

Méthode de conception de l'architecture d'un capteur de vision dédié au contrôle qualité

Delphine EMZIVAT, Claude GAGNADRE, Eric MARTIN

Laboratoire LESTER
Centre de Recherche, rue St MAUDE 56100 LORIENT

`delphine.emzivat@univ-ubs.fr, gagnadre@univ-ubs.fr, emartin@univ-ubs.fr`

Résumé- Aujourd'hui l'optique et l'électronique modernes permettent d'envisager l'intégration sur un même circuit des capteurs pour l'acquisition et des processeurs pour le traitement d'images. De tels dispositifs constituent un élément essentiel des recherches actuelles sur les capteurs de vision. Notre but est de concevoir et de valider une méthodologie pour l'adéquation entre algorithmes de traitements d'images et architectures de capteur et de processeur de traitement. L'optimisation recherchée tout au long de notre étude vise à réaliser l'architecture de traitement juste nécessaire aux besoins de l'algorithme de traitements d'images.

Abstract – Today modern optics and electronics make it possible to consider integration of the sensors for acquisition and the processors for the image processing on a same circuit. Such devices constitute an essential element of current research on the sensors of vision. Our aim is thus to formalise and suggest a methodology which provides the complete optimised architecture for a given algorithm. The sought optimisation consists of carrying out the just necessary processing architecture to the image processing algorithm needs.

1. Introduction

Les dispositifs qui admettent sur le même circuit intégré VLSI les fonctions de saisie de l'image et de son traitement, constituent un élément essentiel des recherches actuelles sur les capteurs de vision. Avec l'ajout de fonctions de communication numérique via un réseau local on parle alors de capteur intelligent ou bien encore de Smart Sensor. Selon le domaine d'application, des besoins spécifiques peuvent apparaître qui se traduisent alors par une mise en œuvre de l'acquisition temps réel plus ou moins complexe et par des puissances de calculs plus ou moins importantes. Notre projet s'intègre dans le cadre d'un contrôle qualité industriel pour vérifier par exemple sur une chaîne de production le mouvement ou bien encore la forme supposée simple de produits manufacturés (biscuits, pièces métalliques...). Les informations recherchées sont de qualité réduite c'est-à-dire que le degré de précision requis est relativement faible. Notre but est de formaliser et de proposer une méthodologie qui fournit pour un traitement donné et en parfait accord avec les exigences requises par le demandeur l'architecture complète optimale, à savoir : le nombre de cellules photosensibles constituant la partie optique de notre circuit, la charge de travail de chaque processeur, la capacité mémoire, le nombre de bus d'interconnexions nécessaires. L'objectif majeur de notre travail réside dans l'obtention d'une parfaite adéquation algorithme/architecture qui permet la réalisation d'un système performant et peu coûteux. Il s'agit en fait d'obtenir un capteur intelligent juste «suffisant» pour l'application envisagée. Au cours de cet article, nous présentons dans un premier temps le contexte d'utilisation et les caractéristiques principales requises pour notre rétine. Puis nous décrivons la méthodologie développée dans le cas bien précis d'une estimation de mouvement déterminée à partir du calcul du coefficient d'intercorrélation. Enfin nous présentons les résultats obtenus lors de nos simulations.

2. Démarche entreprise

2.1. Principe de l'acquisition

Pour la conception de notre rétine, des contraintes de temps, de performance ainsi que des contraintes d'intégration guident notre choix. Notre capteur ayant pour application l'inspection et le contrôle qualité, les critères de qualité établis s'avèrent très subjectifs et sont définis à partir de la nature de l'image à traiter et du domaine de travail considéré. Nous devons donc pour chaque application envisagée étudier, définir et évaluer les caractéristiques à considérer. Les modèles de capteurs de vision intelligents apparaissent en général sous forme de tableaux bidimensionnels : il s'agit dans bien des cas de matrices de processeurs élémentaires [1,2] chacun incluant un dispositif photosensible tels que photodiodes ou phototransistors. Le problème essentiel rencontré par cette classe de capteurs réside dans la valeur relativement faible du facteur de remplissage. En effet, les implantations sur le même support de silicium et au sein d'une même cellule du capteur et du processeur conduisent à des dispositifs où la surface recevant effectivement la lumière se révèle restreinte par rapport à la surface totale du circuit. Ainsi pour pallier cet inconvénient, le système d'acquisition intégré dans notre prototype est constitué de deux réseaux linéaires de photocapteurs disposés orthogonalement (fig.1). Cette disposition permet de limiter le nombre de cellules photosensibles utilisées, l'image étant formée par le mouvement de la scène elle-même. Ceci se révèle possible étant donné le type des algorithmes ciblés. En abaissant le nombre de pixels acquis, on réduit ainsi le coût de revient de la partie optique. On décide de privilégier la simplicité au niveau du dispositif optique en dirigeant l'ensemble des données recueillies vers des modules à grande capacité de mémorisation.

On peut démontrer que l'association des informations reçues par les deux barrettes à l'instant t que l'on note sous la forme (x,t) et (y,t) fournit les mêmes résultats que les données bidimensionnelles (x,y) issues d'une matrice 2D pour une même application. Ceci suppose l'existence d'un déplacement de la part de l'objet et l'invariance de la forme de celui-ci dans les différentes images acquises. Un objectif adapté permet de projeter sur nos deux barrettes linéaires l'image de l'objet en mouvement qui se situe partiellement ou entièrement à l'intérieur du domaine d'observation.

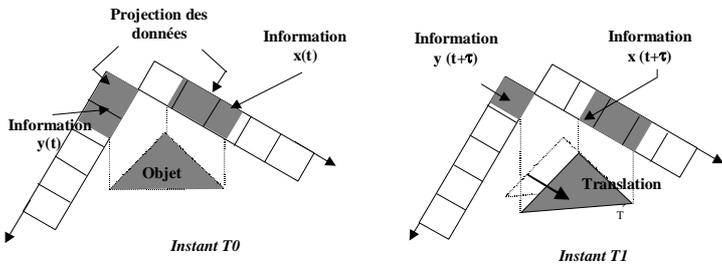


FIG.1 : Evaluation de l'intensité

L'idée poursuivie est de travailler sur des applications dont le comportement est déterministe. Ainsi, si on peut caractériser les mouvements "normaux", il sera alors possible de détecter une rupture dans le déplacement ou une erreur sur la forme de l'objet. Les objets sont constitués d'un ensemble de points dans l'image qui se déplacent dans la scène. La projection de l'image va débiter dès lors que l'objet apparaît dans l'espace recouvert par notre réseau de barrettes linéaires. Plus l'objet sera présent au sein du champ d'observation et plus l'intensité lumineuse va croître jusqu'à atteindre une valeur constante qui sera conservée tant que l'objet figurera intégralement dans la scène à observer. Une décroissance s'opère lorsque l'objet quitte progressivement l'espace d'observation. Tout ceci est illustré par les schémas de la figure 2.

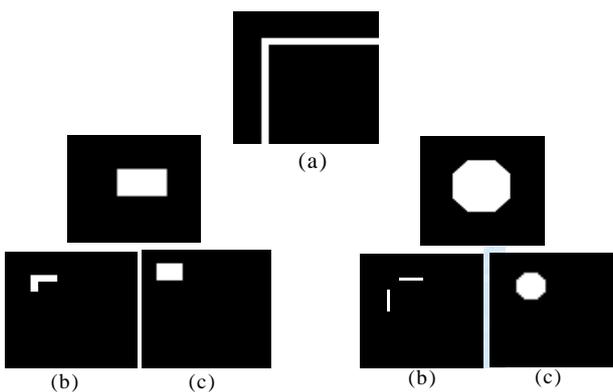


FIG 2 : (a): Représentation de la rétine
(b)/(c): Reconstruction à l'instant t de l'image de l'objet en mouvement qui se situe partiellement (b) ou entièrement (c) dans le champ d'observation

2.2. Principe du traitement

La conception optimisée d'une architecture spécifique pour un algorithme donné nécessite de suivre une certaine démarche méthodique. Nous avons ainsi développé une méthodologie qui permet d'obtenir l'adéquation algorithme-architecture qui limite cette dernière aux besoins de l'algorithme et qui tient compte du compromis entre l'architecture du capteur et celle du traitement.

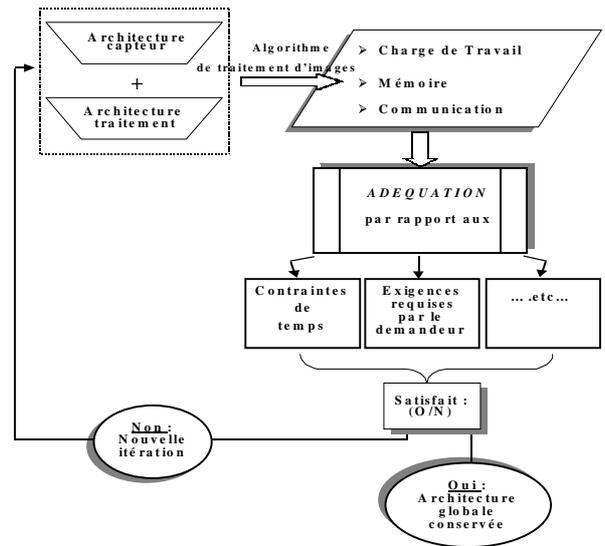


FIG. 3 : Principe de recherche de l'adéquation algorithme/architecture

Pour une architecture donnée de notre capteur, on va procéder à une simulation de la partie traitement qui va fournir comme résultat la charge de travail, la capacité mémoire, les besoins en communications. Si la forme des objets recherchés se révèle inexacte ou bien le temps de calcul incompatible par rapport au rythme d'acquisition, on recherche alors une nouvelle architecture du capteur et du traitement et on réitère l'approche. Cet enchaînement itératif des différentes étapes (fig.3) se déroule jusqu'à l'obtention d'un parfait équilibre entre les ressources matérielles nécessaires, les contraintes exigées et les résultats obtenus. Notre travail sera dédié dans un premier temps à l'estimation de mouvement. On considère dans notre cas qu'une image est correcte lorsqu'elle contient suffisamment d'informations significatives de la scène observée. Diverses techniques d'estimation existent [3,4,5], chacune reposant sur des approches spécifiques. Ainsi pour illustrer l'intérêt de notre méthodologie nous avons recours aux méthodes basées sur la mesure de la corrélation que l'on applique soit à deux images successives soit à l'image courante et à une image de référence. Ce principe de calcul est bien connu pour son efficacité dans l'estimation de mouvement à condition que certaines précautions d'emploi soient respectées (comme l'invariance de l'intensité lumineuse). Le coefficient d'autocorrélation dans le cas d'un mouvement de translation selon l'axe horizontal est défini par l'expression suivante:

$$C_{xx}(\tau) = \vec{X}(d) \cdot \vec{X}(d + \tau)$$

$$\text{soit } C_{xx}(\tau) = \sum_{j=0}^{N-1} X_j^T \cdot X_{j-\tau}^{T_0}$$

avec $\left\{ \begin{array}{l} T_1 = \text{ins tan t courant} \\ T_0 = \text{ins tan t précédent} \\ N = \text{nombre de cellules élémentaires de notre barrette} \\ X = \text{Valeur de l'intensité projetée sur l'axe} \\ d = \text{position} \end{array} \right.$

Chaque élément de l'architecture se révèle être un véritable processeur disposant d'une unité de traitement autonome. Notre dispositif de traitement est partagé en deux réseaux linéaires [6] de processeurs (fig.4), chacun se chargeant de recueillir et de gérer les données issues respectivement des projections sur la barrette horizontale ou verticale.

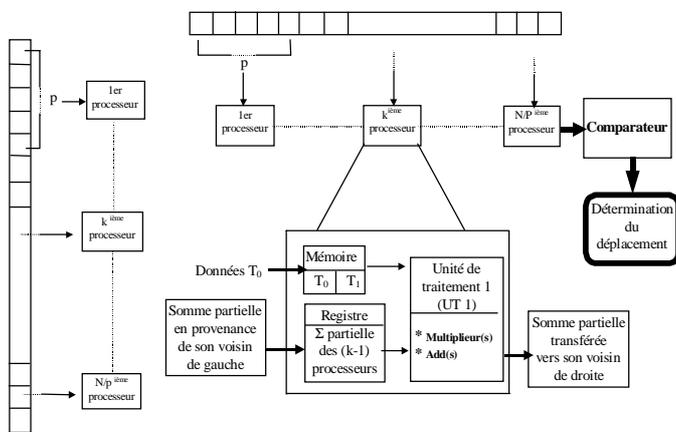


FIG 4 : Architecture requise dans le cadre du calcul du coefficient d'intercorrélacion

Le calcul est distribué à tous les processeurs constituant le réseau. L'algorithme relatif à la corrélation se caractérise par une grande simplicité et une grande régularité des calculs. On procède, sur chaque processeur k du réseau de traitement, au parallélisme des calculs relatifs aux p données T_1 qui lui sont dédiées. On a ainsi :

PARALLELE for all $k = 1, \dots, N/p$

For τ

$$C_{xx}^k(\tau) = 0$$

For $j \in [np; (n+1)p]$

$$C_{xx}^k(\tau) = C_{xx}^k(\tau) + X_j^T \cdot X_{j-\tau}^{T_0}$$

SEQUENTIEL for $k = 1, \dots, N/p$

$$C_{xx}(\tau) = 0$$

$$C_{xx}(\tau) = C_{xx}(\tau) + C_{xx}^k(\tau)$$

$$\text{Max } [C_{xx}(\tau)] \Rightarrow \text{Motion } \tau$$

Si l'algorithme d'autocorrélacion s'avère simple, sa mise en œuvre parallèle sur un tel réseau de processeurs implique des mécanismes de transfert et de partage de données relativement complexes, qu'il s'agit d'optimiser.

2.3. Adéquation Algorithme-Architecture

Une des étapes de conception pour passer d'un algorithme de traitement du signal à sa réalisation pratique est la synthèse architecturale. Cette étape est réalisée en procédant à la description comportementale de notre algorithme via l'outil de synthèse de haut niveau GAUT [7,8,9], logiciel développé au sein de notre laboratoire et qui comporte plusieurs modules. La première étape consiste à écrire notre algorithme dans un langage de haut niveau (dans notre cas en VHDL) tout en imposant un temps de calcul (i.e. une cadence) qui correspond à la contrainte temporelle de l'application. Le module «sélection» intégré dans l'outil permet d'optimiser la surface du circuit par la meilleure sélection d'opérateurs issus de la librairie. On rappelle que les outils de synthèse haut niveau permettent d'accéder à une description RTL du circuit à partir d'une spécification comportementale de l'application. Cette description RTL permet ensuite d'aboutir à un circuit VLSI à l'aide d'outils de synthèse logique.

La flexibilité de l'architecture consiste à faire varier le nombre de processeurs afin d'acquérir une architecture qui ait certes un moindre coût mais des performances justes suffisantes. Au niveau des contraintes relatives à notre application, seule la vitesse de l'objet est fixée avec :

$$\text{Vitesse}_{\text{Max}} = \frac{\tau_{\text{Max}}}{\text{Cadence de l'algorithme}}$$

où cadence = nombre d'itération de notre algorithme

L'adéquation recherchée entre l'algorithme et la surface de traitement consiste non seulement à décliner l'ensemble des valeurs prises par τ dans l'intervalle $[\tau_{\text{min}}; \tau_{\text{max}}]$ mais également à modifier la cadence de l'application tout en respectant la vitesse imposée. Ainsi si on augmente la largeur de l'intervalle de déplacement $[\tau_{\text{min}}; \tau_{\text{max}}]$, la cadence imposée sera plus importante elle aussi afin de respecter la vitesse imposée lors de notre application, ce qui se traduira par une croissance de la surface de traitement. De même l'élément p qui est le nombre de rétines dédiées par processeur influe sur les architectures du capteur et du processeur de traitement. De ce fait l'optimisation recherchée entre les deux architectures consiste à retenir une répartition correcte des surfaces illuminées par rapport à la surface dédiée au traitement. Si on augmente cet indice p, on réduit la cadence imposée à chaque processeur élémentaire ce qui entraîne une diminution de la surface de traitement proposée par l'outil de synthèse. Mais cette surface doit correspondre sensiblement à la surface de chaque élément photosensible. Au vu de ces différentes constatations, nous allons procéder à une exploration des triplets $(\tau, \text{cadence}, p)$ que l'on soumet à l'entrée de notre outil de synthèse architecturale.

$$\left\{ \begin{array}{l} \tau \\ p \end{array} \right\} \begin{array}{l} \longleftrightarrow \text{cadence} \\ \implies \end{array} \text{GAUT} \implies \text{Surface de traitement}$$

Nous avons traduit en langage VHDL l'algorithme relatif à notre estimation de mouvement. Nous obtenons en sortie la surface requise pour le traitement avec notamment la liste des

ressources nécessaires pour mener les calculs ainsi que la complexité en surface pour un processeur élémentaire. Si nous procédons à une normalisation par rapport à une surface photosensible unique on a une surface de traitement équivalente S/p . A une surface photosensible donnée correspond donc une surface de traitement S/p . La figure ci-dessous fait apparaître précisément qu'à l'issue de nos premières simulations et pour différents triplets fixés (τ , cadence, p) l'optimisation est acquise entre les deux surfaces, ceci hors logique de conversion pour l'acquisition et hors routage pour le traitement.

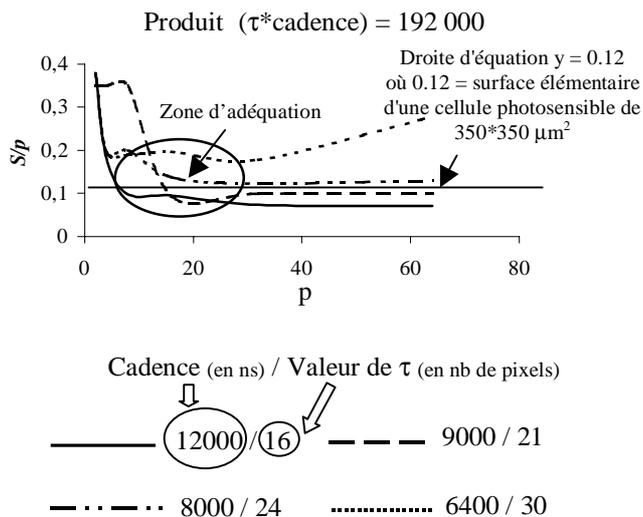


FIG 6 : Optimisation Surface Capteur / Surface Processeur

On observe donc une zone où logiquement la surface requise en regard de la surface de la cellule photosensible est en totale adéquation. Selon les triplets (τ , cadence, p) nous obtenons diverses configurations. Notre travail sera donc de choisir celles qui respectent au mieux les différentes fonctions coûts qui nous ont été imposées.

Il est à noter que les diverses courbes obtenues au cours de ce paragraphe ne concernent que le processeur seul. On va donc étendre ces résultats à l'ensemble du réseau de processeurs. Les simulations réalisées sont exploitées afin de tester différentes configurations et de valider outre l'algorithme, la réalisation pratique qui respecte au mieux les contraintes temps réel et qui minimise les ressources matérielles utilisées. Ceci nous conduit donc à la notion de reconfigurabilité qui constitue une nouvelle mise en œuvre de notre étude.

3. Conclusion

Nous avons décrit une méthode de conception d'architecture dans le cadre d'une estimation de mouvement. L'illustration proposée n'est autre qu'une estimation de mouvement basée sur la mesure du coefficient d'intercorrélacion. La finalité de notre travail est de formaliser la méthodologie qui consiste à obtenir une parfaite adéquation entre l'algorithme et l'architecture pour limiter la puissance de cette dernière aux besoins de l'algorithme. Ainsi la mise en œuvre pratique de cette méthodologie nous a permis d'atteindre un rapport

Surface Eclairée / Surface dédiée au traitement de 75% au lieu des 34% initial et ceci pour une même application. Il s'agira à présent de rendre applicable cette méthodologie à la plupart des algorithmes utilisés en contrôle qualité. Nous allons ensuite procéder à la validation du circuit, laquelle va confirmer ou au contraire infirmer nos choix de conception. Cette validation devra se faire préférentiellement dans l'environnement dans lequel le circuit sera amené à évoluer. On s'est ainsi orienté vers l'outil CADENCE qui offre un large choix de solutions pour la conception analogique, numérique et mixte.

4. Références

- [1] E Goujo. *Etude et réalisation d'un capteur d'images à temps d'exposition variable en technologie CMOS*. Thèse de l'université de Bourgoigne, Juin 1995.
- [2] J-E Eklund, C Svensson, A Aström. *VLSI Implementation of a Focal Plane Image Processor / A realization of the Near-Sensor Image Processing concept*. IEEE Transactions on VLSI Systems, Vol 4, N°3, September 1996.
- [3]. R.C Meitzler, A.G Andreou, K Strohhahn and R.E Jenkins. *A sampled-data motion chip*. 36th Midwest Symp. Circuits and Systems, Vol 1, pp 288-291.
- [4] C-H Hsieh, T-P lin. *VLSI Architecture for Block-matching Motion Estimation Algorithm*. IEEE Transactions on Circuits and Systems for Video Technology, Vol.2, No2, June 1992.
- [5] E.D Frimout, J.N Driessen, Ed.F Deprettere. *Parallel Architecture for a Pel-recursive Motion Estimation Algorithm*. IEEE Transactions on Circuits and Systems for Video Technology, Vol.2, No2, June 1998.
- [6] N.Ranganathan, N.Vijaykrishnan, N.bhavanishankar. *A linear Array processor with dynamic Frequency Clocking for Image processing Applications*. IEEE Transactions on Circuits and Systems for Video Technology, Vol.8, No4, August 1998.
- [7]. E. Martin, O. Sentieys, H. Dubois and J.L. Philippe. *GAUT, An architectural synthesis tool for dedicated signal processors*. Proceeding EURO-DAC93.
- [8] D.Emzivat, C.Gagnadre, E.Martin. *Optical Integrated Circuit for the Quality Control*. International Symposium on Microelectronic Manufacturing Technologies, 19-21 May 1999 EDIMBOURG
- [9] D.Emzivat, C.Gagnadre, E.Martin. *Vision Sensor for the Industrial Quality Control*. 7th International Conference on image on Processing and its Application, 12-15 July 1999 MANCHESTER (à paraître)