

ETUDE D'ARCHITECTURES POUR LA REALISATION D'UN DETECTEUR DE CONTOURS DE TYPE "CANNY-DERICHE"

L.TORRES, M.ROBERT, E.BOURENNANE*, M.PAINDAVOINE*

UNIVERSITE MONTPELLIER II, LIRMM, UMR 9928 CNRS /UM II, CP 79
161 Rue ADA
34392 Montpellier cedex 5, FRANCE
email : torres@lirmm.fr

*UNIVERSITE DE BOURGOGNE, Laboratoire LIESIB.
6 Rue Gabriel
21004 Dijon Cedex, FRANCE

– Résumé –

Nous présentons la conception et l'implantation d'un opérateur de détection de contours basé sur une extension du filtre de Canny-Deriche (modèle de contours non idéal). Le filtre obtenu est un filtre récursif d'ordre 3, ses performances apportent une amélioration de 20% par rapport au filtre de Deriche en ce qui concerne le rapport signal sur bruit dans le cas d'images floues et bruitées. Le processeur dédié (ASIC) ainsi obtenu est réalisé en technologie CMOS 1 μ m, il est capable de traiter un pixel en 30ns pour des images dont les dimensions varient entre 64x64 et 1024x1024 pixels.

I- INTRODUCTION

Pour sélectionner les informations géométriques relatives à des objets présents dans une séquence d'images, une des étapes essentielles est l'extraction de contours des objets considérés (la reconnaissance de caractères, la détection de défauts sur des pièces industrielles sont les principales applications visées).

L'opérateur de détection de contours que nous proposons correspond à une extension du filtre de Canny et de Deriche [1], [2], [3]. Cette étude a permis l'implantation de ce filtre sous une forme récursive (ordre 3), et nous avons mis en évidence une amélioration sensible des performances de détection (au sens des critères de Canny) par rapport à un filtre de type Deriche (optimisé correctement pour le contour choisi).

Cependant l'intégration numérique de cet algorithme en temps réel ne peut être effectuée que par une étude approfondie de l'architecture des filtres numériques, ainsi que par une implantation VLSI appropriée.

Le processeur dédié ainsi obtenu, réalisé en technologie CMOS 1 μ m, est capable d'extraire les contours d'une image en temps réel (traitement par pixel : 30 Mhz/pixel).

L'article est organisé de la manière suivante : la première partie est consacrée à la présentation de l'algorithme et à la définition des critères d'optimisation du filtre. Nous explicitons ensuite deux méthodes d'implantation de filtres récursifs. Finalement nous présentons les spécifications du processeur dédié ainsi obtenu.

– Abstract –

We present the design and the implementation of an edge detection operator based on a Canny-Deriche filter extension (not ideal edge model). The filter obtained is a three order recursive filter, its performances allow to increase the signal/noise ratio about 20% for blurred and noisy images in comparison with the Deriche filter. The integrated circuit, realized in CMOS 1 μ m, is able to process a pixel in 30ns and images size from 64x64 to 1024x1024 pixels.

II- ALGORITHME

L'acquisition d'une scène réelle, en relief, par une même caméra implique la présence d'un certain flou. De plus le flou des contours réels peut provenir de causes différentes (mouvements d'objets dans la scène, faible résolution de l'image, éclairage diffus...) c'est pourquoi le modèle choisi ne pourra, de toute façon, les représenter qu'approximativement.

Le modèle de contour choisi (modèle de contour exponentiel) est représentatif d'une large classe d'images floues, et de ce fait constitue une bonne approximation des contours réels [1], Figure 1 et Figure 2.

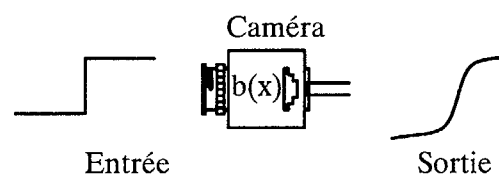


Figure 1 : Fonction de transfert de la caméra

$$\text{Avec } b(x) = \frac{s}{2} e^{-s|x|} \text{ et } B(\omega) = \frac{s^2}{s^2 + \omega^2}.$$

Notre modèle de contour dépend d'un seul paramètre "s". Nous pouvons considérer que "s" est une caractéristique de la caméra et qu'il n'est donc pas nécessaire de le déterminer pour chaque contour séparément ; il sera le même pour tous les contours idéaux rendus flous par la caméra.



$$\begin{cases} C(x) = 1 - \frac{e^{-sx}}{2} & \text{pour } x > 0 \\ & \text{ou } s > 0 \\ C(x) = \frac{e^{sx}}{2} & \text{pour } x < 0 \end{cases}$$

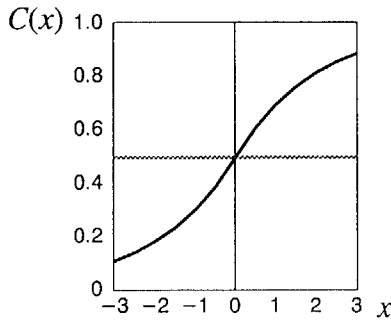


Figure 2 : Modèle de contour exponentiel, (exemple pour s=0.5)

Les critères d'optimisation choisis pour cet algorithme sont les suivants :

- *Le rapport signal sur bruit* : le bruit est considéré blanc, additif, gaussien et centré en zéro.
- *La localisation* : caractérise l'écart moyen entre la position du contour détecté en présence de bruit et sa position exacte.
- *Suppression des réponses multiples* : en présence de bruit le signal détecté peut présenter plusieurs maxima, ces réponses parasites doivent être aussi éloignées que possible de la réponse principale.

Nous transformons le problème d'optimisation en un problème sous contraintes, en utilisant le calcul variationnel.

Suite à cette manipulation mathématique, la réponse impulsionnelle du filtre optimisé obtenue pour des contours flous est la suivante :

$$\begin{cases} f(x) = ke^{\alpha x} \sin \omega x + e^{\alpha x} \cos \omega x - e^{sx} \\ f(-x) = -f(x) \end{cases}$$

$$k_0 = \frac{m(m+1)(m-1)^3}{(1-4m+4m^2-m^4)} \quad \text{pour } 0 < m < 1$$

Où m détermine la bande passante du filtre

Ceci nous conduit à une implantation récursive (passage par la transformée en Z) avec deux filtres récursifs variant en directions opposées.

$$y^+(i) = a_1x(i-1) + a_2x(i-2) + a_3y^+(i-1) - a_4y^+(i-2) + a_5y^+(i-3)$$

$$y^-(i) = -a_1x(i-1) - a_2x(i-2) + a_3y^-(i-1) - a_4y^-(i-2) + a_5y^-(i-3)$$

Avec a_i pour $i \in [1;5]$ coefficients du filtre [1].

Nous avons montré que pour un choix correct de m, et dans le cas d'images bruitées, notre opérateur améliore sensiblement le rapport signal/bruit (Par exemple pour un rapport Signal/Bruit de 2 l'amélioration est de 20%) par rapport au filtre de Deriche (Selon l'indice de Pratt [6]). La comparaison de notre filtre avec celui de Deriche est représentée en Figure 3.

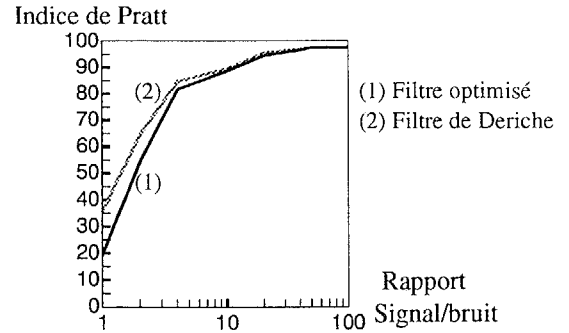


Figure 3 : Comparaison du filtre optimisé et du filtre de Deriche.

III - ETUDE DU FILTRE NUMERIQUE

De nombreuses applications liées au traitement du signal et de l'image utilisent des filtres numériques. Pour des fréquences élevées, l'implantation VLSI des filtres non récursifs est un problème bien maîtrisé. Tel n'est pas le cas pour la conception des filtres récursifs (IIR), en effet la boucle de récursivité limite la fréquence de fonctionnement.

Pour palier à ce problème les techniques d'anticipation par ajout de pôles sont souvent utilisées. Nous présentons une première approche consistant en une anticipation par ajout de pôles dispersés avec une décomposition en puissance de deux, et une redistribution globale des registres internes aux filtres. Pour améliorer le compromis vitesse/surface notre seconde approche consiste en une technique de redistribution locale des registres.

III.1 - Première approche : Redistribution globale des registres

Cette première approche consiste à favoriser la fréquence de traitement au détriment de la surface (silicium) du circuit, en utilisant l'accélération des calculs par anticipation. La transformée en Z du filtre (causal et anti-causal) est donnée par :

$$\begin{cases} TZ^+ = \frac{Y^+(z)}{X(z)} = \frac{a_1z^{-1} + a_2z^{-2}}{1 - a_3z^{-1} + a_4z^{-2} - a_5z^{-3}} \\ TZ^- = \frac{Y^-(z)}{X(z)} = -\frac{a_1z + a_2z^2}{1 - a_3z + a_4z^2 - a_5z^3} \end{cases}$$

TZ^+ est la transformée en Z de la partie causale du filtre, et TZ^- la transformée de la partie anti-causale.

La fonction de transfert de ce filtre $T(z)$ est transformée par anticipation [1] en 4 étages de "pipeline" (limitée par la capacité d'intégration dans un ASIC). Le filtre résultant de la transformation par anticipation est un filtre d'ordre 12 dont l'équation de transfert $H(z)$ est donné par :

$$H(z) = \frac{(a_1Z^{-1} + a_2Z^{-2})(1 + a_3Z^{-1} + a_4Z^{-2} + a_5Z^{-3})}{1 - [(a_3^2 - 2a_4)^2 + (-a_4^2 + 2a_3a_5)]Z^{-4} \frac{[1 + (a_3^2 - 2a_4)Z^{-2} - (-a_4^2 + 2a_3a_5)Z^{-4} + a_5^2Z^{-6}]}{+ (-(-a_4^2 + 2a_3a_5) + 2a_3^2(a_3^2 - 2a_4))Z^{-8} + a_5^4Z^{-12}}}$$

Cette fonction de transfert peut être vue comme le produit de 4 fonctions de transfert. De ce fait, nous montrons comment la partie récursive de $H(z)$ peut être accélérée par une redistribution de registres, [4],[5] et par l'insertion de registres à l'intérieur des multiplieurs comme cela est décrit dans la Figure 4 ($e(n)$ correspondant à la sortie de la partie non récursive).

IV – SPECIFICATIONS DE L'ASIC

L'implantation de cet ASIC a été réalisée à l'aide du logiciel OPUS (Cadence Design Framework II) avec une approche de cellules précaractérisées pour une technologie CMOS 1µm.

IV-1 Principe de l'architecture

L'architecture proposée est présentée en Figure 6. Les principaux blocs utilisés sont : le filtre causal et anti-causal, un micro-contrôleur permettant de gérer les coefficients des filtres, et la partie de mémorisation des coefficients.

a) Les Filtres

Chaque filtre est composé de cinq multiplieurs 12x12 bits, quatre additionneurs 12 bits et six registres de 12 bits. Le choix respectif de l'algorithme de Booth et du "carry-select" pour le multiplieur et l'additionneur offre le meilleur compromis vitesse/surface en cellules précaractérisées.

b) Les Coefficients

Tous les coefficients sont codés sur 12 bits en complément à deux, et sont chargés en série par le bus vidéo en 10 cycles d'horloge dans des registres statiques de 12 bits. Une fois ces coefficients chargés ils sont alors disponibles à l'entrée du multiplieur correspondant.

c) Le Test

Chaque filtre peut être testé indépendamment, en utilisant une méthode de test appelée "full scan". Tous les registres sont chaînés pour former, en mode test, un long registre à décalage permettant ainsi d'imposer des vecteurs de test.

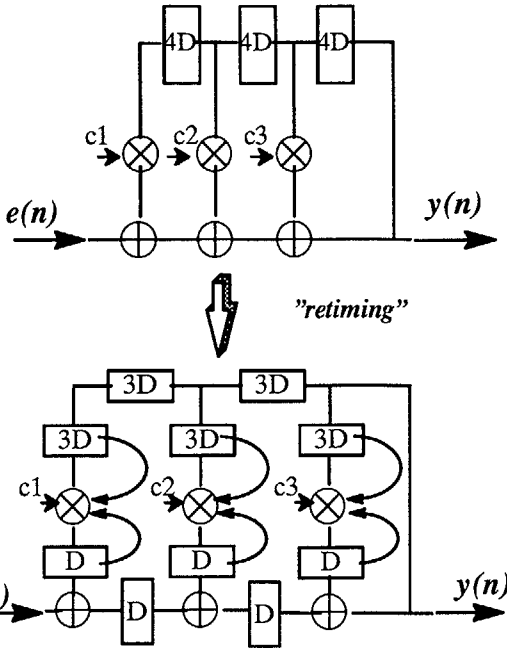


Figure 4 : Insertion de pipeline grâce à une redistribution globale des registres

Cette technique permet d'accélérer les calculs (fréquence de traitement de 65 Mhz), mais elle est trop complexe pour permettre une optimisation de la surface de silicium. En effet le nombre de multiplieurs est de 11 et le nombre d'additionneurs de 10 avec plus de 30 registres de 12 bits.

III.2-Deuxième approche : Redistribution locale des registres

Cette nouvelle approche consiste à implanter le filtre f(x) directement, en réalisant une redistribution locale des registres sur la partie récursive, comme cela est représenté sur la Figure 5.

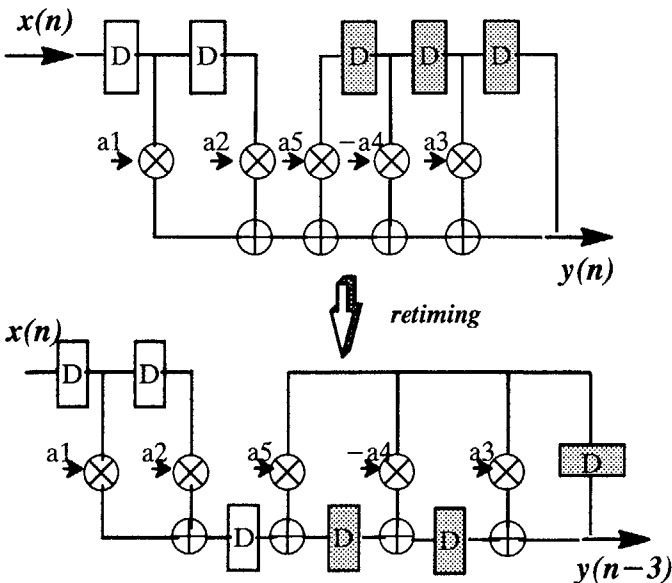


Figure 3 : Redistribution locale des registres

La fréquence de fonctionnement est alors donnée par le temps de calcul d'une multiplication et d'une addition. Le gain en temps correspond à l'élimination de trois additionneurs 12 bits, soit un gain de l'ordre de 30%. Le choix pour la conception du processeur dédié s'est porté sur cette architecture.

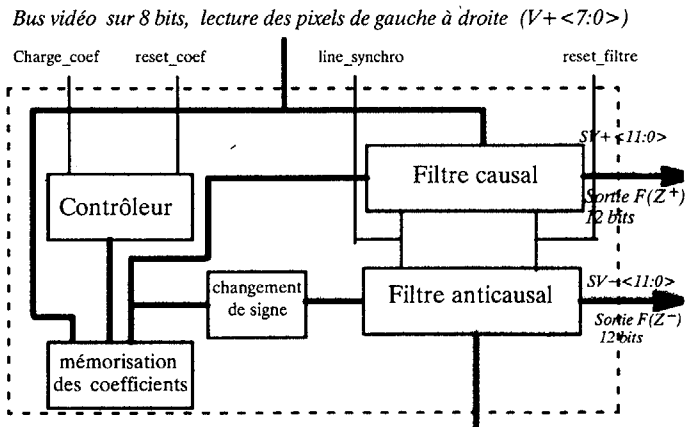


Figure 6 : Architecture de l'ASIC

IV-2 Caractéristiques du processeur dédié

Les prototypes réalisés avec cette approche, dans une technologie CMOS 1µm, ont été testés avec succès. En effet le processeur dédié ainsi obtenu (Figure 7) est capable de traiter un pixel en 30ns, avec des images de taille variable (64x64 à 1024x1024 pixels). La complexité du circuit est de 72 000 transistors pour une surface de 29 mm² (5.6 x 5.3 mm²).

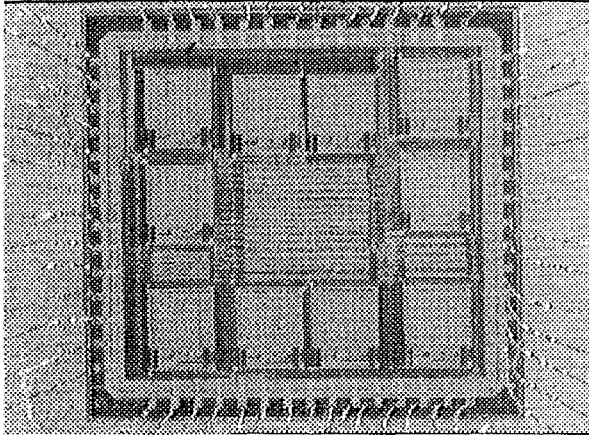


Figure 7 : Photographie du processeur dédié :

- Opérateur de détection de contours en temps réel
- Fréquence : 33Mhz
- Surface : 29 mm² (avec les plots d'entrées/sorties)
- Nombre de transistors : 72 000 Tr
- Opérateur : algorithme de Canny–Deriche optimisé
- Signal vidéo d'entrée sur : 8 bits
- Précision interne : 12 bits
- Sortie sur : 12 bits
- Codage : en complément à deux
- Technologie : Eurochip CMOS 1 μm (ES2)
- Nombre de plots : 57 (Package : LCC 68 pads)

V – CONCLUSION

L'accélération des filtres numériques récursifs est une étape nécessaire pour l'implantation d'algorithmes en temps réel. Nous avons mis en évidence deux techniques de conception de filtres récursifs. Cette étude a débouché sur un processeur dédié à haut débit. Les résultats satisfaisants obtenus nous permettent de développer actuellement une carte de traitement d'images autour de ce processeur, ainsi qu'un nouveau ASIC mémorisant l'équivalent de deux lignes de 512 pixels.

REFERENCES

- [1] E.BOURENNANE, M.PAINDAVOINE et F.TRUCHETET : " Amélioration du filtre de Canny–Deriche pour la détection de contours sous forme de rampe", Traitement du Signal, No 4, 1993.
- [2] J.CANNY : " A computational approach to Edge detection", IEEE trans. on Patt. AN. and Machin. Int, Vol 8, No 6, 1986
- [3] R.DERICHE : " Using Canny Criteria to derive a recursively implemented optimal Edge detector", Int. Journal of Comp. Vision, p167–187,1987
- [4] C.LEISERSON, F.FROSE, J.SAXE, " Optimizing synchronous circuitry by retiming", in Proc 3rd Caltech Conference, Pasadena, 1983
- [5] K.K PARHI, " Pipeline Interleaving and Parallelisme in Recursive Digital Filters–Part I : Pipelining using scattered Look Ahead and Decomposition", Journal IEEE Trans. Acoust. ,Speech, Signal processing, vol 37, No 7, pp 1099–1117, july 1989.
- [6] W.K PRATT, " Digital Image Processing", Wiley–Interscience, New York 1978.