Implantation temps réel d'un algorithme de segmentation par modélisation de l'architecture du système visuel biologique

F. Devillard, B. Heit et T. Cecchin

Centre de Recherche en Automatique de Nancy - CNRS UPRESA 7039
Université HENRI POINCARE, NANCY I
11, rue de l'université 88100 SAINT-DIE, FRANCE
Francois.Devillard@cran.iutsd.u-nancy.fr, heit@iutnb.u-nancy.fr,
Thierry.Cecchin@cran.iutsd.u-nancy.fr

Résumé — La segmentation d'images en temps réel pose encore à l'heure actuelle des problèmes d'implantations et de performances. Les méthodes utilisant une modélisation du système visuel biologique présentent de bonnes performances mais restent pénalisantes en termes de calculs sur machines séquentielles. Dans cet article, on propose un algorithme utilisant une architecture visuelle biologique pour réaliser une détection de contours. Notre travail a consisté à évaluer, en vue d'une implantation DSP, chaque opérateur en terme de coût de calculs, mémoires et communications. Leur répartition est étudiée pour un environnement multiprocesseurs. La solution matérielle proposée répond aux besoins temps réel nécessaires aux applications embarquées où l'on recherche un opérateur homogène, performant et sans réglage de paramètres de vision s'accommodant aux différentes caractéristiques des images naturelles.

Abstract — At the present time, realtime image segmentation provides us many problems concerned its implementation and its achievements. A good alternative to the classical segmentation methods are those deduced from biological visual systems but they are very long to perform on Von Neuman architecture. Our algorithm uses a biological visual modelisation to perform an edge detection. We evaluate their characteristics as for instance computing time, memory or communication costs at every stage of processing. A scheduling of the main tasks is designed in a multiprocessor system. The obtained calculator would be useful to on-board systems that need efficient and robust algorithms for a large range of images and that without parameters settings.

1 Introduction

1.1 Objectif

Les contours en vision artificielle sont souvent les primitives de base des processus de reconnaissance de formes. Dans une chaîne classique de traitement, la démarche consiste à rechercher parmi l'ensemble des algorithmes existants, l'ordonnancement qui présente les meilleurs résultats au regard des objectifs à atteindre. En utilisant des opérateurs connus et en les modifiant selon les propriétés du modèle biologique, Girod [1] a proposé une chaîne continue, dépourvue de réglages et peu sensible aux caractéristiques des images à traiter. Le modèle visuel biologique présenté peut être considéré comme un ensemble complet et cohérent de traitements permettant d'obtenir, entre autres, les contours et leur orientation. Actuellement, l'algorithme de segmentation est implanté sur PC et programmé en langage C optimisé. Sur un Pentium avec une fréquence d'horloge de 100 MHz, le temps complet d'exécution sans affichage reste encore trop important (1 minute). Un calculateur spécialisé possédant une architecture adaptée à l'algorithme devrait, même cadencé plus lentement, permettre d'obtenir des performances supérieures. L'objectif de notre travail est donc de déterminer une architecture d'implantation adaptée à ce modèle en vue d'une utilisation en temps réel.

1.2 Le modèle biologique

Le modèle repris est le schéma visuel simplifié de Thorpes [2] représenté sur la figure 1. Ce schéma décrit sommairement le cheminement des informations visuelles dans les différentes aires cérébrales humaines.

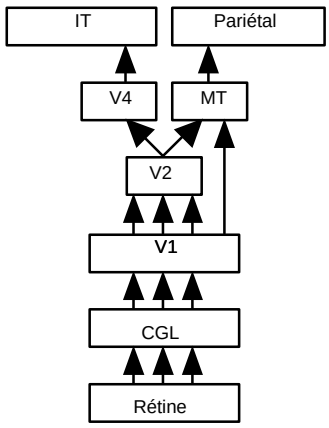

FIG. 1 : Les différentes aires cérébrales selon le modèle visuel biologique de Thorpes

3 canaux d'informations partent de la rétine par le nerf optique et le corps genouillé latéral (CGL) pour gagner l'aire V1. Les informations sont la couleur, la luminance et le mouvement. La liaison entre l'aire V1 et V2 est également multiple, les informations transitant sont maintenant la couleur, les formes et le mouvement. A partir de l'aire V2, ces

mêmes informations modifiées et complétées par la notion de profondeur s'orientent vers 2 régions :

- la région IT (Inférotemporale) dédiée à la reconnaissance de formes en passant par l'aire V4 (couleur).
- la région pariétale centre d'analyse des mouvements et de la situation spatiale en passant au préalable par l'aire MT (mouvement et stéréovision).

Notre algorithme décrit par des opérateurs simples le fonctionnement de la rétine, du corps genouillé latéral et de l'aire V1.

2 Présentation de l'algorithme

Soucieux d'obtenir la meilleure résolution possible, notre modélisation (cf figure 2) concerne exclusivement la partie centrale de la fovéa et nous considérons que la topologie corticale est rectangulaire et homogène. Par ailleurs, dans un but de simplification, nous faisons abstraction de la notion de couleurs, pour ne traiter que l'aspect luminance du signal. L'image capturée par la rétine est notée I.

2.1 Modélisation de la rétine

Le premier élément de l'ensemble proposé simule le fonctionnement de la partie centrale de la rétine. Cette fonction est modélisée par D. Marr et E. Hildreth [3] sous le nom d'opérateur DOG ou "Difference Of Gaussian". L'image résultat X, obtenue à partir de I, est similaire à l'application d'un opérateur laplacien. Cependant, nous réglons l'opérateur DOG en filtre passe-bande large qui autorise le traitement de transitions sur une large gamme de résolutions allant du détail fin au contour flou. Parallèlement à X, une image de luminance W est calculée par un filtrage passe-bas gaussien appliqué sur l'image de la rétine I. Ces images X et W transitent ensuite pour gagner l'élément simulant le corps genouillé latéral et la zone corticale V1.

2.2 Modélisation de l'aire V1

Le modèle proposé pour l'aire V1 simule les caractéristiques de sélectivité à l'orientation des stimuli visuels que présentent certaines cellules du cortex. L'algorithme correspondant comporte deux étapes pour chacunes des images X et W.

La première étape effectue un filtrage directif gaussien des images selon 8 canaux de directions échelonnées au pas de 22.5°. Ce traitement permet de rechercher des alignements de pixels pré-contours dans l'image X au voisinage des contours, et de lisser le signal W de luminance le long de ceux-ci pour atténuer le bruit. X et W filtrées donnent chacune 8 images résultat, respectivement Xd et Wd (d est l'indice de direction variant de 0 à 7).

La seconde étape réalise l'opération de localisation des passages par 0 sur les Xd et le calcul du contraste sur les Wd ; ces opérations s'effectuent orthogonalement à la direction du filtrage précédent. Ainsi, un contour en marche est trouvé au passage par 0 du signal Xd. On obtient alors une image Z'd pour chacune des orientations. Notre approche repose sur l'hypothèse que la voie W subit les mêmes traitements que la voie X. La directivité des images Z'd n'est pas suffisante pour une bonne immunité au bruit, c'est pourquoi nous utilisons les informations contenues dans les images de contraste C'd pour réduire le nombre de faux contours. Pour ce faire, les images Z'd et C'd interagissent par validation mutuelle. Les informations extraites sont appelées pré-contours, car les points détectés par les passages par 0 dans Xd et validés par un contraste cohérent sur Wd, sont des points de contour potentiels de l'image I de la rétine.

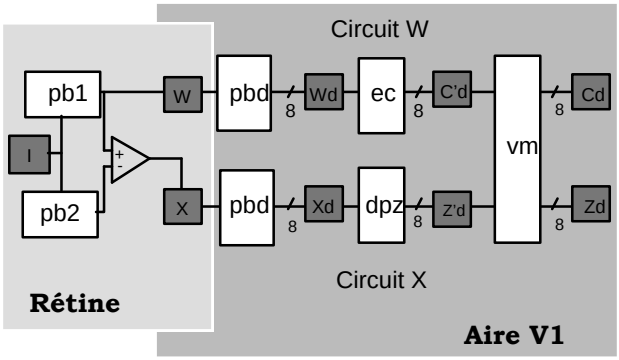

FIG. 2 : L'algorithme biologique adapté pour son implantation. La rétine est composée de filtres passe-bas (pb1 et pb2). L'aire V1 est constituée de banques de passe-bas directionnels (pbd), du calcul du contraste (ec), de la détection des passages par 0 (dpz) et enfin de la validation mutuelle des zéros et des contours (vm).

2.3 Détection des contours finaux

Une méthode de sélection de ces pré-contours en contours finaux est proposée dans les travaux de Girod. Les résultats obtenus démontrent l'efficacité et la souplesse d'utilisation de l'approche biologique comparée aux classiques comme, par exemple, Canny-Deriche [4]. Une illustration des résultats obtenus est montrée en annexe.

3 Implantation sur ADSP 21020

L'algorithme de segmentation possède une architecture qui se prête parfaitement à l'implantation sur un calculateur rapide, possédant à la fois une architecture parallélisée et pipelinée. De plus il ne comporte que des opérations de voisinage sur les pixels comme les convolutions, les détecteurs d'extrema, des passages par 0 …

La difficulté majeure réside dans la quantité de données à calculer, stocker et à manipuler. Cet algorithme dans sa version originale (8 canaux de direction) nécessite environ 90 opérations de voisinage sur les images et le stockage de 60 images. Nous avons donc adopté un compromis consistant à traiter des images de 256 x 256 pixels en 256 niveaux de gris afin d'obtenir un système final compact et à un coût raisonnable. Nous avons évalué le temps d'exécution et les besoins en mémoire de toutes les opérations élémentaires de l'algorithme sur le DSP ADSP21020 de Analog Devices cadencé à 33 MHz. Les modules décrits à la suite sont développés et optimisés en assembleur. La programmation en langage C est une solution envisageable pour superviser l'ensemble du traitement mais l'optimisation en assembleur des opérations itératives reste à l'heure actuelle incontournable pour la rapidité des calculs. En C, des contraintes de compilation ou le choix d'un compilateur (si

possible) améliorent les temps de traitement sensiblement [5] mais sans commune mesure avec les résultats obtenus en assembleur.

3.1 La rétine

L'opérateur DOG constituant la rétine est un filtre passe-bande dont la bande passante est réglée pour couvrir la détection d'une large gamme de contours de périodes spatiales allant de 2 à 5 pixels. La bande-passante de notre filtre étant positionnée dans les fréquences élevées, l'implantation des filtres gaussiens sous la forme FIR peut-être envisagée avec de petits noyaux de convolution. Le filtre gaussien bidimensionnel est séparable en direction ce qui nous permet de le calculer en 2 convolutions monodimensionnelles successives selon 2 axes orthogonaux (horizontal et vertical). La figure ci-dessous donne la structure finale du filtre DOG, X étant sa sortie. On a 2 filtres gaussiens monodimensionnels FIR nommés g1 (1x5) et g2 (1x13) de variance respective σ_1=0.5 et σ_2=1.5.

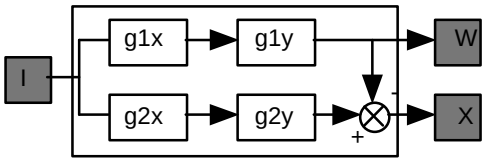

FIG. 3 : Bloc fonctionnel RETINE simulant la rétine.

Le tableau ci-dessous donne les temps et besoins mémoires nécessaires au calcul de la rétine sur un processeur. Les calculs sont menés sur 32 bits en virgule fixe.

TAB. 1 : Caractéristiques du système RETINE intégré avec un processeur ADSP21020-33MHz.

Traitements	Temps (msec)	Mémoires
g1x/y	21.54 msec	**Internes**
g2x/y	37.51 msec	**4*64Kword**
Différence	4.37 msec	**E/S**
RETINE	**122.48 msec**	**3*64Kword**

3.2 L'aire V1

Le corps genouillé latéral et l'aire V1 sont des zones sensibles à l'orientation locale des formes dans les scènes. Le calcul est donc organisé en blocs traitant chacun des directions différentes. Le traitement du canal de direction j est représenté dans la figure ci-dessous. Il est constitué de 2 banques de filtres passe-bas directifs traitant W et X, d'une détection des passages par 0 sur Xi et d'un bloc d'estimation de contraste et de validation mutuelle donnant en sortie Cj et Zj. L'ensemble constitue un bloc fonctionnel V1j parallèlisable autant de fois qu'il y a de canaux de directions. Chaque bloc est détaillé dans les paragraphes qui suivent.

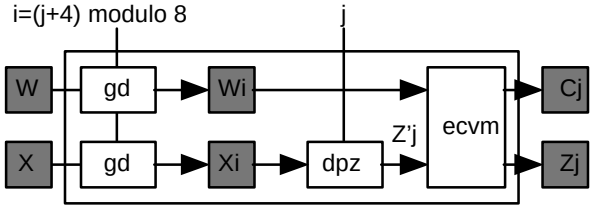

FIG. 4 : Bloc fonctionnel V1j simulant un canal de direction j de l'aire V1.

3.2.1 Les filtres directionnels

Les filtres directionnels constituant l'aire V1 sont du type passe-bas gaussiens monodimensionnels et appliqués perpendiculairement à la direction sélectionnée. La bande-passante d'un filtre est réglée pour que sa sélectivité en direction soit de l'ordre de 22°.5. On remplit cette condition avec un filtre gaussien monodimensionnel FIR nommé gdi (1x25) et de variance σ_d=3.

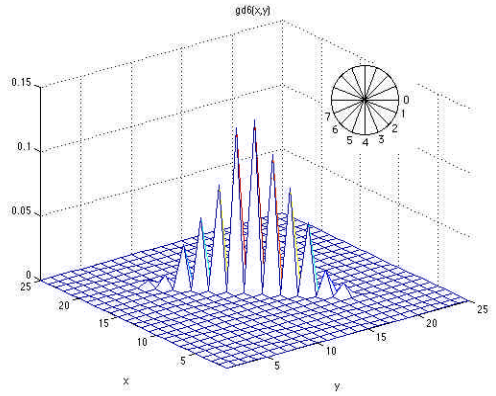

FIG. 5 : Codage des directions et réponse impulsionnelle du filtre gd6 (canal de direction 6 soit 135° ± 11.25°).

3.2.2 Passage au format entier pour Xi

Une étape nécessaire est la fonction f2int. Elle convertit et arrondit en entier les images Xi avant qu'elles ne soient traitées par la détection des passages par 0. Cette conversion, nous permet de gagner sur les prochains traitements en simplifiant les procédures de test sur les pixels dans les boucles.

3.2.3 La détection des passages par 0

La détection des passages par 0 consiste à rechercher dans Xi les changements de signe dans la direction perpendiculaire soit j=(i+4) modulo 8. Quand un passage par 0 est localisé, on recherche l'amplitude des 2 lobes qui l'entourent dans la direction fixée par j. L'amplitude maximale est recherchée à une distance de 4 pixels de part et d'autre du pixel courant. L'extemum calculé constitue un résultat de Z'j.

3.2.4 Le calcul de contraste et validation mutuelle

Les blocs ec et vm présentés figure 2 sont pipelinées dans notre algorithme et regroupés en une seule fonction ecvm.

L'estimation du contraste

Pour chaque pixel, l'estimation du contraste du contour est réalisée sur Wi à la condition où il y a un passage par 0 détecté dans Z'j. L'estimation du contraste est obtenue par la recherche de l'extremum d'amplitude sur une distance de 4 pixels de part et d'autre du pixel courant dans l'image Wi. L'extremum calculé constitue un résultat temporaire de Cj repris ensuite par la validation mutuelle.

La validation mutuelle

vm valide les points contour Cj à partir du résultat temporaire de Cj et de Zj'. La validité du contour est vérifiée pour chaque pixel par la cohérence des signes observés dans ces 2 images. L'image ainsi traitée est filtrée d'un certain

nombre de points contour qui sont des artefacts provenant du bruit au voisinage des contours. Les artefacts provenant des effets de bords des préfiltrages passe-bas sont corrigés par des traitements qui à l'heure actuelle ne sont encore pas implantés.

TAB. 2 : Caractéristiques du système V1j (canal j de V1) intégré avec un processeur ADSP21020-33MHz.

Traitements	Temps (msec)	Mémoires
gd	100.50	Internes
f2int	4.37	6*64Kword
dpz	73.85	
ecvm	69.75	E/S
Canal V1j	**348.97**	4*64Kword

4 Implantation multiprocesseurs

Au vue des temps du système monoprocesseur et des contraintes architecturales de l'algorithme, l'objectif de l'implantation multi-processeurs est de réduire le temps global de calcul de l'aire V1 qui très pénalisant pour obtenir un traitement temps réel. La parallélisation se réalise simplement le schéma ci-dessous illustre une solution à base de 4 processeurs, à titre d'exemple.

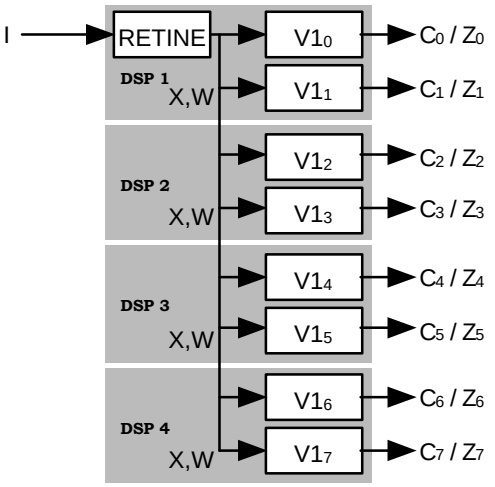

FIG. 6 : Architecture utilisant 4 processeurs pour la modélisation complète.

TAB. 3 : Caractéristiques du système RETINE+V1 intégré avec 2^N processeurs ADSP21020-33MHz.

2^N processeurs	Temps (msec) RETINE/V1	Temps (msec) GLOBAL
N=0	122.48 / 2791.80	**2914.28**
N=1	79.40 / 1395.90	**1475.30**
N=2	79.40 / 697.95	**777.35**
N=3	79.40 / 348.98	**428.37**

L'algorithme se prête bien à la parallélisation. En confiant le calcul de 2 directions à chaque DSP, on obtient une architecture matérielle où le calcul est distribué de façon relativement homogène. Seul le 1er DSP est chargé en plus du calcul de la rétine et de la diffusion du résultat dans les 3 autres calculateurs (DSP2…4). Des temps de transfert peuvent éventuellement s'ajouter au temps de calcul global mais il restent relativement modestes (chargement en DMA de 64Ko réalisable en 4.37 msec [6]). D'autres dispositifs de transfert transparents sur le temps de traitement sont envisageables mais ils sont plus coûteux matériellement.

5 Perpectives et conclusion

Les temps indiqués dans le tableau 3 restent encore importants pour nos objectifs et quelques améliorations sont encore à apporter à ce jour.

Les temps donnés sont obtenus individuellement pour chaque opérateur sur des images 256x256 pixels alors qu'un effet de bord apparaît rapidement dés les premiers filtrages (pour N=0 gain de 590 msec sur les 2914 msec annoncés).

Les opérateurs dtp et ecvm sont encore optimisables, si l'on gère les sorties de boucle prématurées possibles sous certaines conditions (gain variable en fonction de l'image).

En contre-partie, les temps de transferts ne sont pas pris en compte pour les échanges (4.37 msec/64Ko) entre cartes mais ils restent mineurs en comparaison au temps de filtrage directionnel (100 msec).

En conclusion, les améliorations apportées peuvent considérablement réduire le temps de traitement (environ 20%). Si l'on ne change pas de processeur et d'architecture, le problème réside dans le filtre directionnel qui crée un temps de calcul incompressible de 100msec. La nature du filtre mis en oeuvre (FIR) est peut-être à remettre en question dans l'avenir.

6 Annexe

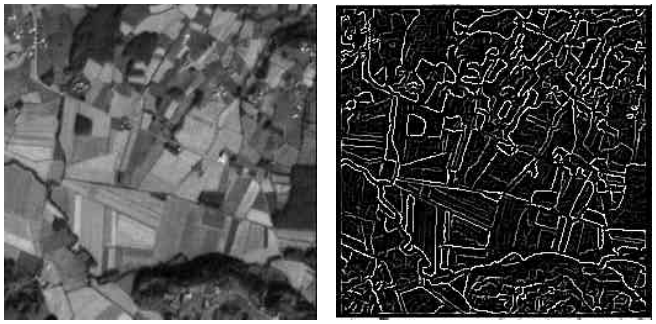

FIG. 7 : Image AQUITAINE (GDR-PRC ISIS) et Image des contours

Références

[1] J.P. Girod. *Extraction de primitives en traitement d'images par modélisation des systèmes visuels biologiques*. Thèse de doctorat de l'Université H. Poincaré, Nancy I. 13 octobre 1994.

[2] S.J. Thorpe. *Traitement d'images chez l'homme*. Techniques et Sciences Informatiques, vol. 7, n°6, p. 517 à 525, 1988.

[3] D. Marr, E. Hildreth. *Theory of edge detection*, Proc of Roy. Soc. of London, n° B207, p. 187 à 217, 1980.

[4] R. Deriche. *Extraction de composants connexes, basée sur une détection optimale des contours*, CESTA, Paris, 1987.

[5] T. Ea, L. Lacassagne, P. Garda. *Exécution temps réel des détecteurs de contours de Deriche par des processeurs RISC.*, congrès AAA 98 Saclay. France. p. 251 à 258, 1998.

[6] Analog Devices. ADSP21010-20 User's Manual, 1993.