

Calcul Instantané d'une Transformée Spatiale d'Image pour l'Estimation du Mouvement par Recherche de Blocs de Pixels

Guy Cathébras, David Navarro

► **To cite this version:**

Guy Cathébras, David Navarro. Calcul Instantané d'une Transformée Spatiale d'Image pour l'Estimation du Mouvement par Recherche de Blocs de Pixels. JFAAA: Journées Francophones sur l'Adéquation Algorithme Architecture, Monastir, Tunisie. 2003. <lirmm-00269476>

HAL Id: lirmm-00269476

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00269476>

Submitted on 3 Apr 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Calcul instantané d'une Transformée Spatiale d'Image Pour l'Estimation du Mouvement Par Recherche de Blocs de Pixels

G. Cathébras, D. Navarro

Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

UMR UM2 / CNRS 5506

161 Rue Ada 34392 MONTPELLIER

navarro@lirmm.fr

Résumé

Cet article traite de l'intégration d'un traitement d'image dans un capteur en technologie CMOS standard. Ce traitement est une estimation de mouvement basée sur la mise en correspondance de transformées spatiales entre 2 images. Cette transformée est la Census Transform, qui permet de faciliter l'estimation du mouvement de chaque pixel entre deux images acquises successivement. Il s'agit d'un codage de la texture d'un voisinage de pixels (9x9) obtenu en comparant les luminosités de pixels proches. La Census Transform a déjà été implémentée à l'aide de FPGA [1] et informatiquement [2], en utilisant des caméras de type standard. Nous présentons ici une architecture permettant d'intégrer le calcul de cette transformée au sein même du capteur.

Abstract

This paper is devoted to the integration of an image processing in a standard CMOS sensor. This processing is a movement estimation, based on images' spatial transforms matching. That transform is called the Census Transform, which allows to estimate the pixels' movements between two successive images with more accuracy. It consists in coding the texture of a neighbouring of pixels (9x9), obtained by comparing the luminosity of close pixels. The Census Transform has already been implemented with FPAGA card[1] and with a computer [2], using standard image sensors. We present here an architecture that allows to integrate this transform in the sensor itself.

1. Introduction

Les capteurs d'images CMOS sont, de nos jours, des produits couramment utilisés, du fait de leur faible coût, de leur facilité d'utilisation, et de leur capacité à intégrer des fonctionnalités de traitement d'image dès le capteur. Cette intégration permet de développer des capteurs d'images spécifiques, effectuant des opérations de traitement bas niveau [3]. Nous nous focalisons ici sur une architecture où le traitement d'image est effectué au niveau des pixels, en utilisant la force de calcul des structures parallèles, dans le but d'obtenir une représentation du flot optique, à partir des deux images acquises successivement.

2. Codage de blocs de pixels pour l'estimation du mouvement

L'estimation de mouvement est un processus entraînant des calculs fortement itératifs. Calculs qui peuvent comprendre une phase de pré-codage (transformée locale) de l'image, facilitant la recherche de structures invariantes. La pertinence de tels algorithmes fait de ce type d'estimation du mouvement une solution robuste et utilisée.

Dans leur principe, ces algorithmes consistent à remplacer chaque pixel de l'image par une information extraite de son voisinage. Ainsi, lors de la comparaison de deux images successives, l'identification du déplacement d'un motif se fait

simplement en comparant deux à deux les pixels des transformées des images.

Plusieurs types de transformées existent, nous nous intéressons ici à une transformée dont le codage a comme propriété une bonne efficacité en même temps que des comparaisons locales. Ces caractéristiques en font un codage qui se prête bien à l'intégration.

Nous avons choisi la Census Transform, qui est une transformée locale non paramétrique conservant les structures de l'image. Elle est utilisée dans des applications de robotique mobile, à cause de sa robustesse face aux scènes fortement texturées. De plus, les brusques variations de luminosité, comme par exemple celle produite par l'entrée du robot dans une zone d'ombre ou le reflets des objets, n'influent pas sur les calculs. Cette transformée a d'abord été proposée par Zabih et Woodfill dans [4], nous présentons ici son principe général. Elle est caractérisée par une comparaison de luminosité entre les pixels voisins d'une image. La transformation se fait à partir d'une image en niveaux de gris, telle que celle représentée figure 1.

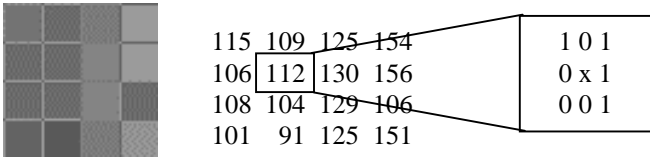


Figure 1 : Une image, ses niveaux de gris et le codage du pixel "112"

En se focalisant sur un pixel précis, considéré comme pixel central par rapport à son voisinage, on effectue une série de comparaisons. Les résultats de ces comparaisons ('1' si la luminosité du pixel 'voisin' est supérieure à celle du pixel central, et '0' sinon) constituent le codage du pixel considéré. Ainsi, dans la figure 1, le pixel dont la valeur est "112", sera codé 10101001, qu'on peut écrire A9 en hexadécimal (l'ordre de lecture des bits est arbitraire, il importe seulement qu'il soit systématique).

A chaque pixel est donc associé un code de huit bits (au lieu de le caractériser par les 8 bits du niveau de gris). L'image constituée de ces codes est la Census Transform de l'image originale. Une mémoire d'un demi mégabit est normalement nécessaire pour stocker la transformée d'une image de 256x256 pixels. Notons que la Census Transform peut se calculer sur des voisinages plus importants (5x5 par exemple), le nombre de bits du codage, et donc la taille de l'image transformée, reflétant alors l'augmentation de la taille du voisinage.



Figure 2 : Codes des pixels dans deux images successives

Pour déterminer le mouvement le plus probable entre deux images I1 et I2 successives, on utilise les Census Transform C^1 et C^2 de ces deux images. La figure 3 montre un extrait des Transformées correspondant à deux images (codage hexadécimal). Pour chaque pixel C^{1ij} de C^1 , on calcule la distance de Hamming entre C^{1ij} et C^{2ij} ainsi qu'avec chacun des pixels voisins de C^{2ij} . La taille de ce voisinage dépend de l'estimation faite sur l'amplitude du mouvement entre les deux images. Nous nous limitons ici à un voisinage de 3x3 autour de C^{2ij} . Le vecteur mouvement attribué au pixel ij est alors celui qui amène C^{2ij} sur le pixel de son voisinage le plus proche, au sens de la distance de Hamming (dH). En cas d'égalité, c'est la distance spatiale qui sert de discrimination. Ainsi, si $dH(C^{1ij}, C^{2ij}) = 0$, alors le pixel ij est noté sans mouvement, quelles que soient les autres valeurs. Dans l'exemple de la figure 3, le déplacement mesuré est celui qui amène le pixel de valeur 56 dans C^1 sur le pixel de valeur 56

dans C^2 , c'est à dire un déplacement horizontal de 1 pixel vers la droite.

Nous présentons dans ce papier l'intégration du calcul et de la mémorisation des Census Transforms de deux images successives dans une architecture de capteur CMOS.

3. Intégration dans un capteur CMOS

Nous avons choisi d'intégrer cette transformée spatiale au coeur même des pixels, c'est à dire d'introduire de l'électronique de traitement à côté de chaque zone photosensible.

Nous avons retenu deux solutions pour effectuer ce calcul de manière électronique. La première structure propose de comparer 2 tensions analogiques à un instant précis t_c , alors que la deuxième compare les 2 instants de commutation pour une tension de seuil V_s donnée.

Comme le montre la figure 3, après une phase de précharge à Vdd (initialisation), chaque photodiode (par exemple A et B) présente, sur sa cathode, une tension décroissante avec le temps (il s'agit de pixels fonctionnant en mode d'intégration du photocourant). La valeur absolue de la pente de cette rampe est une fonction croissante de l'éclairement.

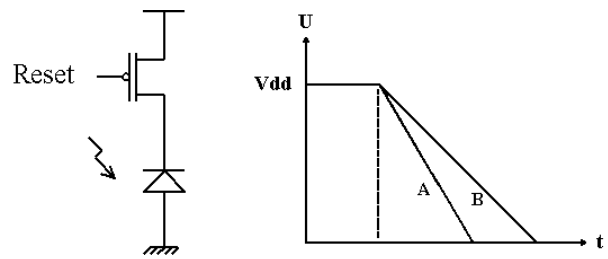
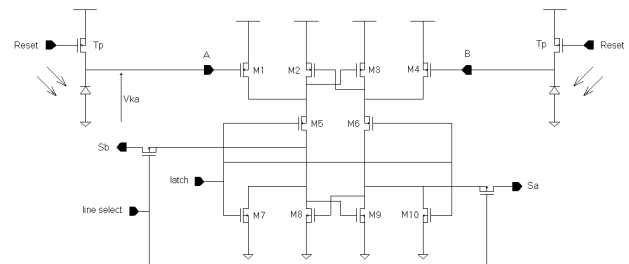


Figure 3 : Schéma d'un pixel et décharge des pixels A et B à lumière constante

Pour la première architecture (a), la comparaison inter-pixels est réalisée par un comparateur de tension analogique à transistors minimaux. A un instant donné t_c , on compare les tensions fournies, A et B, par deux photosites adjacents. Ce comparateur est construit autour d'un bistable formé de deux inverseurs rebouclés. Le grand gain de la structure amplifie même une très faible différence (quelques mV) sur les tensions d'entrées et provoque le basculement de la structure dans un état mémoire, '1' ou '0', reflétant le signe de la différence des tensions d'entrée A-B.



(a)

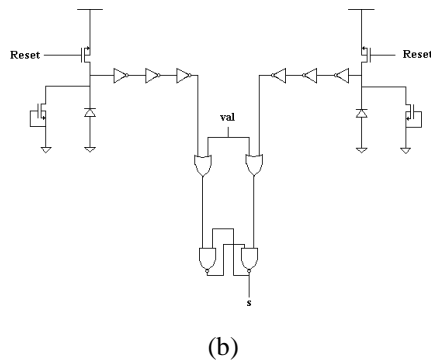


Figure 4 : Schémas des comparateurs (a) et (b) avec 2 photodiodes

Pour la deuxième solution (b), les décharges des photodiodes créent deux fronts sur les deux entrées de la structure. Le premier front passant de '1' à '0' force la bascule de type RS dans un état mémoire. Les schémas utilisés sont donnés Figure 4. A noter qu'un transistor de précharge, connecté entre Vdd et la cathode de la photodiode, permet d'initialiser cette dernière.

Le résultat obtenu en sortie de tels comparateurs correspond à l'un des bits du mot de huit bits représentant le résultat de la comparaison d'un pixel avec chacun de ses huit voisins. A l'analyse, le nombre de comparateurs peut être réduit à quatre du fait de la symétrie évidente qui existe entre B-A et A-B (on sacrifie là un peu de la redondance dans les codes de la Census Transform, mais ce sacrifice est fait de telle façon qu'elle reste aisée à reconstituer, par simple concaténation, lors de la lecture de l'image). Les deux structures ont été simulées, elles sont assez sensibles pour traiter des images ayant plus de 256 niveaux de gris (jusqu'à 3000 niveaux). On observe sur la figure 5 le résultat de la simulation comparant les 2 pixels A et B.

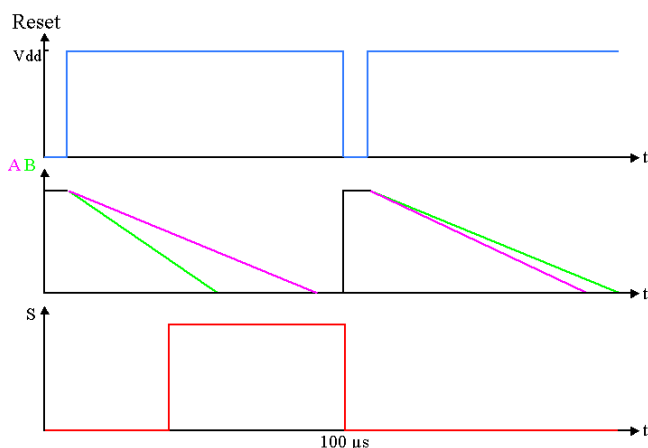
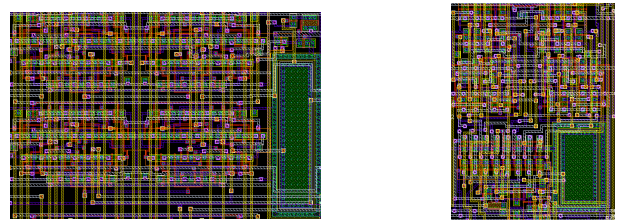


Figure 5 : Simulation de comparaison de 2 pixels A et B

Ces simulations ont été obtenues avec les 2 structures de comparateurs. La sortie réagit instantanément, soit dès que le signal de comparaison a été appliqué, pour la structure (a) ou dès qu'un signal a franchi le seuil, pour la structure (b).

Lorsque le pixel A est moins éclairé que le pixel B, la pente de sa décharge est moins prononcée, la sortie S passe donc à 1, signifiant que $I_A < I_B$. A l'inverse, quand le pixel A est plus éclairé, $I_A > I_B$, la sortie reste à 0.

Ces cellules ont été dessinées en technologie AMS 0.35μm, et utilisent une surface de respectivement pour les 2 structures : 3421μm² (a) et 2441μm² (b). Ces pixels, représentés figure 6, imposent pour des matrices de 100x100 pixels des surfaces de 34 mm² et 24 mm², ce qui prouve leur faisabilité. Les simulations ont montré que la structure axée autour de la bascule de type RS est plus compacte et plus précise. Cependant, le résultat n'est accessible que lorsque l'acquisition est terminée, alors que la structure de comparateur déclenché permet des comparaisons très rapides, le temps de commutation d'une bascule. La validation par des mesures sur un circuit de test, en cours de fabrication, montrera quelle est la meilleure architecture.



(a)

(b)

Figure 6 : Vue des layouts: comparateur à structure 'bistable' (a) et comparateur à structure 'RS' (b)

Un bloc numérique additionnel - pour compléter la reconnaissance du mouvement - a été développé en VHDL. Ce bloc a pour fonction de rechercher la distance de Hamming minimale dans les codes des deux images transformées, afin que l'estimation du mouvement soit réalisée totalement dans le capteur. Pour tout code de Census Ci de l'image 1, une recherche est effectuée sur les 9 positions possibles (de la fenêtre 3x3) dans l'image 2.

Un exemple de représentation du mouvement est donné en figure 7.

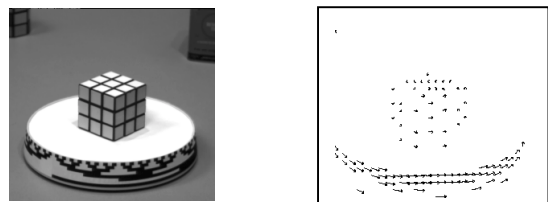


Figure 7 : Scène acquise et représentation du mouvement

Pour le capteur, nos simulations indiquent que le calcul complet sur 2 images prend 100ns, alors que deux expositions successives requièrent, pour une (forte) luminosité donnée, 100μs. Le temps de calcul est donc négligeable par rapport au fonctionnement 'normal' d'un capteur d'images. Ce résultat indique aussi de possibles utilisations en vidéo rapide. Dans un premier temps, dès la réception du circuit, le code VHDL

du bloc additionnel nous permettra de programmer un FPGA afin de faire des validations sur le circuit de test.

4. Conclusion

Nous avons présenté un algorithme d'estimation du mouvement qui utilise une transformée spatiale d'image, la Census Transform, afin d'ajouter de la robustesse quant au traitement d'images fortement texturées. Cette transformée code les pixels de deux images, pour faciliter leur reconnaissance afin d'estimer leur déplacement. Sa robustesse en fait un algorithme utilisé en robotique. Nous présentons ici son intégration au sein même d'un capteur d'images réalisé en technologie CMOS standard 0.35 μ m. A ce jour, les schémas et layouts des différents éléments sont fixés et un circuit est en cours de fabrication. La simulation a permis de valider ces structures, et les mesures sur le circuit permettront de confirmer la fonctionnalité et d'étudier les différentes architectures de calcul électronique.

5. Bibliographie

- [1] J. Woodfill, B. Von Herzen – Real-Time Stereo Vision on the PARTS Reconfigurable Computer – IEEE Symposium on Field-Programmable *Custom Computing Machines*, pp242-250, 1997.
- [2] L. Qiu, L. Li – *Contour Extraction of Moving Objects* – 14 th International Conference on Pattern Recognition, Australia, pp 1427-1432, 1998.
- [3] L. Lew Yan Voon et al – *Optical Correlation allows silicon retinas to recognize patterns stored in memory* – International Conference on Quality Control by Artificial Vision, pp.238-242, 2001.
- [4] R. Zabih, J. Woodfill – *Non-Parametric Local Transforms For Computing Visual Correspondance* – 3rd European Conference on Computer Vision, pp 151-158, 1994.
- [5] A. Moini – *Vision chips or seeing silicon* – Technical Report, Centre for High Performance Integrated Technologies and Systems, The University of Adelaide, 1997.
- [6] S.S. Beauchemin, J.L. Barron – *The computation of Optical Flow* – ACM Computing Surveys, Vol 27, 1995.