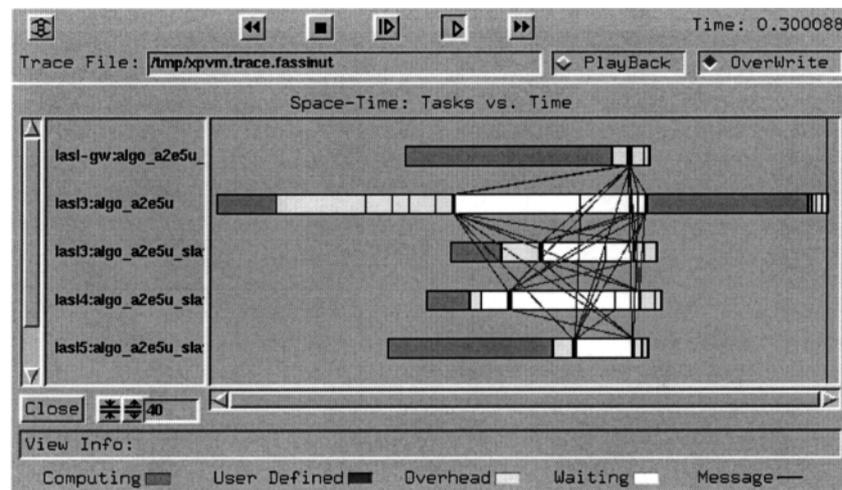


(b) 3 stations de travail

Figure 11. – Activité des tâches maître-esclaves PVM : répartition des tâches sur 2 et 3 stations de travail.



(a) 4 stations de travail



(b) 5 stations de travail

Figure 12. – Activité des tâches maître-esclaves PVM : répartition des tâches sur 4 et 5 stations de travail.

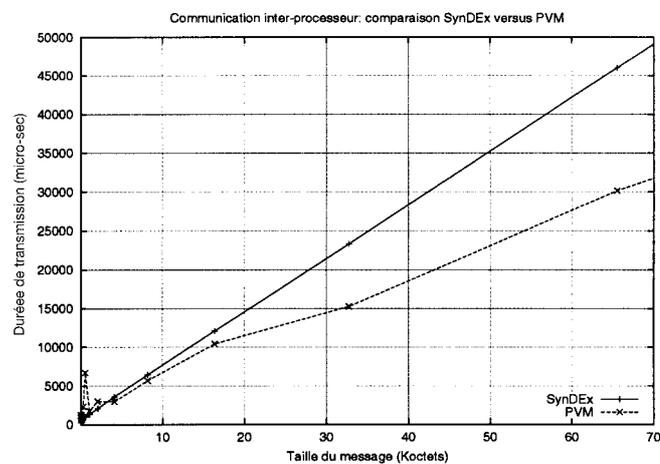


Figure 13. – Durée de transmission inter-processeurs.

Les figures 14(a) et 14(b) décrivent, quant à elles, pour les deux environnements de programmation SynDEX et PVM, respectivement la durée d'exécution et l'efficacité des implantations parallèles en fonction de la configuration matérielle du réseau de stations de travail.

Tout d'abord, il faut remarquer que l'ordre de grandeur des mesures des durées d'exécution de SynDEX n'est pas la même que celui de PVM. Les mesures effectuées avec l'implantation PVM vont de quelques dizaines de milli-secondes à la centaine de milli-secondes. En revanche, les durées d'exécutions avec SynDEX sont de l'ordre de la centaine de micro-secondes.

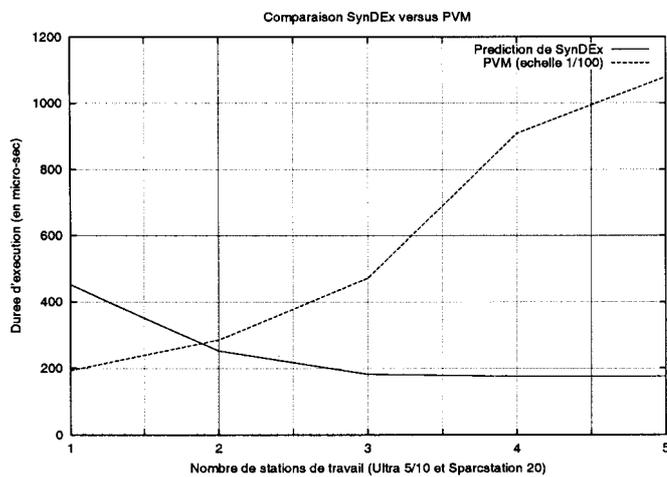
Dans la figure 14(a), nous remarquons également, que l'allure de la courbe de PVM est croissante au lieu d'être décroissante. Cette croissance est d'autant rapide que le nombre de stations de travail constituant la machine virtuelle augmente. La parallélisa-

tion avec PVM s'avère donc très peu rentable. Cela est d'autant plus vrai que l'efficacité de l'implantation passe en deçà de 30% avec deux tâches maître et esclave susceptibles d'être exécutées sur 2 stations de travail. Elle est de moins de 1% au maximum de rentabilité attendue par la description flot de données retenue de l'application.

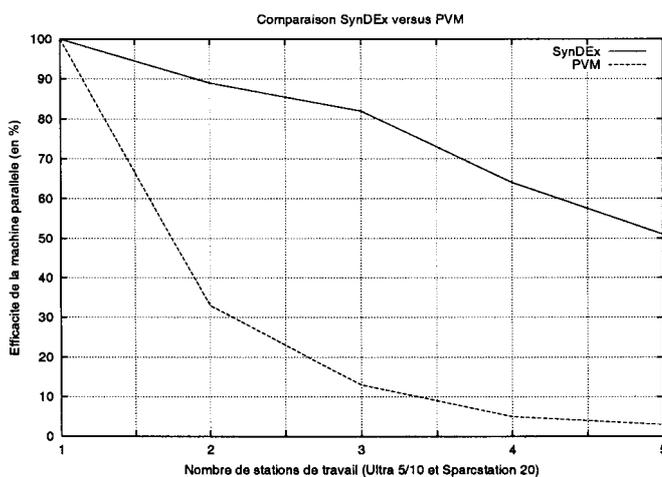
Par contre, l'allure de la courbe de la durée d'exécution avec SynDEX est décroissante. Elle atteint un seuil à partir duquel l'augmentation du nombre de stations de travail dans le réseau est sans influence. On peut dire, dans ce cas, que la parallélisation de l'algorithme A2E est excellente puisque la durée d'exécution est quasiment divisée par le nombre de stations de travail. Bien que le graphe flot de données soit optimal pour 5 stations de travail, l'implantation est plus rentable avec 3 stations de travail.

## 7. conclusion

Dans cet article, nous avons proposé une méthode de fusion probabiliste reposant sur des critères entropiques dont l'intérêt est la réduction de l'espace de combinaison en représentant explicitement le caractère dual de la complémentarité et de la redondance des sources d'information. Afin de mettre en œuvre cette approche, nous avons développé un algorithme de fusion entropique : l'algorithme d'Agrégation Adaptative Entropique (A2E). Nous avons également présenté la parallélisation du code de l'algorithme A2E à partir d'une application de la robotique mobile sur un réseau de stations de travail fonctionnant en mode MIMD-NORMA avec une communication par passage de messages. Dans un contexte peu privilégié (quantité de calculs dont près de 60%<sup>17</sup> est parallélisable, granularité importante, messages de petites tailles inférieurs à quelques kilo-octets) l'utilisation de SynDEX pour l'écriture et l'implantation du code exécutable a permis d'obtenir d'excellentes performances par rapport à PVM. Le placement-ordonnancement des tâches de calcul et le séquençement des communications en utilisant PVM, tout en exploitant au mieux le recouvrement calcul-communication, sont d'une part à la charge du programmeur, et d'autre part, ils sont difficiles à mettre en œuvre. L'utilisation de SynDEX avant PVM, nous a permis de connaître facilement la manière dont devaient être écrites les communications inter-processeurs dans le programme PVM.



(a) Durée de calcul et de communication



(b) Efficacité du réseau de stations de travail

Figure 14. – Évaluation de performance sur SynDEX et PVM.

<sup>17</sup> Si  $PE \rightarrow +\infty$  alors  $pc \rightarrow \frac{1}{Acc(PE)_{max}}$  (loi d'Amdahl) où  $pc$  est le pourcentage de calcul séquentiel. Pour une accélération maximale  $Acc(PE)_{max} = 2,56$  (atteint par SynDEX),  $pc \approx 39\%$ .

## RÉFÉRENCES

- [Aiglon96] C. Aiglon, C. Lavarenne, Y. Sorel, et A. Vicard, Utilisation de SynDex pour le traitement d'images temps-réel. Rapport de Recherche 2968, INRIA Rocquencourt, Septembre 1996.
- [Blaze00] J. Blazewicz, K. Ecker, et T. D. Plateau, *Handbook on parallel and distributed processing*, Springer, 2000.
- [Bloch94] I. Bloch et H. Maître, Fusion de données en traitement d'images : modèles d'information et de décision. *Traitement du Signal*, 11(6):435-446, 1994.
- [Cappel98] F. Cappelletto et O. Richard, Architectures parallèles à partir de réseaux de stations de travail : réalités, opportunités, enjeux. *Calculateurs Parallèles, Hermes*, 10(1):9-34, Février 1998.
- [Demp68] A. P. Dempster, A generalization of bayesian inference. *Journal of the Royal Statistical Society*, 30:205-247, 1968.
- [Dubois88] D. Dubois et H. Prade, *Théorie des possibilités, applications à la représentation des connaissances en informatique*. Éditions Masson, Paris, 1988.
- [Duda73] R. O. Duda et P. E. Hart, *Pattern classification and scene analysis*. John Wiley and Sons publication, New-York, Édition Wiley Interscience, January 1973.
- [Fass99] B. Fassinut-Mombot, M. Zribi, et J. B. Choquel, Fusion of information multisensor heterogeneous using an entropy criterion. Dans *Proceedings of the Second International Conference on Information Fusion, FUSION'99*, pages 1204-1210, San Francisco, USA, July 1999.
- [Fass00] B. Fassinut-Mombot et J. B. Choquel, An entropy method for multi-source data fusion. Dans *Proceedings of the Third International Conference on Information Fusion, FUSION 2000*, volume II, pages ThC5 17-23, Paris, France, July 2000.
- [Fass01a] B. Fassinut-Mombot, J. B. Choquel, et M. Paidavoine, Using Entropy Model for Identity Information Fusion. Dans *International ICSC Congress, ISI'2001, Proceedings of the International Workshop on Autonomous Artificial Systems Exploring Hostile Environments, AASE-HE'2001*, page à paraître, Université Américaine de Dubai, U.A.E., March 2001. ICSC Academic Press.
- [Fass01b] B. Fassinut-Mombot, M. Zribi, et J. B. Choquel, Distributed estimation and joint probabilities estimation by entropy model. Dans A. Mohammad-Djafari, éditeur, *Bayésien Inference and Maximum Entropy Methods In Science and Engineering, 20<sup>th</sup> International Workshop*, pages 292-302. American Institute of Physics, 2001.
- [Fass02] Bienvenu Fassinut-Mombot, *Contribution à la fusion d'informations par une approche entropique : algorithme et architecture dans un contexte de perception multi-sources*. Thèse de Doctorat 3<sup>ème</sup> cycle, Université du Littoral Côte d'Opale (ULCO), Février 2002. N° d'ordre ULCO 2002.02.
- [Gatin00] Laurent Gatineau, *Algorithmes concurrents pour les problèmes de mise en correspondance dans les séquences d'images*. Thèse de Doctorat 3<sup>ème</sup> cycle, Université Evry Val d'Essonne, Janvier 2000.
- [Geist94] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, et V. Sunderam, *PVM: Parallel Virtual Machine - An user's guide tutorial for networked parallel computing*. MIT Press, Édition Scientific and Engineering Computation, 1994.
- [Grand98] T. Grandpierre, C. Lavarenne, et Y. Sorel, Modèle d'exécutif distribué temps-réel pour SynDex. Rapport de Recherche 3476, INRIA Rocquencourt, Août 1998.
- [Hoare85] C.A.R. Hoare, *Communicating sequential processes*. Printice-Hall, 1985.
- [Lars98] T. Dall Larsen, N. A. Andersen, et O. Ravn, Sensor management for identity fusion on a mobile robot. Dans *Proceedings of the 5<sup>th</sup> International Conference on Control, Automation, Robotics and Vision, ICARCV'98*, Singapore, December 1998.
- [Lava91] C. Lavarenne, O. Seghrouchni, Y. Sorel, et M. Sorine, SynDex un environnement de programmation pour application de traitement du signal. Dans *13<sup>ème</sup> Colloque sur le Traitement du Signal et des Images, GRETSI*, 1991.
- [Lava93] C. Lavarenne et Y. Sorel, Performance optimization of multiprocessor real-time applications by graphs transformations. Dans *Proceedings of the PARCV'93 Conference*, France, 1993.
- [Luo89] R. C. Luo et M. G. Kay, Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):901-931, October 1989.
- [Perche95] P. B. Perche, D. Pomorski, et M. Staroswiecki, Méthodes d'induction par arbre de décision utilisant un critère entropique. Dans *2<sup>ème</sup> Conférence Internationale sur l'automobile Industrielle*, CRAN, Nancy, 7-9 Juin 1995.
- [Quin83] J.R. Quinlan, Learning efficient classification procedures and their application to chess and games. Dans J.G. Carbonell R.S. Michalski et T.M. Mitchell, éditeurs, *Machine Learning: an artificial intelligence approach*, pages 463-482. Tioga Publishing Company, Palo Alto, 1983.
- [Quin86] J.R. Quinlan, Induction of decision trees. *Machine Learning*, 1:81-106, 1986.
- [Shafer76] G. Shafer, *A mathematical theory of evidence*. Princeton University Press, New Jersey, Édition Princeton, 1976.
- [Shan48] C. E. Shannon, A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379-423 et 623-656, Juillet 1948. <http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html>
- [Shan49] C. E. Shannon, Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656-715, Octobre 1949.
- [Smets90] P. Smets, The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):447-458, 1990.
- [Sorel94] Yves Sorel, Massively parallel systems with real time constraints, the Algorithm Architecture Adequation. Dans *Proceedings Massively Parallel Computing Systems, the Challenges of Genreal-Purpose and Special-Purpose Computing Conference*, Ischia, Italy, May 1994.
- [Sorel96] Yves Sorel, Real time embedded image processing application using the A<sup>3</sup> methodology. Dans *IEEE International Conference on Image Porcessing*, November 1996.
- [Toro82] Victor Manuel Toro Cordoba, *Contribution à l'analyse structurale de systèmes complexes à l'aide de l'entropie et ses généralisations*. Thèse de Doctorat 3<sup>ème</sup> cycle, USTLFA, Mars 1982. No d'ordre 955.
- [Vic99] Annie Vicard, *Formalisation et optimisation des systèmes informatiques distribués temps réel embarqués*. Thèse de Doctorat 3<sup>ème</sup> cycle, Université Paris 13, Juillet 1999.
- [Zadeh78] Lofti A. Zadeh, Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3-28, 1978.

Manuscrit reçu le 6 mai 2002

### LES AUTEURS

#### Bienvenu FASSINUT-MOMBOT



Bienvenu FASSINUT-MOMBOT est titulaire d'une Licence et Maîtrise EEA option EMO de l'Université de Rouen (1996), du DEA Instrumentation et Informatique de l'Image de l'Université de Bourgogne (1997) et du diplôme de docteur en Génie Informatique Automatique et Traitement du Signal de l'Université du Littoral Côte d'Opale (2002). Son activité de recherche s'est développé à cheval sur deux domaines complémentaires : la Fusion d'Informations et l'Adéquation Algorithme-Architecture. Il est actuellement Ingénieur de Recherche en Calcul Scientifique à l'Université du Littoral Côte d'Opale (ULCO) au sein du Laboratoire d'Analyse des Systèmes du Littoral (LASL). Ses nouvelles activités de recherche au sein du groupe fusion d'informations du LASL portent actuellement sur les systèmes de perception et de modélisation GPS-Estime pour la navigation terrestre.

#### Jean-Bernard CHOQUEL



Jean-Bernard CHOQUEL a obtenu son Doctorat en électronique en 1994, dans le domaine du traitement du signal et des images, à l'Université des Sciences et Technologies de Lille. Il est actuellement Maître de Conférences à l'Université du Littoral Côte d'Opale. Auparavant il a travaillé 9 ans dans l'industrie sidérurgique. Il est entré avec son diplôme d'Ingénieur EUDIL à l'IUT du Littoral à Calais, puis à l'IUP. Il a entrepris de nombreux travaux correspondant aux activités d'Enseignement dans les disciplines concernant l'Automatique, l'Informatique Industrielle et l'Électronique, exécuté des contrats industriels, et réalisé de nombreuses tâches administratives. Au Laboratoire d'Analyse des Systèmes du Littoral, il poursuit sa Recherche dans le domaine de la « Fusion d'Informations et Adéquation Algorithme-Architecture ». Il est Président du Groupe Nord de la SEE depuis septembre 2002, et il a obtenu le grade de Senior SEE en 2002.