



HAL
open science

Adéquation Algorithme architecture : Aspects logiciels, matériels et cognitifs

Dominique Ginhac

► **To cite this version:**

Dominique Ginhac. Adéquation Algorithme architecture : Aspects logiciels, matériels et cognitifs. Traitement du signal et de l'image [eess.SP]. Université de Bourgogne, 2008. tel-00646480

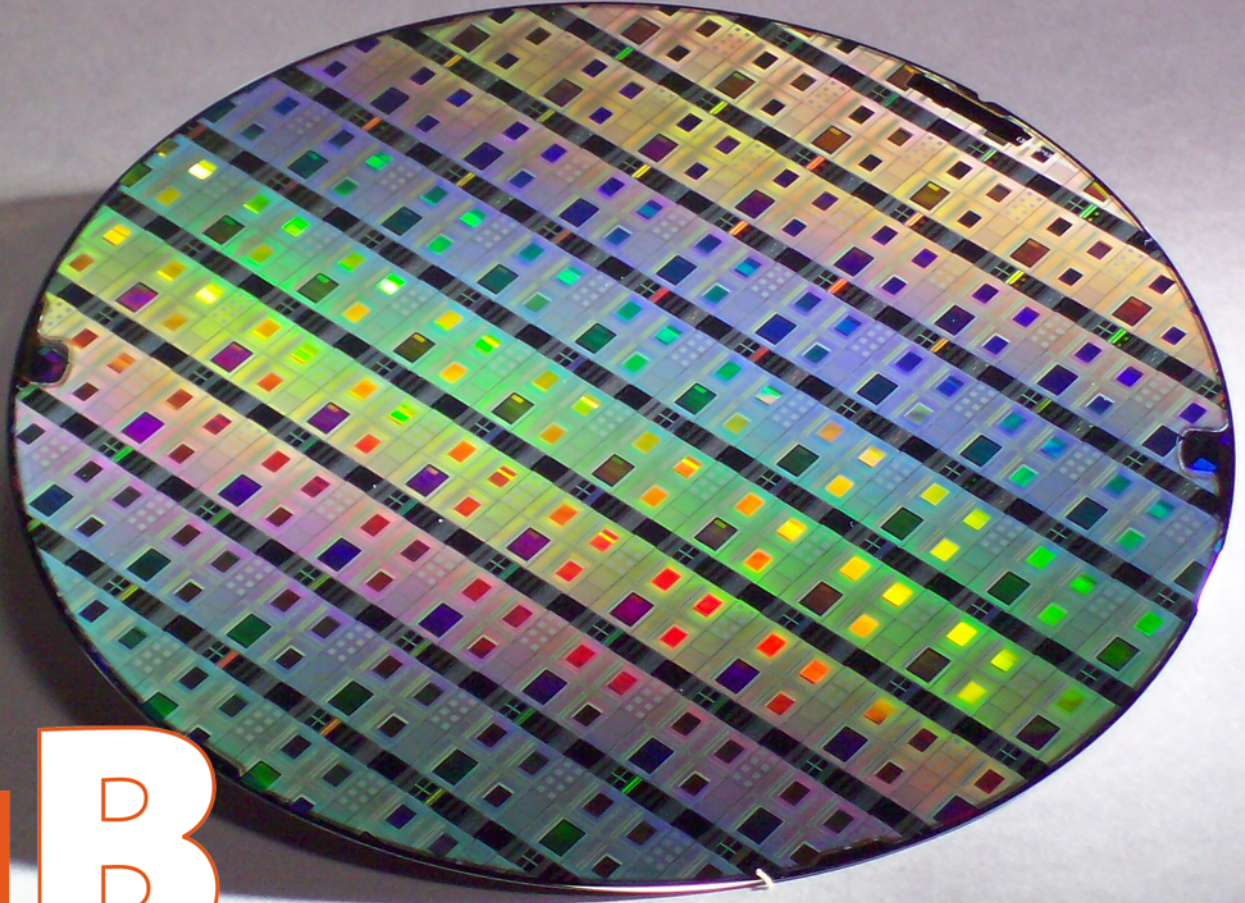
HAL Id: tel-00646480

<https://theses.hal.science/tel-00646480v1>

Submitted on 30 Nov 2011

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.



LE2I UMR CNRS 5158

UNIVERSITÉ DE BOURGOGNE

ADÉQUATION ALGORITHME ARCHITECTURE ASPECTS LOGICIELS, MATÉRIELS ET COGNITIFS

présentée par

DOMINIQUE GINHAC

Composition du Jury

Pr. PIERRE MAGNAN - Institut Supérieur de l'Aéronautique et de l'Espace - Rapporteur

Pr. ALAIN MÉRIGOT - Université Paris Sud - Rapporteur

Pr. MICHEL ROBERT - Université Montpellier II - Rapporteur

DR AXEL CLEEREMANS - Université Libre de Bruxelles - Examineur

Pr PATRICK MARQUIE - Université de Bourgogne - Examineur

Pr MICHEL PAINDAVOINE - Université de Bourgogne - Directeur de l'HDR

Remerciements

Sommaire du dossier

Remerciements	3
Sommaire du dossier	5
Introduction générale	9
Partie 1 : Dossier administratif	13
1 Introduction	15
2 Curriculum Vitae	17
2.1 Fonctions Actuelles	17
2.2 Formation Universitaire	18
2.3 Parcours Professionnel.....	18
2.4 Publications Significatives.....	19
2.5 Renseignements complémentaires.....	19
3 Synthèse des activités d'enseignement	21
3.1 Enseignement en cycle ingénieur de l'ESIREM	21
3.2 Enseignement à l'UFR Sciences de l'Université de Bourgogne	22
4 Synthèse des activités administratives	25
4.1 Activités informatiques	25
4.2 Activités pédagogiques.....	25
5 Animations et responsabilités scientifiques.....	27
5.1 Animation au niveau local	27
5.2 Animation au niveau national et international.....	27
5.3 Organisation de colloques.....	28
5.4 Conception de site web.....	28
6 Encadrements de thèses et de DEA.....	29
6.1 Thèses soutenues	29
6.2 Thèses en cours	30
6.3 Encadrements de DEA et Master Recherche	30
7 Activités contractuelles	31
7.1 Récapitulatif	31
7.2 Contrats internationaux publics (2 en cours, 1 en évaluation, 1 non retenu):	31
7.3 Contrats nationaux publics (3 en cours, 5 non retenus, 6 terminés)	32
7.4 Contrats privés (4 terminés).....	35
8 Collaborations scientifiques (en dehors des projets ANR)	37
8.1 Au niveau national (6)	37

8.2	Au niveau international (3)	38
9	Publications et Communications	39
9.1	Récapitulatif	39
9.2	Publications dans une revue internationale ou nationale - Travaux de recherche au LE2I (7)	40
9.3	Publications dans une revue internationale ou nationale - Travaux de thèse au LASMEA (3)	40
9.4	Chapitres de livre (2)	41
9.5	Communications à un congrès international avec actes (21)	41
9.6	Communications à un congrès national avec actes (11)	42
9.7	Publications soumises dans une revue internationale (3)	43
9.8	Publications en cours de rédaction (4)	43
10	Récapitulatif des travaux de recherche et responsabilités administratives	45
10.1	Publications	45
10.2	Co encadrement de thèse (à 70 %) et de DEA	45
10.3	Activités contractuelles	45
10.4	Responsabilités administratives	45
Partie 2 : Dossier de recherche		47
1	Introduction	49
2	Travaux de recherche en conception de capteurs CMOS	51
2.1	Contexte scientifique	51
2.2	Développement de capteurs CMOS dédiés à la reconnaissance de visages (2001 - 2005)	53
2.3	Développement de capteurs CMOS de type Gamma Caméra dédiés à des applications médicales (2002 - 2007)	55
2.4	Intégration de traitements d'images bas niveau au sein de capteurs CMOS programmables (2003 - 2008)	56
2.5	Projet ANR PACS : Programmable Architecture for CMOS Sensor (2008 - 2011)	58
2.6	Projet ANR PANINI : Programme Architectures Nano-électroniques Intégrées Neuro Inspirées (2008 - 2011)	60
2.7	Projet européen HiDRaLoN : High Dynamic Range Low Noise CMOS imagers (2009 - 2012)	61
2.8	Publications jointes	63
3	Travaux de recherche en Adéquation Algorithme Architecture	89
3.1	Contexte scientifique	89
3.2	Développement d'outils de prototypage rapide d'applications de traitement d'images (2000 - 2002)	90
3.3	Développement d'applications de traitement d'images à fortes contraintes temporelles (2000 - 2006)	92
3.4	Projet ANR : Nouvelles technologies et méthodes pour la rééducation motrice (2007 - 2011)	95

3.5	Projet européen : Développement de capteurs embarqués sans fil pour applications médicales (2006 - 2010).....	96
3.6	Publications jointes.....	97
4	Travaux de recherche en Algorithmie pour les Sciences Cognitives	137
4.1	Contexte scientifique.....	137
4.2	Développement algorithmique pour le projet « Enfant et réalité virtuelle » (2002-2004)	138
4.3	COGSCI : Apport des Sciences Cognitives à l'Apprentissage et la Reconnaissance de Forme (2002-...).....	139
4.4	Publication jointe	140
5	Conclusions et perspectives de recherche	151
5.1	Bilan des recherches effectuées.....	151
5.2	Directions de recherche future	152

Introduction générale

Les travaux présentés dans le cadre de cette Habilitation à Diriger des Recherches s'inscrivent principalement dans la problématique dite d'« **Adéquation Algorithme Architecture** » (A³) de la **section 61 du CNU** « Génie informatique, automatique et traitement du signal ». Ils ont pour objectif commun la mise en œuvre de systèmes matériels et logiciels dédiés à la vision artificielle à fortes contraintes temporelles. Ces systèmes matériels reposent principalement sur des capteurs d'images spécifiquement développés en technologie CMOS et interfacés avec des structures de calculs plus traditionnelles telles que des FPGA ou des DSP. De fait, ces activités de conception microélectronique s'inscrivent également dans les champs disciplinaires de la **section 63 du CNU** « Electronique, Optronique et Systèmes » tels que « **Conception assistée des circuits intégrés et de microsystèmes** », « **Capteurs** » ou « **Traitement du Signal** ».

Ces travaux de recherche se focalisent sur différents aspects cruciaux tels que l'acquisition d'images par des capteurs dédiés, le développement et la programmation d'architectures optimisées de traitement des images et l'implantation d'algorithmes de traitement du signal et d'images en temps réel sur ces architectures. Ces systèmes matériels de vision doivent répondre à des contraintes cruciales telles que limitation du bruit des capteurs, débit et accès rapide à l'information, rapidité de traitement, embarquabilité, facilité de programmation des applications, ... Satisfaire de telles exigences impose inévitablement la mise en place d'activités de modélisation, de développement et de conception de capteurs et d'architectures complètement dédiés. De fait, une approche « Adéquation Algorithme Architecture » est abordée à tous les niveaux hiérarchiques, allant du niveau le plus fin lors de la conception microélectronique de capteurs ou de circuits dédiés jusqu'au niveau le plus élevé avec le développement d'outils logiciels de prototypage rapide d'applications de traitement d'images. L'originalité de ces travaux est de confirmer l'approche théorique par une approche expérimentale systématique à travers la réalisation de démonstrateurs à base de systèmes électroniques et informatiques sur lesquels s'exécutent des applications complexes de traitement d'images. Plus particulièrement, les applications visées concernent le domaine des Sciences et Technologies de l'Information comme la reconnaissance de visages en temps réel, les Sciences de la Vie et de la Santé comme les dispositifs de mesure temps réel de pression pour les maladies veineuses ainsi que les Sciences de l'Ingénieur comme le contrôle qualité en temps réel de produits manufacturés.

Depuis 1995, mes travaux dans le domaine de l'adéquation algorithme architecture se situent dans 3 axes complémentaires.

Le premier axe a débuté lors de mes travaux de thèse (1995-1998) au LASMEA - Clermont Ferrand et s'est poursuivi après mon recrutement à l'Université de Bourgogne en 2000. Il concernait le développement de **méthodologies formelles et d'outils de prototypage rapide** permettant d'optimiser l'implantation d'applications de traitement d'images tout en réduisant fortement les temps de développement. En effet, la programmation des architectures dédiées au traitement d'images demeure un exercice délicat encore réservé à des spécialistes. La résultante est des temps de conception-implantation-validation

importants d'où une incapacité à évaluer rapidement un large spectre de solutions vis à vis d'un problème donnée. Or, dans le contexte des applications de traitement d'images, seule une approche expérimentale fondée sur la validation in situ des solutions permet de juger de leur validation fonctionnelle et opérationnelle. La seule solution passe donc par une diminution drastique des temps de développement ce qui suppose l'existence de formalismes de haut niveau et des outils d'aide à l'implantation d'applications. Dans ce contexte, l'approche visée dans le cadre de mes travaux de thèse concernait le développement et la mise en œuvre d'un outil d'aide à la parallélisation d'applications de traitement d'images à fortes contraintes temporelles. En septembre 2000, ma nomination comme Maître de Conférences (61^{ème} section) à l'Université de Bourgogne m'a offert l'opportunité de renforcer cette activité de recherche sur un aspect peu abordé durant mes travaux de thèse, à savoir la mise au point d'**applications complexes de traitement d'images** à fortes contraintes temporelles, et en particulier la reconnaissance de visages.

Parallèlement à ces activités de développement d'applications, le deuxième axe de mes recherches concerne la **conception microélectronique** de capteurs d'images dédiés appelés **rétines artificielles**. Une rétine artificielle est un circuit intégré associant sur un même substrat de silicium un capteur d'images et une architecture parallèle de traitements dans le but de constituer un système de vision compact. L'amélioration permanente des procédés technologiques permet d'associer aux capteurs CMOS des opérations de plus en plus complexes de traitement d'images et donc de surpasser les capteurs classiques de type CCD. Ces propriétés sont particulièrement intéressantes dès lors que l'on s'intéresse à la conception de systèmes embarqués capables à la fois d'acquérir des images, de les prétraiter et d'extraire des informations pertinentes du flot vidéo. Introduire de l'intelligence sous forme de traitement d'images programmables au sein même des pixels d'un capteur CMOS constitue un axe de recherche particulièrement innovant pouvant conduire à court et moyen terme au développement de nouvelles applications haute performance de traitement d'images embarqué.

La problématique d'adéquation algorithme architecture est abordée ici à une granularité extrêmement fine au plus près du silicium composant les pixels élémentaires des capteurs. Les travaux se sont focalisés, d'une part sur la modélisation de capteurs et des circuits de traitement associés et d'autre part, sur le développement microélectronique d'architectures dédiées à la vision artificielle intégrant sur un substrat silicium à la fois le capteur dédié à l'acquisition des images et l'unité de calcul analogique chargée de traiter l'information image. L'objectif ambitieux de ces travaux est de concevoir de nouvelles générations de capteurs intelligents surclassant les capteurs existants en réalisant des opérations complexes tout en garantissant une embarquabilité maximale.

Le troisième axe de mes recherches concerne des travaux de recherche en **Algorithmie pour les Sciences Cognitives** dans le cadre de collaborations avec le laboratoire de psychologie de l'Université de Bourgogne et le laboratoire de Sciences Cognitives de l'Université Libre de Bruxelles. La collaboration entreprise avec les spécialistes des sciences cognitives a pour objectif de mieux comprendre les mécanismes cognitifs mis en œuvre par les êtres humains dans les phénomènes d'apprentissage, de reconnaissance, d'attention visuelle. Ces travaux ont conduit au développement de différentes applications informatiques dédiées à l'expérimentation et à l'étude de ces phénomènes cognitifs. A plus long terme, l'idée première est d'essayer de tirer parti de ces approches cognitives afin de donner à une rétine artificielle des capacités importantes d'interprétation des scènes visuelles.

En conclusion, mes travaux de recherche se situent au cœur de la thématique **Adéquation Algorithme Architecture** qui est abordée selon trois points de vue distincts mais complémentaires. En effet, l'activité se situe 1) au **niveau logiciel** par le développement d'outils de prototypage et d'applications de traitement d'images temps réel, 2) au **niveau**

matériel par la conception de capteurs d'images CMOS intégrant des traitements d'images programmables et 3) au **niveau cognitif** par l'étude des processus cognitifs élémentaires impliqués dans l'apprentissage ou la reconnaissance.

Au final, le défi majeur se situera au niveau de l'intégration des travaux menés dans ces trois directions dans le but de modéliser et de concevoir une rétine artificielle dotée de fonctions bio inspirées de haut niveau lui permettant d'apprendre et d'interpréter une scène visuelle complexe.

Les travaux présentés dans ce rapport sont le fruit d'une douzaine d'années de recherche initiée au LASMEA à Clermont-Ferrand pendant mes 3 années de thèse et poursuivie au LE2I à Dijon depuis mon recrutement en 2000. Durant ces 8 dernières années, j'ai co-encadré 4 thèses dont trois ont déjà été soutenues dans le domaine de la conception microélectronique de capteurs d'images.

La première partie de ce dossier présente d'une manière succincte les différentes activités liées au métier d'enseignant-chercheur. Dans un premier temps, seront abordées successivement une synthèse de mes activités d'enseignement, de mes activités administratives et de mes activités d'animation et de responsabilités scientifiques. Dans un deuxième temps, un résumé des activités directement liées à la recherche sera présenté avec notamment une description détaillée des encadrements d'étudiants, des contrats de recherche, des collaborations nationales et internationales ainsi que la liste exhaustive des publications.

La deuxième partie de ce dossier est consacrée à la présentation complète des différents projets de recherche dans lequel je suis impliqué. Cette deuxième partie est divisée en trois principaux chapitres consacrés respectivement au développement microélectronique de capteurs d'images, à la problématique d'adéquation algorithme architecture en traitement d'images en temps réel et aux travaux de recherche en algorithmie pour les sciences cognitives. A la fin de chacun de ses chapitres, une ou deux publications de référence viennent étayer les propos développés tout au long du chapitre. Un chapitre final réalise la synthèse de l'ensemble de ces recherches et présente les aspects prospectifs de ces travaux à court et moyen terme.

Partie 1 :

Dossier administratif



The image shows a close-up of a hand holding a grey pen, writing on a medical form. The form is titled "DETAILS" and contains several fields for patient information. The fields are filled with the following text:

- Name: HUSBAND
- House No & Street Name: 11 BARTHIST
- Postcode: 3803
- Contact name: [blank]

Other visible text on the form includes:

- * Check one per group
- Please tick boxes (complete one box)
- Witnessed Yes No
- Time of day [] [] [] []
- Location [] [] [] []
- ULANCE ARRIVED Not started [] [] [] []
- tion of CPR [] [] [] []
- nder

1 Introduction

Après un premier cycle universitaire en 1992 à l'Université Blaise Pascal de Clermont Ferrand, j'ai obtenu un diplôme d'ingénieur du CUST en Génie Electrique en 1995. En parallèle avec la 3^{ème} année du cycle ingénieur, j'ai suivi les enseignements du DEA en Vision Artificielle à l'Université Blaise Pascal et réalisé mon stage d'initiation à la recherche au laboratoire LASMEA UMR 6602 (Laboratoire des Sciences des Matériaux pour l'Electronique et d'Automatique).

Après avoir effectué une thèse en Electronique au LASMEA de 1995 à 1998, j'ai occupé un poste d'ATER en Informatique en 1999-2000 avant d'être recruté en 2000 en tant que Maître de Conférences en 61^{ème} section du CNU à l'Université de Bourgogne. Depuis cette date, je suis enseignant à l'école d'ingénieurs ESIREM et membre de l'équipe « Capteurs et Architectures » du laboratoire LE2I UMR 5158 (Laboratoire Electronique Informatique et Image).

Du point de vue recherche, j'ai initié et développé au sein du LE2I une thématique de recherche centrée sur la conception de systèmes matériels et logiciels dédiés à la vision artificielle à fortes contraintes temporelles. Ces travaux se focalisent sur différents aspects cruciaux tels que l'acquisition d'images par des capteurs dédiés, le développement et la programmation d'architectures optimisées de traitement des images et l'implantation d'algorithmes de traitement du signal et d'images en temps réel sur ces architectures.

Du point de vue enseignement, j'ai mis en place et assuré la responsabilité de nombreux modules d'enseignement dans le domaine de la conception microélectronique de circuits intégrés ainsi qu'en informatique.

Du point de vue administratif, je suis fortement impliqué dans les différentes structures d'enseignement et de recherche. Du point de vue de l'enseignement, j'assume plus particulièrement la responsabilité pédagogique de la dernière année du cycle ingénieur du département InfoTronique de l'ESIREM. Du point de vue de la recherche, je suis co-responsable de la thématique « Rétines CMOS » au sein du LE2I en charge de différents projets de recherche (ANR, projets européens).

La première partie de ce mémoire présente de manière détaillée ces différentes activités pédagogiques, scientifiques et administratives.

2 Curriculum Vitae

Dominique GINHAC

Maître de Conférences 61^{ème} section - Classe normale

ESIREM / LE2I - Université de Bourgogne



Adresse Professionnelle

LE2I - Université de Bourgogne
Aile de l'Ingénieur -BP 47870
21078 Dijon Cedex

Tel: + 33 (0)3 80 39 38 86

Email : dginhac@u-bourgogne.fr

Adresse Personnelle

13 Impasse de La Grande Fin
21110 THOREY EN PLAINE

Tel : +33 (0)3 80 79 18 93

Née le 26 Mai 1972 (36 ans) à Saint-Flour (15)

Nationalité : Française

Marié, 1 enfant (5 ans)

2.1 Fonctions Actuelles

Recherche : Chercheur - Laboratoire Electronique Informatique Image (LE2I UMR 5158) :

- Conception microélectronique de **capteurs d'images CMOS** avec traitements intégrés
- Conception de plates-formes logicielles dédiées à l'**Adéquation Algorithme Architecture** pour le Traitement d'images temps réel

Chercheur Visiteur au CONsciousness, COgnition & COmputation Group (CO3) - Université Libre de Bruxelles :

- Etude des **mécanismes cognitifs** impliqués dans les situations d'apprentissage et de reconnaissance de formes

Enseignement : Responsable de différents modules d'enseignement :

- Electronique : Conception Micro Electronique, Technologie des composants, Conception de Circuits Electronique
- Informatique : Programmation en langage C, Conception web

Administration : Responsable pédagogique - 5^{ème} année Département InfoTronique - ESIREM

2.2 Formation Universitaire

1995-1998 **Doctorat d'Electronique** de l'Université Blaise Pascal de Clermont-Ferrand (Mention Très Honorable avec les félicitations à l'unanimité du jury) sur le sujet :

**« Prototypage rapide d'applications parallèles
de vision artificielle par squelettes fonctionnels »**

Thèse préparée au sein du Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique) (LASMEA UMR 6602 CNRS)

Jury :

Bertrand Zavidovique (Université Paris Sud) - Président
Michel Paindavoine (Université de Bourgogne) - Rapporteur
Yves Sorel (INRIA) - Rapporteur
Guy Cousineau (Université Paris VII) - Examineur
Michel Dhome (Université Blaise Pascal) - Examineur
Jocelyn Sérot (Université Blaise Pascal) - Examineur
Jean-Pierre Dérutin (Université Blaise Pascal) - Directeur de Thèse

1994-1995 **DEA Electronique et Systèmes - Option Vision pour la Robotique -** Université Blaise Pascal de Clermont-Ferrand (Mention Assez Bien)

1992-1995 **Ingénieur Génie Electrique du CUST -** Université Blaise Pascal de Clermont-Ferrand (Mention Bien)

1990-1992 **DEUG Electronique Electrotechnique Automatique** (Mention Bien) - Université Blaise Pascal de Clermont-Ferrand

2.3 Parcours Professionnel

2007-2009 **Chercheur Visiteur, COnciousness, COgnition & COmputation Group (CO3)** -Université Libre de Bruxelles (Belgique)

Depuis 2000 **Maître de conférences ESIREM / LE2I**, Université de Bourgogne

1999-2000 **Attaché Temporaire d'Enseignement et de Recherche (ATER)** en Informatique (section n°27 CNU), Université Blaise Pascal de Clermont-Ferrand

1998-1999 **Service National, informaticien**, Clermont-Ferrand
Administration réseau et développement d'applications Internet

1995-1998 **Allocataire de recherches** à l'Université Blaise Pascal de Clermont-Ferrand

2.4 Publications Significatives

1. J. Dubois, **D. Ginhac**, M. Paindavoine, B. Heyrman *A 10 000 frames/s CMOS Image Sensor with Multi-Processing Pixel*. IEEE Journal of Solid-State Circuits, 43(3), 706-717, 2008.
2. S. Chambaron, **D. Ginhac**, P. Perruchet. *Methological issues and computational software dedicated to SRT tasks*. Behavior Research Methods, 40(2), 493-502, 2008.
3. **D. Ginhac**, F. Yang, M. Paindavoine. *Design, Implementation and Evaluation of Hardware Vision Systems dedicated to Real-Time Face Recognition*. In Delac and Mislav Grgic (eds), Face Recognition, Vienna, Austria, pp. 123-148, 2007.
4. J. Sérot, **D. Ginhac**. *Skeletons for parallel image processing: an overview of the SKiPPER project*, Parallel Computing, 28(12), 1785-1808, 2002.
5. J. Sérot, **D. Ginhac**, R. Chapuis, J.P. Dérutin. *Fast prototyping of parallel vision applications using functional skeletons*. Journal of Machine Vision and Applications, 12(6), 271-290, 2001.

2.5 Renseignements complémentaires

- 2008-2009 Bénéficiaire de 2 semestres d'**Accueil en Délégation** au CNRS
- 2007-2008 Bénéficiaire de 2 semestres de **Congé pour Recherche ou Conversion Thématique** (CRCT) au titre de la 61^{ème} section du CNU
- 2005-2009 Bénéficiaire de la **Prime d'Encadrement Doctoral et de Recherche** (PEDR) depuis octobre 2005
- 2000-2008 Co-Encadrement de **4 thèses** (dont 3 soutenues) avec accord du Conseil Scientifique de l'Université de Bourgogne
Encadrement de **4 DEA** (2001, 2004, 2005, 2006)
- 2002-2008 Membre suppléant de la **Commission de Spécialistes** (61/63^{ème} section) de l'Université de Bourgogne
- 2002-2006 Membre élu du **Conseil d'Administration** de l'ESIREM
- 2000-2008 Administrateur **réseau et systèmes informatiques** de l'ESIREM

3 Synthèse des activités d'enseignement

Depuis mon recrutement en tant que Maître de Conférences en 2000, j'ai effectué principalement mes activités d'enseignements au sein des deux départements Matériaux et InfoTronique de l'école d'ingénieurs ESIREM (cycle ingénieur Bac +3 à Bac +5) à laquelle je suis rattaché administrativement. J'ai également été impliqué dans différents modules d'enseignement au sein du département Informatique Electronique Mécanique de l'UFR Sciences et Techniques de l'Université de Bourgogne.

Le volume global de ces enseignements est de 1820 heures (536h CM - 723h TD - 561 h TP). Ceci représente 1901 heures équivalent TD réalisées en 7 années universitaires (de Septembre 2000 à Septembre 2007), soit un service d'environ 271 h équivalent TD par année universitaire. Depuis septembre 2007, je suis intégralement déchargé de mes activités d'enseignement du fait de l'obtention d'un Congé de Recherche et Conversion Thématique (CRCT) obtenu au titre du CNU 61^{ème} section (2007 - 2008), puis d'un accueil en délégation au CNRS (2008 - 2009).

Les enseignements réalisés pendant ces 7 années ont été dispensés à un public varié, et porte sur des matières allant de l'initiation à l'informatique jusqu'aux techniques avancées de conception de circuits VLSI.

J'ai eu la chance de me voir confier la responsabilité de nombreux modules d'enseignement, en particulier dans le département InfoTronique de l'ESIREM. Ces modules n'existaient pas pour la plupart et ont été spécifiquement créés pour les 3 années du cycle ingénieur InfoTronique. Ceci a nécessité non seulement de préparer les Cours Magistraux, mais également les Travaux Dirigés ainsi que mettre en place les Travaux Pratiques adéquats. La responsabilité de ces modules implique également la gestion de l'équipe pédagogique en charge de tous ces enseignements.

Je ne présente ici que les modules dont je suis le responsable et pour lesquels j'assure les Cours Magistraux.

3.1 Enseignement en cycle ingénieur de l'ESIREM

- Responsable de différents modules d'enseignement en **électronique** :

- Conception CMOS numérique (depuis 2004)

Mots clés : Design CMOS de circuits intégrés, fonctions CMOS de base (portes logiques), fonctions CMOS classiques (arithmétiques, registres, compteurs)

Objectif du module : Former les étudiants aux techniques de conception des Circuits Intégrés Numériques

Volume horaire : 20h de Cours Magistraux par an

Public : 4^{ème} année du département InfoTronique

- Conception Micro Electronique avancée et Technologie des composants (depuis 2005)
 - Mots clés : Technologie de fabrication des circuits CMOS, fonctions CMOS élaborés (mémoires SRAM et DRAM)
 - Objectif du module : Former les étudiants aux techniques avancées de conception des Circuits Intégrés Numériques
 - Volume horaire : 16h de Cours Magistraux par an
 - Public : 5^{ème} année du département InfoTronique
- Responsable de différents modules d'enseignement en **informatique** :
 - Programmation en langage C (2002-2005)
 - Mots clés : Algorithmie, outils fondamentaux de programmation en langage C
 - Objectif du module : Former les étudiants à la programmation en langage C
 - Volume horaire : 16h de Cours Magistraux par an
 - Public : 3^{ème} année du département Matériaux
 - Conception de sites web (depuis 2002)
 - Mots clés : Conception graphique, développement de site web, langage HTML
 - Objectif du module : Former les étudiants au développement de sites web
 - Volume horaire : 16 h de Cours Magistraux par an
 - Public : 4^{ème} année du département Matériaux

3.2 Enseignement à l'UFR Sciences de l'Université de Bourgogne

- Responsable de différents modules d'enseignement en **électronique** :
 - Conception CMOS numérique (depuis 2000)
 - Mots clés : Design CMOS de circuits intégrés, fonctions CMOS de base (portes logiques), fonctions CMOS classiques (arithmétiques, registres, compteurs)
 - Objectif du module : Former les étudiants aux techniques de conception des Circuits Intégrés Numériques
 - Volume horaire : 24h de Cours Magistraux par an
 - Public : Master STIC, 1^{ère} année, Parcours Electronique
 - Conception CMOS analogique et Technologie des composants (2001-2004)
 - Mots clés : Initiation au Design de circuits intégrés analogiques (amplificateurs, filtres, convertisseurs)
 - Objectif du module : Donner aux étudiants une compétence dans la conception de circuits intégrés analogiques
 - Volume horaire : 30h de Cours Magistraux par an

Public : Master STIC, 2^{ème} année Spécialité professionnelle Signal Electronique et Automatique

- Architectures parallèles dédiées au traitement d'images (depuis 2002)

Mots clés : Traitement d'images en temps réel, architectures parallèles, circuits spécialisés

Objectif du module : Initiation à la problématique d'implantation d'algorithmes de traitement d'images à fortes contraintes temporelles sur architectures spécialisées

Volume horaire : 6h de Cours Magistraux par an

Public : Master STIC, 2^{ème} année, Spécialité recherche Instrumentation et Informatique de l'Image

4 Synthèse des activités administratives

4.1 Activités informatiques

Depuis mon recrutement en tant que Maître de conférences en 2000 à l'ESIREM, j'ai toujours été impliqué dans le fonctionnement de l'école.

Entre 2000 et 2007, j'ai été responsable Réseaux et Systèmes informatiques de l'ESIREM. En raison de l'augmentation constante du nombre d'élèves et de la mise en place du département InfoTronique en 2003, le parc des machines de l'école est passé de 20 postes en 2000 à plus de 100 machines en 2008 organisées en une dizaine de salles.

En tant que responsable informatique, j'ai donc assuré les fonctions suivantes :

- Définition des besoins informatiques en fonction des enseignements
- Montage des dossiers de financement du matériel
- Commande, installation et maintenance des logiciels et des matériels
- Formation des personnels à l'utilisation de matériels et logiciels spécifiques

En 2006, j'ai pris la responsabilité d'une équipe chargée du développement du nouveau site web de l'école¹. Cette équipe est composée de 2 étudiants du département InfoTronique et de 3 enseignants de l'école. Ce projet a permis de mettre en ligne un nouveau site en mars 2007. A l'heure actuelle, je m'occupe principalement de la coordination de l'équipe, du développement de nouveaux modules, de la rédaction et de la mise en ligne des contenus.

4.2 Activités pédagogiques

En 2002, l'ESIREM a souhaité ouvrir un deuxième département dans les domaines de l'électronique et de l'informatique. En tant qu'enseignant dans ces disciplines, j'ai été fortement impliqué dans le montage du dossier transmis à la Commission des Titres Ingénieurs (CTI).

En 2003, le département InfoTronique accueillait ses premiers étudiants et j'ai pris en 2005 la responsabilité pédagogique de la 3^{ème} année du cycle Ingénieur de ce département. A ce titre, j'assume un certain nombre de responsabilités administratives :

- Définition des services des intervenants et organisation de l'emploi du temps
- Constitution de l'équipe pédagogique et choix des enseignants vacataires
- Participation aux jurys de soutenance de stage ingénieur
- Gestion des projets de fin d'étude
- Recrutement des nouveaux élèves

¹ <http://esirem.u-bourgogne.fr>

5 Animations et responsabilités scientifiques

5.1 Animation au niveau local

- Co-responsable de la thématique « Rétines CMOS » au sein du laboratoire LE2I. Je participe activement à l'animation de l'équipe Architecture Dijon composée de 7 enseignants-chercheurs.
- Responsable de projets transversaux en collaboration avec le LEAD UMR CNRS 5022 de l'Université de Bourgogne. Ces projets sont le Programme Interdisciplinaire CNRS « Enfant et Réalité Virtuelle » et la thématique « Apprentissage ».
- Membre de la commission de spécialistes des 61^{ème} et 63^{ème} sections de l'Université de Bourgogne.
- Membre du conseil de laboratoire du LE2I.

5.2 Animation au niveau national et international

- Membre actif du groupe « Adéquation Algorithme Architecture » au sein du Thème C du GDR ISIS depuis 1995. Ce groupe a pour objectif l'étude simultanée des aspects algorithmiques et architecturaux en prenant en compte leurs interactions. J'ai présenté mes travaux à plusieurs reprises lors des réunions de ce groupe.
- Membre actif du groupe « Rétine » au sein du Thème C du GDR ISIS depuis 2000. L'objectif principal est l'étude et la conception d'une rétine dédiée à la détection et à la mesure des mouvements dans des séquences d'images. J'ai présenté mes travaux à plusieurs reprises lors des réunions de ce groupe.
- Membre actif en 2000 de l'Action Spécifique 29 « SOC : Systèmes d'exploitation et architectures multiprocesseurs » du Réseau Thématique Pluridisciplinaire « System On Chip » (RTP SOC) du département STIC du CNRS. Les travaux de cette Action Spécifique consistaient à cerner les difficultés rencontrées dans l'élaboration d'un système sur puce afin d'identifier celles nécessitant des travaux de recherche soutenus.
- Reviewer pour la revue IEEE Transactions on Circuits and Systems (Part I)
- Reviewer pour différents congrès internationaux :
 - « International Congress on High Speed Photography and Photonics 2002 » (HSPP'02)
 - « Journées francophones sur l'Adéquation Algorithme Architecture » (JFAAA'05)
 - « Advanced Concepts for Intelligent Vision Systems » (ACIVS'2004)
 - « Workshop on Distributed Smart Cameras » (DSC06)

5.3 Organisation de colloques

- Co-organisateur du 22^{ème} colloque GRETSI 2009 - Traitement du signal et des images qui aura lieu à Dijon en septembre 2009
- Co-organisateur du colloque international NDES2006 (14th International IEEE Workshop on Nonlinear Dynamics of Electronic Systems) qui a eu lieu en juin 2006 à Dijon.
- Membre du comité d'organisation du colloque international HSPP'02 (High Speed Photography and Photonics) qui a eu lieu en septembre 2002 à Beaune
- Membre du comité d'organisation du colloque international JFFAAA'05 (Journées Francophones sur l'Adéquation Algorithme Architecture) qui a eu lieu en janvier 2005 à Dijon.

5.4 Conception de site web

- Concepteur du site web du colloque GRETSI 2009 (<http://www.gretsi2009.org>)
- Concepteur du site web du colloque NDES2006 (<http://cerbere.u-bourgogne.fr/ndes2006>)
- Concepteur du site web de l'école d'ingénieurs ESIREM (<http://esirem.u-bourgogne.fr>)

6 Encadrements de thèses et de DEA

Depuis mon recrutement en tant que Maître de Conférences en 2000, j'ai eu l'opportunité de suivre les travaux de quatre étudiants en DEA / Master Recherche.

J'ai également co-encadré quatre thèses dont trois ont déjà été soutenues (2005, 2007 et 2008) et une est en cours. Les thèses sont co-encadrées sous la responsabilité du Professeur M. Paindavoine avec l'accord du Conseil Scientifique de l'Université de Bourgogne. De plus, une nouvelle thèse a démarré en octobre 2008 dans le cadre de l'ANR Architecture du Futur.

Ces encadrements se situent principalement dans la thématique « Conception CMOS de capteurs d'images » ainsi que dans la thématique « Développement d'applications complexes de traitement d'images à fortes contraintes temporelles ».

6.1 Thèses soutenues

1. Jérôme Dubois (encadrement à 70%)

Sujet : « Architecture de rétine CMOS : vers un capteur intelligent programmable »

Financement : Allocation de recherche - ENS Cachan

Début : Septembre 2005

Soutenance : Aout 2008

Situation actuelle : Chercheur CEA - LIST

2. Nur Salahuddin (encadrement à 70%)

Sujet : « Conception d'un capteur CMOS de type Gamma Camera dédié à des applications médicales »

Financement : Bourse de 4 ans du gouvernement indonésien

Début : Octobre 2002

Soutenance : Juillet 2007

Situation actuelle : Enseignant Chercheur Université Gunadarma - Indonésie

3. Eri Prasetyo (encadrement à 70%)

Sujet : « Localisation et reconnaissance de visages : Etude et réalisation d'une implantation sur silicium »

Financement : Bourse de 4 ans du gouvernement indonésien

Début : Octobre 2001

Soutenance : Octobre 2005

Situation actuelle : Enseignant Chercheur Université Gunadarma - Indonésie

6.2 Thèses en cours

1. Auguste Ngoua

Sujet : « Nouvelle génération de capteurs CMOS massivement parallèles »

Financement : ANR Architectures du Futur

Début : Octobre 2008

Soutenance : prévue en Octobre 2011

2. Julien Steinbrunn (encadrement à 70%)

Sujet : « Optimisation, qualification et intégration de capteurs dans une chaîne de mesure pour dispositifs médicaux souples et extensibles »

Financement : Contrat Privé – Société Urgo

Début : Janvier 2007

Soutenance : prévue en Décembre 2009

6.3 Encadrements de DEA et Master Recherche

1. Olivier Brousse (encadrement à 100%)

Sujet : « Adéquation Algorithme Architecture : Application à la reconnaissance de visages en temps réel »

Soutenance : Juin 2006

Situation actuelle : Thèse LIRMM - Montpellier

2. Jérôme Dubois (encadrement à 100%)

Sujet : « Architecture de rétine CMOS : vers un capteur intelligent programmable »

Soutenance : Juin 2005

Situation actuelle : Chercheur CEA - LIST

3. Moussa Witt (encadrement à 100%)

Sujet : « Etude de la faisabilité microélectronique d'une cellule élémentaire d'un réseau non linéaire dédié au traitement d'images temps réel »

Soutenance : Juin 2004

Situation actuelle : inconnue

4. Damien Bray (encadrement à 100%)

Sujet : « Etude de rétines analogiques appliquées à la reconnaissance de visages »

Soutenance : Juin 2001

Situation actuelle : Poste Ingénieur SSII

7 Activités contractuelles

7.1 Récapitulatif

Type de projet	Nombre de demandes	Nombre de projets retenus	Financement accordé LE2I
International public	3	1 (+ 1 en évaluation)	40 k€
National public	14	9	1010 k€
Privé	4	4	10 k€
Total	21	14	1060 k€

7.2 Contrats internationaux publics (2 en cours, 1 en évaluation, 1 non retenu):

1. Projet européen Eureka Catrene (projet en cours d'évaluation)

Sujet : « **HiDRaLoN : High Dynamic Range Low Noise CMOS imagers** »

Résumé : Le projet HiDRaLoN vise à développer une nouvelle génération de capteurs CMOS à grande dynamique, grande résolution, faible bruit et haute qualité d'images.

Période : 2009 - 2012.

Montant : non encore défini

Partenaires : 15 partenaires dont Thomson, Thales, Philips, E2V, Fraunhofer Institute

Porteur du projet : Grass Valley (NL)

Responsable LE2I : D. Ginhac

2. Projet européen Large-scale integrating project (IP) FP7- ICT-2007.3.5 (projet non retenu)

Sujet : « **HIDRIC : High Dynamic Range Image Capture** »

Résumé : Le projet HIDRIC vise à développer une nouvelle génération de capteurs CMOS à grande dynamique, grande résolution, faible bruit et haute qualité d'images.

Période : 2008 - 2012

Montant : 14.5 M€ (dont 766 k€ pour LE2I)

Partenaires : 12 partenaires européens dont Thomson, Thales, E2V, Fraunhofer Institute

Porteur du projet : Grass Valley (NL)

Responsable LE2I : D. Ginhac

3. Projet européen Large-scale integrating project (IP) FP6-IST-2004-2.4.2

Sujet : « **STELLA : Stretchable Electronics for Large Area Applications** »

Résumé : La collaboration avec URGO a pour objectif de développer un prototype expérimental embarqué fin, extensible, sans fil, capable de mesurer des pressions à l'interface de la peau et d'une bande de contention.

Période : 2006 – 2010

Montant : 13.1 M€ (dont 40 k€ pour LE2I)

Partenaires : 10 partenaires européens dont Urgo, Philips, CEA, IMEC

Porteur du projet : Freudenberg

Responsable LE2I : M. Paindavoine

7.3 Contrats nationaux publics (3 en cours, 5 non retenus, 6 terminés)

1. Projets Exploratoires PluridisciplinaireS - PEPS (projet non retenu)

Sujet : « **Nouveau concept de détecteurs photoniques rapides à haute résolution et haute sensibilité appliqués à l'imagerie Gamma** »

Résumé : Ce projet vise à explorer un nouveau concept de détecteur photonique à grande sensibilité en technologie CMOS basé sur l'imagerie par rayonnement gamma diffusé.

Période : 2008 – 2010

Montant : 10k€ (dont 5 k€ pour LE2I)

Partenaires : ETIS

Porteur du projet : ETIS

Responsable LE2I : D. Ginhac

2. ANR Architectures du futur 2007

Sujet : « **PANINI : Programme Architectures Nano-électroniques Intégrées Neuro Inspirées** »

Résumé : Le projet PANINI vise à étudier l'utilisation et la programmation d'assemblées de composants nanoélectroniques dans l'optique d'implémenter des applications de traitements des images et de reconnaissance des formes

Période : 2008 – 2011

Montant : 1.4M€ (dont 163 k€ pour LE2I)

Partenaires : 5 partenaires dont CEA LEM, CEA LIST, IEF, IMS

Porteur du projet : IEF

Responsables LE2I : M. Paindavoine, B. Heyrman

3. ANR Architectures du futur 2007

Sujet : « **PACS : Programmable Architecture for CMOS Sensor** »

Résumé : Le projet PACS a pour objectif de développer une architecture de calcul innovante capable de supporter des traitements d'images évolués au plus près du capteur d'image.

Période : 2008 – 2011

Montant : 1.3M€ (dont 295 k€ pour LE2I).

Partenaires : 4 partenaires dont CEA LIST, IEF, BERTIN Technologies

Porteur du projet : CEA LIST

Responsables LE2I : M. Paindavoine, D. Gin hac

4. ANR Technologie pour la Santé - TecSan2006

Sujet : « **SIMACTION : Reconditionnement moteur par observation et simulation mentale** »

Résumé : Le projet SIMACTION vise à développer des plateformes matérielles et logicielles basées sur la Réalité Virtuelle et dédiées à la rééducation et au réapprentissage moteur.

Période : 2007 – 2010

Montant : 1.3 M€ (dont 404 k€ pour LE2I)

Partenaires : 6 partenaires dont LEAD, ERM207, CHU Dijon

Porteur du projet : ERM207 INSERM

Responsable LE2I : F. Merienne

5. ANR Architectures du futur 2006 (projet non retenu)

Sujet : « **SAVES: Smart Architectures for Vision Embedded Systems** »

Résumé : Le projet SAVES vise à définir une architecture matérielle innovante dédié au traitement d'image embarqué capable d'exécuter à la fois des traitement de proximité très réguliers et des traitements de haut niveau très irréguliers dépendant des données.

Période : 2006 – 2009

Montant : 3.2M€ (dont 528 k€ pour LE2I)

Partenaires : 9 partenaires dont France Telecom, CEA, LASMEA, IEF, CMM

Porteur du projet : CMM

Responsables LE2I : M. Paindavoine, B. Heyrman

6. ANR Programme systèmes Interactif et Robotique – PsiROB (projet non retenu)

Sujet : « **COAP² : Cognitive Action Perception Project** »

Résumé : Le projet COAP² a pour objectif de concevoir une plateforme matérielle et logicielle de recherche autour de la dualité perception/action à partir des outils des neurosciences cognitives.

Période : 2006 – 2009

Montant : 4.4M€ (dont 918 k€ pour LE2I)

Partenaires : 7 partenaires dont LASMEA, LAAS, LIRMM, LPPA, CEA, LENA

Porteur du projet : LASMEA

Responsable LE2I : D. Gin hac

7. Contrat de Plan Etat Région - CPER

Sujet : « **Développement et caractérisation de capteurs d'images CMOS** »

Résumé : Ce financement a permis la fabrication de capteurs CMOS en technologie CMOS 0.35 μm et l'acquisition de matériel de caractérisation des circuits fabriqués.

Période : 2006 - 2007

Montant : 50 k€

Porteur du projet : D. Ginhac (LE2I)

8. Programme « ATIP Jeunes Chercheurs » du département STIC du CNRS (projet non retenu)

Sujet : « **NOLIRETINA : Vers une rétine non linéaire reconfigurable pour systèmes de vision embarqués** »

Résumé : L'objectif de ce projet est de développer un capteur CMOS massivement parallèle intégrant au sein de chaque pixel des traitements d'images non linéaires.

Période : 2005 - 2007

Montant : 50 k€

Porteur du projet : P. Marquié (LE2I)

9. Programme « ACI Jeunes Chercheurs » du Ministère de la Recherche (projet non retenu)

Sujet : « **Environnement matériel et logiciel d'aide au prototypage rapide d'applications de traitement d'images temps réel** »

Résumé : Ce projet vise la conception d'un système de vision complexe (capteur CMOS/FPGA/DSP) et le développement d'outils de haut niveau dédiés à l'implantation d'applications de traitement d'images sur ce système.

Période : 2004 - 2006

Montant : 150 k€ (dont 100 k€ pour LE2I)

Partenaires : LASMEA

Porteur du projet : D. Ginhac (LE2I)

10. Programme thématique « Calcul Logiciel » de la Région Rhône-Alpes

Sujet : « **Implantation d'algorithme de traitement d'images dans des rétines CMOS** »

Résumé : Le projet a pour objectif le développement d'algorithmes de traitement d'images et leur implantation au sein d'un système de vision de type rétine CMOS.

Période : 2003 - 2005

Montant : 374 k€ (dont 10 k€ pour LE2I)

Partenaires : LTSI, ATMEL

Porteur du projet : LTSI

Responsable LE2I : F. Yang

11. Programme « ATIP Jeunes Chercheurs » du département STIC du CNRS

Sujet : « **Localisation et Reconnaissance de visages : vers une implantation sur silicium** »

Résumé : Le projet concernait le développement d'un capteur d'images CMOS dédié à l'extraction de vecteurs caractéristiques pour la reconnaissance de visages.

Période : 2002 - 2004

Montant : 30 k€

Porteur du projet : D. Ginhac (LE2I)

12. Programme interdisciplinaire « Cognition et Traitement de l'Information »

Sujet : « **Enfant et Réalité Virtuelle** »

Résumé : L'objectif du projet est de créer un environnement virtuel adapté aux enfants et de tester l'influence du virtuel sur les performances motrices des jeunes enfants.

Période : 2002 - 2004

Montant : 47 k€ (dont 20 k€ pour LE2I)

Partenaires : LEAD, LE2I, TIMC-IMAG

Porteur du projet : LEAD

Responsable LE2I : D. Ginhac

13. Projet « Jeunes Chercheurs » du GDR ISIS

Sujet : « **Etude et évaluation d'un composant de vision CMOS** »

Résumé : Le projet concerne la définition et l'étude d'un capteur CMOS à accès rapide totalement aléatoire.

Période : 2002 - 2003

Montant : 8 k€ (dont 4 k€ pour LE2I)

Partenaires : LASMEA

Porteur du projet : LASMEA

Responsable LE2I : D. Ginhac

14. Contrat de Plan Etat Région - CPER

Sujet : « **Développement de l'activité conception CMOS de capteurs d'images** »

Résumé : L'objectif de ce projet est l'acquisition de matériel de CAO électronique (stations, logiciels) dédié à la conception de capteurs CMOS.

Période : 2002 - 2003

Montant : 34 k€

Porteur du projet : D. Ginhac (LE2I)

7.4 Contrats privés (4 terminés)

1. Contrat Immersion

Sujet : « **Définition et Conception d'un gant de données pour enfant de 6 ans** »

Résumé : Dans le projet « Enfant et Réalité Virtuelle », j'ai collaboré avec les sociétés Immersion et 5DT pour définir et tester un gant de données pour des enfants de 6 ans.

Période : 2002 – 2003

Porteur du projet : D. Ginhac (LE2I)

2. Contrat Filiber

Sujet : « **Détection de défauts sur des étiquettes imprimées** »

Résumé : Le projet consiste en une étude de faisabilité d'un système de contrôle et de détection de défaut en sortie d'impression d'étiquettes de vins

Période : 2002 – 2003

Montant : 2 k€

Porteur du projet : D. Ginhac (LE2I)

3. Contrat Mecerlec

Sujet : « **Détection de défauts sur des pièces plastiques** »

Résumé : Le projet consiste en en une étude de faisabilité d'un système de contrôle et de détection de défaut en sortie de la chaîne de fabrication de pièces plastiques.

Période : 2002 – 2003

Montant : 2 k€

Porteur du projet : D. Ginhac (LE2I)

4. Contrat Kormann

Sujet : « **Conception de papier peint à la demande** »

Résumé : Le projet consiste en en une étude de faisabilité d'un système de conception de papier peint à la demande.

Période : 2002 – 2003

Montant : 6 k€

Porteur du projet : D. Ginhac (LE2I)

8 Collaborations scientifiques (en dehors des projets ANR)

8.1 Au niveau national (6)

1. Collaboration avec **M.D. Giraud** (depuis 2006)

Laboratoire : UMR CNRS 6152 « Mouvement et Perception », Université Aix Marseille II

Sujet : Apprentissage de régularités dans des tâches motrices continues complexes

2. Collaboration avec **T. Pozzo** (2006 – 2007)

Laboratoire : INSERM/ERM 207 « Motricité Plasticité », Université de Bourgogne

Sujet : Apprentissage implicite de mouvements biologiques / non biologiques dans des tâches de pointage

3. Collaboration avec **C. Ferrel-Chapus** (2002 – 2006)

Laboratoires : UMR CNRS 5022 « Laboratoire d'Etude de l'Apprentissage et du Développement », Université de Bourgogne et Equipe EA 2931 « Sport et Culture », Université Paris X - Nanterre

Sujet : Développement d'environnement de réalité virtuelle et d'applications dédiées

4. Collaboration avec **P. Perruchet** (2002 – 2007)

Laboratoire : UMR CNRS 5022 « Laboratoire d'Etude de l'Apprentissage et du Développement », Université de Bourgogne

Sujet : Apprentissage implicite en situations discrètes et continues

5. Collaboration avec **J. Sérot** (2000 – 2002)

Laboratoire : UMR CNRS 6602 « Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique », Université Blaise Pascal de Clermont-Ferrand

Sujet : Prototypage rapide d'applications parallèles de traitement d'images

6. Collaboration avec **F. Berry** (depuis 2000)

Laboratoire : UMR CNRS 6602 « Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique », Université Blaise Pascal de Clermont-Ferrand

Sujet : Conception de circuits CMOS pour la vision active

8.2 Au niveau international (3)

1. Collaboration avec **S. Chambaron** et **A. Cleeremans** (depuis 2006)

Laboratoire : Cognitive Science Research Unit, Université Libre de Bruxelles - Belgique

Sujet : Apprentissage Implicite de séquences, modélisation connexionniste

2. Collaboration avec **P. Peigneux** (depuis 2006)

Laboratoires : Unité de Recherches en Neuropsychologie et Neuroimagerie Fonctionnelle (UR2NF), Université Libre de Bruxelles - Belgique

Sujet : Apprentissage Implicite et Imagerie cérébrale

3. Collaboration avec **E. Prasetyo** et **N. Sultan** (depuis 2005)

Laboratoire : Université de Gunadarma - Indonésie

Sujet : Conception de circuits VLSI

9 Publications et Communications

Les paragraphes suivants sont consacrés à la présentation exhaustive de toutes les publications et communications réalisées depuis 1995. Le classement a été fait en quatre principales catégories :

- Publications dans une revue internationale ou nationale :
 - Articles directement liés aux travaux de thèses réalisés au LASMEA et publiés jusqu'en 2001
 - Articles directement liés aux travaux de recherche effectués depuis mon recrutement en tant que Maître de conférences au LE2I à Dijon
- Chapitres de livre
- Communications dans un congrès international avec actes
- Communications dans un congrès national avec actes

A titre d'information, les publications internationales soumises et les publications en cours de rédaction ont été également ajoutées en fin de liste.

Pour toutes les publications dans une revue, les facteurs d'impacts publiés dans Journal Citation Report² sont systématiquement indiqués. Pour les articles récents (2008), le facteur d'impact indiqué est celui de 2007 (dernière valeur connue). De plus, le nombre de citations évalué par Scholar Google³ est également donné pour les articles déjà parus.

9.1 Récapitulatif

Articles :	7 (+3 soumis)
Articles de thèse	3
Chapitres de livres :	2
Conférences internationales :	21
Conférences nationales :	11
Total des publications :	44

² voir <http://isiwebofknowledge.com> pour plus d'informations. Le facteur d'impact représente, pour une année donnée, le rapport entre le nombre de citations sur le nombre d'articles publiés par un journal, sur une période de référence de deux ans.

³ Voir <http://scholar.google.com> pour plus d'informations.

9.2 Publications dans une revue internationale ou nationale – Travaux de recherche au LE2I (7)

1. J. Dubois, **D. Ginhac**, M. Paindavoine, B. Heyrman *A 10 000 fps CMOS Sensor With Massively Parallel Image Processing*. IEEE Journal of Solid-State Circuits, 43(3): 706-717, 2008
(Impact Factor 2007 : 2.254)
2. S. Morfu, P. Marquié, B. Nofiélé, **D. Ginhac**. *Nonlinear systems for Image Processing*. Advances in Imaging and Electron Physics, Elsevier, 152 : 79-153, 2008.
(Impact Factor 2007 : 1.026)
3. S. Chambaron, **D. Ginhac**, P. Perruchet. *Methodological issues and computational software dedicated to SRT tasks*. Behavior Research Methods, 40(2): 493-502, 2008.
(Impact Factor 2007 : 0.963)
4. S. Chambaron, **D. Ginhac**, P. Perruchet. *Variations procédurales dans une tâche de TRS : quel est l'impact sur l'apprentissage*. L'Année Psychologique, 108(3):, 2008.
(Impact Factor 2007 : 0.380)
5. S. Chambaron, **D. Ginhac**, C. Ferrel-Chapus, P. Perruchet. *Implicit learning of a Repeated Segment in Continuous Tracking: A Reappraisal*. The Quarterly Journal of Experimental Psychology, 59A: 845-854, 2006
(Impact factor 2006 : 2.154 – Citations Scholar Google : 6)
6. F. Yang, M. Paindavoine, **D. Ginhac**, J. Dubois. *Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques*. Traitement du Signal, 22 (5) : 549-562, 2005.
7. J. Sérot, **D. Ginhac**. *Skeletons for parallel image processing: an overview of the SKiPPER project*, Parallel Computing, 28(12) : 1785-1808, 2002.
(Impact Factor 2002 : 0.625 - Citations Scholar Google : 46)

9.3 Publications dans une revue internationale ou nationale – Travaux de thèse au LASMEA (3)

8. J. Sérot, **D. Ginhac**, R. Chapuis, J.P. Dérutin. *Fast prototyping of parallel vision applications using functional skeletons*. Journal of Machine Vision and Applications, 12(6) : 271-290, 2001.
(Impact Factor 2001 : 0.373 - Citations Scholar Google : 9)
9. J. Sérot, **D. Ginhac**, J.P. Dérutin. *SKiPPER: a skeleton-based parallel programming environment for real-time image processing applications*. Parallel Computing Technologies Lecture Notes on Computer Science 1662: 296-305. Springer, 1999.
(Impact Factor 1999 : 0.872 - Citations Scholar Google : 26)
10. **D. Ginhac**, J. Sérot, and J.P. Dérutin. *Evaluation de l'outil SynDEX en vue de prototypage rapide d'applications de traitement d'images sur machine MIMD-DM*. Traitement du Signal, 14 (6) : 605-613, 1997
(Citations Scholar Google : 7).

9.4 Chapitres de livre (2)

11. **D. Ginhac**, F. Yang, X. Liu, J. Dang, M. Paindavoine. *Robust Face Recognition System based on a multi-views face database* In K. Delac, M. Grgic and M.S. Bartlett (eds), *Face Recognition - State of the Art in Cognitive and Computational Processes*, Vienna, Austria, In press
12. **D. Ginhac**, F. Yang, M. Paindavoine. *Design, Implementation and Evaluation of Hardware Vision Systems dedicated to Real-Time Face Recognition*. In K. Delac and M. Grgic (eds), *Face Recognition*, Vienna, Austria, pp. 123-148 – Juillet 2007.

9.5 Communications à un congrès international avec actes (21)

13. J. Dubois, **D. Ginhac**, M. Paindavoine. *A Programmable Vision Chip with High Speed Image Processing*. 28th International Congress on High-Speed Imaging and Photonics (ICHSIP 28)
14. S. Chambaron, A. Destrebecqz, **D. Ginhac**, A. Cleeremans. *Influence of response-stimulus interval (RSI) on sequence learning*. XXIX International Congress of Psychology ICP, Berlin, July 20-25, 2008.
15. S. Chambaron, A. Destrebecqz, **D. Ginhac**, A. Cleeremans. *The role of time and pace in sequence learning*, 12th annual meeting of the Association for the Scientific Study of Consciousness (ASSC12), Taipei (Taiwan), June 19-22, 2008.
16. S. Chambaron, **D. Ginhac**, A. Cleeremans, P. Peigneux. *Learning discrete and continuous regularities in two-dimensional settings*. BAPS 2008 annual meeting, University of Leuven, May 26, 2008.
17. S. Chambaron, A. Destrebecqz, **D. Ginhac**, A. Cleeremans. *Influence of the response-stimulus interval on implicit sequence learning: constant vs. variable RSIs*. BAPS 2008 annual meeting, University of Leuven, May 26, 2008.
18. J. Dubois, **D. Ginhac**, M. Paindavoine. *A Multi-Processing 10 000 frames/s CMOS Image Sensor* Workshop on Design and Architectures for Signal and Image Processing, DASIP 2007, Grenoble, France, Novembre 2007.
19. M. Paindavoine, J. Dubois, R. Mosqueron, B. Heyrman, J. Dubois, **D. Ginhac**. *High speed camera with embedded image processing*. 6th International Workshop on Embedded System, September 6-7, 2007, Vaasa, Finland, Septembre 2007.
20. S. Chambaron. **D. Ginhac**. *Procedural Variations around a SRT task* BAPS 2007 - Belgian Association for Psychological Science, Louvain la Neuve, Belgique, 1 Juin 2007.
21. S. Chambaron. **D. Ginhac**. *Implicit learning of sequences: discreteness versus continuity* European Workshop On Movement Science - EWOMS 2007, Amsterdam, Pays-Bas, Juin 2007
22. J. Dubois, **D. Ginhac**, M. Paindavoine. *VLSI design of a high-speed CMOS image sensor with in-situ 2D programmable processing*. EUSIPCO, Florence, Italy, Sept 2006
23. J. Dubois, **D. Ginhac**, M. Paindavoine. *A single-chip 10 000 frames/s CMOS sensor with in-situ 2D programmable image processing*. IEEE International Workshop on CAMPS 2006, Montreal, Quebec, Canada, Sept 2006

24. J. Dubois, **D. Ginhac**, M. Paindavoine. *Design of a 10 000 frames/s CMOS sensor with in-situ image processing*. ReCoSoC06, Montpellier, France, July 2006
25. S. Chambaron, **D. Ginhac**, P. Perruchet. *Is Learning in SRT Tasks Robust Across Procedural Variations?* The Twenty-Eighth Annual Conference of the Cognitive Science Society - CogSci 2006, Vancouver, Canada, July 2006.
26. E. Prasetyo, **D. Ginhac**, M. Paindavoine. *Design and Implementation of a 8 bits Pipeline Analog to Digital Converter in a 0.6 μm CMOS technology*, Indonesian Student's Scientific Meeting in Europe 2005, Paris, Sept 2005.
27. **D. Ginhac**, E. Prasetyo, M. Paindavoine, B. Heyrman. *Principles of a CMOS sensor dedicated to face tracking and recognition*. Proceedings: IEEE International Workshop on CAMP 2005, Palermo, Italy, pp. 33-38, July 2005.
28. S. Chambaron, **D. Ginhac**, P. Perruchet. *Implicit motor learning in discrete vs continuous tasks*. European Workshop On Movement Science, Vienna, Austria, June 2005.
29. **D. Ginhac**, E. Prasetyo, M. Paindavoine. *CMOS sensor for face tracking and recognition*, 26th International Congress on High Speed Photography and Photonics (HSPP'04), Alexandria (Virginia USA), Sept 2004.
30. N.S. Salahuddin, **D. Ginhac**, M. Paindavoine, M. Parmentier, N. Tamda. *A CMOS image sensor dedicated to medical gamma camera application*, 26th International Congress on High Speed Photography and Photonics (HSPP'04), Alexandria (Virginia USA), Sept 2004.
31. N. Malasne, F. Yang, M. Paindavoine, **D. Ginhac**. *Face Tracking and Recognition: from algorithm to implementation*. SPIE's 47th Annual Meeting, Advanced Signal Processing Algorithms, Architecture and Implementations, Seattle, USA, Aug 2002.
32. N. Malasne, F. Yang, M. Paindavoine, **D. Ginhac**. *RBF Neural Networks Applied to Face Tracking and Recognition* QCAV 2001, Le Creusot, France, May 2001.
33. **D. Ginhac**, J. Sérot, and J.P. Dérutin. *Fast prototyping of image processing applications using functional skeletons on MIMD-DM architecture*. IAPR Workshop on Machine Vision Applications, pages 468-471, Chiba, Japan, Nov. 1998.

9.6 Communications à un congrès national avec actes (11)

34. Chambaron, S., Delbecque, L., **Ginhac, D.**, Holender, D. & Cleeremans, A. *Action, Observation et Imagerie Mentale : Apports de l'apprentissage implicite au domaine moteur*, Journées de la Société Française de Psychologie du Sport SFPS 2008, Quiberon, 25-29 Mars, 2008.
35. J. Dubois, **D. Ginhac**, M. Paindavoine. *Un Capteur d'Images Reconfigurable dédié à l'Imagerie Rapide, aux Traitements d'Images Linéaires et Réseaux Convolutifs*. 8^{ème} colloque sur le Traitement Analogique de l'Information, du Signal et ses Applications, TAISA 2007, Lyon, France, Octobre 2007
36. N.S Salahuddin, M. Paindavoine, **D. Ginhac**, M. Parmentier, N Tambda. *Conception de photodiodes CMOS dédiées à l'Imagerie Gamma*. Imagerie pour les Sciences du Vivant - Instrumentation IMVIE-3, ECRIN, Paris, France, Juin 2006
37. S. Chambaron, **D. Ginhac**, P. Perruchet. *Apprentissage moteur implicite: variations autour d'une tâche TRS*. Congrès national de la Société Française de Psychologie, Nancy, Sept. 2005.

38. P.K Tahej, **D. Ginhac**, I. Olivier, C. Ferrel-Chapus. *Structuration de l'espace et conversion des informations visuelles en coordonnées motrices*. 11ème Congrès International de l'ACAPS, Paris, Oct. 2005.
39. **D. Ginhac**, E. Prasetyo, M. Paindavoine. *Localisation et Reconnaissance de visages : Vers une implantation sur silicium*, IEEE Signaux, Circuits et Systèmes 2004, Monastir (Tunisie) Mars 2004.
40. N. Malasne, F. Yang, M. Paindavoine, **D. Ginhac**. *Implantation temps réel d'un algorithme de localisation et de reconnaissance de visages sur un FPGA*. 18ème Colloque GRETSI sur le traitement du signal et des images, Toulouse, Sept 2001
41. **D. Ginhac**, J. Sérot, J.P. Dérutin, R. Chapuis. *SKiPPER ; un environnement de programmation parallèle fondé sur les squelettes et dédié au traitement d'images*. 17ème Colloque GRETSI sur le traitement du signal et des images, pages 1209-1212, Vannes, Sept 1999.
42. **D. Ginhac**, J. Sérot, J.P. Dérutin. *Utilisation de squelettes fonctionnels au sein d'un outil d'aide à la parallélisation*. 4èmes Journées Adéquation Algorithme Architecture en Traitement du Signal et Image, Saclay, Jan 1998.
43. **D. Ginhac**, J. Sérot, J.P. Dérutin. *Vers un outil d'aide à la parallélisation fondé sur les squelettes*. 16ème Colloque GRETSI sur le traitement du signal et des images, Grenoble, Sept 1997.
44. **D. Ginhac**, J. Sérot, J.P. Dérutin. *Evaluation de l'outil SynDEx pour l'implantation d'un algorithme d'étiquetage en composantes connexes sur la machine Transvision*. 3èmes Journées Adéquation Algorithme Architecture en Traitement du Signal et Image, Toulouse, Jan 1996.

9.7 Publications soumises dans une revue internationale (3)

45. **D. Ginhac**, J. Dubois, M. Paindavoine, B. Heyrman *A SIMD Programmable Vision Chip with High Speed Focal Plane Image Processing*. Soumis en mars 2008 à EURASIP Journal of Embedded Systems - Special Issue on Design and Architectures for Signal and Image Processing (DASIP 2007) - Article en révision mineure (Impact factor non encore calculé car la revue a été créée en 2006)
46. **D. Ginhac**, E. Prasetyo, F. Yang, M. Paindavoine. *VLSI Design of a smart camera for face recognition*. Soumis en avril 2008 à Journal of Electronic Imaging. (Impact factor 2007 : 0.455)
47. **D. Ginhac**, J. Dubois, M. Paindavoine, B. Heyrman *A high speed programmable focal-plane SIMD vision chip*. Soumis en avril 2008 à Analog Integrated Circuits and Signal Processing - Special Issue on Traitement Analogique de l'Information, du Signal et ses Applications (TAISA 2007) (Impact factor 2007 : 0.279)

9.8 Publications en cours de rédaction (4)

48. **D. Ginhac**, J. Dubois, M. Paindavoine. *Pixel Responsivity enhancement by photodiode shape optimization*. Soumission prévue à IEEE Transactions on Electron Devices. (Impact factor 2007 : 2.165)

-
49. **D. Ginhac**, J. Dubois, M. Paindavoine. *Design of a CMOS analog Four-Quadrant Multiplier*. Soumission prévue à Electronics Letters.
(Impact factor 2007 : 1.009)
 50. **D. Ginhac**, F. Yang, M. Paindavoine. *A massively parallel smart image sensor dedicated to face recognition*. Soumission prévue à IEEE Transactions on Neural Networks.
(Impact factor 2007 : 2.769)
 51. **D. Ginhac**, N.S. Salahuddin, M. Paindavoine. *VLSI Design of CMOS Photodiodes for Gamma Camera Application*. Soumission prévue à IEEE Transactions on Nuclear Science.
(Impact factor 2007 : 1.107)

10 Récapitulatif des travaux de recherche et responsabilités administratives

10.1 Publications

- Nombre d'articles publiés dans des revues scientifiques : **10**
 - 7 dans des revues d'audience internationale
 - 3 dans des revues d'audience nationale
- Nombre d'articles soumis : **3** dont 1 en révision mineure
- Nombre de chapitres de livres internationaux: **2**
- Nombre d'articles publiés dans des conférences internationales : **21**
- Nombre d'articles publiés dans des conférences nationales : **11**

10.2 Co encadrement de thèse (à 70 %) et de DEA

- Thèses soutenues : **3**
 - Eri Prasetyo (2005), Nur Salahuddin (2007), Jérôme Dubois (2008)
- Thèses à soutenir : **2**
 - Julien Steinbrunn (2009), Auguste N'Goua (2011)
- Stages de recherche de DEA : **4**

10.3 Activités contractuelles

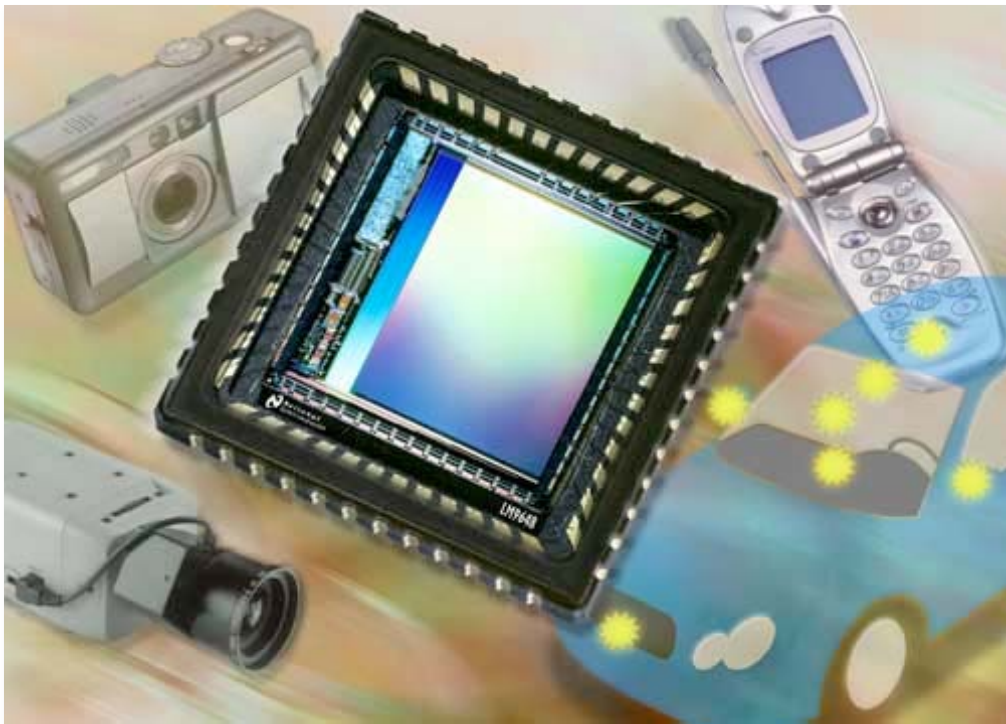
- Contrats de type public : **10**
- Contrats de type privé : **4**
- Montant total : **1060 k€**

10.4 Responsabilités administratives

- Responsable pédagogique de la 5^{ème} année du Département InfoTronique de l'ESIREM

Partie 2 :

Dossier de recherche



1 Introduction

En septembre 2000, ma nomination comme Maître de Conférences (61^{ème} section) à l'ESIREM, école d'ingénieurs de l'Université de Bourgogne m'a permis d'intégrer le Laboratoire Electronique Informatique et Image (LE2I - UMR CNRS 5158). Ce recrutement s'inscrivait dans une double problématique de recherche menée par l'équipe « Architecture et capteurs » de ce laboratoire et constituait pour moi une nouvelle orientation scientifique.

Premièrement, il s'agissait de développer une nouvelle thématique concernant la **conception microélectronique de capteurs CMOS intelligents**, appelés rétines artificielles. Une rétine artificielle peut être vue comme un circuit intégré associant sur un même substrat de silicium un capteur d'images et une architecture parallèle de traitements dans le but de constituer un système de vision compact qui évite les transferts de l'un vers l'autre. On donne souvent le qualificatif de « capteur intelligent » aux rétines artificielles en raison de leurs capacités à traiter l'information directement au plus proche du capteur et extraire les caractéristiques pertinentes de l'image. Les améliorations successives de la technologie CMOS rendent aujourd'hui possible l'intégration d'opérateurs sophistiqués de traitement des images au niveau du circuit, des colonnes, voire au niveau de chaque pixel du capteur. Ces propriétés sont particulièrement intéressantes dès lors que l'on s'intéresse à la conception de systèmes embarqués capables à la fois d'acquérir des images, de les prétraiter et d'extraire des informations pertinentes du flot vidéo. Introduire de l'intelligence sous forme de traitements programmable au niveau du plan focal constitue un axe de recherche particulièrement innovant pouvant conduire à court et moyen terme au développement de nouvelles applications haute performance de traitement d'images embarqué.

Deuxièmement, il s'agissait de renforcer la thématique « **Adéquation Algorithme Architecture** » du LE2I. Cet axe majeur de l'équipe concerne aussi bien la mise en œuvre de systèmes d'acquisition d'images avec des architectures matérielles adéquates que le développement d'applications temps réel de traitement d'images à fortes contraintes. L'objectif majeur de l'équipe est la validation des concepts théoriques par la réalisation de démonstrateurs à base de systèmes électroniques et informatiques. Cette activité de recherche constituait une suite logique à mes travaux réalisés pendant ma thèse au sein du laboratoire LASMEA (LABoratoire des Sciences des Matériaux pour l'Electronique et d'Automatique) de l'Université Blaise Pascal de Clermont Ferrand. Cela m'offrait l'opportunité de renforcer mon activité 1) sur le développement d'outils pour des plateformes de prototypage rapide et 2) sur la mise au point d'applications de traitement d'images à fortes contraintes temporelles.

Troisièmement, depuis 2002, une partie de mes activités est consacrée à des travaux de recherche en **Algorithmie pour les Sciences Cognitives**. Ces activités de recherche ont été initiées dans le cadre d'une collaboration avec le laboratoire de psychologie de l'Université de Bourgogne. De plus, depuis 2007, je suis Chercheur Visiteur au laboratoire de Sciences Cognitives de l'Université Libre de Bruxelles dans le cadre d'un Congé de Recherche (CRCT) obtenu au titre du CNU 61^{ème} section (2007-2008), puis d'un accueil en délégation au CNRS (2008-2009). Les collaborations entreprises avec ces laboratoires ont pour objectif de mieux cerner les mécanismes cognitifs mis en œuvre dans les phénomènes d'apprentissage, de reconnaissance, d'attention visuelle. Mon implication dans ces travaux porte principalement sur la mise en œuvre de systèmes informatiques et électroniques spécifiques et sur le développement logiciel de différentes applications informatiques dédiées à l'expérimentation et à l'étude de ces phénomènes cognitifs.

Les chapitres suivants sont dédiés à la présentation des différents projets effectués dans le cadre de ces 3 axes de recherche :

- Description de 6 projets liés à la conception microélectronique de capteurs d'images CMOS parmi lesquels trois ont déjà reçu le soutien du CNRS (Atip Jeunes Chercheurs - 2002) ou de l'Agence Nationale de la Recherche (2 projets ANR Architectures du futur - 2007) et enfin, un projet européen en cours d'expertise (CATRENE 2008).
- Description de 4 projets dans la thématique Adéquation Algorithme Architecture parmi lesquels deux sont financés par un projet ANR (Technologie pour la santé - 2006) et un projet européen du 6^{ème} PCRDT (STELLA - 2006).
- Description de 2 projets consacrés au développement logiciel pour les sciences cognitives dont un a reçu le soutien du CRNS (Programme Interdisciplinaire - 2002).

Un chapitre final présente une synthèse de ces travaux et décrit les perspectives et directions de recherche futures à explorer pour les thématiques présentées dans ce mémoire.

J'attire l'attention du lecteur sur le fait que les aspects techniques sont rarement abordés dans le détail. Seuls le contexte et les résultats obtenus sont décrits. Les détails techniques pourront être consultés dans les articles joints en fin de chapitres.

2 Travaux de recherche en conception de capteurs CMOS

2.1 Contexte scientifique

À l'heure actuelle, le marché de l'imagerie en pleine croissance repose principalement sur deux technologies de capteurs : CCD et CMOS. La technologie CCD (à transfert de charge) a dominé le monde des capteurs pendant plus de trente ans. Elle nécessite la mise en œuvre de procédés de fabrications particuliers largement maîtrisés et permet d'obtenir des images de haute résolution avec un bruit très faible. Les améliorations successives de la technologie CMOS dans le domaine des mémoires RAM et des processeurs en particulier ont largement contribué à rendre les capteurs d'images CMOS beaucoup plus attractifs voire très compétitifs par rapport à leurs concurrents de type CCD. Les capteurs CMOS rattrapent aujourd'hui leur retard en terme de sensibilité, dynamique de fonctionnement, qualité d'image. Ainsi, les capteurs de vision CMOS possèdent certains avantages primordiaux comparativement à leurs concurrents : intégration plus grande, consommation plus faible, coût de fabrication moindre. Ainsi, dans certains secteurs d'activités, tels que les marchés des webcams, de la téléphonie mobile, des appareils photos grand public, la technologie CMOS est devenue primordiale (cf. Fig. 1).

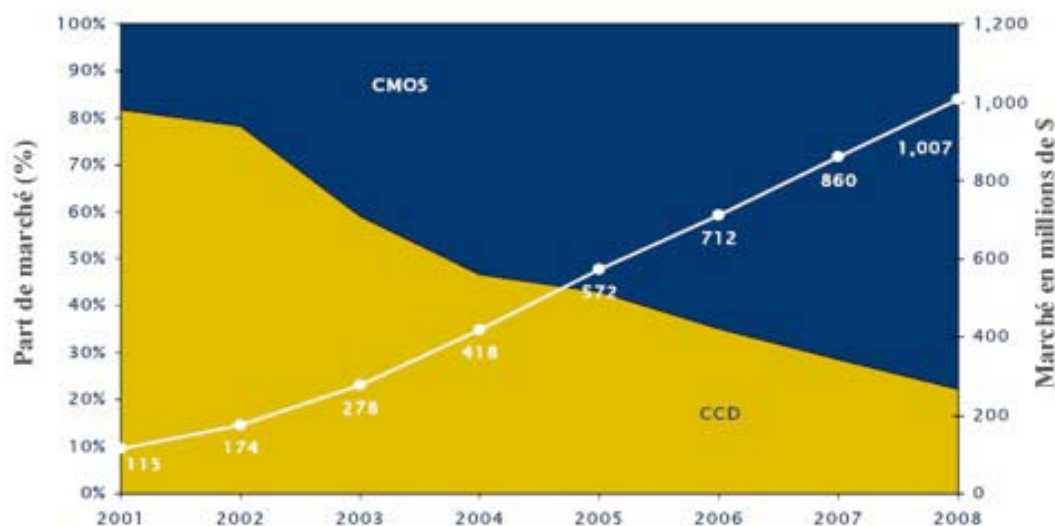


Fig. 1: Evolution du marché des technologies CCD et CMOS

De plus, les capteurs CMOS possèdent une plus grande richesse fonctionnelle. Premièrement, alors que les capteurs CCD imposent un adressage séquentiel de l'ensemble des points de l'image, les capteurs CMOS autorisent l'accès aléatoire à chaque point de l'image. L'image accessible en sortie du capteur peut être l'image complète ou être constituée uniquement d'une portion de l'image totale, selon le concept de région d'intérêt, permettant ainsi d'accroître les cadences d'acquisition.

Deuxièmement, le concept de base d'un capteur CMOS est celui du « pixel actif » (ou APS pour Active Pixel Sensor), qui permet d'associer au sein de chaque photosite au minimum un élément photosensible capable de collecter les photons et un amplificateur chargé de produire un signal électrique lié à l'illumination reçue. Partant de ce concept basique, il est tout à fait possible d'intégrer aujourd'hui des transistors supplémentaires au sein de chaque pixel permettant par exemple d'améliorer la qualité de l'image produite. Des fonctionnalités plus avancées peuvent également être intégrées soit localement au sein de chaque pixel (obturateur électronique par exemple), soit ailleurs sur le circuit (convertisseur analogique numérique, processeur de traitement de signal par exemple).

Cet avantage primordial comparativement aux capteurs CCD permet d'envisager la conception de systèmes complets intelligents intégrant un capteur d'image, et une ou plusieurs unités de traitement analogiques ou numériques, simultanément dans un même circuit électronique (cf Fig. 2).

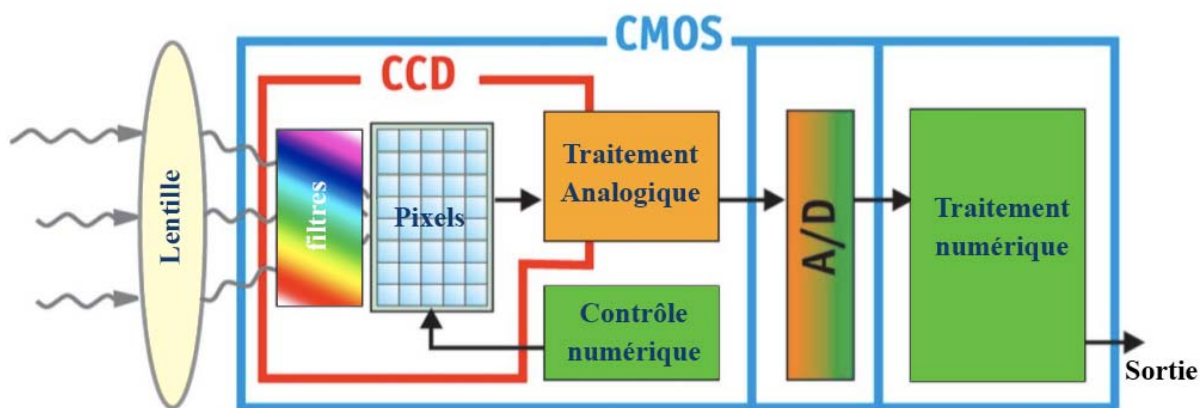


Fig. 2: Intégration des fonctionnalités des capteurs CCD et CMOS

L'intégration d'unités de traitement au sein des capteurs CMOS est un champ de recherche très actif depuis une dizaine d'années. Cependant, la plupart des travaux dans le domaine se sont focalisés jusqu'à présent sur l'intégration soit d'un processeur unique au sein d'un circuit, soit d'un processeur dédié à chaque colonne de la matrice d'acquisition. L'approche consistant à intégrer un processeur par pixel a souvent été délaissée en raison notamment de la surface non négligeable occupée par l'unité de traitement comparativement à celle du pixel. Cependant, l'avènement de technologies récentes permettant d'accroître la densité d'intégration des transistors rend maintenant cette approche tout à fait envisageable et offre de sérieux avantages. Chaque nouvelle technologie offre soit la possibilité d'intégrer plus de fonctionnalités pour une surface identique de silicium, soit la possibilité d'intégrer les mêmes fonctionnalités sur une surface moindre. La conséquence directe est l'amélioration des capteurs en terme de résolution, en terme de qualité d'image par intégration de fonctions spécifiques d'amélioration du signal (double échantillonnage corrélé, anti éblouissement, grande dynamique, ...). De plus, intégrer un processeur élémentaire par pixel offre des possibilités de calcul massivement parallèle au sein du capteur, facilitant l'acquisition des images à grande vitesse jusqu'à plusieurs milliers d'images par seconde et permettant l'implantation d'applications complexes de traitement d'images à des cadences standards.

Au sein de l'équipe Capteurs du LE2I, nos travaux dans ce domaine ont toujours été motivés par l'intégration de fonctions électroniques de traitement d'images au plus proche du plan

focal. Les travaux réalisés dans une première phase (thèses d'Eri Prasetyo et de Nur Sultan Salahuddin) concrétisés par 4 prototypes de circuits en technologie CMOS 0.6 μm , avaient des objectifs multiples. Le premier objectif était directement lié aux aspects technologiques. Il s'agissait d'acquérir des compétences et du savoir faire sur le plan de la conception des capteurs d'images CMOS et de leur caractérisation. Le deuxième objectif, du point de vue recherche, était de concevoir des capteurs d'images innovants. Dans le cadre de la thèse d'Eri Prasetyo, il s'agissait de développer un capteur d'images intégrant des traitements arithmétiques simples sur des régions d'intérêt de taille rectangulaire variable positionnées aléatoirement dans l'image globale. La thèse de Nur Sultan Salahuddin était, quant à elle, dédiée au développement et à la caractérisation de photodiodes spécifiques pour les gamma caméras.

La deuxième phase de nos travaux de recherche a débuté en 2004 avec le DEA puis la thèse de Jérôme Dubois. Ces travaux visaient le développement d'une architecture de calcul innovante capable de supporter des traitements d'images évolués au plus près du capteur d'image. L'approche retenue était l'intégration d'un processeur élémentaire de calcul au sein de chaque pixel du capteur. Les 2 prototypes réalisés en technologies CMOS 0.35 μm ont permis de valider expérimentalement des traitements d'images de type Sobel ou Laplacien à une cadence de plusieurs milliers d'images par seconde en tirant bénéfice du parallélisme massif des processeurs de calcul.

2.2 Développement de capteurs CMOS dédiés à la reconnaissance de visages (2001 - 2005)

Les systèmes électroniques autonomes dédiés à la localisation et la reconnaissance de visages en temps réel permettent d'envisager dans un avenir très proche de multiples applications dans les domaines des interfaces homme machine, de la vidéoconférence, de la téléphonie mobile ou du contrôle d'accès sécurisé par exemple. L'émergence de tels systèmes exige la mise en œuvre d'algorithmes complexes nécessitant des volumes de calculs élevés, difficilement compatibles avec une architecture de calcul classique. Une solution possible est alors de réaliser une architecture électronique dédiée à cette application. Sa spécificité réside surtout dans le fait de paralléliser massivement les calculs, permettant ainsi de traiter dans un temps limité une grande quantité de données.

Cependant, définir un tel système répondant à des critères d'embarquabilité (faible volume, consommation maîtrisée), de fiabilité (taux de reconnaissance très élevé dans n'importe quel environnement) et de rapidité (reconnaissance temps réel d'une ou plusieurs personnes en mouvement dans une scène) reste encore un problème très ouvert de nos jours. Des travaux antérieurs réalisés au sein du laboratoire LE2I (Thèse de N. Malasné en 2002) ont permis de réaliser un premier système embarqué de localisation et d'identification de visages. Reposant sur un capteur CMOS classique associé à un FPGA implémentant un réseau de neurones, cette réalisation a mis en avant le besoin de développer un capteur d'images dédié permettant l'extraction de vecteurs caractéristiques servant de point d'entrée du réseau de neurones.

C'est ce point qui a fait l'objet des travaux préliminaires menés par D. Bray lors de son DEA (2000 - 2001) suivis de la thèse d'Eri Prasetyo (2001 - 2005).

L'objectif principal était la réalisation d'un système complet de vision constitué principalement d'un capteur d'images de type CMOS et d'un composant programmable de type FPGA. Le capteur est dédié à l'acquisition des images et au prétraitement de ces images sous la forme d'une extraction de vecteurs de paramètres représentant les images. Ces

vecteurs paramètres servent ensuite d'entrée à un réseau de neurones de type RBF implanté sur le FPGA (cf. Fig. 3).

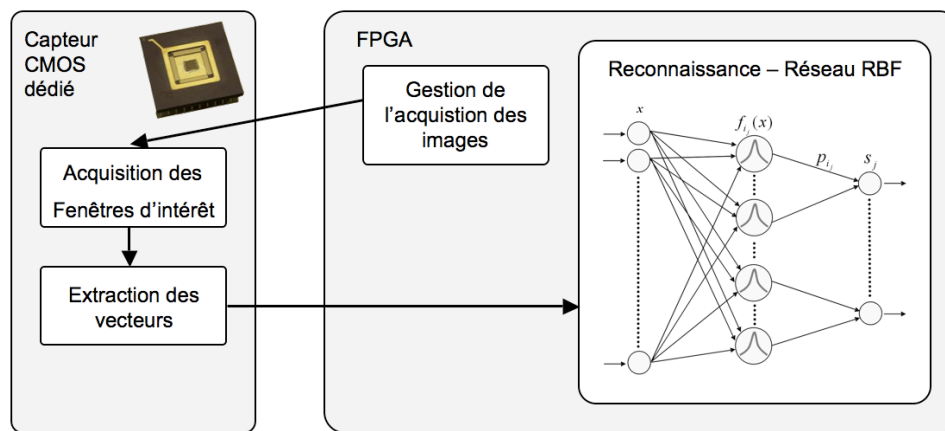


Fig. 3: Schéma du système de vision dédié à la reconnaissance de visages

Les travaux de D. Bray ont permis de réaliser une étude approfondie de l'état de l'art concernant les rétines artificielles (rétines existantes, technologies employées, applications réalisées, performances, ...) et de définir les fondements et les caractéristiques essentielles d'une rétine dédiée à la reconnaissance des visages.

Entre 2001 et 2005, j'ai co-encadré (70 %) les travaux de thèse d'E. Prasetyo. Cette thèse a été financée par le gouvernement indonésien et s'inscrit dans le cadre d'une collaboration étroite entre notre laboratoire et l'université de Djakarta (Indonésie). Cette thèse a été soutenue en octobre 2005.

L'objectif final de la thèse était la conception, la modélisation et la réalisation d'un capteur d'images en technologie CMOS 0.6 μm dédié à l'extraction de vecteurs caractéristiques des images. Tous les principaux points de la conception CMOS d'un capteur d'image dédié ont été successivement abordés dans cette thèse :

- Conception d'un photo élément de type photodiode de type N,
- Réalisation d'un pixel à limitation du temps d'intégration basé sur 4 transistors,
- Développement d'une unité de contrôle permettant l'accès aléatoire à des régions d'intérêt,
- Etude de convertisseurs analogiques numériques de type pipeline,
- Implémentation de traitement numérique des signaux sous la forme d'opérateurs arithmétiques.

Seuls les trois premiers points ont été caractérisés et validés expérimentalement sur les circuits prototypes. L'étude de la conversion analogique numérique et le traitement des signaux numériques ont seulement été simulés.

Ces travaux ont également reçu le soutien du Département STIC du CNRS sous la forme d'un programme ATIP Jeunes Chercheurs dont j'étais le bénéficiaire. Ce programme d'une durée de 2 ans (2002 - 2004) et d'un montant de 30 k€ a permis de financer la fabrication des deux prototypes de circuits en 2003 et 2004.

La validation des concepts retenus a conduit à la rédaction de 3 articles de conférences internationales [26, 27, 29], de 1 conférence francophone [39] et d'un chapitre de livre [12].

De plus, un article [46] a été soumis en avril 2008.

2.3 Développement de capteurs CMOS de type Gamma Caméra dédiés à des applications médicales (2002 - 2007)

Le principe de l'exploration radio isotopique utilisée en médecine nucléaire consiste à introduire dans l'organe étudié une molécule marquée par un radioélément, appelé traceur. On acquiert ensuite une image bidimensionnelle de la répartition volumique du traceur. Le détecteur utilisé est appelé Gamma Caméra. Le premier prototype d'une gamma caméra a été proposé par H. Anger en 1958. Il était constitué d'un cristal scintillateur et d'un ensemble de tubes photomultiplicateurs (PMTs) permettant de convertir les photons gamma en photons visibles puis en signaux électriques. Dans le but d'atteindre des résolutions spatiales de l'ordre du millimètre, tout en maintenant une bonne résolution énergétique, de nombreux travaux ont été effectués tout au long de ces dernières années sur les technologies des détecteurs utilisables en imagerie radio isotopique.

Entre 2002 et 2007, j'ai co-encadré les travaux de thèse de Nur Sultan Salahuddin (co-encadrement à 70 %). Comme celle d'Eri Prasetyo, cette thèse a été financée par le gouvernement indonésien pour une durée de quatre ans (2002-2006). La soutenance de ces travaux a eu lieu en juillet 2007.

L'objectif de cette thèse était de faire rencontrer les deux domaines de l'imagerie radio isotopique et des capteurs d'images CMOS. En effet, compte tenu des progrès importants obtenus avec ce type de capteurs, notre but a été d'étudier la possibilité d'utiliser ces nouvelles technologies pour concevoir de nouveaux détecteurs applicables en imagerie radio isotopique. L'objectif des recherches était le remplacement des photomultiplicateurs par un capteur d'images CMOS. Une telle approche présente de nombreux avantages : 1) résolution spatiale fortement accrue, 2) distance d'acquisition diminuée et plus grandes possibilités de prises de vue en raison de la compacité du système, 3) faible coût de production des procédés CMOS pour des grandes séries, ...

A partir de l'état de l'art sur les détecteurs utilisés en imagerie gamma, nous avons pu spécifier qu'il était nécessaire de concevoir un capteur capable de détecter des photons lumineux émettant dans le bleu avec une faible énergie ($10\text{nW}/\text{mm}^2$) et ceci avec une constante de temps maximale de 0,1ms. Un tel objectif est tout à fait envisageable en utilisant des photodiodes CMOS en mode courant associées avec des pré-amplificateurs de type miroirs de courants. La modélisation et la simulation en VHDL-AMS de différentes structures de photodiodes ont montré la faisabilité d'utilisation de la technologie des détecteurs d'images CMOS en imagerie gamma. Ces simulations ont été validées par la conception et à la réalisation physique de deux circuits en technologie CMOS $0.6\mu\text{m}$ (cf. Fig. 4). La caractérisation des performances des circuits a révélé que les capteurs sont en limite de sensibilité et qu'il serait nécessaire d'insérer dans le circuit un amplificateur faible bruit.

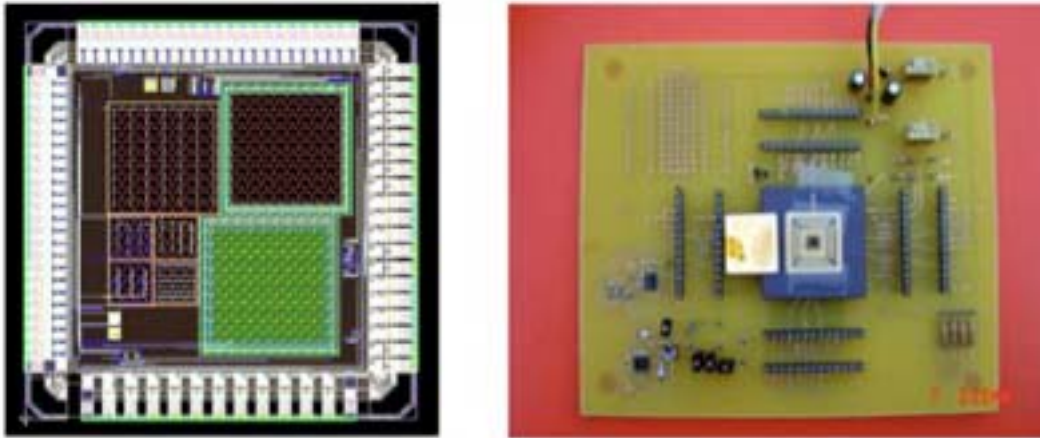


Fig. 4 : Layout du capteur CMOS et dispositif expérimental

Les premiers résultats ont été publiés en 2004 dans une conférence internationale [30] et une conférence nationale [36].

Un article en cours de rédaction [51] devrait être soumis avant la fin de l'année 2008.

2.4 Intégration de traitements d'images bas niveau au sein de capteurs CMOS programmables (2003 - 2008)

De nos jours, l'amélioration des procédés technologiques permet aux imageurs CMOS de concurrencer pleinement les capteurs de type CCD. Les capteurs CMOS ont des avantages déterminants : coût très favorable, résolution potentiellement très élevée, faible consommation électrique, accès aléatoire aux pixels et surtout intégration possible de traitements d'images au plus près des structures d'acquisition. Ces propriétés sont particulièrement intéressantes dès lors que l'on s'intéresse à la conception de systèmes embarqués capables à la fois d'acquérir des images, de les prétraiter et d'extraire des informations pertinentes du flot vidéo.

Ces capteurs dits « intelligents », irréalisables en technologie CCD, sont nommés « rétines artificielles ». Une rétine CMOS peut être vue comme une grille bidimensionnelle de « cellules », chaque cellule étant formée de l'association d'un élément photosensible (photorécepteur), et d'un élément de calcul (processeur élémentaire). Du fait du caractère massivement parallèle, une rétine peut avoir des performances en vitesse, en intégration et en consommation surpassant les solutions traditionnelles de systèmes de vision (architecture de traitement connectée à une caméra). L'approche proposée dans ces travaux consiste à associer, dans un même circuit les photo capteurs et des fonctions programmables de prétraitements des images. La plupart des travaux dans le domaine s'intéresse soit à l'intégration d'une seule unité de calcul pour le circuit VLSI total, soit à la mise en œuvre d'un ensemble d'unités dédiées à la gestion de plusieurs colonnes de l'image. En effet, intégrer un processeur élémentaire pour chaque pixel du circuit implique bien souvent un pixel surdimensionné et de fait inutilisable. Malgré cela, l'amélioration des technologies submicroniques permet d'envisager à court terme des perspectives très intéressantes en terme d'intégration de processeur élémentaire par pixel, notamment sur les aspects de traitement massivement parallèle des données acquises. Bien évidemment, cela entraîne des bénéfices majeurs pour l'acquisition à haute cadence (plusieurs milliers d'images par

seconde) et pour l'intégration au sein des circuits d'applications complexes de traitement d'images à cadence vidéo standard.

Nos travaux s'articulent donc autour de la faisabilité de rétines artificielles présentant des tailles de pixels raisonnables tout en y intégrant des traitements bas niveaux programmables. Les premiers travaux relatifs à ce projet ont démarré en 2003 avec le stage de DEA de M. Wittl (encadrement à 100%) avec le double objectif suivant : effectuer une étude approfondie de l'état de l'art du domaine (circuits existants, technologies employées) et proposer une solution d'intégration d'une unité de traitement par pixel. Ils se sont poursuivis en 2004 avec le stage de Master Recherche de J. Dubois (encadrement à 100%) qui avait pour objectif la réalisation en technologie CMOS 0.35 μm d'un pixel intégrant un élément photodétecteur associé à une unité de traitement analogique.

Depuis 2005, nos travaux (thèse de J. Dubois co encadrée à 70% et financée par une allocation de recherche ENS) ont concerné la conception et la réalisation d'un système embarqué programmable de type VSOC (Vision System On Chip) entièrement dédié à des traitements bas niveaux des images à haute cadence tels que filtrage, gradient, rehaussement de contrastes, ... Ceci nous a conduit à reconsidérer la géométrie même des photodiodes utilisées classiquement et à pousser au maximum l'intégration de transistors. Cela a permis le développement d'une première rétine 64 x 64 pixels intégrant une unité de traitement programmable composée de quelques dizaines de transistors (cf. Fig. 5).

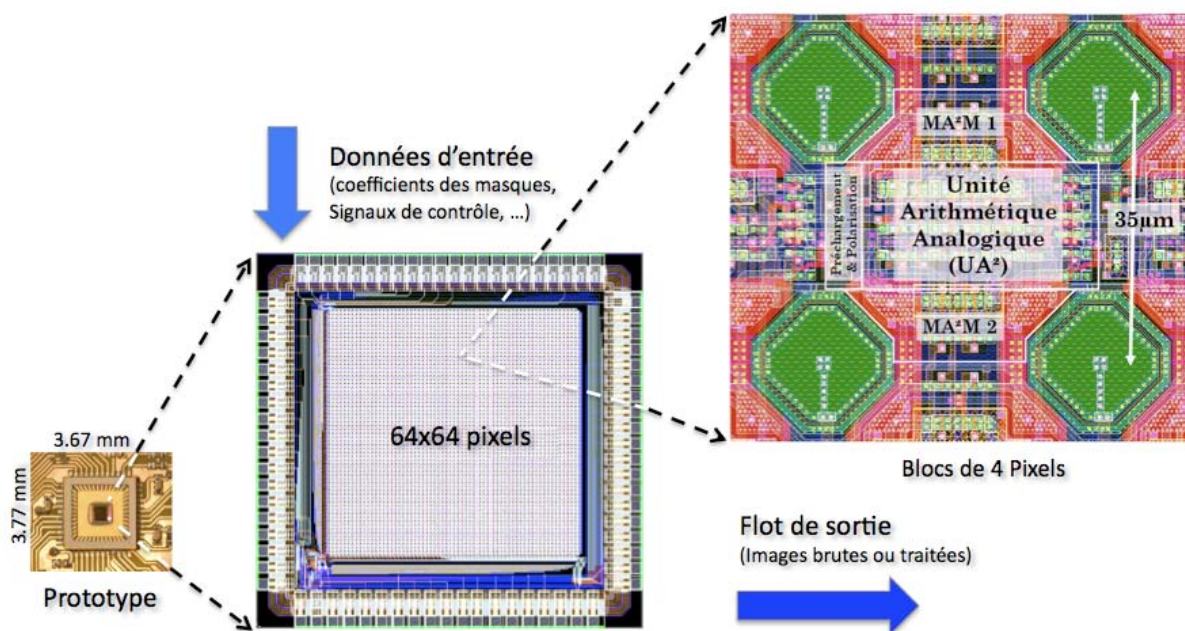


Fig. 5 : Vue d'ensemble du circuit avec un processeur élémentaire par pixel

L'objectif principal de ces travaux est (1) d'évaluer la vitesse d'acquisition du capteur afin d'atteindre la cadence de 10000 images par seconde, (2) de valider la faisabilité et l'intégration d'une unité de traitement au sein des pixels et (3) de fournir une plate-forme originale dédiée au traitement d'images embarqué.

L'évaluation des 2 circuits prototypes (en 2006 et 2007) a conduit à de très bons résultats en termes de performances d'acquisition brutes ou de traitement à très haute cadence. À titre d'exemple, la Fig. 6 montre sur la partie gauche l'image acquise à 10 000 im/s d'un

ventilateur tournant à 3700 trs/min. Sur la partie droite, on voit une succession de 16 images acquises à 5000 im/s après évaluation au sein des pixels d'un gradient de type Roberts.

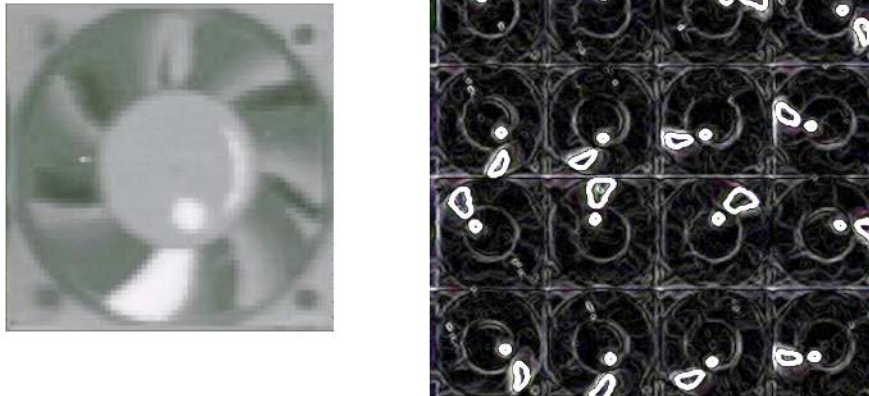


Fig. 6 : Acquisition et traitement d'images à plusieurs milliers d'images par seconde

Depuis 2003, ces travaux ont conduit à la rédaction de 2 articles de revue internationale [1, 2], de 6 conférences internationales [13, 18, 19, 22, 23, 24] et de 1 conférence nationale [35].

Deux articles supplémentaires ont été soumis en février 2008 [45 - article en révision mineure] et en mars 2008 [47].

De plus, trois nouveaux articles [48, 49, 50] sont en cours de rédaction et devraient être soumis avant la fin de l'année 2008.

2.5 Projet ANR PACS : Programmable Architecture for CMOS Sensor (2008 - 2011)

Ces travaux de recherches s'intègrent dans le cadre du projet PACS (Programmable Architecture for CMOS Sensor). Ce projet est financé par l'ANR "Architectures du futur 2007". Il est réalisé en collaboration avec plusieurs autres équipes de notoriété internationale (CEA LIST - LETI - IEF). Pour notre part, une thèse (A. Ngoua) financée dans le cadre de ce projet débutera en octobre 2008.

Les architectures de calcul dédiées au traitement d'images dans le domaine des systèmes embarqués doivent toujours rechercher le meilleur compromis entre le besoin en puissance de calcul toujours plus grand et la consommation électrique. Dans ces architectures, un processeur est souvent affecté au traitement d'une image entière, entraînant de facto une consommation électrique conséquente. Les prochaines générations de systèmes intégrant un capteur vidéo vont profiter des nouvelles technologies d'intégration améliorant ainsi les performances, l'efficacité énergétique et le coût. Les avancées récentes des technologies d'intégration permettent d'introduire près du capteur beaucoup plus de portes logiques destinées à des traitements évolués notamment grâce à l'utilisation du silicium amorphe.

Le contexte applicatif du projet PACS est principalement celui de la télésurveillance et des systèmes de sécurité. Les systèmes de sécurité connaissent, du fait des évolutions technologiques et du contexte international, une diffusion croissante. Dans ces systèmes, des chaînes de traitement automatisées intègrent des capteurs d'images intelligents permettant de limiter le flux d'informations sans cesse croissant remonté aux équipes de surveillance, et de systématiser les alertes en cas d'incident. L'intérêt principal d'une architecture de type rétine artificielle programmable réside dans la possibilité de réaliser des traitements

d'images pour un coût énergétique faible dans un volume optimisé. Les objectifs attendus des technologies de capteurs images intelligents à base de rétine, sont de disposer de capacités de traitement d'images et d'analyse de situation locales au capteur, afin de fiabiliser et enrichir l'alerte par des informations pertinentes et d'éviter de transmettre les images en continu en se limitant aux images correspondant à l'événement.

La participation du LE2I dans ce projet concerne la définition et le développement microélectronique des fonctions de traitement d'images embarqués au plus près du capteur. Les recherches menées sont dans la continuité directe de la thèse de J. Dubois. S'appuyant sur les nouvelles possibilités offertes par les technologies amorphes, les travaux de thèse d'A. Ngoua ont pour objectif le développement d'une architecture de calcul innovante capable de supporter des traitements d'images évolués au niveau des pixels avec une consommation électrique très faible grâce au parallélisme massif des processeurs de calcul. Il s'agit donc d'étudier une architecture de calcul se basant sur un tableau de processeurs, chacun connecté à un sous-ensemble de la matrice des pixels (cf. Fig. 7).. Les possibilités d'intégration 3D de la matrice photosensible sur la matrice des processeurs seront étudiées afin de pouvoir introduire des processeurs de granularité moyenne offrant des fonctions complexes et donc plus de flexibilité. La régularité des structures SIMD et leur simplicité électronique en font de bons candidats pour intégrer un grand nombre de processeurs près de la rétine.

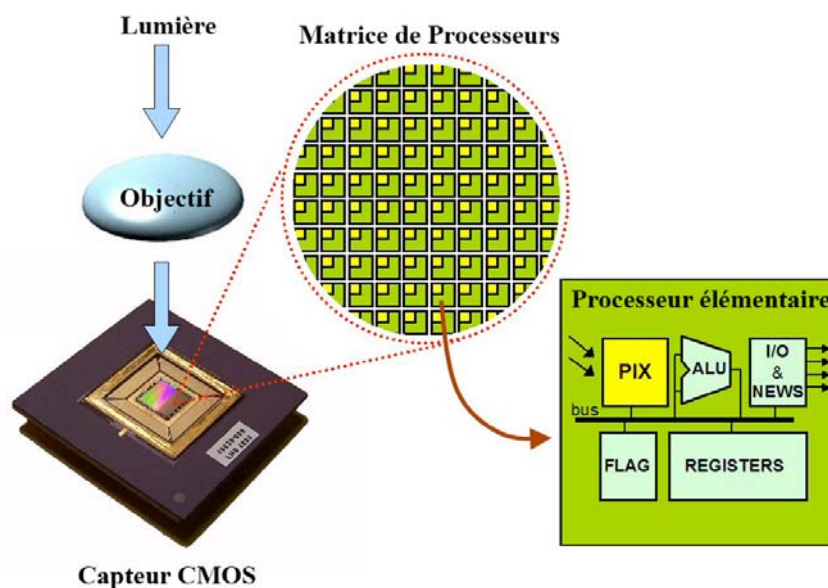


Fig. 7 : Principe de l'architecture de calcul intégrée au sein du capteur CMOS

L'objectif ambitieux de cette thèse est de développer une nouvelle génération de capteurs intelligents surclassant les capteurs existants en réalisant des opérations de traitement complexes tout en maintenant une consommation électrique basse par un parallélisme massif des unités de calcul. Une telle architecture se doit d'être à la fois très compacte, faible consommation, et facilement programmable pour permettre le développement de produits à forte intégration de services, basé sur l'analyse vidéo temps-réel. L'enjeu pour cette architecture générique étant de pouvoir réaliser plus efficacement un système de traitement vidéo, que ne le permet aujourd'hui l'association d'un capteur CMOS avec un DSP ou processeur de traitement d'image spécialisé, avec ses bus de communications et sa mémoire associée. Elle doit offrir une accessibilité aux pixels et une puissance de calcul plus

importantes que les systèmes s'appuyant sur un processeur vidéo et une plus grande flexibilité que les solutions basées ASIC.

2.6 Projet ANR PANINI : Programme Architectures Nano-électroniques Intégrées Neuro Inspirées (2008 - 2011)

Ces travaux de recherches s'intègrent dans le cadre du projet PANINI (Programme Architectures Nano-électroniques Intégrées Neuro Inspirées). Ce projet est financé par l'ANR "Architectures du futur 2007". Il est réalisé en collaboration avec plusieurs autres équipes de notoriété internationale (CEA LEM, CEA LIST, IEF, IMS). Pour notre part, un financement d'un post-doc est prévu pour l'année 2009-2010.

L'utilisation d'assemblées de composants nanoélectroniques, auto-organisées ou non, obtenues par une méthode d'assemblage « bottom-up » permettrait d'enrayer l'explosion des coûts de fabrication des circuits intégrés. Néanmoins, la programmation et l'utilisation de telles assemblées pour remplir une fonction donnée requièrent la mise en place d'une architecture d'ensemble et le développement de procédures originales. Au cours de cette procédure de programmation, l'adressage individuel de chaque composant n'est pas souhaitable pour des questions de vitesse et de coût. De plus, il est acquis que les nanocomposants ainsi construits présenteront des dispersions de caractéristiques importantes. Aussi il est important d'ajouter des propriétés d'auto-organisation et d'auto-ajustement aux assemblées ainsi construites. La difficulté liée à l'adressage individuel de ces nanocomposants nous a naturellement conduits à nous intéresser tout particulièrement aux algorithmes d'apprentissage utilisés dans le cadre des réseaux de neurones formels. Afin de placer cette étude en relation avec des assemblées réalistes de nanocomposants, nous avons choisi d'étudier des assemblées de nanotubes de carbone et de transistors à nanotubes de carbone - CNTFET (cf. Fig. 8). Dans ce projet, nous porterons une attention particulière aux composants qui permettent d'envisager un apprentissage comme les CNTFETs à commande optique (OG-CNTFET). De plus, nous étudierons les assemblées aléatoires ou auto-organisées telles que les réseaux de nanotubes multiconnectés (MCCNN) afin de déterminer leur mode de fonctionnement et identifier les fonctions, éventuellement complexes, susceptibles d'être réalisées par de tels dispositifs. Nous étudierons les possibilités de programmation d'un tel réseau en explorant notamment la possibilité d'une sensibilisation localisée par la lumière afin de définir une architecture.

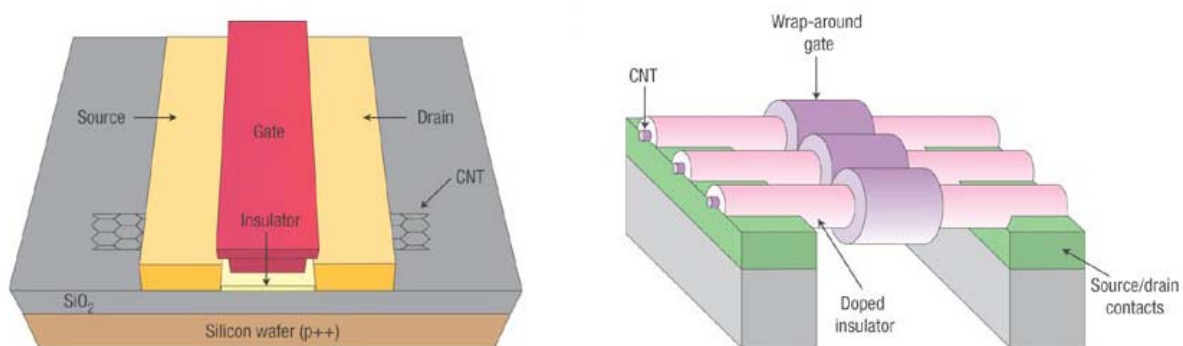


Fig. 8 : Exemples de structures de type CNTFET

Afin de pouvoir concevoir et simuler des architectures de circuits adaptés à l'apprentissage neuro- inspiré, nous adapterons des modèles compacts physiques de nanocomposants. Le caractère physique des modèles de simulation de nanocomposants permettra d'évaluer la sensibilité des caractéristiques électriques des circuits aux variations des paramètres physiques. De plus, nous pourrons ainsi vérifier par simulation l'efficacité de l'apprentissage pour la compensation des dispersions de caractéristiques physiques des composants.

La participation du LE2I dans ce projet se situe principalement au niveau de l'étude des réseaux auto-organisés qui seront exploités directement comme des classifieurs dans des applications adaptées en traitement d'images. Ces réseaux auto-organisés pourront être utilisés comme des blocs logiques booléens configurables, à l'image des LUT dans les FPGA, mais leur nature neuronale permet aussi de les exploiter directement comme des classifieurs dans des applications de traitement d'images et de reconnaissance de formes. Dans le même esprit, l'analyse du fonctionnement des réseaux aléatoires peut révéler des modes de fonctionnement inédits associés à la minimisation stochastique d'une énergie ou à un algorithme d'apprentissage non supervisé comme une carte de Kohonen.

2.7 Projet européen HiDRaLoN : High Dynamic Range Low Noise CMOS imagers (2009 - 2012)

Le projet HiDRaLoN (High Dynamic Range Low Noise CMOS imagers) est un projet européen déposé en avril 2008 dans le cadre du programme EUREKA CATRENE (Cluster for Application and Technology Research in Europe on NanoElectronics). Ce projet a été évalué positivement par le comité scientifique de CATRENE réuni en assemblée en juillet 2008 et proposé pour labellisation par le comité central CATRENE qui doit se réunir à la fin septembre 2008. Ce projet a été préparé en collaboration avec des entreprises européennes (Philips, Thomson, Thales, E2V) et avec des instituts de recherche (Fraunhofer Institute) très renommés. Il fait suite à un premier projet (HIDRIC - High Dynamic Range Image Capture) déposé à la Commission Européenne en Octobre 2007 dans le cadre du 7^{ème} PCRD, mais qui n'a pas été retenu (classement 7^{ème} sur 18 projets).

Le projet HiDRaLoN s'intéresse au développement de nouvelles approches de conception de capteurs d'images CMOS. L'objectif est de concevoir, de développer et de mettre au point de nouvelles méthodes permettant d'accroître de manière drastique la dynamique des capteurs d'images CMOS tout en conservant, voire améliorant les autres caractéristiques importantes que sont la résolution, la qualité d'images, la sensibilité. Les champs de recherche de ce projet sont largement pluridisciplinaires. On peut citer la conception et la modélisation de pixels, le développement de nouvelles stratégies de correction et d'amélioration du signal, l'intégration de convertisseurs analogiques numériques, la mise en œuvre d'un système de lecture très rapide de l'image, ...

La participation du LE2I dans ce projet se situe à deux niveaux. Le premier est le développement d'algorithmes de correction d'images permettant d'éliminer les valeurs aberrantes des pixels. Il s'agit ici de veiller à fournir en sortie de la matrice de pixels une image de la plus grande qualité possible après avoir supprimé les pixels défectueux et les non-uniformités de type bruit fixe de colonne (FPN). Deuxièmement, le projet HiDRaLoN vise à augmenter de manière significative la dynamique des capteurs CMOS. Pour cela, les travaux menés par le LE2I visent à développer un système innovant de lecture de l'image réalisant plusieurs lectures partielles de l'image tout au long de l'acquisition. Au final, l'image à la sortie du capteur est constituée de l'assemblage des différentes zones de l'image acquises à différents moments. Un tel système permettrait d'une part d'éviter les problèmes

de saturation pour les zones fortement éclairées et d'autre part de disposer d'informations suffisantes pour les zones faiblement éclairées.

Les retombées potentielles de ce projet et les champs d'applications de ces capteurs CMOS innovants sont très nombreux. On peut citer les caméras professionnelles pour la télévision haute définition ou le cinéma numérique dans le domaine artistique et culturel. Mais également, on peut envisager diverses applications pour le contrôle et la surveillance des routes, pour la surveillance des lieux publics, pour les systèmes automatisés de contrôle de qualité par vision artificielle, ... Selon les applications visées, les résultats issus de ce projet, et plus particulièrement les travaux sur le pixel, pourront permettre de concevoir des systèmes de vision parfaitement adaptés au domaine souhaité selon plusieurs directions différentes (cf Fig. 9) :

- La diminution de la taille des pixels associée pour une diagonale du capteur constante permet de libérer de l'espace en périphérie de chaque pixel, espace qui peut être utilisé pour intégrer au niveau du plan focal des traitements d'images spécifiques. Le champ d'application visé ici est principalement le domaine des machines de vision pour applications industrielles.
- La réduction de la taille des pixels pour une même diagonale de capteur permet de créer des sous groupes de pixels sur lesquels sont disposées des microlentilles dans l'objectif de pouvoir créer des images 3D des scènes acquises.
- Dans le cas d'une diagonale du capteur constante, l'utilisation de pixels de taille plus petite permet également d'augmenter la résolution horizontale et/ou verticale du capteur pour toutes les applications nécessitant une grande finesse de détails (surveillance, télévision, cinéma, photographie, ...).
- Pour une résolution constante, des pixels de taille plus petite permettent de concevoir des capteurs plus compacts et donc de réduire le coût de fabrication. Ceci est très bénéfique pour les systèmes grand public comme les webcams, la téléphonie mobile, ...

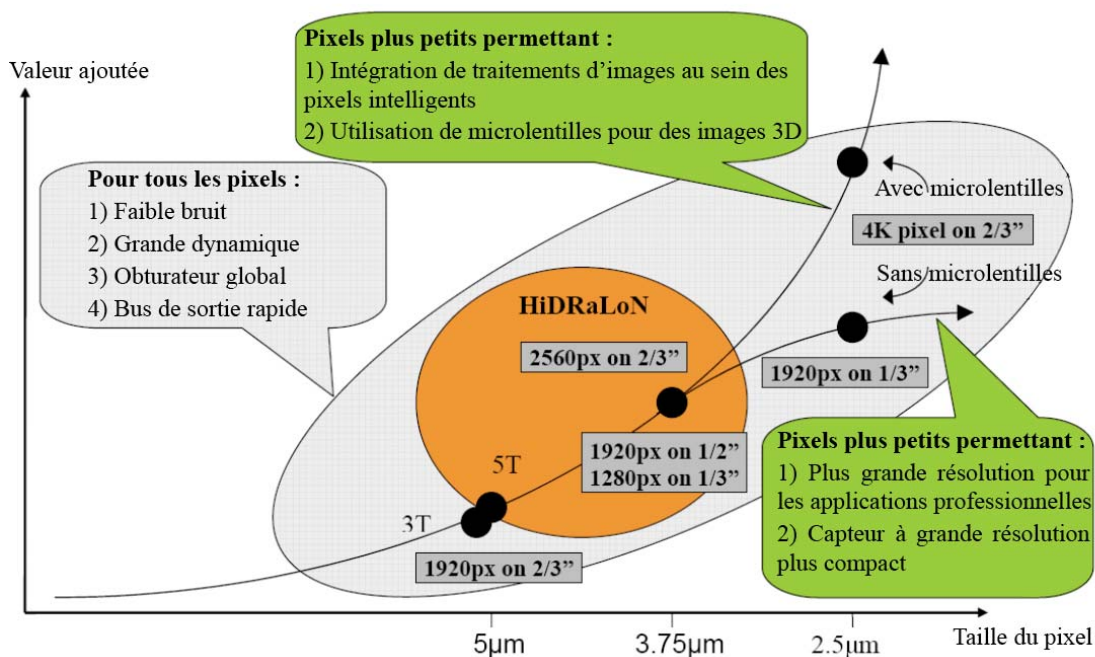


Fig. 9 : Différents champs d'applications du projet HiDRaLoN

2.8 Publications jointes

J. Dubois, **D. Ginhac**, M. Paindavoine, B. Heyrman *A 10 000 fps CMOS Sensor With Massively Parallel Image Processing*. IEEE Journal of Solid-State Circuits, 43(3): 706-717, 2008

D. Ginhac, J. Dubois, M. Paindavoine, B. Heyrman *A SIMD Programmable Vision Chip with High Speed Focal Plane Image Processing*. Soumis en mars 2008 à EURASIP Journal of Embedded Systems - Special Issue on Design and Architectures for Signal and Image Processing (DASIP 2007) - Article en révision mineure

A 10 000 fps CMOS Sensor With Massively Parallel Image Processing

Jérôme Dubois, *Student Member, IEEE*, Dominique Ginjac, *Member, IEEE*, Michel Paindavoine, *Member, IEEE*, and Barthélémy Heyrman

Abstract—A high-speed analog VLSI image acquisition and pre-processing system has been designed and fabricated in a $0.35\ \mu\text{m}$ standard CMOS process. The chip features a massively parallel architecture enabling the computation of programmable low-level image processing in each pixel. Extraction of spatial gradients and convolutions such as Sobel or Laplacian filters are implemented on the circuit. For this purpose, each $35\ \mu\text{m} \times 35\ \mu\text{m}$ pixel includes a photodiode, an amplifier, two storage capacitors, and an analog arithmetic unit based on a four-quadrant multiplier architecture. The retina provides address-event coded output on three asynchronous buses: one output dedicated to the gradient and the other two to the pixel values.

A 64×64 pixel proof-of-concept chip was fabricated. A dedicated embedded platform including FPGA and ADCs has also been designed to evaluate the vision chip. Measured results show that the proposed sensor successfully captures raw images up to 10 000 frames per second and runs low-level image processing at a frame rate of 2000 to 5000 frames per second.

Index Terms—CMOS image sensor, parallel architecture, high-speed image processing, analog arithmetic unit.

I. INTRODUCTION

TODAY, improvements in the growing digital imaging world continue to be made with two main image sensor technologies: charge coupled devices (CCD) and CMOS sensors. The continuous advances in CMOS technology for processors and DRAMs have made CMOS sensor arrays a viable alternative to the popular CCD sensors. New technologies provide the potential for integrating a significant amount of VLSI electronics into a single chip, greatly reducing the cost, power consumption, and size of the camera [1]–[4]. This advantage is especially important for implementing full image systems requiring significant processing such as digital cameras and computational sensors [5]–[7].

Most of the work on complex CMOS systems deals with the integration of sensors providing a processing unit at chip level (*system-on-chip* approach) or at column level by integrating an array of processing elements dedicated to one or more columns [8]–[11]. Indeed, pixel-level processing is generally dismissed because pixel sizes are often too large to be of practical use. However, as CMOS image sensors scale to $0.18\ \mu\text{m}$ processes and under, integrating a processing element at each pixel or group of neighboring pixels becomes feasible. More significantly, employing a processing element per pixel offers the

opportunity to achieve massively parallel computations and thus the ability to exploit the high-speed imaging capability of CMOS image sensors [12]–[15]. This also benefits the implementation of new complex applications at standard rates and improves the performance of existing video applications such as motion vector estimation [16]–[18], multiple capture with dynamic range [19]–[21], motion capture [22], and pattern recognition [23].

As integrated circuits keep scaling down following Moore's Law, recent trends show a significant number of papers discussing the design of digital pixels [24]–[27] that take advantage of the increasing number of available transistors at the pixel in order to perform analog to digital conversion. This trend is mainly motivated by the significant advantages of pixel-level analog-to-digital (A/D) conversion such as high SNR, lower power consumption, and very low conversion speeds. Nevertheless, the resulting implementations of in-pixel analog-to-digital converter (ADC) are rather area consuming, strongly restricting the image processing capability of CMOS sensors.

In this paper, we discuss hardware implementation issues of a high-speed CMOS imaging system embedding low-level image processing. For this purpose, we designed, fabricated, and tested a proof-of-concept 64×64 pixel CMOS analog sensor with per-pixel programmable processing element in a standard $0.35\ \mu\text{m}$ double-poly quadruple-metal CMOS technology. The main objectives of our design are: 1) to evaluate the speed of the sensor, and, in particular, to reach a 10 000 frames/s rate; 2) to demonstrate a versatile and programmable processing unit at pixel-level; and 3) to provide an original platform dedicated to embedded image processing.

The rest of the paper is organized as follows. Section II is dedicated to the description of the operational principle at pixel-level in the sensor. The main characteristics of the sensor architecture are described in Section III. Section IV talks about the design of the circuit. The details of the photodiode structure, the embedded analog memories, and the arithmetic unit are successively described. Finally, some experimental results of high-speed image acquisition with pixel-level processing are presented in Section V.

II. EMBEDDED ALGORITHMS AT PIXEL LEVEL

Low-level image processing consists of simple operations executed on a very large data set, such as the whole set of pixel values or a region of interest of the whole image. Embedding low-level tasks at focal plane is quite interesting for several aspects. First, the key features are the capability to operate in accordance with the principles of single instruction multiple data (SIMD) computing architectures [13]. This enables massively

Manuscript received May 8, 2007; revised October 2007.

The authors are with the LE2I Laboratory, Burgundy University, 21078 Dijon, France (e-mail: dginjac@u-bourgogne.fr).

Digital Object Identifier 10.1109/JSSC.2007.916618

parallel computations allowing high framerates up to thousands of images per second, with a rather low power consumption. Moreover, the parallel evaluation of the pixels by the SIMD operators leads to processing times which are not dependent of the resolution of the sensor. In a classical system, in which low-level filters are externally implemented after digitization, processing times are proportional to the resolution leading to lower framerates as resolution increases. Secondly, having hardware processing operators, along with the sensor's array, enables to remove the classical input output bottleneck between the sensor and the external processors in charge of processing the pixel values. Indeed, the bandwidth of the communication between the sensor and the external processors is known as a crucial aspect, especially with high-resolution sensors. In such cases, the sensor output data rate can be very high, and needs a lot of hardware resources to convert, process and transmit a lot of information. So, integrating image processing at the sensor level can solve this problem because the pixel values are pre-processed on-chip by the SIMD operators before sending them to the external world via the communication channel. This will result in data reduction, which allows sending the data at lower data-rates, and reduces the effect of the computational-load bottleneck.

Thirdly, one of the main drawbacks to design specific circuits integrating sensing and processing on the same chip is that the image processing operators are often designed for a specific application and not reusable in another context. On the other side, digital processors are characterized by an important versatility and their easy programming. However, in our approach, a new analog processing architecture has been designed. It highlights a compromise between versatility, parallelism, processing speeds and resolution. The analog processing operators are fully programmable devices by dynamic reconfiguration. They can be viewed as a software-programmable image processor dedicated to low-level image processing.

From a traditional point of view, a CMOS smart sensor can be seen as an array of independent pixels, each including a photodetector (PD) and a processing element (PE) built upon a few transistors. Existing works on analog pixel-level image processing can be classified into two main categories. The first one is intrapixel, in which processing is performed on the individual pixels in order to improve image quality, such as the classical Active Pixel Sensor or APS [8], [28] as shown in Fig. 1(a). The second category is interpixel, where the processing is dedicated to groups of pixels in order to perform some early vision processing and not merely to capture images. The transistors, which are placed around the photodetector, can be seen as a real on-chip analog signal processor which improves the functionality of the sensor. This typically allows local and/or global pixel calculations. Our work takes place in this second category because our main objective is the implementation of various *in situ* image processing using local neighborhoods (such as spatial gradients, and Sobel and Laplacian filters). Based on this design concept, this forces a re-thinking of the spatial distribution of the processing resources, so that each computational unit can easily use a programmable neighborhood of pixels. Consequently, in our design each processing element takes place in the middle of four adjacent pixels, as shown in the Fig. 1(b). The

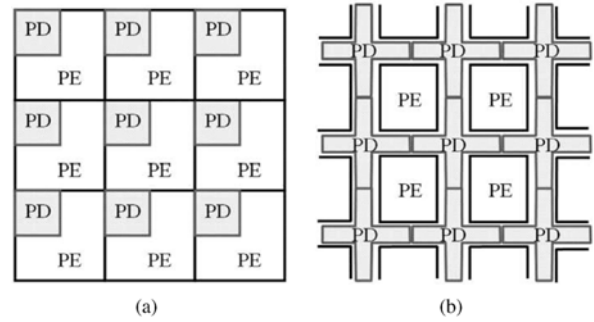


Fig. 1. Photosites with (a) intra-pixel and (b) inter-pixel processing.

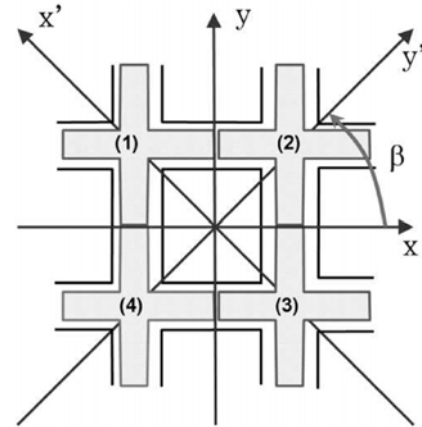


Fig. 2. Evaluation of spatial gradients.

key to this distribution of the pixel-level processors is to realize both compactness of the metal interconnections with pixels and generality of high-speed processing based on neighborhood of pixels.

A. Spatial Gradients

The structure of our processing unit is tailor-made for the computation of spatial gradients based on a 4-neighborhood pixel algorithm, as depicted in Fig. 2.

The main idea for evaluating the spatial gradients [29] is based on the definition of the first-order derivative of a 2-D function performed in the vector direction ξ , which can be expressed as

$$\frac{\partial V(x, y)}{\partial \xi} = \frac{\partial V(x, y)}{\partial x'} \cos(\beta) + \frac{\partial V(x, y)}{\partial y'} \sin(\beta) \quad (1)$$

where β is the vector's angle of outline.

A discretization of (1) at the pixel-level, according to Fig. 2, would give

$$\frac{\partial V}{\partial \xi} = (V_2 - V_4) \cos(\beta) + (V_1 - V_3) \sin(\beta) \quad (2)$$

where $V_i, i \in \{1; 4\}$ is the luminance at pixel i , i.e., the photodiode output. In this way, the local derivative in the direction of vector ξ is continuously computed as a linear combination of two basis functions, the derivatives in the x' and y' directions.

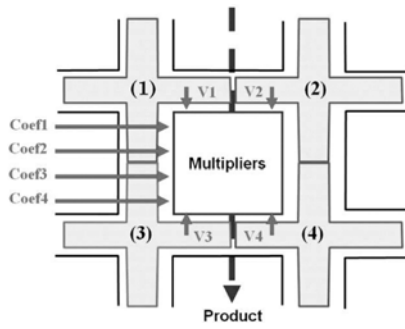
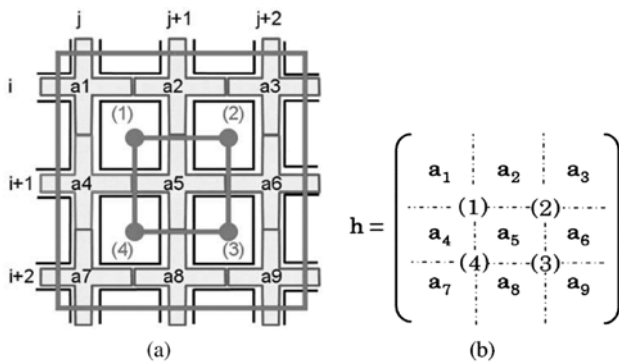


Fig. 3. Implementation of multipliers at pixel-level.

Fig. 4. (a) Array architecture. (b) 3×3 mask used by the four processing elements.

Using a four-quadrant multiplier [30], [31] (see Section IV-C for details of design and implementation), the product of the derivatives by a cosine function can easily be computed. The output product P , as shown in Fig. 3, is given by

$$P = V_1 \sin(\beta) + V_2 \cos(\beta) - V_3 \sin(\beta) - V_4 \cos(\beta). \quad (3)$$

Consequently, the processing element implemented at the pixel-level carries out a linear combination of the four adjacent pixels by the four associated weights ($\text{coef}_i, i \in \{1; 4\}$). In order to evaluate (3), the following values have to be given to the coefficients:

$$\begin{pmatrix} \text{coef1} & \text{coef2} \\ \text{coef3} & \text{coef4} \end{pmatrix} = \begin{pmatrix} \sin(\beta) & \cos(\beta) \\ -\sin(\beta) & -\cos(\beta) \end{pmatrix}. \quad (4)$$

From such a viewpoint, horizontal and vertical gradients can be straightforwardly evaluated by respectively fixing the value of β as 0° and 90° .

B. Sobel Operator

The structure of our architecture is also well-adapted to various algorithms based on convolutions using binary masks on a neighborhood of pixels. For example, the evaluation of the Sobel algorithm with our chip leads to the result directly centered on the photosensor and directed along the natural axes of the image according to Fig. 4(a). In order to compute the mathematical operation, a 3×3 neighborhood is applied on the whole image, as described in Fig. 4(b).

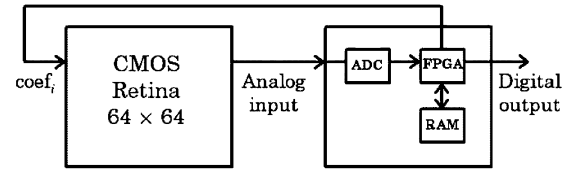


Fig. 5. Schematic imager system bloc.

In order to carry out the discretized derivatives in two dimensions (along the horizontal and vertical axes) it is necessary to build two 3×3 matrices called h_1 and h_2 :

$$h_1 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \\ h_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}. \quad (5)$$

Within the four processing elements numbered from 1 to 4, as shown in Fig. 4(a), four 2×2 masks act locally on the image. According to (5), this allows the evaluation of the following series of operations:

$$h_1 : \begin{aligned} V_{11} &= -(V_1 + V_4) \\ V_{12} &= +(V_3 + V_6) \\ V_{13} &= +(V_6 + V_9) \\ V_{14} &= -(V_4 + V_7) \end{aligned} \\ h_2 : \begin{aligned} V_{21} &= -(V_1 + V_2) \\ V_{22} &= -(V_2 + V_3) \\ V_{23} &= +(V_8 + V_9) \\ V_{24} &= +(V_7 + V_8) \end{aligned} \quad (6)$$

with the values V_{1k} and V_{2k} provided by the processing element (k). Then, from these trivial operations, the discrete amplitudes of the derivatives along the vertical axis ($V_{h1} = V_{11} + V_{12} + V_{13} + V_{14}$) and the horizontal axis ($V_{h2} = V_{21} + V_{22} + V_{23} + V_{24}$) can be computed. The evaluation of the horizontal and vertical gradients spends four retina cycles, two for each gradient.¹

In the first frame, in order to evaluate $V_{11} + V_{14}$, the following values have to be given to the coefficients:

$$\begin{pmatrix} \text{coef1} & \text{coef2} \\ \text{coef3} & \text{coef4} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ -1 & 0 \end{pmatrix}. \quad (7)$$

Then, in the second frame $V_{12} + V_{13}$ is evaluated by using the following coefficient values:

$$\begin{pmatrix} \text{coef1} & \text{coef2} \\ \text{coef3} & \text{coef4} \end{pmatrix} = \begin{pmatrix} 0 & +1 \\ 0 & +1 \end{pmatrix}. \quad (8)$$

So, the Analog Arithmetic Units (A^2U) implementing these computations at pixel-level (see Section IV-B for details) drastically decrease the number of calculation carried out by the external processor (FPGA) as shown in Fig. 5. Indeed, in the

¹A retina cycle is defined as the time spent between two successive acquisition frames including thus acquisition and preprocessing of the image.

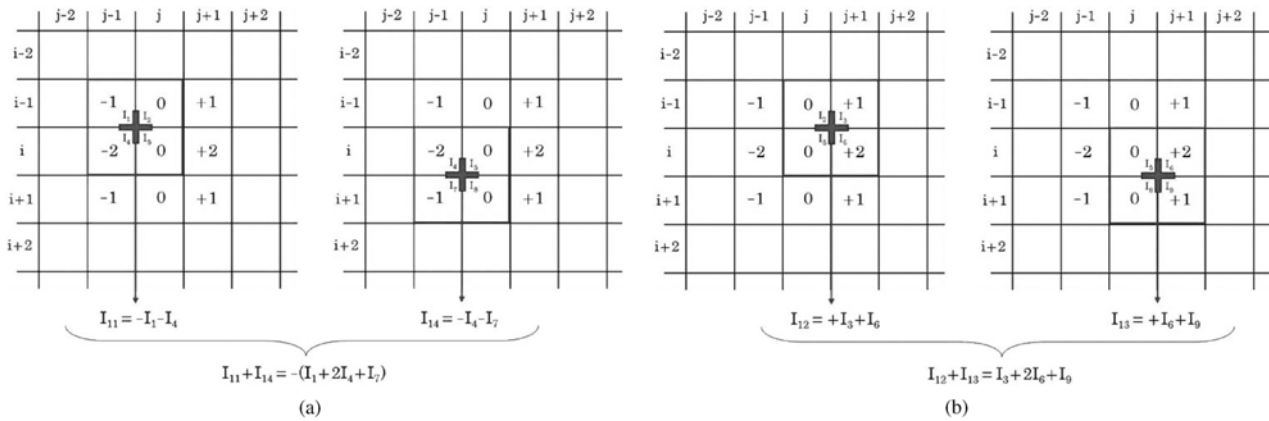


Fig. 6. Dynamic reconfiguration sequence for vertical Sobel filter.

case of our experimental 64×64 pixel sensor, the peak performance is equivalent to 4 parallel signed multiplications by pixel at 10 000 frames/s, i.e., more than 160 million multiplications per second. With a VGA resolution (640×480), the performance level would increase to a factor of 75, leading to about 12 billion multiplication per second. Processing this data flow by external processors will imply important hardware resources in order to cope with the temporal constraints.

Moreover, with our chip, the assignment of coefficient values from the external processor towards the retina, gives the system some interesting dynamic properties. The system can be easily reconfigured by changing the internal coefficients for the masks between two successive frames. First, this allows the possibility to dynamically change the image processing algorithms embedded in the sensor. Second, this enables the evaluation of some complex pixel-level algorithms, implying different successive convolution masks. For example, as depicted in Fig. 6, the coefficient values are reconfigured twice in order to evaluate the vertical Sobel filter. During the first frame, V_{11} and V_{14} are evaluated whereas the second frame allows the computation of V_{12} and V_{13} . The FPGA is only used for the final addition of the four values.

C. Second-Order Detector: Laplacian

Edge detection based on some second-order derivatives such as the Laplacian can also be implemented on our architecture. Unlike previously described spatial gradients, the Laplacian is a scalar [see (9)] and does not provide any indication about edge direction:

$$\Delta = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \quad (9)$$

From this 3×3 mask, the following operations can be extracted according to the principles previously used for the evaluation of the Sobel operator:

$$\begin{aligned} \Delta : V_{11} &= V_4 - V_5 \\ V_{12} &= V_2 - V_5 \\ V_{13} &= V_6 - V_5 \\ V_{14} &= V_8 - V_5. \end{aligned} \quad (10)$$

The discrete amplitudes of the second-order derivative is given by $V_{\Delta} = V_{11} + V_{12} + V_{13} + V_{14}$. These operations can be carried out in four retina cycles.

D. General Spatial Filter and Strategies

In the preceding sections, we focused on 2×2 and 3×3 convolution masks. In the case of a 2×2 mask, the coefficients are fixed once before the beginning of the acquisition frame. In the case of a 3×3 mask, two possibilities can occur. First, the 3×3 mask presents some symmetrical properties (such as the Sobel or Laplacian masks) and then the coefficients values can be fixed as in a 2×2 mask. Second, if the mask is not symmetric, it is necessary to dynamically reconfigure the coefficients during the acquisition frame. For masks which size is greater than 3×3 and more generally in the case of an $N \times N$ mask, a dynamic reconfiguration of coefficients is necessary during the acquisition frame in order to evaluate the successive values of the linear combinations of pixels.

III. OVERVIEW OF THE CHIP ARCHITECTURE

As in a traditional image sensor, the core of the chip presented in this paper is constructed of a 2-D pixel array, here of 64 columns and 64 rows with random pixel ability, and some peripheral circuits. It contains about 160 000 transistors on a $3.675 \text{ mm} \times 3.775 \text{ mm}$ die. The full layout of the retina is depicted in Fig. 7 and the main chip characteristics are listed in Table I.

Each individual pixel contains a photodiode for the light-to-voltage transduction and 38 transistors integrating all the analog circuitry dedicated to the image processing algorithms. This amount of electronics includes a preloading circuit, two Analog Memory, Amplifier and Multiplexer structures ([AM]²) and an Analog Arithmetic Unit (A²U) based on a four-quadrant multiplier architecture. The full pixel size is $35 \mu\text{m} \times 35 \mu\text{m}$ with a 25% fill factor.

Fig. 8 shows a block diagram of the proposed chip. The architecture of the chip is divided into three main blocks as in many circuits widely described in the literature. First, the array of pixels (including photodiodes with their associated circuitry for performing the analog computation) is placed at the center. Second, placed below the chip core are the readout circuits with the three asynchronous output buses: the first one is dedicated

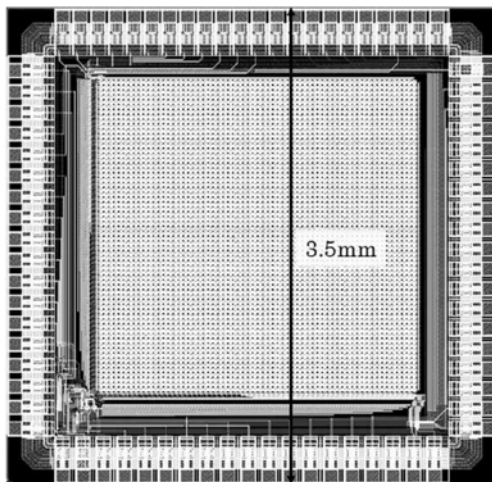


Fig. 7. Layout of the full retina.

TABLE I
CHIP CHARACTERISTICS

Technology	0.35 μm 2-poly 4-metal CMOS
Array size	64 \times 64
Chip size	11 mm ²
Number of transistors	160 000
Number of transistors / pixel	38
Pixel size	35 μm \times 35 μm
Sensor Fill Factor	25 %
Dynamic power consumption	110 mW
Supply voltage	3.3 V
Frame rate	10 000 fps

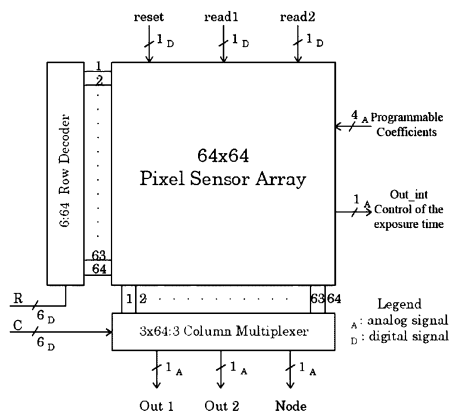


Fig. 8. Block diagram of the chip.

to the image processing results whereas the other two provide parallel outputs for full high rate acquisition of raw images. Finally, the left part of the sensor is dedicated to a row decoder for addressing the successive rows of pixels. The pixel values are selected one row at a time and read out to vertical column buses connected to an output multiplexer.

The chip also contains test structures used for detailed characterization of the photodiodes and processing units. These test structures can be seen on the bottom left of the chip.

The operation of the imaging system can be divided into four phases: reset, integration, image processing, and readout. The reset, integration, and pixel-level processing phases all occur in parallel over the full array of pixels (*snapshot* mode) in order to avoid any distortion due to a row-by-row reset. The control of the integration time can be supervised with the global output signal called *Out_int*. This signal provides the average incidental illumination of the whole matrix of pixels. Indeed, the currents issued from all the pixels of the matrix are summed to produce *Out_int*. So, this signal is directly linked to the average level of the image. A low value of *Out_int* implies a dark image, whereas an important value traduces a bright image. Following the values of *Out_int*, the integration time can be adapted in order to obtain the most appropriate images: if the average level of the image is too low, the exposure time may be increased. On the contrary, if the scene is too luminous, the integration period may be reduced.

IV. DESIGN OF THE CIRCUIT

A. Photodiode Structure

As previously described in Section II, each pixel of our chip includes a photodiode and a processing unit dedicated to low-level image processing based on neighborhoods. One of our main objectives focuses on the optimization of the pixel-level processor mapping in order to facilitate the access to the values of adjacent pixels. Therefore, an original structure [as previously depicted in Fig. 1(b)] was chosen. The major advantage of this structure is the minimization of the length of metal interconnection between adjacent pixels and the processing units, contributing to: 1) a better fill factor and 2) a higher framerate.

In order to achieve high-speed performance, one of the key elements is the photodiodes which should be designed and optimized carefully. Critical parameters in the design of photodiodes are the dark current and the spectral response [32]. The shape of photodiode layout, the structure of the photodiode, and the layout have significant influences on the performance of the whole imager [33], [34].

In our chip, photodiodes consist of N-type photodiodes based on an n^+ -type diffusion in a p-type silicon substrate. The depletion region is formed in the neighborhood of the photodiode cathode. Optically generated photocarriers diffuse to neighboring junctions [35]. We have analyzed and tested three photodiodes shapes: the square photodiode classically used in literature, the cross shape which is perfectly adapted to the optimized pixel-level processors mapping, and finally the octagonal shape based on 45° structures.

Fig. 9 illustrates these different photodiodes structures. For each of these shapes, the active area (displayed in gray dots) and the inter-element isolation area with external connections (filled in gray) are represented. The active area absorbs the illumination energy and turns that energy into charge carriers. This active area must be large as possible in order to absorb a maximum of photons whereas the inter-element isolation area must be as small as possible in order to obtain the best fill factor (i.e., the ratio between the active area and the total pixel area). In the follow-up to this paper, we use the term *Active layer surfaces (Als)* when talking about the active area of the photodiode and

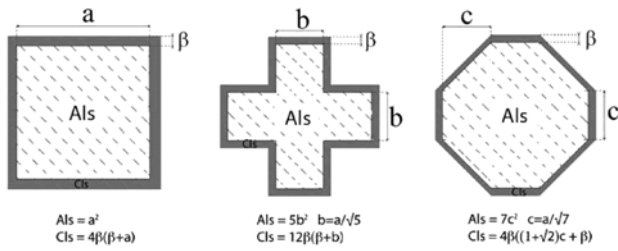


Fig. 9. Photodiode structures. (a) Square shape. (b) Cross shape. (c) Octagonal shape.

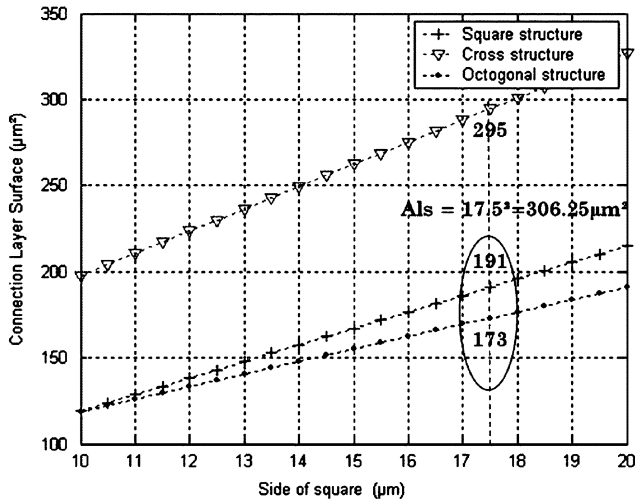


Fig. 10. Cl_s for the three different shapes expressed as a function of the side a of the square photodiode.

the term *Connection layers surfaces* (Cl_s) for the connections of the photodiodes. Based on the geometrical parameters a , b , c , and β , we can easily define the Cl_s and Al_s mathematical expressions (as depicted in the Fig. 10). Furthermore, according to the design rules of the AMS-CMOS 0.35 μm process, the minimal value of β was evaluated to 2.35 μm . Starting from this result, we can plot comparative graphs of Cl_s for the three photodiodes shapes, as shown in Fig. 10.

In our design, we have fixed the fill factor to 25% with a total pixel size of 35 $\mu\text{m} \times 35 \mu\text{m}$. So, the values of Al_s and a can be easily inferred: $Al_s = 306.25 \mu\text{m}^2$ and $a = 17.5 \mu\text{m}$. From Fig. 10, we can see 1) that the cross shape appears to be unrealistic because of the large value of Cl_s ($Cl_s = 295 \mu\text{m}^2$) and 2) that the square and the octagonal shapes have similar values (respectively, 191 μm^2 and 173 μm^2). Finally, the octagonal shape was chosen because the surface dedicated to the interconnections is about 12% lower compared to a square shape, allowing a best integration of the photodiodes. This also implies a better spectral response compared to the square photodiode as shown in Fig. 11. A detailed characterization of spectral responses of the different photodiodes has been performed by using a light generator with wavelength of 400 nm to 1100 nm. The experimental data reveal that the octagonal structure has better performance than the square shape for all wavelengths. Our results are complementary and similar to those obtained by [33] in their study of dark current.

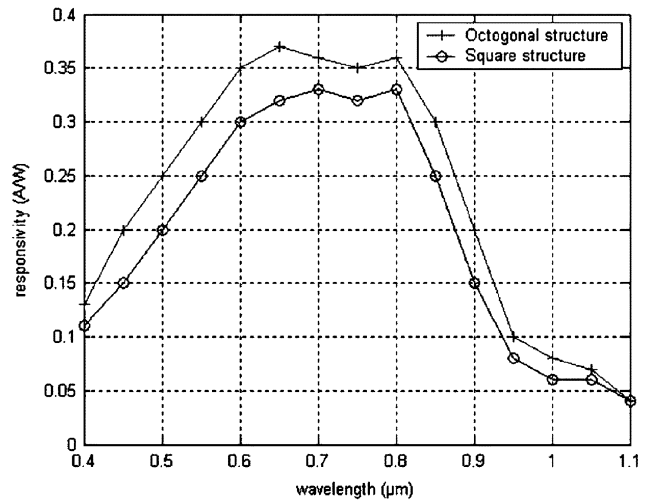


Fig. 11. Spectral responses in the photodiode structures of type square, and type octagonal.

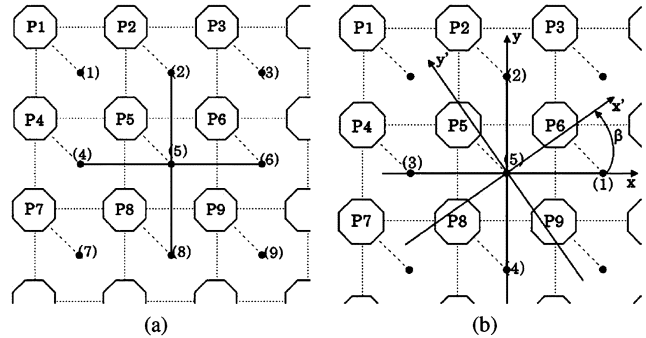


Fig. 12. (a) Array of pixel based on octagonal photodiodes. (b) Evaluation of spatial gradients.

From the above measurement results, the octagonal type photodiode structure was chosen as the photodetector for our chip. Fig. 12 illustrates the arrangement of pixels and the computation of spatial gradients in this configuration, as previously described in this paper.

B. Pixel-Level $[AM]^2$

In order to increase the algorithmic possibilities of the architecture, the key point is the separation of the acquisition of the light inside the photodiode and the readout of the stored value at pixel-level [36]. Thus, the storage element should keep the output voltage of the previous frames whereas the sensors integrate photocurrent for a new frame. So, for each pixel of our chip we have designed and implemented two specific circuits, including an analog memory, an amplifier, and a multiplexor as shown in Fig. 14.

With these $[AM]^2$ circuits, the capture sequence can be made in the first memory in parallel with a readout sequence and/or processing sequence of the previous image stored in the second memory, as shown in Fig. 13.

Such a strategy has several advantages:

- 1) The framerate can be increased (up to 2x) without reducing the exposure time.

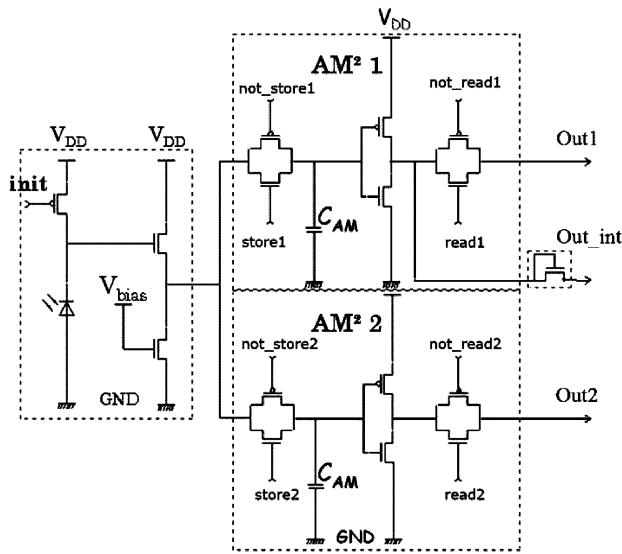
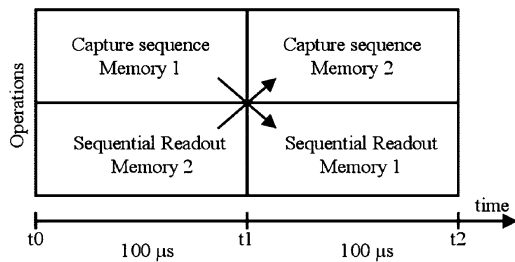
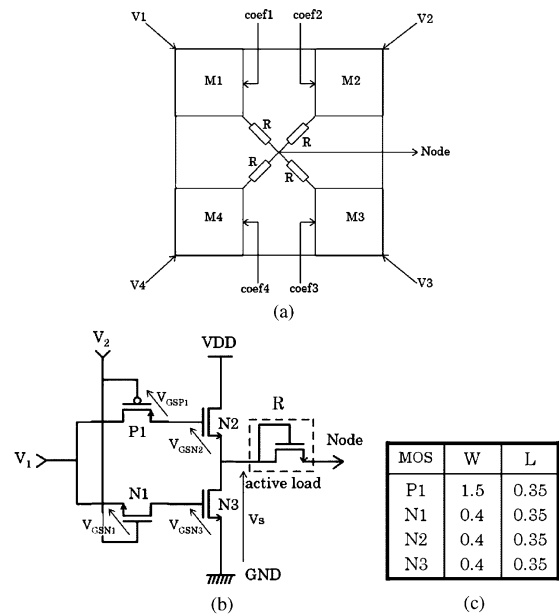

 Fig. 14. Schematic of the $[AM]^2$ structure.


Fig. 13. Parallelism between capture sequence and readout sequence.

- 2) The image acquisition is decorrelated from image processing, implying that the architecture performance is always the highest, and the processing framerate is maximum.
- 3) A new image is always available without spending any integration time.

The chip operates with a single 3.3 V power supply. In each pixel, as shown in Fig. 14, the photodiode is a nMOS photodiode associated with a pMOS transistor reset, which represents the first stage of the acquisition circuit. The pixel array is held in a reset state until the *init* signal goes high. Then, the photodiode discharges according to incidental luminous flow. This signal is polarized around $V_{DD}/2$ (i.e., half the power supply voltage). Behind this first stage of acquisition, two identical subcircuits take place. One of these subcircuits is selected when either the *store1* signal or the *store2* signal is turned on. Then, the associated analog switch is open allowing the capacitor to integrate the pixel value. Consequently, the C_{AM} capacitors are able to store the pixel values during the frame capture either from switch 1 or switch 2. The capacitors are implemented with double-polysilicium. The size of the capacitors is as large as possible in order to respect the fill factor and the pixel size requirements. The capacitors values are about 40 fF and are able to store the pixel values for 20 ms with an error lower than 4%. Each of the capacitors is followed by an inverter, polarized on $V_{DD}/2$. This


 Fig. 15. The A^2U structure.

inverter serves as an amplifier of the stored value. It provides a value which is proportional to the pixel incidental illumination. Finally, the readout of the stored values are activated by a last switch controlled by the *read1* and *read2* signals.

C. Pixel-Level Analog Arithmetic Unit: A^2U

The analog arithmetic unit (A^2U) represents the central part of the pixel and includes four multipliers (M1, M2, M3, and M4), as illustrated in Fig. 15. The four multipliers are all interconnected with a diode-connected load (i.e., an nMOS transistor with gate connected to drain). The operation result at the *node* point is a linear combination of the four adjacent pixels.

Assuming that MOS transistors operate in subthreshold region, the output node V_S of a multiplier can be expressed as a function of the two inputs V_1 and V_2 as follows:

$$k_r(V_{THN} - V_1)(V_1 - V_2 - V_{THN}) = (V_1 - V_S - V_{THN})(V_2 - V_S - V_{THN} - V_{THP}) \quad (11)$$

where k_r represents the transconductance ratio, V_{THN} and V_{THP} are the threshold voltage for the nMOS and pMOS transistors. Around the operating point ($V_{DD}/2$), the variations of the output node mainly depend on the product V_1V_2 . So, (11) can be simplified and finally, the output node V_S can be expressed as a simple first-order of the two input voltages V_1 and V_2 :

$$V_S = MV_1V_2 \quad \text{with } M = \frac{k_r - 1}{2V_{THN} + V_{THP}} \approx 8.07/V. \quad (12)$$

The important value of the coefficient M gives to the structure a good robustness by limiting the impact of the second-order intermodulation products. The first consequence is a better linearity of our multiplier design integrating only five transistors.

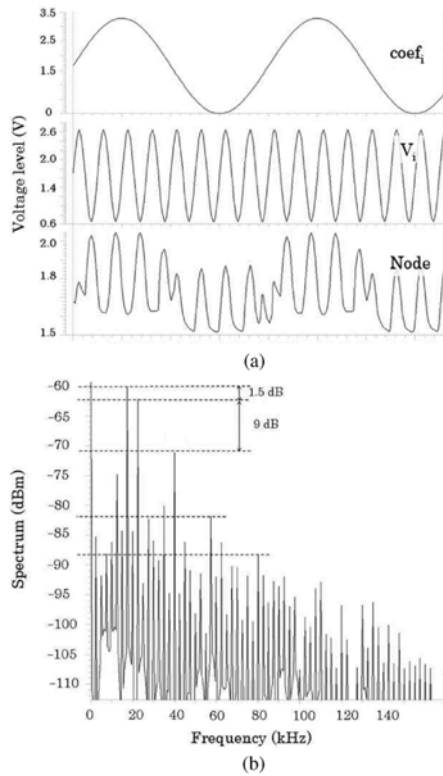


Fig. 16. Benchmark of the four-quadrant multiplier.

Fig. 16 shows the experimental results of this multiplier structure with cosine signals as inputs:

$$\text{coef}_i = A \cos(2\pi f_1) \text{ with } f_1 = 2.5 \text{ kHz} \quad (13)$$

$$V_i = B \cos(2\pi f_2) \text{ with } f_2 = 20 \text{ kHz}. \quad (14)$$

In an ideal case, the output node value can be written as follows:

$$\text{Node} = \frac{AB}{2} [\cos(2\pi(f_2 - f_1)) + \cos(2\pi(f_2 + f_1))]. \quad (15)$$

The signal's spectrum, represented in Fig. 16(b), contains two main frequencies (17.5 kHz and 22.5 kHz) around the carrier frequency. The residues which appear in the spectrum are known as inter-modulations products. They are mainly due to the nonlinearity of the structure (around 10 kHz and 30 kHz) and the insulation defects of input pads (at 40 kHz). However, the amplitude of these inter-modulation products is significantly lower than the two main frequencies. Indeed, the spectral line level at 40 kHz is 9 dB under the level of the main frequencies. Therefore, the contribution of the insulation defect is eight times smaller than the main signals. Furthermore, experimental measures on the chip revealed that the best linearity of the multiplier is obtained for amplitudes of the signal V_i in the range of 0.6–2.6 V. In the chip, the signal V_i corresponds to the voltage coming from the pixel. The pixel values can be included in this range by means of by the biasing voltage V_{bias} of the $[AM]^2$ structure.

TABLE II
CHIP MEASUREMENTS

Conversion gain	54 $\mu\text{V}/e^-$ RMS
Sensitivity	0.15 V/lux.s
Fixed Pattern Noise retina (FPN), dark	225 μV RMS
Thermal reset noise	68 μV RMS
Output levels disparities	4.3%
Voltage gain of the amplifier stage	12
Linear flux	98.5%
Dynamic range	68 dB

V. EXPERIMENTAL RESULTS

An experimental 64×64 pixel image sensor has been developed in a $0.35 \mu\text{m}$, 3.3 V, standard CMOS process with poly-poly capacitors. This prototype was sent to foundry at the beginning of 2006 and was available at the end of the third quarter of the year. Its functional testing and its characterization were performed using a specific hardware platform. The hardware part of the imaging system contains a one million Gates Spartan-3 FPGA board with 32 MB SDRAM embedded. This FPGA board is the XSA-3S1000 from XESS Corporation. An interface acquisition circuit includes three ADC from Analog Device (AD9048), high-speed LM6171 amplifiers and others elements such as the motor lens. Fig. 17 shows the schematic and some pictures of the experimental platform.

A. Characterization

The sensor was quantitatively tested for conversion gain, sensitivity, fixed pattern noise, thermal reset noise, output levels disparities, voltage gain of the amplifier stage, linear flux, and dynamic range. Table II summarizes these imaging sensor characterization results. To determine these values, the sensor included specific test pixels in which some internal node voltages can be directly read. The test equipment hardware is based on a light generator with wavelength of 400 nm to 1100 nm. The sensor conversion gain was evaluated to $54 \mu\text{V}/e^-$ RMS with a sensitivity of 0.15 V/lux.s, thanks to the octagonal shape of the photodiode and the fill factor of 25%. At 10 000 frames/s, measured nonlinearity is 0.12% over a 2 V range. These performances are similar to the sensor described in [25]. According to the experimental results, the voltage gain of the amplifier stage of the two $[AM]^2$ is $A_v = 12$ and the disparities on the output levels are about 4.3 %.

Image sensors always suffer from technology related non-idealities that can limit the performances of the vision system. Among them, fixed pattern noise (FPN) is the variation in output pixel values, under uniform illumination, due to device and interconnect mismatches across the image sensor. Two main types of FPN occur in CMOS sensors. First, offset FPN which takes place into the pixel is due to fluctuations in the threshold voltage of the transistors. Second, the most important source of FPN is introduced by the column amplifiers used in standard APS systems. In our approach, the layout is symmetrically built in order to reduce the offset FPN among each block of four pixels and

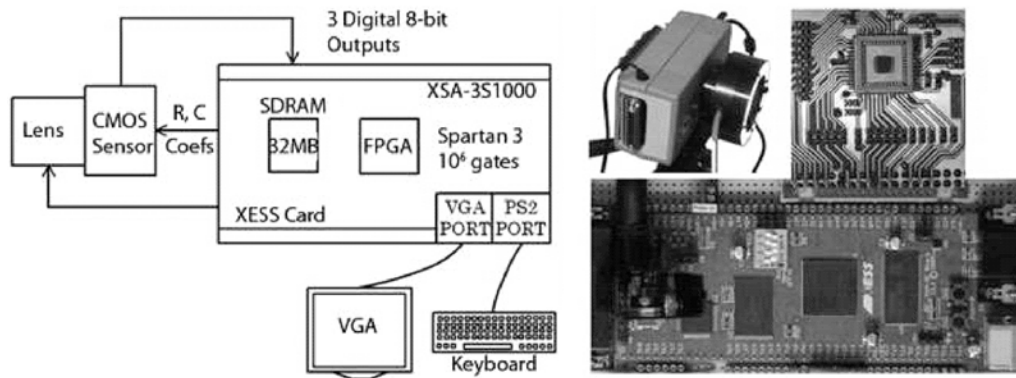


Fig. 17. Block diagram and pictures of the hardware platform including FPGA board and CMOS sensor.

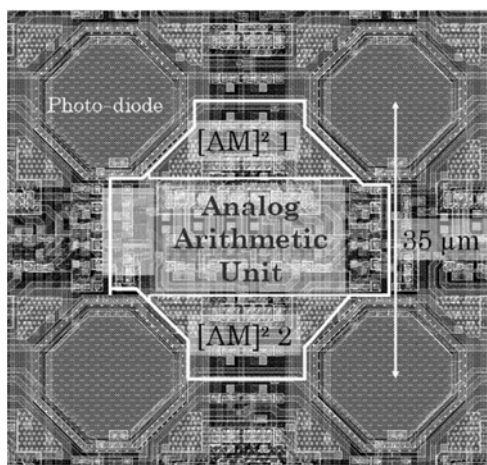


Fig. 18. Layout of four pixels.

to ensure uniform spatial sampling, as depicted in the layout of a 2×2 pixel block in Fig. 18.

Furthermore, our chip does not include any column amplifier since the amplification of the pixel values takes place into the pixel by means of an inverter. So, the gain FPN is very limited and only depends on the mismatch of the two transistors. FPN can be reduced by correlated double sampling (CDS). To implement CDS, each pixel output needs to be read twice, once after reset and a second time at the end of integration. The correct pixel signal is obtained by subtracting the two values. A CDS can be easily implemented in our chip. For this purpose, the first analog memory stores the pixel value just after the reset signal and the second memory stores the value at the end of integration. Then, at the end of the image acquisition, the two values can be transferred to the FPGA, responsible for producing the difference. In Fig. 19, the two images show fixed pattern noise with and without CDS using a 1 ms integration time. On the left image, the FPN is mainly due to the random variations in the offset voltages of the pixel-level analog structures. The experimental benchmarks of our chip reveal a FPN value of $225 \mu\text{V}$ RMS. The right picture shows the same image after analog CDS, performed as described above. The final FPN has been reduced by a factor of 34 to $6.6 \mu\text{V}$. In the rest of the results, CDS has

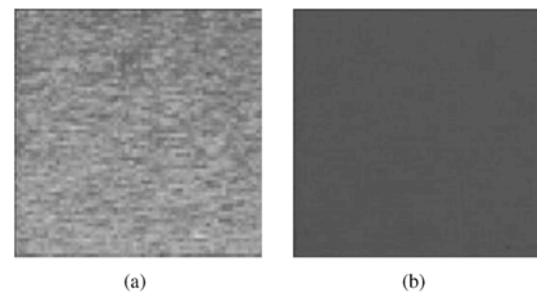


Fig. 19. Images of fixed pattern noise (a) without CDS and (b) with CDS for an integration time of 1 ms.

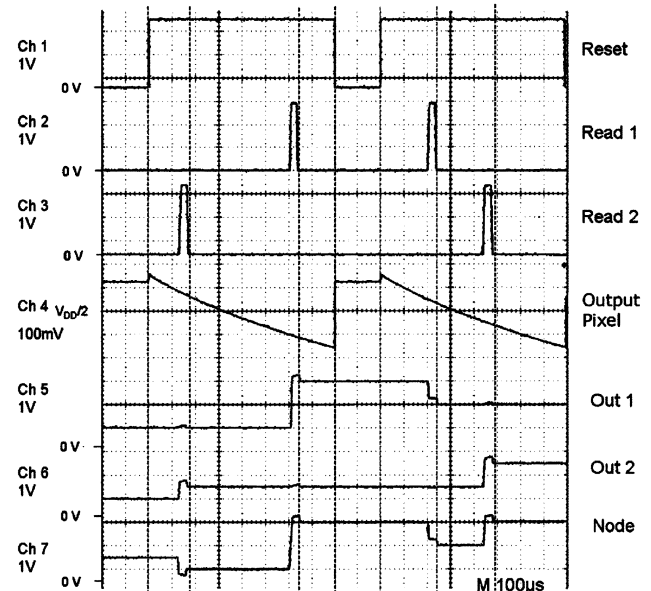


Fig. 20. High-speed sequence capture with basic image processing.

not been implemented since FPN has low values. Only, an entire dark image is subtracted from the output images on the FPGA. Focus has been made on the development of low-level image processing using the two analog memories and the associated processing unit.

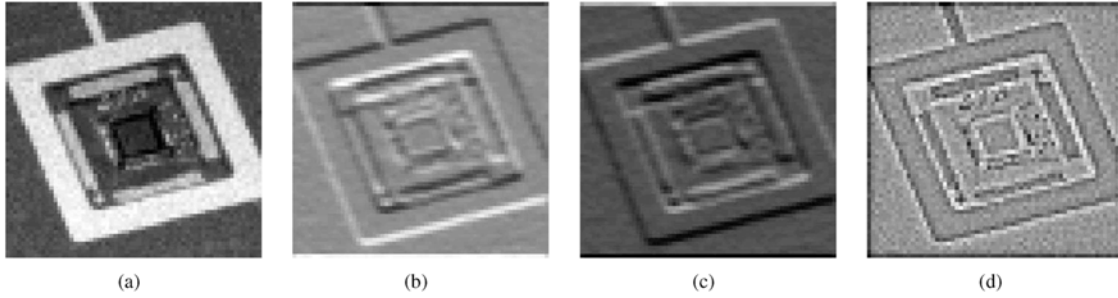


Fig. 21. (a) Raw image at 10 000 fps. (b) Output Sobel horizontal image. (c) Output Sobel vertical image. (d) Output Laplacian image.

B. Sample Images

Fig. 20 describes the experimental results of successive acquisitions and signal processing in an individual pixel. Each acquisition occurs when one of the two signals *read 1* or *read 2* goes high. For each of these acquisitions, various levels of illumination are applied. The two outputs (*out 1* and *out 2* give a voltage corresponding to the incidental illumination on the pixels. The calibration of the structure is ensured by the biasing ($V_{bias} = 1.35$ V). Moreover, in this characterization, the output *node* computes the difference between *out 1* and *out 2*. For this purpose, the coefficients are fixed at the following values: $coef1 = -coef2 = V_{DD}$ and $coef3 = coef4 = V_{DD}/2$.

Fig. 21 shows experimental image results. Fig. 21(a) shows an image acquired at 10 000 frames/s (integration time of 100 μ s). Except for amplification of the photodiodes signal, no other processing is performed on this raw image. Fig. 21(b)–(d) shows different images with pixel-level image processing at a frame rate of about 2500 frames/s. From left to right, horizontal and vertical Sobel filter and Laplacian operator images are displayed. Some of these image processing algorithms imply a dynamic reconfiguration of the coefficients. We can note that there is no energy spent for transferring information from one level of processing to another because only a frame acquisition is needed before the image processing take place.

In order to estimate the quality of our embedded image processing approach, we have compared results of horizontal and vertical Sobel and Laplacian operators obtained with our chip and with digital operators implemented on a computer. In each case, the image processing is applied on real images obtained by our chip. For the comparison of the results, we have evaluated the likelihood between the resulting images by using the cross correlation coefficient. The correlation coefficient is given by

$$r = \frac{\sum_{j=1}^N \sum_{i=1}^N (R(i,j) - \bar{R})(P(i,j) - \bar{P})}{\sqrt{\sum_{j=1}^N \sum_{i=1}^N (R(i,j) - \bar{R})^2} \sqrt{\sum_{j=1}^N \sum_{i=1}^N (P(i,j) - \bar{P})^2}} \quad (16)$$

where R is the resulting image obtained with the analog arithmetic units on the retina, and P is the resulting image obtained with an external processor. \bar{P} and \bar{R} are respectively the average matrices P and R . N is the array size ($N = 64$). Table III summarizes the results of the cross correlation coefficient obtained with horizontal and vertical Sobel filters and Laplacian operators.

TABLE III
IMAGE CORRELATION COEFFICIENT

Operator	Correlation coefficient (r)
Vertical Sobel	0.928
Horizontal Sobel	0.930
Laplacian	0.939

The cross correlation coefficient can be viewed as a good indicator of the linearity of the pixel-level analog arithmetic units. In our case, this coefficient is 93.2% on average. The likelihood, specifically for the Laplacian operator, is greater because of the perfect symmetry of this operator. Overall, the analog arithmetic unit has good performance compared to external operators implemented on a computer.

VI. CONCLUSION AND PERSPECTIVES

An experimental pixel sensor implemented in a standard digital CMOS 0.35 μ m process was described. Each 35 μ m \times 35 μ m pixel contains 38 transistors implementing a circuit with photocurrent integration, two [AM]², and an A²U.

Experimental chip results reveal that raw image acquisition at 10 000 frames per second can be easily achieved using the parallel A²U implemented at pixel-level. With basic image processing, the maximal frame rate slows to about 5000 fps.

The next step in our research will be the design of a similar circuit in a modern 130 nm CMOS technology. The main objective will be to design a pixel of less than 10 μ m \times 10 μ m with a fill factor of 20%. Thus, with the increasing scaling of the transistors in a such technology, we could consider the implementation of more sophisticated image processing operators dedicated to face localization and recognition. Previous works of our team [37] have demonstrated the needs of dedicated CMOS sensors embedding low-level image processing such as features extraction. Moreover, actual works [38] focus on a recent face detector called the Convolutional Face Finder (CFF) [39], which is based on a multi-layer convolutional neural architecture. The CFF consists of six successive neural layers. The first four layers extract characteristic features, and the last two perform the classification. Our objective would be to implement at pixel-level the first layers based on convolutions by different masks from 2 \times 2 to 5 \times 5.

In order to evaluate this future chip in some realistic conditions, we would like to design a CIF sensor (352 \times 288 pixels), which leads to a 3.2 mm \times 2.4 mm in a 130 nm technology. In

the same time, we will focus on the development of a fast ADC. The integration of this ADC on future chips will allow us to provide new and sophisticated vision systems on chip (ViSOC) dedicated to digital embedded image processing at thousands of frames per second.

REFERENCES

- [1] E. Possum, "Active pixel sensors: Are CCDs dinosaurs?," *Int. Soc. Opt. Eng. (SPIE)*, vol. 1900, pp. 2–14, 1993.
- [2] E. Possum, "CMOS image sensor: Electronic camera on a chip," *IEEE Trans. Electron Devices*, vol. 44, no. 10, pp. 1689–1698, Oct. 1997.
- [3] P. Seitz, "Solid-state image sensing," *Handbook of Computer Vision and Applications*, vol. 1, pp. 165–222, 2000.
- [4] D. Litwiler, "CCD versus CMOS: Facts and fiction," *Photonics Spectra*, pp. 154–158, Jan. 2001.
- [5] M. Loinaz, K. Singh, A. Blanksby, D. Inglis, K. Azadet, and B. Ackland, "A 200 mv 3.3 V CMOS color camera IC producing 352 × 288 24-b video at 30 frames/s," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 2092–2103, Dec. 1998.
- [6] S. Smith, J. Hurwitz, M. Torrie, D. Baxter, A. Holmes, M. Panaghistor, R. Henderson, A. Murray, S. Anderson, and P. Denyer, "A single-chip 306 × 244-pixel CMOS NTSC video camera," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, 1998, pp. 170–171.
- [7] A. El Gamal, D. Yang, and B. Fowler, "Pixel level processing—why, what and how?," in *Proc. SPIE Electronic Imaging '99 Conf.*, Jan. 1999, vol. 3650, pp. 2–13.
- [8] O. Yadid-Pecht and A. Belenky, "In-pixel autoexposure CMOS APS," *IEEE J. Solid-State Circuits*, vol. 38, no. 8, pp. 1425–1428, Aug. 2003.
- [9] P. Acosta-Serafini, M. Ichiro, and C. Sodini, "A 1/3 VGA linear wide dynamic range CMOS image sensor implementing a predictive multiple sampling algorithm with overlapping integration intervals," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1487–1496, Sep. 2004.
- [10] L. Kozlowski, G. Rossi, L. Blanquart, R. Marchesini, Y. Huang, G. Chow, J. Richardson, and D. Standley, "Pixel noise suppression via SoC management of target reset in a 1920 × 1080 CMOS image sensor," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2766–2776, Dec. 2005.
- [11] M. Sakakibara, S. Kawahito, D. Handoko, N. Nakamura, M. Higashi, K. Mabuchi, and H. Sumi, "A high-sensitivity CMOS image sensor with gain-adaptative column amplifiers," *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 1147–1156, May 2005.
- [12] A. Krymsky and T. Niarong, "A 9-V/Lux 5000-frames/s 512 × 512 CMOS sensor," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 136–143, Jan. 2003.
- [13] G. Cembrano, A. Rodríguez-Vazquez, R. Galan, F. Jimenez-Garrido, S. Espejo, and R. Dominguez-Castro, "A 1000 FPS at 128 × 128 vision processor with 8-bit digitized I/O," *IEEE J. Solid-State Circuits*, vol. 39, no. 7, pp. 1044–1055, Jul. 2004.
- [14] L. Lindgren, J. Melander, R. Johansson, and B. Miller, "A multiresolution 100-GOPS 4-Gpixels/s programmable smart vision sensor for multi-sense imaging," *IEEE J. Solid-State Circuits*, vol. 40, no. 6, pp. 1350–1359, Jun. 2005.
- [15] Y. Sugiyama, M. Takumi, H. Toyoda, N. Mukozaka, A. Ithori, T. Kurashina, Y. Nakamura, T. Tonbe, and S. Mizuno, "A high-speed CMOS image with profile data acquiring function," *IEEE J. Solid-State Circuits*, vol. 40, pp. 2816–2823, 2005.
- [16] D. Handoko, K. S. Y. Takokoro, M. Kumahara, and A. Matsuzawa, "A CMOS image sensor for local-plane motion vector estimation," in *Symp. VLSI Circuits Dig. Papers*, Jun. 2000, vol. 3650, pp. 28–29.
- [17] S. Lim and A. El Gamal, "Integrating image capture and processing—beyond single chip digital camera," in *Proc. SPIE Electronic Imaging 2001 Conf.*, San Jose, CA, Jan. 2001, vol. 4306.
- [18] X. Liu and A. El Gamal, "Photocurrent estimation from multiple non-destructive samples in a CMOS image sensor," in *Proc. SPIE Electronic Imaging 2001 Conf.*, San Jose, CA, Jan. 2001, vol. 4306.
- [19] D. Yang, A. El Gamal, B. Fowler, and H. Tian, "A 640 × 512 CMOS image sensor with ultra wide dynamic range floating-point pixel-level ADC," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1821–1834, Dec. 1999.
- [20] O. Yadid-Pecht and E. Possum, "CMOS APS with autoscaling and customized wide dynamic range," in *IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, Jun. 1999, vol. 3650, pp. 48–51.
- [21] D. Stoppa, A. Somoni, L. Gonzo, M. Gottardi, and G.-F. Dalla Betta, "Novel CMOS image sensor with a 132-dB dynamic range," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1846–1852, Dec. 2002.
- [22] X. Liu and A. El Gamal, "Simultaneous image formation and motion blur restoration via multiple capture," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2001, vol. 3, pp. 1841–1844.
- [23] C.-Y. Wu and C.-T. Chiang, "A low-photocurrent CMOS retinal focal-plane sensor with a pseudo-bjt smoothing network and an adaptative current schmitt trigger for scanner applications," *IEEE Sensors J.*, vol. 4, no. 4, pp. 510–518, Aug. 2004.
- [24] D. Yang, B. Fowler, and A. El Gamal, "A Nyquist-rate pixel-level ADC for CMOS image sensors," *IEEE J. Solid-State Circuits*, vol. 34, no. 3, pp. 348–356, Mar. 1999.
- [25] S. Kleinfelder, S. Lim, X. Liu, and A. El Gamal, "A 10 000 frames/s CMOS digital pixel sensor," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 2049–2059, Dec. 2001.
- [26] A. Harton, M. Ahmed, A. Beuhler, F. Castro, L. Dawson, B. Herold, G. Kujawa, K. Lee, R. Mareachen, and T. Scaminaci, "High dynamic range CMOS image sensor with pixel level ADC and *in situ* image enhancement," in *Sensors and Camera Systems for Scientific and Industrial Applications VI. Proc. SPIE*, Mar. 2005, vol. 5677, pp. 67–77.
- [27] Y. Chi, U. Mallik, E. Choi, M. Clapp, G. Gauwenberghs, and R. Etienne-Cummings, "Cmos pixel-level ADC with change detection," in *Proc. Int. Symp. Circuits and Systems (ISCAS)*, May 2006, pp. 1647–1650.
- [28] O. Yadid-Pecht, B. Pain, C. Staller, C. Clark, and E. Possum, "CMOS active pixel sensor star tracker with regional electronic shutter," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 285–288, Feb. 1997.
- [29] M. Barbara, P. Burgi, A. Mortara, P. Nussbaum, and F. Heitge, "A 100 × 100 pixel silicon retina for gradient extraction with steering filter capabilities and temporal output coding," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 160–172, Feb. 2002.
- [30] C. Ryan, "Applications of a four-quadrant multiplier," *IEEE J. Solid-State Circuits*, vol. 5, no. 1, pp. 45–48, Feb. 1970.
- [31] S. Liu and Y. Hwang, "CMOS squarer and four-quadrant multiplier," *IEEE Trans. Circuits Syst. I, Fundam. Theory Applicat.*, vol. 42, no. 2, pp. 119–122, Feb. 1995.
- [32] C. Wu, Y. Shih, J. Lan, C. Hsieh, C. Huang, and J. Lu, "Design, optimization, and performance analysis of new photodiode structures for CMOS active-pixel-sensor (APS) imager applications," *IEEE Sensors J.*, vol. 4, no. 1, pp. 135–144, Feb. 2004.
- [33] I. Shcherback, A. Belenky, and O. Yadid-Pecht, "Empirical dark current modeling for complementary metal oxide semiconductor active pixel sensor," *Opt. Eng.*, vol. 41, no. 6, pp. 1216–1219, Jun. 2002.
- [34] I. Shcherback and O. Yadid-Pecht, "Photoresponse analysis and pixel shape optimization for CMOS active pixel sensors," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 12–18, Jan. 2003.
- [35] J. Lee and R. Hornsey, "CMOS photodiodes with substrate openings for higher conversion gain in active pixel sensor," in *IEEE Workshop on CCDs and Advanced Image Sensors*, Crystal Bay, NV, Jun. 2001.
- [36] G. Chapinal, S. Bota, M. Moreno, J. Palacin, and A. Herms, "A 128 × 128 CMOS image sensor with analog memory for synchronous image capture," *IEEE Sensors J.*, vol. 2, no. 2, pp. 120–127, Apr. 2002.
- [37] F. Yang and M. Paindavoine, "Implementation of an RBF neural network on embedded systems: Real-time face tracking and identity verification," *IEEE Trans. Neural Networks*, vol. 14, no. 5, pp. 1162–1175, Sep. 2003.
- [38] N. Farrugia, F. Mamalet, S. Roux, F. Yang, and M. Paindavoine, "A parallel face detection system implemented on FPGA," in *In IEEE Int. Symp. Circuits and Systems (ISCAS 2007)*, New Orleans, May 2007, pp. 3704–3707.
- [39] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 11, pp. 1408–1423, Nov. 2004.



Jérôme Dubois is a Normalien of the 2001 promotion. He obtained a competitive examination, in electrical engineering, for post on the teaching staff of first cycle universities in July 2004. He receive Masters degree in image processing in June 2005. He is currently a Mph.D. student and Instructorship at Laboratory LE2I and University of Burgundy.

His research interests include the design, development implementation, and testing of silicon retinas for multi-processing and high-speed image sensor.



Dominique Ginjac received the Ph.D. degree in electronics and image processing from Clermont-Ferrand University, France, in 1999.

He is currently an Associate Professor at the University of Burgundy, France, and member of LE2I UMR CNRS 5158 (Laboratory of Electronic, Computing and Imaging Sciences). His main research topics are image acquisition and embedded image processing on CMOS VLSI chips.



Barthélemy Heyrman received the Ph.D. degree in electronics and image processing from Burgundy University, France, in 2005.

He is currently an Associate Professor at the University of Burgundy, France, and a member of LE2I UMR CNRS 5158 (Laboratory of Electronic, Computing and Imaging Sciences). His main research topics are system-on-chip smart camera and embedded image processing chips.



Michel Paidavoine received the Ph.D. degree in electronics and signal processing from Montpellier University, France, in 1982.

He was with Fairchild CCD Company for two years as an engineer specializing in CCD sensors. He joined Burgundy University in 1985 as maitre de conférence and is currently full Professor at LE2I UMR-CNRS, Laboratory of Electronic, Computing and Imaging Sciences, Burgundy University, France. His main research topics are image acquisition and real-time image processing. He is also one of the

main managers of ISIS (a research group in signal and image processing of the French National Scientific Research Committee).

A SIMD Programmable Vision Chip with High Speed Focal Plane Image Processing

Dominique Ginjac, Jérôme Dubois, Michel Paindavoine, and Barthélémy Heyrman

Abstract—A high speed analog VLSI image acquisition and low-level image processing system is presented. The architecture of the chip is based on a dynamically reconfigurable SIMD processor array. The chip features a massively parallel architecture enabling the computation of programmable mask-based image processing in each pixel. Extraction of spatial gradients and convolutions such as Sobel operators are implemented on the circuit. Each pixel include a photodiode, an amplifier, two storage capacitors, and an analog arithmetic unit based on a four-quadrant multiplier architecture. A 64×64 pixel proof-of-concept chip was fabricated in a $0.35 \mu\text{m}$ standard CMOS process, with a pixel size of $35 \mu\text{m} \times 35 \mu\text{m}$. A dedicated embedded platform including FPGA and ADCs has also been designed to evaluate the vision chip. The chip can capture raw images up to 10 000 frames per second and runs low-level image processing at a framerate of 2000 to 5000 frames per second.

Index Terms—CMOS Image Sensor, Parallel architecture, SIMD, High-speed image processing, Analog arithmetic unit.

I. INTRODUCTION

TODAY, digital cameras are rapidly becoming ubiquitous, due to reduced costs and increasing demands of multimedia applications. Improvements in the growing digital imaging world continue to be made with two main image sensor technologies: charge coupled devices (CCD) and CMOS sensors. Historically, CCDs have been the dominant image-sensor technology. However, the continuous advances in CMOS technology for processors and DRAMs have made CMOS sensor arrays a viable alternative to the popular CCD sensors. This led to the adoption of CMOS image sensors in several high-volume products, such as webcams, mobile phones, PDAs for example. Furthermore, new recent technologies provide the ability to integrate complete CMOS imaging systems at focal plane, with analog to digital conversion, memory and processing [1]–[5]. By exploiting these advantages, innovative CMOS sensors have been developed and have demonstrated fabrication cost reduction, low power consumption, and size reduction of the camera [6]–[8].

The main advantage of CMOS image sensors is the flexibility to integrate processing down to the pixel level. As CMOS image sensors technologies scale to $0.18 \mu\text{m}$ processes and under, processing units can be realized at chip level (*system-on-chip* approach), at column level by dedicating processing elements to one or more columns, or at pixel-level by integrating a specific unit in each pixel or local of neighboring pixels. Most of the researches deals with chip and column-level [9]–[12]. Indeed, pixel-level processing is generally dismissed

because pixel sizes are often too large to be of practical use. However, as CMOS scales, integrating a processing element at each pixel or group of neighboring pixels becomes more feasible. Each new technology process offers 1) to integrate more processing functions in a given silicon area, or 2) to integrate the same functionalities in a smaller silicon area. This can benefit the quality of imaging in terms of resolution, noise for example by integrating specific processing functions such as correlated double sampling [13], anti blooming [14], high dynamic range [15], and even all basic camera functions (color processing functions, color correction, white balance adjustment, gamma correction) onto the same camera-on-chip [16]. Furthermore, employing a processing element per pixel offers the ability to exploit the high speed imaging capabilities of the CMOS technology by achieving massively parallel computations [17]–[20].

In this paper, we discuss hardware implementation issues of a high speed CMOS imaging system embedding low-level image processing. For this purpose, we designed, fabricated, and tested a proof-of-concept 64×64 pixel CMOS analog sensor with per-pixel programmable processing element in a standard $0.35 \mu\text{m}$ double-poly quadruple-metal CMOS technology.

The rest of the paper is organized as follows. The Section II is dedicated to the description of the high speed algorithms embedded at pixel-level. The Section III is a general description of the characteristics of the sensor. These characteristics are well detailed in the Section IV, which talks about the design of the circuit, with a full description of the main components such as the photodiode structure, the embedded analog memories, and the arithmetic unit are successively described. In the section V, we describe the test hardware platform and the chip characterization results, including an analysis of the fixed pattern noise. Finally, some experimental results of high speed image acquisition with pixel-level processing are provided in the last section of this paper.

II. HIGH SPEED FOCAL PLANE IMAGE-PROCESSING CAPABILITIES

In an increasingly digital world, the most part of imaging systems has become almost entirely digital, using only an analog-to-digital (ADC) between the sensor and the processing operators. However, low-level image processing usually involves basic operations using local masks. These local operations are spatially dependent on other pixels around the processed pixel. Since the same type of operations is applied to a very large data set, these low-level tasks are computationally intensive and require a high bandwidth between the image

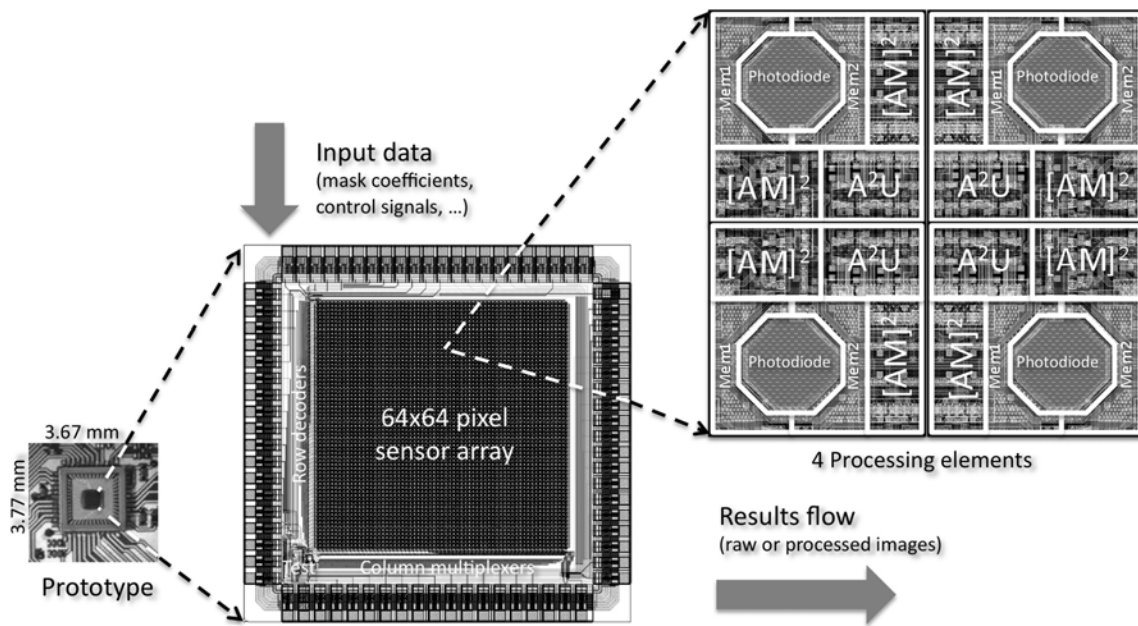


Fig. 1. Overview of the image sensor with a processor-per-pixel array

memory and the digital processor. In this case, an analog or a mixed-approach can offer superior performance leading to a smaller, faster, and lower power solution than a digital processor [21]. Low-level image processing tasks are inherently pixel-parallel in nature. Integrating a processing element within each pixel based on a single instruction multiple data (SIMD) architecture is a natural candidate to cope with the processing constraints [18]. This approach is quite interesting for several aspects. First, SIMD image-processing capabilities at focal plane have not been fully exploited because the silicon area available for the processing elements is very limited. Nevertheless, this enables massively parallel computations allowing high framerates up to thousands of images per second. The parallel evaluation of the pixels by the SIMD operators leads to processing times, independent of the resolution of the sensor. In a classical system, in which low-level image processing is externally implemented after digitization, processing times are proportional to the resolution leading to lower framerates as resolution increases. Several papers have demonstrated the potentially outstanding performance of CMOS image sensors [22]–[24]. Krymski et al. [22] describe a high speed (500 frames/s) large format 1024×1024 Active Pixel Sensor (APS) with 1024 ADCs. Stevanovic et al. [23] describe a 256×256 APS which achieves more than 1000 frames/s with variable integration times. Kleinfelder et al. [24] describe a 352×288 Digital Pixel Sensor (DPS) with per pixel bit parallel ADC achieving 10,000 frames/s or 1 Giga-pixels/s.

Secondly, the high speed imaging capability of CMOS image sensors can benefit the implementation of new complex applications at standard rates and improve the performance of existing video applications such as motion vector estimation [25]–[27], multiple capture with dynamic range [28]–[30], motion capture [31], and pattern recognition [32]. Indeed,

standard digital systems are unable to operate at high framerates, because of the high output data rate requirements for the sensor, the memory, and the processing elements. Integrating the memory and processing with the sensor on the same chip removes the classical input output bottleneck between the sensor and the external processors in charge of processing the pixel values. Indeed, the bandwidth of the communication between the sensor and the external processors is known as a crucial aspect, especially with high resolution sensors. In such cases, the sensor output data flow can be very high, and needs a lot of hardware resources to convert, process and transmit a lot of information. So, integrating image processing at the sensor level can alleviate the high data rate problem because the pixel values are pre-processed on-chip by the SIMD operators before sending them to the external world via the communication channel. This will result in data reduction, which allows sending the data at lower data-rates, and reduces the effect of the computational-load bottleneck.

Thirdly, one of the main drawbacks to design specific circuits integrating sensing and processing on the same chip is that these vision chips are often built as special-purpose devices, performing specific and dedicated tasks, and not reusable in another context [33]. So, it can be widely beneficial to integrate a versatile device, whose functionality can be easily modified. Moreover, except the basic operations such as convolutions with small masks, the majority of computer vision algorithms requires the sequential execution of different successive low level image processing on the same data. So, each processing element must be built around a programmable execution unit, communication channels, and local memories dedicated to intermediate results. Because of the very limited silicon area, the processing units are necessarily very simple, providing the best compromise between various factors such

as versatility, complexity, parallelism, processing speeds and resolution.

To sum up, the flexibility to integrate processing down to the pixel level allows us to rearchitect the entire imaging system to achieve much higher performances [34]. The key idea is 1) to capture images at a very high framerate, 2) to process the data on each pixel with a SIMD programmable architecture exploiting the high on-chip bandwidth between the sensor, the memory and the elementary processors and 3) to provide results at the best framerate depending on the complexity of the image processing. In this paper, we present our approach to the design of a massively parallel, SIMD vision chip based implementing low level image processing based on local masks. Our analog processing operators are fully programmable devices by dynamic reconfiguration, and can be viewed as a software-programmable image processor dedicated to low-level image processing. The main objectives of our design are: 1) to evaluate the potential for high speed snap-shot imaging and, in particular, to reach a 10 000 frames/s rate, 2) to demonstrate a versatile and reconfigurable processing unit at pixel-level, and 3) to provide an original platform for experimenting with low-level image processing algorithms that exploit high-speed imaging.

III. DESCRIPTION OF THE ARCHITECTURE

The proof-of-concept chip presented in this paper is depicted in Fig. 1. The core includes a two-dimensional array of 64×64 identical processing element (PE). It follows the single instruction multiple data (SIMD) computing paradigm. Each of the PE is able to convolve the pixel value issued from the photodiode by applying a set of mask coefficients to the image pixel values located in a small neighborhood. The key idea is that a global control unit can dynamically reconfigure the convolution kernel masks and then implements the most part of low-level image processing algorithms. This confers the functionality of programmable processing devices to the PEs embedded in the circuit. Each individual PE includes the following elements:

- a photodiode dedicated to the optical acquisition of the visual information and the light-to-voltage transduction,
- two *Analog Memory, Amplifier and Multiplexer* structures called $[AM]^2$, which serve as intelligent pixel memories and are able to dissociate the acquisition of the current frame in the first memory and the processing of the previous frames in the second memory,
- an *Analog Arithmetic Unit* named A^2U based on four analog multipliers, which performs the linear combination of the four adjacent pixels using a 2×2 convolution kernel.

In brief, each PE includes 38 transistors integrating all the analog circuitry dedicated to the image processing algorithms. The global size of the PE is $35 \mu\text{m} \times 35 \mu\text{m}$ ($1225 \mu\text{m}^2$). The active area of the photodiode is $300 \mu\text{m}^2$, giving a fill-factor of 25 %. The chip has been realized in a standard $0.35 \mu\text{m}$ double-poly quadruple-metal CMOS technology and contains about 160 000 transistors on a $3.67 \text{ mm} \times 3.77 \text{ mm}$ die (13.83 mm^2). The chip also contains test structures on the

bottom left of the chip. These structures are used for detailed characterization of the photodiodes and processing units.

IV. CIRCUIT DESIGN

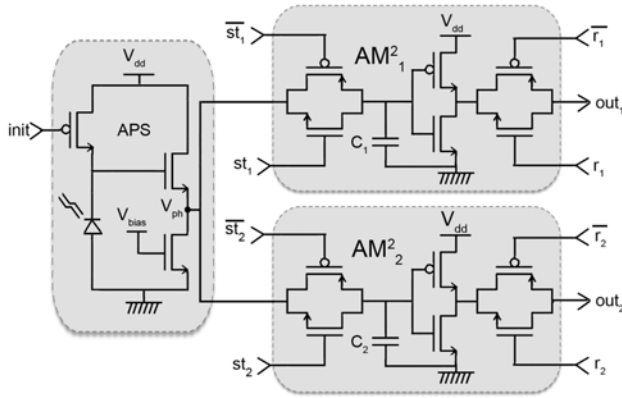
A. Pixel Structure

Each pixel in the CMOS image sensor array consists of a photodiode and a processing unit dedicated to low-level image processing based on neighborhoods. In our chip, the type of photodiodes is one of the simplest photo element in CMOS image sensor technology. It consists of N-type photodiodes based on an n^+ -type diffusion in a p-type silicon substrate. The depletion region is formed in the neighborhood of the photodiode cathode. Optically generated photocarriers diffuse to neighboring junctions [35]. In order to achieve good performances, the photodiodes should be designed and optimized carefully, in order to minimize critical parameters such as the dark current and the spectral response [36]. The shape of photodiode layout, the structure of the photodiode, and the layout have significant influences on the performance of the whole imager [37], [38]. The active area of the photodiode absorbs the illumination energy and turns that energy into charge carriers. This active area must be large as possible in order to absorb a maximum of photons. In the mean time, the control circuitry required for the readout of the collected charges and the inter-element isolation area must be as small as possible in order to obtain the best fill factor. We have theoretically analyzed, designed and benchmarked different photodiodes shapes [39], and finally, an octagonal shape based on 45° structures was chosen (see Fig. 1). More details about the photodiodes design can be found in the aforementioned paper. Here, only the basic concept behind the photodiodes design has been briefly overviewed.

The second part of the pixel is the analog processing unit. Existing works on analog pixel-level image processing can be classified into two main categories. The first one is intrapixel, in which processing is performed on the individual pixels in order to improve image quality, such as the classical Active Pixel Sensor or APS [9], [40]. The second category is interpixel, where the processing is dedicated to groups of pixels in order to perform some early vision processing and not merely to capture images. Our work clearly takes place in this category because our main objective is the implementation of various in-situ image processing using local neighborhoods. Based on this design concept, this forces a re-thinking of the spatial distribution of the processing resources, so that each computational unit can easily use a programmable neighborhood of pixels. For this purpose, the pixels are mirrored about the horizontal and the vertical axis in order to share the different *Analog Arithmetic Units* (A^2U). As example, a block of 2×2 pixels is depicted in Fig. 1. The main feature of its innovative distribution is to optimize the compactness of the metal interconnections with pixels, to contribute to a better fill factor, and to provide generality of high speed processing based on neighborhood of pixels.

B. Analog Memory, Amplifier and Multiplexer : $[AM]^2$

In order to increase the algorithmic possibilities of the architecture, the key point is the separation of the acquisition

Fig. 2. The $[AM]^2$ structure

of the light inside the photodiode and the readout of the stored value at pixel-level [41]. Thus, the storage element should keep the output voltage of the previous frames whereas the sensor integrates photocurrent for a new frame. So, we have designed and implemented dedicated pixels including a light sensitive structure and two specific circuits called *Analog Memory, Amplifier, and Multiplexer* ($[AM]^2$), as shown in Fig. 2.

The system has five successive operation modes: reset, integration, storage, amplification, and readout. All these phases are externally controlled by global signals common to the full array of pixels. They all occur in parallel over the sensor (*snapshot* mode) in order to avoid any distortion due to a row-by-row reset. In each pixel, the photosensor is a N-type photodiode associated with a PMOS transistor reset. This switch resets the integrating node to the fixed voltage V_{dd} . The pixel array is held in the reset mode until the *init* signal raises, turning the PMOS transistor off. Then, the photodiode discharges for a fixed period, according to the incidental luminous flow. The first NMOS transistor acts as a transconductance, producing the voltage V_{ph} , directly proportional to the incident light intensity. The integrated voltage is polarized around $V_{dd}/2$ by the second NMOS transistor. The calibration of the structure is ensured by the positive reference bias voltage ($V_{bias} = 1.35V$).

Following the acquisition stage, two identical subcircuits $[AM]^2_i$ (with $i = 1, 2$) take place to realize the storage phase of V_{ph} . Each $[AM]^2$ includes three pairs of NMOS and PMOS transistors and a capacitor which acts as an analog memory. The subcircuit $[AM]^2_i$ is selected when the st_i signal is turned on. Then, the associated analog switch is open allowing the integration of the photogenerated current in the corresponding C_i capacitor. Consequently, the capacitors are able to store the pixel value during the frame capture from one of the two switches. The capacitors are implemented with double-polysilicium. The size of the capacitors is as large as possible in order to respect the fill-factor and the pixel size requirements. The capacitors values are about 40 fF. They are able to store the pixel value for 20 ms with an error lower than 4 %. Behind the storage subcircuit, a basic CMOS inverter is integrated. This inverter serves as a linear high-gain amplifier since the pixel signal is polarized around $V_{dd}/2$. Finally, the

last phase consists in the readout of the stored values in the capacitors C_i . The integrated voltage across the capacitor C_i can be readout on the output out_i through one of the two switches, controlled by the r_i signals.

Fig. 3 describes the experimental results of successive acquisitions in an individual pixel. The acquisitions occur when one of the two signals st_1 or st_2 goes high. The two combinations of acquisitions are presented in this example. After the first reset, st_2 is followed by st_1 whereas the inverted sequence of acquisitions is realized after the second reset. The signal V_{ph} gives the voltage corresponding to the incidental illumination on the pixel and the two outputs (out_1 and out_2) give the voltage stored in each of the capacitors when the associated readout signal raises.

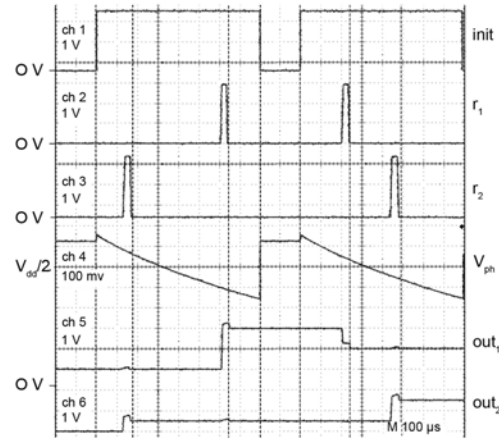


Fig. 3. High speed sequence capture with basic image processing

One of the main advantages of the two $[AM]^2$ structures is that the capture sequence can be made in the first memory in parallel with a readout sequence and/or processing sequence of the previous image stored in the second memory, as shown in Fig. 4.

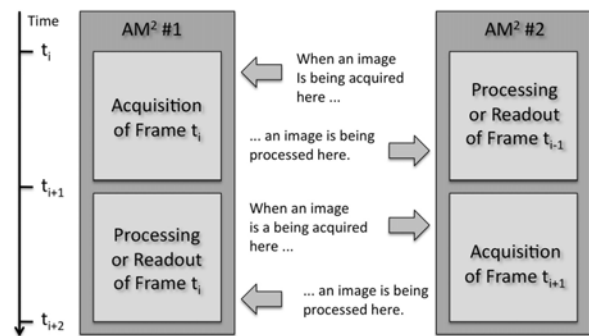


Fig. 4. Parallelism between capture sequence and readout sequence

Such a strategy has several advantages:

- 1) The framerate can be increased (up to 2x) without reducing the exposure time.
- 2) The image acquisition is time-decorrelated from image processing, implying that the architecture performance

is always the highest, and the processing framerate is maximum,

- 3) A new image is always available without spending any integration time.

C. Analog Arithmetic Unit: A^2U

When designing a pixel-level processing unit, you should consider adopting efficient strategies to minimize the silicon area occupied by the processor. To that end, we designed an analog arithmetic unit (A^2U) which is able to perform convolution of the pixels with a 2×2 dynamic kernel. This unit is based on four-quadrant analog multipliers [42], [43] named M_1 , M_2 , M_3 , and M_4 , as illustrated in Fig. 5. Each multiplier requires 5 transistors. So, the transistor count of the complete unit is only 22 transistors. The last two transistors not depicted on the Fig. 5 serve as an output switch, driven by the column signal. It features relative small area, simplicity, and high speed. These characteristics make it an interesting choice in low-level image processing embedded at focal plane. Each multiplier M_i (with $i = 1, \dots, 4$) takes two analog signals V_{i1} and V_{i2} and produces an output V_{iS} which is their product. The outputs of multipliers are all interconnected with a diode-connected transistor employed as load. Consequently, the global operation result at the V_S point is a linear combination of the four products V_{iS} . Image processing operations such as spatial convolution can be easily performed by connecting the inputs V_{i1} to the kernel coefficients and the inputs V_{i2} to the corresponding pixel values.

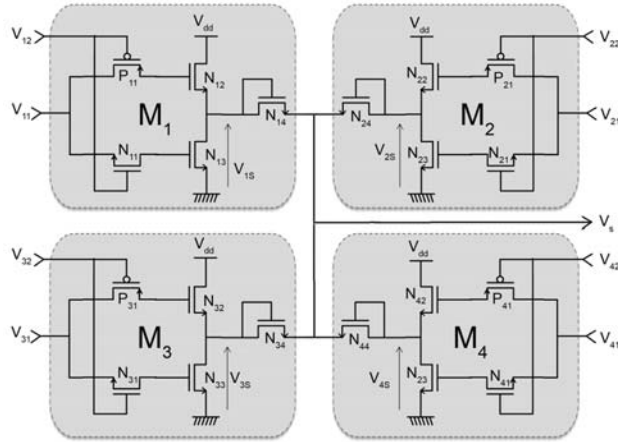


Fig. 5. The A^2U structure

In order to keep the analysis simple, it is assumed in this section that the contribution of parasitic capacitances is negligible. Considering the MOS transistors operating in sub-threshold region, the output node V_{iS} of a multiplier can be expressed as a function of the two inputs V_{i1} and V_{i2} as follows:

$$k_r (V_{ThN} - V_{i1}) (V_{i1} - V_{i2} - V_{ThN}) = (V_{i1} - V_{iS} - V_{ThN}) (V_{i2} - V_{iS} - V_{ThN} - V_{ThP}) \quad (1)$$

with k_r represents the transconductance factor, V_{ThN} and V_{ThP} are the threshold voltage for the NMOS and PMOS

transistors. Around the operating point ($V_{dd}/2$), the variations of the output node mainly depend on the product $V_{i1}V_{i2}$. So, the equation 1 can be simplified and finally, the output node V_{iS} can be expressed as a simple first-order of the two input voltages V_{i1} and V_{i2} .

$$V_{iS} = MV_{i1}V_{i2} \quad \text{with,} \quad M = \frac{k_r - 1}{2V_{ThN} + V_{ThP}} \approx 8.07/V \quad (2)$$

The important value of the coefficient M gives to the structure a good robustness by limiting the impact of the second-order intermodulation products. The first consequence is a better linearity of our multiplier design integrating only 5 transistors.

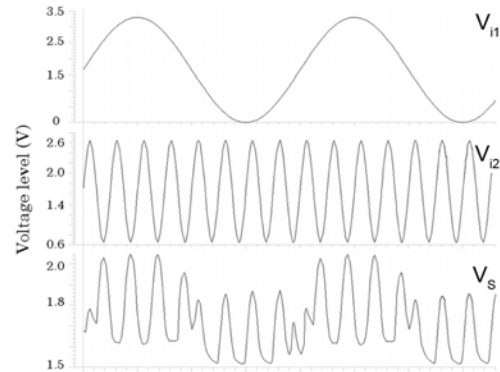
The theoretical analysis has been validated by an experimental study realized on the test structures embedded on the chip. Fig. 6 shows the experimental measures obtained with two cosine signals as inputs of the multiplier structure.

$$V_{i1} = A \cos(2\pi f_1) \quad \text{with } f_1 = 2.5 \text{ kHz} \quad (3)$$

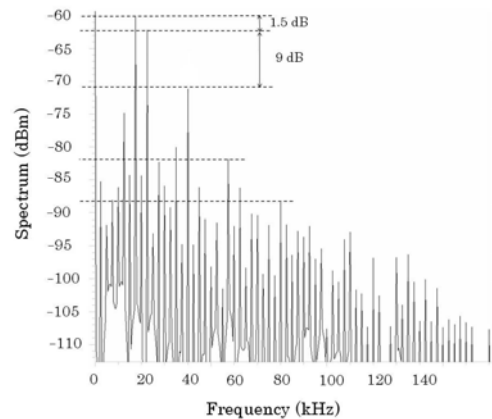
$$V_{i2} = B \cos(2\pi f_2) \quad \text{with } f_2 = 20 \text{ kHz} \quad (4)$$

In an ideal case, the voltage V_S at the output of the multiplier can be expressed as:

$$V_S = \frac{AB}{2} [\cos(2\pi(f_2 - f_1)) + \cos(2\pi(f_2 + f_1))] \quad (5)$$



(a) Multiplication of cosine signals



(b) Frequency spectrum of the result of multiplication

Fig. 6. Experimental results of the multiplication of two cosine signals by the four-quadrant multiplier

The frequency spectrum, represented in Fig. 6(b) contains two main frequencies (17.5 kHz and 22.5 kHz) around the carrier frequency. The residues which appear in the spectrum are known as intermodulations products. Intermodulation is caused by non-linear behaviour of the of the structure (around 10 kHz and 30 kHz) and the insulation defects of input pads (at 40 kHz). Nevertheless, the amplitude of these intermodulation products is significantly lower than the two main frequencies. Indeed, at 40 kHz, the level of intermodulation is 9dB under the level of the main frequencies. Therefore, the contribution of the insulation defect is eight times smaller than the main signals. Furthermore, the experimental measures highlighted that the linearity of the multiplier is maximum for input signals with amplitude range between 0.6 V and 2.6 V. This corresponds to the typical range of values coming from the pixel since the $[AM]^2$ structures provide values around $V_{dd}/2=1.65V$.

V. CHIP CHARACTERIZATION

An experimental 64×64 pixel image sensor has been developed in a $0.35 \mu\text{m}$, 3.3 V, standard CMOS process with poly-poly capacitors. Its functional testing and its characterization were performed using a specific hardware platform. The hardware part of the imaging system contains a one million Gates Spartan-3 FPGA board with 32MB SDRAM embedded. This FPGA board is the XSA-3S1000 from XESS Corporation. An interface acquisition circuit includes three ADC from Analog Device (AD9048), high speed LM6171 amplifiers and others elements such as the motor lens. Fig. 7 shows the schematic and some pictures of the experimental platform.

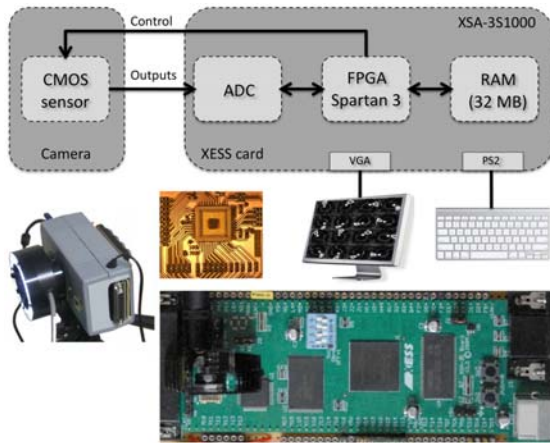


Fig. 7. Block diagram and pictures of the hardware platform including FPGA board and CMOS sensor

A. Electrical characterization

The sensor was quantitatively tested for conversion gain, sensitivity, fixed pattern noise, thermal reset noise, output levels disparities, voltage gain of the amplifier stage, linear flux, and dynamic range. Table I summarizes the main chip properties and the characterization results.

To determine these values, the sensor included specific test pixels in which some internal node voltages can be directly read. The test equipment hardware is based on a light generator with wavelength of 400 nm to 1100 nm. The sensor conversion gain was evaluated to $54 \mu\text{V}/e^-$ RMS with a sensitivity of 0.15 V/lux.s, thanks to the octagonal shape of the photodiode and the fill factor of 25%. At 10 000 frames/s, measured non-linearity is 0.12% over a 2 V range. These performances are similar to the sensor described in [24]. According to the experimental results, the voltage gain of the amplifier stage of the two $[AM]^2$ is $A_v = 12$ and the disparities on the output levels are about 4.3 %.

TABLE I
CHIP CHARACTERISTICS AND MEASUREMENTS

Technology	0.35 μm 2-poly 4-metal CMOS
Array size	64×64
Chip area	13.8 mm^2
Pixel size	$35 \mu\text{m} \times 35 \mu\text{m}$
Number of transistors	160 000
Number of transistors / pixel	38
Sensor Fill Factor	25 %
Dynamic power consumption	110 mW
Conversion gain	$54 \mu\text{V}/e^-$ RMS
Sensitivity	0.15 V/lux.s
Fixed Pattern Noise retina (FPN), dark	225 μV RMS
Thermal reset noise	68 μV RMS
Output levels disparities	4.3%
Voltage gain of the amplifier stage	12
Linear flux	98.5%
Dynamic range	68 dB

B. Fixed pattern noise

Image sensors always suffer from technology related non-idealities that can limit the performances of the vision system. Among them, fixed pattern noise (FPN) is the variation in output pixel values, under uniform illumination, due to device and interconnect mismatches across the image sensor. Two main types of FPN occur in CMOS sensors. First, offset FPN which takes place into the pixel is due to fluctuations in the threshold voltage of the transistors. Second, the most important source of FPN is introduced by the column amplifiers used in standard APS systems. In our approach, the layout is symmetrically built in order to reduce the offset FPN among each block of four pixels and to ensure uniform spatial sampling, as already depicted in the layout of a 2×2 pixel block in Fig. 1.

Furthermore, our chip does not include any column amplifier since the amplification of the pixel values takes place into the pixel by means of an inverter. So, the gain FPN is very limited and only depends on the mismatch of the two transistors. FPN can be reduced by correlated double sampling (CDS). To implement CDS, each pixel output needs to be read twice, once after reset and a second time at the end of integration. The correct pixel signal is obtained by subtracting the two values. A CDS can be easily implemented in our chip. For this purpose, the first analog memory stores the pixel value

just after the reset signal and the second memory stores the value at the end of integration. Then, at the end of the image acquisition, the two values can be transferred to the FPGA, responsible for producing the difference. In Fig. 8, the two images show fixed pattern noise with and without CDS using a 1-ms integration time. On the left image, the FPN is mainly due to the random variations in the offset voltages of the pixel-level analog structures. The experimental benchmarks of our chip reveal a FPN value of $225 \mu\text{V RMS}$. The right picture shows the same image after analog CDS, performed as described above. The final FPN has been reduced by a factor of 34 to $6.6 \mu\text{V}$. In the rest of the results, CDS has not been implemented since FPN has low values. Only, an entire dark image is subtracted from the output images on the FPGA. Focus has been made on the development of low-level image processing using the two analog memories and the associated processing unit.

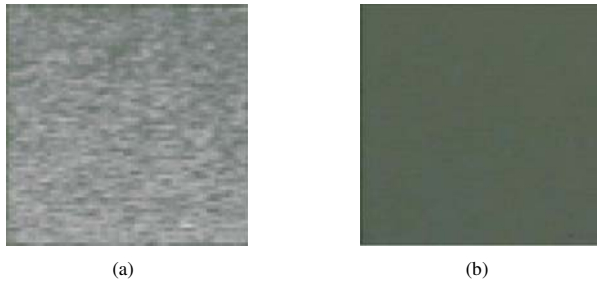


Fig. 8. Images of fixed pattern noise (a) without CDS and (b) with CDS for an integration time of 1 ms

VI. HIGH-SPEED IMAGE PROCESSING APPLICATIONS

In this section, we provide experimental results of image processing implemented on our high-speed vision chip. First, we demonstrate the possibility of acquisition of raw images at different framerates, up to 10000 frames/s. Secondly, we present an implementation of edge detection, based on the well-known Sobel operator.

A. Sample Images

The prototype chip was used for acquisition of raw images. First, sample raw images of stationary scenes were captured at different framerates, as shown in Fig. 9. In the three views, no image processing is performed on the video stream, except for amplification of the photodiodes signal. From left to right, we can see a human face obtained at 1000 frames/s, a static electric fan at 5000 frames/s, and a electronic chip at 10000 frames/s.

Fig. 10 represents different frames of a moving object, namely, a milk drop splashing sequence. In order to capture the details of such a rapidly moving scene, the sensor operates at 2500 frames/s. and stores a sequence of 50 images. The frames 1, 5, 10, 15, 20, 25, 30 and 40 are shown on the Figure.

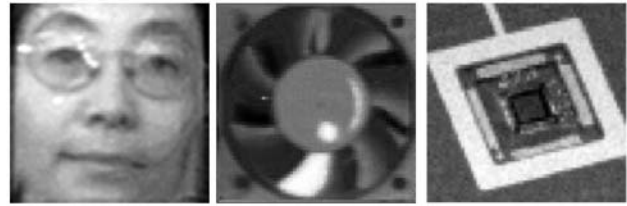


Fig. 9. Various raw images acquisition at 1000, 5000 and 10000 frames/s

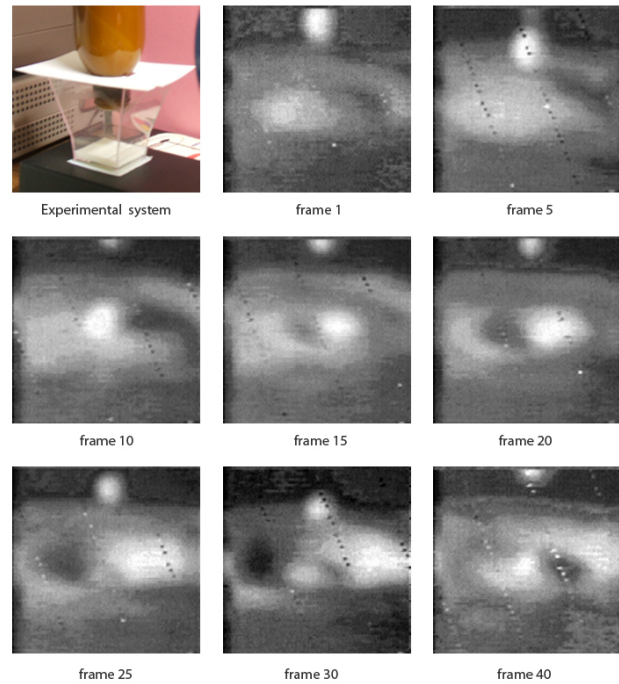


Fig. 10. A 2500 frames/s video sequence of a milk drop splashing

B. Sobel operator

The characteristics of our sensor, especially the analog processing unit, make it extremely useful for low-level image processing based on convolution masks. In this paragraph, we report an edge detector, the Sobel detector. The Sobel operator estimates the gradient of a 2D image. This algorithm is used for edge detection in the preprocessing stage of computer vision systems. The classical algorithm uses a pair of 3×3 convolution kernels (see Eq. 6), one to detect changes along the vertical axis (h_1) and another to detect horizontal contrast (h_2). For this purpose, the algorithm performs a convolution between the image and the sliding convolution mask over the image. It manipulates 9 pixels for each value to produce. The value corresponds to an approximation of the gradient centered on the processed image area.

$$h_1 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad h_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (6)$$

The structure of our architecture is well-adapted to the

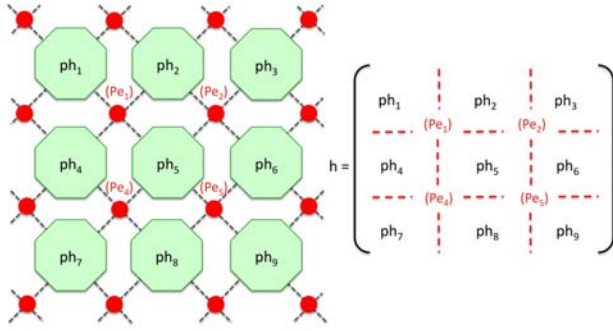


Fig. 11. 3x3 kernel used by the 4 processing elements

evaluation of the Sobel algorithm. It leads to the result directly centered on the photo-sensor and directed along the natural axes of the image. The gradient is computed in each pixel of the image by performing successive linear combinations of the 4 adjacent pixels. For this purpose, each 3x3 kernel mask is decomposed into two 2x2 masks that successively operate on the whole image. For the kernel h_1 , the corresponding 2x2 masks are:

$$m_1 = \begin{pmatrix} -1 & 0 \\ -1 & 0 \end{pmatrix} m_2 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \quad (7)$$

The Fig 11 represents the 3x3 mask centered on the pixel ph_5 . Each octagonal photodiode ph_i ($i = 1, \dots, 9$) is associated with a processing element PE_i , represented with a circle on the Figure. Each PE_i is positioned on the bottom right of its photodiode, as in the real layout of the circuit (see Fig 1). The first mask m_1 contributes to evaluate the following series of operations for the four PE_i s:

$$\begin{aligned} V_{11} &= -(V_{ph_1} + V_{ph_4}) \\ V_{12} &= -(V_{ph_2} + V_{ph_5}) \\ V_{14} &= -(V_{ph_4} + V_{ph_7}) \\ V_{15} &= -(V_{ph_5} + V_{ph_8}) \end{aligned} \quad (8)$$

and the second mask m_2 computes:

$$\begin{aligned} V_{21} &= +(V_{ph_2} + V_{ph_5}) \\ V_{22} &= +(V_{ph_3} + V_{ph_6}) \\ V_{24} &= +(V_{ph_5} + V_{ph_8}) \\ V_{25} &= +(V_{ph_6} + V_{ph_9}) \end{aligned} \quad (9)$$

with V_{ij} corresponding to the result provided by the processing element PE_j ($j = 1, 2, \dots, 9$) with the mask m_i ($i = 1, 2$), and V_{ph_k} ($k = 1, 2, \dots, 9$), the voltages representing the incidental illumination on each photodiode ph_k . Then, the evaluation of the gradient at the center of the mask can be computed by summing the different values on the external FPGA. Note that $V_{12} = -V_{21}$ and $V_{15} = -V_{24}$. So, the final sum can be simplified and written as $V_{h1} = V_{11} + V_{22} + V_{25} + V_{14}$. If we define a retina cycle as the time spent for the configuration of the coefficients kernel and the preprocessing of the image, the evaluation of the gradient on the vertical direction only spends a frame acquisition and two retina cycles. By generalization, the estimation of the complete

gradient along the two axis spend 4 cycles because it involves 4 dynamic configurations.

In short, the dynamic assignment of coefficient values from the external processor gives the system some interesting dynamic properties. The system can be easily reconfigured by changing the internal coefficients for the masks between two successive computations. First, this allows the possibility to dynamically change the image processing algorithms embedded in the sensor. Secondly, this enables the evaluation of some complex pixel-level algorithms, implying different successive convolutions. The images can be captured at higher framerate than the standard framerate, processed by exploiting the the analog memories and the reconfigurable processing elements and output at a lower framerate depending of the number of the dynamic reconfigurations. Moreover, the analog arithmetic units implementing these pixel-level convolutions drastically decrease the number of single operations such as additions and multiplications executed by an external processor (a FPGA in our case) as shown in Fig. 7. Indeed, in the case of our experimental 64×64 pixel sensor, the peak performance is equivalent to 4 parallel signed multiplications by pixel at 10000 frames/s, i.e. more than 160 million multiplications per second. With a VGA resolution (640×480), the performance level would increase to a factor of 75, leading to about 12 billion multiplications per second. Processing this data flow by external processors will imply important hardware resources in order to cope with the temporal constraints.

As an illustration of the Sobel algorithm, Fig. 12 is an example sequence of 16 images of a moving object, namely, an electric fan. Two white specific markers are placed on the fan, i.e a small circle near the rotor and a painted blade. The speed rotation of the fan is 3750 rpm. In order to capture such a rapidly moving object, a short integration time ($100 \mu s$) was used for the frames acquisition. The Sobel algorithm allow to distinguish clearly the two white markers even with a high framerate.

C. Strategies used for general spatial filter

In the preceding sections, we focused on 2x2 and 3x3 convolution masks. In the case of a 2x2 mask, the coefficients are fixed once before the beginning of the acquisition frame. In the case of a 3x3 mask, two possibilities can occur. First, the 3x3 mask presents some symmetrical properties (such as the Sobel kernel) and then the coefficients values can be fixed as in a 2x2 mask. Second, if the mask is not symmetric, it is necessary to dynamically reconfigure the coefficients during the acquisition frame. For masks which size is greater than 3x3 and more generally in the case of an NxN mask, a dynamic reconfiguration of coefficients is necessary during the acquisition frame in order to evaluate the successive values of the linear combinations of pixels.

VII. COMPARISON WITH OTHER SIMD VISION CHIPS

Table II shows an overview of some representative SIMD vision chips based on different alternatives for implementing vision processing at focal plane. The first column corresponds to the chip described in this paper. On the second column,

TABLE II
COMPARISON WITH OTHER SIMD SENSORS

Chip	This work	SCAMP-3 [44], [45]	MIMD IP Chip [46]	ACE16k [18], [47]	VCS-IV [48], [49]	PVLSAR2.2 [50]
Technology	0.35 μm	0.35 μm	1.2 μm	0.35 μm	0.35 μm	0.8 μm
Resolution	64 \times 64	128 \times 128	80 \times 78	128 \times 128	64 \times 64	128 \times 128
Pixel pitch	35 $\mu\text{m}\times$ 35 μm	49.35 $\mu\text{m}\times$ 49.35 μm	45.6 $\mu\text{m}\times$ 45 μm	75.5 $\mu\text{m}\times$ 75.3 μm	67.4 $\mu\text{m}\times$ 67.4 μm	60 $\mu\text{m}\times$ 60 μm
Fill Factor	25 %	5.6 %	33 %	?	10 %	30 %
Transistors/PE	38 tr.	128 tr.	9 tr.	198 tr.	84 tr.	50 tr.
PE Type	Analog	Analog	Analog	Analog	Digital	Digital
Framerate	10000 fps	1000 fps	9600 fps	1000 fps	1000 fps	1000 fps
Image Processing	Mask-based Image Processing	Low-level Image Processing	Spatial Convolutions	Spatial Convolutions	Low-level Image Processing	Low to mid-level Image Processing

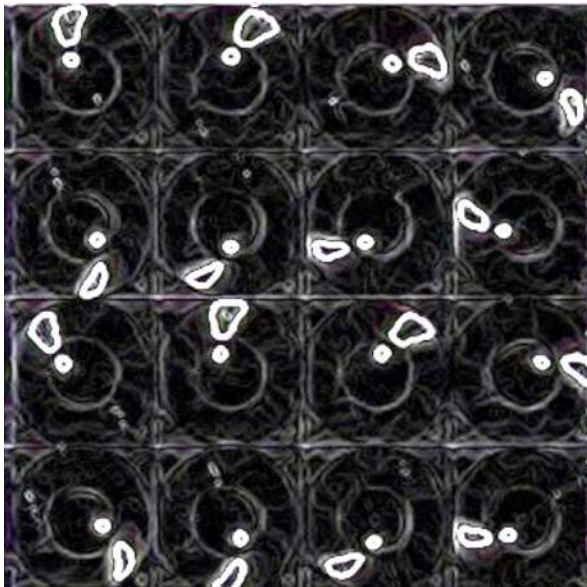


Fig. 12. Sequence of 16 images with Sobel operator

we can find the main characteristics of SCAMP-3, a general purpose processor array implementing a variety of low-level image processing tasks at a high framerate. The third column describes a Multiple Instruction Multiple Data Image Processing Chip (MIMD IP Chip) performing spatial convolutions using kernels from 3×3 to 11×11 . The fourth column presents a vision chip integrating an imager and an array of mixed-signal SIMD processing elements that can process grayscale images with Cellular Neural Network Universal Machines (CNN-UM) mode of operation. The fifth vision chip in the table is a programmable SIMD vision chip that can perform various early visual processing such as edge detection, smoothing or filtering by chaining processing elements and reconfiguring the hardware dynamically. Finally, the last one is a programmable artificial retina in which each pixel contains a tiny digital processing element capable of grey-level image processing from low to mid-level vision (motion detection, segmentation, pattern recognition).

Compared to this state-of-the-art of high-speed CMOS image sensors, one easily sees that the chip reported in this paper

lead to some major improvements. With a pixel size of 35 μm by 35 μm , the pixel pitch is almost 1.5 more compact than the smallest pixel described in [44], [45]. This contributes either to a better resolution of the sensor for a given chip area or a lower cost for a given resolution. At the same time, as compared with the other processing elements, our solution relies on a compact analog arithmetic unit based on a four-quadrant multiplier architecture using only 38 minimal size transistors. This leads to a fill factor of about 25 %. Consequently, the active area of our photodiode is bigger compared to the major part of the other chips, providing a best sensibility to the sensor at high framerates of thousands of images per second.

From the performance point of view, all the chips on the table implement low-level image processing by programming or dynamically configuring the processing elements. Our solution is able to capture image, run user-defined 3×3 convolution masks, and provide the results on the sensor output bus in less than 200 μs , giving a framerate of 5000 images per second. These temporal performances are compatible and even slightly higher than the other sensors since each of them is only able to provide low-level image processing results at 1000 frames per second, except the MIMD IP Chip.

From the programmability point of view, our chip can suffer from less versatility compared to other programmable sensors. Indeed, the implementation of complex low-level image processing require successive reconfigurations of the internal masks coefficients of the processing elements. This task may be more difficult in comparison with algorithms easily written in a machine-level language for the programmable sensors, such as the SCAMP-3 sensor.

VIII. CONCLUSION AND PERSPECTIVES

An experimental pixel sensor implemented in a standard digital CMOS 0.35 μm process has been described in this paper. Each 35 $\mu\text{m}\times$ 35 μm pixel contains 38 transistors implementing a circuit with photo-current integration, two [AM]² (Analog Memory, Amplifier, and Multiplexer), and an A²U (Analog Arithmetic Unit).

Experimental chip results reveal that raw image acquisition at 10000 frames per second can be easily achieved using the parallel A²U implemented at pixel-level. With basic image processing, the maximal framerate slows down to about

5 000 fps. The potential for dynamic reconfiguration of the sensor was also demonstrated in the case of the Sobel operator.

The next step in our research will be the design of a similar circuit in a modern 130nm CMOS technology. The main objective will be to design a pixel of less than $10 \mu\text{m} \times 10 \mu\text{m}$ with a fill factor of 20%. A second possibility would be the design of a sensor with emergent technologies using amorphous silicon in which 3D pixels can be built. With a photosensitive layer placed just behind the optical part, the pixel fill factor can reach 100% since the processing elements can be packed in the empty space below the photodiode.

Thus, with the increasing scaling of the transistors in such technologies, we could consider the implementation of more sophisticated image processing operators dedicated to face localization and recognition. Previous works of our team [51] have demonstrated the needs of dedicated CMOS sensors embedding low-level image processing such as features extraction. Moreover, actual works [52] focus on a recent face detector named Convolutional Face Finder (CFF) [53]. CFF is based on a multi-layer convolutional neural architecture. The CFF consists of six successive neural layers. The first four layers extract characteristic features, and the last two perform the classification. Our objective would be to implement at pixel-level the first layers based on convolutions by different masks from 2×2 to 5×5 .

In order to evaluate this future chip in some realistic conditions, we would like to design a CIF sensor (352×288 pixels), which leads to a $3.2 \text{ mm} \times 2.4 \text{ mm}$ in a 130 nm technology. The exploitation of high FPS capability with this sensor could be achieved with the integration of a dedicated output module able to cope with a gigapixel per second bandwidth. Another possible solution is to assemble 64×64 pixel modules with a dedicated output bus for each of them. In the same time, we will focus on the development of a fast analog to digital converter (ADC). The integration of this ADC on future chips will allow us to provide new and sophisticated vision systems on chip (ViSOC) dedicated to digital embedded image processing at thousands of frames per second.

REFERENCES

- [1] E. Fossum, "Active pixel sensors: Are CCDs dinosaurs?" *International Society for Optical Engineering (SPIE)*, vol. 1900, pp. 2–14, 1993.
- [2] —, "CMOS Image Sensor : Electronic Camera On A CHIP," *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1689–1698, October 1997.
- [3] A. El Gamal, D. Yang, and B. Fowler, "Pixel level processing – Why, What and How?" in *Proceedings of the SPIE Electronic Imaging '99 conference*, vol. 3650, January 1999, pp. 2–13.
- [4] P. Seitz, "Solid-State Image Sensing," *Handbook of computer Vision and Applications*, vol. 1, pp. 165–222, 2000.
- [5] D. Litwiller, "CCD vs. CMOS: Facts and Fiction," *Photonics Spectra*, pp. 154–158, January 2001.
- [6] C. H. Aw and B. Wooley, "A 128128-pixel standard-cmos image sensor with electronic shutter," *IEEE Journal of Solid State Circuits*, vol. 31, no. 12, pp. 1922–1930, 1996.
- [7] M. Loinaz, K. Singh, A. Blanksby, D. Inglis, K. Azadet, and B. Ackland, "A 200mv 3.3v CMOS Color Camera IC Producing 352×288 24-b Video at 30 Frames/s," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 2092–2103, 1998.
- [8] S. Smith, J. Hurwitz, M. Torrie, D. Baxter, A. Holmes, M. Panaghiston, R. Henderson, A. Murray, S. Anderson, and P. Denyer, "A single-chip 306×244 -pixel CMOS NTSC video camera," in *In ISSCC Digest of technical papers*, San Francisco, CA, 1998, pp. 170–171.
- [9] O. Yadid-Pecht and A. Belenky, "In-Pixel Autoexposure CMOS APS," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 8, pp. 1425–1428, August 2003.
- [10] P. Acosta-Serafini, M. Ichiro, and C. Sodini, "A 1/3" VGA Linear Wide Dynamic Range CMOS Image Sensor Implementing a Predictive Multiple Sampling Algorithm With Overlapping Integration Intervals," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1487–1496, September 2004.
- [11] L. Kozlowski, G. Rossi, L. Blanquart, R. Marchesini, Y. Huang, G. Chow, J. Richardson, and D. Standley, "Pixel Noise Suppression via SoC Management of Target Reset in a 1920×1080 CMOS Image Sensor," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2766–2776, December 2005.
- [12] M. Sakakibara, S. Kawahito, D. Handoko, N. Nakamura, M. Higashi, K. Mabuchi, and H. Sumi, "A High-Sensitivity CMOS Image Sensor With Gain-Adaptative Column Amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 5, pp. 1147–1156, May 2005.
- [13] R. H. Nixon, S. E. Kemeny, C. O. Staller, and E. R. Fossum, " 128×128 CMOS photodiode-type active pixel sensor with on-chip timing, control, and signal chain electronics," in *Proc. SPIE Vol. 2415, p. 117-123, Charge-Coupled Devices and Solid State Optical Sensors V, Morley M. Blouke; Ed., ser. Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, M. M. Blouke, Ed., vol. 2415, Apr. 1995, pp. 117–123.
- [14] S. Wu, H. Chien, D. Young, C. Tseng, C. Wang, C. Chang, and Y. Hsiao, "A high performance active pixel sensor with 0.18um cmos color imager technology," *Electron Devices Meeting, 2001. IEDM Technical Digest, International*, pp. 555–558, 2001.
- [15] S. Decker, D. McGrath, K. Brehmer, and C. Sodini, "A 256256 cmos imaging array with wide dynamic range pixels and column-parallel digital output," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 2081–2091, Dec 1998.
- [16] K. Yoon, C. Kim, B. Lee, and D. Lee, "Single-chip cmos image sensor for mobile applications," *IEEE Journal of Solid-State Circuits*, Dec 2002.
- [17] A. Krymsky and T. Niarong, "A 9-V/Lux 5000-Frames/s 512×512 CMOS Sensor," *IEEE Transactions on Electron Devices*, vol. 50, no. 1, pp. 136–143, January 2003.
- [18] G. Cembrano, A. Rodriguez-Vazquez, R. Galan, F. Jimenez-Garrido, S. Espejo, and R. Dominguez-Castro, "A 1000 FPS at 128×128 Vision Processor With 8-Bit Digitized I/O," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1044–1055, July 2004.
- [19] L. Lindgren, J. Melander, R. Johansson, and B. Miller, "A Multiresolution 100-GOPS 4-Gpixels/s Programmable Smart Vision Sensor for Multi-sense Imaging," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1350–1359, June 2005.
- [20] Y. Sugiyama, M. Takumi, H. Toyoda, N. Mukozaka, A. Ithori, T. Kurashina, Y. Nakamura, T. Tonbe, and S. Mizuno, "A High-Speed CMOS Image With Profile Data Acquiring Function," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 2816–2823, 2005.
- [21] D. Martin, H.-S. Lee, and I. Masaki, "A mixed-signal array processor with early vision applications," *Solid-State Circuits, IEEE Journal of*, Mar 1998.
- [22] A. Krymski, D. Van Blerkom, A. Andersson, N. Bock, B. Mansoorian, and E. Fossum, "A high speed, 500 frames/s, 1024×1024 cmos active pixel sensor," *VLSI Circuits, 1999. Digest of Technical Papers. 1999 Symposium on*, pp. 137–138, 1999.
- [23] N. Stevanovic, M. Hillebrand, B. Hosticka, and A. Teuner, "A cmos image sensor for high-speed imaging," *Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International*, pp. 104–105, 449, 2000.
- [24] S. Kleinfelder, S. Lim, X. Liu, and A. El Gamal, "A 10 000 Frames/s CMOS Digital Pixel Sensor," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 2049–2059, December 2001.
- [25] D. Handoko, K. S. Y. Takokoro, M. Kumahara, and A. Matsuzawa, "A CMOS image sensor for local-plane motion vector estimation," in *Symposium of VLSI Circuits*, vol. 3650, June 2000, pp. 28–29.
- [26] S. Lim and A. El Gamal, "Integrating Image Capture and Processing – Beyond Single Chip Digital Camera," in *Proceedings of the SPIE Electronic Imaging '2001 conference*, vol. 4306, San Jose, CA, January 2001.
- [27] X. Liu and A. El Gamal, "Photocurrent estimation from multiple non-destructive samples in a CMOS image sensor," in *Proceedings of the SPIE Electronic Imaging '2001 conference*, vol. 4306, San Jose, CA, January 2001.
- [28] D. Yang, A. El Gamal, B. Fowler, and H. Tian, "A 640×512 CMOS Image Sensor with Ultra Wide Dynamix Range Floating-Point Pixel

- Level ADC," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 1821–1834, December 1999.
- [29] O. Yadid-Pecht and E. Fossum, "CMOS APS with autoscaling and customized wide dynamic range," in *IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, vol. 3650, June 1999, pp. 48–51.
- [30] D. Stoppa, A. Somoni, L. Gonzo, M. Gottardi, and G.-F. Dalla Betta, "Novel CMOS Image Sensor With a 132-dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1846–1852, December 2002.
- [31] X. Liu and A. El Gamal, "Simultaneous image formation and motion blur restoration via multiple capture," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, 2001, pp. 1841–1844.
- [32] C.-Y. Wu and C.-T. Chiang, "A Low-Photocurrent CMOS Retinal Focal-Plane Sensor With a Pseudo-BJT Smoothing Network and an Adaptive Current Schmitt Trigger for Scanner Applications," *IEEE Sensors Journal*, vol. 4, no. 4, pp. 510–518, August 2004.
- [33] P. Dudek and P. Hicks, "An analogue simd focal-plane processor array," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, vol. 4, 6–9 May 2001, pp. 490–493 vol. 4.
- [34] A. El Gamal and H. Eltoukhy, "Cmos image sensors," *Circuits and Devices Magazine, IEEE*, May–June 2005.
- [35] J. Lee and R. Hornsey, "CMOS Photodiodes with Substrate Openings for Higher Conversion Gain in Active Pixel Sensor," in *IEEE Workshop on CCDs and Advanced Image Sensors*, Crystal Bay, Nevada, June 2001.
- [36] C. Wu, Y. Shih, J. Lan, C. Hsieh, C. Huang, and J. Lu, "Design, optimization, and performance analysis of new photodiode structures for CMOS active-pixel-sensor (APS) imager applications," *IEEE Sensors Journal*, vol. 4, no. 1, pp. 135–144, February 2004.
- [37] I. Shcherback, A. Belenky, and O. Yadid-Pecht, "Empirical dark current modeling for complementary metal oxide semiconductor active pixel sensor," *Optical Engineering*, vol. 41, no. 6, pp. 1216–1219, June 2002.
- [38] I. Shcherback and O. Yadid-Pecht, "Photoresponse analysis and pixel shape optimization for CMOS active pixel sensors," *IEEE Transactions on Electron Devices*, vol. 50, no. 1, pp. 12–18, January 2003.
- [39] J. Dubois, D. Ginjac, M. Paindavoine, and B. Heyrman, "A 10 000 fps cmos sensor with massively parallel image processing," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, p. In Press, March 2008.
- [40] O. Yadid-Pecht, B. Pain, C. Staller, C. Clark, and E. Fossum, "CMOS Active Pixel Sensor Star Tracker with Regional Electronic Shutter," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 2, pp. 285–288, February 1997.
- [41] G. Chapinal, S. Bota, M. Moreno, J. Palacin, and A. Herms, "A 128 × 128 CMOS Image Sensor With Analog Memory for Synchronous Image Capture," *IEEE Sensors Journal*, vol. 2, no. 2, pp. 120–127, April 2002.
- [42] C. Ryan, "Applications of a four-quadrant multiplier," *IEEE Journal of Solid-State Circuits*, vol. 5, no. 1, pp. 45–48, Feb 1970.
- [43] S. Liu and Y. Hwang, "CMOS Squarer and Four-Quadrant Multiplier," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 42, no. 2, pp. 119–122, Feb 1995.
- [44] P. Dudek, "Implementation of simd vision chip with 128×128 array of analogue processing elements," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, vol. 5, 2005, pp. 5806–5809.
- [45] P. Dudek and S. Carey, "General-purpose 128×128 SIMD processor array with integrated image sensor," *Electronic Letters*, vol. 42, no. 12, pp. 678–679, Jan 2006.
- [46] R. Etienne-Cummings, Z. Kalayjian, and D. Cai, "A programmable focal-plane simd image processor chip," *Solid-State Circuits, IEEE Journal of*, vol. 36, no. 1, pp. 64–73, Jan 2001.
- [47] A. Rodriguez-Vasquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona, F. Jimenez-Garrido, R. Dominguez-Castro, and S. Meana, "ACE16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 5, pp. 851–863, 2004.
- [48] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida, "A digital vision chip specialized for high-speed target tracking," *IEEE Transactions on Electron Devices*, vol. 50, no. 1, pp. 191–199, 2003.
- [49] T. Komuro, S. Kagami, and M. Ishikawa, "A dynamically reconfigurable SIMD processor for a vision chip," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 265–268, 2004.
- [50] F. Paillet, D. Mercier, and T. Bernard, "Second generation programmable artificial retina," 1999, pp. 304–309.
- [51] F. Yang and M. Paindavoine, "Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1162–1175, Sept 2003.
- [52] N. Farrugia, F. Mamalet, S. Roux, F. Yang, and M. Paindavoine, "A Parallel Face Detection System Implemented on FPGA," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, New Orleans, USA, May 27–30 2007, pp. 3704–3707.
- [53] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1408–1423, Nov 2004.

3 Travaux de recherche en Adéquation Algorithme Architecture

3.1 Contexte scientifique

Nous assistons à une évolution rapide des architectures de traitement du signal et des images, construites par assemblage de processeurs standards, de circuits spécifiques programmables (DSP, FPGA) et de circuits intégrés dédiés (ASIC). Cette évolution associée à une amélioration aussi rapide des méthodes et outils logiciels d'aide à l'implantation, ont permis de réaliser à des coûts raisonnables certaines implantations d'algorithmes complexes que l'on osait même pas envisager il y a quelques années. On est peu à peu passé d'une étude séparée des algorithmes et des architectures, à une approche méthodologique globale de plus en plus formalisée.

L'Adéquation Algorithme Architecture (AAA) consiste en l'étude simultanée des aspects algorithmiques et architecturaux en prenant en compte leurs interactions dans le but d'effectuer une implantation la plus optimisée possible des algorithmes sur des plateformes matérielles bien déterminées. Ces plateformes dédiées peuvent aller de la simple carte électronique embarquable aux systèmes complexes comportant plusieurs centaines de processeurs communiquant via un réseau spécifique.

Le souci d'une exploitation maximale de la puissance de calcul offerte par de telles architectures et la volonté de réduction du coût de l'implantation ont amené les chercheurs du domaine à étudier conjointement les algorithmes et les architectures cibles. En France, cette thématique a été soutenue dès sa création en 1990 par le GDR TDS, puis par le GDR ISIS sous forme de groupes de travail permettant à la communauté francophone de se rencontrer et d'échanger de nombreuses idées. Dans ce contexte, les travaux menés par les différentes équipes en jeu ont permis de développer des méthodologies plus ou moins formelles permettant d'effectuer l'implantation optimisée d'un algorithme (minimisation des composants logiciels et matériels) tout en réduisant les temps de développement et les coûts de l'application étudiée. Les méthodologies d'AAA sont principalement basées sur un modèle unifié d'algorithme, d'architecture et d'implantation, prenant en compte les contraintes (temps-réel, embarquabilité, ...), la nature distribuée des informations à traiter (multi-capteurs et actionneurs, données distribuées, ...), la nécessité de réutiliser et de porter facilement algorithmes et architectures. Une telle approche formelle permet d'effectuer des vérifications le plus tôt possible dans le cycle de développement de l'application, de poser des problèmes d'optimisation pour dimensionner au mieux les architectures de circuits spécialisés et/ou de machines, et enfin de générer automatiquement du code exécutable. Cette approche permet d'une part d'améliorer les techniques de prototypage rapide, et d'autre part d'aborder de manière plus claire le problème de la conception conjointe logiciel-matériel. Ces deux points constituent les grands enjeux du futur.

Satisfaire la complexité toujours croissante des applications temps réel embarquées en traitement d'images nécessite à la fois des outils de spécification de haut niveau et des architectures multi-composants. La méthodologie AAA rend possible la réduction du nombre d'erreurs de spécification des algorithmes et la limitation au maximum des tests matériels. Elle permet aussi à l'utilisateur de se concentrer sur les aspects temporels qui sont cruciaux dans le domaine du temps réel (réactivité du programme et temps de réponse contraint), d'étudier les relations entre le parallélisme potentiel au niveau de l'algorithme de

l'application et celui disponible au niveau de l'architecture matérielle distribuée, tout en étant déchargé de la programmation bas niveau souvent fastidieuse.

Le laboratoire LE2I à travers l'équipe « Capteurs et Architecture » possède un axe de recherche clairement identifié dans le domaine de l'Adéquation Algorithme Architecture. Cet axe concerne l'acquisition d'images et l'implantation d'algorithmes de traitements du signal et d'images en temps réel. Cette équipe, de par les nombreuses thèses soutenues et les articles publiés dans le domaine des « Capteurs » et de « l'Architecture », mais aussi de par sa participation active aux travaux du GDR ISIS thème C (Adéquation Algorithme Architecture) a acquis une notoriété scientifique sur le plan national et international dans le domaine de la recherche en Architecture des Systèmes de traitement du Signal et des Images Temps Réel. L'originalité de l'équipe est de confirmer l'approche théorique par une approche expérimentale à travers la réalisation de démonstrateurs à base de systèmes électroniques et informatiques.

Depuis mon recrutement en 2000, je suis intervenu de manière très active dans cette thématique qui constituait une suite logique à mes travaux antérieurs effectués au LASMEA (Clermont Ferrand) lors de ma thèse. Ces activités m'ont offert l'opportunité de renforcer mon activité 1) sur le développement d'outils pour des plates-formes de prototypage rapide 2) sur la mise au point d'applications de traitement d'images à fortes contraintes temporelles et 3) de participer au développement de nouvelles architectures matérielles et logicielles pour le traitement du signal et des images.

3.2 Développement d'outils de prototypage rapide d'applications de traitement d'images (2000 - 2002)

Malgré la puissance sans cesse accrue des machines séquentielles, les systèmes complexes de vision artificielle nécessitent souvent l'utilisation d'architectures parallèles dédiées afin de répondre aux contraintes temporelles. Toutefois, la programmation efficace de telles architectures demeure un exercice délicat requérant du programmeur qu'il résolve explicitement un problème d'adéquation algorithme architecture, c'est-à-dire de mise en correspondance efficace de son algorithme avec les caractéristiques des différents éléments de l'architecture matérielle. Ceci suppose typiquement l'identification au sein de l'application des différentes séquences de calcul pouvant être exécutées concurremment, la définition des interfaces de communication permettant d'échanger des données entre ces calculs et enfin la répartition des calculs et des communications sur le réseau physique de processeurs. Dans la plupart des cas, une approche purement manuelle de ces problèmes n'est pas envisageable, d'où l'existence de modèles de programmation parallèle de haut niveau destinés à décharger le programmeur de tout ou partie de ces problèmes. Cependant, la quasi totalité de ces modèles repose sur des hypothèses qui ne s'appliquent pas dans le contexte des machines de vision embarquées. Ces dernières sont en effet le plus souvent construites "sur mesure" à partir de processeurs spécialisés (DSP, FPGA par exemple) et ne bénéficient pas du support logiciel (système d'exploitation, bibliothèques de communication en particulier) traditionnellement offert pour des machines plus conventionnelles (PC et stations de travail).

Dans ce contexte, concilier sur de telles machines prototypage rapide des algorithmes – i.e. l'évaluation rapide d'une solution vis à vis d'un problème donné – et efficacité des applications reste de fait un domaine de recherche encore très ouvert. Le défi majeur se situe donc dans l'élaboration de nouveaux outils de haut niveau dédiés à l'implantation d'applications de traitement d'images sur de tels systèmes hétérogènes, de manière à faciliter le prototypage rapide d'applications avec une efficacité optimale.

Dans nos travaux initiés durant ma thèse (1995 - 1998), nous avons proposé une méthodologie de programmation parallèle fondée sur le concept de squelettes de parallélisation. Ceux-ci représentent des constructeurs génériques de haut niveau encapsulant des formes communes de parallélisme tout en dissimulant les détails relatifs à l'exploitation de ce parallélisme sur la plate-forme cible.

La principale originalité de cette approche est de limiter le travail de parallélisation au choix et à l'instanciation de constructeurs génériques en dehors de toute considération sur les caractéristiques physiques de l'architecture cible. Du point de vue du programmeur, un squelette apparaît comme une spécification incomplète d'une forme de parallélisme commune à un grand nombre d'applications, que le programmeur va spécialiser avec les fonctions de calcul séquentiel propres à son application. Cette encapsulation des détails relatifs à la mise en œuvre du parallélisme offre des propriétés extrêmement intéressantes. Premièrement, le programmeur d'applications voit son travail de parallélisation fortement s'amoinrir puisqu'il n'a plus à traiter les aspects bas niveau d'implantation. Deuxièmement, l'implantation d'un squelette sur une architecture donnée, étant réalisée une fois pour toutes, peut être précisément étudiée et optimisée garantissant ainsi une bonne efficacité. Troisièmement, le travail du programmeur d'applications se réduisant au développement de fonctions de calcul séquentiel (par exemple dans un langage impératif classique comme le C), il en résulte une plus grande portabilité des applications. En cas de modification des caractéristiques architecturales, voire de migration vers une autre plate-forme, le travail de réimplantation est limité au portage des squelettes, la spécification des applications demeurant inchangée. Quatrièmement, l'implantation d'un squelette sur une architecture étant parfaitement connue, il est envisageable de modéliser son comportement et d'en déduire un modèle analytique de performances paramétré à la fois par les caractéristiques matérielles (nombre de processeurs, vitesse des liens, etc.) et par les caractéristiques algorithmiques (temps d'exécution des fonctions de calcul, type et taille des données, etc.).

Toutefois, et malgré ces qualités, l'applicabilité des squelettes en tant que modèle général de programmation reste à ce jour un problème ouvert. Ils imposent en effet au programmeur de construire ses applications à partir d'une collection finie de constructeurs et rien ne peut garantir que cette collection permettra de couvrir toutes les applications possibles. La profusion de propositions, montre clairement les difficultés rencontrées pour définir une collection "idéale" de squelettes. Face à ce constat, notre réponse a été d'ordre essentiellement pragmatique: elle a consisté à restreindre délibérément le domaine d'applications. La définition d'une collection de squelettes a alors pu être réalisée de manière ascendante à partir de l'analyse a posteriori d'applications implantées manuellement d'une part et de l'expérience accumulée par les programmeurs œuvrant dans le domaine d'autre part.

L'outil SKiPPER (SKEletal Parallel Programming EnviRonment) est un environnement complet de développement, supportant cette méthodologie (cf. Fig. 10). Il a été entièrement développé pendant mes 3 années de thèse. Il comprend une bibliothèque de squelettes dédiés au traitement d'images et un ensemble d'outils permettant de transformer la spécification haut niveau des applications en un code parallèle optimisé pour différentes machines cibles (partie gauche de la Fig. 10). Des outils dédiés à l'exécution séquentielle des applications sur stations de travail ont également été développés dans le but de pouvoir faciliter la mise au point des algorithmes en dehors de toute considération d'implantation parallèle (partie droite de la Fig. 10).

L'applicabilité des concepts mis en œuvre dans SKiPPER et des outils développés conjointement a été démontrée pendant mon année d'ATER et durant les deux premières années après mon recrutement en tant que Maître de Conférences. Diverses applications de complexité réaliste : étiquetage en composantes connexes, détection et suivi de lignes

blanches en milieu autoroutier et segmentation d'image en régions par technique de division/fusion ont été parallélisées automatiquement par l'environnement SKiPPER validant ainsi l'objectif initial de prototypage rapide d'applications parallèles de vision artificielle à fortes contraintes temporelles sur architecture dédiée.

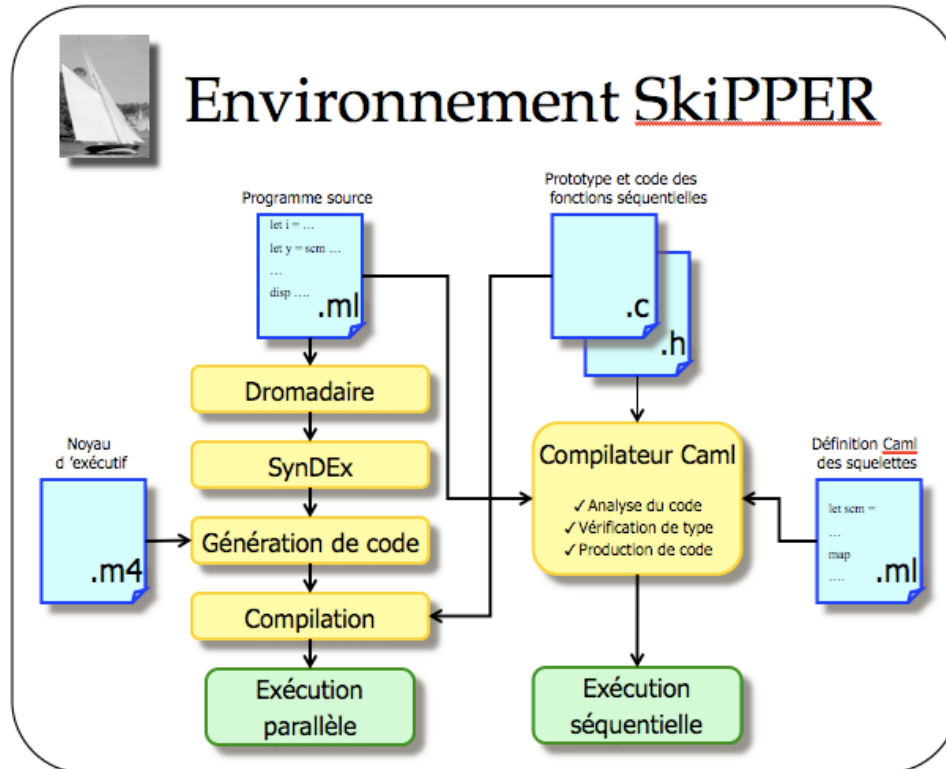


Fig. 10 : Environnement de développement SKiPPER

L'ensemble de ces travaux a conduit à la rédaction de 3 articles de revues internationales [7, 8, 9], de 1 article en revue nationale [10], de 1 conférence internationale [33] et de 4 conférences nationales [41, 42, 43, 44].

3.3 Développement d'applications de traitement d'images à fortes contraintes temporelles (2000 - 2006)

Depuis une dizaine d'années, notre équipe au sein du LE2I travaille sur des applications en reconnaissance de formes temps réel et plus particulièrement sur les aspects de localisation et d'identification d'images de visages. Dans ce contexte, nous nous sommes focalisés sur l'analyse en temps réel de séquences vidéo pour répondre aux questions suivantes : les personnes intéressantes se trouvent-elles dans la scène ? Si la réponse est Oui, quelle est alors leur position à chaque instant ? La mise en œuvre de tels systèmes de vision permet d'envisager dans un avenir très proche de multiples applications dans les domaines des interfaces homme machine, de la vidéoconférence, de la téléphonie mobile ou du contrôle d'accès sécurisé par exemple.

La reconnaissance de visages est une activité aisée pour toute personne. L'homme est en effet capable de localiser et d'identifier les visages très rapidement, dans des situations

extrêmement variées et avec une grande fiabilité. La mise au point de systèmes électroniques capables de reconnaître des visages avec un taux de réussite similaire exige toutefois la mise en œuvre d'algorithmes complexes impliquant des volumes de calculs élevés. La littérature du domaine comprend de nombreux travaux reposant sur une approche purement algorithmique. Ces solutions logicielles ont l'avantage de proposer une certaine facilité de développement en se basant sur les outils standard du marché. Cependant, les performances de ces réalisations sont très rapidement limitées par les capacités de la machine hôte (traditionnellement un PC) et ne peuvent prétendre à être utilisées dans un contexte à fortes contraintes temps réel. A l'opposé, les architectures matérielles dédiées offrent une moins grande souplesse quant au choix des algorithmes à implanter. En revanche, leur avantage est de permettre une vitesse de traitement largement supérieure aux machines de type PC, satisfaisant de fait les contraintes temporelles.

Dans notre approche, nous visons à réaliser des systèmes de localisation et de reconnaissance de visages qui peuvent être embarquables. Cela nécessite donc de trouver un juste milieu entre la complexité algorithmique et la rapidité d'exécution pour permettre le traitement en temps réel des images issues du capteur. L'adéquation algorithme architecture est ici étudiée à la fois au niveau de la plate forme matérielle et également au niveau des algorithmes à implanter. Du côté matériel, il s'agit de pouvoir concevoir un système de vision embarqué en respectant au mieux les contraintes de volume, d'énergie et de puissance de calcul. Du côté logiciel, il s'agit de mettre au point des algorithmes les plus simples possibles tout en optimisant la fiabilité de reconnaissance des visages

Un premier système de tracking dynamique et d'identification de visages en temps réel a été conçu en 2002 durant la thèse de N. Malasné. Les systèmes traditionnels utilisent deux modules indépendants, l'un pour la détection et l'autre pour la reconnaissance. Dans notre approche, le système ne requiert qu'un seul module de reconnaissance (cf. Fig. 11). Il faut noter que dans ce cas, les visages non appris par le système ne seront même pas détectés.

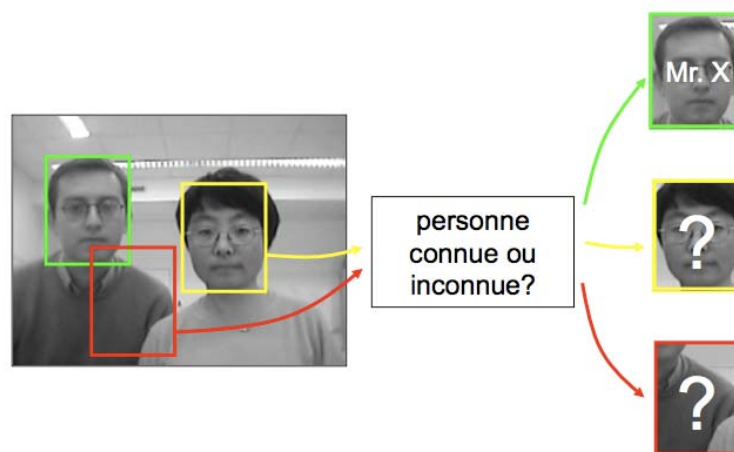


Fig. 11 : Processus proposé pour la localisation et la reconnaissance de visages

L'image est d'abord découpée en un ensemble de fenêtres d'intérêt. Chacune de ses fenêtres est alors transformée en un vecteur caractéristique. Ce vecteur peut simplement être le déroulement ligne par ligne de la fenêtre traitée. Dans un souci de réduction de taille, ce vecteur peut aussi correspondre à un sous échantillonnage de la fenêtre. Les vecteurs ainsi obtenus servent de données d'entrée au module de reconnaissance. Ce module met en œuvre un réseau de neurones de type RBF (Radial Basis Function). Ce modèle de réseau neuronal a

démonstré son efficacité dans les applications de reconnaissances de visages car il a l'avantage de posséder une bonne capacité de généralisation tout en engendrant un volume de calcul modéré.

Cet algorithme de reconnaissance de visages a été implanté sur trois systèmes électroniques embarqués, respectivement à base de processeurs neuronaux ZISC, de FPGA Xilinx Vitex-II et de DSP Texas TMS320C6x. Les résultats de ces travaux ont été publiés en conférences internationales [31, 32] et en conférence nationale [40].

Plus récemment, nous nous sommes intéressés à améliorer la fiabilité de nos systèmes en étendant nos modèles de traitement 2D vers des traitements d'images panoramiques (3D). En effet, la plupart des techniques de reconnaissance de visages utilise uniquement le visage de face comme signal d'entrée, ce qui les rend sensibles aux poses et aux conditions d'éclairage. Afin de pallier ce problème majeur, nous nous sommes intéressés à la construction de visages panoramiques obtenus à partir de différentes caméras. Pour cela, nous avons développé un système simple composé de 5 caméras standard capables d'effectuer 5 acquisitions simultanées (cf. Fig. 12).

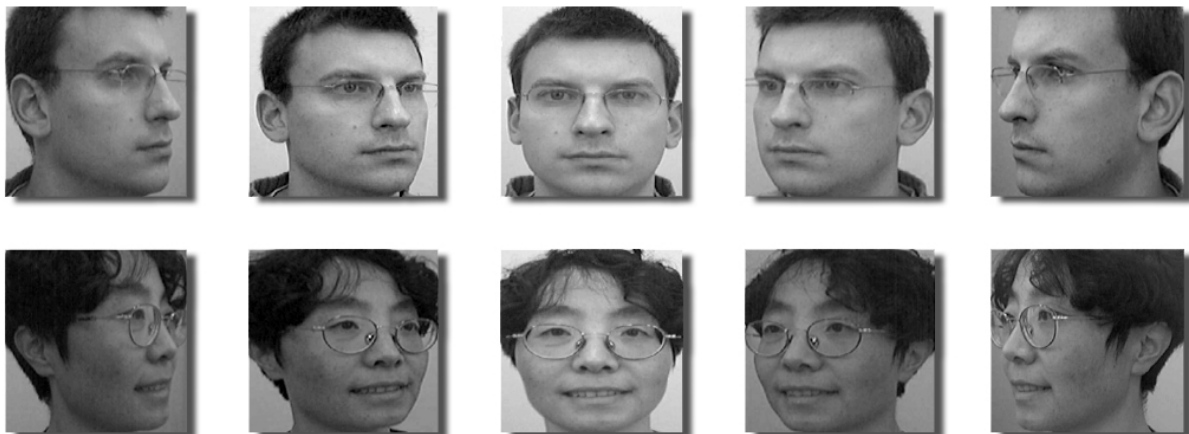


Fig. 12 : Exemples de différentes images prises par le système d'acquisition

L'algorithme implanté sur ce système est basé sur des transformations linéaires successives permettant de recomposer un visage panoramique de 150° à partir de ces 5 vues. Dans le but d'étudier les performances de reconnaissance de notre système, nous avons constitué une base de données de 3600 visages. Nos méthodes de reconnaissance de visages sont basées sur l'Analyse en Composantes Principales (ACP) et l'Analyse en Composantes Indépendantes (ACI). Les premiers résultats expérimentaux ont révélé des taux de reconnaissances très satisfaisants (de l'ordre de 90%) et ont montré la faisabilité et la viabilité du système proposé permettant d'envisager une future implantation matérielle sur une architecture dédiée.

Ces travaux ont été publiés dans un article de revue nationale [6] et dans deux chapitres de livre [11, 12].

3.4 Projet ANR : Nouvelles technologies et méthodes pour la rééducation motrice (2007 - 2011)

Ce travail de recherche s'inscrit dans le cadre d'un projet de type ANR Technologie pour la Santé (TecSan2006) intitulé « SIMACTION : Reconditionnement moteur par observation et simulation mentale ».

L'objectif est de développer de nouvelles méthodes de prévention et de rééducation afin d'améliorer en qualité et en rapidité la fonction motrice de sujets atteints de déficiences due à des lésions centrales ou au vieillissement. Le projet SIMACTION vise à explorer le champ des limites conceptuelles et technologiques en termes de réapprentissage moteur, et notamment de permettre aux patients de se rééduquer en utilisant les avancées récentes des STIC, et favoriser l'autonomie du patient lors de séance individuelle, hors d'une présence médicale. Les produits qui seront développés dans le cadre de ce projet reposent sur l'intégration des données récentes en neurosciences qui mettent en évidence un couplage étroit entre observation et exécution du mouvement avec les avancées de la capture 3D du mouvement et l'imagerie virtuelle (cf. Fig. 13). La chaîne de compétences nécessaires à la mise en œuvre de ce projet comprend la scénarisation des mouvements, la capture du mouvement, la mise en forme des points 3D capturés et leur intégration dans un environnement virtuel, l'interface de dialogue entre le système, le clinicien et le patient, la mise en œuvre des dispositifs de restitution des images.

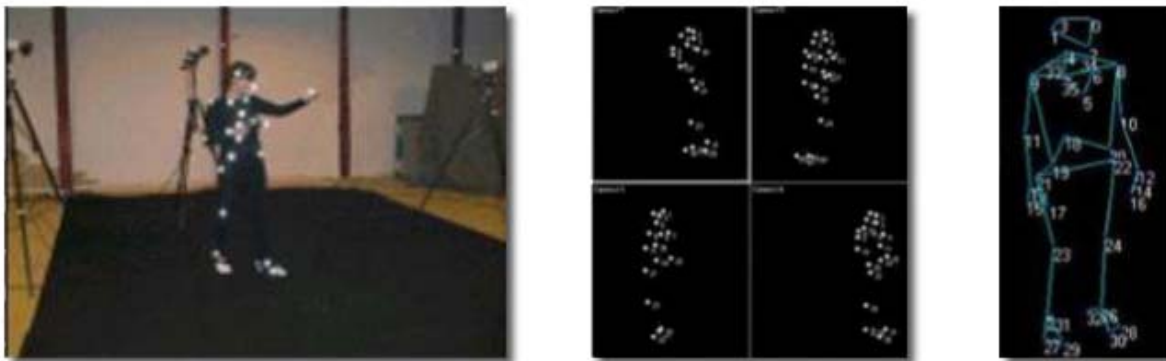


Fig. 13 : Capture de mouvements : Acquisition de points caractéristiques d'un acteur afin d'animer de manière réaliste un acteur virtuel

Deux prototypes basés sur le mécanisme de l'empathie et l'apprentissage implicite par l'observation seront développés pour le reconditionnement physique à domicile (en version légère) et en clinique (dispositif de réalité virtuelle/RV simplifié). Ce projet interdisciplinaire (STIC, SDV), vise à prévenir la dépendance des sujets âgés, permettre aux patients déficients moteur de se rééduquer en utilisant les avancées récentes des STIC, et réduire les dépenses de santé public en favorisant l'autonomie du patient lors de séances individuelles, hors d'une présence médicale. La rééducation par l'image est ici envisagée comme complément à la pratique dans un souci de faire gagner du temps au praticien et au patient en offrant un moyen d'améliorer l'espérance de vie sans incapacité ou bien d'accélérer le processus de guérison avec une relative autonomie. C'est la réactivation de réseaux neuronaux existants ou la mise en place de réseaux compensateurs qui sont les éléments clés de cette application. Si la multiplication des essais favorise l'apprentissage de séquences motrices, pour des raisons évidentes de temps et de fatigue, le nombre de

répétitions demandées au patient est limité. C'est sur ce point que réside l'intérêt de SIMACTION qui tire profit du mécanisme de simulation mentale via un support visuel.

Le LE2I intervient à deux niveaux principaux dans le projet SIMACTION. Premièrement, l'équipe de recherche en immersion virtuelle du LE2I est chargée de la conception de l'interface de réalité virtuelle, de l'optimisation de cette plate-forme (gestion de données de la maquette numérique) et de la mise au point de techniques d'interaction optimisées (utilisation de la multimodalité, développement de méthodes et outils de la réalité augmentée) permettant la meilleure immersion pour le patient.

Deuxièmement, une partie de l'équipe « Capteurs et Architectures » (M. Paindavoine et D. Ginhac) est chargée de mettre en œuvre différentes architectures matérielles d'acquisition d'images à haute cadence ainsi que leur traitement. Il s'agit d'apporter au consortium une expertise et un savoir faire dans le domaine de l'acquisition et du traitement d'images temps réel et plus particulièrement dans la mise en œuvre de capteurs d'images sophistiqués, dans l'implantation en temps réel d'algorithmes de traitements d'images sur des circuits électroniques spécifiques ainsi que dans la parallélisation et l'optimisation d'algorithmes de traitement d'images.

3.5 Projet européen : Développement de capteurs embarqués sans fil pour applications médicales (2006 – 2010)

Les plaies chroniques sont un problème de santé important dans le monde. De plus, le vieillissement de la population et les traitements plus adéquats des plaies expliquent que le marché de la prévention et du traitement des plaies ne cesse de se développer. D'où l'importance pour la société URGO d'innover dans ce domaine pour donner des informations aux cliniciens, de plus en plus rapidement, sur l'état de santé d'un patient ou encore pour permettre à ce dernier de se soigner à domicile, ce qui tend à être de plus en plus demandé de nos jours.

La compression de la jambe inférieure est un traitement efficace dans la prévention et le traitement des maladies veineuses et lymphatiques, avec ses effets en grande partie dépendants de la quantité de compression appliquée. Bien que celle-ci puisse être mesurée par des capteurs disponibles sur le marché, ceci est rarement pratiqué dans le milieu médical. Typiquement, les infirmiers comptent sur la classe de compression prescrite par le fabricant de bandes et sur leur expérience pour appliquer de tels dispositifs. Les capteurs disponibles sur le marché ne sont pas extensibles (c.-à-d. peu ou pas confortables) et presque tous câblés.

Ce travail de recherche s'inscrit dans le cadre d'un projet européen du 6^{ème} PCRD intitulé STELLA⁴ (STretchable ELectronics for Large Area applications) qui a débuté en février 2006, d'une durée de 4 ans. Ce projet réunit 11 entreprises et laboratoires de recherche européens (Urgo, Philips, IMEC, CEA-Léti, Freudenberg, ...). Dans le cadre de ce projet, la société Urgo, spécialiste des dispositifs médicaux, collabore avec le LE2I (Dominique Ginhac et Michel Paindavoine).

Le but est de développer un prototype expérimental de capteur embarqué, fin, extensible, sans fil, capable de mesurer des pressions à l'interface de la peau et d'une bande de contention et de communiquer sans fil les données acquises à destination d'un système maître externe (cf. Fig. 14). Les applications visées sont le traitement et la prévention, par la thérapie de compression, des maladies veineuses et lymphatiques. Un tel dispositif détecteur

⁴ <http://www.stella-project.de/>

de pression peut permettre aussi bien la mesure statique que dynamique. Le médecin et le patient lui-même peuvent vérifier la pression à la pose de la bande puis la dérive dans le temps. Un signal d'alarme peut être émis quand la pression dépasse un certain seuil. Un boîtier externe se charge de stocker et de traiter les données émises par le système embarqué sur la jambe du patient.

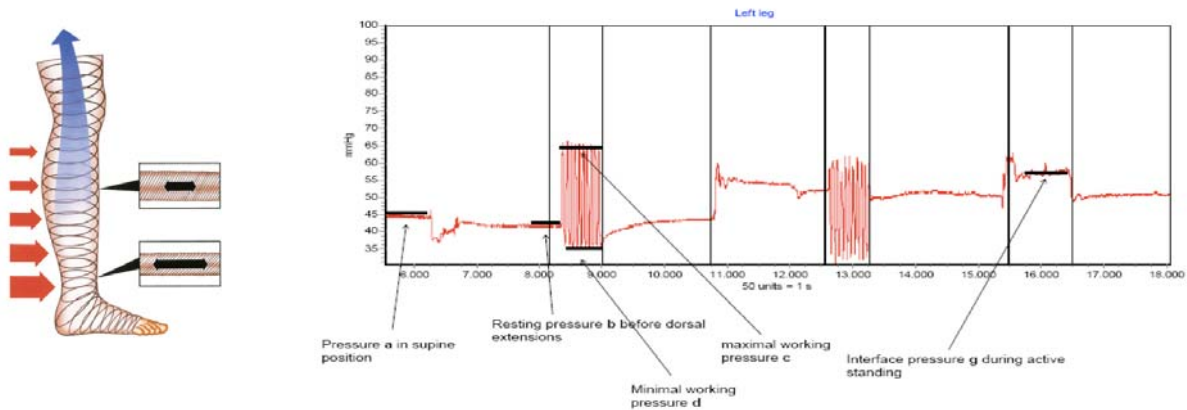


Fig. 14 : Exemple de capteur embarqué souple et mesures typique

Ces travaux font l'objet de la thèse de J. Steinbrunn (début en janvier 2007 - co encadrement à 70 %). Les objectifs de cette thèse sont multiples, à savoir 1) la mise en œuvre d'un capteur adéquat, 2) le développement d'une architecture électronique miniaturisée optimisée en termes d'énergie, 3) la recherche de nouveaux algorithmes pour le traitement du signal et 4) l'étude des applications potentielles pour ce produit.

Le LE2I n'étant pas partenaire du consortium original STELLA, un contrat spécifique entre URGO et notre laboratoire a été établi afin d'officialiser notre rôle dans ce projet européen. Ce contrat est d'un montant de 40 k€ et correspond au contrat d'encadrement de la thèse de J. Steinbrunn.

3.6 Publications jointes

J. Sérot, **D. Ginhac**. *Skeletons for parallel image processing: an overview of the SKiPPER project*, *Parallel Computing*, 28(12) : 1785-1808, 2002.

F. Yang, M. Paindavoine, **D. Ginhac**, J. Dubois. *Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques*. *Traitement du Signal*, 22 (5) : 549-562, 2005.

Skeletons for parallel image processing: an overview of the SKIPPER project

Jocelyn Sérot ^{a,*}, Dominique Ginhac ^b

^a *LASMEA, UMR 6602 CNRS, University Blaise Pascal de Clermont-Ferrand,
Campus des Cézeaux, F-63177 Aubière, France*

^b *LE2I, FRE 2309 CNRS, University of Burgundy, F-21078 Dijon, France*

Received 18 February 2002; received in revised form 22 May 2002; accepted 17 June 2002

Abstract

This paper is a general overview of the SKIPPER project, run at Blaise Pascal University between 1996 and 2002. The main goal of the SKIPPER project was to demonstrate the applicability of *skeleton-based* parallel programming techniques to the fast prototyping of reactive vision applications. This project has produced several versions of a full-fledged integrated parallel programming environment (PPE). These PPEs have been used to implement realistic vision applications, such as road following or vehicle tracking for assisted driving, on embedded parallel platforms embarked on semi-autonomous vehicles. All versions of SKIPPER share a common front-end and repertoire of skeletons—presented in previous papers—but differ in the techniques used for implementing skeletons. This paper focuses on these implementation issues, by making a comparative survey, according to a set of four criteria (efficiency, expressivity, portability, predictability), of these implementation techniques. It also gives an account of the lessons we have learned, both when dealing with these implementation issues and when using the resulting tools for prototyping vision applications.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Parallelism; Skeleton; Computer vision; Fast prototyping; Data-flow

1. Introduction

The general context of the SKIPPER project is the development of realistic vision applications for embedded platforms. These applications may be found for instance

* Corresponding author.

E-mail addresses: jocelyn.serot@lasmea.univ-bpclermont.fr (J. Sérot), dginhac@u-bourgogne.fr (D. Ginhac).

in remote inspecting robots or vehicles equipped with assisted-driving systems, as presented in [16,25,27]. Although relying on algorithms and programming paradigms encountered in the mainstream of computer vision, these applications raise two specific issues. First, they implement *reactive systems*, operating “on the fly” on digital *streams* of images. This means that they must be able to absorb input data and output results at a minimum frequency and produce responses within a maximal latency. For assisted-driving applications, for instance, the typical frequencies are in the range of 10–30 frame/s and the maximal latency rarely exceeds 50 ms. Second, they must meet stringent operational constraints in terms of volume or power consumption, which often rules out implementations based upon stock-hardware.

These requirements can be met by resorting to embedded parallel machines. The TRANSVISION [15,19] platforms, built between 1992 and 1998 at LASMEA, are examples of this approach. These MIMD architectures, built upon Transputer and Alpha processors, could deliver significant computing power with a limited volume and power-consumption, and provided built-in facilities for video i/o. More recently, we have been investigating the feasibility of a embedded Beowulf-style cluster built upon PowerPC G4 processors and using the IEEE-1394 interface for fast video i/o.

But relying on parallel machines places severe strains on programmers: in the absence of high-level parallel programming models and environments, they have to explicitly take into account every aspect of parallelism such as task partitioning and mapping, data distribution, communication scheduling or load-balancing. Having to deal with these low-level details results in long, tedious and error-prone development cycles—especially when the persons in charge of developing the algorithms are image processing and not parallel programming specialists—thus hindering a true experimental approach. For *reactive* applications, the problem is reinforced by the fact that the need to evaluate the *dynamic* properties of the algorithm at realistic frame-rate effectively rules out any prototyping phase solely based upon off-line, sequential simulation on stock hardware. Parallel programming at a low level of abstraction also limits code reusability and portability.

The SKIPPER project was developed in response to the aforementioned problems. Basically, its goal was to “capture”—in an efficient and portable way—the expertise gained by programmers when implementing reactive vision applications using low level parallel constructs, to make it readily available to algorithmicians and image processing specialists. This project has been run at LASMEA from 1996 to now and has produced four skeleton-based parallel programming environments: SKIPPER-O, SKIPPER-I, SKIPPER-II and SKIPPER-D. These results have been described in previous papers [9,16,24,25,27] but in a rather separate manner. The goal of this paper is to provide a global presentation of these separate accounts and to provide a comparative assessment of the successive versions of SKIPPER. It explains in particular why these versions, which share a common formalism for *specifying* parallel programs, differ significantly in the techniques used for *implementing* skeletons. It is organized as follows. Section 2 is a brief recall of SKIPPER principles and general architecture. Section 3 presents the successive versions of SKIPPER and the criteria used to assess them. Section 4 summarizes the results of this assessment. Section 5 is a brief review of related work and Section 6 concludes this paper and outlines directions for future work.

2. SKIPPER generic architecture

The SKIPPER programming methodology is based upon the concept of *algorithmic skeletons* [7,8]. Skeletons are high-level program constructs that abstract common patterns of parallel computation in a parametric way. With this approach, the structure of a parallel application is expressed only as a combination of the skeletons provided. The repertoire of skeletons acts as a sort of “parallel toolbox” from which parallel programs can be built with a minimal concern for low-level details. Fig. 1 gives the general software architecture of the SKIPPER parallel programming environments. The application programmer provides a skeletal, structured description of the parallel program, the set of application-specific sequential functions used to instantiate the skeletons and a description of the target architecture. The SKIPPER suite of tools turns these descriptions into executable parallel code. The main software components are: a library of skeletons, a compile-time system (CTS) for generating the parallel C code and a run-time system (RTS) providing support for executing this parallel code on the target platform. The CTS can be further decomposed into a front-end, whose goal is to generate a target-independent intermediate representation of the parallel program, and a back-end system, in charge of mapping this intermediate representation onto the target architecture.

2.1. The skeleton library

The SKIPPER library of skeletons was built “bottom-up”, from a careful analysis of a large corpus of existing low-to-mid level vision applications hand-coded in parallel C [27]. It consists of three skeletons:¹

- the **SCM** (split–compute–merge) skeleton is devoted to fixed data-parallelism, for instance to “geometric” processing of iconic data, in which the input image is split into a fixed number of subimages, each subimage being processed independently, and the final result is obtained by merging the results computed on subimages;
- the **DF** (data-farming) skeleton handles variable data-parallelism, i.e., situations in which the number of data to process is not known at compile time;
- the **TF** skeleton is a generalisation of the **DF** skeleton, in which the processing of one data item may recursively generate new items to be processed. It is generally used to implement *divide-and-conquer* strategies.

Each skeleton comes with two semantics: a *declarative semantics*, which gives its “meaning” to the application programmer in an implicitly parallel manner, i.e.,

¹ A fourth skeleton (**ITERMEM**) is described in previous papers. This skeleton does not actually encapsulate parallel behavior, but is used whenever the *iterative* nature of the real-time vision algorithms—i.e., the fact that they do not process single images but continuous *streams* of images—has to be made explicit. It will not be discussed here.

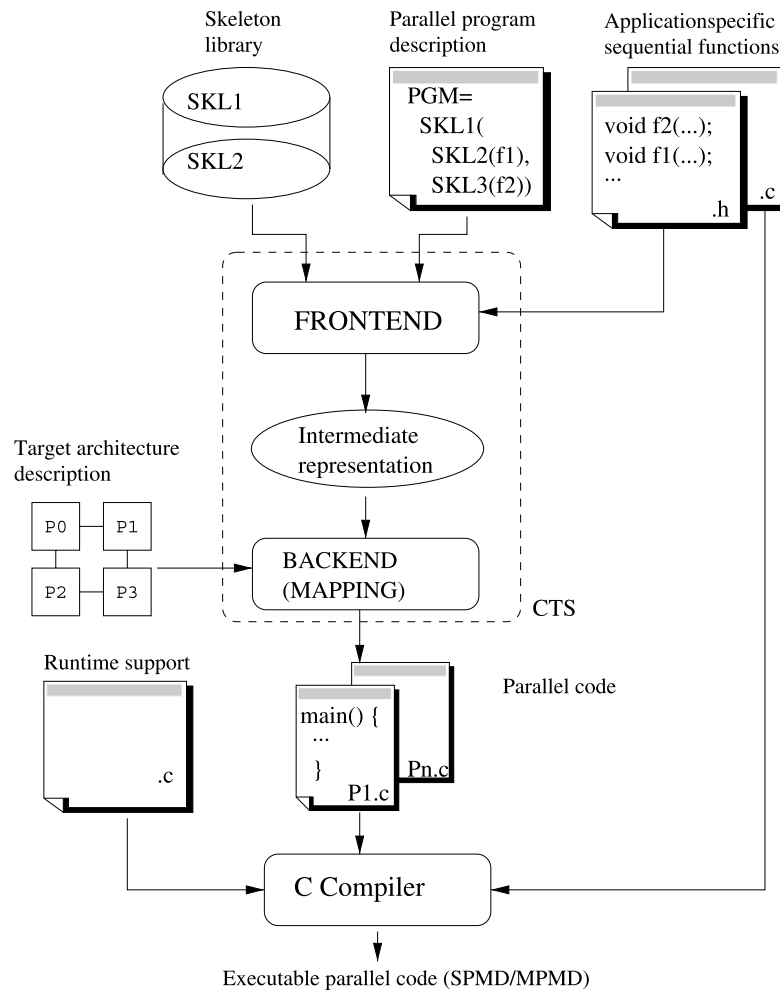


Fig. 1. SKIPPER general software architecture.

without any reference to an underlying execution model, and an *operational semantics* which provides an explicitly parallel description of the skeleton.

The *declarative semantics* of each skeleton is shared by all SKIPPER versions. It is conveyed using the CAML language, using higher-order polymorphic functions. The corresponding definitions are given in Fig. 2. Potential (implicit) parallelism arises from the use of the `map` and `foldll` higher-order functions.²

² These higher-order functions apply a function and iterate a (commutative, associative) binary operator over a list of elements, respectively, $map\ f\ [x_1, x_2, \dots, x_n] = [f(x_1), f(x_2), \dots, f(x_n)]$ and $foldll\ \oplus\ [x_1, x_2, \dots, x_n] = x_1 \oplus x_2 \oplus \dots \oplus x_n$.

J. Sérot, D. Ginjac / Parallel Computing 28 (2002) 1685–1708

1689

```

let scm split comp merge x = merge (map comp (split x))

let df comp acc xs = foldl1 acc (map comp xs)

let rec tf triv solve divide comb xs =
  let f x =
    if (triv x) then solve x
    else tf triv solve divide comb (divide x)
  in foldl1 comb (map f xs)

```

Fig. 2. Declarative semantics of SKIPPER skeletons.

The *operational semantics* of a skeleton varies according to the nature of the intermediate representation used by the CTS. In the successive versions of SKIPPER, we have been experimenting with four types of intermediate representation: static data-flow graphs (for SKIPPER-O), parametric process networks (for SKIPPER-I), hierarchical task graphs (for SKIPPER-II) and tagged-token data-flow graphs (for SKIPPER-D). These representations will be discussed in turn in Sections 3.1–3.4.

SKIPPER also relies on the CAML language for *expressing the parallel (skeletal) structure of the programs*. The programmer indicates which skeletons are used, in what order and, for each skeleton, the sequential functions and/or numeric values given as parameters.³ This is illustrated in Fig. 3, with a small program making use of the scm skeletons to process an image. Here `get`, `splitrow`, `filt`, `conv`, `mergerow` and `disp` are the application specific, sequential functions: `splitrow` decomposes an image into horizontal sub-images, `filt` and `conv` respectively apply a median filter and a convolution mask on a (sub)image and `mergerow` concatenates subimages into a single one. The `get` function retrieves the next image from the video input stream and the `disp` function displays the result image on the screen. `◦` is the CAML infix operator denoting function composition.

In the previous example, the *application-specific sequential functions* are written in C. This point is of great practical importance since we do not want application programmers to recode their algorithms from scratch (and especially in CAML). The prototype of the functions used in the previous example are given in Fig. 4.

The role of the *back-end* in the CTS is to map the intermediate representation of the parallel program (data-flow graph, process network, etc.) onto the target architecture. For an MIMD target with distributed memory, for example, this involves finding a distribution of the operations/processes on the processors and a scheduling of the communications on the provided medium (bus, point-to-point links, etc.). The distribution and the scheduling can be *static*—i.e., done at compile time—or *dynamic*—i.e., postponed until run-time. Both approaches require some kind of

³ The need to pass and return functions and values from various types to/from other functions explains the choice of a higher-order, polymorphic language, such as CAML, for specifying skeletons and skeletal programs in SKIPPER.

1690

J. Sérot, D. Ginjac / Parallel Computing 28 (2002) 1685–1708

```

let img = get ();;
let res = scm splitrow (conv o filt) mergerow img;;
let main = disp res;;

```

Fig. 3. A sample skeletal program.

```

void get(/*out*/ img *im);
void splitrow(/*in*/ img *im, /*out*/ imgList *ims);
void filt(/*in*/ img *im1, /*out*/ img *im2);
void conv(/*in*/ img *im1, /*out*/ img *im2);
void mergerow(/*in*/ imgList *ims, /*out*/ img *im);
void disp(/*in*/ img *im);

```

Fig. 4. Prototype of sequential functions.

RTS. For static approaches, the RTS can take the form of a reduced set of *primitives*, providing mechanisms for synchronizing threads of computations and exchanging messages between processors.⁴ For dynamic approaches, it must include more sophisticated mechanisms for scheduling threads and/or process and dynamically managing communication buffers, etc. For this reason, static approaches generally lead to better (and more predictable) performances. But, as evidenced in Section 3.1, they may lack expressivity. Dynamic approaches, on the other hand, do not suffer from this limitation but this is generally obtained at the expense of reduced performances and predictability (as evidenced in Sections 3.3 and 3.4). The SKIPPER project has covered a wide spectrum of distribution and scheduling techniques, ranging from entirely static to fully dynamic, making it possible to assess the relative merits and flaws of these techniques in the context of a skeleton-based methodology.

Depending on the distribution and scheduling technique used in the back-end, the *parallel code* takes the form of a set of either MPMD (one distinct program per processor) or SPMD (the same program for all processors) programs. These programs are linked with the code of the RTS and the definition of the application specific sequential functions to produce the executable parallel code.

3. Comparative assessment

All versions of SKIPPER share the general architecture described in the previous section. They differ in the type of *intermediate representation* produced by the front-end and in the distribution/scheduling technique used by the back-end. The

⁴ These primitives can use architecture-specific instructions or portable OS-level facilities such as MPI for example.

consequences of these implementation choices will be analysed in turn in the following sections according to four criteria:

Efficiency. Efficiency will be assessed either by observing the obtained speedups on realistic or synthetic⁵ test applications or by comparing the run-time performances of the “skeletalized” application to those obtained with a hand-crafted parallel version using C+MPI.⁶

Portability. Here, we mean the ability to port a given version of the SKIPPER suite of tools onto a new parallel platform. Given the layered software architecture of SKIPPER, these portability issues mainly concern the RTS: the smaller (and the simpler) this RTS, the more portable the corresponding SKIPPER version will be. Moreover, in our context (embedded vision applications), we must eventually consider the possibility of targeting architectures with *little or no OS-level support*,⁷ such as machines built from specialized or digital signal processors (DSPs).

Performance predictability. This refers to the possibility to predict the run-time behavior of an application (its latency and frequency for example) without actually running it on the target parallel platform, on the basis of application-specific parameters (such as the duration of the sequential functions) and architecture-specific parameters (such as communication latency). Performance prediction is generally carried out using analytical *cost models* and estimated (typical) durations (as in SKIPPER-I or in most existing skeleton-based PPEs). In the context of *reactive* applications, one may need a more *deterministic* approach, in which *strict* temporal bounds can be computed at compile-time.

Expressivity. This refers to the ability to implement an application expressed as an arbitrary combination of skeletons. In practice, experience has shown that the critical point here is whether the intermediate representation supports *nesting* or not, i.e., the ability for a skeleton to take another skeleton as an argument. Although it is still unclear whether realistic applications really need nesting (see [8]), its support has always been perceived as a challenge by skeleton implementors.

3.1. Static data-flow—SKIPPER-0

The first version of SKIPPER used an intermediate representation of skeletal programs as static data-flow graphs (DFG). Skeletons were viewed as parameterisable data-flow *graph patterns*, encoded directly in CAML as higher-order functions thanks to a tool called CAMLFLOW. An in-depth description of CAMLFLOW (which is based upon *abstract interpretation*) can be found in [26]. The mapping of the DFG onto the target architecture was handled by a third party software called SYNDEX [17]. Both

⁵ By “synthetic”, we mean an application written solely for benchmarking purposes, as opposed to an application corresponding to a “real” algorithm. Synthetic applications allow an easier adjustment of key parameters such as ratio between communication and computation costs or data distribution.

⁶ The first versions of SKIPPER were developed for the TRANSVISION platforms, for which we did not have a MPI layer at our disposal.

⁷ By OS-level support, we mean the facilities typically provided by multi-tasking, Unix-like operating systems: multi-processing, inter-process communication and synchronization, virtual memory, etc.

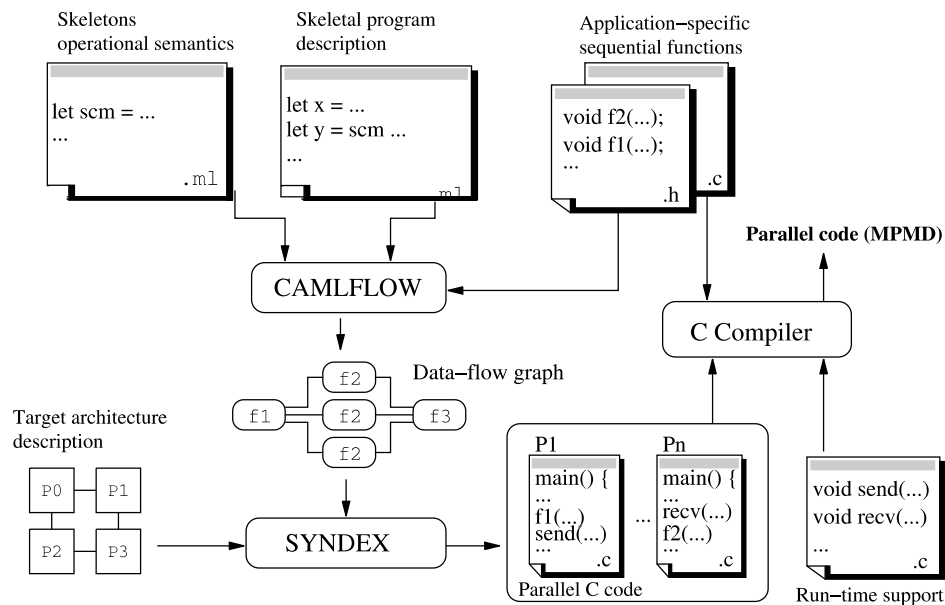


Fig. 5. Compilation path in SKIPPER-O.

the distribution of the operations (the sequential functions associated with nodes) onto the processors and the scheduling of communications onto inter-processor channels were *static*. The result of SYNDEX distribution and scheduling is a set of processor-independent programs,⁸ one per processor, built from a small kernel of primitives. These primitives offers support for static thread creation, thread communication and call of user-supplied sequential functions. The final parallel C code was obtained by simply providing definitions for these kernel primitives according to the available hardware facilities.⁹ The complete compilation path for SKIPPER-O is illustrated in Fig. 5.

The SKIPPER-O environment is further illustrated in Fig. 6, which shows the SYNDEX session used for implementing the program given in Fig. 3. The left window shows the corresponding data-flow graph along with the target architecture (four ring-interconnected C40 processors here). The right window illustrates the static mapping of operations onto processors computed by SYNDEX (oval boxes represent operations, diagonal lines communications and columns processors).

Assessment. With SKIPPER-O, the overhead of the run-time system was virtually zero, since all decisions regarding distribution and scheduling were taken at compile-time. This resulted, at least for programs relying on mid and coarse grain fixed data-

⁸ *m4* macro-code.

⁹ For the TRANSVISION platforms, for example, the primitives used the built-in process switching and channel *i/o* of the *Transputer*. But the Kernel can be easily ported to other systems, for instance Unix/Linux-based multi-processors communicating through TCP/IP sockets or MPI.

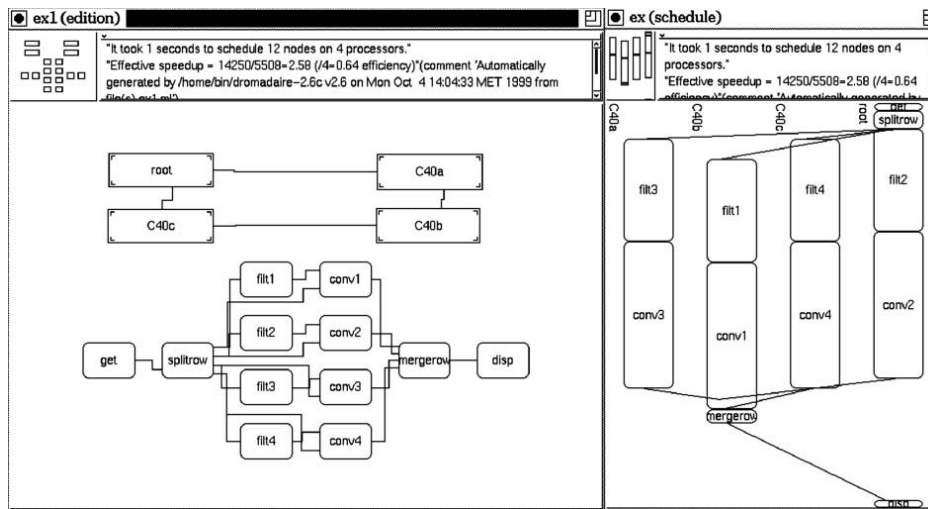


Fig. 6. A SYNDEX session in SKIPPER-O.

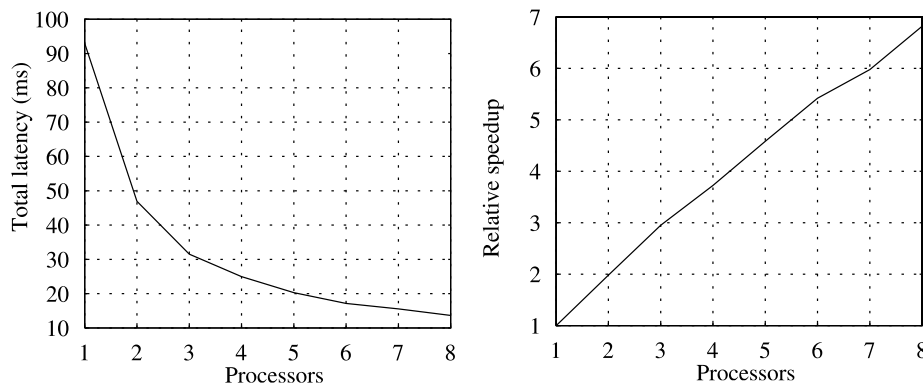


Fig. 7. SKIPPER-O performance figures (scm skeleton).

parallelism, in high efficiency. This is evidenced in Fig. 7 which gives the total latency and relative speedup for an application computing histograms of gray level images and implemented on a multi-transputer machine.¹⁰

For the same reason, predictability was very good, with measured performances never differing from those predicted by more than 5%. Performance prediction in SKIPPER-O actually required two passes: in the first pass, rough estimates of the durations of the sequential functions were given to SYNDEX, which generated a first, sub-optimal, parallel program but with automatic profiling instructions inserted in

¹⁰ T800, 25 MHz processors, 10 Mb/s point-to-point links. At the time SKIPPER-O was designed, this platform was the only one available.

it. This program could then be run on typical data to extract the real durations. These durations were used in turn to obtain the final program by means of a mapping and scheduling heuristic based upon minimization of the total latency. One could also use upper bounds for function durations in order to predict worst case behavior, in order to satisfy hard real time constraints for instance.¹¹ Finally, portability was also good: because the output macro-code was built on a small set of kernel primitives, re-targeting an application on an architecture built from a new processor type only required (re)writing this set of kernel primitives. This proved to be a straightforward task for the platform we had to deal with.¹²

The main problem with *SKIPPER-O* was expressivity. Indeed, giving an operational semantics to the *DF* and *TF* skeletons in terms of static DFG was problematic. Consider the *DF* skeleton, for instance. This skeleton is used to apply a function to a list of data items when the size of the list is unknown and/or the time to process one item can vary significantly.¹³ In this case, a static allocation of the operations (items) to processors is not always possible and would result, anyway, in an uneven work-load between processors (which in turn results in a poor efficiency). The classical solution is therefore to give the operational semantics of the *DF* skeleton as a process network and to rely on a *farming* protocol to ensure load-balancing: a *master* process dynamically doles out items to a pool of *worker* processes and collects results back, on a “first done, first served” basis. This model, however, cannot be implemented using a static mechanism, in which all communications must be scheduled at compile-time.

3.2. Template-based implementation—*SKIPPER-I*

In *SKIPPER-I*, the limitations of *SKIPPER-O* were overcome by relying on *process networks* for the intermediate representation of skeletal programs and on *implementation templates* for skeletons. This approach is the most widely used for existing skeleton-based PPEs (like those cited in Section 5). Implementation templates are “*known parametric parallel process networks that efficiently implement a skeleton on a particular parallel target architecture at hand*” [13]. They generally take the form of process graphs that can be parameterized in the parallelism degree (the number of *worker* nodes for instance) and the sequential function(s) associated with each node. The intermediate representation of the application as a process network is then obtained by *instantiating* the skeleton templates.¹⁴ The most often claimed advantage of template-based approaches is that, being written once and for all for a given

¹¹ To our knowledge, *SKIPPER-O* is the only realization of a skeleton-based PPE capable of handling such hard real-time timing constraints.

¹² The kernel definition for the Transputer processor was less than 300 lines of m4 code. Kernels have been written for several well-know DSPs and also for clusters of Unix machines running TCP/IP communication layers.

¹³ This situation is frequent in reactive vision, where a varying number of *regions of interest*, of varying size, often have to be processed in each frame.

¹⁴ This instantiation is done on the basis of the provided application-specific sequential functions. It can also take architectural parameters into account, to adjust the declared parallelism degree of the skeleton to the one actually offered by the architecture for instance.

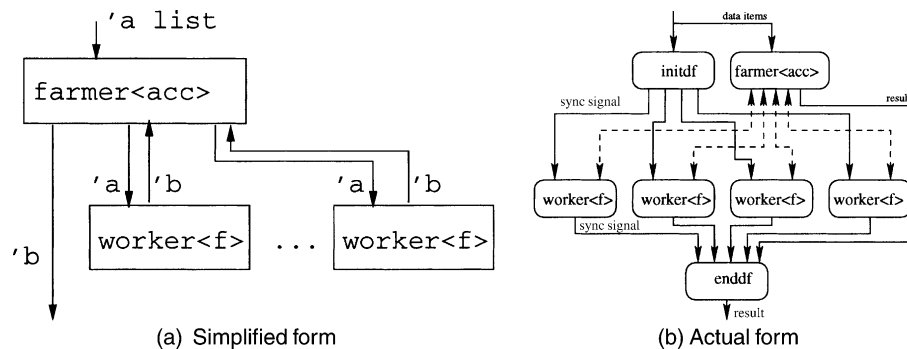


Fig. 8. The parametric process network of the DF skeleton.

architecture, they can be carefully hand-crafted to make them both reliable and highly efficient.

The CAMLFLOW front-end of SKIPPER was therefore modified to produce process networks out of CAML skeletal descriptions instead of data-flow graphs. For this purpose, each skeleton was described (in CAML, again) as a *parametric process network*.¹⁵ Fig. 8a gives a parametric process network (PPN) for the DF skeleton.¹⁶ This graph is parametric in the number of *worker* nodes, in the type of data items exchanged between nodes (denoted with type variables 'a...b) and in the sequential functions run on the nodes *farmer* and *worker* (this “parameterization” being denoted with brackets).

The behavior of the *farmer* and *worker* processes was stored separately as a *parametric process template* (PPT). A PPT is a piece of sequential code whose behavior can be specialized by providing numeric parameters, data types and/or functional parameters.¹⁷

The compilation path in SKIPPER-I was similar to the one depicted in Fig. 5 for SKIPPER-O, except that the CAMLFLOW front-end produced an intermediate representation in the form of a *process network* instead of a data-flow graph.¹⁸ The back-end tasks were still handled by the SYNDEX software. This may seem contradictory since, as stated in Section 3.1, SYNDEX can only handle static data flow graphs and not process graphs. The solution adopted in SKIPPER-I was in fact a hybrid solution: process graphs were “viewed” by SYNDEX as data-flow graphs and mapped/scheduled as data-flow graphs. In particular, SYNDEX only scheduled (at compile-time) “static” communications (the ones that mark the start and the end of a farming skeleton for instance). The “dynamic” communications (the ones occurring between the master and the workers during the activity of a farming skeleton) were handled by ad hoc processes

¹⁵ To facilitate cross-referencing, we use here the terms introduced in [27]. Conceptually, *parametric process networks* are *implementation templates*.

¹⁶ This graph is a simplified version of the PPN actually in SKIPPER-I, which appears in Fig. 8b (see later).

¹⁷ Specialization is carried out using macro substitution.

¹⁸ A detailed presentation of SKIPPER-I can be found in [27].

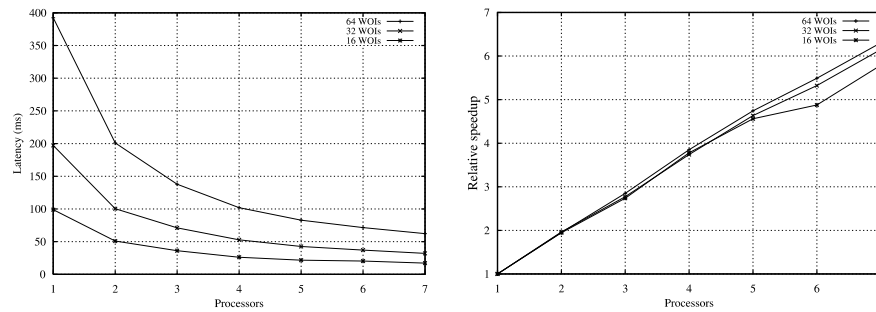


Fig. 9. SKIPPER-I performance figures.

“hidden” in the data-flow nodes. This technique—which amounts to tolerating “critical sections” of dynamically scheduled code within a globally statically scheduled application—is set out in detail in [15]. It is illustrated in Fig. 8b, where “static” communications are denoted with plain lines and “dynamic” ones with dashed lines.

Assessment. The SKIPPER-I version was the first to support the complete set of skeletons described in Section 2 and has been used to implement several realistic reactive vision applications, most noticeably those described in [16] (segmentation by connected component labeling), [25] (vehicle tracking) and [27] (road tracking). Thanks to the SYNDEX back-end, efficiency remained high (with an overhead never exceeding 25% compared with hand-written parallel C code for the applications implemented). For applications making use only of “static” skeletons (such as SCM), this overhead was almost zero, as for SKIPPER-O. Fig. 9 gives the measured performances on a synthetic application, which consists in applying a dummy processing function to a list of windows of interests (WOIs) in an image. Tests were performed on a cluster of eight Sun Ultra-5 workstations with a switched Fast Ethernet connection. Total latency and relative speedup are given for three values (16, 32, 64) of NBW , the number of WOIs processed in each image (the greater the number, the more the dynamic farming capabilities of the DF skeleton are solicited).

Predictability of performances relied on a set of analytical cost models [15] that provided accuracy in the range of 10–20%. But, unlike SKIPPER-O, strict timing bounds could not always be exhibited: this is clearly the price to pay for accepting dynamically scheduled skeletons such as DF.

The main problem with SKIPPER-I lay in the hybrid nature of the intermediate representation. Because dynamic communications were transparent to SYNDEX, the routing of these communications between distant processors had to be handled explicitly by auxiliary processes (whereas it is done automatically by SYNDEX for static communications). It turned out that including the description of these auxiliary processes in the SYNDEX kernel without compromising too much efficiency was a difficult task. To make the problem tractable, the SKIPPER-I compilation process therefore made assumptions on the topology of the target architecture (it had to be ring-interconnected). These assumptions, along with the increased size and complexity of the SYNDEX kernel, lowered the portability of the applications developed

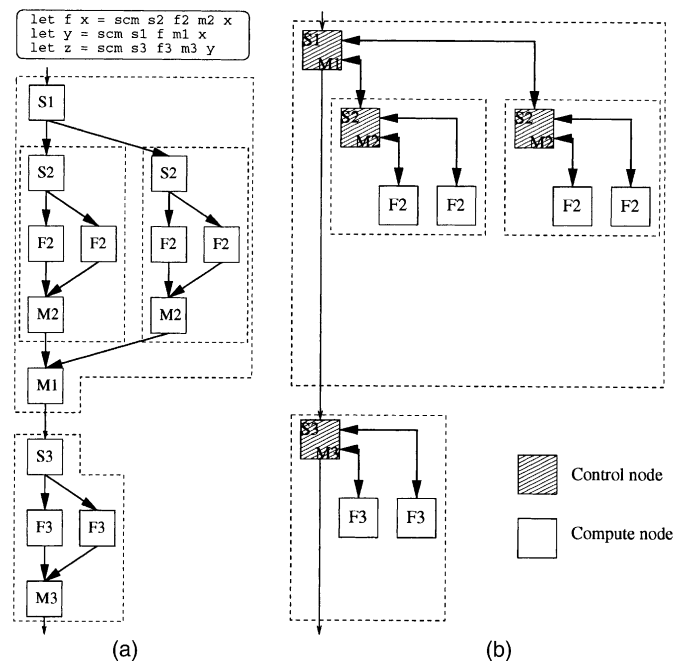


Fig. 10. Intermediate representation of skeletal programs within SKIPPER-II: (a) original program (b) intermediate representation as a tree of TF-II.

with SKIPPER-I (compared to SKIPPER-O). Finally, the hybrid intermediate representation of SKIPPER-I implicitly relied on a “flat” execution model and was definitely not suited for implementing nested skeletons.

3.3. Hierarchical task graphs—SKIPPER-2

In SKIPPER-II, we turned to a homogeneous intermediate representation of programs as *hierarchical task graphs*. This design choice was made in order to overcome the difficulties raised by hybrid representations (such as that used in SKIPPER-I) and to solve the problem of skeleton nesting in a systematic way. To do so, all skeletons of the SKIPPER repertoire were viewed, *at the implementation level*, as specialized *instances* of a generic skeleton, called TF-II.¹⁹ The operational semantics of the TF-II skeleton is basically the one of a task farming skeleton: a *master* process doles out tasks to a pool of *worker* (slave) processes, but here a task can be either a sequential function to be computed or another skeleton to be run. The intermediate representation takes the form of a tree of TF-II skeletons. It is computed by another version of the CAMLFLOW front-end, which uses alternate definitions of the SCM, DF and TF skeletons as specialized calls to the TF-II higher-order function. This step is illustrated in Fig. 10 where a program making using of three SCM skeletons (two of them

¹⁹ For Task Farming, version II.

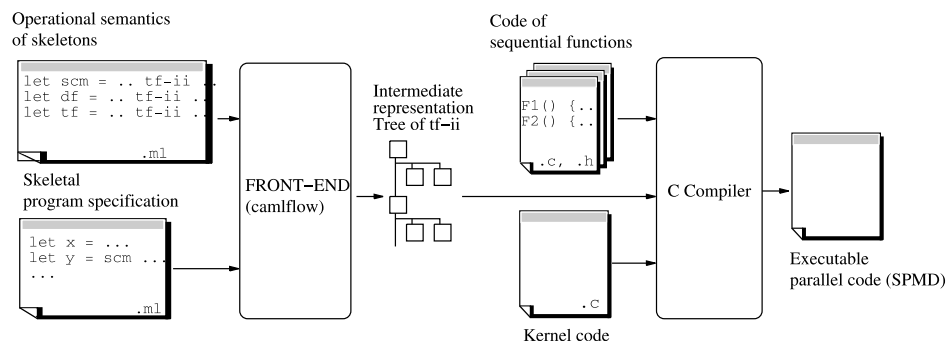


Fig. 11. Compilation path in the SKIPPER-II parallel programming environment.

nested) is turned into a tree of TF-II descriptors. In this tree, nodes correspond to skeleton control processes and leaves to sequential functions (a detailed presentation of the SKIPPER-II system can be found in [9] or in the forthcoming [10]).

Interpretation of the intermediate representation within SKIPPER-II is done at run-time by a specialized program (the “kernel”) running in SPMD mode on all processors (see Fig. 11). This kernel—written in C—provides dynamic support for three kind of services: concurrent execution of *master* and *worker* processes, inter-process communication (using a subset of MPI-conformant routines) and handling of shared resources such as the worker pool. Whenever a skeleton needs to be run, either as a “top-level” node (on the spine of the TF-II tree) or as a nested instance, a new copy of the kernel is launched on the local processor. This copy acts as the *master* of the skeleton. It uses the free resources (idle processors) to allocates new workers. When all resources are busy, the execution of *worker* processes is sequentialized on the processor running the *master* process.

Assessment. SKIPPER-II was the first version to use a fully *dynamic* implementation mechanism for skeleton-based programs. This has several advantages. First, in terms of expressivity, since arbitrary nesting of skeletons is naturally supported. The introduction of new skeletons is also facilitated, since it only requires giving their translation in terms of TF-II. Portability remains acceptable since porting applications to new architectures only requires the porting of the run-time kernel. This, in practice, turned out to be a relatively straightforward task. The approach used in SKIPPER-II also provides automatic load-balancing, since all mapping and scheduling decisions are taken at run-time, depending on the available physical resources. In the same vein, sequential emulation is obtained “for free” by just running the program on a single processor. The major problems with SKIPPER-II are efficiency and predictability.

As regards efficiency, several experiments [9,10] have shown that the dynamic process distribution used in SKIPPER-II may entail a significant performance penalty. This has been proved to be true specially for

- applications exhibiting a low compute vs communication ratio (compared to a C+MPI implementation, the SKIPPER-II kernel performs more communications, for exchanging data between inner and outer *masters* in particular);

- applications relying on fine-grain parallelism (because shared resources are handled in a centralized manner in **SKIPPER-II**, each worker allocation requires a pair of communications to a particular processor; this becomes a bottleneck when the grain decreases, i.e., when the number of *worker* processes increases);
- platforms not supporting multi-processing at the processor level (in this situation, some processors may end up running only one *master* process, with a very small load factor, leading to poor global efficiency.²⁰).

Furthermore, the fully dynamic approach used in **SKIPPER-II** makes performance prediction very difficult because, in this model, processors can switch from *master* to *worker* behavior depending only on actual input data (there is no “fixed” mapping for dynamic skeletons as in **SKIPPER-I**). Even the interpretation of execution profiles, generated by an instrumented version of the kernel, turned out to be far from trivial. This point raises a pragmatic problem within a programming methodology based upon experimental validation of solutions: here, one not only needs to obtain a running prototype quickly, but also to be able to understand why a given prototype exhibits poor run-time performances.²¹

3.4. Dynamic data-flow—**SKIPPER-D**

The implementation of **SKIPPER-D** started in 2000 in response to the problems identified with **SKIPPER-II** version. The design of **SKIPPER-D** was inspired by results obtained by Danelutto on the Macro Data-Flow (MDF) execution model for skeletons [12]. This model is very similar to the one used in **SKIPPER-O**: skeleton-based parallel programs are compiled down to data-flow graphs, in which nodes correspond to sequential functions (“macro-instructions”) and arcs to data dependencies between these functions. But, like Danelutto and unlike **SKIPPER-O**, a *dynamic* interpretation mechanism is used for executing these graphs. This mechanism relies on a set of distributed data-flow interpreters, running in SPMD mode on all processors of the target architecture. **SKIPPER-D** extends the MDF execution model proposed by Danelutto in order to implement arbitrary nested data or task farm skeletons. For this purpose, the **SKIPPER-D** runtime relies on the *tagged-token* data-flow interpretation technique [1,2]. This technique basically allows several concurrent activations of a single sequential node to overlap in time; it associates a unique *tag* with each activation, and each data token also carries a tag that specifies the particular activation to which it belongs. Skeletons involving run-time bounded iterations and/or recursion, and nested in an arbitrary way, can then be represented as cyclic data-flow graphs. This is illustrated in Fig. 12 with the formulation as a tagged-token MDF graph of a program involving two nested **DF** skeletons (in this figure, tags are denoted as superscripts). The MDF graph uses a pair of special nodes called *iter* and *endf*. *Iter* accepts a list of data items and generates distinct result tokens, each

²⁰ In the multi-processing case, the processor can be shared between *master* and *worker* processes.

²¹ By contrast, the profiling facilities offered by **SKIPPER-I** (and set out in detail in [27]) were much easier to exploit.

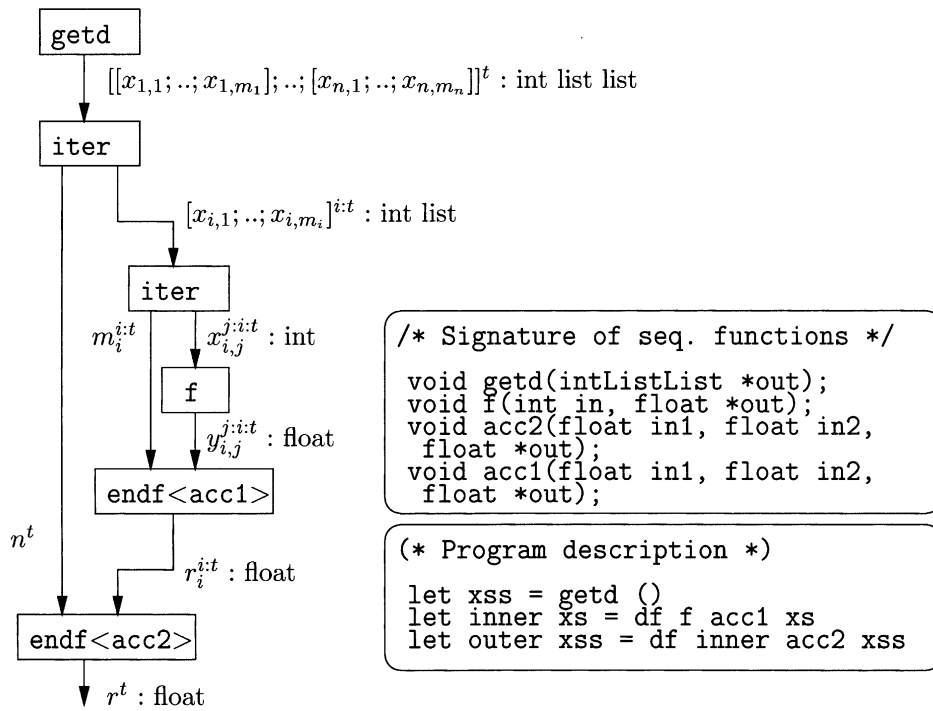


Fig. 12. Nested farm skeletons under the tagged-token MDF model.

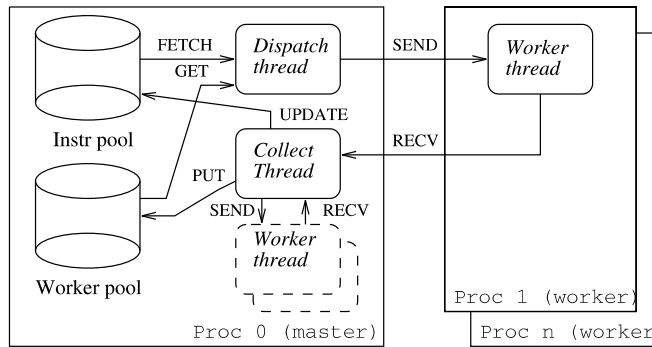


Fig. 13. The run-time system of SKIPPER-D.

carrying one data item and a distinct tag. These tokens trigger distinct firings of the subsequent nodes. The tokens resulting from these firings are collected by the `endf` node and accumulated using the `acc` sequential function. A more detailed account of this mechanism can be found in [24].

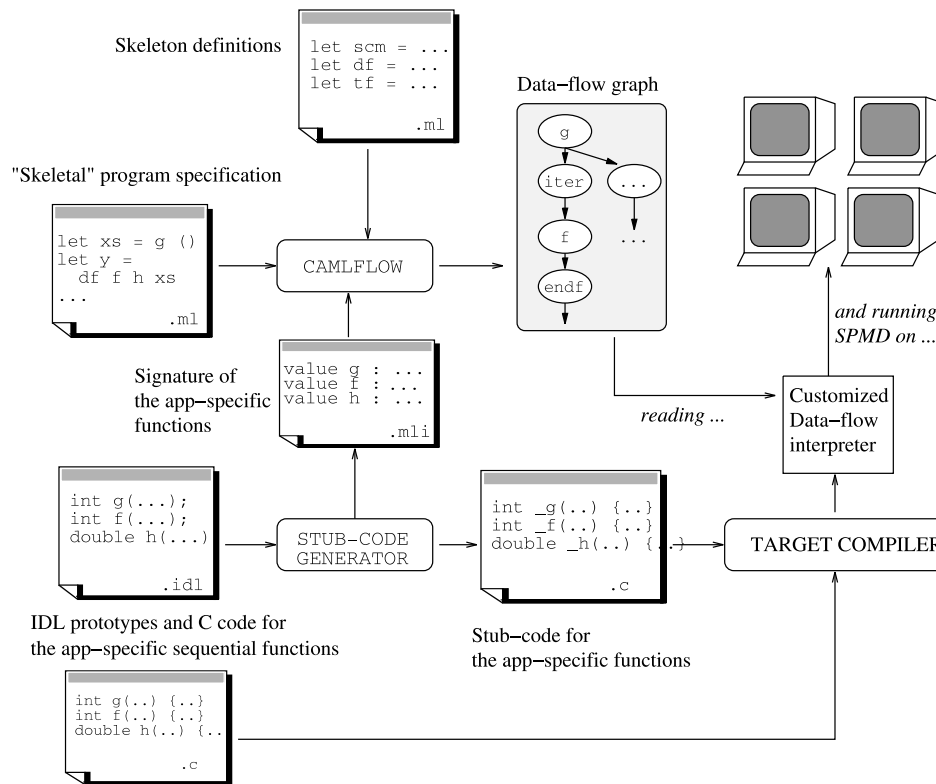


Fig. 14. The compile-time system of SKIPPER-D.

Like its predecessors, the SKIPPER-II system can be divided into a compile-time system and a run-time system. The latter implements a (centralized) tagged-token data-flow interpreter and the former produces the MDF graph for this interpreter from a high-level skeletal program specification.

The *run-time system* of SKIPPER-D is sketched in Fig. 13. Like Danelutto's system, it relies on an SPMD approach: all the processors (nodes) of the target architecture run the same program, which is the result of the compilation of the user code (C sequential functions) and the interpreter code. The interpreter itself involves several *threads* of execution: a `dispatch` thread, which fetches macro-instructions (sequential functions to be computed) from a pool of fireable instructions and sends them to the worker threads, a `collect` thread, which receives results from the worker threads and updates the instruction pool accordingly and several ²² worker threads for computing sequential functions. The `dispatch` thread fetches idle workers from a centralized pool, in which all worker threads register at initialization and which is subsequently updated by the `update` thread upon reception of results.

²² At least one per processor.

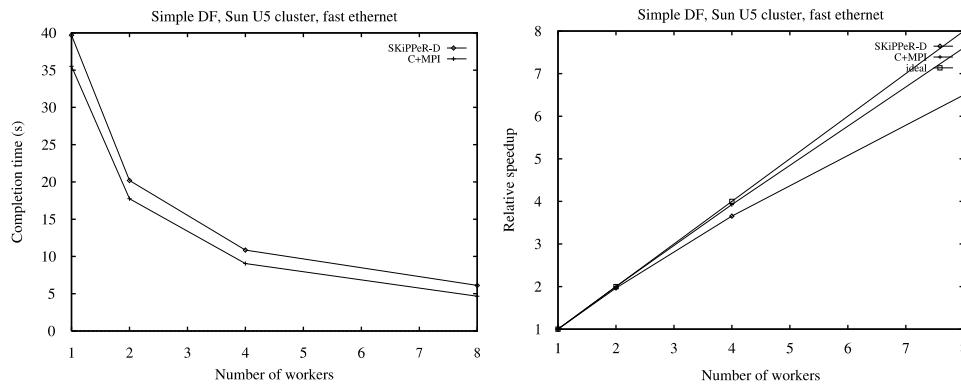


Fig. 15. SKIPPER-D performance figures (DF skeleton).

The *compile-time system* is sketched in Fig. 14. It produces the application-specific data needed to customize the run-time interpreter, i.e., the MDF representation of the program used to build the initial instruction pool and the code of the sequential C functions to be integrated with the custom run-time interpreter. The MDF graph is generated by the CAMLFLOW tool. This offers a way, as in previous versions of SKIPPER, to describe skeletons entirely in CAML as higher-order functions.

Assessment. The main contribution of SKIPPER-D is to provide an all-encompassing intermediate representation for all skeletons. This representation allows arbitrary combination (including nesting) of skeletons, thanks to the tagged-token interpretation mechanism. SKIPPER-D therefore definitely solves the expressivity problem, at least for our repertoire of skeletons. Experimental results, obtained with a prototype run-time system (written in OBJECTIVE CAML) for synthetic applications on a cluster of workstations are reported in [24] and in Fig. 15. They show good speedups and a small overhead (compared with hand-written C+MPI code). Moreover, these experiments have shown that, at least for coarse and medium-grained computation schemes, and contrary to SKIPPER-II, the mechanism used for handling nesting does not entail a significant performance penalty. Together with those reported by Danellutto in [13], these results confirm the merits of dynamic MDF execution models with respect to template-based ones. SKIPPER-D run-time performances could be further improved by integrating some optimization techniques described in [13]. These techniques include a more sophisticated management strategy of the instruction pool (based on high/low water marks), local caching of data on worker nodes and, most noticeably, a distributed interpreter implementation. The current SKIPPER-D implementation relies on a *centralized* data-flow interpreter and a rudimentary scheduling strategy for fireable instructions and is unlikely to provide comparable performances in cases of very irregular fine-grained computations. Performance predictability is clearly harder to obtain than with template-based implementation systems but does not seem an intractable problem (as in SKIPPER-II). The interpretation of profiling results is also easier than with SKIPPER-II, especially if sophisticated visualization tools such as jumpshot [28] are provided. The portability of the SKIPPER-D run-

Table 1
Comparative assessment of SKIPPER versions

	IR	D/S	Eff	Expr	Port	Pred
SKIPPER-O	SDFG	S/S	4	1	4	4
SKIPPER-I	PPN	S/S+D	3	2	1	3
SKIPPER-II	HTG	D/D	2	4	3	1
SKIPPER-D	DDFG	D(S)/D	3	4	4	2

time system on architectures built from specialized or digital signal processors is currently limited by the fact that it is written in OBJECTIVE CAML and uses *bytecode* threads. But the runtime could easily be rewritten in C for these systems.²³ In this case, threads can be emulated using hardware context switching mechanisms (as evidenced by the implementation of the SYNDEX kernel for DSPs [18]).

4. Comparative assessment

Table 1 summarizes our assessment of the successive versions of SKIPPER. In this table, we have tried to rate each version in terms of the four criteria explicated in Section 3: efficiency (*Eff*), expressivity (*Expr*), portability (*Port*) and predictability (*Pred*). For this we use a relative “score” between 1 (poor) and 4 (excellent). The first column recalls the underlying intermediate representation (IR): Synchronous Data Flow Graphs, Parametric Process Networks, Hierarchical Task Graphs and Dynamic Data Flow Graphs. The second column gives the distribution and scheduling strategy (S = static, D = dynamic).

The evolution from SKIPPER-O to SKIPPER-D can be viewed as a progressive shift—evidenced by the growing part of the run-time system in the implementation—from static approaches, offering excellent performances and predictability at the price of limited expressivity, to more dynamic approaches, trading off efficiency and/or predictability in favor of expressivity.

Fully static approaches, as in SKIPPER-O, are attractive in our context of embedded reactive applications because they minimize the resources needed to implement the algorithm and allow strict real-time bounds to be computed. But within a programming methodology dedicated to the fast prototyping of solutions—and mainly intended for algorithmicians, not parallel programming specialists—these approaches were finally found to be too restrictive. For instance, it is often possible to reformulate an existing vision algorithm—defined in terms of dynamically allocated data structures as lists or trees—so that it only uses fixed-size arrays and can be parallelized using a “static” skeleton (like SCM); but we found that it is not

²³ The current implementation is less than 500 lines of OBJECTIVE CAML code. We think that a re-implementation in C would be in the range of 1000-2000 loc, perfectly suited for processors with limited memory.

reasonable, even desirable, to do this reformulation at the prototyping level, when being able to quickly test various algorithmic and/or parallel implementation schemes turns out to be more important than obtaining optimal performances. Moreover, some algorithms are intrinsically not amenable to static implementation because the size of the input data and/or the duration of the sequential functions cannot be reliably estimated at compile time.

On the other hand, the conclusions given in Section 3.3 show that approaches relying on a fully dynamic run-time system, as *SKIPPER-II*, may raise efficiency and predictability or observability problems that conflict with our prototyping goals and/or target platforms (although these approaches might prove useful in other application domains).

In this light, we believe that the *SKIPPER-D* approach offers the best trade-off between the conflicting above-mentioned criteria. The data-flow interpretation mechanism is “mostly dynamic”²⁴ but its run-time behavior can be more easily modeled and performances do not suffer from hardly understandable performance drops due to unpredictable process allocation.²⁵

5. Related work

In the past decade, the issues related to skeleton-based parallel programming have been investigated by several research groups.²⁶ But few of them have produced full-fledged software environments that can be used to implement complex, realistic applications.

The Pisa Parallel Programming Language (P3L) project [3] is one of these projects. The P3L system includes both task parallelism (*farm, pipe*) and data parallelism (*map, reduce, scan*). Some control skeletons (*loop, seq*) allow the definition of sequential P3L modules and the iteration of skeleton compositions. Like *SKIPPER*, P3L uses C to express the sequential parts of the application but, unlike *SKIPPER*, the skeletal structure of the application is denoted using C-like syntax of data types and skeletons. The first compilers generated code for a Transputer-based Meiko CS/1 MIMD machine and for PVM running on a cluster of UNIX workstations. A more recent version [6] generates C+MPI code for PC running Linux and Fujitsu AP1000. P3L has been used to implement applications such as optical character recognition [11], ray tracing and circuit test generation.

The Heriot-Watt group has investigated the use of skeletal-based methodology for the parallelisation of vision algorithms [20,22,23]. Parallelism is extracted and exploited from programs written entirely in Standard ML. Unlike *SKIPPER* or P3L, in which skeletons are viewed as *explicit* indications to the compiler of which parallelism will be deployed and where—they take an *implicit* approach, in which skeletons

²⁴ Scheduling is done at run-time but mapping of threads to processors is done at compile-time.

²⁵ As in *SKIPPER-II*.

²⁶ See for example [29] for a comprehensive survey.

are viewed as *possible* realizations of common higher-order functions (the decision is taken by the compiler, on the basis of profiling information collected by an *instrumentation* phase). Results have been given for a Meiko CS, a Fujitsu AP-1000 and a 32-node Beowulf.

The Skil project [4] is another system relying on skeletons to provide high level parallel programming. Skil is an imperative, C-based language enhanced with a series of functional features such as higher-order functions and polymorphism. Compile-time instantiation of these features results in very efficient code (approaching the efficiency of direct C implementations). Skil focuses on data-parallelism and provides built-in types for manipulating distributed data-structures. On numerical applications such as PDE solvers [5] Skil has demonstrated good absolute performances and scalability (24 speedup for 32 processors, 87 on 128 processors) on a 1024-node Parsytec multi-processor.

6. Conclusions and future work

Several lessons were learnt when developing and using the SKIPPER system, both at the application level (from a user's point of view) and at the implementation level (from an implementor's point of view).

At the application level, the SKIPPER project has provided a convincing demonstration of the merits of skeleton-based parallel programming techniques. These conclusions are supported by realistic case studies, carried out with the help of full-fledged parallel programming environments, by people who were not parallel programming specialists in the first place. First, the “off-the-shelf” style provided by the skeleton approach effectively provides dramatic savings in development efforts. These savings make it possible to adopt a truly experimental approach in the design and implementation of applications, a key property in our context. The price to pay is a decrease in performance (compared to hand-crafted parallel code) but, for most of the realizations presented here this can be kept reasonable and was viewed as acceptable, anyway, with regard to the above-mentioned benefits. Second, within a given application domain, such as reactive embedded vision, skeletons may be viewed as a very effective way to *encapsulate* and reuse the expertise gained by skilled parallel programmers. This pragmatically solves the classic “completeness” problem often associated with skeleton-based parallel programming methodologies—namely the fact that, in theory, nothing can guarantee that a given set of skeletons will be sufficient to express every parallel algorithm: in our case, the definition of the skeleton basis was made in a *bottom-up* manner starting from an identifiable corpus of applications and/or expert knowledge and was explicitly targeted towards low to mid-level vision algorithms. Finally, the explicit, “menu-driven” approach proposed by SKIPPER could be criticised for requiring a minimum understanding of the skeleton operational semantics to be used and therefore that it cannot be used as a fully automatic parallelizing tool. Our answer, motivated by our experience in developing complex vision applications with algorithmicians, is that skeletons actually provides an effective *common ground for sharing*

expertise between image processing and parallel programming specialists: the former no longer have to deal with implementation details and the latter can treat application-specific functions as black boxes.

At the implementation level, the SKIPPER project has led us to thoroughly investigate the relative merits and flaws of *static* and *dynamic* approaches for implementing skeletons. As stated in Section 3.4, we now believe that a macro data-flow representation of skeleton-based parallel programs is probably the best choice, because it can be associated with a wide spectrum of operational semantics (from purely static synchronous to dynamic tagged-token). This conclusion is similar to that drawn by Najjar et al. in [21] who underline the “universality” of the data-flow model by exhibiting potential application domains both in the “software” domain (parallel programming on clusters of workstations for instance) and in the “hardware” domain (design of application-specific circuits for instance). In this context, we are now investigating the possibility of developing transformational rules to derive a static formulation of an algorithm automatically (using a *synchronous* data-flow execution model) from a dynamic one (based upon a *tagged-token* execution model). Our ultimate goal, motivated by our experience and needs in reactive vision applications, is to be able to specify, with the *same* skeletal formalism both “hard” (time-critical) parallel applications (built from static skeletons such as SCM) and “softer” applications (built from dynamic skeletons such as DF) which can tolerate the run-time unpredictability implied by interpreter-based implementation techniques. Recent work on *graph factorization* techniques [14] has provided some insights on how to do this in the context of compile-time bounded iterations. We are currently working to extend this scheme to generic data and task farming skeletons (the fundamental issue being: what constraints do we have to put on the tagged-token data-flow graph formulation of an algorithm—that can always be interpreted dynamically—to make it amenable to static implementation).

References

- [1] Arvind, K.P. Gostelow, The U-interpreter, *IEEE Computer* 15 (2) (1982) 42–49.
- [2] Arvind, R. Nikhil, Executing a program on the MIT tagged-token dataflow architecture, *IEEE Transactions on Computers* 39 (3) (1990) 300–318.
- [3] B. Bacci, M. Danelutto, S. Orlando, S. Pelagatti, M. Vanneschi, P³L: a structured high level programming language and its structured support, *Concurrency: Practice and Experience* 7 (3) (1995) 225–255.
- [4] G.H. Botorog, H. Kuchen, SKIL: an imperative language with algorithmic skeletons for efficient distributed programming, in: *International Symposium on High Performance Distributed Computing*, *IEEE Computer*, pp 243–252.
- [5] G.H. Botorog, High-level parallel programming and the efficient implementation of numerical algorithms. Ph.D. Thesis, RWTH-Aachen, 1998.
- [6] S. Ciarpaglini, M. Danelutto, L. Folchi, C. Manconi, S. Pelagatti, anacleto: a template-based p3l compiler, in: *Proceedings of the Seventh Parallel Computing Workshop (PCW '97)*, Australian National University, Canberra, 1997.

- [7] M. Cole, *Algorithmic Skeletons: Structured Management of Parallel Computation*, MIT Press, Cambridge, MA, 1989.
- [8] M. Cole, Algorithmic skeletons, in: G.J. Michaelson, K. Hammond (Eds.), *Research Directions in Parallel Functional Programming*, Springer-Verlag, Berlin, 1999.
- [9] R. Coudarcher, J. Sérot, J.P. Déruvin, Implementation of a Skeleton-based Parallel Programming Environment Supporting Arbitrary Nesting. in: *Proceedings of the 6th International Workshop on High-Level Parallel Programming Models and Supportive Environments*, San Francisco, Lecture Notes in Computer Science, vol. 2026, Springer, Berlin, April 2001, pp. 71–85.
- [10] R. Coudarcher, *Composition de squelettes algorithmiques: application au prototypage rapide d'applications de vision*. Ph.D. Thesis, Université Blaise Pascal Clermont-Ferrand (France), 2002, in press.
- [11] M. Danelutto, S. Pelagatti, R. Ravazzolon, A. Riaudo, Parallel OCR in P3L: a case study, in: *High Performance Computing and Networking*, vol 1067, Lecture Notes in Computer Science, Springer, Berlin, 1996, pp. 1017–1019.
- [12] M. Danelutto, Dynamic run time support for skeletons, in: *Proceedings of the ParCo99 Conference*, Delft, The Netherlands, August 1999.
- [13] M. Danelutto, Efficient run-time support for skeletons on workstation clusters, *Parallel Processing Letters* 11 (1) (2001) 41–56.
- [14] A. Dias, C. Lavarenne, M. Akil, Y. Sorel, Optimized implementation of real-time image processing algorithms on field programmable gate arrays, in: *ICSP'98 Fourth International Conference on Signal Processing*, Beijing, China, 1998.
- [15] D. Ginhac, *Prototypage rapide d'applications parallèles de vision artificielle par squelettes fonctionnels*. Ph.D. Thesis, Université Blaise Pascal Clermont-Ferrand (France), 1999.
- [16] D. Ginhac, J. Sérot, J. Déruvin, Fast prototyping of image processing applications using functional skeletons on an MIMD-DM architecture, in: *IAPR Workshop on Machine Vision and Applications*, Chiba, Japan, 1998, pp. 468–471.
- [17] T. Grandpierre, C. Lavarenne, Y. Sorel, Optimized rapid prototyping for real time embedded heterogeneous multiprocessors, in: *Proceedings of 7th International Workshop on Hardware/Software Co-Design*, Rome, 1999.
- [18] C. Lavarenne, Y. Sorel, *Modèle d'exécutif distribué temps-réel pour SynDEX* INRIA Research Report, RR-3476, 1998.
- [19] P. Legrand, R. Canals, J.P. Déruvin, Edge and region segmentation processes on the parallel vision machine Transvision, in: *Computer Architecture for Machine Perception*, New-Orleans, USA, 1993, pp. 410–420.
- [20] G.J. Michaelson, N.R. Scaife, Prototyping a parallel vision system in standard ML, *Journal of Functional Programming* 5 (3) (1995) 345–382.
- [21] W.A. Najjar, E.A. Lee, G.R. Gao, Advances in the dataflow computational model, *Parallel Computing* (25) (1999) 1907–1929.
- [22] N. Scaife, P. Bristow, G. Michaelson, P. King. Engineering a parallel compiler for SML. *Proceedings of the 10th International Workshop on Implementation of Functional Languages*, September, 1998, pp. 213–226.
- [23] N. Scaife, *A dual source parallel architecture for computer vision* Ph.D. Thesis, U. Heriot-Watt, Edinburgh, 2000.
- [24] J. Sérot, Tagged-token data-flow for skeletons, *Parallel Processing Letters* 11 (4) (2001).
- [25] J. Sérot, D. Ginhac, J. Déruvin. Skipper: a skeleton-based parallel programming environment for real-time image processing applications, in: *Proceedings of 5th International Conference on Parallel Computing Technologies*, 6–10 September, 1999, Lecture Notes in Computer Science, vol. 1662, Springer, Berlin, 1999, pp. 296–305.
- [26] J. Sérot, CamlFlow: a Caml to data-flow graph translator, in: S. Gilmore (Ed.), *Trends in Functional Programming*, Intellect, vol 2, 2001.
- [27] J. Sérot, D. Ginhac, R. Chapuis, J. Déruvin, Fast prototyping of parallel vision applications using functional skeletons, *Journal of Machine Vision and Applications* 12 (6) (2001) 271–290.

1708

J. Sérot, D. Ginjac / Parallel Computing 28 (2002) 1685–1708

- [28] O. Zaki, E. Lusk, W. Gropp, D. Swider, Toward scalable performance visualization with Jumpshot, *High Performance Computing Applications* 13 (2) (1999).
- [29] Online bibliography available at <http://hypatia.dcs.qmw.ac.uk/SEL-HPC/Articles/SkeletonArchive.html>.

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

Development of a fast panoramic face mosaicing and recognition system

F. Yang, M. Paindavoine, D. Gin hac, J. Dubois

Laboratoire Le2i, Aile de l'Ingénieur - Mirande Université de Bourgogne, BP 400 - 21011 Dijon cedex
Tél: 03 80 39 36 08, Fax: 03 80 39 59 10
fanyang@u-bourgogne.fr, paindav@u-bourgogne.fr

Manuscrit reçu le

Résumé et mots clés

Dans cet article, nous présentons quelques résultats sur le développement d'un système de mosaïquage de visages panoramiques. Notre objectif est d'étudier la faisabilité de construction de visages panoramiques en temps réel. Ceci nous a conduit tout d'abord à concevoir un système d'acquisition très simple, composé de 5 caméras standards qui réalise la prise de 5 vues simultanément sous différents angles. Puis, nous avons choisi un algorithme facilement implantable sur des systèmes embarqués. Cet algorithme est basé sur des transformations linéaires successives, pour composer un visage panoramique de 150° à partir de ces 5 vues. La méthode a été testée sur une centaine de visages. Nous avons aussi effectué une étude préliminaire sur la reconnaissance de visages panoramiques dans le but de valider notre système de mosaïquage de visages. La reconnaissance est basée sur le modèle de « visages propres ». Les résultats expérimentaux ont montré la faisabilité et la viabilité du système proposé permettant d'envisager une future implantation matérielle. Nous pensons aussi utiliser notre système de mosaïquage dans d'autres applications comme la reconstruction rapide de visages 3D et la catégorisation des expressions basée sur le mouvement.

Vision panoramique, mosaïquage d'images, reconnaissance de visages, analyse en Composantes Principales, FFT.

Abstract and key words

In this article, we present some development results of a system that performs mosaicing of panoramic faces. Our objective is to study the feasibility of panoramic face construction in real-time. This led us to conceive of a very simple acquisition system composed of 5 standard cameras and 5 face views taken simultaneously at different angles. Then, we chose an easily hardware-achievable algorithm: successive linear transformation, in order to compose a panoramic face of 150° from these 5 views. The method has been tested on hundreds of faces. In order to validate our system of panoramic face mosaicing, we also conducted a preliminary study on panoramic faces recognition, based on the «eigenfaces» method. Experimental results obtained show the feasibility and viability of our system. This allows us to envisage later a hardware implantation. We also are considering applying our system to other applications such as human expression categorization using movement estimation and fast 3D face reconstruction.

Panoramic vision, image mosaicing, face recognition, principal Component Analysis, FFT.

Remerciements

Nous remercions Monsieur Dominique Arnoult du laboratoire LE2I pour la réalisation technique du système d'acquisition et la constitution de la base de données d'images.

1. Introduction

La biométrie est actuellement un champ de recherches très actif qui concerne plusieurs disciplines comme par exemple, le traitement d'images, la reconnaissance de formes et la vision artificielle. Le principal objectif de la biométrie est d'établir des systèmes capables d'identifier les personnes en fonction des caractéristiques intrinsèques comme le visage, les empreintes digitales, l'iris, etc. La reconnaissance de visages est un sujet beaucoup étudié dans le domaine de la biométrie par le fait que les êtres humains utilisent quotidiennement l'information faciale pour se reconnaître entre eux. Différentes techniques ont été proposées pour reconnaître automatiquement les visages. Nous pouvons citer les réseaux de neurones[1][2], les visages propres [3][4], et les champs de Markov[5]. Comme le «*vendor test*» [6][7][8] le montre, la plupart de ces modèles utilisent seulement le visage de face comme le signal d'entrée. Par conséquent, beaucoup de méthodes sont sensibles aux poses et aux conditions d'éclairage. Une des pistes pour surmonter ces limites est de combiner plusieurs modalités : couleur, profondeur, surface faciale 3D ... [9][10][11][12][13].

La plupart des systèmes d'acquisition 3D utilisent des appareils professionnels comme la caméra «*traveling*» ou le scanner 3D[11][12][14]. Typiquement, ces systèmes exigent que le sujet reste immobile pendant quelques secondes pour avoir des informations 3D. Ils ne sont pas adaptés aux applications en temps réel comme la catégorisation des expressions faciales basée sur le mouvement. En plus, leur coût est souvent prohibitif pour des applications orientées grand public. Pour simplifier ces systèmes d'acquisition 3D, certains modèles de reconnaissance de visages génèrent des visages 3D à partir de la stéréovision [15]. Dans ce cas, des calculs relativement complexes sont nécessaires afin d'assurer l'auto calibrage et l'extraction des informations 3D [16]. Une autre possibilité est d'obtenir des informations 3D à partir d'un ensemble d'images de visages 2D, sans construire la structure complète de visages 3D[9][17].

Depuis une dizaine d'années, notre laboratoire travaille sur le traitement automatique d'images de visages et a obtenu des résultats pour la détection, la reconnaissance et le suivi dynamique de visages 2D [18][19][20][21]. L'un des objectifs du projet «*visages panoramiques*» est d'étendre nos modèles de traitement 2D vers des traitements d'images panoramiques (3D). Dans cet article, nous décrivons le développement d'un système simple et efficace, permettant de traiter des visages 3D en temps réel. La méthode que nous proposons construit des visages panoramiques donnant des informations sur la surface 3D. Le système d'acquisition composé de 5 caméras standards permet d'obtenir 5 vues d'un visage simultanément. Ce système

d'acquisition possède des avantages comme la simplicité de mise en oeuvre et le très faible coût.

Dans ce papier, nous présentons d'abord, le système d'acquisition pour les 5 vues. Ensuite, nous décrivons la méthode de création des visages panoramiques basée sur une transformation linéaire successive. Nous donnons aussi des résultats expérimentaux de la reconnaissance de visages panoramiques avant de conclure.

2. Présentation du système d'acquisition

Notre système d'acquisition est composé de 5 caméras USB logitech 4000 avec une résolution maximale de 640×480 pixels chacune. Ces caméras sont fixées sur une glissière circulaire réglable en hauteur pour pouvoir s'adapter à la taille du sujet (voir la figure 1 gauche). Les paramètres de chaque caméra (Format des images, Luminosité, Contraste ...) peuvent être ajustés indépendamment. Le programme d'acquisition (sous Visual C++) déclenche la capture de 5 images simultanément (voir la figure 1 droite). Ces images sont ensuite stockées sur le disque dur du PC avec un flot de données de $20 \times 5 = 100$ images par seconde.

Le sujet humain est placé devant le banc d'acquisition, face à la caméra centrale (caméra 3). Il possède sur son visage des marqueurs de différentes couleurs qui permettent de récupérer plus tard des points communs entre chaque vue du visage. La position de ces marqueurs correspond approximativement aux points caractéristiques du visage (coin des yeux, menton ...). La figure 2 montre la position des points choisis. Au total, il y a 10 marqueurs sur chaque visage avec au moins 3 marqueurs en commun entre chaque vue.

Avec la caméra utilisée, selon le positionnement de la personne, 1 pixel correspond à une surface qui varie entre 0.7 mm^2 et 1 mm^2 . Chaque marqueur a une surface équivalente de 20 mm^2 et contient donc au moins 20 pixels. Notre méthode de localisation des marqueurs basée sur un calcul de centres de gravités (voir la section 3.1) garantit une précision minimale au pixel près, soit une erreur de la localisation inférieure à 1 mm^2 ce qui est suffisant pour notre application.

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

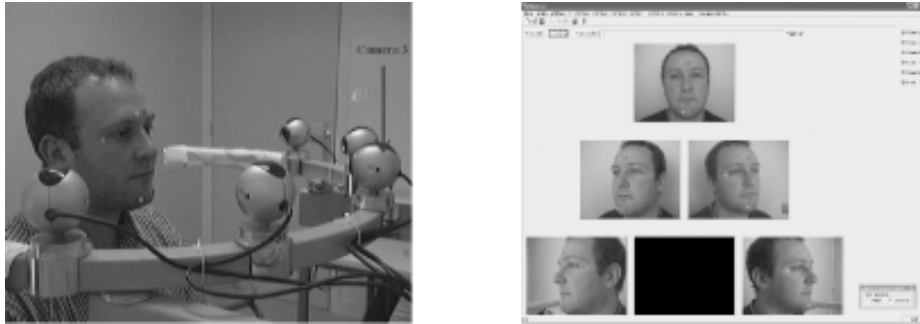


Figure 1. Système d'acquisition : la figure gauche montre les 5 caméras et leur support, la figure droite affiche les 5 images prises avec un sujet. La taille de chaque image est de 240×320 pixels.



Figure 2. Répartition des 10 marqueurs sur les 5 vues (dans le sens des aiguilles d'une montre à partir du haut à gauche) : image 1, image 2, image 3, image 4, et image 5.

3. Construction de visages panoramiques

Plusieurs algorithmes de construction de mosaïques ont été proposés dans la littérature [17] [22] [23] [24] [25] [26] [27]. A. Jain et A. Ross [25] ont développé une technique de mosaïquage d'images permettant de construire une image plus complète d'empreintes digitales à partir de deux images d'une même empreinte. Pour cela, ils alignent d'abord les 2 images en utilisant les points communs de minutie. Ensuite, une version modifiée de l'algorithme ICP (*Iterative Closest Point*) [28] est appliquée sur cet alignement afin de calculer la matrice de transformation qui définit la relation spatiale entre les deux images. À la fin du traitement, une image composite est générée grâce à la matrice de transformation qui possède 6 paramètres indépen-

dants : 3 angles de rotation (α, β, γ) autour des axes x, y , et z et 3 composantes de translation (t_x, t_y, t_z) suivant ces 3 axes.

X. Liu et T. Chen [17] ont utilisé la géométrie faciale pour améliorer les résultats de mosaïquage de visages. Ils ont créé une projection sphérique (au lieu de la projection cylindrique) qui représente mieux le mouvement de la tête suivant les deux directions (horizontale et verticale). Un algorithme de «*matching*» géométrique a été développé pour décrire la correspondance de l'image 2D entre l'espace plane QUV et la surface sphérique $O\alpha\beta$. Le point Q se situe à la pointe du nez d'un visage, QU, QV représentent respectivement les axes horizontaux et verticaux du plan 2D QUV . Le point d'origine O , l'angle horizontal α et l'angle vertical β constituent les coordonnées de l'espace sphérique $O\alpha\beta$.

L'algorithme *Levenberg-Marquardt* a été utilisé pour trouver les deux paramètres de «*matching*» $[\Delta\alpha, \Delta\beta]^T$ afin d'accomplir la transformation depuis l'espace 2D en surface sphérique. Pour

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

réduire le volume de calcul, X. Liu et T. Chen approximent un visage par un ensemble de triangles en appliquant une technique de maillage.

En général, les méthodes utilisant des transformations non linéaires et des algorithmes récursifs obtiennent des résultats très corrects en terme de précision géométrique. En revanche, ces méthodes demandent des volumes de calculs relativement importants et leur réalisation matérielle en temps réel n'est pas toujours facile. Notre objectif pour ce projet « visages panoramiques » est d'aboutir à un système en temps réel. Nous avons décidé d'appliquer une transformation linéaire simple, mais efficace pour construire le mosaïquage des visages. Cet algorithme est divisé en 3 étapes (voir la figure 3) :

1. Détection des marqueurs et calcul de leurs coordonnées,
2. Estimation de la matrice de transformation et transformation de l'image,
3. Construction du visage panoramique.

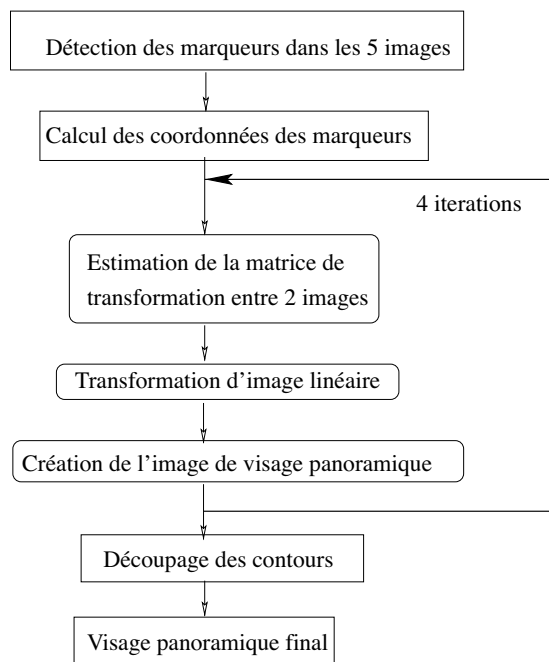


Figure 3. Construction de visages panoramiques basée sur la transformation linéaire successive.

3.1. Détection des marqueurs et calcul de leurs coordonnées

La première étape de la méthode proposée consiste à détecter les marqueurs sur les visages de sujets. Ces marqueurs sont fabriqués avec des papiers autocollants. Nous avons utilisé 3 couleurs pour les 10 marqueurs (4 bleus, 3 jaunes et 3 violets, voir la figure 2 pour une illustration). Ces trois couleurs ont été choi-

sies car elles sont facilement identifiables par rapport à la couleur de peau. Ces marqueurs sont utilisés comme des points de référence pour coller les différentes vues d'un visage.

Nous avons effectué une segmentation des couleurs en utilisant les composantes Teinte et Saturation pour chaque image. Cette procédure permet une forte sélection des couleurs et en même temps une faible sensibilité à la variation de luminance. La figure 4 illustre les étapes de la détection des marqueurs jaunes. Tout d'abord, à partir de l'image d'origine (Figure 4 à gauche), la segmentation de couleurs nous donne une image binaire qui contient des marqueurs (Figure 4 au milieu). Ensuite, nous calculons le centre des zones détectées (Figure 4 à droite). Ces centres nous permettent de trouver les coordonnées des marqueurs dans l'image.

3.2. Estimation de la matrice de transformation et Transformation linéaire de l'image

Nous avons décidé de représenter un visage comme une mosaïque. Chaque mosaïque de visage est fabriquée par la concaténation de différentes vues collées sur une surface plane. Ainsi, pour créer un visage panoramique, les 5 vues sont combinées l'une après l'autre. Nous commençons par la vue centrale et collons une vue latérale à chaque étape (voir la figure 3). Notre méthode consiste à transformer l'image latérale et à ajuster celle-ci avec la vue centrale de manière à ce que les points communs de ces deux vues se superposent.

Nous obtenons l'image transformée par une multiplication de l'image avec la matrice de transformation. Cette matrice de transformation est estimée en fonction des coordonnées des 3 marqueurs communs entre les deux vues. C_1 et C_2 représentent, respectivement, les coordonnées des marqueurs de la première et de la seconde image :

$$C_1 = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \quad (1)$$

$$C_2 = \begin{bmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \end{bmatrix} \quad (2)$$

Nous obtenons la matrice de transformation comme suit :

$$T = C_1 \times (C_2^*)^{-1} \quad (3)$$

avec

$$C_2^* = \begin{bmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{bmatrix} \quad (4)$$

et

$$T = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} \quad (5)$$

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques



Figure 4. Détection des marqueurs jaunes (de gauche à droite) : image d'origine, image binaire obtenue avec le filtrage de couleurs sur les composants Teinte et Saturation et la localisation des marqueurs.



Figure 5. Image 4 avant (gauche) et après (droite) la transformation : la matrice de transformation est estimée en utilisant l'image 4 et l'image 3 (vue centrale, voir les figures 2 et 6).



Figure 6. Superposition de l'image 3 et de l'image 4 : image 3 d'origine (gauche), et superposition de l'image 4 transformée et de l'image 3 d'origine (droite).

Puis, nous généralisons cette transformation sur l'image entière :

$$\begin{aligned} x &= a_1x' + b_1y' + c_1 \\ y &= a_2x' + b_2y' + c_2 \end{aligned} \tag{6}$$

Cette transformation linéaire correspond à une combinaison de la rotation, de la translation et de la dilatation de l'image (voir la figure 5). La figure 6 affiche le résultat de la superposition de l'image 3 (non transformée) et de l'image 4 (transformée en utilisant les coordonnées des marqueurs jaunes).

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

3.3. Construction de visage panoramique

Nous commençons la construction du visage panoramique par la vue centrale (image 3 voir la figure 2). A partir de la superposition de l'image 3 d'origine et de l'image 4 transformée (voir la figure 6), nous supprimons les pixels redondants afin d'obtenir une image panoramique temporaire image 3-4 (voir la figure 7 gauche). Nous avons créé deux lignes de découpage à partir des deux marqueurs jaunes pour éliminer les pixels redondants. Cette image 3-4 devient temporairement notre image cible.

Nous répétons cette opération sur chaque vue. D'abord, l'image 2 est collée avec l'image 3-4 afin d'obtenir une nouvelle image panoramique temporaire (image 2-3-4) (voir la figure 7 droite). La matrice de transformation est générée en utilisant les 3 marqueurs violets qui sont communs avec les deux vues. Ensuite, nous calculons la matrice de transformation permettant de construire l'image 2-3-4-5 (voir la figure 8 gauche) en utilisant les 2 marqueurs bleus et le marqueur jaune le plus haut. Finalement, l'image 1 est collée avec l'image 2-3-4-5 panoramique temporaire à l'aide des 2 marqueurs bleus et le marqueur

violet le plus haut (voir la figure 8 droite).

La figure 9 gauche affiche le visage panoramique final construit à partir des 5 vues. La construction préserve la forme du visage. Par exemple, le menton possède plus de courbures que les autres parties du visage, la partie basse du visage panoramique est composée avec les 5 vues: 1, 2, 3, 4 et 5. En revanche, pour composer la partie haute, du visage (front), les 3 vues suffisent (1, 3 et 5).

La figure 9 droite montre la mosaïque finale du visage obtenue après le découpage automatique des contours. Pour ce processus, nous appliquons d'abord un cercle qui englobe la zone de visage (cheveux compris) et nous effaçons la zone de fond qui se situe à l'extérieur de ce cercle. En parcourant, à l'intérieur de ce cercle, de gauche à droite et de droite à gauche, les lignes centrales, nous détectons automatiquement les points extrêmes des deux oreilles et en même temps, nous réduisons le rayon du cercle. Ensuite, nous remplaçons les segments de ce cercle par des courbes polynomiales en utilisant les coordonnées des points extrêmes localisés à l'aide de la position des marqueurs. Par rapport à la méthode proposée par Liu et Chen [17], les visages panoramiques obtenus avec notre système sont moins

S

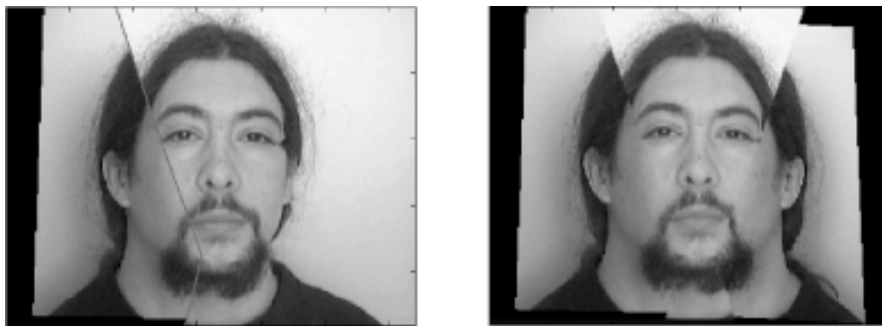


Figure 7. Résultats du mosaïquage : image 3-4 (gauche), et image 2-3-4 (droite).



Figure 8. Résultats du mosaïquage : image 2-3-4-5 (gauche), et image 1-2-3-4-5 (droite).

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

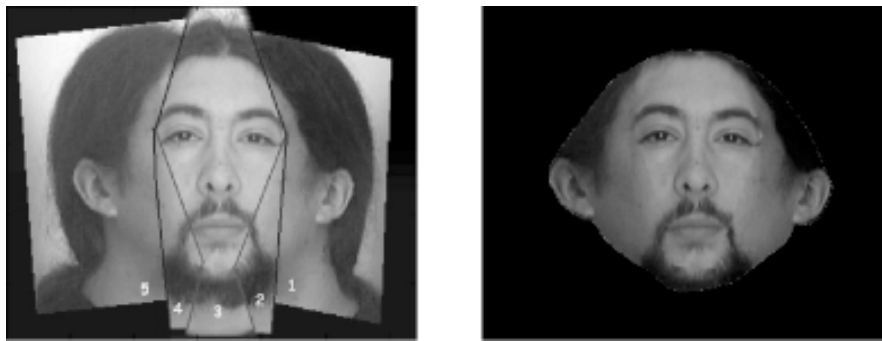


Figure 9. Résultats du mosaïquage : visage panoramique construit à partir des 5 vues (gauche), et mosaïque finale (droite).

précis géométriquement. Par exemple, Liu et Chen utilisent un maillage de triangles pour représenter un visage. Chaque triangle possède ses propres paramètres de transformation obtenus à l'aide d'un algorithme récursif. Dans notre système, une matrice unique de transformation est appliquée sur une image entière. Notre objectif est d'étudier un algorithme simple, mais efficace pour de futures implantations matérielles. Le système proposé crée un visage panoramique à partir de 5 vues en 1.2 seconde avec le logiciel Matlab (Pentium 4B, 2.4 GHz). Ceci nous permet d'envisager une application en temps réel sur des systèmes embarqués.

Afin de tester et de valider notre algorithme de mosaïquage de visage panoramique, nous proposons dans la section suivante, une étude de reconnaissance de visages basée sur le modèle de « visages propres » proposé par Turk et Pentland [3].

4. Reconnaissance des visages panoramiques

Nous avons créé une base de données de visages panoramiques en utilisant la méthode décrite dans la section 3. Cette base de données est composée de 12 personnes \times 4 expressions \times 2 sessions = 96 visages panoramiques. Les deux sessions d'acquisition ont été réalisées dans un intervalle d'un mois. Les 4 expressions de visages sont les suivantes : neutre, souriant, soucieux, endormi (voir la figure 10). Dans le but de valider notre système de mosaïquage de visages panoramiques, nous avons implémenté un algorithme de reconnaissance de visages que nous avons testé en utilisant notre base de données. Nous présentons successivement dans cette section notre algorithme de reconnaissance de visages et les résultats expérimentaux obtenus à partir de notre base de données de visages panoramiques.

4.1. Description de l'algorithme de reconnaissance de visages: ACP

Durant les 25 dernières années, plusieurs techniques de reconnaissance de visages ont été proposées en relation avec un besoin croissant d'applications liées à la sécurité mais aussi avec le développement des approches de modélisation, utilisées en sciences cognitives, pour étudier le comportement humain. L'une des techniques les plus utilisées est basée sur l'Analyse en Composantes Principales (ACP) adaptée aux images de visages [29]. Par exemple, Abdi [30], Turk et Pentland [3] ont utilisé des vecteurs propres de visages pour la détection et l'identification de visages 2D. L'Analyse en Composantes Principales est basée sur l'idée que la reconnaissance de visages peut être réalisée avec un faible nombre de données qui caractérisent d'une manière optimale un ensemble de visages connus (appris). L'application de cette méthode pour la reconnaissance de visages consiste à réaliser tout d'abord l'ACP sur un jeu d'images de visages connus (chaque personne est caractérisée par plusieurs images de visages correspondant à différentes expressions). A partir de cette analyse, un jeu de K composantes principales est obtenu. La projection de nouveaux visages sur ces composantes est réalisée dans le but de calculer les distances entre les nouveaux visages et les visages déjà appris. Ces distances permettent de vérifier si les visages en cours d'analyse sont des nouveaux visages (visages inconnus) ou des visages déjà appris. L'ACP appliquée à la reconnaissance de visages fonctionne de la manière suivante. Les visages à apprendre sont représentés par des vecteurs \mathbf{a}_k où k est le numéro du visage [31]. Ces vecteurs sont obtenus en transformant l'image 2D des visages en une image 1D par la vectorisation d'une matrice. L'ensemble des images de visages transformées en vecteurs est rassemblé au sein de la matrice d'apprentissage \mathbf{A} dans laquelle chaque colonne correspond à un visage.

En utilisant les propriétés de l'algèbre linéaire, cette matrice d'apprentissage peut être exprimée comme :

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques



Figure 10. Exemples de visages panoramiques de la base de données obtenus lors de la première session de mesures.

$$A = P\Delta Q^T \tag{7}$$

où P est la matrice des vecteurs propres de AA^T , Q est la matrice des vecteurs propres de $A^T A$, et Δ est la matrice diagonale des valeurs singulières de A , $\Delta = \Lambda^{1/2}$ avec Λ qui est la matrice des valeurs propres de AA^T et $A^T A$.

En considérant la matrice des vecteurs propres (visages propres), chaque visage de la base de données est représenté par la projection de l'image de visage sur la matrice des vecteurs propres :

$$PROJ_x = x^T P \Delta^{-1} \tag{8}$$

où x est une image de visage dépliée sous forme de vecteur. Il est à noter que l'approche des visages propres permet d'aligner et de normaliser des images pour l'apprentissage et la reconnaissance.

L'identification d'un visage est obtenue en localisant, dans la base de données, l'image d'un visage connu dont le vecteur de projection a la plus petite distance Euclidienne avec celui de l'image testée. Cet algorithme proposé par Turk and Pentland [3] est nommé « Règle de classification des plus proches voisins ». La figure 11 représente la projection de la matrice d'apprentissage A sur les deuxième et troisième composantes principales. Dans cet exemple, 48 visages panoramiques sont analysés.

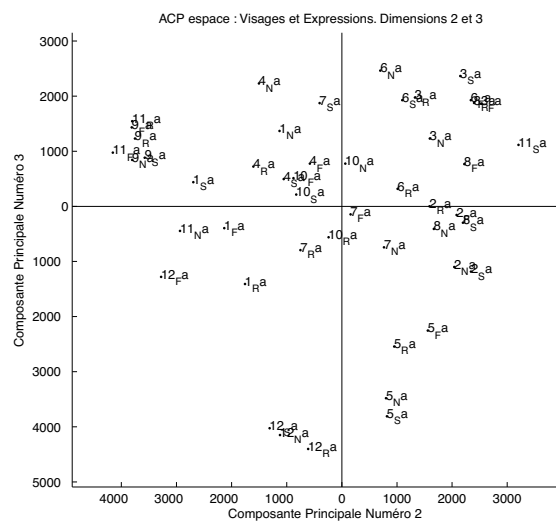


Figure 11. Représentation 2D de la projection de la matrice d'apprentissage A sur les deuxième et troisième composantes principales ; chaque visage panoramique est caractérisé par son identité (numéros 1-12), son expression (N,S,R,F), et sa session d'acquisition (a, b).

sés. Chaque visage panoramique est caractérisé par son identité (12 personnes numérotées de 1 à 12), par son expression (4 expressions différentes codées comme: N-neutre, S-sourire, R-sourieux, F-yeux fermés) et par le numéro de session d'acquisition d'images de visages (deux sessions a et b acquises à un mois d'intervalle). Dans cet exemple nous présentons uniquement le traitement des visages acquis lors de la session a.

4.2. Résultats expérimentaux de la reconnaissance de visages panoramiques

4.2.1. Représentation spatiale

Pour ces tests, nous avons utilisé des images de visages panoramiques avec un format de 240 × 320 pixels (représentation dans le domaine spatial). Nous avons constitué une base de données de 12 personnes, ce qui correspond à 12 × 4 expressions × 2 sessions = 96 visages panoramiques. Nous avons divisé notre base de données en deux blocs. Le premier bloc est utilisé comme base d'apprentissage et le deuxième permet de tester les performances de reconnaissance de notre système. Comme cela est illustré en figure 10, les images de visages panoramiques ont été acquises avec un fond uniforme et ceci sous un éclairage de jour.

À partir de notre base de données de 96 visages panoramiques, nous choisissons d'une manière aléatoire, pour chacune des 12 personnes, des images de visages panoramiques. Nous avons réalisé 4 expériences différentes basées sur le nombre de visages appris pour chacune des 12 personnes. Ce nombre p varie de 1 à 4 images de visages par personne et constitue le premier bloc (bloc d'apprentissage). Le reste de la base de données constitue le deuxième bloc et est utilisé comme base de visages testés en mode reconnaissance. Par exemple, quand $p = 1$, le nombre de visages pour l'apprentissage est égal à 1×12 personnes = 12 et le nombre de visages qui sont utilisés dans ce cas pour tester notre algorithme de reconnaissance est égal à $96 - 12 = 84$.

Nous présentons en Tableau 1 les résultats de reconnaissance de visages que nous avons obtenus avec notre programme Matlab et ceci pour les 4 expériences ($p = 1$ à 4). Nous avons ajouté à la phase d'apprentissage une étape d'analyse discriminante [32] permettant de déterminer le nombre de vecteurs propres utiles pour la reconnaissance de visages. On ré-ordonne les vecteurs

propres non pas en fonction de leurs valeurs propres mais en fonction de leur importance pour la reconnaissance (voir [33] pour la description technique). Pour l'exemple montré à la figure 12, on voit bien que 23 vecteurs propres suffisent pour avoir le taux de reconnaissance maximum (93.75%), les autres vecteurs propres n'apportant rien de plus à la reconnaissance.

4.2.2. Représentation fréquentielle

Nous avons testé aussi le comportement fréquentiel de notre système de reconnaissance de visages panoramiques. La figure 13 gauche représente le spectre d'amplitude d'un visage panoramique. Nous pouvons observer que ce spectre est centré autour des basses fréquences ce qui nous permet d'appliquer un filtre passe-bas dans le but de réduire la taille des données à traiter (voir la figure 13 droite). Ainsi, seulement 80 × 80 données du spectre d'amplitude sont utilisées par notre système de reconnaissance.

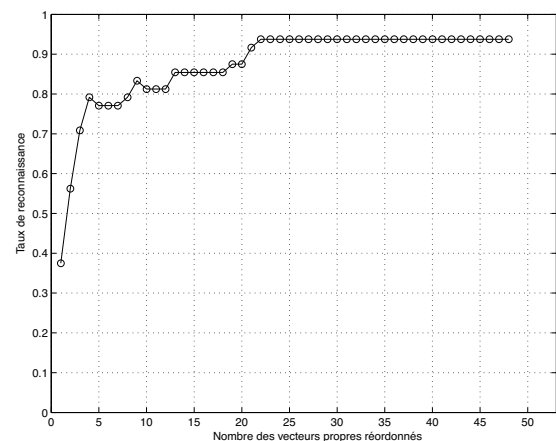


Figure 12. Résultat d'une analyse discriminante pour une base d'apprentissage de 48 visages : nombre de vecteurs propres ré-organisés en fonction de leur importance pour la reconnaissance.

Tableau 1. Résultats de la reconnaissance de visages panoramiques représentés dans le domaine spatial : le nombre de vecteurs propres utilisés correspond à la valeur moyenne obtenue pendant plusieurs exécutions avec l'analyse discriminante.

Nombre exemples appris/personne	$p = 1$	$p = 2$	$p = 3$	$p = 4$
Nombre total d'exemples appris	12	24	36	48
Nombre de vecteurs propres de visages utilisés	9	13	18	25
Nombre de tests pour la reconnaissance	84	72	60	48
Taux de reconnaissance correcte	70 %	85.08 %	90.1 %	93.21 %

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

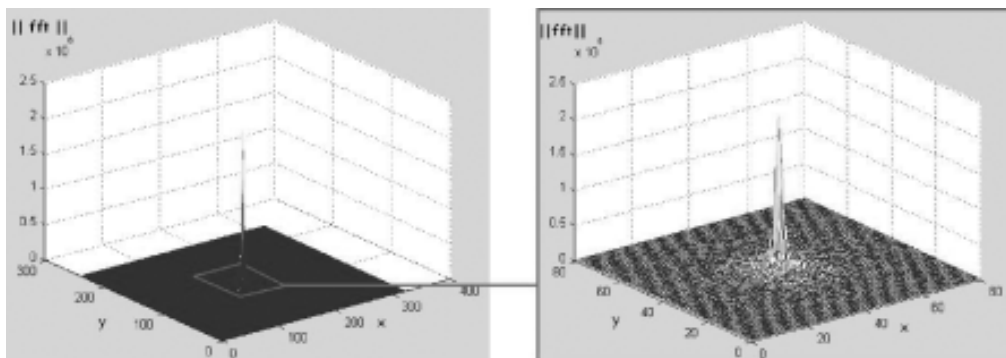


Figure 13. Représentation 3D du spectre d'amplitude original (gauche), et spectre d'amplitude avec les fréquences faibles uniquement $80 \times 80 = 6400$ valeurs (droite) qui sont utilisées par notre système de reconnaissance.

Tableau 2. Résultats de la reconnaissance de visages panoramiques avec la représentation fréquentielle : le nombre de vecteurs propres utilisés correspond à la valeur moyenne obtenue pendant plusieurs exécutions avec l'analyse discriminante.

Nombre exemples appris/personne	$p = 1$	$p = 2$	$p = 3$	$p = 4$
Nombre total d'exemples appris	12	24	36	48
Nombre de vecteurs propres de visages utilisés	8	13	18	24
Nombre de tests pour la reconnaissance	84	72	60	48
Taux de reconnaissance correcte	76.83%	91.26%	93.25%	97.46%

Nous avons réalisé les mêmes tests que ceux que nous avons effectué avec la représentation spatiale. Les résultats sont présentés dans la Table 2. En comparaison avec la représentation spatiale, nous obtenons un meilleur taux de reconnaissance avec la représentation fréquentielle, limitée au spectre d'amplitude (97.46% en comparaison avec 93.21% pour la représentation spatiale).

Ceci est dû au fait que pour les images de visages, le spectre d'amplitude est moins sensible au bruit (ou aux variations) que le spectre de phase. Pour illustrer ce phénomène, nous avons bruité volontairement une image de visage panoramique. La figure 14a représente une image d'un visage panoramique et la figure 14b représente ce même visage avec un ajout de bruit. Nous avons calculé la transformée de Fourier directe de ces deux images puis la transformée de Fourier inverse selon deux configurations différentes :

1) En utilisant le spectre d'amplitude de l'image bruitée et le spectre de phase de l'image originale (voir la figure 14c),

2) En utilisant le spectre de phase de l'image bruitée et le spectre d'amplitude de l'image originale (voir la figure 14d).

Ces résultats montrent que le visage obtenu avec la première configuration est plus proche du visage original que celui obtenu avec la deuxième configuration. Ceci confirme que l'image de visage obtenue uniquement à partir du spectre d'amplitude

est moins sensible au bruit que celle obtenue directement dans le domaine spatial (l'image spatiale initiale possède naturellement les spectres d'amplitude et de phase). Par ce fait les performances en reconnaissance sont supérieures avec la représentation fréquentielle limitée au spectre d'amplitude en comparaison avec les performances obtenues avec la représentation spatiale qui contient les informations en amplitude et en phase. Tsalakanidou *et al.* [9] ont évalué trois approches pour la reconnaissance de visages en utilisant les facteurs suivants : la couleur, la profondeur et la combinaison des deux. Ils ont quantifié la contribution de la mesure de profondeur pour la reconnaissance de visages. Les images de couleurs sont stockées avec une résolution de 720×576 pixels. Pour chaque personne, ils ont créé une carte de profondeur à l'aide d'une approche «éclairage structuré» qui capte la surface 3D d'un visage. Leur expérience montrent qu'un gain significatif de 5% sur le taux de reconnaissance a été apporté par l'information 3D. La technique de reconnaissance utilisée est basée sur l'approche ACP (similaire par rapport à la notre). Ils ont obtenu un taux de reconnaissance de 97.5% en combinant les 2 facteurs (couleur et profondeur). Nous avons établi un système simple et efficace pour construire un visage panoramique à partir de 5 vues et les performances de reconnaissance de visages sont très proches de celles annoncées par Tsalakanidou *et al.*

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

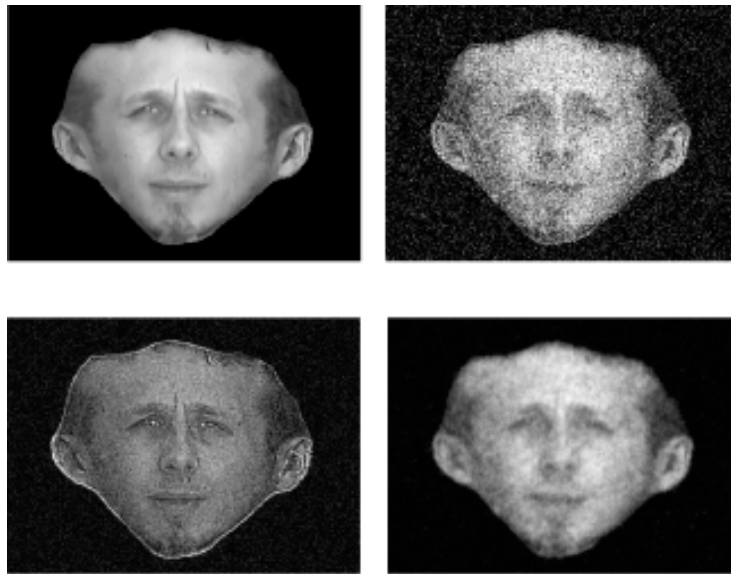


Figure 14. Le spectre d'amplitude est moins sensible au bruit que le spectre de phase : **a**) image originale de visage panoramique (en haut à gauche); **b**) image originale avec un ajout de bruit blanc gaussien (moyenne=0 et variance=0.05) (en haut à droite); **c**) FFT inverse de l'image en utilisant le spectre d'amplitude de **b**) et le spectre de phase de **a**) (en bas à gauche); **d**) FFT inverse de l'image en utilisant le spectre d'amplitude de **a**) et le spectre de phase de **b**) (en bas à droite). Nous pouvons constater que l'image **c**) ressemble plus à l'image **a**) que l'image **d**).



Figure 15. Visages panoramiques de quatre personnes non apprises par le système de reconnaissance.

4.2.3. Reconnaissance de visages panoramiques avec des échantillons négatifs

Dans le but de tester le comportement de notre système de reconnaissance de visages avec des personnes inconnues (non apprises), nous avons ajouté 4 nouvelles personnes dans notre base de données (voir la figure 15). Ces visages panoramiques ont été obtenus selon la méthode décrite dans la section 3.

Dans ce cas, les performances de notre système sont décrites dans la Table 3. Nous avons ajouté 4 personnes \times 4 expressions \times 2 sessions = 32 nouveaux visages panoramiques dans chaque jeu de test. Dans le but de rejeter ces visages inconnus, nous avons introduit un seuillage sur le calcul de la distance Euclidienne. Comme nous envisageons d'utiliser notre système pour des applications de contrôle d'accès pour lesquelles la

confusion est plus grave que la non-reconnaissance, notre seuil de validation de visage connu est élevé et ceci dans le but d'éliminer les intrusions. Pour tous les tests, le seuil de validation est resté constant. Pour cette étude préliminaire de reconnaissance de visages panoramiques, ce seuil est imposé par l'utilisateur de façon expérimentale.

Afin de qualifier notre système, nous avons utilisé les paramètres suivants :

Taux de reconnaissance : reconnaissance correcte des visages panoramiques appris,

Taux de Non-reconnaissance : visage panoramique appris non reconnu,

Taux de Confusion : intrus confondu avec un visage panoramique appris.

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

Tableau 3. Résultats de la reconnaissance de visages panoramiques avec des exemples négatifs: le nombre de vecteurs propres utilisés correspond à la valeur moyenne obtenue pendant plusieurs exécutions avec l'analyse discriminante. Ces performances sont obtenues en utilisant la représentation fréquentielle.

Nombre exemples appris/personne	$p = 1$	$p = 2$	$p = 3$	$p = 4$
Nombre total d'exemples appris	12	24	36	48
Nombre de vecteurs propres de visages utilisés	8	13	18	24
Nombre de tests pour la reconnaissance	116	104	92	80
Taux de Non-reconnaissance	25.4 %	12.74 %	7.58 %	4.82 %
Taux de Confusion	5.85 %	4 %	3.5 %	2.8 %
Taux de reconnaissance	68.75 %	83.26 %	88.92 %	92.38 %

5. Conclusions et Perspectives

Dans cet article, nous avons proposé une méthode simple et rapide pour réaliser un mosaïquage d'image. Le système d'acquisition d'images est constitué d'un ensemble de cinq caméras connectées à un même PC. La fabrication des images de visages panoramiques est obtenue grâce à un algorithme simple basé sur des transformées linéaires rapides ce qui permet d'envisager des applications temps réel.

Dans le but d'étudier les performances de reconnaissance de notre système, nous avons constitué une base de données de visages panoramiques. Notre méthode de reconnaissance de visages est basée sur l'Analyse en Composantes Principales (ACP). Nous avons testé deux types de représentations d'images: spatiale et fréquentielle. La représentation fréquentielle est celle qui nous a apporté les meilleures performances en reconnaissance (97.46 % en comparaison avec 93.21 % pour la représentation spatiale). Un autre avantage de la représentation fréquentielle est la réduction du volume de données à traiter ce qui permet une accélération dans les calculs. Nous avons aussi utilisé des échantillons négatifs de visages panoramiques dans le but de tester la robustesse aux confusions de notre système. Nous avons pu ainsi montrer que dans le cas d'intrusions, le taux de reconnaissance reste supérieur à 92 %. Ainsi, les résultats expérimentaux que nous avons obtenus montrent la validité de notre système de mosaïquage pour traiter des visages dans des scènes 3D. Les performances de notre système sont proches, voire supérieures, à celles déjà décrites dans la littérature [5][1][2][9].

Notre système d'acquisition permet d'obtenir des images de visages selon 5 vues. Avec la méthode proposée, l'utilisateur dispose des 5 vues séparées, mais aussi de la recombinaison panoramique des visages. Dans cet article, nous avons montré les résultats obtenus à partir de l'image panoramique qui représente des avantages comme la simplicité, l'économie de stockage et aussi la rapidité de traitement dus au fait que les 5 images sont condensées en une seule.

Avec notre dispositif, il est aussi possible d'envisager une approche mixte associant les reconnaissances à partir des 5 vues séparées et de l'image panoramique. Des travaux sont en cours actuellement dans notre équipe pour étudier, selon une approche de type fusion d'information, la combinaison de ces 2 approches et leur impact sur l'augmentation du taux de reconnaissance.

En perspectives, nous avons aussi l'intention de simplifier notre système d'acquisition. Nous sommes en train de mettre en place un système composé de 2 rétro-projecteurs pour projeter des marqueurs virtuels de différentes couleurs sur le visage au moment de l'acquisition. Ceci apportera un caractère dynamique à notre dispositif et améliorera l'aspect ergonomique. Nous souhaitons aussi utiliser notre système sans marqueurs. Pour cela, nous allons utiliser des algorithmes de détection de coins et de courbures maximales dans le but de détecter les points caractéristiques des visages. Enfin nous envisageons d'utiliser notre système de reconnaissance de visages panoramiques pour la catégorisation temps réel des expressions humaines, aussi bien dans le cadre d'applications liées à la modélisation en sciences cognitives (étude de l'expression des visages en relation avec le goût des aliments par exemple) que dans le cas d'applications en vidéoconférence.

Références

- [1] A.J. HOWELL and H. BUXTON, *Learning identity with radial basis function networks*, Neurocomputing, Vol.20, pp.15-34, 1998.
- [2] T. SIM, R. SUKTHANKAR and al., *Memory-Based face recognition for visitor Identification*, 4th IEEE International Conf. On automatic face and gesture recognition, Grenoble, France, 26-30 March 2000.
- [3] M. TURK and A. PENTLAND, *Eigenfaces for recognition*, Journal Cognitive neuroscience, Vol.3, pp.71-86, 1991.
- [4] H. ABDI, D. VALENTIN and A. O'TOOLE, *A generalized auto-associator model for face semantic process*, in *Optimization and neural network*, edited by D.Levine (Erlbaum, Hillsdale), 1997.
- [5] M. SLIMANE, T. BROUARD and al., *Unsupervised learning of pictures by genetic hybridization of hidden Markov chain*, Signal Processing, Vol.16, No.6, pp.461-475, 1999.
- [6] P.J. PHILLIPS, P. GROTHOR and al., *Face recognition Vendor Test 2002*, IEEE International workshop on Analysis and Modeling of Faces and Gestures (AMFG), 2003.

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques

- [7] <http://www.frvt.org/FRVT2002/default.htm>
- [8] <http://www.cs.colostate.edu/evalfacerec/>
- [9] F. TSALAKANIDOU, D. TZOVARAS and M.G. STRINTZIS, *Use of delth and colour eigenfaces for face recognition*, Pattern recognition Letters, Vol.24, pp.1427-1435, 2003.
- [10] C. BEUMIER, M. ACHEROY, *Face verification from 3D and grey level clues*, Pattern recognition letters, Vol.22, pp.1321-1329, 2001.
- [11] C. HEHSER, A. SRIVASTAVA and G. ERLEBACHER, *A nouvel technique for face recognition using range imaging*, 7th International Symposium on Signal Processing and its Applications (ISSPA), 2003.
- [12] X. LU, D. COLBRY and A.K. JAIN, *Three-Dimensional model based face recognition*, Proc. International Conference on Pattern Recognition, Cambridge, UK, August, 2004.
- [13] K.W. BOWYER, K. CHANG and P. FLYNN, *A survey of 3D and multi-modal 3D+2D face recognition*, International Conference on Pattern Recognition (ICPR), 2004.
- [14] V. BLANZ and T. VETTER, *Face recognition based on fitting a 3D morphable model*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.25, pp.1063-1074, September, 2003
- [15] J.G. WANG, R. VENKATESWARLU and E.T. LIM, *Face tracking and recognition from stereo sequence*, Computer Science, Vol. 2688, pp.145-153, 2003.
- [16] R. HARTLY and A. ZISSERMAN, *Multiple View Geometry in Computer vision*, Cambridge University Press, Second Edition, 2003.
- [17] X. LIU and T. CHEN, *Geometry-assisted statistical modeling for face mosaicing*, IEEE International Conference on Image Processing (ICIP), Vol.2, pp.883-886, Barcelona, Spain, 2003.
- [18] F. YANG et M. PAINDAVOINE, *Détection et Localisation de Visages dans une Scène : Implantation Parallèle sur un Réseau de DSPs*, Revue Traitement du signal. Vol.17, N° 2, 2000.
- [19] F. YANG, M. PAINDAVOINE, N. MALASNÉ, *Localisation et Reconnaissance de Visages en temps Réel avec un Réseau de Neurones RBF : Algorithme et Architecture*, Revue Traitement du signal. Volume 20, No 4, 2003.
- [20] F. YANG, M. PAINDAVOINE, H. ABDI and J. MITÉLAN, *A new filtering technique combining a wavelet transform with a linear neural network: application to face recognition*, Optical Engineering SPIE. Vol.39, No.11, 2000.
- [21] F. YANG, M. PAINDAVOINE, *Implementation of a RBF neural network on embedded systems: Real time face tracking and identity verification*, IEEE Transactions on Neural Networks, Vol.14, No.5, pp.1162-1175, September 2003.
- [22] Y. KANAZAWA and K. KANATANI, *Image mosaicing by stratified matching*, Image and Vision computing, Vol.22, pp.93-103, 2004.
- [23] Y. ZHOU, H. XUE and M. WAN, *Inverse image alignment methode for image mosaicing and video stabilization in fundus indocyanine green angiography under confocal scanning laser ophthalmoscope*, Computerized Medical Imaging and Graphics, Vol.27, pp.513-523, 2003.
- [24] P.F. MCLAUCHLAN and A. JAENICKE, *Image mosaicing using sequential bundle adjustment*, Image and Vision computing, Vol.20, pp.751-759, 2002.
- [25] A.K. JAIN and A. ROSS, *Fingerprint Mosaicking*, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, Florida, May, 2002.
- [26] D.N. WOOD, A. FINKELSTEIN, J.F. HUGHES, C.E. THAYER and D.H. SALESIN, *Multiperspective Panoramas for Cel Animation*, Proceedings of SIGGRAPH 97, pp.243-250, august, 1997.
- [27] P. RADEMACHER and G. BISHOP, *Multiple-Center-of-Projection Images*, Proceedings of SIGGRAPH, pp.199-206, 1998.
- [28] C. LANGIS, M. GREENSPAN and G. GODIN, *The Parallel Iterative Closest Point Algorithm*, Proceedings of the Third International Conference on 3D Digital Imaging and Modeling (3DIM), Québec, Canada, 2001.
- [29] D. VALENTIN, H. ABDI, A.J. O'TOOLE and G.W. COTTRELL, *Connectionist models of face processing: A survey*, Pattern Recognition, Vol.27, 1208-1230, 1994.
- [30] H. ABDI, *A generalized approach for connectionist auto-associative memories: interpretation, implications and illustration for face processing*. In J. Demongeot (Ed.), *Artificial Intelligence and Cognitive Sciences*. Manchester: Manchester University Press, (1988).
- [31] H. ABDI, D. VALENTIN and B. EDELMAN, *Neural Networks*, Sage, Thousand Oaks, 1999.
- [32] H. ABDI, D. VALENTIN and A. O'TOOLE, *A generalized auto-associator model for face semantic process*, in *Optimization and neural network*, edited by D. Levine (Erlbaum, Hillsdale), 1997.
- [33] A.J. O'TOOLE, F. JIANG, H. ABDI and J.V. HAXBY, *Partially distributed representations of objects and faces in ventral temporal cortex*, Journal of Cognitive Neuroscience, Vol.17, (in press, 2005).

Développement d'un système rapide pour le mosaïquage et la reconnaissance de visages panoramiques



Fan Yang

Fan Yang est maître de conférences de l'Université de Bourgogne depuis 2000. Elle enseigne actuellement l'informatique industrielle, l'automatisme et la robotique à l'UT de Dijon. Ses travaux de recherche au sein du groupe Architecture du LE2I-CNRS UMR 5158 s'intéressent aux méthodes de traitement automatique d'images de visages 2D et 3D, et aux systèmes embarqués en temps réel.



Michel Paidavoine

Michel Paidavoine est professeur à l'Université de Bourgogne. Il enseigne le traitement du signal et des images à l'Ecole d'Ingenieurs ESIREM et à l'IUP Electronique et Image. Il est le directeur du laboratoire LE2I-CNRS UMR 5158 (Laboratoire d'Electronique, d'Informatique et d'Image). Il effectue ses travaux de recherche dans le domaine de l'Adéquation Algorithmes Architectures en traitement d'images.



Dominique Ginhac

Dominique Ginhac est maître de conférences à l'Université de Bourgogne depuis 2000. Il effectue ses travaux de recherches dans le domaine de la conception micro-électronique de capteurs d'images dédiés intégrant des traitements analogiques et/ou numériques.



Julien Dubois

Julien Dubois a rejoint le Laboratoire Electronique, Informatique et Image (LE2I) depuis 2003, où il occupe les fonctions de Maître de Conférence. Ses thématiques principales sont la conception et le développement de caméra rapide et intelligente. A travers ces développements matériels, il s'intéresse plus particulièrement à l'intégration de nombreux algorithmes de traitements d'images, tels que la compression d'images ou la reconnaissance de visage, sur des cibles de type FPGA et DSP (co-design).

4 Travaux de recherche en Algorithmie pour les Sciences Cognitives

4.1 Contexte scientifique

Les sciences cognitives sont un ensemble de disciplines scientifiques visant à l'étude et la compréhension des mécanismes de la pensée humaine, animale ou artificielle, et plus généralement de tout système cognitif, c'est-à-dire tout système complexe de traitement de l'information capable d'acquérir, conserver, et transmettre des connaissances. Les sciences cognitives reposent donc sur l'étude et la modélisation de phénomènes aussi divers que la perception, l'intelligence, le langage, le calcul, le raisonnement ou même la conscience... En tant que domaine interdisciplinaire, les sciences cognitives utilisent conjointement des données issues d'une multitude de branches de la science et de l'ingénierie, en particulier : la linguistique, l'anthropologie, la psychologie, les neurosciences, la philosophie, l'intelligence artificielle.

Au sein des sciences cognitives, le champ de recherche de la psychologie cognitive a pour objectif d'étudier les grandes fonctions psychologiques de l'être humain que sont la mémoire, le langage, l'intelligence, le raisonnement, la résolution de problèmes, la perception ou l'attention. La psychologie cognitive est véritablement née dans les années 1950, en même temps que l'intelligence artificielle. Les débuts de l'informatique ont permis de fournir un arsenal conceptuel permettant d'étudier les grandes fonctions cognitives : la notion d'information et de traitement de l'information. A l'heure actuelle, la modélisation informatique joue un rôle important. Les approches connexionnistes représentent un courant fort pour la modélisation de phénomènes complexes tels que les processus cérébraux en sciences cognitives. La psychologie cognitive utilise préférentiellement l'expérimentation et les mesures comportementales qui comprennent notamment la mesure de temps de réaction (TR), ou du temps nécessaire à une opération (temps de réalisation de la tâche, temps d'exposition en lecture), la précision de la réponse (par exemple taux de bonnes ou mauvaises réponses), ou même l'oculométrie cognitive ou des données physiologiques (imagerie fonctionnelle, potentiels évoqués, etc.).

Ces recherches sont par nature pluri et interdisciplinaires à la frontière des départements Sciences Du Vivant (SDV) et Sciences et Technologies de l'Information et de l'Ingénierie (ST2I). Elles visent une meilleure compréhension des processus cognitifs grâce à l'élaboration de nouveaux outils pour la compréhension du fonctionnement cérébral, en utilisant les méthodes de l'informatique, du traitement d'images, de la réalité virtuelle, les outils statistiques, les techniques de mathématiques appliquées, ...

C'est dans ce contexte que s'inscrit le troisième axe de mes recherches au LE2I. Cet axe concerne des travaux de recherche en **Algorithmie pour les Sciences Cognitives** dans le cadre de collaborations avec le laboratoire de psychologie de l'Université de Bourgogne (Laboratoire d'Etude de l'Apprentissage et du Développement - LEAD UMR CNRS 5022) et le laboratoire de Sciences Cognitives de l'Université Libre de Bruxelles (CONsciousness, COgnition & COmputation Group - CO3). Depuis 2007, je suis Chercheur Visiteur au laboratoire CO3 dans le cadre d'un Congé de Recherche (CRCT) obtenu au titre du CNU 61^{ème} section (2007-2008), puis d'un accueil en délégation au CNRS (2008-2009). La collaboration entreprise avec les spécialistes des sciences cognitives a pour objectif de mieux comprendre les mécanismes cognitifs mis en œuvre dans les phénomènes d'apprentissage,

de reconnaissance, d'attention visuelle. Ces travaux ont conduit à la mise en œuvre de systèmes informatiques et électroniques spécifiques et au développement de différentes applications informatiques dédiées à l'expérimentation et à l'étude de ces phénomènes cognitifs.

Deux principaux projets vont être présentés dans la suite de ce mémoire. Le premier concerne un projet de développement d'environnement de réalité virtuelle spécifique aux enfants qui a été réalisé dans le cadre d'un Programme Interdisciplinaire du CNRS « Cognition et Traitement de l'Information » en 2002. Le deuxième concerne des recherches débutées en 2002 sur le développement de logiciels informatiques dédiés à l'étude des phénomènes d'apprentissage et de reconnaissance.

4.2 Développement algorithmique pour le projet « Enfant et réalité virtuelle » (2002-2004)

Ce premier projet initié en 2002 est le résultat de la collaboration entre 3 enseignants chercheurs de différentes thématiques :

- C. Ferrel du Laboratoire d'Etude de l'Apprentissage et du Développement (LEAD UMR CNRS 5022) de l'Université de Bourgogne
- I. Olivier du Laboratoire « Techniques de l'Ingénierie Médicale et de la Complexité - Informatique, Mathématiques et Applications de Grenoble » (TIMC IMAG UMR CNRS 5525) de l'Université Joseph Fourier de Grenoble,
- D. Ginjac du Laboratoire Electronique Informatique et Image (LE2I UMR CNRS 5158) de l'Université de Bourgogne

Ce projet a reçu le soutien du Programme Interdisciplinaire du CNRS « Cognition et Traitement de l'Information » en 2002 pour une durée de 2 ans, sous la forme d'une aide financière de 50 k€.

L'évolution des technologies permet maintenant à un être humain de réaliser une tâche dans un environnement inaccessible parce que distant, restreint ou virtuel. De telles situations contraignent l'être humain à s'adapter par l'intermédiaire d'une interface. Ces dispositifs modifient la coordination visuo-manuelle normale : les informations visuelles et proprioceptives sont altérées et entrent en conflit. De nombreuses recherches en sciences et technologies de l'information et de la communication s'intéressent au développement et à l'optimisation des dispositifs de réalité virtuelle. Il est toutefois intéressant de constater que les enfants ont été oubliés dans cette démarche. Alors qu'ils représentent un marché prometteur, les dispositifs actuels sont inadaptés à leur morphologie et à leur développement sensori-moteur et cognitif. Il semble donc indispensable de reconsidérer la notion d'environnement virtuel pour les enfants. Le premier objectif de ces recherches a donc été de créer un environnement virtuel adapté aux enfants et à leur développement. L'idée qui est développée est de garder le minimum de matériel standard de réalité virtuelle et d'utiliser aux mieux les possibilités du traitement d'images en temps réel. Dans un deuxième temps, un tel dispositif est également un moyen privilégié pour appréhender le développement de la coordination visuo-motrice chez l'enfant. Tout d'abord, les recherches se sont intéressées aux référentiels spatiaux utilisés par l'enfant pour effectuer un geste. Elles ont cherché à savoir si certaines contraintes pouvaient modifier le choix des enfants. D'autre part, les recherches ont étudié les mécanismes permettant de s'adapter à un espace virtuel. Le transfert d'apprentissage du virtuel au réel a été testé afin d'évaluer si ce type de tâche peut avoir des conséquences sur le développement de la coordination normale.

En résumé, les principaux objectifs des travaux menés étaient premièrement de créer un environnement virtuel adapté aux enfants utilisant le moins de matériel possible, et deuxièmement de mesurer l'influence du virtuel sur les performances motrices des jeunes enfants, en particulier dans le domaine de la préhension d'objets.



Fig. 15 : Dispositif expérimental de réalité virtuelle

L'année 2003 a été consacrée essentiellement au choix du matériel de réalité virtuelle. J'ai mené une collaboration étroite avec la société Immersion afin de développer et tester un gant de données dédié aux jeunes enfants cf. Fig. 15). Les résultats sont extrêmement satisfaisants puisque ce gant est maintenant validé et commercialisé par cette société.

Par la suite, en 2004, je me suis consacré à la définition des caractéristiques de l'environnement virtuel, à la mise en œuvre des différents matériels et à la programmation de l'ensemble des différentes tâches motrices au sein de cet environnement.

L'étape suivante a consisté à valider les concepts retenus sous la forme de passations d'expériences utilisant le matériel de réalité virtuelle au sein d'écoles primaires. Ces passations ont eu lieu en 2005 et ont été réalisées par les autres partenaires impliqués dans le projet. Malheureusement, les premiers résultats obtenus n'ont pas été aussi prometteurs que ce que nous pouvions espérer et n'ont donné lieu qu'à la rédaction d'une communication dans une conférence nationale [38].

4.3 COGSCI : Apport des Sciences Cognitives à l'Apprentissage et la Reconnaissance de Forme (2002-...)

Parallèlement aux travaux sur les architectures matérielles dédiées à la reconnaissance de visages, j'ai entrepris une collaboration avec le Laboratoire d'Etude de l'Apprentissage et du Développement (LEAD UMR CNRS 5022), spécialisé dans les sciences cognitives. L'idée première était d'essayer de tirer parti des approches cognitives afin d'améliorer nos systèmes matériel et logiciel de reconnaissances de visage.

Entre 2002 et 2006, une première collaboration dans le domaine de l'Apprentissage Implicite a été menée avec Pierre Perruchet (Directeur de Recherche CNRS) et Stéphanie Chambaron (Doctorante de 2002 à 2005, puis ATER en 2006) au LEAD. Ces recherches ont conduit au développement de logiciels dédiés à la génération de stimuli visuels, de mesures en temps réel de l'apprentissage et d'analyses statistiques des données recueillies. Ce projet interdisciplinaire a donné lieu à la rédaction de 1 article de revue internationale [5] et de plusieurs conférences internationales [25, 28] et nationales [37].

Cette collaboration avec Stéphanie Chambaron (Chargé de Recherche FNRS - Université Libre de Bruxelles depuis 2007) sur l'Apprentissage Implicite est en pleine expansion à l'heure actuelle et a contribué à la mise en place de nouvelles collaborations, et plus particulièrement avec le laboratoire « Mouvement et Perception », UMR CNRS 6152 de l'Université Aix Marseille II, le laboratoire « Motricité Plasticité », INSERM/ERM 207 de l'Université de Bourgogne. Ces travaux ont donné lieu à la rédaction de 1 article de revue internationale [3], 1 article de revue nationale [4], 6 conférences internationales [14, 15, 16, 17, 20, 21] et une conférence nationale [34].

De plus, depuis 2007, j'effectue une partie de mes travaux de recherche au sein du laboratoire CO3 de l'Université Libre de Bruxelles dans le cadre de mon CRCT puis de ma délégation au CNRS. La collaboration entreprise avec les spécialistes des sciences cognitives de ce laboratoire a pour objectif de mieux comprendre les mécanismes cognitifs mis en œuvre par les êtres humains pour apprendre, localiser et reconnaître n'importe quel objet dans n'importe quelle scène. L'idée sous jacente est 1) d'étudier les mécanismes d'apprentissage des régularités présentes dans les objets situés dans l'environnement visuel, 2) de comprendre comment ces régularités sont utilisées pour permettre la localisation et la reconnaissance des objets dans la scène et 3) de doter un système de vision de telles fonctionnalités. Le défi majeur est donc d'adapter les travaux sur l'apprentissage de séquences menés au CO3 à ma problématique de traitement d'images afin de concevoir à plus long terme une rétine artificielle de fonctions de haut niveau lui permettant d'apprendre et d'interpréter une scène visuelle complexe.

4.4 Publication jointe

S. Chambaron, D. Ginhac, P. Perruchet. *Methodological issues and computational software dedicated to SRT tasks*. Behavior Research Methods, 40(2): 493-502, 2008.

gSRT-Soft: A generic software application and some methodological guidelines to investigate implicit learning through visual–motor sequential tasks

STÉPHANIE CHAMBARON, DOMINIQUE GINHAC, AND PIERRE PERRUCHET
Université Libre de Bruxelles, Brussels, Belgium

Serial reaction time tasks and, more generally, the visual–motor sequential paradigms are increasingly popular tools in a variety of research domains, from studies on implicit learning in laboratory contexts to the assessment of residual learning capabilities of patients in clinical settings. A consequence of this success, however, is the increased variability in paradigms and the difficulty inherent in respecting the methodological principles that two decades of experimental investigations have made more and more stringent. The purpose of the present article is to address those problems. We present a user-friendly application that simplifies running classical experiments, but is flexible enough to permit a broad range of nonstandard manipulations for more specific objectives. Basic methodological guidelines are also provided, as are suggestions for using the software to explore unconventional directions of research. The most recent version of gSRT-Soft may be obtained for free by contacting the authors.

Implicit learning is usually defined as the process by which people learn without intent and without being able to clearly articulate what they are learning (for reviews, see Perruchet & Pacton, 2006; Shanks, 2005). The growing interest in implicit learning stems from its crucial role in the acquisition of one's mother language and in the development of other cognitive, social, and motor abilities. Another attractive feature of implicit learning is that it has proven to be relatively insensitive to age (e.g., Curran, 1997; D. V. Howard & J. H. Howard, 1989; Kotchoubey, Haisst, Daum, Schugens, & Birbaumer, 2000) and is preserved in a number of neuropsychological disorders (e.g., McDowall & Martin, 1996; Smith, Siegert, McDowall, & Abernethy, 2001; Stevens et al., 2002; Zillmer & Spiers, 2001). As a consequence, the phenomenon is a focus of investigation not only for laboratory researchers, but also for those oriented toward educational or clinical objectives.

Although several tasks have been used to investigate implicit learning (e.g., the artificial grammar learning task proposed by Reber, 1967, and the dynamic control task used by Berry & Broadbent, 1984), motor sequence-learning tasks are increasingly popular. In the most typical paradigm, coined the serial reaction time (SRT; Nissen & Bullemer, 1987) task, a target stimulus appears on successive trials at one of a limited number of positions. Participants are asked to react to the appearance of the target by pressing a key that spatially matches the location of the target on a keyboard. Unknown to participants, the sequence of events is not random. It usually consists of the continuous cycling of the same sequence. Learning

is attested by the fact that reaction times (RTs) progressively decrease with practice of the repeated sequence and suddenly increase when a random sequence is unexpectedly inserted. This indicates that participants have acquired knowledge about the structured nature of the repeated sequence. However, even if it has been shown that participants demonstrate sequence learning, the debate about the nature of the acquired knowledge—implicit versus explicit—remains open. Consequently, various tests of awareness have been proposed for evaluating explicit and implicit sequence knowledge. First, Perruchet and Amorim (1992) developed a recognition task in which participants are presented some short sequences and have to discriminate between the sequences that follow the learned structure and the sequences that violate it. Subsequent studies have established that participants are able to recognize and discriminate correct sequences (Perruchet, Bigand, & Benoit-Gonin, 1997; Shanks, 2003; Shanks, Wilkinson, & Channon, 2003). Second, other researchers have proposed various generation tasks in which participants have to reproduce either the whole learned sequence or some fragments of it. In the case of the *free generation* task (Destrebecqz & Cleeremans, 2001; Perruchet & Amorim, 1992; Shanks & Johnstone, 1999), participants have to freely generate a sequence that is as similar as possible to the sequence learned during the training phase. This implies that participants have to remember a substantial amount of the structure of this learned sequence. On the contrary, in a *cued generation* task (Willingham, Nissen, & Bullemer, 1989), participants are presented ele-

S. Chambaron, schambar@ulb.ac.be

ments of the sequences and are instructed to press the button corresponding to where they think the next stimulus will appear. The stimulus remains present until the participant makes the correct response. A variation of this procedure is the *trial-by-trial sequence generation* task proposed by Wilkinson and Shanks (2004). Participants observe a short, five-element sequence of targets from the training sequence and then have to produce a single generation response corresponding to the correct continuation response. Norman, Price, Duff, and Mentzoni (2007) have proposed a novel generation task, the *generation rotation* task. This task is a modification of the existing trial-by-trial generation task, but with the addition of a randomly varying contextual cue. According to Norman et al., this task provides a more robust measure of explicit sequence knowledge.

Several reasons explain the success of the SRT paradigm. There is no doubt that sequential behavior is involved in virtually any world-wide ability, from language to the organization of movements, thus ensuring a good ecological validity to sequential tasks. The use of a visual-motor implementation makes a quantitative assessment of learning easy to get, and robust learning has proven to be possible within a short time in a large variety of populations, from children (Vinter & Perruchet, 2000) to elderly people (J. H. Howard & D. V. Howard, 1997). Another advantage of the SRT task over some other tasks of implicit learning is that participants are in truly incidental conditions of learning, because the effect of regularities can be assessed without participants having been informed about the presence of hidden regularities (Cleeremans, 1993; Destrebecqz & Cleeremans, 2001). Finally, it has been shown that the reliability of SRT tasks is pretty good in comparison with other tasks of implicit learning (Salt-house, McGuthry, & Hambrick, 1999). This property is essential when the aim of the researcher is to compare the learning abilities of different samples of participants and, moreover, when the residual learning abilities of patients need to be assessed on an individual basis.

A potential difficulty in the development of dedicated SRT task software is the large number of to-be-implemented procedures that are becoming increasingly complex. Indeed, since the initial study by Nissen and Bullemer (1987), SRT tasks have been the object of a large number of investigations that have led to both the emergence of a number of variants and the growing sophistication of methodological controls (e.g., Bischoff-Grethe, Goedert, Willingham, & Grafton, 2004; Chamberon, Ginhac, & Perruchet, 2006; Curran & Keele, 1993; Osman, Bird, & Heyes, 2005; Perruchet et al., 1997; Ziessler & Nattkemper, 2001).

Another possible obstacle against a still larger extension of SRT tasks, however, is the increasing difficulty of their implementation. SRT tasks require accurate time measurements, and this has been shown to be problematic in a multitask environment, such as Microsoft Windows. It has been recommended to use MS-DOS only, in which millisecond accuracy is possible (Myors, 1999). However, modern operating systems, such as Microsoft Windows XP, offer sophisticated mechanisms for recording time measures with

submillisecond accuracy (more detailed explanations are given in the section dealing with records of RTs).

The purpose of the present article is to help researchers address all of these difficulties. We propose flexible, easy-to-use software and some guidelines for using it in accordance with up-to-date methodological principles.

OVERVIEW OF GSRT-SOFT

gSRT-Soft is a user-friendly application that makes it very easy to run classical SRT experiments, but is flexible enough to permit a broad range of nonstandard manipulations for more specific objectives. By default, the software has been designed to allow a straightforward and time-saving implementation of the most popular paradigms (e.g., a four-choice SRT task, with participants' responses being entered on the computer keyboard). Thus, it can be used, for instance, by a neuropsychologist wishing only to add a standard SRT task to a large-scale test battery of cognitive functions. However, it also integrates a number of features that enable users to implement nonstandard situations. For instance, the targets can be arranged in order to draw a labyrinth that covers all the bidimensional space of the screen, and a computer mouse or graphic tablet can be used as a manipulandum instead of the keyboard.

gSRT-Soft was developed in C++, which was chosen because it is a general-purpose programming language with high-level and low-level capabilities. The high-level capabilities allow easy development of large pieces of software, whereas the low-level capabilities allow the programmer to manage real-time aspects of the software, such as high-resolution measures of RTs.

The software is a Windows-based program using menus, buttons, and selection boxes accessible with a mouse. It includes two main menus—*Configuration* and *Participant*. Configuration determines the general parameters for a given experiment, which need to be set before the Participant menu can be accessed. First, the Configuration menu allows either the generation of a new configuration, the loading of an existing configuration stored on a file, or the saving of the current configuration to a file. These files are standard INI files frequently used for Microsoft Windows-based applications. In the second menu, the *New Participant* menu specifies the particular values used for each participant of a previously configured experiment.

Configuration

Figure 1 shows the available possibilities in the Configuration dialog box. The main choice concerns whether participants' responses are made on the keyboard or with a mouse. As anywhere else in the program, the user indicates his or her choice by clicking the appropriate radio button. When the keyboard option (default) is selected, two suboptions follow. Because an overwhelming proportion of SRT studies involve the same stimulus-response pattern (namely, four target locations and four spatially congruent response keys), this configuration is proposed as a default (the researcher can choose either the European AZERTY or the U.S. QWERTY keyboard). However, the configuration can be customized with regard to

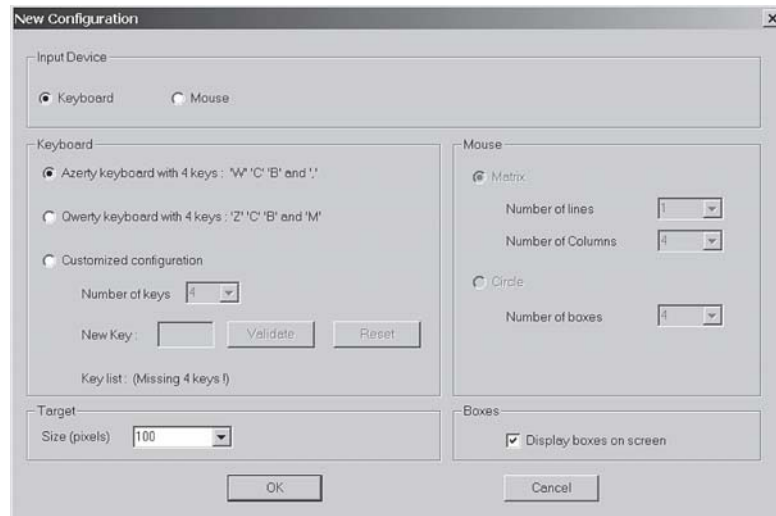


Figure 1. The New Configuration dialog box.

which keys match which target. The selection of the keys may serve several objectives. For instance, in most studies, there is a direct mapping between the location of the target on the screen and the spatial arrangement of the keys on the keyboard. Whether stimulus–response mapping is direct or not, however, has proven to be influential in a large number of motor control studies (e.g., Deroost & Soetens, 2006; Stöcker, Sebald, & Hoffmann, 2003). This software allows the exploration of the effect of this variable in SRT studies. It is possible, for instance, to pair the leftmost target with the rightmost key, or to design any other stimulus–response pairings.

During keyboard configuration, changing the number of keys is also possible, although this option may turn out to have limited utility. This is not because the nearly ubiquitous use of four target locations would be rooted in a principled advantage of this specific configuration; rather, it is due to the fact that using more than four keys causes a considerable slowing down of the responses and/or a dramatic increase in the error rate, for obvious reasons (for the use of six keys, see, e.g., Cleeremans, 1995; Heyes & Foster, 2002; Jiménez, Méndez, & Cleeremans, 1996). Limiting the number of possible events to four (or even six) can be damaging, restricting the ecological validity of the SRT task in a skill-learning context. Also, a sequence involving a larger number of different events

allows the exploration of learning of much richer statistical regularities.

The use of a computer mouse is essentially aimed at overcoming the practical constraints linked to the use of a computer keyboard. The number of events is no longer limited, except by the spatial constraints linked to the positioning of the targets on a computer screen. In gSRT-Soft, the targets can be disposed either in a matrix (X rows and Y columns, maximum) or along a circle (X targets, maximum). The maximum values for rows and columns in a matrix layout or in a circle layout are dependent on screen resolution and target size. For example, with a standard XGA resolution ($1,024 \times 768$ pixels) and a 100-pixel target size, the maximum number of rows is six and the maximum number of columns is nine. In the case of a circle layout, the maximum number of targets is 12. These maximum values are automatically evaluated in order to avoid an overlap between two consecutive locations on the screen.

Three different configurations (standard 4-position, 3×4 matrix, and 12-position circle) are depicted in Figure 2.

Note that with a matrix pattern, the sequence can be ordered in such a way that the target draws a labyrinth on the screen. A legitimate question, however, may be whether using a mouse recruits the same learning process as when keypressing is involved. To address this question,

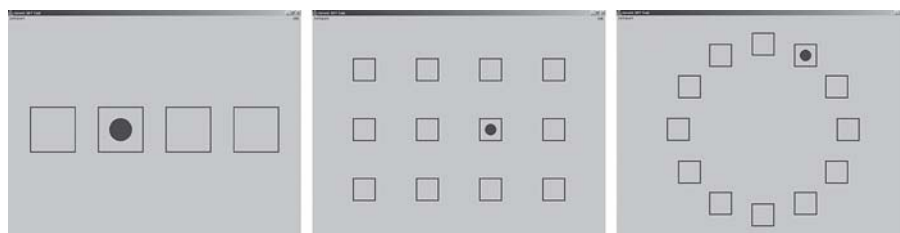


Figure 2. Different configurations for the targets.

we (Chambaron et al., 2006) have borrowed the design of a prior SRT study (Shanks, 2003), using either a keyboard or a mouse, as a function of participants. Although the mean RT was significantly larger with a mouse, evidence of learning was observed in both conditions, with no significant difference in the amount of learning. This result suggests that using a mouse may represent a promising alternative to the use of the keyboard whenever the researcher wishes to include more than a very few different targets in the trained sequence.

Finally, the size of the target needs to be defined. The default value is 100 pixels. However, the target can be reduced (to 1 pixel!) if the researcher wishes to degrade the perceptual discrimination of the stimuli—a procedure that may strengthen the role of location anticipation. Note that with the mouse option, a small target also increases motor accuracy constraints. The size of the target can also be increased when the opposite objectives are aimed. For obvious reasons, the maximum size depends on the number of targets and on their spatial configuration. When an illegal (i.e., too large) value is entered, the program returns an error message indicating the maximum value allowed, given the specific configuration.

By default, the possible locations of the targets are marked on the screen throughout the session by empty boxes in which the targets appear. However, an option can be selected in the Configuration menu, preventing these boxes from being displayed on the screen throughout the session.

New Participant

After the configuration is complete, clicking on the New Participant element menu opens the window displayed in Figure 3. The first choice concerns the task to be run. The label “SRT task” designates the main training task. This task may be followed by a recognition task. The general organization is the same for the two tasks.

The training sequences that a researcher may wish to explore may differ along a virtually unlimited number of features. They may be composed from the repetition of a sequence, but they also may be generated by a finite-state grammar (e.g., Cleeremans & McClelland, 1991) or other set of rules. They may be deterministic or probabilistic (see below), and they may differ in length and in a number of other parameters. This boundless variety makes it unmanageable to elaborate the sequence through an interactive set of options. The problem has been solved in gSRT-Soft by dissociating the generation of the sequence from the main program, with the consequence that the program reads only a previously prepared sequence of trials stored in the *stimuli input file*. A few typical sequences are included in the package (available on request). In most other cases, a standard spreadsheet, such as Microsoft Excel, prepares a set of original data quite well. For more sophisticated objectives, the researcher may use his or her preferred programming language, provided the resulting values are stored as a text file. In the case of a repetition of a sequence, the researcher only needs to generate a single text file that includes all the successive positions of the target. This text file will be used for all of the participants in the experiment. On the contrary, if the researcher

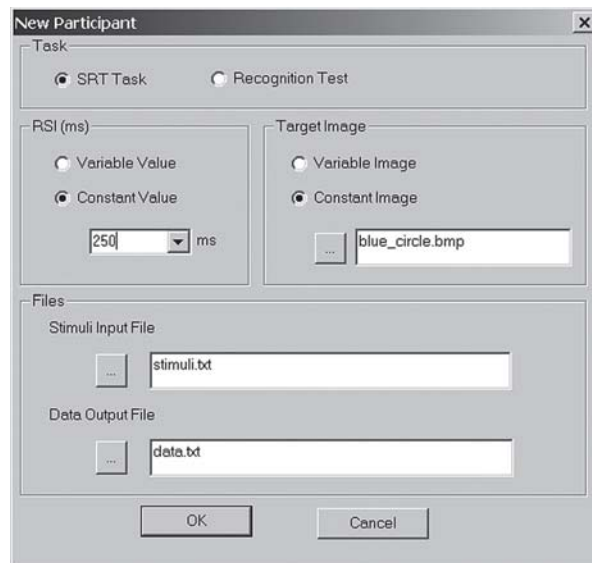


Figure 3. The New Participant dialog box.

designs individual, different blocks that vary randomly between participants, for example, the researcher must create one stimuli input file for each participant, and must include every block of stimuli in the desired order.

The stimuli input file must include a list of target locations. In the file, there is only one stimulus per line. The separator of the stimuli is the end-of-line (EOL) character. Instead of a stimulus, a line may include a “P” character, which indicates the presence of a self-paced pause. For instance, when the software reads the following 10 lines:

```
1
2
3
4
P
1
2
3
4
P
```

it generates two sequences, in which the target moves from the left to the right, and the two sequences are separated by a pause.

Moreover, concerning the recognition test, participants are classically presented with small sequence fragments and are asked to classify them as instances of the training material or not. In this case, the “R” character is inserted in the stimuli input file between each sequence to be recognized. The user decides the number of sequences to be recognized and the order in which these sequences are to be presented to the participants. In other words, the user has to prepare the set of sequences in the stimuli input file in such a way that “old” versus “new” sequences appear in a specific order. For instance, if we consider the sequence 1–2–3–4 to be the training sequence (i.e., *old* sequence), then the stimuli input file that reads as follows:

```

1
2
3
4
R
2
1
3
4
R
4
2
3
1
R
1
2
3
4
R

```

presents four sequences to be recognized in the following order: old, new, new, old. For simplicity, a two-alternative scale has been chosen in gSRT-Soft with default values for the question *Do you recognize this sequence?* and the two possible answers (*yes, no*). These default values can be modified by the user in the INI file. At present, gSRT-Soft does not allow recognition tasks based on gradual scales—for example, from 0 (*highly unfamiliar sequence*) to 10 (*highly familiar sequence*). Future releases may include such an option if this is requested.

The stimuli input file may be the same for all of the participants of a given experiment (note that by default, the program displays the path and the name of the last file that has been loaded, so this information does not need to be typed for each participant). However, in many cases, the stimuli input file differs from participant to participant, generally at the end of counterbalancing or randomizing.

An important parameter in the SRT task is the response–stimulus interval (RSI), which is the time that elapses between the participant’s response (which, as a rule, triggers the suppression of the current target) and the onset of the next target. This value is usually kept constant throughout the session. A standard value of 250 msec is set by default in the software. However, it may be interesting to use variable RSIs (see below). In this case, the sequence of RSIs needs to be prepared. In the stimuli input file, each stimulus is followed by the associated RSI value. For instance, consider the following configuration:

```

1 0
2 500
3 0
4 0
P
1 0
2 500
3 0
4 0
P

```

There is no delay between Locations 1 and 2, nor between Locations 3 and 4, but there is a 500-msec interval between Locations 2 and 3.

Another important parameter in gSRT-Soft is the possibility of displaying any bitmap image on the screen as a target stimulus. Some basic bitmap files are distributed with the package, including geometric shapes (circle, square, triangle) in various colors (blue, yellow, red, black, etc.). However, any bitmap file can be used as a target. The image is automatically resized in order to be correctly displayed in the boxes on the screen. In the same manner as with the RSI value, it is possible to choose either a constant picture for all the stimuli or a variable picture associated to each stimulus. Consider the following example:

```

1 grey_circle.bmp
2 black_circle.bmp
3 yellow_circle.bmp
4 blue_circle.bmp
P
1 grey_triangle.bmp
2 black_triangle.bmp
3 yellow_triangle.bmp
4 blue_triangle.bmp
P

```

In the first sequence 1–2–3–4, a circle of different colors (grey, black, yellow, blue) is used, whereas in the second sequence, a triangle is displayed.

This option is innovative, making gSRT-Soft flexible enough to allow the programming of novel SRT procedures. We can imagine realizing SRT tasks with sequences of letters, words, or pictures. Such a possibility is used in the serial naming task developed by Goschke and Bolte (2007) in order to investigate implicit learning of repeated sequences of abstract semantic categories. In this task, participants have to name pictures of objects displayed on a screen (e.g., table, shirt, etc.). Unknown to the participants, the semantic categories of the objects (e.g., furniture, clothing) follow a repeating sequence.

For obvious reasons, we envision generating some sequences based on variable RSI values and variable target images. In such a case, the stimuli input file must be written in the manner of the following:

```

1 0 grey_circle.bmp
2 500 black_circle.bmp
3 0 yellow_circle.bmp
4 0 blue_circle.bmp
P
1 0 grey_triangle.bmp
2 500 black_triangle.bmp
3 0 yellow_triangle.bmp
4 0 blue_triangle.bmp
P

```

However, the present release of gSRT-Soft does not allow the display of multiple stimulus displays at the same time, as was seen in the procedure used by Norman et al. (2007). In their experiments, the four possible target positions were indicated by four shape outlines (heart, circle, square, cross) colored red, green, or blue on a white background. On each

trial, one shape was filled solid with the same color as its outline, and the other three remained unfilled. The filled shape was the target stimulus. Such an option will be considered with much attention by gSRT-Soft developers for future releases if enough researchers request it.

Before running the experiment, the path and the name of the data output file need to be specified. If this information is kept unchanged from one participant to the next, the data will be appended in the specified file. First, the file contains various information, such as the name and version number of the software, the date, and the time the task was performed. Moreover, the configuration parameters of the experiment are stored: input device, configuration of the input device, number of boxes, and size of the target. Then, the data about the participant are saved. They include the nature of the task, the name of input and output files, the type of RSI, and the target images. Finally, in the case of the training phase, the collected data are presented in a matrix comprising, on each row, the index of the trial, the location of the stimulus for a trial, the participant's answer, a binary value in the *good* column indicating whether this response is correct (1) or not (0), the RTs in milliseconds, the RSI (if variable), and the target image (if variable). The example matrix in Figure 4 includes eight rows corresponding to the eight trials previously described. Between the fourth and the fifth trial, a pause is inserted, and no data is saved in this case.

In the case of the recognition task, two different kinds of data are automatically recorded. First, RTs are stored in a matrix similar to the one previously described for the practice phase. Second, for each sequence to be recognized, the response given by the participant is stored in the output file.

As depicted in Figure 4, RTs are stored in milliseconds, with a high degree of accuracy. High-resolution timing is supported in modern operating systems (such as Microsoft Windows XP) that offer sophisticated mechanisms for recording time measures at submillisecond accuracy. In our case, high-resolution timing is supported by the two main following functions: *QueryPerformanceCounter* and *QueryPerformanceFrequency*. The first call, *QueryPerformanceCounter*, returns the amount of time that has elapsed since the system was booted. This amount of time is expressed in ticks of the processor clock. The second function, *QueryPerformanceFrequency*, returns the frequency of the processor clock. To retrieve the elapsed time of a code section, one has to get the actual value of the

high-resolution performance counter immediately before and immediately after the section of code to be timed. The difference of these values would indicate the number of clock ticks that had elapsed while the code executed. The elapsed time can be computed then, by dividing this difference by the frequency of the processor. Such a method allows gSRT-Soft to record RTs with very high accuracy.

SOME METHODOLOGICAL GUIDELINES

The remainder of this article outlines some basic methodological principles for making the best use of the software. We focus on the most standard SRT paradigms, which are also those on which our methodological knowledge is the most developed. To give a first hint about the difficulties inherent in planning an SRT experiment, let us consider the seminal study by Nissen and Bullemer (1987), in which the same 10-element sequence (4-2-4-1-3-2-4-3-2-1) was continuously repeated. Note that this sequence includes Locations 2 and 3 three times, and Locations 1 and 4 two times. This raw frequency information can be used by participants to improve their performance (Shanks, 2003). Moreover, even if one considers that sequential information is learned, it remains difficult to gain more knowledge about the learning capabilities of participants. An essential question about sequence learning is whether participants take into account the information provided by the immediately preceding event, by the two prior events, or by still higher order information. In the sequence used by Nissen and Bullemer, Location 3 allows one to predict Location 2, and thus it is possible that improved performance simply reflects knowledge of the first-order dependency 3-2. On other parts of the sequence, however, predicting the next trial requires considering at least two, and occasionally three, successive events (predicting the event following 3-2 implies considering whether the prior location is 1 or 4). The sequence makes it impossible to know whether participants actually learn more than first-order dependency rules.

The Order of the Dependency Rules

The example above makes it obvious that a proper assessment of what participants learn requires that the order of the dependency rules be homogeneous throughout the sequence, although a few authors have used hybrid sequences embedding relations of different order (e.g., Cohen, Ivry, & Keele, 1990). Many well-controlled studies now employ a 12-element sequence of targets known as the second-order

Index	Stimuli	Answer	Good	RT	RSI	Target
1	1	1	1	838.247	0	grey_circle.bmp
2	2	2	1	330.454	500	black_circle.bmp
3	3	3	1	598.474	0	yellow_circle.bmp
4	4	4	1	750.548	0	blue_circle.bmp
5	1	1	1	794.300	0	grey_triangle.bmp
6	2	2	1	362.484	500	black_triangle.bmp
7	3	3	1	686.425	0	yellow_triangle.bmp
8	4	4	1	694.526	0	blue_triangle.bmp

Figure 4. Example of a data output file.

conditional (SOC) sequence (Reed & Johnson, 1994). An example of the SOC sequence is 3-1-4-3-2-4-2-1-3-4-1-2 (Shanks, 2003). In this sequence, participants cannot learn to predict the next element on the basis of its raw frequency, because each location occurs an equal number of times (3). Neither can they predict the next element on the basis of the immediately preceding element, because all of the possible successions occur equally (e.g., 2 may be followed by 1, 3, and 4; note that a ubiquitous constraint in SRT tasks is that the target does not appear in the same location on two successive trials). In SOC sequences, two prior elements of context (prior locations) are required in order to fully predict the next target. For instance, 4-2 is always followed by 3, and, in addition, 3 is always preceded by 4-2. SOC sequences are now prevalent; some studies have used first-order conditional (FOC) sequences, in which the nature of any element is fully predicted by a single prior element (e.g., Schvaneveldt & Gomez, 1998, Experiment 1). Note that those sequences are especially easy to learn. Indeed, it is sufficient to learn pairwise relations, and, moreover, the length of the sequence is limited to the number of possible locations. For instance, with four locations, only four-element FOC sequences are possible (e.g., 2-4-3-1).

Generating a Transfer Sequence

The prior section concerned the choice of the to-be-learned sequence. At first glance, examining whether participants' speed improves throughout the session could provide a reliable measure of sequence learning. This is not the case, however, because performance improvement may be due to other factors—for instance, nonspecific familiarization with the task. In order to reveal learning, RTs on the trained sequence need to be compared with RTs on another sequence. This other sequence may be randomly generated. However, the procedure is not optimal, because it is possible for a random sequence to share some of the features displayed in the to-be-learned sequence. The best method consists of carefully selecting another sequence, called the *transfer sequence*. Considering the SOC sequence above (hereafter SOC1), a convenient transfer sequence (hereafter SOC2) would be 4-3-1-2-4-1-3-2-1-4-2-3 (Shanks, 2003; Wilkinson & Shanks, 2004). Worthy of note, SOC1 and SOC2 differ only by their second-order transitions (note that they are also equal with respect to other potentially influential features, such as the number of back-and-forth movements). Thus, comparing RTs on these two sequences should provide a reliable measure of whether participants have learned second-order dependency rules. An additional precaution consists of counterbalancing SOC1 and SOC2—half of the participants being trained with SOC1 (and tested with SOC2), and the other half being trained with SOC2 (and tested with SOC1)—in order to cancel the effect of a possible difference in difficulty between the two sequences. The principles described here for SOC sequences are obviously generalizable to any other sequences.

When Should the Transfer Sequence Be Introduced?

In Nissen and Bullemer (1987), a separate group of participants was trained with random sequences. Beyond

the need for a large number of participants, a between-participants design is ill suited to neuropsychological investigations, in which an individual assessment of learning ability is often desirable. Most recent studies use a within-participants design. However, they often differ with regard to the moment at which the transfer sequence is displayed during the training session.

The most standard procedure consists in showing the transfer sequence toward the end of the training session. Assuming that the whole session is divided into n blocks, the transfer sequence may be introduced on block $n-1$. The RT average on the transfer block is then compared with the RT average on the surrounding blocks (block $n-2$ and block n). Learning is shown by the presence of a selective increase of RTs (or error rate) on block $n-1$ in relation to RTs on the surrounding blocks. Although widespread, this method is limited by the fact that it provides no indication of the time course of learning, hence making results heavily dependent on the (largely arbitrary) choice of n . It is quite possible to imagine that a continuous measure of learning would have revealed differences in the learning curves of two groups of participants who are found to perform at the same level on a final test.

When learning curves appear desirable, displaying the transfer sequence (or parts of it) throughout the training session provides a solution. In a few studies (e.g., Meulemans, Van der Linden, & Perruchet, 1998), a random sequence is intercalated between successive occurrences of the to-be-learned sequence. However, this method requires the use of an ever-changing random sequence to prevent learning of the transfer sequence; we saw above that this choice is not optimal for assessing the exact content of learning. In another method, developed by Schvaneveldt and Gomez (1998), an element of the repeated sequence is randomly substituted with an element of the transfer sequence. Using the SOC1 and SOC2 sequences reported above as training and transfer sequence, respectively, a sample of the final sequence may be, for instance, 3-1-4-2-1-3-4-3-2-4-2-1-... , in which the italicized locations are transfer elements (i.e., they respect the second-order dependency rules of SOC2, whereas the other locations respect the rules of SOC1). Averaging—for each block of trials—the RTs on regular elements on the one hand and the RTs on transfer elements on the other allows us to obtain two separate curves, the difference of which provides evidence of learning.

It is worth stressing that dispatching transfer items within the trained sequence not only opens a window on the level of learning reached at those points, but also changes the to-be-learned material. Instead of being continuously cycled in deterministic ways, the sequence becomes probabilistic. Assuming that 10% of the transfer elements have been randomly introduced, the location of the next target at any point in the sequence can be predicted only with a probability of .90. The few available studies that use both deterministic and probabilistic sequences (e.g., Shanks, Channon, Wilkinson, & Curran, 2006) show that learning occurs for both types of sequences. Nevertheless, unsurprisingly, learning probabilistic sequences appears more difficult. The level of difficulty obviously

depends on the proportion of transfer elements introduced during training. For instance, Schvaneveldt and Gomez (1998) reported that they had failed to obtain a reliable performance improvement with 20% of transfer trials with SOC sequences, but that they were successful with 10%. Shanks et al. (2006) reported quick learning of SOC sequences with 15% of transfer elements—even in amnesic patients. The fact that the repetition structure is less salient is often interpreted as an advantage, with the idea that the difficulty of detecting the repetition structure may disrupt the explicit mode of learning (e.g., Shanks et al., 2006). An additional advantage of probabilistic sequences is that they may be more representative of sequential events in the real world (Hunt & Aslin, 2001).

The Number of Trials per Block and the Number of Blocks

As a rule, the whole training session is divided into blocks of about 100 trials separated by self-paced pauses. The exact number of trials per block often depends on particularities of the repeated sequence. For instance, with 12-element sequences, the length of the block may be set to 96, in order to include eight full sequences. It is also possible to add a few random trials at the beginning of each block, in order to make the repetition structure less salient. The number of blocks is more difficult to select. It depends on the nature of the repeated sequence, the sample of participants, the objective of the researchers, and so on. However, it is worth stressing that studies using a small number of blocks (e.g., Perruchet et al., 1997) or using probabilistic sequences that allow the elaboration of learning curves (e.g., Shanks et al., 2006) have revealed that learning emerges very early during the session, and often does not improve with further practice. Of course, this does not mean that further practice does not induce any changes. It is possible, for instance, that performance in subsequent tests of explicit knowledge depends on the amount of practice. However, if the main objective of the researchers is to give evidence of implicit learning through RT measures, using only a few blocks of trials may be sufficient. For instance, with SOC sequences, three or four 100-trial blocks appear sufficient for getting evidence of reliable learning.

The Response–Stimulus Interval (RSI)

Most SRT studies have used RSIs around 200 or 250 msec. However, a few studies have used longer intervals. For instance, Frensch, Buchner, and Lin (1994) used RSIs of up to 1,500 msec and still observed learning. On the other hand, a few studies have used a 0-msec RSI, with various outcomes. Perruchet et al. (1997) reported no RT improvement with intact explicit knowledge; Destrebecqz and Cleeremans (2001) reported an exactly inverse pattern; and Shanks et al. (2006) reported improved performance both on RTs and on tests of explicit learning. Because these empirical results do not provide us with reliable guidelines for a choice, it may be better to start from theoretical considerations. It may be thought that a long RSI gives participants the opportunity of elaborating conscious strategies, rehearsing the sequence, or engaging

in other controlled activities that are generally construed as being undesirable in the context of implicit learning studies. The choice of RSI values depends on the purpose of the experiment. Indeed, with gSRT-Soft, researchers interested in comparing performance patterns across different levels of awareness have the capability of using a between-participants RSI manipulation.

The Tests of Explicit Knowledge

SRT studies have often been used to compare motor performance in the incidental training task—often construed to be a measure of implicit knowledge—with performance in a subsequent task aimed at capturing participants' explicit knowledge about the structure of the training materials. There is consensus that the test of explicit learning needs to be as sensitive as possible in order to reveal even fleeting evidence of explicit knowledge (see Shanks & St. John, 1994, about the sensitivity criterion). In this regard, a yes/no recognition test appears to be a reasonable choice. Small parts of the repeated sequence, intermixed with small parts of the transfer sequence, are displayed to participants, who have to respond to the target just as they had in the training phase. Note that the resulting RTs provide an additional measure of motor performance that does not necessarily converge with performance data collected during the training session (see Shanks et al., 2006). The length of the to-be-recognized sub-sequences generally comprises between three and six trials. After each sub-sequence, participants are asked to judge whether they had seen the target during the training session.

Various generation tasks are also used to assess the amount of explicit knowledge. In this type of task, the relationship between the target appearance and participants' responses is reversed in such a way that participants' keypresses (or, alternatively, clicks in a target, when the mouse option is selected) elicit the appearance of the selected target on the screen. In free generation tasks, participants are asked to reproduce at best the sequence they saw during the training session. This task may be thought of as a recall task adapted to the case of sequential materials (for a detailed analysis of the resulting data, see Perruchet & Amorim, 1992). However, other instructions are possible. Participants may be told to generate the first sequence that comes to mind, and, in this case, the generation task becomes a nominally implicit task. In the case of cued or trial-by-trial generation tasks, additional cues are given to the participants, whose task is to produce the correct continuation response in the sequence. If performance is at chance, this indicates that participants have learned implicitly. However, there is much evidence that participants are able to generate the learned sequence, suggesting that learning requires at least some explicit knowledge (Perruchet & Amorim, 1992; Shanks & Johnstone, 1999).

The use of this kind of generation task does not show the researcher whether good performance is due only to explicit knowledge or is partly mediated by implicit knowledge. In order to measure the respective contributions of implicit and explicit knowledge, the process dissociation procedure (PDP) initially implemented by Jacoby (1991) can be used. By varying the instructions given to partici-

pants, the procedure is aimed at dissociating conscious and automatic influences. In the inclusion task, participants are asked to generate the learned sequence by remembering it as clearly as possible. The exclusion task, however, requires participants to avoid generating the regularities embedded in the learned sequence. Destrebecqz and Cleeremans (2001, 2003) were the first to conduct a series of experiments applying the PDP to SRT tasks. They demonstrated that, under certain circumstances, participants could not avoid reproducing the learned sequence, despite having been instructed not to do so.

Generation tasks are not implemented in the current release of gSRT-Soft. Nevertheless, actual development of the gSRT-Soft focuses on the implementation of a free generation task and a trial-by-trial generation task. Moreover, these two tasks could be used with inclusion and exclusion instructions. In the case of the free generation task, participants will be instructed to freely generate the learned sequence (inclusion condition) or to generate a different sequence (exclusion condition). In the case of the trial-by-trial generation task, participants will be asked to respond to a five-element sequence and to produce the sixth element. In the inclusion condition, participants are instructed by a green question mark displayed on the screen to produce the next element of the training sequence. In the exclusion condition, participants have to generate a different continuation response. In this case, a red question mark will appear on the screen.

Further releases of gSRT-Soft that include these new features will soon be available.

SUMMARY

In short, the gSRT-Soft should allow the user to run sequence-learning experiments quite easily and with excellent time accuracy. Obviously, the value of the results will depend on the care with which each researcher plans his or her research. We have provided some guidelines for designing experiments involving standard paradigms. There is no doubt that SRT studies, and learning studies in general, raise tricky methodological problems when new issues are explored. We hope that the capabilities of the software will encourage researchers to investigate unexplored directions of research in learning.

Evaluation copies of gSRT-Soft may be obtained free of charge by contacting the authors. Future releases of gSRT-Soft will be published as free software under the terms of the general public license (see www.gnu.org for more details about the GPL). Binary software and source code of the SRT task software will be available from a dedicated Web site. The authors of gSRT-Soft ask that publications involving the use of the original or modified versions of the software cite this article.

AUTHOR NOTE

This work was supported by the Centre National de la Recherche Scientifique (CNRS, UMR 5158 and UMR 5022), the Université de Bourgogne (LE21 and LEAD), and the Region de Bourgogne (AAFE). The authors also thank Axel Cleeremans and the anonymous reviewers

for their help at various stages of elaboration. Correspondence concerning this article should be addressed to S. Chambaron, Cognitive Science Research Unit, Université Libre de Bruxelles, Av. F. D. Roosevelt, 50, CP 191, 1050 Brussels, Belgium (e-mail: schambar@ulb.ac.be).

REFERENCES

- BERRY, D. C., & BROADBENT, D. E. (1984). On the relationship between task performance and associated verbalizable knowledge. *Quarterly Journal of Experimental Psychology*, **36A**, 209-231.
- BISCHOFF-GRETHER, A., GOEDERT, K. M., WILLINGHAM, D. T., & GRAFTON, S. T. (2004). Neural substrates of response-based sequence learning using fMRI. *Journal of Cognitive Neuroscience*, **16**, 127-138.
- CHAMBARON, S., GINHAC, D., & PERRUCHET, P. (2006). Is learning in SRT tasks robust across procedural variations? In R. Sun & N. Miyake (Eds.), *Proceedings of the 28th Annual Conference of the Cognitive Science Society* (pp. 148-153). Mahwah, NJ: Erlbaum.
- CLEEREMANS, A. (1993). Attention and awareness in sequence learning. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society* (pp. 330-335). Hillsdale, NJ: Erlbaum.
- CLEEREMANS, A. (1995). Implicit learning in the presence of multiple cues. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society* (pp. 298-303). Mahwah, NJ: Erlbaum.
- CLEEREMANS, A., & MCCLELLAND, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, **120**, 235-253.
- COHEN, A., IVRY, R. I., & KEELE, S. W. (1990). Attention and structure in sequence learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **16**, 17-30.
- CURRAN, T. (1997). Effects of aging on implicit sequence learning: Accounting for sequence structure and explicit knowledge. *Psychological Research*, **60**, 24-41.
- CURRAN, T., & KEELE, S. W. (1993). Attentional and nonattentional forms of sequence learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **19**, 189-202.
- DEROOST, N., & SOETENS, E. (2006). Spatial processing and perceptual sequence learning in SRT tasks. *Experimental Psychology*, **53**, 16-30.
- DESTREBECQZ, A., & CLEEREMANS, A. (2001). Can sequence learning be implicit? New evidence with the process dissociation procedure. *Psychonomic Bulletin & Review*, **8**, 343-350.
- DESTREBECQZ, A., & CLEEREMANS, A. (2003). Temporal effects in sequence learning. In L. Jiménez (Ed.), *Attention and implicit learning* (pp. 181-213). Amsterdam: John Benjamins.
- FRENSCH, P. A., BUCHNER, A., & LIN, J. (1994). Implicit learning of unique and ambiguous serial transitions in the presence and absence of a distractor task. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **20**, 567-584.
- GOSCHKE, T., & BOLTE, A. (2007). Implicit learning of semantic category sequences: Response-independent acquisition of abstract sequential regularities. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **33**, 394-406.
- HEYES, C. M., & FOSTER, C. L. (2002). Motor learning by observation: Evidence from a serial reaction time task. *Quarterly Journal of Experimental Psychology*, **55A**, 593-607.
- HOWARD, D. V., & HOWARD, J. H., JR. (1989). Age differences in learning serial patterns: Direct versus indirect measures. *Psychology & Aging*, **4**, 357-364.
- HOWARD, J. H., JR., & HOWARD, D. V. (1997). Age differences in implicit learning of higher order dependencies in serial patterns. *Psychology & Aging*, **12**, 634-656.
- HUNT, R. H., & ASLIN, R. N. (2001). Statistical learning in a serial reaction time task: Access to separable statistical cues by individual learners. *Journal of Experimental Psychology: General*, **130**, 658-680.
- JACOBY, L. L. (1991). A process dissociation framework: Separating automatic from intentional uses of memory. *Journal of Memory & Language*, **30**, 513-541.
- JIMÉNEZ, L., MÉNDEZ, C., & CLEEREMANS, A. (1996). Comparing direct and indirect measures of sequence learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **22**, 948-969.
- KOTCHOUBEY, B., HAISS, S., DAUM, I., SCHUGENS, M., & BIRBAUMER, N. (2000). Learning and self-regulation of slow cortical potentials in older adults. *Experimental Aging Research*, **26**, 15-35.
- MCDOWALL, J., & MARTIN, S. (1996). Implicit learning in closed head

- injured subjects: Evidence from an event sequence learning task. *New Zealand Journal of Psychology*, **25**, 1-6.
- MEULEMANS, T., VAN DER LINDEN, M., & PERRUCHET, P. (1998). Implicit sequence learning in children. *Journal of Experimental Child Psychology*, **69**, 199-221.
- MYORS, B. (1999). Timing accuracy of PC programs running under DOS and Windows. *Behavior Research Methods, Instruments, & Computers*, **31**, 322-328.
- NISSEN, M. J., & BULLEMER, P. (1987). Attentional requirements of learning: Evidence from performance measures. *Cognitive Psychology*, **19**, 1-32.
- NORMAN, E., PRICE, M. C., DUFF, S. C., & MENTZONI, R. A. (2007). Gradations of awareness in a modified sequence learning task. *Consciousness & Cognition*, **16**, 809-837.
- OSMAN, M., BIRD, G., & HEYES, C. (2005). Action observation supports effector-dependent learning of finger movement sequences. *Experimental Brain Research*, **165**, 19-27.
- PERRUCHET, P., & AMORIM, M.-A. (1992). Conscious knowledge and changes in performance in sequence learning: Evidence against dissociation. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **18**, 785-800.
- PERRUCHET, P., BIGAND, E., & BENOIT-GONIN, F. (1997). The emergence of explicit knowledge during the early phase of learning in sequential reaction time tasks. *Psychological Research*, **60**, 4-13.
- PERRUCHET, P., & PACTON, S. (2006). Implicit learning and statistical learning: One phenomenon, two approaches. *Trends in Cognitive Sciences*, **10**, 233-238.
- REBER, A. S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning & Verbal Behavior*, **5**, 855-863.
- REED, J., & JOHNSON, P. (1994). Assessing implicit learning with indirect tests: Determining what is learned about sequence structure. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **20**, 585-594.
- SALTHOUSE, T. A., MCGUTHRY, K. E., & HAMBRICK, D. Z. (1999). A framework for analyzing and interpreting differential aging patterns: Application to three measures of implicit learning. *Aging, Neuropsychology, & Cognition*, **6**, 1-18.
- SCHVANEVELDT, R. W., & GOMEZ, R. L. (1998). Attention and probabilistic sequence learning. *Psychological Research*, **61**, 175-190.
- SHANKS, D. R. (2003). Attention and awareness in "implicit" sequence learning. In L. Jiménez (Ed.), *Attention and implicit learning* (pp. 11-42). Amsterdam: John Benjamins.
- SHANKS, D. R. (2005). Implicit learning. In K. Lamberts & R. Goldstone (Eds.), *Handbook of cognition* (pp. 202-220). London: Sage.
- SHANKS, D. R., CHANNON, S., WILKINSON, L., & CURRAN, H. V. (2006). Disruption of sequential priming in organic and pharmacological amnesia: A role for the medial temporal lobes in implicit contextual learning. *Neuropsychopharmacology*, **31**, 1768-1776.
- SHANKS, D. R., & JOHNSTONE, T. (1999). Evaluating the relationship between explicit and implicit knowledge in a sequential reaction time task. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **25**, 1435-1451.
- SHANKS, D. R., & ST. JOHN, M. F. (1994). Characteristics of dissociable human learning systems. *Behavioral & Brain Sciences*, **17**, 367-447.
- SHANKS, D. R., WILKINSON, L., & CHANNON, S. (2003). Relationship between priming and recognition in deterministic and probabilistic sequence learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **29**, 248-261.
- SMITH, J., SIEGERT, R. J., MCDOWALL, J., & ABERNETHY, D. (2001). Preserved implicit learning on both the serial reaction time task and artificial grammar in patients with Parkinson's disease. *Brain & Cognition*, **45**, 378-391.
- STEVENS, A., SCHWARZ, J., SCHWARZ, B., RUF, I., KOLTER, T., & CZEKALLA, J. (2002). Implicit and explicit learning in schizophrenics treated with olanzapine and with classic neuroleptics. *Psychopharmacology*, **160**, 299-306.
- STÖCKER, C., SEBALD, A., & HOFFMANN, J. (2003). The influence of response-effect compatibility in a serial reaction time task. *Quarterly Journal of Experimental Psychology*, **56A**, 685-703.
- VINTER, A., & PERRUCHET, P. (2000). Implicit learning in children is not related to age: Evidence from drawing behavior. *Child Development*, **71**, 1223-1240.
- WILKINSON, L., & SHANKS, D. R. (2004). Intentional control and implicit sequence learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **30**, 354-369.
- WILLINGHAM, D. B., NISSEN, M. J., & BULLEMER, P. (1989). On the development of procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **15**, 1047-1060.
- ZIESSLER, M., & NATTKEMPER, D. (2001). Learning of event sequences is based on response-effect learning: Further evidence from a serial reaction task. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **27**, 595-613.
- ZILLMER, E. A., & SPIERS, M. V. (2001). *Principles of neuropsychology*. Belmont, CA: Wadsworth.

(Manuscript received May 9, 2007;
revision accepted for publication December 21, 2007.)

5 Conclusions et perspectives de recherche

5.1 Bilan des recherches effectuées

Les travaux présentés dans ce mémoire d'Habilitation à Diriger des Recherches s'inscrivent dans la problématique dite d'« Adéquation Algorithme Architecture » en traitement du signal et des images en temps réel. Ils se focalisent sur de multiples aspects tels que l'acquisition d'images par des capteurs dédiés, le développement et la programmation d'architectures matérielles ainsi que l'implantation d'algorithmes de traitement d'images sur ces architectures.

Ce mémoire synthétise environ une douzaine d'années de recherche qui ont commencé en 1995 lors de mon DEA et qui m'ont permis d'acquérir un large spectre de compétences, allant du développement logiciel à la conception microélectronique de circuits intégrés en passant par le domaine de l'électronique numérique et les architectures matérielles de traitement d'images en temps réel. Le dénominateur commun à tous ces travaux est bien évidemment l'axe Adéquation Algorithme Architecture qui a été abordée sous différents aspects. Dans tous les cas, ces travaux de recherche ont toujours été motivés par la volonté de mettre en œuvre des démonstrateurs capables de valider expérimentalement les questions théoriques étudiées.

Trois axes de recherche ont ainsi été abordés dans ce mémoire :

- Le premier axe s'intéresse aux **aspects matériels** de l'Adéquation Algorithme Architecture et vise la conception microélectronique de rétines artificielles. L'idée est d'utiliser au mieux les améliorations des technologies submicroniques afin de pouvoir intégrer des traitements d'images au niveau même des pixels. L'objectif ambitieux de ces travaux est de concevoir de nouvelles générations de capteurs intelligents surclassant les capteurs existants en réalisant des opérations complexes tout en garantissant une embarquabilité maximale.
- Le deuxième axe s'intéresse aux **aspects logiciels** de l'Adéquation Algorithme Architecture et concerne le développement de méthodologies formelles et d'outils permettant d'optimiser l'implantation d'applications de traitement d'images sur des architectures dédiées de calcul. Le prototypage rapide d'applications complexes de traitement d'images à fortes contraintes temporelles (reconnaissance de visages en particulier) a également été longuement abordé depuis mon recrutement en tant que Maître de Conférences à Dijon en 2000.
- Le troisième axe s'intéresse aux **aspects cognitifs** de l'Adéquation Algorithme Architecture et concerne des travaux de recherche en Algorithmie pour les Sciences Cognitives dans le cadre de collaborations avec différentes équipes internationales du domaine. L'objectif est ici de mieux comprendre les mécanismes cognitifs utilisés en apprentissage ou en reconnaissance d'objets.

Les travaux et les résultats présentés dans ce mémoire n'auraient pu être aussi aboutis sans l'aide des 4 doctorants que j'ai coencadrés depuis 2001. Les trois thèses déjà soutenues dans le domaine de la conception CMOS de capteurs d'images nous ont permis d'acquérir un savoir faire et une maîtrise technologique des outils de conception, condition indispensable à la réalisation de capteurs innovants. Les travaux les plus récents nous ont permis de concevoir des capteurs capables de réaliser des traitements d'images programmables bas

niveau à plusieurs milliers d'images par seconde au sein même des pixels. De plus, le soutien du CNRS (ATIP Jeune chercheur en particulier), des ANR Architectures du futur et de la Région Bourgogne (CPER) a permis de financer les premiers prototypes de capteurs et de valider expérimentalement nos choix technologiques. La publication de ces résultats dans les conférences majeures du domaine et dans les revues de notoriété internationale nous a donné une reconnaissance scientifique importante sur le plan national et international dans le domaine de la recherche en Architecture des Systèmes de traitement du Signal et des Images Temps Réel.

5.2 Directions de recherche future

Les thèmes abordés dans ce mémoire bien que se rapportant tous à la problématique d'Adéquation Algorithme Architecture en traitement du signal et des images sont extrêmement divers et pluridisciplinaires. Notre objectif à moyen terme est d'arriver à établir des collaborations fortes entre ces différentes activités. Toutefois, cet objectif de cohésion ne répond pas seulement à des contraintes de confort intellectuel mais également à une complémentarité réelle entre les différents thèmes abordés. L'idée majeure est de recentrer les activités de recherche sur la conception microélectronique de capteurs d'images tout en apportant à ce domaine de recherche un certain nombre d'innovations provenant des autres thèmes abordés.

L'évolution constante des technologies microélectroniques a rendu possible l'intégration dans un seul circuit d'algorithmes sophistiqués de traitement d'images. Historiquement, les premiers traitements intégrés au sein des capteurs étaient de bas niveau et avaient pour objectif de corriger les imperfections introduites par le capteur et d'extraire certaines primitives visuelles. En raison de la loi de Moore qui garantit que l'intégration des transistors sur silicium augmente de 50 % par an, il est aujourd'hui tout à fait envisageable de réaliser des opérations de moyen niveau telles que la localisation et la reconnaissance de formes dans des scènes complexes. Le défi à relever dans les prochaines années sera donc d'intégrer toute la chaîne de traitement de l'image allant de la perception jusqu'à l'interprétation de la scène. Cette interprétation permettra de donner un sens à la scène visuelle, en passant d'une représentation sous forme de pixels à un schéma plus élaboré (correspondant à des descriptions du type « les clés de la voiture sont posées sur la table à côté du journal »).

Si nous voulons donner à une rétine artificielle de telles capacités dans un futur proche, une solution envisageable consiste à s'intéresser de plus près aux systèmes biologiques de perception et de traitement. Il ne s'agit pas seulement de vouloir copier à l'identique le système visuel humain mais plutôt de s'en inspirer et de focaliser nos travaux de recherche sur les mécanismes cognitifs mis en œuvre par les êtres humains pour apprendre, localiser et reconnaître n'importe quel objet dans n'importe quelle scène. Pour cela, il sera nécessaire de développer et d'intégrer au niveau du silicium des fonctions cognitives essentielles telles que l'attention visuelle. L'attention visuelle peut être définie comme la capacité d'un système de vision à sélectionner rapidement les informations les plus pertinentes de l'environnement dans lequel il opère. Le rôle principal de ce mécanisme est d'accélérer le processus de vision en se focalisant sur la zone contenant de l'information utile et en réduisant ainsi sensiblement la quantité d'informations visuelles à traiter. La plupart des méthodes logicielles mises en œuvre dans la littérature sont généralement très coûteuses en opérations de calcul. Les rétines artificielles programmables, grâce à leur parallélisme massif, semblent très bien adaptées pour de telles tâches.

Toutefois, le développement d'un tel capteur passe par la résolution de nombreux problèmes qui restent encore largement ouverts à l'heure actuelle :

Concernant l'aspect « **Adéquation Algorithme Architecture** », la question fondamentale consiste une nouvelle fois à tenter de répondre au problème difficile de partitionnement logiciel / matériel. Dans le cas d'un capteur CMOS intelligent, ce problème se situe à deux granularités différentes, à savoir au niveau le plus fin (plan focal) et au niveau le plus haut (système global). Quelles sont les fonctions de traitement des images qui doivent être intégrées au sein des pixels ? Quelles sont les fonctions qui doivent être positionnées en périphérie du capteur ? Quelles sont celles qui doivent être réalisées de manière matérielle ? Quelles sont celles qui doivent être implémentées sur un processeur programmable ? L'objectif des recherches à mener est de tenter d'apporter des éléments de réponse précis à ces questions. S'il semble impossible de répondre dans un cas général et universel, la possibilité de restreindre l'approche à notre problématique clairement identifiée de conception de capteur intelligent doit permettre d'apporter à court terme des éléments de réponse.

Concernant l'aspect « **Algorithmie pour le traitement d'images** », la volonté de développer un capteur capable d'attention visuelle passe par la mise en œuvre d'algorithmes de vision de plus haut niveau. A l'heure actuelle, nos capteurs implémentent des traitements de bas niveau utilisant des masques de convolution. Ces traitements sont exécutés de manière synchrone en parallèle sur l'ensemble des pixels. L'idée est maintenant d'aller plus loin et de proposer des circuits pouvant exécuter des traitements de moyen niveau sur des régions d'intérêt (étiquetage en composantes connexes, chaînage de points contours, ...). De tels traitements sont par nature irréguliers en complexité et en temps d'exécution car ils dépendent des données. Ceci imposera de mettre en œuvre des stratégies de distribution des données, de collecte des résultats, de synchronisation entre toutes les ressources de calculs mises en œuvre.

Concernant l'aspect « **Sciences cognitives** », l'accent doit être mis sur le développement de modèles connexionnistes donnant au système de vision des capacités d'analyse et de décision de très haut niveau. L'idée est de mettre en place un système expert capable d'interagir avec son environnement, de choisir quelles sont les zones de l'image à analyser, quels sont les algorithmes à appliquer sur ces zones.

Concernant l'aspect « **Conception microélectronique** », les directions de recherche sont bien évidemment multiples pour tenter d'implanter sur le silicium les fonctionnalités précédemment énoncées. Premièrement, sur le plan technologique, une piste à suivre est la mise à disposition de nouvelles technologies 3D permettant d'empiler des photodiodes au dessus des transistors. Une telle approche permet d'une part d'augmenter de manière considérable le taux de remplissage des pixels (jusqu'à 100%) et d'autre part libère de la surface de silicium au sein de chaque pixel afin d'intégrer de nouvelles fonctions au sein des processeurs élémentaires. Deuxièmement, l'utilisation optimale de la surface passe inévitablement par la mise au point de processeurs numériques conçus à base de transistors aux dimensions minimales, impliquant donc de convertir au plus tôt le signal analogique issu de la photodiode en signal numérique. Des travaux sur la conversion analogique numérique et sur le stockage en mémoire devront être menés préliminairement à tout développement de processeur numérique. Enfin, des réflexions plus approfondies doivent être engagées concernant ces processeurs élémentaires. Une piste intéressante serait de pouvoir adapter dynamiquement la puissance de calcul des processeurs en fonction du contexte algorithmique et des données à traiter. En effet, doter une rétine artificielle de la possibilité de consacrer l'intégralité de ses ressources de calcul sur une ou plusieurs zones particulières de l'image constitue une approche très innovante en donnant l'opportunité d'implanter des traitements d'images de moyen niveau. Pour cela, une piste potentielle serait

de créer au sein de chaque pixel un processeur élémentaire qui constitue la brique de base minimale pour effectuer des traitements. L'assemblage dynamique de ces processeurs par groupes de 2, 4 ou plus permettrait de concevoir des structures de calculs plus puissantes, capables de répondre aux fortes contraintes temporelles.

Au final, le défi majeur se situe au niveau de l'intégration des travaux menés dans toutes ces directions (logicielle, matérielle et cognitive) dans le but de modéliser et de concevoir une rétine artificielle dotée de fonctions bio inspirées de haut niveau lui permettant d'appréhender, d'apprendre et d'interpréter une scène visuelle complexe.