

PROCESSEUR INTEGRE D'EXTRACTION DE CONTOURS D'UNE IMAGE TELEVISION EN TEMPS REEL

Gérard BOUVIER*, François DEVILLARD*, Abdellah LAMINE**

* L T I R F, INPG 46, avenue Félix Viallet 38031 - GRENOBLE-CEDEX (France)

** Ecole Nationale Supérieure d'Electricité et de Mécanique, route d'el Jadida, CASABLANCA (Maroc)

RÉSUMÉ

L'identification et la localisation d'objets dans une image, et plus généralement l'analyse automatique de scènes, nécessitent une phase préalable de détection de contours. Compte-tenu de la très grande quantité d'informations à manipuler, la mise en oeuvre d'algorithmes de détection de contours sur un ordinateur classique entraîne des temps de calcul toujours très longs pour des applications en robotique. Les auteurs présentent un processeur numérique qui travaille à la volée sur le signal vidéo numérisé d'une caméra TV CCD. Ce processeur calcule en temps réel les composantes du vecteur gradient associé à chaque pixel d'image grâce à un filtre de Sobel. L'image traitée apparaît en synchronisme avec le signal vidéo, avec un retard de 1 ligne TV et 1 pixel par rapport à l'image d'origine.

I. Introduction

Nous présentons dans cet article un processeur de détection de contours opérant en temps réel sur le signal vidéo numérisé d'une caméra de télévision.

Ce processeur est un circuit intégré à application spécifique (ASIC), réalisé en technologie CMOS. Il constitue l'opérateur de traitement bas niveau d'un système de vision.

Nous considérons que les images sont constituées par la juxtaposition de régions de luminosité homogène et que ces régions sont séparées par des zones de contraste. Le processeur sélectionne les pixels d'image appartenant aux contours des régions.

Le chaînage de ces pixels ainsi que les traitements de vision de plus haut niveau seront réalisés par un ordinateur hôte à partir des données transférées dans une mémoire d'image commune pour des raisons évidentes de synchronisation du flux de données.

ABSTRACT

Objects identification and localization and more generally scene analysis, need a preliminary step of edge detection. Because of the very important amount of data to be processed, implementing any edge detection algorithm on a conventional computer leads to a high computational cost, too high for many applications in robotics. The authors present here an on-line real time processor which works upon the digitized signal of a TV camera. This processor computes the two components of the gradient associated with each pixel, using a Sobel filter. The processed image appears synchronously with the video signal, delayed by 1 TV lines and 1 pixel duration compared to the original image.

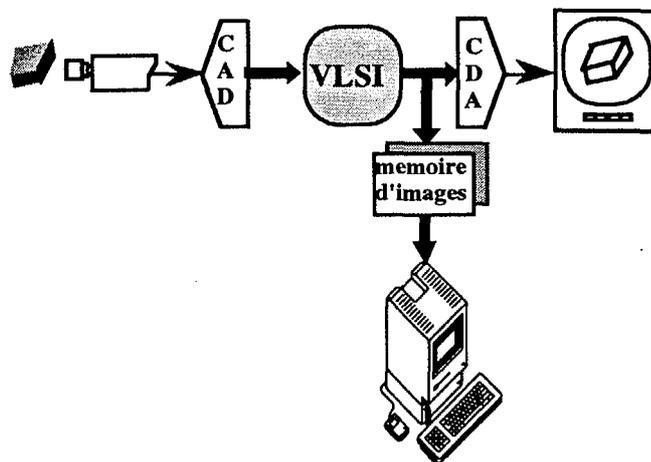


Figure 1 : Exemple de réalisation d'une chaîne de traitement d'image avec l'opérateur VLSI présenté.



II - Détection des zones de contraste

Notre objectif était de réaliser le traitement en temps réel donc sans mémorisation préalable de l'image. Nous nous sommes intéressés aux méthodes fondées sur une estimation du gradient de luminosité, mettant en oeuvre des opérateurs locaux de traitement.

A chaque pixel $I(m,n)$ de l'image est associée une approximation $G(m,n)$ du vecteur obtenu par différence finie sur un voisinage.

Les composants $G_h(m,n)$ et $G_v(m,n)$ du vecteur gradient selon les directions horizontales et verticales sont obtenues par convolution de l'image avec des masques M_h et M_v en application des propriétés des différences finies dans des images échantillonnées.

Le module de gradient est lors :

$$|G(mn)| = \sqrt{G^2_h(m,n) + G^2_v(m,n)}$$

et sa direction du gradient est donnée par :

$$\phi(mn) = \text{Arctg} \frac{G_v(m,n)}{G_h(m,n)}$$

Parmi tous les gradients non nuls, les zones de contraste sont définies comme des zones à fort gradient de luminance, alors le pixel $I(m,n)$ appartient à une zone de contraste si $|G(m,n)| > S$ ou S : seuil qui peut être déterminé d'après les caractéristiques globales de l'image sur un histogramme.

Les zones de contraste obtenues selon cette méthode ont une épaisseur dépendant de l'étendue de la zone de transition entre les régions et de la taille des masques M_h et M_v utilisés pour le calcul.

Un algorithme de détermination du contour à l'intérieur de la zone de contraste est nécessaire et dans ces conditions le choix du seuil S n'est pas critique pour peu qu'il conserve toutes les zones de contraste.

Opérateur de calcul

Parmi les filtres non récursifs, les masques de Prewitt et de Sobel s'avèrent les mieux adaptés du point de vue du bruit et de la facilité de mise en oeuvre. Le filtre de Sobel est utilisé dans bon nombre d'applications déjà réalisées au laboratoire et il a déjà largement démontré son efficacité.

Masque de Sobel

$$M_h = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad M_v = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Ces masques possèdent la propriété d'être séparables.

$$M_h = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}^T = V_1 \cdot V_2^T \quad M_v = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -2 & -1 \end{bmatrix}^T = V_2 \cdot V_1^T$$

$$\text{et } G_h = I * M_h = I * V_1 \cdot V_2^T$$

$$G_v = I * M_v = I * V_2 \cdot V_1^T$$

avec I matrice 3×3 du voisinage du pixel $I(m,n)$ de l'image.

Cette propriété de séparabilité des masques de convolution permet de donner à l'opérateur une structure pipeline parfaitement adaptée au traitement des images T.V.

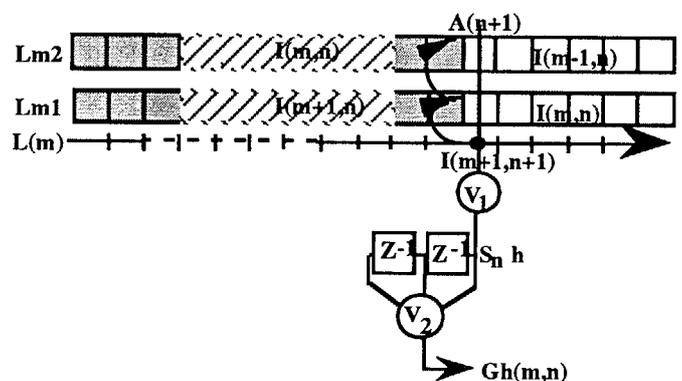


Figure 2 : Organigramme de calcul

Lorsque $I(m+1,n)$ est la ligne courante (figure 2) , si les deux lignes précédentes $I(m,n)$ et $I(m-1,n)$ sont disponibles en mémoire dans des registres Lm1 et Lm2, à l'adresse $A(n+1)$ du pixel courant se trouvent les pixels $I(m, n+1)$ et $I(m-1, n+1)$. Alors :

$$S_{(n+1)h} = I * V_1 = \sum_{j=-1}^{+1} I(m+j, n+1) V_{1(j)}$$

aux adresses précédentes :

$$S_{(n)h} = I * V_1 = \sum_{j=-1}^{+1} I(m+j, n) \cdot V_{1(j)}$$

$$S_{(n-1)h} = I * V_1 = \sum_{j=-1}^{j+1} I(m+j, n-1) \cdot V_{1(j)}$$

$$\text{alors: } Gh(m,n) = \sum_{k=-1}^{+1} S_{(n+k)h} \cdot V_{2(k)}$$

$$Gh(m,n) = S_{(n-1)h} \cdot V_{2(-1)} \cdot Z^{-2}$$

$$+ S_{(n)h} \cdot V_{2(0)} \cdot Z^{-1} + S_{(n+1)h} \cdot V_{2(1)}$$

$$\text{soit: } Gh(m,n) = \sum_{k=-1}^{+1} S_{(n+k)h} V_{2(k)} Z^{-1-k}$$

III - Blocks fonctionnels

Pour la taille des noyaux de convolution choisie, les pixels nécessaires à l'opérateur de calcul se trouvent situés sur 3 lignes d'images télévision consécutives : la ligne courante et les 2 précédentes. Pour chacune de ces lignes le calcul nécessite 3 pixels consécutifs. Le processeur doit donc comprendre :

- deux registres de mémorisation de 2 lignes de 768 pixels (8 bits) pour répondre à la définition des caméras CCD modernes,

- deux multiplieurs accumulateurs MAC 1 et MAC 2 pour les produits matriciels.

$$MAC\ 1(I) = I \cdot V_1^T$$

$$MAC\ 2(I) = I \cdot V_2^T$$

Chacun est utilisé pour les calculs des deux composantes du gradient,

- deux registres de mémorisation des convolutions colonne S_{nh} et S_{nv}

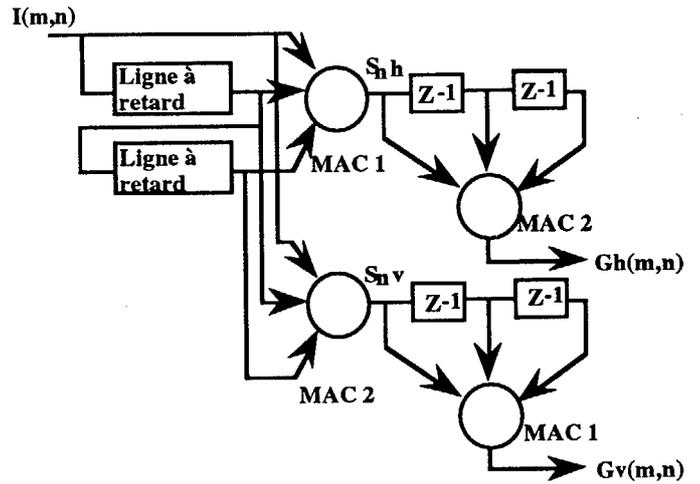


Figure 3 : Structure de l'opérateur

III.1. Registre lignes

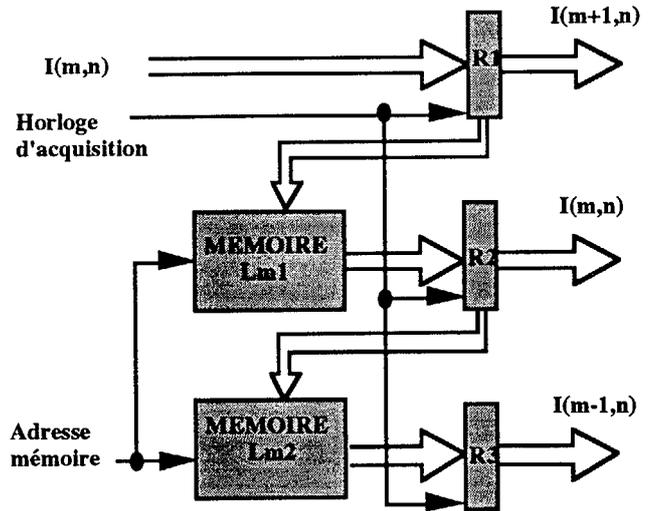


Figure 4 : Registres de mémorisation lignes

Un compteur d'adresses sur 10 bits maximum, extérieur au circuit pour des raisons de souplesse de mise en oeuvre sélectionne sur les 2 registres Lm1, Lm2 la colonne pixel.

L'acquisition et la lecture des pixels est faite par 3 registres R1, R2, R3 au rythme de l'horloge d'acquisition.

Le transfert des contenus des registres R1 vers Lm1 et R2 vers Lm2 assure le balayage vertical de l'image. Ces produits S_{nh} , S_{nv} sont mémorisés dans des registres spécifiques.

Ces mémoires ont été dessinées à l'aide d'un compilateur de silicium en technologie CMOS 1,2 μ , ce qui donne une taille de cellule de 4,3 mm^2 avec un temps d'accès de 12 ns ce qui est compatible avec les 67 ns d'une horloge à 15 MHz pour une définition de 768 pixels utiles par ligne TV.



IV - REALISATION

III.2. Multiplieurs-accumulateurs

Les calculs des gradients $G_h(m,n)$ et $G_v(m,n)$ dans le cas d'un filtre de Sobel et compte-tenu des propriétés de séparabilité des matrices, nécessitent 2 multiplieurs-accumulateurs pour effectuer les produits de convolution :

$$\text{MAC 1} = 1.e_1 + 2.e_2 + 1.e_3$$

$$\text{et } \text{MAC 2} = -1.e_1 + 0.e_2 + 1.e_3$$

soit en calcul numérique :

$$\text{MAC 1} = e_1 + e_3 + T(e_2)$$

$$\text{MAC 2} = c(e_1) + e_3$$

avec $T(\)$ = opérateur de translation pour la multiplication par 2,

avec $C(\)$ = complément à 2 pour la négation.

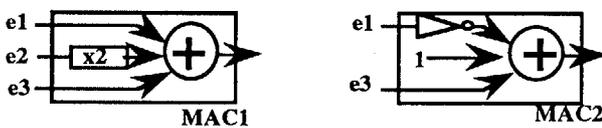


Figure 5 : Structure des opérateurs de calculs

Ces opérateurs ont été réalisés en logique combinatoire en technologie CMOS 1,2 μ .

MAC 1 occupe une surface 0,9 mm² et nécessite un temps de propagation de 9 ns,

MAC 2 occupe une surface de 0,4 mm² et nécessite un temps de propagation de 23 ns.

Les résultats délivrés par chaque MAC sont stockés dans 2 registres colonnes pour chaque composante du gradient. Cette structure pipe-line est particulièrement bien adaptée à l'analyse des images et permet la parfaite synchronisation des opérateurs.

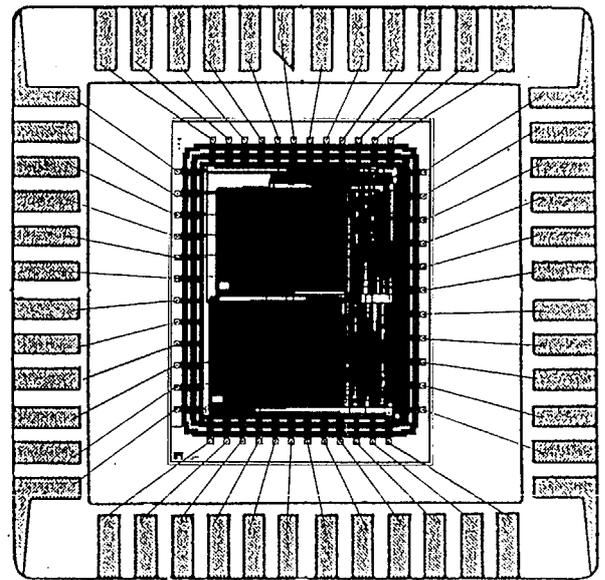


Figure 6 : Circuit réalisé

L'ensemble du circuit a été réalisé en technologie CMOS 1,2 μ avec l'outil de placement routage des cellules pré-caractérisées de ES2 : SOLO 1400. Cette méthode de design ne permet pas une réduction maximum de la surface de silicium mais optimise le rapport : temps de conception/surface de silicium. Le circuit complet a une taille de 6 mm x 5 mm = 30 mm² pour 47 plots d'entrées/sorties pour des registres permettant le traitement d'images TV de 768 pixels réels digitalisés sur 8 bits, quel que soit le nombre de lignes. Pour chaque pixel, les composantes du gradient sont délivrées avec une ligne, une colonne et 53 ns de retard.

BIBLIOGRAPHIE

- [1] M. BASSEVILLE *Détection de contours : méthodes et études comparatives*. Ann. Télécomm., **34**, n° 11-12, pp. 559-579, Fr (1979)
- [2] J. CANNY, *A computational approach to edge detection*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. **PAMI-8** (6), pp. 679-698 (1986)
- [3] CHEN Yi L., F.V.M. CATTLOOR H.J. DE MAN, *An efficient ASIC architecture for real-time edge detection*. IEEE Transactions on Circuits and Systems, vol. **36**, n° 10, pp. 1350-1359 (1989)
- [4] R. CYPHER & J.C.L. SANZ, *SIMD architectures and algorithms for image processing and computer vision*. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. **37**, n° 12, pp. 2158-2173 (1989)
- [5] R. DERICHE, *Using Canny's criteria to derive a recursively implemented optimal edge detector*. International Journal of Computer Vision, pp. 167-187 (1987)