

# Optimisation sur ordinateur d'un modèle d'architecture de couche de cellules rétinienne

François DEVILLARD<sup>1</sup>, Frédéric GRANDIDIER<sup>2</sup>, Bernard HEIT<sup>2</sup>

<sup>1</sup>Centre de Recherche en Automatique de Nancy UMR CNRS 7039 (CRAN - CNRS)  
11 rue de l'Université, F-88100 Saint-Dié-des-Vosges

<sup>2</sup>Centre de Recherche en Automatique de Nancy UMR CNRS 7039 (CRAN - CNRS)  
2 rue Jean Lamour, F-54500 Vandœuvre-lès-Nancy Cedex

francois.devillard@iutds.uhp-nancy.fr, bernard.heit@esstin.uhp-nancy.fr

**Résumé** – Cet article traite de l'implémentation sur ordinateur d'une modélisation des couches cellulaires de la rétine biologique vue de son application en vision artificielle. En vision artificielle ces traitements permettent d'extraire dans un flot continu les indices visuels avec une bonne robustesse. Le coût de calcul résultant nécessite par contre des algorithmes adaptés et des machines spécialisées pour une mise en oeuvre performante. La solution proposée utilise une adaptation de l'algorithme pour accélérer le calcul des couches rétiniennes tout en conservant leurs propriétés locales adaptatives.

**Abstract** – This article presents a work about computer implementation of a retinal layer modelisation that is dedicated to machine vision. In computer vision this kind of processing allows to extract visual indices in a continuous flow with a good robustness. To minimize its computing cost, the implementation require well suited algorithms and hardware systems to implement efficient applications. The proposed method uses an adapted algorithm to accelerate the perform of the retinal layer modelisation while preserving its adaptive local properties.

## 1 Introduction

La rétine fovéale est constituée d'un *empilement* de couches de cellules traitant le signal lumineux pour en extraire des indices visuels (contrastes de chrominance statique et dynamique, luminance...). Dans l'objectif de modéliser son fonctionnement, une couche élémentaire de cellules nerveuses rétiniennes a été définie et sert de base à la conception d'une architecture de rétine utilisable en vision artificielle. Cette couche élémentaire est modélisée par des outils de traitement du signal et permet de reproduire une approche comportementale de la couche rétinienne biologique. Elle est composée d'un réseau de cellules de traitement paramétrables localement pour reproduire les mécanismes d'adaptation biologiques observables. L'assemblage des différentes couches de la rétine doit satisfaire à des contraintes d'implémentation temps réel dans le cadre d'applications de vision artificielle; l'algorithme correspondant décrit la structure des couches dans la section 2. La section 3 présente une approche plus efficace sur ordinateurs de la programmation des couches rétiniennes. La démarche des modifications apportées sur l'algorithme est présentée. Le gain observé sur le coût de calcul est évalué à chaque étape. La section 4 donne quelques résultats qualitatifs obtenus avec l'algorithme appliqué sur une séquence d'images d'entrée statique. Enfin la dernière section conclut et donne les perspectives envisageables à court terme pour nos travaux.

## 2 Architecture de la rétine

### 2.1 Modélisation de la PLE

Les mécanismes de la perception visuelle [4] réduits à la couche plexiforme externe (*PLE*) du système parvocellulaire (zone fovéale de la rétine) sont schématisés dans la figure 1. L'acquisition de la scène observée est réalisée par une matrice de photorécepteurs (cellules de type cône) qui transforment l'énergie lumineuse  $I$  en énergie électrochimique  $C$ . Deux autres couches de cellules effectuent à la suite une première analyse du message par le réhaussement des contrastes présents dans l'image. La couche des cellules horizontales est une couche de régularisation estimant localement un référentiel de luminance  $H$  de la scène. La couche des cellules bipolaires différencie la réponse des cellules cône et horizontale pour fournir une réponse *Bip* mettant en évidence les contrastes caractérisant les objets observés. Chaque couche est constituée d'un maillage plus ou moins diffus de cellules interagissant latéralement. Chaque cellule fournit une réponse, fonction d'une somme pondérée et non linéaire de ses entrées synaptiques. De plus, la réponse est une combinaison de filtrages passe-bas spatial et temporel. Par conséquent, chaque cellule peut être modélisée par un processeur ou filtre élémentaire à entrées multiples (les synapses cellulaires) et dont la sortie (l'activité cellulaire) est donnée par un filtre passe-bas spatio-temporel [1]. L'association de filtres élémentaires travaillant en maillage permet de construire une couche de traitement cellulaire aux comportements comparables à celui d'une couche de la rétine.

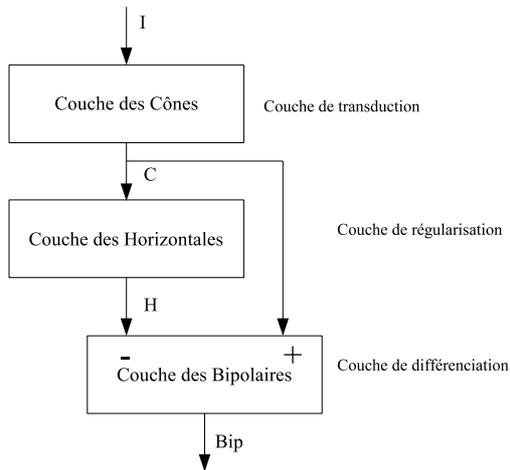


FIG. 1 – Schéma fonctionnel de l'architecture de la couche PLE de la rétine biologique photopique (vision diurne).

## 2.2 Modélisation des couches rétiniennes

La mise en oeuvre de notre solution originale repose sur un filtrage spatio-temporel numérique distribué et adaptatif. La fonction de sortie de la cellule  $s_{x,k}$  est donnée dans l'équation 1 (pour une seule dimension spatiale  $x$ ). Elle traduit l'activité cellulaire pour une entrée  $e_{x,k}$  à la position  $x$  et à l'instant  $k$ . Les différentes couches de cellules présentées figure 1 utilisent la même équation avec des paramètres différents. Les périodes d'échantillonnage spatiale et temporelle sont unitaires pour simplifier la formulation.

$$s_{x,k} = (1 - r_{x,k}) \cdot \left[ b_{x,k} \sum_{i=-\lambda}^{\lambda} \frac{C_{i+\lambda}^{2\lambda}}{4^\lambda} \cdot e_{x+i,k} \right] + r_{x,k} \cdot s_{x,k-1} \quad (1)$$

Avec  $\lambda$  représentant  $n_{x,k}$  et  $0 \leq r_{x,k} < 1$ .

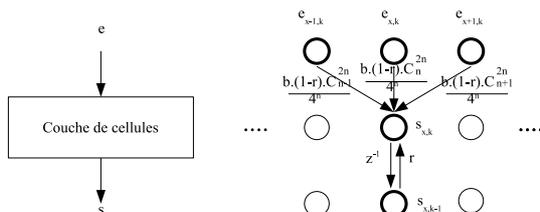


FIG. 2 – Graphe de fluence du traitement du processeur élémentaire équivalent à une cellule à synapses positionnée en  $x$  à l'instant  $k$ . Seules sont représentées les connexions avec le proche voisinage cellulaire. L'opérateur  $z^{-1}$  représente un retard pur et  $C$  la fonction statistique de combinaison.

Ce filtre linéaire et séparable est une association de structures de type *RIF* (Réponse Impulsionnelle Finie) et *RII* (Réponse Impulsionnelle Infinie) appliquées respectivement dans les domaines spatial et temporel. Les paramètres influençant la réponse spatiale de la cellule sont  $n$  et  $b$  (respectivement le couplage latéral entre cellules et le facteur de gain de cellule). Le coefficient  $b$  agit sur l'excitation de la cellule  $x$  comme un gain et le coefficient  $n$  traduit l'influence de la cellule  $x$  sur ses voisines. Le pa-

ramètre  $n$  d'étalement fixe le diamètre de l'arborescence dendritique à  $2n + 1$ .

Ces coefficients agissent différemment sur la réponse spatiale et temporelle de la couche. L'intensité de l'influence de  $n$ , en fonction de la distance, est obtenue à l'aide d'une fonction binomiale de manière à fournir une réponse impulsionnelle passe-bas, centrée et pseudo gaussienne [5]. Le paramètre influençant la réponse temporelle de la cellule est  $r$ . Il permet de reproduire l'effet intégrateur d'une constante de temps sur la sortie de la cellule (due à la résistance et à la capacité membranaire de la cellule) [6]. La figure 2 représente l'équation 1 graphiquement sous la forme d'un maillage. La couche de cellules cône prend en entrée la succession d'images formant la séquence vidéo (1 pixel = 1 cône). Les cônes dans notre version d'algorithme sont considérés achromatiques.

## 3 Mise en oeuvre algorithmique

La mise en oeuvre de l'équation 1 nécessite une adaptation algorithmique et/ou architecturale pour obtenir une rétine minimisant les temps de calcul. En effet, la composante spatiale du filtrage est la principale consommatrice de ressources de calcul. Les deux paragraphes qui suivent décrivent la décomposition du filtrage *RIF* spatial en une cascade de filtres élémentaires. Cette décomposition est réalisée d'une part pour accélérer le calcul et d'autre part pour respecter les consignes locales  $n_{x,k}$  des cellules de filtrage.

### 3.1 Décomposition en filtres élémentaires

L'équation 1 (pour  $r_{x,k} = 0$ ) étendue aux deux dimensions spatiales conduit à une convolution binomiale locale équivalente à un masque de taille  $(2n_{x,k} + 1) \times (2n_{x,k} + 1)$ .

D'un point de vue algorithmique cette opération peut être facilement optimisée par une décomposition en cascade de convolutions élémentaires bidimensionnelles appliquées dans l'image (masque  $b_{in}[3 \times 3]$  de l'équation 2). Par cette méthode, nous passons d'un coût en opérations *MAC* (Multiplication Accumulation) de  $(2n + 1)^2$  à  $9n$ .

D'un point de vue architectural, cette solution programmée sous forme d'itérations simples permet de mieux optimiser l'utilisation du code dans l'architecture des processeurs par un adressage rapide et simple des opérandes et d'exploiter pleinement le cache d'instructions, même limité, de certains de calculateur ([3] *DSP, CPU...*). Pour ce qui est de la composante temporelle de filtrage décrite par l'équation 1, elle est cascadée avec la composante spatiale.

### 3.2 Filtrage spatial

Si on applique les optimisations précédentes le temps de calcul, fonction de  $n_{x,k}$ , reste toujours important. Nous avons donc mis en oeuvre une architecture de filtrage multi-échelle pour la couche illustrée figure 3.

#### 3.2.1 Architecture du filtrage

Le filtrage multi-échelle effectue des traitements dans les bandes de fréquence adaptées de façon à réduire les

temps de calcul. Il est réalisé à 2 échelles de fréquence ; l'échelle 0 qui est le niveau de représentation de l'image originale et l'échelle  $i$ , celui d'une représentation de plus basse résolution de l'image dans un spectre de fréquences divisé par  $2^i$ . Dans les conditions fixées par la section 3.2.3, le filtrage est effectué en 3 étapes :

1. Cette étape prépare les images au passage de l'échelle 0 à  $i$ . Pour cela, elle filtre les hautes fréquences pour éviter un recouvrement de spectre puis elle sous-échantillonne les images opérands de la couche (images  $e$ ,  $n$  et  $b$ ) ;
2. Cette étape réalise le filtrage passe-bas dans l'échelle  $i$  ;
3. Pour finir cette dernière étape à l'inverse de l'étape 1 prépare l'image résultat de la couche au passage de l'échelle  $i$  à 0 ;

Le filtrage dans l'échelle de fréquence  $i$  est effectué par  $n_i$  itérations successives. Pour une échelle  $i$ , un pixel  $e_{x,k}$  de l'image d'entrée est convolué avec le noyau  $b_{in}$  à la condition où le pixel  $n_{x,k}$  est supérieur strictement à 0. Dans la condition inverse la convolution n'est pas faite.

$$\text{Avec } b_{in} = \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2)$$

$b_{in}$  est l'opérateur binomial élémentaire de convolution.

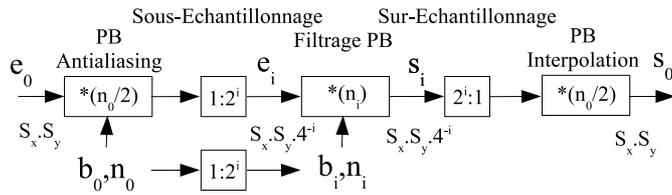


FIG. 3 – Architecture du filtre spatial multi-échelle. La notation  $*$  est l'opérateur de convolution.  $e_i$  et  $s_i$  sont respectivement l'image d'entrée et l'image résultat provisoire de la couche dans l'échelle  $i$ .

### 3.2.2 Fonction coût calcul

L'équation 3 donne le calcul du nombre d'itérations  $n_{eq}$  de filtrage équivalent fait à pleine résolution avec les conditions sur  $n_0$  pour l'antialiasing. L'ordre de grandeur de la fonction *coût* s'exprime à partir de deux paramètres de filtrage  $n_0$  et  $n_i$  pour une image de taille  $S_x \times S_y$ . Les temps de filtrage d'antialiasing et d'interpolation sont pris en compte. Le coefficient  $\alpha$  entier et supérieur à  $2^i$  [2] et représente le nombre de filtrages dans les étapes de sous ou bien sur-échantillonnage.

$$n_{eq} = n_0 + n_i \cdot 2^i \quad \text{avec } n_0 = 2 \cdot \alpha$$

$$\text{Coût} = S_x \cdot S_y \cdot ((n_0 + 3) + (n_i + 1) \cdot 4^{-i}) \quad (3)$$

On constate dans la figure 4 qu'avec une telle architecture de filtre qu'il existe un temps minimal de traitement correspondant à une échelle  $i_{opt}$ . Tout filtrage de couche en fonction du paramétrage sera effectué à une échelle adaptée  $i_{opt}$  afin de minimiser son temps global de calcul.

### 3.2.3 Choix de l'échelle de fréquence

Les paramètres  $n_0$  et  $n_i$  doivent être choisis pour respecter à la fois les conditions de Shannon et obtenir un temps de calcul minimisé. L'algorithme 4 donne la méthode utilisée. La valeur minimale de  $n$  restreint éventuellement le choix de l'échelle  $i_{opt}$  détecté à l'aide de la fonction *Coût*. La valeur  $\beta$  est le minimum souhaité d'itérations de filtrage dans l'échelle  $i$ .

```

if ( $Min(n) - 2 \cdot \alpha > (\beta \cdot 2^i)$ ) then
     $n_i = \text{Partie Entière}(2^{-i} \cdot (n - n_0))$ ;
else
     $n_i = 0$ ;  $n_0 = n$ ;
end if

```

(4)

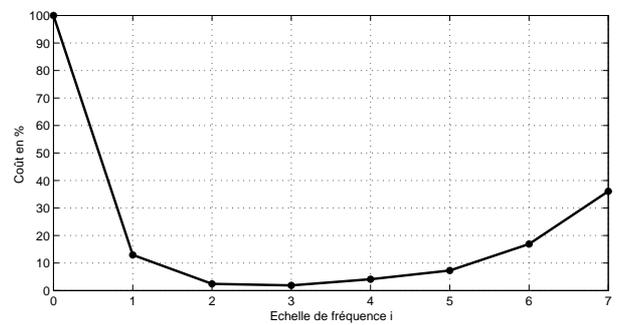


FIG. 4 – La fonction *Coût* en % par rapport au calcul en résolution 0 est représentée pour une image  $1024 \times 1024$  filtrée spatialement de façon homogène avec  $n = 1000$ . Elle passe par une valeur minimale  $i_{opt} = 3$  détectable et utilisée pour minimiser le temps de calcul en fonction du contexte de la couche.

### 3.3 Filtrage temporel

Le filtrage est réalisé à la suite du spatial à pleine résolution. Le coût de calcul de ce filtre, bien moindre que le spatial, est de l'ordre au mieux de  $2 \cdot S_x \cdot S_y$  traité dans un calculateur à jeu d'instructions arithmétiques de type MAC.

## 4 Résultats

Nous présentons ci-dessous un exemple de traitement obtenu sur une couche simulant la perception des cônes de la rétine. Les cônes sont disposés dans la rétine avec une densité variable en fonction de leur position [7]. Le centre de la rétine appelée fovéa est une zone très dense en cellules cônes pour la vision de précision ( $150000/mm^2$ ). La périphérie dédiée plutôt à la vision dynamiques est moins précise car elle possède une densité de cellules cônes vingt fois moindre. Nous simulons le résultat de perception des cônes figure 6 en réponse d'une image statique de luminance représentée figure 5. Notre maillage de cellule étant uniforme, nous agissons sur le couplage latéral entre les cellules pour créer cet effet. Ainsi on observe figure 6 une

zone centrale de netteté (région fovéale) et une périphérie floue (macula).

La figure 7 présente un comparatif de réponses. La référence est un filtre passe-bas gaussien de variance fréquentielle  $\sigma_f^2 = 2 \cdot 10^{-2} \text{pixel}^{-2}$  équivalent à un filtrage de  $n = 100$  itérations du noyau  $b_{in}$ . Ce filtre gaussien est comparé figure 7.b déjà avec ce dernier (résultats similaires). La figure 7.a présente l'écart de réponse créé par l'utilisation du multi-échelle pour  $i = 2$  avec  $n_0 = 16$  et  $n_2 = 21$ .



FIG. 5 – Image *château* de luminance originale.



FIG. 6 – Réponse de la couche cône.

## 5 Conclusions et perspectives

Nous avons présenté une solution à l'implémentation de couches rétiniennes constituées de cellules de filtrage. Une méthode de filtrage est proposée pour calculer l'algorithme caractérisant la fonction d'une couche rétinienne. Elle propose une organisation de la forme du filtre (décomposition en filtres élémentaires et le multi-échelle) pour accélérer les temps de calcul. L'utilisation de filtres élémentaires nous fait gagner des facteurs importants de temps (section 3.1). Le multi-échelle accentue ce gain illustré figure 4 sachant que l'inverse de la fonction *Coût* en % nous donne le facteur de temps gagné. Cependant le choix de l'échelle de filtrage dépend dans notre approche du paramètre  $n_{x,k}$

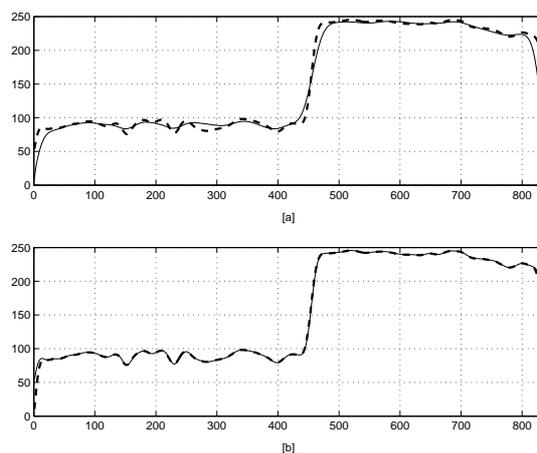


FIG. 7 – Réponses observées sur le profil de la ligne centrale de l'image *château* par trois méthodes de calcul. La courbe en pointillée est la réponse de référence (filtre gaussien avec  $\sigma_f^2 = 2 \cdot 10^{-2} \text{pixel}^{-2}$ ).

des cellules. Il est important que ce paramètre ne varie pas dans des proportions importantes car le choix de  $i_{opt}$  est fonction de la taille de l'image et de la valeur minimale de  $n_{x,k}$ .

Les difficultés de faire évoluer la structure de filtre d'inspiration *traitement d'images*, nous ont incitées à nous orienter vers une architecture d'automates cellulaires [8]. Une première réalisation portant sur la modélisation de la couche *PLE* a déjà été menée [3]. A court terme nous envisageons, afin d'accéder à plus de puissance de calcul, d'utiliser les architectures parallèles présentes dans les processeurs graphiques (*GPU*).

## Références

- [1] W.H.A Beaudot. *Le traitement neuronal de l'information dans la rétine des vertébrés – Un creuset d'idées pour la vision artificielle*. PhD thesis, Institut National Polytechnique de Grenoble, France, 1994. 228 pages.
- [2] A. Chelikian. Fast computation of optimal semi-octave pyramids. *7th S.C.I.A.*, August 1991.
- [3] F. Devillard. Implémentation par automates cellulaires d'une modélisation architecturale de rétine biologique. *R.E.A.D. 2005*, Juin 2005.
- [4] J.-M. Jolion. *Les systèmes de vision*. Traitement du Signal et de l'Image. HERMES Science Europe, Paris, 2001.
- [5] David Marr. *Vision*. W. H. Freeman and Company, New York, 1982.
- [6] J.-F. Risse. *Exploration de la fonction visuelle – Applications au domaine sensoriel de l'œil normal et en pathologie*. Masson - Société Française d'Ophthalmologie, 1999.
- [7] W.F. Schreiber. *Fundamentals of Electronic Imaging Systems – Some Aspects of Image Processing*. Springer-Verlag, 3rd edition, 1993.
- [8] S. Wolfram. *A new kind of science*. Wolfram Media Inc., 1st edition, 2002.