

Évaluation des performances d'un système de vision à base d'un capteur intelligent

A. ELOUARDI, S. BOUAZIZ, A. DUPRET, L. LACASSAGNE, J.O. KLEIN et R. REYNAUD

Institut d'Electronique Fondamentale, Bât. 220, Université Paris Sud, 91405 Orsay, FRANCE
Tél.: +33 (0)1.69.15.41.06, Fax: +33 (0)1.69.15.40.00, Email: elouardi@ief.u-psud.fr.

Résumé - Une des solutions pour résoudre la complexité des traitements d'images est d'exécuter quelques traitements, bas niveau, sur le plan focal du capteur. Cet article présente un système de vision basé sur un capteur intelligent : PARIS1 (Programmable Analog Retina-like Image Sensor). Le circuit intègre un vecteur de processeurs analogiques qui permet d'augmenter la vitesse d'exécution des calculs tout en réduisant les flux de données et les échanges avec un processeur numérique. Un premier prototype est employé pour évaluer l'architecture d'un système de vision monopuce (SoC) à base d'un tel capteur et un processeur numérique. Un système de vision a été mis en oeuvre comme preuve de concept. Ceci nous a permis d'évaluer les performances requises pour une implantation possible d'un processeur numérique sur le même circuit. L'approche est comparée à d'autres architectures mettant en application des capteurs CMOS/APS interfacés au même processeur. La comparaison est liée au temps de calcul des traitements d'images, à la programmabilité et à la puissance consommée.

Mots-clés : Architecture des systèmes de vision, capteurs CMOS/APS, capteurs intelligents, SoC, modélisation et instrumentation.

1. INTRODUCTION

La plupart des systèmes de vision actuels sont équipés de circuits de traitements d'images utilisant des fonctions spécifiques et une puissance de calcul significative associée à des moyens de communication importants. L'approche classique pour effectuer ce genre de tâches consiste à acquérir l'image par une caméra CCD ou un capteur CMOS et la traiter sur une plateforme numérique (PC ou DSP par exemple). De tels systèmes ne remplissent pas les conditions de coût et de portabilité nécessaires aux applications de la robotique mobile et des véhicules intelligents. Ces systèmes sont, malheureusement, aussi coûteux que grands consommateurs d'énergie.

Pour faire face à la complexité des traitements et des algorithmes, il est donc souvent avantageux d'effectuer ces traitements directement dans le plan focal du capteur. L'une des solutions consiste à reporter une partie du traitement dans le circuit du capteur lui-même.

Cela donne un capteur doté d'une certaine intelligence et qui sera appelé une "rétine intelligente". Ces rétines ambitionnent d'implanter sur un même circuit l'imageur et une bonne partie des opérateurs de prétraitements avancés [1]. Cela permet de réduire les flots de données extra-circuit au bénéfice d'une puissance de calcul importante intra-circuit et d'effectuer des calculs massivement parallèles à basse consommation d'énergie.

De tels circuits intègrent des opérateurs de traitement du signal numérique et/ou analogique au sein des pixels [2][3] ou bien au bord de la matrice des pixels [4] pour réaliser certains traitements bas niveau et sont souvent dédiés à des applications spécifiques [5].

Dans ce contexte, plusieurs approches ont été étudiées. La solution proposée par P. Dudek [6] consiste à implanter des pixels intégrant un microprocesseur analogique, qui traite des données analogiques échantillonnées, ainsi qu'un processeur numérique. Cela produit un système mixte analogique/numérique capable d'opérer en mode SIMD. Dans une telle architecture, chaque processeur dispose de plusieurs mémoires analogiques, un registre de communication avec les pixels voisins et un multiplieur. En conséquence, le pixel souffre d'un faible facteur de remplissage et sa surface est trop grande pour considérer des rétines de haute résolution.

Une autre solution proposée par Arias Estrada [7] consiste à utiliser un capteur CMOS standard en l'interfaçant à des opérateurs câblés à base de circuits en technologie FPGA. L'inconvénient de cette solution est le goulot d'étranglement concernant le flux de données entre le capteur CMOS et les circuits de traitement. Cette solution nécessite des méthodes strictes pour passer de l'algorithme à sa mise en oeuvre matérielle sur les circuits de traitements FPGA ou DSP.

Le capteur que nous avons choisi présente un concept d'architecture faisant un compromis entre la versatilité et le parallélisme. Notre solution vise à prendre en compte le temps de réponse des algorithmes avec une résolution du capteur importante, tout en réduisant la consommation en énergie pour des raisons d'embarquabilité.

L'approche que nous allons présenter se situe entre les deux approches évoquées précédemment (celle de P. Dudek et de A. Estrada). Elle consiste à placer les principaux opérateurs, habituellement intégrés dans le pixel, aux abords de la matrice. En conséquence, les fonctions de ces opérateurs sont partagées par un groupe

de pixels (ligne ou colonne). Ces opérateurs sont exécutés en parallèle dans ce groupe et séquentiellement sur l'ensemble des groupes. Cette architecture se présente sous forme d'une matrice de pixels associée à un vecteur de processeurs analogiques/numériques mixtes. Chaque processeur peut effectuer, in situ, des traitements de bas niveau. L'avantage certain de cette solution est d'augmenter le facteur de remplissage des pixels et d'accroître la complexité des opérateurs à intégrer sans incidence sur le Fill Factor.

Nous avons mis en œuvre deux systèmes prototypes qui ont constitué la base de notre évaluation architecturale. L'un des prototypes se base sur le circuit PARIS1 et l'autre se base sur un capteur logarithmique CMOS APS. Ces deux systèmes ont été sujets d'une évaluation métrique. Pour avoir des comparaisons raisonnables, nous avons mis en œuvre les différentes architectures utilisant le même processeur numérique dédié pour le calcul et les traitements haut niveau. Cette condition nous a permis d'avoir la même base pour des comparaisons au niveau de l'étiquetage temporel et les durées de traitements des algorithmes testés.

2. L'ARCHITECTURE PARIS

2.1 Architecture générale

La schématique générale de l'architecture PARIS est représentée sur la figure 1. Elle est constituée d'une matrice de pixels passifs et d'un vecteur de processeurs mixtes analogiques/numériques qui forment les éléments de traitement partagés par l'ensemble des pixels [8][9][10]. De ce fait, la résolution de la matrice reste indépendante du type des traitements implantés au sein des processeurs et de leurs complexités.

Notre objectif est de concevoir une architecture d'un système de vision basé sur un capteur qui peut multiplexer spatialement les opérateurs de traitement et temporellement les différents traitements. Nous avons mis notre choix d'évaluation sur ce type d'architecture puisqu'il se base sur des processeurs conçus sur le modèle des unités arithmétiques et logiques intégrées au sein des processeurs de traitement de signal (DSP) et accessible à un grand nombre de pixels ou groupes de pixels. Cette architecture semblable à celle de R.Etienne Cumming [11] présente la possibilité de rééquilibrer les performances de calcul et de communication au cœur du capteur en programmant une exécution partiellement séquentielle au niveau de la lecture de la matrice et de l'exécution d'un programme.

Les pixels présentent eux même une architecture qui permet la mémorisation des valeurs des photodétecteurs ainsi que des résultats de traitements intermédiaires. Les processeurs mixtes A/N quant à eux présentent une architecture similaire à celle des ALUs. Ils permettent de réaliser une grande variété de traitements d'images bas niveau. Les calculs effectués par ces processeurs sont déportés sur la colonne correspondante en plus des deux colonnes voisines. Ce déportement de calcul au bord de la matrice, sans qu'il y ait une séparation totale entre les

pixels et les opérateurs de traitements, permet de réduire les dimensions des pixels, d'augmenter le Fill Factor et de concevoir des rétines à grande résolution.

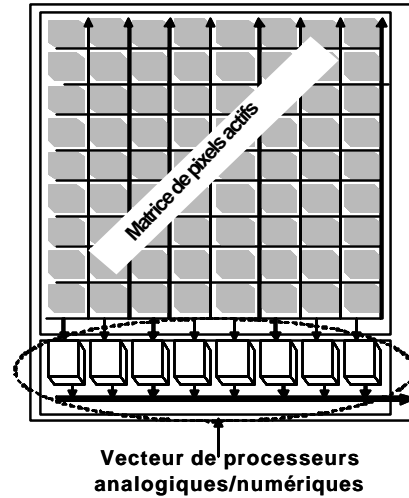


Figure 1 : Schématique générale de l'architecture PARIS

Afin d'associer l'efficacité et la compacité du pixel, les éléments d'acquisition et de mémorisation ont été regroupés. Le pixel comporte donc un photodétecteur, trois éléments de mémorisation et un dispositif de lecture (un amplificateur du signal de lecture et un multiplexeur pour adresser les différentes mémoires). L'architecture du pixel est détaillée dans la figure 2.

Le photodétecteur est constitué de deux transistors bipolaires verticaux associés en parallèle formant un phototransistor. Cette disposition permet d'augmenter la sensibilité et de conserver une large bande passante. La surface photosensible occupée est de 11% de la surface totale du pixel.

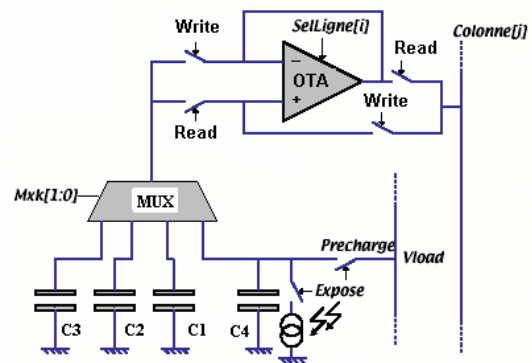


Figure 2 : Schéma du pixel de l'architecture PARIS

2.2 Architecture du processeur mixte

La matrice des pixels est associée à un vecteur de processeurs mixtes analogiques/numériques. Chaque processeur intègre :

- Un multiplexeur bidirectionnel qui assure l'accès aléatoire aux pixels de trois colonnes successives.
- Une unité de calcul analogique (UCA) pour réaliser

les combinaisons des opérations élémentaires (addition, soustraction et multiplication accumulation) nécessaires pour un traitement analogique.

- Un comparateur (CMP) à seuil programmable qui permet d'obtenir des informations booléennes.
- Une unité booléenne (UB) qui peut traiter les résultats issus du comparateur.
- Des registres pour mémoriser les résultats intermédiaires au niveau du processeur mixte.

Le schéma de la figure 3 représente l'architecture d'un processeur mixte et l'interaction de chaque processeur avec la matrice des pixels à travers le multiplexeur intégré dans le processeur. Un deuxième multiplexeur est intégré pour choisir le type de sortie du processeur (sortie de l'UCA, sortie de l'UB ou sortie du comparateur).

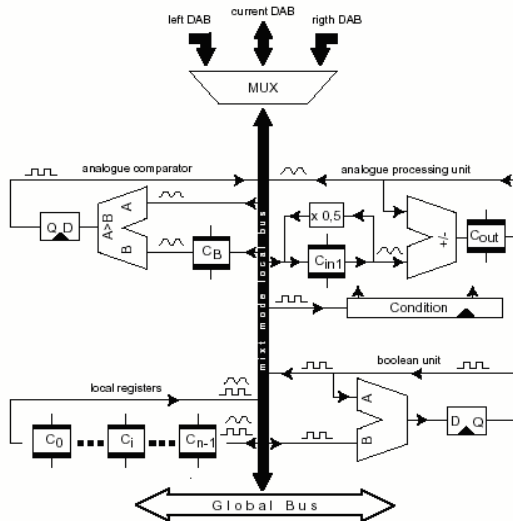


Figure 3 : Architecture du processeur mixte

L'architecture de l'unité de calcul analogique (UCA) est schématisée dans la figure 4. Pour effectuer les instructions relatives aux opérations d'addition, de soustraction et de multiplication, trois capacités et un jeu d'interrupteurs sont intégrés dans l'UCA. Deux capacités d'entrée de la même valeur permettent, par la mise en court-circuit de l'une d'elles, puis par la répartition des charges entre elles, d'effectuer des divisions par deux. La capacité de sortie, de même valeur que celles d'entrée, accumule les charges transférées grâce à un l'amplificateur OTA. Le jeu d'interrupteurs permet d'effectuer ce transfert de charge de façon à ce que les charges s'additionnent ou se retranchent à celles de la capacité de sortie.

Les pixels issus d'une seule ligne sont simultanément traités par l'unité de calcul analogique (UCA), l'unité booléenne (UB) ou le comparateur (CMP). Ensuite ces lignes sont transférées colonne par colonne. Le résultat du calcul est dirigé, via un multiplexeur, vers un registre, pour qu'il soit réutilisé pour un nouveau calcul, stocké dans l'un des plans mémoires ou transmis sur le bus de sortie.

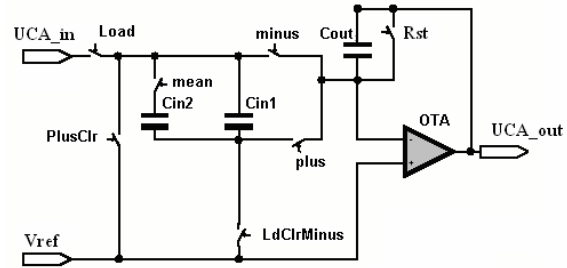


Figure 4 : Architecture de l'unité de calcul analogique

Le circuit PARIS1 est un premier prototype conçu en technologie CMOS 0.6µm [12]. Il a une résolution de 16x16 pixels aléatoirement adressables. La matrice de pixels est associée à un vecteur de 16 processeurs fonctionnant en mode analogique. Ce premier circuit permet d'évaluer les fonctions des opérateurs intégrés à travers certains algorithmes de traitements d'images bas niveau tel que la détection de contours ou de mouvement. Les processeurs analogiques intégrés exécutent les opérations ou les instructions d'un programme déroulé par un séquenceur.

3. ARCHITECTURE DU SYSTEME

L'avantage des rétines est la capacité d'intégrer des circuits additionnels sur la même puce aux abords ou bien au sein même des pixels. Les microprocesseurs RISC, quant à eux, sont devenus de plus en plus répandus, leur coût a diminué et leur capacité de calcul a augmenté. Ceci rend possible l'exécution, sur ces microprocesseurs, de certains traitements des valeurs de pixels "à la volé". De plus, l'intégration de ce type de microprocesseurs avec une rétine permettra de faciliter la gestion d'un tel circuit. Ceci suggère qu'il est possible d'associer un circuit rétine à un microprocesseur RISC pour former un dispositif monopuce.

Pour notre application, nous avons choisi une carte à base d'un microprocesseur ARM [13]. La logique est d'utiliser ce microprocesseur en y ajoutant les fonctionnalités nécessaires pour concevoir une architecture d'un système de vision SoC (l'ensemble des parties d'un tel système peut être intégré dans une puce unique).

Afin d'évaluer donc un système dont l'architecture est basée essentiellement sur la rétine PARIS1 et un processeur de type ARM et lorsque les critères de tailles mémoires ou de vitesses d'exécution sont déterminantes, l'utilisation d'un composant 16 ou 32 bits est particulièrement adaptée. Nous avons choisi une carte intégrant un processeur ARM7TDMI 16/32 bits à basse consommation et qui opère à une fréquence de 50 MHz.

La figure 5 présente, dans une forme simplifiée, l'architecture résultante de l'interfaçage PARIS1-ARM. Dans cette architecture, les traitements bas niveau sont effectués par la rétine. Les traitements haut niveau sont effectués par le microprocesseur ARM. L'avantage de ce type d'architecture demeure dans l'exécution parallèle d'un nombre conséquent d'opérations de bas niveau dès

le pixel en intégrant des opérateurs partagés entre groupes de pixels, ce qui permet d'économiser des ressources de calculs, très coûteuses, et de faire une certaine économie d'énergie. Cette structure est plus avantageuse qu'une solution à base d'un imageur CCD associé à un microprocesseur en terme de puissance de calcul.

Pour évaluer cette architecture, nous avons mis au point un prototype d'instrumentation. Le système est constitué de trois parties : une carte mère (dont le contrôleur intègre le microprocesseur ARM) produisant les signaux de contrôle et une carte fille intégrant un module de conversion A/N et le circuit PARIS1 équipé d'une optique (figure 6).

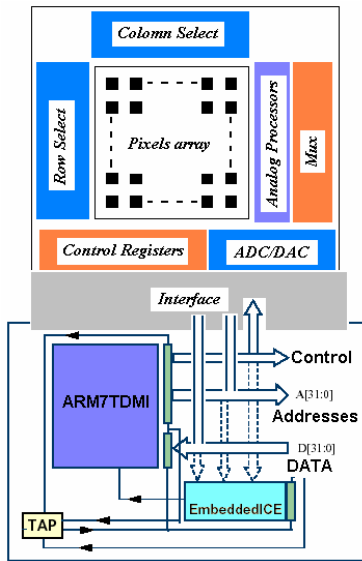


Figure 5 : Architecture du système PARIS1-ARM



Figure 6 : Système de vision prototype PARIS1-ARM

4. ÉVALUATION

4.1 Contrôle automatique du temps d'exposition

Pour un système de vision, la variation de certains paramètres cause des non-uniformités lors de l'étape d'acquisition. Ces effets aléatoires sur le plan focal pourraient projeter de fausses interprétations sur les résultats du traitement ultérieur. Une seule calibration de

ces paramètres est inappropriée lorsqu'un tel phénomène varie au cours du temps. Ainsi le calibrage d'un système de vision devient un facteur important pour le traitement de l'information.

Le temps d'exposition est l'un des paramètres clés dans beaucoup d'applications impliquant la vision. Nous avons mis au point un algorithme de contrôle automatique du temps d'exposition pour notre système. Ceci évite la saturation des pixels et donne une amplification adaptative de l'image, nécessaire pour avoir une bonne dynamique. Le système peut alors s'adapter aux conditions d'éclairément et peut ainsi acquérir les séquences indépendamment de l'éclairément de la scène.

Le concept du calibrage est basé sur le fait que les photodiodes sont utilisées dans un mode d'intégration. Un éclairément donné mène à une variation de tension qui change suivant le temps d'exposition. Si la luminosité est élevée, le temps d'exposition doit diminuer, par contre si la luminosité est faible le temps d'exposition devra augmenter. Par conséquent, plus le temps d'exposition est court plus le gain de temps pour le traitement d'images est grand.

Nous avons effectué plusieurs mesures avec notre système de vision, de sorte à établir un algorithme de contrôle automatique du temps d'exposition en fonction de la luminosité de la scène. La figure 7 présente la variation du maximum du niveau gris en fonction du temps d'exposition.

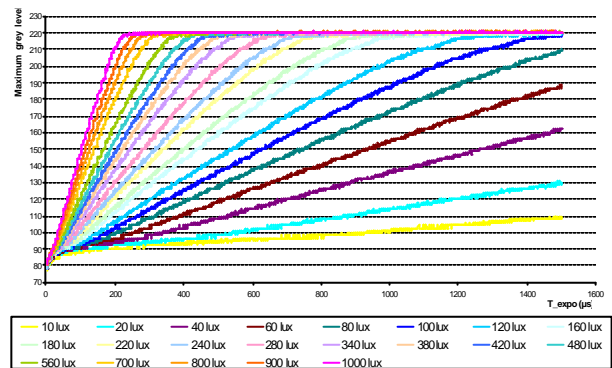


Figure 7 : Evolution du maximum du niveau de gris en fonction du temps d'exposition

Chaque courbe, présente une zone linéaire et une zone de saturation. Ainsi nous déduisons la variation du gradient ($dMax/dt$) des courbes en fonction de la luminosité. La deuxième courbe de la figure 8 donne alors une variation quasi-linéaire (la droite approximée est obtenue par régression linéaire de la courbe expérimentale). L'algorithme consiste à maintenir le temps d'exposition dans l'intervalle où les différentes courbes sont linéaires et où le temps d'exposition est le plus petit. La commande est alors initialisée par un temps d'exposition appartenant à cet intervalle. Quand un maximum est mesuré, la luminosité correspondante est déduite et le contrôle automatique consiste en une

boucle d'asservissement qui ajuste le temps d'exposition relativement à la luminosité. La figure 9 représente une séquence d'images montrant l'adaptation du temps d'exposition à la luminosité.

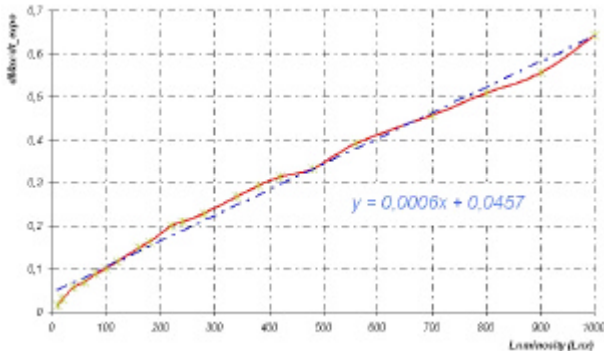


Figure 8 : Rapport dMax/dt_expo en fonction de la luminosité

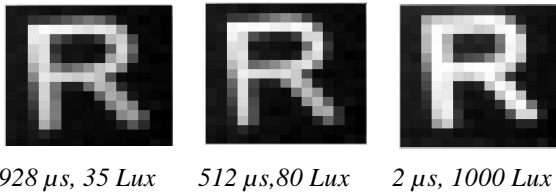


Figure 9 : Séquence montrant l'adaptation du temps d'exposition à la luminosité

4.2 Traitement d'images sur puce

Nous avons évalué des algorithmes de traitements d'images dont nous avons conclu l'avantage de notre architecture PARIS1-ARM. Ces algorithmes sont implantés complètement au sein des unités de calculs analogiques. Ces algorithmes sont de type : convolution, filtrage linéaire, détection de contours et détection de mouvement...etc. Quelques exemples sont présentés dans ci-dessous. Les images sont traitées avec différentes valeurs de luminosité en utilisant le contrôle automatique du temps d'exposition. La figure 10 donne des exemples des images traitées dans une gamme de luminosité de [10Lux, 1000Lu x].

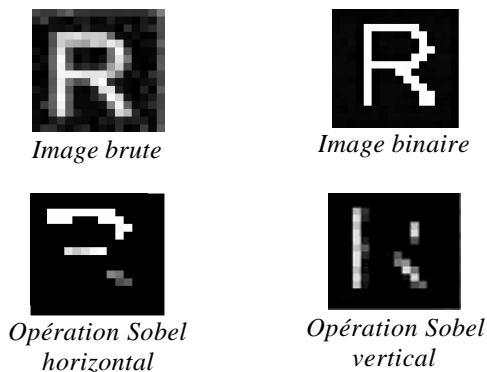


Figure 10 : Exemples d'images traitées utilisant le contrôle du temps d'exposition

4.3 Comparaison avec d'autres architectures

Nous avons concentré les comparaisons sur plusieurs aspects (temps de traitements, consommation d'énergie, complexité, programmabilité...) [14][15].

Ces paramètres s'avèrent importants pour un système SoC vu les raisons exigées d'embarquabilité et de temps réel. Cependant, les comparaisons effectuées restent relatives au type d'architecture évaluée et notre intérêt est de formuler une idée générale sur la différence de comportement entre un système à base d'une rétine tel que PARIS1 et un autre basé sur un capteur sans aucun opérateur de traitement intégré.

Nous avons implanté plusieurs filtres sur un système FUGA-ARM (système à base d'un capteur APS à réponse logarithmique et un processeur ARM) ainsi que sur un système commercial ETHERCAM (système à base d'un capteur APS à réponse linéaire et un processeur ARM). Les mêmes filtres ont été implantés sur le modèle PARI1-ARM avec les codes optimisés correspondants. Pour les deux premiers systèmes, les traitements sont faits sur le processeur ARM, pour le système PARIS1-ARM, les traitements sont faits sur le vecteur de processeur analogique du circuit PARIS1.

Comme exemple, nous avons implanté, sur le modèle FUGA-ARM, un filtre passe bas comme celui implanté sur la rétine PARIS1. La figure 11 donne une comparaison des temps de traitement faits sur les deux architectures. Les temps de calcul pour le circuit PARIS1 ont été calculés à partir des résolutions inférieures ou égales à 16x16 pixels et par extension des calculs sur des matrices de grandes tailles [10].

Au moment où les temps de calculs sont largement importants et croissent d'une manière quadratique pour l'architecture FUGA-ARM, ils sont faibles pour l'architecture basée sur le circuit PARIS1 et croissent d'une manière linéaire. Dans cet exemple, le temps de calcul pour le circuit PARIS1 est de 100µs par ligne de pixels. Ce gain en temps de traitement est dû à la forme du parallélisme réalisé grâce au vecteur de processeurs analogiques intégrés au sein de la rétine PARIS1. Cette architecture permet non seulement de réduire les temps de calculs, mais de fournir l'information recherchée, même de bas niveau, et allège du coup l'ensemble des traitements faits sur le processeur principal du système.

Nous avons calculé le CPP (Cycle par Pixel) de chaque traitement et sur différentes architectures. La figure 12 regroupe les courbes d'évolution du CPP en fonction de la résolution de l'image.

Alors que le CPP des architectures basées sur un capteur APS et un processeur reste relativement constant pour toutes les résolutions, le CPP de l'architecture basée sur PARIS1 débute avec des valeurs largement grandes pour des faibles résolutions mais décroît pour se maintenir constant à une valeur plus petite que celle des autres architectures pour les grandes résolutions.

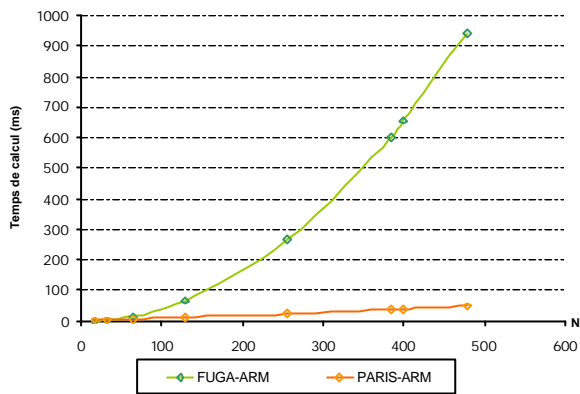


Figure 11 : Temps de calcul d'un filtre passe bas pour les deux architectures FUGA-ARM et PARIS1-ARM

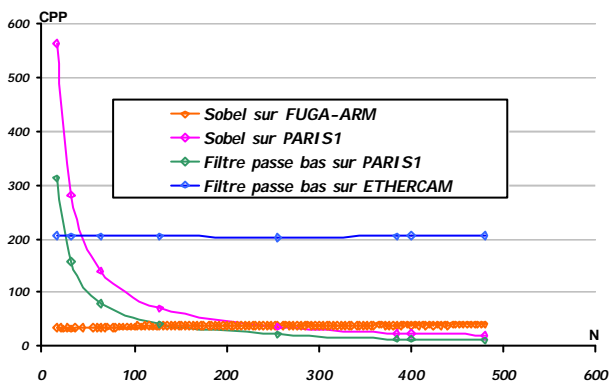


Figure 12 : CPP des différents filtres pour différentes architectures

5. CONCLUSION

La variété des évaluations effectuées sur différents modèles de vision a permis de mettre en relief les avantages d'utilisation des rétines comme capteurs intelligents constituant un module parmi l'ensemble des modules d'un système. Certes, la réalisation d'un système basé sur une rétine nécessite une définition d'une architecture appropriée et le développement de librairie de traitements d'images spécifiques relativement à l'architecture du capteur lui-même, cependant la méthodologie suivie montre la capacité des rétines pour devenir des candidats potentiels afin de concevoir des systèmes de vision monopuce à haute résolution. Les performances qu'elles présentent en terme de temps d'exécution, de consommation d'énergie et de programmabilité, les rendent une solution prometteuse pour une adéquation algorithme architecture.

REFERENCES

[1] Alireza Moini, "Vision chips or seeing silicon" Technical Report, Centre for High Performance Integrated Technologies and Systems, The

University of Adelaide, March 1997. Kluwer Academic Publishers, ed. I. 0-7923-8664-7. 2000.

[2] Abbas El Gamal, and al, "Pixel Level Processing: - Why, what and how?" SPIE Vol.3650, 1999, pp. 2-13.

[3] R. Burns, C. Thomas, P. Thomas, R. Hornsey "Pixel-parallel CMOS active pixel sensor for fast objects location", SPIE International Symposium on Optical Science and Technology, 3 - 8 Aug. 2003, San Diego, CA USA.

[4] Y. Ni, J.H Guan "A 256x256-pixel Smart CMOS Image Sensor for Line based Stereo Vision Applications", IEEE, J. of Solid State Circuits, Vol. 35 No. 7, Juillet 2000, pp. 1055-1061.

[5] A. Elouardi, S. Bouaziz, A. Dupret, L. Lacassagne, J.O. Klein, R. Reynaud "Evaluation of an artificial CMOS retina sensor for tracking systems". Proceeding of IEEE Intelligent Vehicles Symposium, June 2002, Versailles, France.

[6] P. Dudek, "A programmable focal-plane analog processor array" Ph.D. University of Manchester Institute of Science and Technology (UMIST), May 2000.

[7] Miguel Arias-Estrada. "A Real-time FPGA Architecture for Computer Vision", Journal of Electronic Imaging (SPIE - IS&T), Vol. 10, No. 1, January 2001, pp. 289-296.

[8] A. Dupret, J.O. Klein et A. Nshare "A DSP-like Analogue Processing Unit for Smart Image Sensors" International Journal of Circuit Theory and Application, 2002, vol 30, pp. 595-609.

[9] J.O. Klein, A. Dupret, S. Moutault, A. Nshare "Vers une nouvelle génération de rétines programmables". GRETSI 99, Septembre 1999, Vannes, France.

[10] A. Dupret, J.O Klein, A. Nshare "A programmable vision chip for CNN based algorithms" CNNA 2000, Catania, Italy: IEEE 00TH8509.

[11] R. Etienne-Cummings and Z.K. Kalayjian "A Programmable Focal-Plane MIMD Image Processor Chip". IEEE Journal of Solid-State Circuit, 2001. 36 no 1: p. 64-72.

[12] A. Nshare "Définition et conception d'une nouvelle génération de rétines programmables". Thèse de doctorat de l'université Paris Sud, septembre 2002.

[13] ARM System-on-Chip Architecture (2nd Edition), Steve Furber, Septembre 2000.

[14] A. Elouardi, S. Bouaziz, A. Dupret, J.O. Klein, R. Reynaud, "On Chip Vision System Architecture Using a CMOS Retina". Proceeding of IEEE Intelligent Vehicle Symposium, IV'04. Pages 206-211. ISBN 0-7803-8311-7. June 14-17, 2004. Parma, Italy.

[15] A. Elouardi, S. Bouaziz, A. Dupret, L. Lacassagne, J.O. Klein, R. Reynaud, "CMOS Image Sensor versus Retina Experience". Proceeding of IEEE Sensors '04, ISBN 0-7803-8693-0. October 24-27 2004, Vienna, Austria.