



## GENERATION D'IMAGES PAR LANCE DE RAYONS SUR UN CALCULATEUR PARALLELE

FORGUE Marie-Claire,\*GIRAUDON Gérard,BOERI Fernand,AUGUIN Michel

Laboratoire des Signaux et Systèmes, 41.boulevard Napoleon III, 06041 NICE Cedex  
\* Laboratoire PASTIS, INRIA, Sophia Antipolis, 06565 VALBONNE Cedex

Pour des problèmes complexes et coûteux en temps calcul, comme la génération d'images de synthèse par lancé de rayons, il est nécessaire de considérer des machines parallèles pour obtenir des temps de réponse acceptables. Nous montrons dans cet article qu'une utilisation conjointe des deux modes de fonctionnement SIMD (Single Instruction Multiple Data) et SPMD (Single Program Multiple Data) de la machine OPSILA, a permis d'élaborer un schéma de parallélisation simple de cet algorithme. En effet, en mode SPMD, les processeurs exécutent indépendamment l'algorithme séquentiel sur une partie de l'image. Les échanges sont effectués par affectation vectorielle indirecte en mode SIMD.

For complex and time consuming problems, like computer generated images held by ray-casting, it's necessary to take account of parallel machines. We show, in this paper that using both the two operating modes of the OPSILA machine, that are SIMD mode (Single Instruction Multiple Data) and SPMD mode (Single Program Multiple Data), allow us to elaborate a simple parallelization scheme of this algorithm. In SPMD mode, the processors independently execute the sequential algorithm on a part of the image. Exchanges are performed in SIMD mode by indirect accesses to vector elements.

### 1. Présentation

L'objectif principal des travaux effectués au LASSY, en ce qui concerne l'opération " calculateurs parallèles ", est de montrer qu'une machine multiprocesseurs, reconfigurable dynamiquement suivant les deux modes de fonctionnement SIMD (Single Instruction Multiple Data) et SPMD (Single Program Multiple Data), permet de traiter efficacement une grande variété d'applications par utilisation du parallélisme qui leur est associé. En effet, le mode SIMD est particulièrement bien adapté aux calculs vectoriels, et le mode SPMD, en alternance avec l'autre mode, permet l'exécution d'algorithmes non vectoriels mais toutefois parallélisables. Dans le paragraphe suivant, nous rappelons le principe de l'algorithme du lancé de rayons. Puis, nous présentons la machine OPSILA et ses modes de fonctionnement. Enfin, nous proposons un schéma de parallélisation qui tient compte des spécificités matérielles et logicielles de la machine OPSILA.

### 2. Le lancé de rayons

Demier né des algorithmes de synthèse d'images, le lancé de rayons permet la génération d'images de synthèse très réalistes, car il modélise assez bien les phénomènes optiques comme la transparence et la spécularité. Le prix de cette sophistication s'en ressent dans les coûts calcul puisqu'en le comparant à d'autres techniques de synthèse, cet algorithme est 2 à 3 fois plus lent [KAJ83].

L'image d'une scène projetée est reconstruite en analysant le parcours des rayons lumineux parvenant sur l'écran : il suffit de suivre le trajet inverse des rayons lumineux en partant du point d'observation et en passant par chaque pixel de l'image (figure 1). Quand un rayon frappe un objet, de nouveaux rayons peuvent être créés, du fait de la réflexion, de la transmission, et de l'existence d'une ou plusieurs sources lumineuses [WHI80]. Ainsi, le processus est arborescent à partir de l'oeil de l'observateur, les noeuds représentent les points d'intersection rayon-surface (objet

et les feuilles simulent les rayons quittant l'espace de la scène.

### 3. OPSILA

Ce démonstrateur de calculateur parallèle a été conçu et réalisé au LASSY en liaison avec la DRET et la société CIMSA-SINTRA [AUG85].

Le synoptique d'OPSILA est illustré sur la figure 2. L'unité de commande (UC), composée d'un processeur scalaire (PS) et d'un processeur d'instruction (PI), effectue toutes les opérations scalaires, et gère l'ensemble de la machine et notamment la partie vectorielle pour qu'elle traite des vecteurs de taille quelconque. La partie vectorielle est composée de 16 processeurs élémentaires (PE(s)). Cette partie est classique d'une architecture SIMD [HOC81] puisque la mémoire vectorielle est composée de seize bancs de mémoire (BM(s)) et qu'il existe un réseau d'interconnexion (RI) permettant les échanges d'informations de BM à BM ou entre BM et PE.

#### Mode SIMD

Ce mode est particulièrement bien adapté aux calculs vectoriels. L'UC envoie la même instruction à tous les PE(s). L'échange de données peut être effectué dans ce mode grâce au réseau d'interconnexion. On est en exécution synchrone.

#### Mode SPMD

Chaque PE exécute le même programme sur des données différentes indépendamment des autres processeurs. Le passage du mode SIMD au mode SPMD s'effectue en un cycle de la machine et le retour est effectif un cycle après la fin d'exécution du PE dont la charge de travail est la plus importante. Le réseau d'interconnexion est non utilisable dans ce mode. On est en exécution asynchrone.

### 4. Parallélisation du lancé de rayons

La manière la plus naturelle de paralléliser le lancé de rayons consiste à attribuer un arbre de rayons par processeur, c.a.d. un pixel par PE : on engendre alors des architectures typiquement



MIMD (Multiple Instruction Multiple Data). L'inconvénient majeur de ce type d'architecture est la duplication des données de toute la scène dans les mémoires propres de tous les processeurs : l'algorithme de lancé de rayons doit être capable de traiter des centaines voire des milliers d'objets.

Ainsi, en admettant un partage équitable de l'espace scénique pour tous les processeurs, suivant une subdivision spatiale régulière (une région par PE : figure 3), le problème d'optimisation reste le partage équitable des données de la scène dans les bancs de mémoire de manière à ne pas défavoriser la charge de travail d'un ou de plusieurs processeurs : optimisation du taux d'occupation des PE(s).

En réponse à cette optimisation, nous admettons une équité-répartition de tous les objets de la scène entre tous les PE(s). Un rayon primaire ou secondaire pénétrant dans une région ne coupe que les objets contenus dans le banc de mémoire associé à cette région.

Nous avons élaboré un schéma de parallélisation en exploitant le concept de volumes englobants. Cette méthode de réduction de calculs a été introduite par Weghorst et all [WEG84]. En effet, sachant que l'origine et la direction d'un rayon secondaire sont arbitraires, tous les objets (et donc toutes les facettes les composant) doivent être pris en compte. Il est alors commode d'englober les objets par des volumes sphériques ou parallélépipédiques. Si le rayon coupe un de ces volumes, l'objet correspondant sera considéré. En ce qui concerne les rayons primaires, les volumes englobants des objets sont projetés sur le plan écran et le rayon ne sera lancé que si le pixel lui appartenant est bien contenu dans la surface projetée. Pour tous les objets de la scène, on associe des parallélépipèdes pour les rayons primaires et des sphères pour les rayons secondaires.

Le moyen de communication entre PE(s) se fait par descripteurs de rayons (voir ci-après).

On s'attache à rendre les procédures dans les modes SPMD et SIMD les plus longues possibles de manière à réduire le nombre de passages d'un mode à l'autre et, par conséquent, réduire globalement l'attente des processeurs en fin d'exécution de leur programme SPMD. Les phases SIMD assurent le passage des descripteurs de rayons, créés par les phases SPMD, entre les PE(s).

#### 4.1. Paquets de rayons

Chaque rayon est représenté par un paquet, appelé descripteur de rayon, lui-même représenté par une structure composée de six champs (figure 4) :

- le premier contient les informations associées à la racine de l'arbre, c'est à dire, le pixel (x,y) situé dans l'un des PE(s).
- le second contient le numéro du niveau de génération en profondeur de l'arbre.
- le troisième stocke l'intensité directe provenant des sources lumineuses.
- le quatrième champ décrit la direction du rayon.
- le cinquième contient la valeur d'un des coefficients photométriques multiplicateur ( $k_r$  ou  $k_t$ ) correspondant à la surface d'où part le rayon (d'après le modèle d'illumination).
- enfin, le dernier champ permet d'accéder au numéro du PE qui recevra ce descripteur.

#### 4.2. Schéma de parallélisation

On crée des boîtes sur les objets et sur les facettes projetés, telles que leurs dimensions correspondent aux coordonnées extrêmes de l'objet ou de la facette. Pour les rayons primaires, on évite le passage en revue de toutes les facettes de la scène en testant pour chaque pixel son appartenance à une ou plusieurs

boîtes-objets englobantes projetées. Si le test est vérifié, on cherche à combien de facettes il appartient grâce aux boîtes-facettes englobantes elles aussi projetées. Donc pour chaque rayon primaire lancé, on calcule très peu d'intersections. Cette première phase SIMD correspond à une phase de prétraitement des données. Les autres phases SIMD assurent le passage des rayons à traiter entre les PE(s).

Les PE(s) connaissent l'existence des autres objets de la scène par la connaissance des équations de leurs sphères englobantes. La phase SPMD correspond aux lancés des rayons et aux calculs d'intensité aux points d'intersection retenus ou bien à la détermination des sphères englobantes interceptées appartenant aux autres PE(s). Ce schéma se traduit par le squelette algorithmique suivant :

Début

*phase SIMD*

*pour i = 1 à N par pas de N/16*

*pour j = 1 à N*

on considère la tranche de pixels [ (i,j) , (i+1,j) , ..., (i+15,j) ]  
tests successifs d'appartenance aux boîtes-objets, puis aux boîtes-facettes

création de couples de labels (o,f) , respectivement label de l'objet et label de la facette.

*fin\_pour j*

*fin\_pour i*

Après traitement de tous les pixels de l'image, on obtient une matrice de labels couplés (principe de l'item buffer [WEG84]). A partir de cette matrice, on construit des listes de paquets descripteurs de rayons primaires.

*pour k = 1 à profondeur de l'arbre maximum autorisée*

*phase SPMD*

- calculs d'intersection rayon-facette
- calculs d'intensité directe au point d'impact trouvé
- calcul des directions des rayons réfléchis (R) et/ou transmis (T)
- calculs d'intersection de toutes les sphères englobantes avec les rayons R et T successivement
- construction des paquets descripteurs de rayons pour R et/ou T tel que leurs champs "PE destination" contiennent les sphères englobantes coupées.

*phase SIMD*

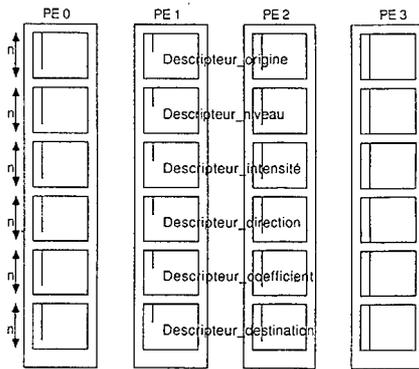
La phase d'échange des paquets de rayons entre les PE(s) est effectuée grâce à la primitive *scatter* définie ci-dessous.

*fin\_pour k*

Fin

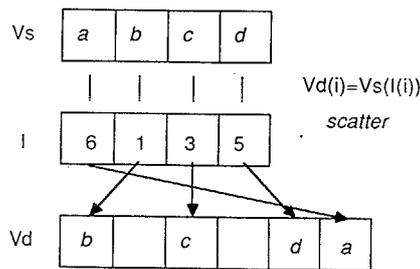
#### 4.3. Procédure d'échanges

A la fin de chaque phase SPMD, chacun des 16 PE(s) a généré au maximum n paquets de rayons. Ils sont rangés dans la mémoire vectorielle sous la forme de six tableaux de n éléments dans chaque banc de mémoire (figure ci-dessous) .



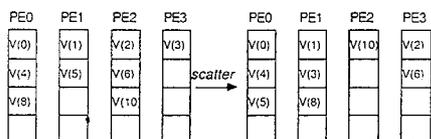
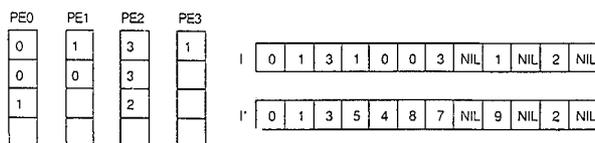
Vu du mode SIMD, chaque tableau peut être considéré comme un vecteur de taille 16\*n dont les composantes sont consécutives en mémoire vectorielle. On assimile l'ensemble des descripteurs de rayons à 6 vecteurs correspondant chacun à un des champs. Nous utilisons la primitive d'OPSILA *scatter* [SCH87], qui réalise des accès mémoire indirects suivant un vecteur d'indice. Dans la définition qui va suivre, Vs désigne le vecteur source, Vd le vecteur destination et I un vecteur d'indice injectif.

instruction *scatter*(Vd,Vs,I) :  $Vd(I(i)) = Vs(i)$  oppA i



Pour notre application, le vecteur d'indice I est le vecteur associé au champ "Descripteur\_destination". Une fois le vecteur d'indice initialisé, on applique le *scatter* aux cinq autres champs des descripteurs de rayons. Le facteur d'accélération du *scatter* (temps séquentiel sur temps vectoriel) est donné en fonction de la longueur du vecteur d'indice (maximum de 32767 composantes) sur la figure 6.

On doit cependant modifier I pour le rendre injectif sous peine de recouvrement des informations. Ce problème est illustré ci-dessous pour 4 PE(s). Le vecteur modifié est noté I'.



On remarque que le vecteur d'indice comporte des 'trous', car les PE(s) peuvent ne pas générer le même nombre de rayons à un même niveau de profondeur des arbres de rayons. Cela ne présente pas de problème car le *scatter* ne traite pas les valeurs référencées par NIL.

### 5. Conclusion

Des applications variées, déjà étudiées ou en cours d'étude, concernent la physique des particules [MAI85], les algorithmes de traitement d'images [DUC87] et la méthode des éléments finis [DAL87]. Leurs mises en oeuvre respectives nous permettent d'affirmer que l'utilisation du mode mixte SIMD/SPMD est efficace à condition de choisir judicieusement les sous-problèmes confiés aux processeurs élémentaires. Pour ce qui nous concerne, nous avons pris comme domaine d'application la synthèse d'images et en particulier l'algorithme de lancé de rayons qui est la technique la plus coûteuse pour la réalisation d'images très réalistes. Ce domaine est divisé en 16 sous-domaines associés aux 16 processeurs. L'utilisation efficace des deux modes SIMD et SPMD de la machine est obtenue de la manière suivante :

- Le mode SIMD est utilisé pour les échanges des paquets de rayons aux frontières des sous-domaines, via le réseau d'interconnexion entre les PE(s).
- La phase calculatoire est assurée en mode SPMD.

L'utilisation des deux modes présente l'avantage de ne pas dupliquer la scène dans les bancs de mémoires et réalise ainsi un gain appréciable en capacité de traitement du nombre d'objets qui composent la scène. Il est clair qu'aucun contrôle pour l'instant n'est effectué sur la répartition de la charge de travail des différents processeurs. Cependant pour une scène comportant plusieurs centaines d'objets équitablement répartis dans l'espace, la répartition de la charge de travail est assurée par le volume des données à traiter. Les transferts d'informations nécessaires lors des phases de synchronisation sont réalisés de manière efficace par l'utilisation de l'instruction *scatter*.

La programmation de l'algorithme du lancé de rayons mise en oeuvre dans cet exposé est en cours de réalisation en langage HELLENA.

### Références

[AUG85] M.Auguin  
 " Etude et réalisation de structures de calculs parallèles, applications aux méthodes numériques " . Thèse d'Etat, Université de Nice, 1985

[DAL87] J.P.Dalban  
 " Partitionnement de la méthode des éléments finis : réalisation SIMD/SPMD " . Thèse de 3<sup>ème</sup> cycle, Université de Nice, Mai 1987

[DUC87] P.Duclos  
 " Mise en oeuvre d'algorithmes de transformations géométriques sur une architecture parallèle de type SIMD/SPMD " . Onzième Colloque sur le Traitement du Signal et ses Applications, Juin 1987, Nice

[HOC81] R.W.Hockney , C.R.Jesshope  
 " Parallel computers " . Adam Higer, Bristol, 1981

[KAJ83] J.T.Kajiya  
 " Siggraph83 tutorial on ray-tracing "

[LEN78] J.Lenfant  
 " Parallel permutation of data : a BENES network control algorithm for frequently used permutations " . IEEE, T.C, Vol C27, No 7, July 1978, pp637-647

[MAI85] J.Maillard , W.Jalby , M.Auguin  
 " Use of a SIMD-SPMD machine for simulation in particle physics " . Computing in high energy physics, Amsterdam, Juin 1985

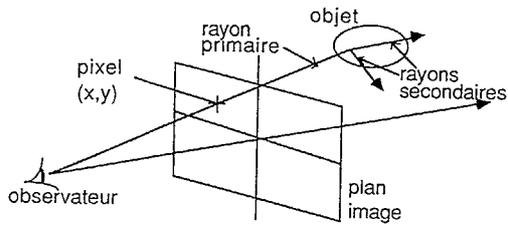


figure 1 : principe du lancé de rayons

origine (x,y) PE i
niveau de génération
intensité directe
direction
coefficient ks ou kt
PE destination

figure 4 : descripteur de rayon

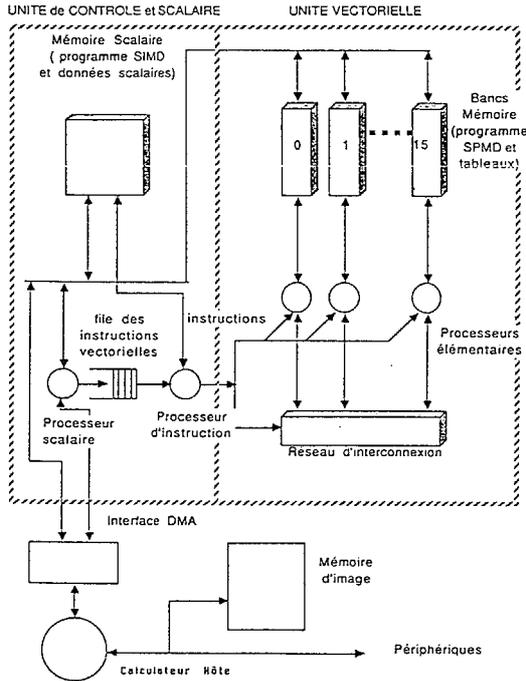


figure 2 : synoptique d'OPSILA

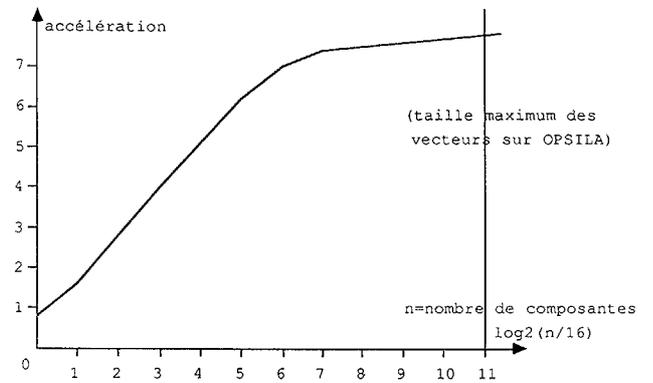


figure 5 : performances du scatter

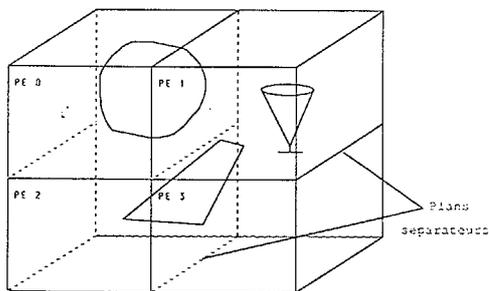


figure 3 : subdivision spatiale pour 4 PE(s)