

TRAITEMENT D'IMAGE COULEUR TEMPS REEL SUR L'ARCHITECTURE PARALLELE GFLOPS

Dominique HOUZET

IRIT-ENSEEIH
118 route de Narbonne
31062 TOULOUSE
Tél: (33) 61 58 83 18
Fax: (33) 61 55 62 58
e-mail: houzet@irit.fr

L'objectif de cet article est de montrer la faisabilité du traitement d'image couleur temps-reel sur une architecture programmable. L'architecture parallèle GFLOPS, développée au sein de notre laboratoire, a été prise comme support. Cette étude concerne la mise en oeuvre et l'évaluation sur cette architecture, d'algorithmes d'extraction de contours sur des images noir et blanc et des images couleur, ces algorithmes étant une étape importante dans de nombreuses applications.

The aim of this paper is to show the feasibility of real-time color Image Processing with programmable architectures. The GFLOPS architecture, developed by our laboratory, has been used to implement and evaluate several edge detection algorithms with black and white images and color images, which are typical Image Processing algorithms.

1. INTRODUCTION

Les applications réelles de Traitement d'Image sont de plus en plus diversifiées, et par là même utilisent des algorithmes aux caractéristiques très diverses. Pour étudier et implémenter de telles applications, il devient primordial de disposer d'architectures programmables capable de traiter ces algorithmes en temps réel. C'est l'objectif de l'architecture GFLOPS développée à l'IRIT. Cette architecture fait suite au projet SYMPATI-2 [1] [2], qui a abouti à la commercialisation début 1993 d'un Processeur Ligne de Traitement d'Image SIMD constitué de 32 à 128 processeurs, développé en collaboration avec le CEA-Saclay.

Nous avons étudié plusieurs algorithmes d'extraction de contours proposés pour les images noir et blanc. Il est intéressant dans de nombreux cas de pouvoir prendre en

compte des images multi-plans comme les images RVB, le but étant d'effectuer les traitements en temps-réel.

La première partie de cet article concerne la présentation de la structure et des caractéristiques de l'architecture utilisée. La seconde partie a pour objectif de présenter et de comparer plusieurs implémentations d'algorithmes d'extraction de contour.

2. PRESENTATION DE L'ARCHITECTURE

L'architecture GFLOPS [3] [4] a été définie pour qu'elle puisse prendre en compte tout type d'algorithmes de Traitement d'Image avec efficacité, par l'utilisation judicieuse des formes de parallélisme qui leur sont associées. Pour cela cette architecture peut fonctionner aussi bien en mode asynchrone (MIMD) qu'en mode synchrone (SIMD). Pour cela



chaque processeur possède sa propre unité de contrôle. Cette architecture est organisée en ligne, chaque processeur possédant un banc mémoire et pouvant accéder à tous les bancs mémoires en plusieurs étapes grâce à un réseau d'interconnexion. Chaque image est répartie sur l'ensemble des bancs mémoire. La figure 1 présente une configuration de l'ensemble de l'architecture. Seules les liaisons de 5 processeurs avec le réseau sont représentées.

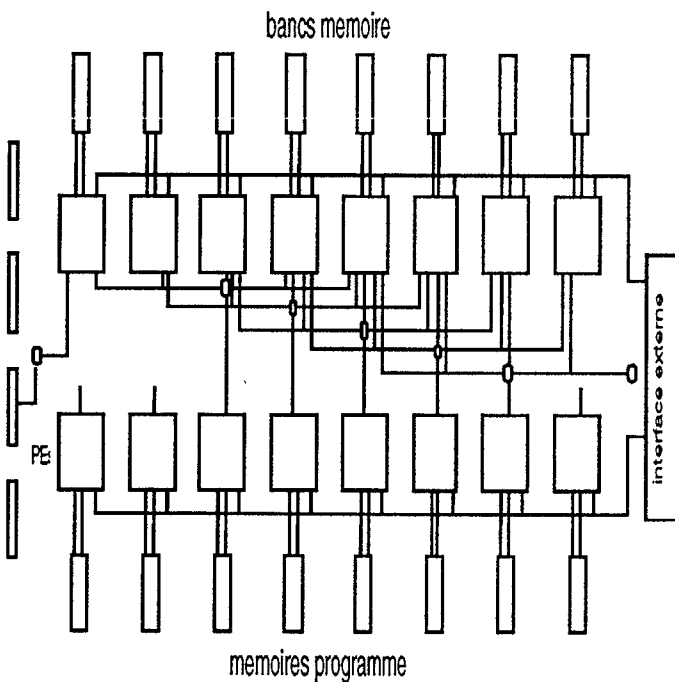


figure 1: architecture générale avec 8 processeurs

Les performances visées pour cette architecture sont celles des super-calculateurs actuels, en ce qui concerne les calculs sur des entiers. Ces performances peuvent être atteintes grâce à l'utilisation combinée du parallélisme globale au niveau de l'architecture, et du parallélisme locale à chaque processeur. Ces processeurs sont de type VLIW (Very Long Instruction Word), composés de 3 unités de traitement générales (ALU 24 bits) fonctionnant en parallèle à chaque cycle horloge. Ces ALUs peuvent traiter aussi bien des données que des adresses. A chaque instruction d'un cycle horloge de durée, le processeur effectue en parallèle, un accès mémoire externe, l'accès à l'instruction suivante, une opération sur chaque ALU et un

accès à la mémoire locale. La figure 2 présente la structure de ce processeur.

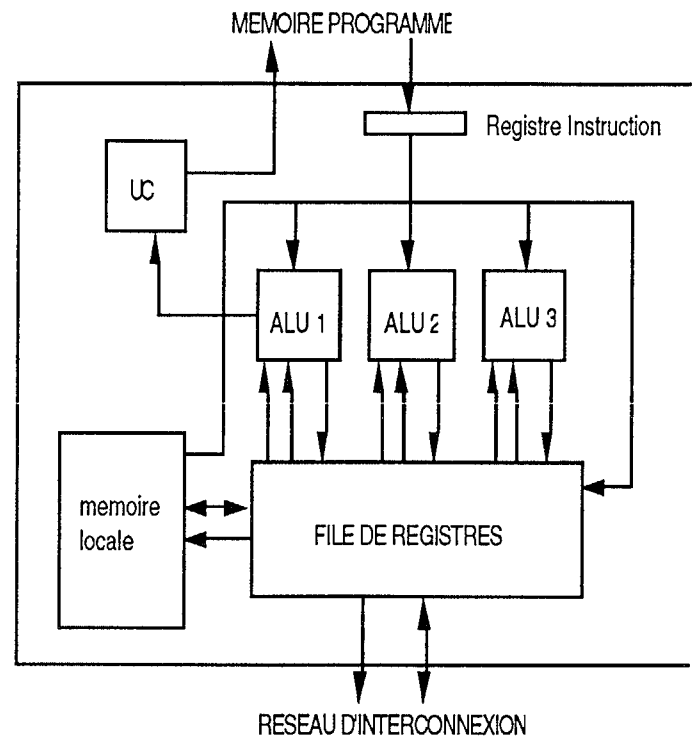


figure 2: schéma interne du processeur

La structure 24 bits de ces processeurs permet de traiter aussi bien des images noir et blanc que des images couleur RVB. De plus les 3 ALUs peuvent servir à traiter en parallèle les 3 composantes couleur d'une image.

3. APPLICATION A LA DETECTION DE CONTOURS

Nous avons tout d'abord implanter quelques algorithmes développés pour des images noir et blanc comme SOBEL, DRF, ISEF [7]. Ces algorithmes ont ensuite été adaptés aux images couleur en les appliquant à chacun des plans RVB, puis en calculant le contour final par d'autres algorithmes: maximum, moyenne, Di Zeno [5] [6]. Les temps d'exécution obtenus dans le cas d'images couleur sont plus importants que pour les images noir et blanc, sans être excessifs. En effet, on ne dépasse jamais un facteur 4 en passant de noir et

blanc à couleur. La figure 3 présente les temps d'exécution de ces différents algorithmes pour des images noir et blanc et des images couleur de taille 256*256 avec une configuration de l'architecture à 16 processeurs. SOBEL correspond à une fenêtre 3*3 appliquée à tous les pixels de l'image en 2 passes de même durée. Dans le cas des images noir et blanc, le réseau est saturé. Par contre, dans le cas des images couleur, ce sont les ALUs qui sont saturées (il y a trois fois plus de calculs), et le réseau n'est utilisé qu'à 50 %. DRF correspond à un filtre récursif par propagations en 4 passes suivi par une différence avec l'image initiale et une binarisation. Les passes se font successivement sur les colonnes puis les lignes de l'image. Il est nécessaire d'effectuer une transposition entre les 2 étapes et à la fin de la seconde étape, pour passer de lignes à colonnes. La binarisation s'opère en effectuant la différence entre la somme des trois composantes d'un pixel et 3 fois le seuil considéré. Le calcul de la moyenne ou du maximum en fin de traitement est négligeable en termes de temps de calcul. Par contre le calcul de Di Zeno est relativement important. Les calculs se sont fait en virgule fixe, avec des tables pour le calcul des fonctions trigonométriques et de $1/x$. 20 cycles sont nécessaire pour traiter un pixel, soit une trentaine d'opérations arithmétiques ou logiques.

	N & B	Couleur + moy. , max.	Couleur + Di Zeno
SOBEL	2 ms	4 ms	7.3 ms
DRF	2.3 ms	3.8 ms	7.1 ms

figure 3: temps d'exécution de la detection de contours

Il est donc possible d'extraire de l'information (des contours par exemple) d'images couleur, en temps réel, sur des architectures programmables. La richesse de ces contours doit permettre d'améliorer les phases suivantes de reconnaissance et d'interprétation. Ces contours d'image couleur pourront être utilisés avec intérêt par exemple dans le calcul de la transformée de HOUGH [8], et dans le cadre d'applications de stéréo-vision,

pour améliorer la mise en correspondance des régions de l'image.

Les évaluations de ces algorithmes ont été effectuées sur un simulateur de la maquette de GFLOPS qui est actuellement en cours de réalisation. Un circuit intégré contenant 1 processeur a été mis au point pour cette maquette [9]. Il sera disponible courant septembre 1993. Il est réalisé en technologie CMOS 1.0 micron par ES2. Les algorithmes étudiés ont été programmés en assembleur GFLOPS, puis exécutés sur le simulateur avec des images de taille 256*256 pixels. La configuration simulée est de 16 processeurs pour une performance brute de 2400 MIPS. Le temps de traitement obtenu pour une extraction de contour est de 2 à 8 ms pour des images couleur. Ce temps d'exécution est directement inversement proportionnel au nombre de processeurs. Par exemple il est possible d'effectuer un SOBEL sur une image couleur à la fréquence trame avec 1 seul processeur. Cette architecture étant extensible linéairement, il est possible d'obtenir une configuration ayant autant de processeurs que de pixels dans une ligne de l'image.

4. REMERCIEMENTS

Je tiens à remercier particulièrement le PRC-GDR ANM ("Architectures Nouvelles de Machines") pour le financement du projet GFLOPS durant ces 3 dernières années.

5. CONCLUSION

Nous avons pu montrer la faisabilité d'une approche programmable pour le traitement d'image couleur dans le cas de la détection de contour. L'architecture utilisée est une architecture générale pour le Traitement d'Image. Une configuration de base à 16 processeurs permet d'obtenir des temps de détection de contours inférieur à 10 ms. Une maquette de cette architecture constituée de 16 VLSIs et de composants programmables XILINX [10] est en cours de réalisation. Plusieurs applications utilisant de la détection de contours



comme la stéréo-vision binoculaire sont à l'étude. Grâce à ce type d'architecture, il sera possible d'effectuer en temps-réel une application complète de Traitement d'Image.

BIBLIOGRAPHIE

- [1] J.L. BASILLE, "SYMPATI'S PROJECT," IEEE Int. Conf. on Syst. Eng., Kobe, Japan, Sept. 1992, pp. 515-518.
- [2] D. Juvin, J.L. Basille, H. Essafi, J.Y. Latil, "Sympati2, a 1.5D Processor array for image applications" in EUSIPCO Signal processing IV: Theories and applications, North-Holland, 1988, pp 311-314.
- [3] D. HOUZET, J.L. BASILLE, J.Y. LATIL, "GFLOPS: a General Flexible Linearly Organized Parallel Structure for Images," IEEE ASAP Int. Conf., Barcelona, Sept. 1991, pp. 431-444.
- [4] D. HOUZET, "PROJET GFLOPS" Rapport technique N°IRIT/92-62-R
- [5] S. DI ZENZO, "A note on the gradient of multi-image," Comput. Graphics Image Processing 33, 1986, pp. 116-125.
- [6] G. HOUZET, "Extraction de contours chromatiques d'un signal image couleur", rapport de DEA, ENSEEIHT Toulouse, 1988.
- [7] J. SHEN, S. CASTAN, " ISEF: A General Description of Linear Derivative Operators for Edge Detection," ICCV'90, Osaka, Japon, Dec. 1990.
- [8] D. HOUZET, "Video Rate Hough Transform Implementation on the SIMD/MIMD Parallel Architecture GFLOPS," IEEE SMC'93 Int. Conf., Le Touquet France, Oct. 1993.
- [9] D. HOUZET, "VLSI Implementation of the Real-Time Image Processing Parallel Architecture GFLOPS," IEEE Workshop on VLSI Signal Processing, Napa, Californie, Oct. 1992, pp. 166-174.
- [10] XILINX, "The Programmable Gate Array Design Handbook," Edition 1992.