

« Round-About » : une architecture de traitement d'images pour la mesure en temps réel de paramètres physiques

J. Dubois, G. Jacquet, G. Motyl, F. Celle, V. Fischer

Laboratoire Traitement du Signal et Instrumentation (LTSI)

Université Jean Monnet

23 Rue du Docteur Michelon, 42023 Saint Etienne Cedex, France

jdubois@univ-st-etienne.fr, jacquet@univ-st-etienne.fr

Résumé - L'architecture Round-About est dédiée à la mesure par traitements d'images de paramètres physiques. Elle a été conçue pour s'adapter au déséquilibre existant entre le volume de données d'entrée (images) et celui des résultats (mesures). Pour cela, les deux flots associés à ces données d'entrée et de sortie ont été séparés. De plus, une topologie différente leur est appliquée, le flot de commandes étant associé à celui des données de sortie. La modularité du système permet d'adapter la configuration matérielle aux performances nécessaires à l'application. Cette modularité est illustrée dans cet article par la description de deux implantations, respectivement sur des cibles de type DSP et FPGA, permettant une mesure de vitesse en temps réel allant d'une fréquence vidéo standard de 25 images/s à 100 images/s.

Abstract – *The Round-About architecture is dedicated to measurement of physical parameters by image processing. The design of this architecture was defined to take into account the disproportion between the flow of input data and the flow of results. These two flows have been split. Moreover, different topologies have been selected for the two previous flows, the control flow being associated to the flow of results. Several levels of performance can be reached by using the modularity of the system. Two examples of implementation are given : one based on DSP and another using FPGA devices.*

1. Introduction

L'architecture Round-About, décrite dans cet article, est destinée à la mesure de paramètres physiques à partir de traitements d'images. La définition de cette architecture a été réalisée en adéquation avec ce type d'applications. En effet, la première spécificité des algorithmes de mesure sur des images réside dans le déséquilibre entre le volume des données d'entrée et celui des données de sortie. En entrée, le système traite un flot d'images généralement à haute résolution et haut débit, alors qu'en sortie, le volume des résultats est faible (mesures). Une autre spécificité réside dans le fait que, dans la plupart des cas, les mesures sont reproduites dans plusieurs régions de l'image. L'image peut alors être découpée en zones (nommées imagettes), de ce fait la parallélisation des calculs est implicite, chaque imagette analysée donnant naissance à un point de mesure. Les mesures étant généralement indépendantes, le traitement de plusieurs imagettes peut être réalisé de manière simultanée, par le même algorithme ou par des algorithmes différents. Ces derniers pouvant être très différents en terme de besoins, l'architecture doit être basée sur un concept modulaire et permettre une modification du nombre et même de la nature des éléments de calcul.

Après une présentation générale de l'architecture et des concepts associés, une mise en œuvre de celle-ci pour des mesures de champ de vitesse en mécanique des fluides est décrite. Deux implantations matérielles, permettant d'atteindre un traitement en temps réel, sont détaillées, l'une utilisant des modules de calcul à base de DSP, l'autre à base de FPGA.

2. Flot de données d'entrée dominant

Dans les applications de mesure, les trois flots parvenant à l'élément de calcul en phase de fonctionnement sont fortement déséquilibrés :

- le flot de données d'entrée est prépondérant ;
- le flot de données de sortie (résultats) est relativement faible ;
- le flot de commande adapté au découpage en imagettes correspond uniquement à des macro-commandes envoyées vers l'élément de calcul, il est donc très faible.

La figure 1 illustre cette répartition dans un modèle où tous les flots sont séparés.

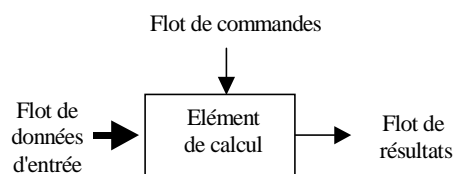


FIG.1 : Modèles à flots séparés

Les modèles habituels de communication (modèle de Von Neuman, de Harvard) ne sont pas adaptés à ce déséquilibre. Un autre modèle peut alors être considéré. Il possède un flot de données d'entrée séparé des deux autres. Ce nouveau modèle s'adapte particulièrement bien à la catégorie d'applications considérée précédemment. En effet, le volume de données étant important en entrée du système il est intéressant de lui accorder une voie d'accès propre, c'est à

dire, a priori, une bande passante plus importante. Le flot de commandes étant limité, il peut être mélangé avec les résultats sans nécessiter une grande bande passante. La figure 2 représente ce modèle nommé « modèle à flots de données d'entrée dominant ».

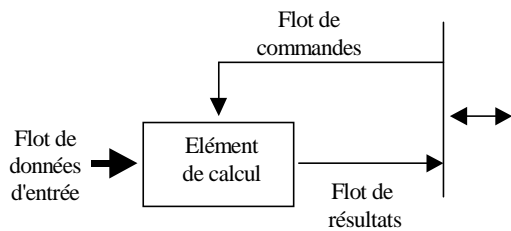


FIG. 2 : Modèle à flot de données d'entrée dominant

La séparation des flots permet d'associer à chacun d'eux des topologies différentes (anneau, bus, pyramide...) adaptées aux caractéristiques de la communication souhaitée et donc de créer des architectures plus performantes.

3. Round-About

Le modèle à flot de données d'entrée dominant a été retenu pour élaborer l'architecture Round-About. La topologie associée au flot de commandes et résultats est celle d'un anneau unidirectionnel, il permet un meilleur contrôle des flots par retour à l'émetteur des données transmises. Pour les données d'entrée, une topologie de type bus a été choisie, elle permet de transférer simultanément les mêmes données vers plusieurs éléments de calcul (transfert sous le mode multi-cast).

L'architecture obtenue a été appelée Round-About en référence à sa forme globale (figure 3a).

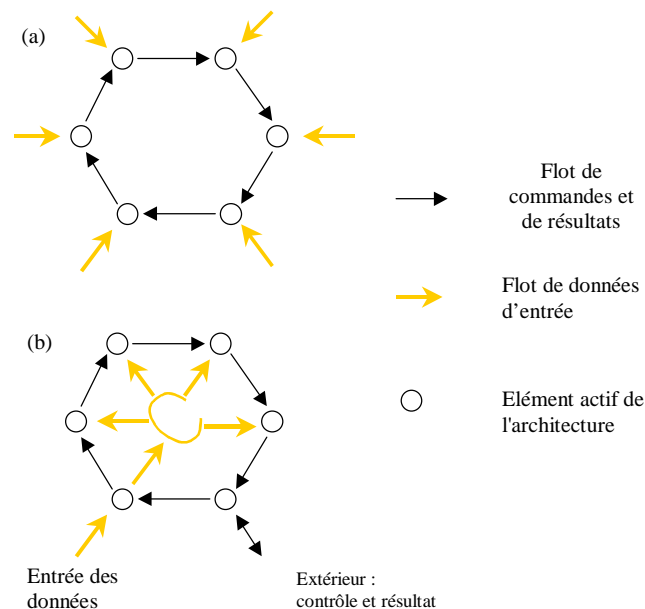


FIG. 3 : Formation de Round-About

Dans la généralité de l'architecture, l'accès extérieur n'est pas représenté. D'un point de vue pratique, un lien privilégié vers l'extérieur de l'architecture doit être prévu. Il permet de

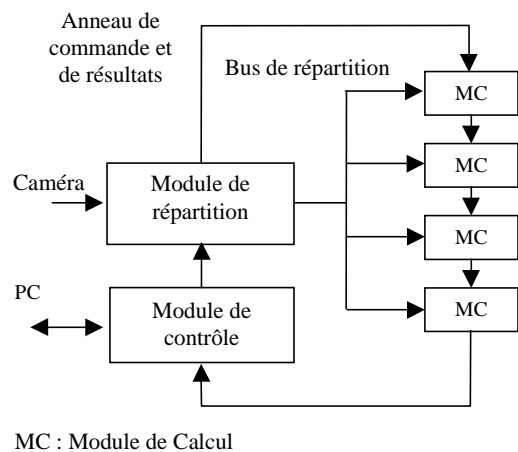
recupérer les commandes à exécuter et de transmettre les résultats vers l'utilisation (figure 3b). Un élément spécifique doit permettre de réaliser cette interface : c'est un élément appelé module de contrôle dans notre architecture.

De la même manière, concernant le flot de données d'entrée, un élément gère l'arrivée des données et les envoie vers tous les éléments ayant en charge le traitement. Ces deux types d'éléments sont respectivement nommés module de répartition et modules de calcul (MC).

Le module de répartition appartient aussi à l'anneau car il doit être contrôlé pour permettre la synchronisation de l'échange des données. Dans le cas où les zones de mesure ne sont pas indépendantes, le module de répartition est aussi capable de gérer du multi-cast par un système de double handshake.

Dans l'implémentation choisie (figure 4), le module de répartition joue le rôle d'interface avec la caméra et les modules de calcul (MC). Le module de contrôle comporte un lien avec un PC. Ce lien permet au module de contrôle de recevoir le séquençement des ordres à effectuer et de transférer les résultats. Il lui permet aussi de gérer les configurations des modules de calcul (unités de programmes pour des DSP ou microprocesseurs, configurations matérielles pour des circuits logiques programmables).

Ce système est très flexible car il est même possible d'effectuer une modification en ligne du traitement à réaliser, seul le MC concerné étant indisponible pendant la modification.



MC : Module de Calcul

FIG. 4 : Architecture retenue

La modification du nombre de MC est relativement aisée : elle est naturelle pour la topologie bus, l'extension de l'anneau se faisant par détection automatique des MC par des trames spécifiques envoyées au démarrage.

Toutes les communications sur l'anneau sont d'ailleurs basées sur ce principe de trames qui transpose une méthode de communication entre ordinateurs à la communication d'éléments d'architectures de traitement. Pour notre architecture, un seul élément (le module de contrôle) génère un trafic continu de trames contenant des commandes pour les MC et de trames vides pour récupérer les résultats, à l'image de certaines normes récentes de réseaux informatiques.

4. Détail de l'implantation du système

4.1 Communication entre modules

L'anneau est constitué d'un ensemble de liaisons point à point unidirectionnelles. Chaque module comporte une unité de communication identique. Elle comporte deux blocs asynchrones qui sont respectivement en charge de la réception et de l'émission des trames. Dans l'implantation choisie la largeur de l'anneau est de 8 bits et chaque trame est composée de 6 octets. La séparation de l'unité de communication en deux parties permet de recevoir un octet tout en transférant un autre. Cela permet de diminuer le temps inter-octets sur l'anneau.

Seul le module de contrôle émet des trames vers les modules de répartition et de calcul. Ils reçoivent ainsi les commandes et ils complètent les trames vides pour transférer des informations au module de contrôle. Ces informations peuvent être de deux types : le signalement d'un dysfonctionnement ou la transmission des résultats. Les trames vides peuvent être remplies par n'importe quel module, cependant une fois complétées, seul le module de contrôle peut les intercepter.

Les trames comportant des commandes sont destinées à un module en particulier. A la réception de la trame, le module extrait la commande, et renvoie la trame marquée au module de contrôle pour confirmer que la commande va être traitée. Dans le cas où le module de calcul concerné est occupé, la trame est retournée sans modification. Le module de contrôle l'émettra à nouveau ultérieurement.

4.2 Module de répartition

Le module de répartition est composé de quatre unités (figure 5) en charge :

- de la numérisation du signal caméra ;
- du stockage temporaire des données résultant de la numérisation ;
- du contrôle du module ;
- de la réception et de l'émission de trames.

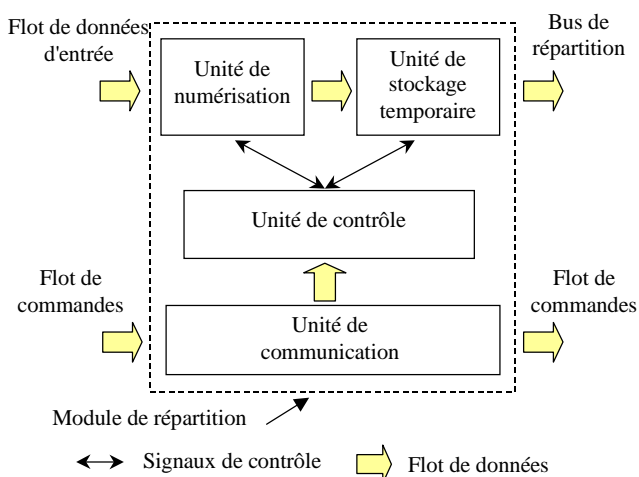


FIG. 5 Module de répartition

Les données, une fois numérisées, sont stockées temporairement de manière à s'affranchir de l'asynchronisme entre le flot issu de la caméra et celui permettant d'approvisionner les MC. Le transfert est déclenché sur commande du module de contrôle. Le transfert de chaque mot de donnée du module de répartition aux MC est réalisé par un protocole de double handshaking. Le bus servant à la répartition des données est de 32 bits.

4.3 Module de contrôle

Le module de contrôle régule le fonctionnement du système. Il assure l'initialisation des modules de calcul, leur commande ainsi que celle du module de répartition, et enfin il permet de transférer les résultats provenant des modules de calcul vers le PC. Le module de contrôle est composé de plusieurs unités permettant de réaliser ces opérations (figure 6).

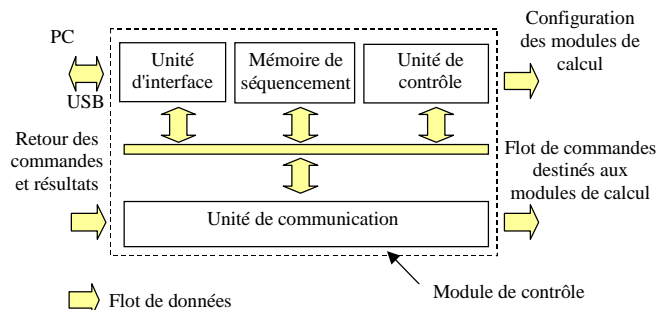


FIG. 6 Organisation du module de contrôle

L'unité de contrôle est le cœur de ce module, elle supervise les autres unités.

L'unité d'interface gère les échanges avec le PC. Des données peuvent être transférées en permanence du PC vers le module de contrôle, ce sont :

- des données de configuration (ou programmation) des modules de calcul, au démarrage ou en cours d'exécution (reconfiguration),
- des trames de commandes destinées aux MC, l'unité de contrôle assure leur stockage temporaire dans une mémoire de séquençement avant de les transmettre,

Les résultats sont aussi transmis en permanence vers le PC, permettant ainsi une restitution ou un contrôle en temps réel.

La communication avec le PC est réalisée à l'aide du bus USB. L'unité de contrôle est constituée d'un micro-contrôleur ST10.

4.4 Module de calcul

Le module de calcul met en œuvre l'algorithme de traitement pour lequel il a été configuré. Il est composé de plusieurs unités (figure 7).

L'unité de contrôle gère le décodage des trames et l'exécution des tâches en fonction des disponibilités.

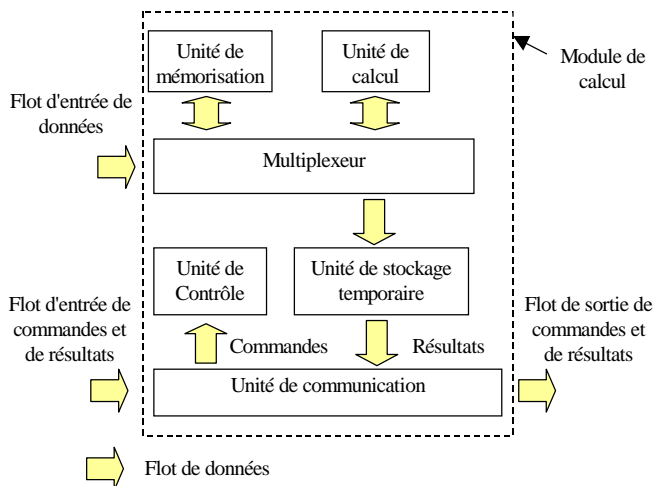


FIG. 7 Organisation du module de calcul

L'unité de mémorisation permet de stocker les données provenant du bus d'entrée. Les données sont alors disponibles dès que l'unité de calcul a achevé un traitement. Elle peut aussi servir de mémoire de travail pour l'unité de calcul si celle-ci n'a pas de mémoire interne. L'aiguillage est réalisé par un multiplexeur.

Les résultats sont stockés temporairement en attente d'une trame vide pour être envoyés.

L'unité de mémorisation est réalisée à l'aide de deux mémoires de 128 K mots de 32 bits. Le multiplexeur, les unités de stockage temporaire, de contrôle et de communication sont implantées sur un FPGA de type Xilinx XC4036XLA. La flexibilité d'une implantation sur un FPGA permet de s'adapter au changement de nature de l'unité de calcul.

5. Application à la mesure de vitesse

La Vélocimétrie par Images de Particules (Particle Image Velocimetry) permet de réaliser des mesures de champs de vitesse (figure 8) dans des écoulements de manière non-intrusive [1].

Des particules sont injectées dans l'écoulement et servent de marqueurs. Une tranche laser illumine un plan à l'intérieur de ce fluide : le plan alors sélectionné représente le plan d'observation.

L'acquisition est réalisée à l'aide d'une caméra CCD qui enregistre les positions des particules dans le fluide. Une comparaison entre la position des particules sur deux images successives de l'écoulement permet l'extraction d'un champ de vecteurs vitesse. Chaque vecteur vitesse est mesuré sur une zone de l'image (imagelette) différente. Le calcul de chaque vecteur est usuellement réalisé par inter-corrélation.

La mesure de ces champs de vecteurs vitesse en temps réel à 25 images/s voire 100 images/s ne peut se faire que sur des architectures matérielles dédiées. L'architecture Round-About permet de s'adapter à la vitesse et à la précision demandées.

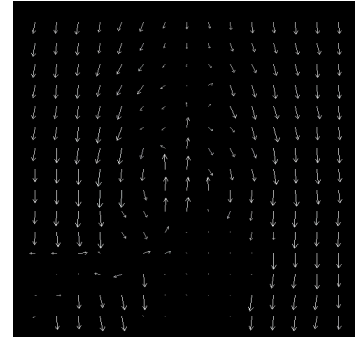


FIG. 8 : Exemple de mesure de vitesse sur des images d'écoulements

5.1 Mesures sur images à niveau de gris

La mesure de champs de vecteurs vitesse peut être faite à partir de paires d'images dont la résolution est de 512x512 pixels et codées sur 8 bits. Afin de limiter l'importance des calculs, l'inter-corrélation est réalisée par FFT [2]. L'implémentation de l'unité de calcul doit alors être faite à partir de DSPs. Nous avons retenu un DSP Texas Instruments de type TMS320C6201. Avec 256 points de mesure par images, la performance atteinte pour 4 modules de calcul est de 6400 vecteurs par seconde soit 25 images par seconde.

5.2 Mesures sur images binaires

Pour obtenir des performances supérieures en terme de vitesse, une technique de corrélation à partir d'images binaires a été développée. Dans ce cas, l'unité de calcul peut être réalisée à l'aide du FGPA déjà présent au sein du module de calcul [3]. Les performances atteintes dans les mêmes conditions que précédemment sont de 28000 vecteurs par seconde soit plus de 100 images par seconde avec 4 modules de calcul.

6. Conclusion

L'architecture développée permet la mesure de paramètres physiques en temps réel. La vélocimétrie par images de particules en temps réel peut être rendu possible grâce aux performances de notre architecture. D'autres applications sont envisagées tels que la profilométrie 3D.

Références

- [1] C.E. Willert, M. Gharib, *Digital particle image velocimetry*, Experiments in fluids, vol. 10, pp. 181-193, 1991.
- [2] J. Dubois, G. Jacquet, V. Fischer, G. Motyl and R. Fouquet, *Parallel Architecture for Physical Measurements by Image Processing*, 15th Conference on Digital Design on Circuits and Integrated Systems, Montpellier, pp. 444-449, 2000.
- [3] J. Dubois, T. Tjahjadi, G. Jacquet et S. Coudert, *Mesure temps réel de mouvement avec une précision sub-pixel en PIV à partir d'images binaires*, Colloque Interdisciplinaire en Instrumentation, Vol.1, pp. 441-449, ed. Hermès, Paris, 2001.