

PARALLELISATION D'UN ALGORITHME DE CODAGE D'IMAGES A BAS DEBIT SUR RESEAU DE TRANSPUTERS

Mactar SECK*, Arnaud LAPREVOTE* et André TOSSER-ROUSSEY**

* TDF-C2R, 1 Rue Marconi, 57070 Technopole 2000 Metz

** Laboratoire de Mécatronique Industrielle ENIM, 57000 Metz

RÉSUMÉ

Dans cet article, après avoir montré les différents formalismes et les mécanismes du parallélisme, nous traiterons de son intérêt et de sa nécessité en traitement d'image.

Ensuite, différentes techniques de parallélisation sur un algorithme de codage d'images à bas débit seront mises au point, dans la perspective d'une implémentation à faible coût sur un réseau de transputers.

1. INTRODUCTION

Le parallélisme permet de concevoir des systèmes utilisant plusieurs processeurs spécialisés, travaillant de façon autonome mais pouvant échanger des informations avec les autres processeurs.

Le parallélisme est devenu indispensable au développement de l'informatique, car les besoins en calcul numérique ne cessent de croître et de nombreux traitements deviennent de plus en plus complexes et très consommateurs en temps de calcul.

Le traitement d'images est sans aucun doute un domaine privilégié pour l'étude et la mise en oeuvre du parallélisme. En effet les algorithmes de traitement d'images nécessitent une puissance de calcul justifiant l'utilisation de machines parallèles.

Tout système de transmission d'images doit être en mesure d'acquérir, de traiter, de transmettre et de visualiser des séquences d'images numérisées dans un temps limité.

ABSTRACT

In this paper, a transputer-based parallel hardware architecture is presented. This has been designed for an image processing application algorithm. However, several and different parallelization techniques have also been studied for a reduced bit rate coding image. Our purpose is to achieve a low-cost transputer system for image coding.

Les contraintes temps réel varient avec le domaine d'application. Pour un service de télévision 25 images par seconde sont nécessaires, par contre pour le visiophone ou la télésurveillance quelques images par seconde suffisent (typiquement entre 10 et 15).

Parmi le vaste champ des techniques du parallélisme, c'est la parallélisation sur machine à mémoire distribuée qui a été retenue. Le processeur de base en est le transputer. Le transputer est le premier microcalculateur spécialement destiné aux applications parallèles et son architecture a été optimisée pour l'exécution de processus concurrents soit au sein d'un même transputer, soit sur plusieurs d'entre eux.

2. LE PARALLELISME

Le parallélisme est caractérisé par l'emploi de plusieurs unités de traitement distinctes qui accomplissent en concurrence le même travail ou des travaux différents. Afin de disposer du



maximum d'avantage du parallélisme, pour chaque application il faut tenir compte de la quantité des données qu'elle contient et de la manière dont les traitements sont organisés. Ainsi nous distinguons trois sortes de parallélisme:

- le parallélisme interne,
- le parallélisme vrai,
- le pipelining.

Le **parallélisme interne** est caractérisé par la présence au sein d'un même processeur de plusieurs opérateurs mis en parallèle ou de plusieurs unités de traitement de l'instruction dupliquée.

Le **parallélisme vrai** ou parallélisme spatial nécessite que les traitements soient indépendants les uns des autres pour pouvoir s'exécuter simultanément sur des unités distinctes. Les unités peuvent échanger des données et regrouper les résultats.

Le **pipelining** ou parallélisme temporel est une technique d'accroissement des performances d'un ordinateur par la division d'une instruction ou d'une opération en plusieurs étapes.

2.1 Techniques de parallélisation

Parmi les applications nécessitant une architecture parallèle, il faut distinguer

- celles où l'on cherche à augmenter la vitesse de traitement afin d'obtenir des temps de réponse aussi courts que possible,
- des applications temps réel, pour lesquelles la durée d'exécution est une contrainte impérative.

Plusieurs techniques de parallélisation existent, mais nous n'en retiendrons que deux:

parallélisation par les données

Les données sont équitablement réparties entre les différents processeurs du réseau. Chaque processeur doit effectuer tous les calculs sur ses propres données.

parallélisation par les calculs

Un sous ensemble du domaine des calculs est affecté à chaque processeur. Les tâches de calcul peuvent utiliser des données situées sur d'autres processeurs par copie.

2.2 Unités de mesure:

Facteur d'accélération :

Soit un problème traité sur un seul processeur en un temps T_{seq} . Ce même problème est traité en parallèle sur p processeurs en un temps T_{par} .

L'accélération $A(p)$ peut être définie de la manière suivante:

$$A(p) = \frac{T_{seq}}{T_{par}}$$

On étudie le facteur d'accélération en faisant varier p .

L'accélération est une indication du gain de temps apporté par la parallélisation.

Facteur d'efficacité

L'efficacité $E(p)$ d'un algorithme parallèle sur p processeurs peut être définie comme suit:

$$E(p) = \frac{A(p)}{p}$$

L'efficacité est une indication du coût de la parallélisation en terme de processeurs.

Il est proche de 1 lorsque les p processeurs sont utilisés de manière optimale. Mais il est ralenti par certains facteurs:

- la partie purement séquentielle de l'algorithme,
- les communications entre les processeurs,
- les temps d'attente.

3. PRESENTATION DE L'ALGORITHME

Une image est caractérisée par sa redondance, son stockage nécessite beaucoup de place mémoire et sa transmission nécessite beaucoup de temps.

Pour réduire l'information transmise par l'intermédiaire d'un canal de transmission d'images numérisées, il est nécessaire de compresser les images transmises. La réduction de l'information peut affecter la qualité de

l'image. Mais de puissantes méthodes de compression d'images numérisées qui n'affectent pas trop la qualité de l'image ont été développées.

Les méthodes de compression se divisent en deux catégories: les méthodes spatiales et les méthodes par transformation. Les méthodes spatiales agissent directement sur les pixels de l'image pour les coder convenablement; tandis que dans les méthodes par transformation, on ne code pas une image mais les coefficients numériques de sa transformée .

C'est dans ce cadre qu'un algorithme de codage d'images à bas débit a été mis au point à TDF-C2R [BAI-92]. Cet algorithme est fondé sur la reconnaissance et la reconstitution des évolutions des luminances dans une image. Cet algorithme est basé sur la TCD et la quantification vectorielle dans le domaine transformé.

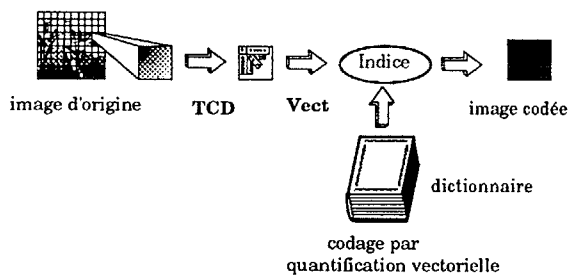


fig 1: schéma de l'agorithme de codage d'image .

L'image est découpée en blocs de 8*8 pixels, et la TCD est appliquée sur chaque bloc.

La **TCD** est une transformation orthogonale bidimensionnelle qui permet de concentrer l'énergie des blocs sur certains coefficients prépondérants dont le premier élément est proportionnel à la moyenne des intensités. Le reste des coefficients fournit une information de type fréquentiel. Un codage avec prédiction différentielle est alors appliqué à la moyenne.

Ensuite on effectue une **classification** du bloc coefficients. L'activité visuelle du bloc de coefficients (sans la moyenne) est calculée pour déterminer son type visuel (homogène, horizontal, vertical,

symétrique,...). A partir de ce type visuel (ou classe) la vectorisation est effectuée. La **vectorisation** consiste à extraire les vecteurs de coefficients les plus pertinents. Ces vecteurs qui sont au nombre d'un, de deux, de trois, de quatre ou de cinq suivant la classe du bloc, sont alors quantifiés avec des dictionnaires adaptés. La quantification vectorielle consiste à représenter un ensemble de pixels appelés vecteur par un représentant (règle du plus proche voisin) choisi dans un dictionnaire. Ce dictionnaire, ainsi composé des éléments les plus représentatifs de l'image, est transmis au décodeur. Une fois la recherche terminée, seul l'index du vecteur du dictionnaire est transmis permettant ainsi un fort taux de compression.

Chaque bloc sera alors codé par sa moyenne, son type, et les codes de ses vecteurs

4 PARALLELISATION DE L'ALGORITHME

Des simulations ont été faites sur une image de 256*256 pixels.

Cet algorithme a été porté sur un réseau de 5 transputers T800 Inmos. Des méthodes de parallélisation par les données ont été utilisées avec des configuration différentes.

4.1 le Farming

Un maître "fermier" distribue à ses esclaves les données sur lesquelles ils appliquent le traitement avant de rendre les résultats au maître. Le nombre de liens limité du transputer nous a conduit à adopter une structure d'arbre binaire.

Avantages:

- taux d'activité des transputers optimal,
- temps d'attente quasiment nul.

Inconvénients:

- taux de communications très élevé,



- impossibilité de transfert direct entre tâches esclaves et maître,
- phénomène de "deadlock" possible,
- tri des blocs résultats obligatoire.

4.2 Le SPMD (Single Program Multiple Data)

Chaque transputer effectue le même traitement sur une partie de l'image. Ce traitement est effectué de manière asynchrone. Mais les transputers se synchronisent après avoir effectué une certaine tâche car, ils peuvent communiquer entre eux s'ils doivent s'échanger des données ou des résultats.

Etant donné que le transputer n'a que quatre liens de communication, une structure en anneau a été retenue.

Avantages:

- bonne performance pour le système de communications,
- exploitation du parallélisme très facile,
- réorganisation automatique des données.

Inconvénients:

- le découpage des données n'est jamais optimal car équilibrer les processeurs en terme de pixels ne signifie pas forcément équilibrer les charges de travail, par exemple le traitement d'un bloc uniforme est moins coûteux en temps de calcul que celui d'un bloc d'une autre classe.
- temps d'attente dans le réseau.

5 CONCLUSION

Cette étude qui est un point d'intersection entre le traitement d'images et le parallélisme montre que:

- le traitement d'images est un domaine privilégié pour la mise en oeuvre du parallélisme,
- le parallélisme apporte des solutions efficaces au traitement d'images.

Au niveau architectural, le SPMD apporte des résultats significatifs mais son efficacité est surtout liée à la nature de l'algorithme. Dans notre cas, elle fut limitée par l'absence

d'équilibrage de la charge de travail des différents processeurs. Tandis que le Farming n'apporte pas de résultats très satisfaisants notamment lorsque le nombre de processeurs augmente. Un problème particulier au traitement d'images est la reconstruction de l'image résultats. En effet les résultats n'arrivent pas toujours dans l'ordre de départ et imposent un travail supplémentaire au transputer de gestion.

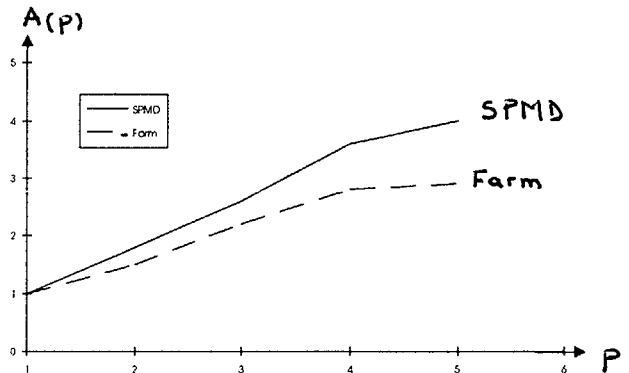


fig 2: Gain en fonction du nombre de transputers

BIBLIOGRAPHIE

- [AHM-74] N.Ahmed, T. Natarajan, K.R. Rao, " Discrete Cosine Transform ", IEEE Trans. on Comm. ,Vol COM.-22, N°1, pp: 90-93, January 1974.
- [PRE-82] Kendall Preston Jr and Leonard Uhr, "Multicomputers and image processing - Algorithms and programs", Academic Press Inc., London, 1982.
- [BAS-85] Jean Luc Basille, "Structures parallèles et traitement d'image", Thèse d'Etat, Université Paul Sabatier de Toulouse, 1985.
- [PIR-90] Alain Pirson, "Conception et simulation d'architectures parallèles et distribuées pour le traitement d'image", Thèse, Université de Grenoble, Mai 1990.
- [BAI-92] Jamal Baina "Codage hybride adaptatif d'images en vue de la transmission à bas débit" Thèse, Université de Metz, Janvier 1992.