



**HAL**  
open science

# Capteur d'images événementiel, asynchrone à échantillonnage non-uniforme

Amani Darwish

► **To cite this version:**

Amani Darwish. Capteur d'images événementiel, asynchrone à échantillonnage non-uniforme. Traitement du signal et de l'image [eess.SP]. Université Grenoble Alpes, 2016. Français. NNT : 2016GREAT071 . tel-01493868

**HAL Id: tel-01493868**

**<https://theses.hal.science/tel-01493868>**

Submitted on 22 Mar 2017

**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.

## THÈSE

Pour obtenir le grade de

### **DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES**

Spécialité : **NanoElectronique et NanoTechnologie**

Arrêté ministériel : 7 août 2006

Présentée par

**Amani Darwish**

Thèse dirigée par **Gilles Sicard** et  
codirigée par **Laurent Fesquet**

préparée au sein du **laboratoire de Techniques de  
l'Informatique et de la Microélectronique pour l'Architecture  
des systèmes intégrés (TIMA)**

dans **l'École Doctorale Électronique, Électrotechnique,  
Automatique et Traitement du Signal (E.E.A.T.S)**

# **Capteur d'images événementiel asynchrone à échantillonnage non-uniforme**

Thèse soutenue publiquement le **27 juin 2016**  
devant le jury composé de :

**M. Yves Leduc**

Professeur à Polytech'Nice-Sophia (UNS), Président

**M. Michel Paindavoine**

Professeur à l'université de Bourgogne, Rapporteur

**M. Olivier Sentieys**

Professeur à l'université de Rennes, Rapporteur

**M. Gilles Sicard**

Chercheur au CEA-LETI, HDR, Directeur de thèse

**M. Laurent Fesquet**

Maître de conférences à l'Université de Grenoble, HDR, Co-Directeur de  
thèse





ISBN : 978-2-11-129215-4



# REMERCIEMENTS

Mon parcours a commencé, il y a presque cinq ans, le 2 septembre 2011 quand je suis arrivée à Grenoble dans le but de poursuivre mes études dans cette très belle ville dynamique et toujours en cours de développement. Ces dernières années me sont extrêmement chères au cœur, elles étaient riches et inoubliables ...

A la fin de ma thèse, je remercie chaleureusement et sincèrement mes directeurs de thèse, Gilles Sicard et Laurent Fesquet. Gilles, je te remercie de m'avoir initialement accueillie au groupe CIS lors de mon stage de recherche à la fin de mon master. Pendant ce stage, j'ai commencé ma découverte du monde des imageurs qui s'est poursuivie en cours de ma thèse. Mon exploration s'est prolongée avec Laurent dans le domaine de circuits asynchrones, ce domaine non conventionnel et dynamique que j'ai tant aimé et apprécié. Vous m'avez donné l'opportunité de faire une thèse et vous m'avez laissée chercher, essayer et expérimenter avec mes propres idées. Je vous remercie pour votre support, votre aide et votre patience pendant les années de cette thèse. Je vous remercie pour toute la confiance que vous m'avez accordée et tout l'effort que vous avez fait.

Je remercie aussi les membres du jury : messieurs les professeurs M. Michel Paindavoine et M. Olivier Sentieys qui m'ont fait l'honneur d'être les rapporteurs de cette thèse, ainsi que M. Yves Leduc, le président du jury.

Je tiens également à remercier Madame Dominique Borrione de m'avoir initialement accueillie dans son laboratoire de recherche en début de ma thèse. Je tiens à remercier M. Salvador Mir, le directeur actuel du laboratoire TIMA. Je remercie également l'équipe administrative de TIMA qui nous facilite toujours la vie quotidienne au laboratoire.

Ensuite, j'adresse mes remerciements à l'équipe CIME NANOTECH pour tout le support logistique notamment à M. Alexandre Chagoya pour son soutien mais aussi pour sa disponibilité, son attention et sa gentillesse qui m'a toujours marquée.

Je continue en remerciant les membres de l'équipe de recherche CIS. J'ai passé des moments agréables avec vous autour du café d'après-midi ainsi que pendant nos escapades en dehors du laboratoire ! Je tiens spécialement à remercier mes compatriotes, mes collègues du domaine et mes amis : Hawra'a et Hassan. Je vous remercie pour tous les conseils que vous m'avez donnés ainsi que pour toutes les longues discussions qu'on a eues, à propos l'électronique, les imageurs et la photographie...

Un grand merci à mes amis au Liban...mes amis d'enfance, mes amis du lycée avec qui j'ai partagé nos premières années, nos premières ambitions et nos premiers rêves. Je remercie également tous mes amis en France.

J'avoue que c'est extrêmement compliqué pour moi de remercier ma famille et particulièrement mes parents qui leurs amours et leurs tendresses infinies m'ont donnée des racines et des ailes. Mes mots seront toujours incapables d'exprimer ma gratitude

## REMERCIEMENTS

envers vos scarifies pendant toutes ces années, votre aide et votre soutien qui m'a permis de voyager, de réaliser un de mes rêves en obtenant mon doctorat et de devenir la personne que je suis ! Malgré la distance entre nous, vous êtes et vous seriez toujours dans mon cœur et dans mon esprit.

Ma pensée et mes remerciements passent naturellement à mes chers frères, Yehya et Mohamad Mahdi, qui étaient toujours à mes côtés à booster ma morale et à m'encourager. Je vous remercie !

Finalement, je tiens à remercier chaleureusement Firas, mon mari et mon meilleur ami, avec qui j'ai partagé nos premiers pas dans le monde des adultes, nos études, nos années en France ainsi que de nombreux moments inoubliables. Firas, ton humour très particulier m'a été toujours nécessaire voire indispensable pendant tous les passages compliqués... Je t'en remercie !

# TABLE DES MATIÈRES

|   |             |
|---|-------------|
| <b>Table des figures</b>  | <b>v</b>    |
| <b>Liste des tableaux</b>   | <b>ix</b>   |
| <b>Resumé</b>   | <b>xi</b>   |
| <b>Abstract</b>   | <b>xiii</b> |
| <b>Chapitre 1 : Introduction</b>                                  | <b>1</b>    |
| <b>Partie I: État de l'art</b>                                    | <b>5</b>    |
| <b>Chapitre 2 : État de l'art des capteurs d'images</b>           | <b>7</b>    |
| 2.1 Les capteurs d'images électroniques . . . . .                 | 8           |
| 2.1.1 Les capteurs d'images CCD . . . . .                         | 8           |
| 2.1.2 Les capteurs d'images CMOS . . . . .                        | 11          |
| 2.2 La photodiode . . . . .                                       | 11          |
| 2.3 Les pixels des capteurs d'images CMOS . . . . .               | 13          |
| 2.3.1 Les pixels passifs (PPS) . . . . .                          | 13          |
| 2.3.2 Les pixels actifs (APS) . . . . .                           | 14          |
| 2.4 Chaîne de lecture du capteur d'images CMOS . . . . .          | 18          |
| 2.4.1 Les décodeurs d'adresses . . . . .                          | 18          |
| 2.4.2 Amplificateur Colonne . . . . .                             | 18          |
| 2.4.3 Conversion analogique numérique . . . . .                   | 18          |
| 2.5 Caractéristiques générales du capteur d'images CMOS . . . . . | 20          |
| 2.5.1 Le facteur de remplissage . . . . .                         | 20          |
| 2.5.2 La sensibilité du pixel . . . . .                           | 20          |
| 2.5.3 La dynamique de fonctionnement . . . . .                    | 21          |
| 2.5.4 La résolution . . . . .                                     | 21          |
| 2.6 Limitations des imageurs conventionnels . . . . .             | 22          |
| 2.6.1 Flots de données en sortie . . . . .                        | 22          |
| 2.6.2 Conversion analogique numérique . . . . .                   | 23          |
| 2.6.3 Dynamique de fonctionnement . . . . .                       | 23          |
| 2.7 Les capteurs d'images asynchrones . . . . .                   | 24          |
| 2.7.1 Les systèmes neuromorphiques . . . . .                      | 24          |
| 2.7.2 Principe des imageurs asynchrones . . . . .                 | 25          |

|  |   |           |
|--|---|-----------|
| 2.7.3  | Protocole de lecture asynchrone : Address Event Representation ( <i>AER</i> ) . . . . . | 26        |
| 2.7.4  | Codage de l'information lumineuse du pixel . . . . .                                    | 29        |
| 2.8  | Conclusion . . . . .  | 33        |
| <b>Chapitre 3 : La Logique Asynchrone</b>        |   | <b>35</b> |
| 3.1  | Problématique des circuits synchrones . . . . .   | 36        |
| 3.2  | Conception des circuits asynchrones . . . . .   | 37        |
| 3.2.1  | Principe . . . . .  | 37        |
| 3.2.2  | La porte Muller . . . . .   | 37        |
| 3.3  | Protocole de communication . . . . .  | 38        |
| 3.3.1  | Protocole à données intégrées . . . . .   | 39        |
| 3.3.2  | Protocole double rails à quatre phases . . . . .  | 39        |
| 3.4  | Classes des circuits asynchrones . . . . .  | 42        |
| 3.4.1  | Insensible aux délais (Delay Inensitive (DI)) . . . . .                                 | 42        |
| 3.4.2  | Quasi-Insensible aux délais (Quasi-Delay Inensitive (QDI)) . . . . .                    | 42        |
| 3.4.3  | Indépendant de la vitesse (Speed Independent (SI)) . . . . .                            | 43        |
| 3.4.4  | Micropipeline . . . . .   | 43        |
| 3.5  | Caractéristiques des circuits asynchrones . . . . .                                     | 46        |
| 3.5.1  | Absence du signal d'horloge . . . . .   | 46        |
| 3.5.2  | Calcul à temps minimal . . . . .  | 46        |
| 3.5.3  | Consommation électrique . . . . .   | 47        |
| 3.5.4  | Modularité . . . . .  | 47        |
| 3.5.5  | Migration technologique . . . . .   | 47        |
| 3.6  | Conclusion . . . . .  | 48        |
| <b>Chapitre 4 : Échantillonnage Non-Uniforme</b> |   | <b>49</b> |
| 4.1  | La théorie de l'échantillonnage . . . . .   | 50        |
| 4.2  | Théorème de Shannon : L'échantillonnage Uniforme . . . . .                              | 51        |
| 4.2.1  | La Quantification . . . . .   | 52        |
| 4.3  | Échantillonnage Non Uniforme . . . . .  | 53        |
| 4.3.1  | Principe . . . . .  | 53        |
| 4.3.2  | Techniques de mise en œuvre . . . . .   | 54        |
| 4.4  | Réduction du flot de données . . . . .  | 58        |
| 4.4.1  | 1 Dimension . . . . .   | 59        |
| 4.4.2  | Extension à 2 Dimensions . . . . .  | 60        |
| 4.5  | Conclusion . . . . .  | 61        |
| <b>Partie II: Fonctionnement Général</b>         |   | <b>63</b> |
| <b>Chapitre 5 : Approches et contributions</b>   |   | <b>65</b> |

|   |  |            |
|---|--|------------|
| 5.1   | Conception des circuits numériques en logique asynchrone . . . . . | 66         |
| 5.2   | Protocole de lecture du capteur d'images . . . . .                 | 67         |
| 5.3   | Suppression des redondances spatiales et temporelles . . . . .     | 67         |
| 5.3.1   | Redondances Temporelles . . . . .                                  | 69         |
| 5.3.2   | Redondances Spatiales . . . . .                                    | 70         |
| 5.4   | Conclusion . . . . .   | 70         |
| <b>Chapitre 6 : Fonctionnement de l'imageur événementiel asynchrone</b> |  | <b>73</b>  |
| 6.1   | Chaîne de lecture de l'imageur événementiel asynchrone . . . . .   | 73         |
| 6.2   | Système de lecture asynchrone . . . . .                            | 76         |
| 6.2.1   | Architecture globale du système de lecture . . . . .               | 76         |
| 6.2.2   | Bloc d'horodatage (Time Stamping Bloc) . . . . .                   | 78         |
| 6.2.3   | Architecture du bloc de traitement des requêtes de lecture . . .   | 79         |
| 6.3   | Réduction des redondances spatiales . . . . .                      | 82         |
| 6.4   | Le pixel événementiel TFS . . . . .                                | 84         |
| 6.4.1   | Fonctionnement du pixel événementiel . . . . .                     | 85         |
| 6.4.2   | Niveau de détection des événements . . . . .                       | 85         |
| 6.4.3   | Initialisation locale . . . . .                                    | 87         |
| 6.4.4   | Temps mort du pixel . . . . .                                      | 87         |
| 6.5   | Conclusion . . . . .   | 88         |
| <b>Chapitre 7 : Simulation Haut Niveau</b>                              |  | <b>89</b>  |
| 7.1   | Images numériques . . . . .  | 89         |
| 7.2   | Simulation Matlab . . . . .  | 90         |
| 7.2.1   | Protocole de simulation . . . . .                                  | 90         |
| 7.2.2   | Modélisation de $\Delta_t$ . . . . .                               | 91         |
| 7.2.3   | Choix des images numériques . . . . .                              | 92         |
| 7.3   | Discussion des résultats . . . . .                                 | 97         |
| 7.4   | Conclusion . . . . .   | 103        |
| <b>Partie III: Design</b>   |  | <b>105</b> |
| <b>Chapitre 8 : Conception du circuit de lecture asynchrone</b>         |  | <b>107</b> |
| 8.1   | Système de lecture . . . . .                                       | 107        |
| 8.1.1   | Encodeur - Bloc de vérification . . . . .                          | 107        |
| 8.1.2   | Logique combinatoire . . . . .                                     | 112        |
| 8.1.3   | Mémoire tampon : FIFO . . . . .                                    | 112        |
| 8.1.4   | Simulation VHDL . . . . .  | 113        |
| 8.2   | Simulation au niveau netlist . . . . .                             | 115        |
| 8.2.1   | Simulation MATLAB : Création des requêtes de lecture . . . .       | 117        |
| 8.2.2   | Simulation VHDL niveau netlist . . . . .                           | 119        |

|   |   |            |
|---|---|------------|
| 8.2.3   | Simulation MATLAB : Reconstruction des images résultantes . . . . . | 119        |
| 8.2.4   | Résultats . . . . .   | 119        |
| 8.3   | Architecture du pixel événementiel . . . . .                        | 122        |
| 8.4   | Conclusion . . . . .  | 123        |
| <b>Chapitre 9 : Conclusion et Perspectives</b>        |   | <b>125</b> |
| 9.1   | Conclusion . . . . .  | 125        |
| 9.2   | Perspectives . . . . .  | 128        |
| <b>Annexe A : Suite des simulations du chapitre 7</b> |   | <b>131</b> |
| <b>Annexe B : Suite des simulations du chapitre 8</b> |   | <b>153</b> |
| <b>Annexe C : Publications de l'auteur</b>            |   | <b>163</b> |
| C.1   | Conférences Internationales . . . . .                               | 163        |
| C.2   | Conférences Invitées . . . . .                                      | 163        |
| C.3   | Conférences Nationales . . . . .                                    | 164        |
| <b>Bibliographie</b>                                  |   | <b>165</b> |

## TABLE DES FIGURES

|             |   |    |
|-------------|---|----|
| FIGURE 2.1  | Répartition du marché des capteurs d'images par applications . . .  | 9  |
| FIGURE 2.2  | Essai du premier capteur d'images électronique CCD par George E. Smith et Willard Boyle au laboratoire Bell . . . . .   | 10 |
| FIGURE 2.3  | Principe de transfert de charges dans un capteur d'images CCD   | 10 |
| FIGURE 2.4  | Exemple d'une photodiode polarisée en inverse (a) Le courant photogénéré en fonction de la tension de polarisation $u$ et le nombre de photons incidents $\epsilon$ (b) . . . . . | 12 |
| FIGURE 2.5  | Le circuit électrique équivalent de la photodiode en mode photoconducteur . . . . .   | 12 |
| FIGURE 2.6  | Pixel Passif ( <i>Passive Pixel Sensor : PPS</i> ) . . . . .  | 13 |
| FIGURE 2.7  | Pixel Actif ( <i>Actif Pixel Sensor : APS</i> ) . . . . .   | 15 |
| FIGURE 2.8  | Diagramme de fonctionnement d'un pixel actif standard ( <i>APS</i> )  | 15 |
| FIGURE 2.9  | Pixel logarithmique basique . . . . .   | 17 |
| FIGURE 2.10 | Réponse typique d'un pixel logarithmique[CC07] . . . . .  | 17 |
| FIGURE 2.11 | Architecture d'un capteur d'images CMOS conventionnel . . . .   | 19 |
| FIGURE 2.12 | Différence entre le protocole de lecture d'un capteur d'images conventionnel et celui d'un capteur d'images asynchrones . . . .   | 26 |
| FIGURE 2.13 | Le protocole de communication d'événement par adresse [Mah92]   | 27 |
| FIGURE 2.14 | L'architecture typique d'un circuit de lecture AER pour un imageur événementiel [Mya+11] . . . . .  | 28 |
| FIGURE 2.15 | Le pixel événementiel <i>TFS</i> . . . . .  | 31 |
| FIGURE 3.1  | Schéma fonctionnel d'un circuit asynchrone . . . . .  | 37 |
| FIGURE 3.2  | La Porte Müller . . . . .   | 38 |
| FIGURE 3.3  | Protocole de communication à données intégrées . . . . .  | 40 |
| FIGURE 3.4  | Protocole de communication double rails à quatre phases . . . .   | 41 |
| FIGURE 3.5  | Codage du protocole de communication double rails à quatre phases . . . . .   | 41 |
| FIGURE 3.6  | Structure de base du circuit de contrôle d'un micropipeline . . .   | 44 |
| FIGURE 3.7  | Micropipeline avec blocs combinatoires . . . . .  | 44 |
| FIGURE 3.8  | Micropipeline indépendant de la vitesse . . . . .   | 46 |
| FIGURE 4.1  | Échantillonnage Uniforme d'un signal aléatoire . . . . .  | 52 |
| FIGURE 4.2  | Échantillonnage aléatoire additif . . . . .   | 55 |
| FIGURE 4.3  | Échantillonnage uniforme à sauts aléatoires . . . . .   | 55 |
| FIGURE 4.4  | Échantillonnage par croisement de signal : <i>Signal Crossing Sampling</i> . . . . .  | 56 |

|             |  |     |
|-------------|--|-----|
| FIGURE 4.5  | Échantillonnage par traversée de niveaux : <i>Level Crossing Sampling</i> . . . . .  | 57  |
| FIGURE 5.1  | Exemple d'une image spatialement redondante . . . . .  | 68  |
| FIGURE 5.2  | Exemple d'une séquence d'images successives temporellement redondantes . . . . .   | 69  |
| FIGURE 5.3  | Récapitulatif de contributions proposées au cours de cette thèse   | 72  |
| FIGURE 6.1  | Les phases de fonctionnement principales du pixel événementiel   | 74  |
| FIGURE 6.2  | Organigramme du fonctionnement du système de lecture . . .   | 75  |
| FIGURE 6.3  | Architecture globale du système de lecture numérique asynchrone  | 77  |
| FIGURE 6.4  | Architecture du bloc d'horodatage ( <i>Time Stamping Bloc</i> ) . . .  | 78  |
| FIGURE 6.5  | Architecture du bloc de traitement asynchrone des requêtes de lecture . . . . .  | 79  |
| FIGURE 6.6  | La signalisation partagée entre les pixels d'un capteur d'images   | 81  |
| FIGURE 6.7  | Chronogramme de trois pixels événementiels . . . . .   | 83  |
| FIGURE 6.8  | Organigramme du fonctionnement du pixel événementiel . . .   | 86  |
| FIGURE 7.1  | Exemple d'une image numérique . . . . .  | 90  |
| FIGURE 7.2  | Fusionnement des niveaux de gris avec l'augmentation de la valeur du coefficient de réduction des redondances spatiales $\Delta_m$   | 91  |
| FIGURE 7.3  | Variation du nombre des niveaux de gris par rapport à $\Delta_m$ . . .   | 92  |
| FIGURE 7.4  | Résultat des simulations MATLAB : Image A . . . . .  | 93  |
| FIGURE 7.5  | Résultat des simulations MATLAB : Image B . . . . .  | 94  |
| FIGURE 7.6  | Résultat des simulations MATLAB : Image C . . . . .  | 95  |
| FIGURE 7.7  | Résultat des simulations MATLAB : Image D . . . . .  | 96  |
| FIGURE 7.8  | Évaluation de la qualité des images résultantes de simulations de l'image A en fonction de $\Delta_m$ . . . . .  | 99  |
| FIGURE 7.9  | Évaluation de la qualité des images résultantes de simulations de l'image B en fonction de $\Delta_m$ . . . . .  | 100 |
| FIGURE 7.10 | Évaluation de la qualité des images résultantes de simulations de l'image C en fonction de $\Delta_m$ . . . . .  | 101 |
| FIGURE 7.11 | Évaluation de la qualité des images résultantes de simulations de l'image D en fonction de $\Delta_m$ . . . . .  | 102 |
| FIGURE 8.1  | Architecture du système de lecture . . . . .   | 108 |
| FIGURE 8.2  | Modèle d'une machine à états finis de type Moore . . . . .   | 110 |
| FIGURE 8.3  | Pipeline synchrone à trois étages . . . . .  | 110 |
| FIGURE 8.4  | Pipeline asynchrone à trois étages . . . . .   | 111 |
| FIGURE 8.5  | Étage d'un micropipeline WCHB linéaire . . . . .   | 112 |
| FIGURE 8.6  | Architecture de la mémoire tampon de type FIFO . . . . .   | 114 |
| FIGURE 8.7  | Chronogramme du système de lecture. (a) La signalisation entre les composantes du bloc de traitement des requêtes de lecture et le bloc d'horodatage (b) Vue élargie de la phase de vérification | 116 |

|             |   |     |
|-------------|---|-----|
| FIGURE 8.8  | Représentation du flot de simulation du circuit de lecture asynchrone par des images numériques . . . . .       | 117 |
| FIGURE 8.9  | Les images de référence utilisées pour les simulations du circuit de lecture niveau <i>netlist</i> . . . . .    | 118 |
| FIGURE 8.10 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9h . . . . .              | 121 |
| FIGURE 8.11 | Architecture du pixel événementiel . . . . .  | 122 |
| FIGURE A.1  | Résultat des simulations MATLAB : Image E . . . . .   | 132 |
| FIGURE A.2  | Évaluation de la qualité des images résultantes de simulations de l'image E en fonction de $\Delta_m$ . . . . . | 133 |
| FIGURE A.3  | Résultat des simulations MATLAB : Image F . . . . .   | 134 |
| FIGURE A.4  | Évaluation de la qualité des images résultantes de simulations de l'image F en fonction de $\Delta_m$ . . . . . | 135 |
| FIGURE A.5  | Résultat des simulations MATLAB : Image G . . . . .   | 136 |
| FIGURE A.6  | Évaluation de la qualité des images résultantes de simulations de l'image G en fonction de $\Delta_m$ . . . . . | 137 |
| FIGURE A.7  | Résultat des simulations MATLAB : Image H . . . . .   | 138 |
| FIGURE A.8  | Évaluation de la qualité des images résultantes de simulations de l'image H en fonction de $\Delta_m$ . . . . . | 139 |
| FIGURE A.9  | Résultat des simulations MATLAB : Image I . . . . .   | 140 |
| FIGURE A.10 | Évaluation de la qualité des images résultantes de simulations de l'image I en fonction de $\Delta_m$ . . . . . | 141 |
| FIGURE A.11 | Résultat des simulations MATLAB : Image J . . . . .   | 142 |
| FIGURE A.12 | Évaluation de la qualité des images résultantes de simulations de l'image J en fonction de $\Delta_m$ . . . . . | 143 |
| FIGURE A.13 | Résultat des simulations MATLAB : Image K . . . . .   | 144 |
| FIGURE A.14 | Évaluation de la qualité des images résultantes de simulations de l'image K en fonction de $\Delta_m$ . . . . . | 145 |
| FIGURE A.15 | Résultat des simulations MATLAB : Image L . . . . .   | 146 |
| FIGURE A.16 | Évaluation de la qualité des images résultantes de simulations de l'image L en fonction de $\Delta_m$ . . . . . | 147 |
| FIGURE A.17 | Résultat des simulations MATLAB : Image M . . . . .   | 148 |
| FIGURE A.18 | Évaluation de la qualité des images résultantes de simulations de l'image M en fonction de $\Delta_m$ . . . . . | 149 |
| FIGURE A.19 | Résultat des simulations MATLAB : Image N . . . . .   | 150 |
| FIGURE A.20 | Évaluation de la qualité des images résultantes de simulations de l'image N en fonction de $\Delta_m$ . . . . . | 151 |
| FIGURE B.1  | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9a . . . . .              | 154 |
| FIGURE B.2  | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9b . . . . .              | 155 |

|            |  |     |
|------------|--|-----|
| FIGURE B.3 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9c . . . . . | 156 |
| FIGURE B.4 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9d . . . . . | 157 |
| FIGURE B.5 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9e . . . . . | 158 |
| FIGURE B.6 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9f . . . . . | 159 |
| FIGURE B.7 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9g . . . . . | 160 |
| FIGURE B.8 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9h . . . . . | 161 |
| FIGURE B.9 | Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9i . . . . . | 162 |

## LISTE DES TABLEAUX

|           |   |     |
|-----------|---|-----|
| TABLE 4.1 | Comparaison entre les techniques d'échantillonnage non-uniforme   | 59  |
| TABLE 4.2 | Caractéristiques de l'échantillonnage uniforme et de l'échantillonnage non uniforme par traversée de niveaux . . . . .                            | 61  |
| TABLE 8.1 | Résultats des simulations niveau <i>netlist</i> des images de référence 8.9 . . . . .   | 120 |
| TABLE 9.1 | Comparaison entre notre capteur d'images événementiel et asynchrone et les principaux capteurs d'images événementiels de la littérature . . . . . | 127 |
| TABLE B.1 | Résultats des simulations niveau <i>netlist</i> des images de référence 8.9 . . . . .   | 153 |



# RESUMÉ

Face aux défis actuels liés à la conception de capteurs d'images à forte résolution comme la limitation de la consommation électrique, l'augmentation du flux de données ainsi que le traitement de données associé, on propose, à travers cette thèse, un capteur d'image novateur asynchrone à échantillonnage non uniforme. Ce capteur d'images asynchrone est basé sur une matrice de pixels événementiels qui intègrent un échantillonnage non uniforme par traversée de niveaux. Contrairement aux imageurs conventionnels, où les pixels sont lus systématiquement lors de chaque trame, les pixels événementiels proposés sont consultés que lorsqu'ils contiennent une information pertinente. Cela induit un flux de données réduit et dépendant de l'image.

Pour compléter la chaîne de traitement des pixels, on présente également une architecture numérique de lecture dédiée conçue en utilisant de la logique asynchrone et destinée à contrôler et à gérer le flux de données des pixels événementiels. Ce circuit de lecture numérique permet de surmonter les difficultés classiques rencontrées lors de la gestion des demandes simultanées des pixels événementiels sans dégrader la résolution et le facteur de remplissage du capteur d'images. En outre, le circuit de lecture proposé permet de réduire considérablement les redondances spatiales dans une image ce qui diminue encore le flux de données.

Enfin, en combinant l'aspect échantillonnage par traversée de niveau et la technique de lecture proposée, on a pu remplacer la conversion analogique numérique classique de la chaîne de traitement des pixels par une conversion temps-numérique (*Time-to-Digital Conversion*). En d'autre terme, l'information du pixel est codée par le temps. Il en résulte une diminution accrue de la consommation électrique du système de vision, le convertisseur analogique-numérique étant un des composants les plus consommant du système de lecture des capteurs d'images conventionnels.

**Mots Clés** – *Capteur d'images CMOS, Échantillonnage Non Uniforme, Événementiel, Logique asynchrone, Réduction du flux de données, Conversion Tension-Temps*



# ABSTRACT

In order to overcome the challenges associated with the design of high resolution image sensors, we propose, through this thesis, an innovative asynchronous event-driven image sensor based on non-uniform sampling. The proposed image sensor aims the reduction of the data flow and its associated data processing by limiting the activity of our image sensor to the new captured information.

The proposed asynchronous image sensor is based on an event-driven pixels that incorporate a non-uniform sampling crossing levels. Unlike conventional imagers, where the pixels are read systematically at each frame, the proposed event-driven pixels are only read when they hold new and relevant information. This induces a reduced and scene dependent data flow.

In this thesis, we introduce a complete pixel reading sequence. Beside the event-driven pixel, the proposed reading system is designed using asynchronous logic and adapted to control and manage the flow of data from event pixels. This digital reading system overcomes the traditional difficulties encountered in the management of simultaneous requests for event pixels without degrading the resolution and fill factor of the image sensor. In addition, the proposed reading circuit significantly reduces the spatial redundancy in an image which further reduces the data flow.

Finally, by combining the aspect of level crossing sampling and the proposed reading technique, we replaced the conventional analog to digital conversion of the pixel processing chain by a time-to-digital Conversion (TDC). In other words, the pixel information is coded by time. This results in an increased reduction in power consumption of the vision system, the analog-digital converter being one of the most consuming reading system of conventional image sensors components.

**Keywords** – *Image Sensors, CMOS, Event-driven, Asynchronous, Non Uniform Sampling, Reduced Data Flow, Level Sampling Scheme, Time-to-Digital Conversion*



# 1

## INTRODUCTION

Dans un monde où la gestion de la consommation électrique est une préoccupation essentielle et devenant de plus en plus critique, les composants électroniques intègrent très souvent des modes de fonctionnement basse consommation (*Low Power*). Les capteurs d'images ont connu, durant ces dernières années, des évolutions majeures notamment au niveau de la qualité de la photoconversion, de la résolution, du bruit, de la vitesse de fonctionnement, et du traitement d'images intégré. Paradoxalement, la consommation électrique de l'imageur demeure très élevée par rapport à d'autres composants électroniques et la minimisation de celle-ci commence juste à être un point d'intérêt pour les industriels qui se focalisaient sur son optimisation afin que les imageurs à forte résolution présentent une consommation raisonnable. De plus, les imageurs, étant intégrés dans la majorité des applications électroniques portables industrielles, médicales et grand public, il est devenu impératif de développer des imageurs à faible consommation électrique. Tout le monde a en effet été déjà confronté à cette problématique, en prenant par exemple des photos avec son téléphone ; on constate que la vitesse de décharge de la batterie augmente !

Pour réduire la consommation électrique d'un imageur conventionnel, on distingue usuellement deux axes de recherche.

Le premier axe consiste à développer un imageur *Low-Power*. En d'autres termes, l'effort consiste ici à optimiser l'architecture de l'imageur et plus particulièrement son convertisseur analogique-numérique qui est l'élément le plus consommant dans le système de lecture de l'imageur. Cette approche exige souvent l'utilisation d'une technologie de fabrication récente et donc chère. Plusieurs équipes de recherche dans le monde travaillent sur cette approche, ce qui explique la présence d'une grande diversité de méthodes et de publications dans la littérature. Cependant, il n'existe toujours pas d'imageurs en production réellement conçus pour la faible consommation.

La deuxième approche consiste à rendre le capteur d'images énergiquement plus efficace en modifiant son protocole de lecture qui consiste à venir lire, pour chaque trame, l'ensemble des pixels. Cette méthode de lecture, séquentielle et exhaustive, génère un flot de données constant, et n'a pas vraiment évolué au cours des dernières années. Cela devient cependant de plus en plus compliqué de maintenir ce mode de lecture (certes confortable) face à l'accroissement continu de la résolution et de la vitesse de lecture des imageurs. Dans ce cadre, les technologies, dites 3D, qui consiste à empiler l'imageur et un circuit dédié à la lecture et à la conversion, sont de plus en plus utilisés, mais elle augmente les coûts et pas vraiment la consommation.

D'une manière générale, un circuit sera considéré comme énergiquement très efficace

s'il limite sa consommation électrique à la détection et à la transmission d'informations pertinentes. Dans le contexte de l'imagerie, cela correspond, idéalement, à la seule lecture des pixels pertinents et éventuellement au traitement de cette information. Cette efficacité peut être obtenue en optimisant la méthode de lecture et en gérant le flot de données en sortie du capteur d'une manière différente et clairement non usuelle.

C'est pourquoi des techniques de réduction des redondances d'informations en sortie du capteur ont été étudiées pour réduire le flot de données. Ces redondances peuvent être temporelles (même valeur pour un même pixel lors de deux trames consécutives) et spatiales (pixels ayant la même valeur lors de la lecture d'une trame). Si elles sont traitées, cela se traduit par des lectures de pixels et des traitements (notamment la conversion analogique-numérique) inutiles. Dans cet axe de recherche, il existe dans la littérature des travaux qui visent la réduction, voire la suppression des redondances dans le flot de données des imageurs afin de réduire leur consommation électrique. Néanmoins, cette dernière demeure une grandeur considérable par rapport aux autres composants électroniques consommant faiblement.

Dans le but de surmonter les limitations des capteurs conventionnels, une nouvelle classe de capteurs a été présentée dans la littérature : les capteurs d'images événementiels. Ces capteurs sont, à l'origine, des systèmes de vision bio-inspirés, conçus et développés afin d'imiter les systèmes biologiques. Ces imageurs sont souvent présentés dans la littérature sous le terme d'imageurs asynchrones car ils s'appuient sur le concept de la lecture des pixels lorsque ces derniers produisent un événement. Ces imageurs sont aussi appelés capteurs événementiels. En effet, un pixel n'est lu que lorsqu'il produit une information pertinente. Dans ce cas, il envoie une requête de lecture au système chargé de l'extraction de l'information du capteur. Cette propriété permet d'optimiser et globalement de réduire le flot de données en sortie d'un capteur. L'évolution et le développement de ces capteurs d'images les rend désormais suffisamment performants pour devenir des alternatives intéressantes aux capteurs d'images conventionnels à la vue de leur avantages : vitesse de lecture, dynamique de fonctionnement, flot de données, consommation électrique. Cependant, les imageurs événementiels présentés dans la littérature souffrent de limitations comme une lecture encore séquentielle de l'ensemble des pixels ou plus classiquement une taille du pixel importante. De plus, le fait d'avoir des pixels événementiels chargés de l'envoi des requêtes de lecture implique le besoin d'un circuit d'arbitrage afin de gérer les requêtes simultanées. Par conséquent, le protocole de lecture employé, dans ce type d'imageur, demeure non déterministe. De plus, les circuits d'arbitrage, dotés d'une circuiterie complexe qui augmente exponentiellement avec la résolution du capteur, insèrent des erreurs temporelles à l'information des pixels actifs.

Au début de cette thèse, on est reparti d'une feuille blanche en visant un capteur d'images événementiel et totalement asynchrone. En d'autres termes, notre capteur d'images implémente une lecture événementielle des pixels en utilisant de la logique asynchrone. Pour cela, la thématique cette thèse est située aux frontières de trois do-

maines : les capteurs d'images CMOS, la logique asynchrone et l'échantillonnage non uniforme.

Le capteur d'images, proposé dans ce manuscrit, emploie un échantillonnage non uniforme au sein du pixel afin que ce dernier soit capable de se comporter comme un détecteur d'événement. De plus, l'information du pixel est temporellement codée grâce à un codage bio-inspiré (*Time-to-First-Spike*). Par conséquent, une conversion tension-temps (*Time-to-Digital Conversion*) remplace la conversion analogique-numérique énergivore. En outre, notre système de lecture numérique, contrairement à ceux proposés dans la littérature, est capable de recevoir simultanément les requêtes de lecture provenant des pixels actifs sans le recours à un circuit d'arbitrage. De ce fait, le protocole de lecture proposé, dans le cadre de cette thèse, est parallèle (simultané) et déterministe. De plus, notre capteur d'images événementiel asynchrone réduit le flot de données en sortie en réduisant, voire supprimant, les redondances spatiales.

La conception de notre capteur d'images a débuté par l'élaboration de l'approche générale qui a été soumise en premier lieu à des simulations haut-niveau afin de garantir son bon fonctionnement. Ensuite, on est passé à la modélisation de notre système de lecture asynchrone ainsi qu'à la mise en place des simulations niveau *RTL* et niveau portes afin de tester et optimiser l'architecture numérique. Par la suite, on a effectué la synthèse à la main du système numérique asynchrone ainsi que le design analogique de notre pixel événementiel. Enfin, on note que la taille (en termes de nombres de transistors) de notre pixel événementiel est très raisonnable par rapport aux pixels événementiels existants dans la littérature avec 48 transistors dont 41 sont des transistors numériques à taille minimales.

De ce fait, le fonctionnement totalement événementiel de notre capteur d'images asynchrone induit une réduction drastique de l'activité du circuit imageur et par conséquent une réduction de la consommation électrique de l'ensemble du système de vision.

Ce manuscrit de thèse est composé de trois parties qui sont réparties de la façon suivante :

La première partie présente l'état de l'art. Le chapitre 2 présente les capteurs d'images CMOS. La chaîne de lecture des imageurs conventionnels y est présentée ainsi que leurs caractéristiques et leurs limitations. Les capteurs d'images événementiels existant dans la littérature sont ensuite décrits. Le chapitre 3 est consacré à la présentation de la logique asynchrone, en insistant sur les différences entre la logique asynchrone et la logique synchrone conventionnelle, les différents protocoles de communication ainsi que les diverses classes de circuits asynchrones. Les protocoles de communication asynchrones présentés dans ce chapitre seront employés au niveau du circuit de lecture et de traitement du capteur d'image. En raison de l'adoption d'un échantillonnage non uniforme, spécifique au capteur d'image proposé, on a consacré le chapitre 4 à l'échantillonnage non uniforme. Ce chapitre expose le théorème de *Shannon*, le principe de l'échantillonnage classique ainsi que l'échantillonnage non uniforme et ses avantages.

Plusieurs techniques de mise en œuvre ainsi que les effets de ce type d'échantillonnage sur le flot de données sont aussi présentés.

La deuxième partie introduit le fonctionnement général de notre capteur d'images événementiel asynchrone ainsi que les résultats de simulations haut niveau. Cette partie commence par le chapitre 5 qui positionne le capteur d'images événementiel asynchrone proposé dans cette thèse par rapport aux imageurs événementiels existant dans la littérature. Le chapitre 6 est, quant à lui, dédié à l'explication du fonctionnement général du système de lecture ainsi que du pixel événementiel. L'architecture globale du système de lecture asynchrone est également présentée dans ce chapitre. Le chapitre 7 présente les résultats de simulations de haut-niveau du protocole de lecture sous MATLAB. La stratégie de lecture et la réduction des redondances spatiales y sont évaluées au travers des simulations présentées.

La troisième partie comporte un seul chapitre, le chapitre 8, dans lequel on reprend l'architecture du système de lecture du capteur d'images. L'objectif de ce chapitre est d'expliquer la conception et la synthèse en logique asynchrone du système de lecture. Des simulations au niveau *netlist* ont été effectuées afin de valider le circuit du système de lecture.

Enfin, une conclusion est donnée à ce travail de thèse dans laquelle des perspectives sont dressées sur les potentiels des imageurs événementiels.

# **Première partie**

## **État de l'art**



# 2

## ÉTAT DE L'ART DES CAPTEURS D'IMAGES

### Sommaire

---

|         |  |           |
|---------|--|-----------|
| 2.1     | Les capteurs d'images électroniques . . . . .                                  | <b>8</b>  |
| 2.1.1   | Les capteurs d'images CCD . . . . .  | 8         |
| 2.1.2   | Les capteurs d'images CMOS . . . . .   | 11        |
| 2.2     | La photodiode . . . . .  | <b>11</b> |
| 2.3     | Les pixels des capteurs d'images CMOS . . . . .                                | <b>13</b> |
| 2.3.1   | Les pixels passifs (PPS) . . . . .   | 13        |
| 2.3.2   | Les pixels actifs (APS) . . . . .  | 14        |
| 2.3.2.1 | Pixel à intégration (Pixel à réponse linéaire) . .                             | 14        |
| 2.3.2.2 | Pixel logarithmique . . . . .  | 16        |
| 2.3.2.3 | Choix . . . . .  | 16        |
| 2.4     | Chaîne de lecture du capteur d'images CMOS . . . . .                           | <b>18</b> |
| 2.4.1   | Les décodeurs d'adresses . . . . .   | 18        |
| 2.4.2   | Amplificateur Colonne . . . . .  | 18        |
| 2.4.3   | Conversion analogique numérique . . . . .                                      | 18        |
| 2.5     | Caractéristiques générales du capteur d'images CMOS . . . . .                  | <b>20</b> |
| 2.5.1   | Le facteur de remplissage . . . . .  | 20        |
| 2.5.2   | La sensibilité du pixel . . . . .  | 20        |
| 2.5.3   | La dynamique de fonctionnement . . . . .                                       | 21        |
| 2.5.4   | La résolution . . . . .  | 21        |
| 2.6     | Limitations des imageurs conventionnels . . . . .                              | <b>22</b> |
| 2.6.1   | Flots de données en sortie . . . . .   | 22        |
| 2.6.2   | Conversion analogique numérique . . . . .                                      | 23        |
| 2.6.3   | Dynamique de fonctionnement . . . . .  | 23        |
| 2.7     | Les capteurs d'images asynchrones . . . . .                                    | <b>24</b> |
| 2.7.1   | Les systèmes neuromorphiques . . . . .   | 24        |
| 2.7.2   | Principe des imageurs asynchrones . . . . .                                    | 25        |
| 2.7.3   | Protocole de lecture asynchrone : Address Event Representation (AER) . . . . . | 26        |
| 2.7.4   | Codage de l'information lumineuse du pixel . . . . .                           | 29        |
| 2.7.4.1 | Codage Temporel Bio-inspiré : Time-to-First-Spike (TFS) . . . . .              | 29        |

|         |   |    |
|---------|---|----|
| 2.7.4.2 | Conversion tension-temps (Time-to-Digital Conversion : TDC) . . . . . | 30 |
| 2.8     | Conclusion . . . . .  | 33 |

Les capteurs d'images sont des composants électroniques très exploités dans le domaine de l'industrie des semi-conducteurs. De plus, les capteurs d'images forment un axe de recherche toujours en cours de développement et d'évolution. Ceci est essentiellement dû à l'augmentation continue de l'intégration des capteurs d'images par des applications diverses. Selon le graphe de la figure 2.1 fourni par la société "ICInsights", on observe la croissance considérable du marché des caméras intégrées dans des appareils électroniques nomades, comme les téléphones mobiles portables (*Smartphones*) et les tablettes. De plus, le marché des caméras dédiées à des applications médicales, de surveillance et d'automobile se démarque par une croissance continue.

Historiquement, le domaine de l'acquisition d'images a commencé avec les systèmes traditionnels de photographie argentique. En fait, pour ce type de caméras, le film photographique effectue simultanément l'acquisition et le stockage de l'image capturée.

Cependant, à partir des années 70, les capteurs d'images électroniques ont progressivement dominé le marché des caméras. Le principe de ces capteurs d'images consiste d'une part à convertir l'information lumineuse en un signal électrique et d'autre part à sauvegarder le signal électrique dans des dispositifs de mémoires analogiques (enregistrement électromagnétique) ou numériques (disque dur externe, carte mémoire).

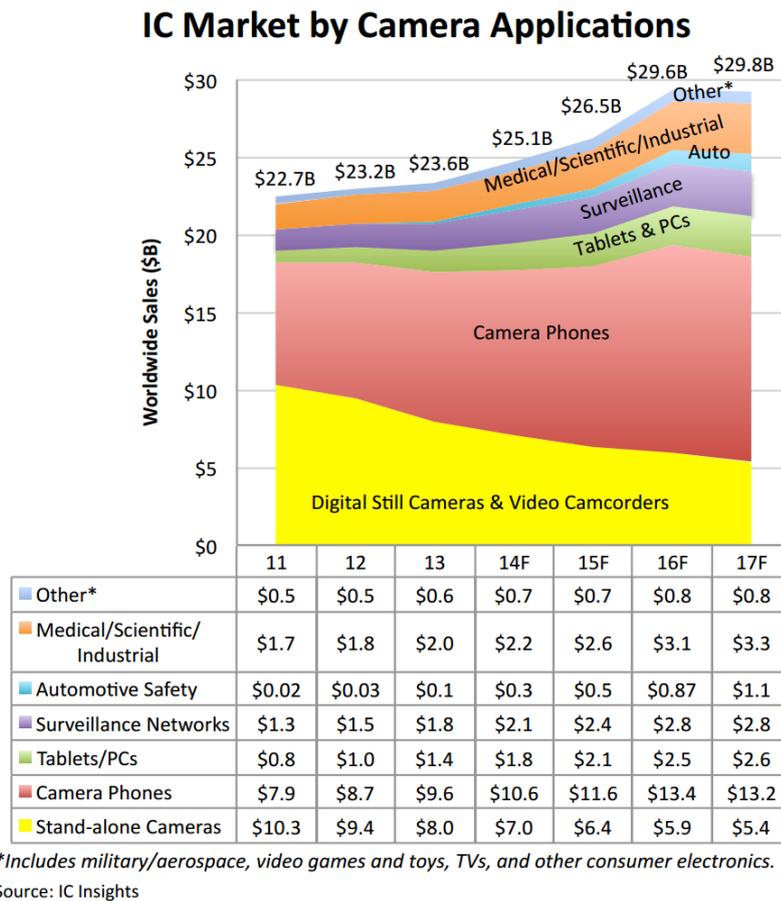
Dans le cadre de cette thèse, on s'intéresse aux capteurs d'images électroniques. Ce premier chapitre est ainsi dédié à l'explication de ces deux types de capteurs d'images électronique : les capteurs CCD (*Charge-Coupled Device*), et les capteurs CMOS (*Complementary Metal-Oxide-Semiconductor*). Ainsi, l'élément de base (le pixel), le système de lecture et de conversion des capteurs d'images électroniques seront détaillés. On abordera ensuite les limitations des capteurs d'images conventionnels. Dans cette thèse, nous allons montrer comment une approche événementielle (asynchrone) peut permettre de surmonter certaines de ces limitations. Ainsi, ce chapitre se termine par l'étude de l'état de l'art des capteurs d'images asynchrones présents dans la littérature.

## 2.1 Les capteurs d'images électroniques

### 2.1.1 Les capteurs d'images CCD

Le premier capteur d'images électronique inventé est le capteur d'images de type "Dispositif à Transfert de Charge : DTC" (*Charge-Coupled Device : CCD*). Ce capteur inventé par George E. Smith et Willard Boyle dans les Laboratoires Bell et datant de l'année 1969, rapportera le prix de Nobel de physique en 2009<sup>1</sup> à ces inventeurs. En

1. [http://www.nobelprize.org/nobel\\_prizes/physics/laureates/2009/](http://www.nobelprize.org/nobel_prizes/physics/laureates/2009/)



**FIGURE 2.1** – Répartition du marché des capteurs d'images par applications

fait, l'invention des capteurs d'images CCD a révolutionné le domaine de l'imagerie. Ces capteurs d'images ont été, en premier lieu, utilisés pour des applications scientifiques. Ensuite, ils se sont intégrés progressivement aux marchés grand public avec les premiers appareils photo électroniques.

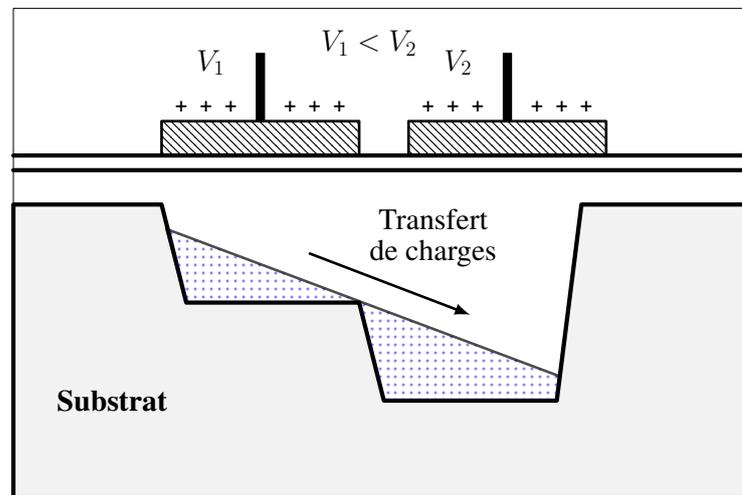
Le principe du capteur d'images CCD est basé sur le concept de transfert de la charge photogénérée d'un pixel vers la sortie de la matrice en passant par les autres pixels de la ligne ou de la colonne. En fait, la cellule de base "photogate" est à la base une capacité MOS (*Metal Oxide Semi-conductor*) dont la grille est polarisée. La tension de polarisation force un dépeuplement de trous dans la zone active sous la grille. Grâce à l'effet photoélectrique, la lumière incidente sur l'électrode de la capacité MOS génère une paire électron-trou dans la zone dépeuplée. Cependant, sous l'effet du champ électrique polarisant, la paire électron-trou est séparée. Par conséquent, l'électron est attiré vers la grille de la capacité tandis que le trou est repoussé dans le volume du silicium. Le pixel du capteur d'images CCD étant composé d'une ou de plusieurs grilles, le transfert des charges accumulées se fait en les transitant d'une capacité à l'autre, créant par la suite



**FIGURE 2.2** – Essai du premier capteur d’images électronique CCD par George E. Smith et Willard Boyle au laboratoire Bell

des puits de potentiel successifs. Le principe de transfert de charge est illustré dans la figure 2.3. Enfin, les charges d’un pixel arrivant en bord de matrice sont converties en une tension selon un facteur de conversion constant par un détecteur de charge. Dans le cas d’un seul pixel, la lecture des charges accumulées est simple. Cependant, la lecture de l’ensemble des pixels d’un capteur d’images CCD n’est pas triviale. En fait, la technique ainsi que la vitesse de lecture limitent les performances du capteur d’images CCD.

Étant les premiers capteurs d’images électronique, les capteurs CCD ont servi de ré-



**FIGURE 2.3** – Principe de transfert de charges dans un capteur d’images CCD

férence pour tous les autres capteurs. Les capteurs CCD présentent un faible niveau de bruit, une grande dynamique de fonctionnement, un grand facteur de remplissage ainsi qu’un bon rendement quantique [Fos93]. Néanmoins, les capteurs d’images CCD sont

limités par leur coût élevé de fabrication en technologie CCD dédié, leur incompatibilité avec l'intégration des circuits CMOS, leur forte consommation ainsi que leur vitesse de lecture limitée [FH14 ; Fos93].

### 2.1.2 Les capteurs d'images CMOS

L'évolution de la technologie de fabrication des transistors CMOS pendant les années 1980 a entraîné le développement d'un nouveau capteur d'images dits capteur d'images CMOS, conçus et fabriqués en technologie CMOS. Ce type de capteurs bénéficient des avantages du développement continu de la technologie CMOS. En outre, le procédé de fabrication des circuits CMOS, permet l'intégration, sur la même puce, de la matrice photosensible, des circuits de lecture ainsi que des éventuels circuits de traitement d'images. En d'autres termes, les capteurs de vision CMOS sont des sortes de systèmes sur puce (*System On Chips : SoC*), essentiellement composés de la matrice de pixels et des circuits de lecture et de traitement (décodeurs, amplificateurs colonne, convertisseur analogique numérique et multiplexeur).

Grâce à leur faible coût de fabrication, les capteurs d'images CMOS ont progressivement remplacé les capteurs CCD surtout pour les appareils photo grand public. De plus, les capteurs d'images CMOS ont une vitesse de lecture plus élevée, une possibilité d'accès aléatoire aux pixels ainsi qu'une consommation électrique plus faible.

Dans le cadre de cette thèse, on s'intéresse aux capteurs d'images CMOS. En effet, grâce à leur chaîne de lecture de la donnée ouverte, la potentialité d'intégrer du traitement d'images ainsi que leur faible consommation électrique, la technologie CMOS est mieux adaptée pour faire évoluer les architectures des capteurs d'images.

Dans ce qui suit, on détaille les différentes structures du capteurs d'images CMOS, débutant par l'élément photosensible (la photodiode), en passant par les divers types de pixels ainsi que la chaîne de lecture complète.

## 2.2 La photodiode

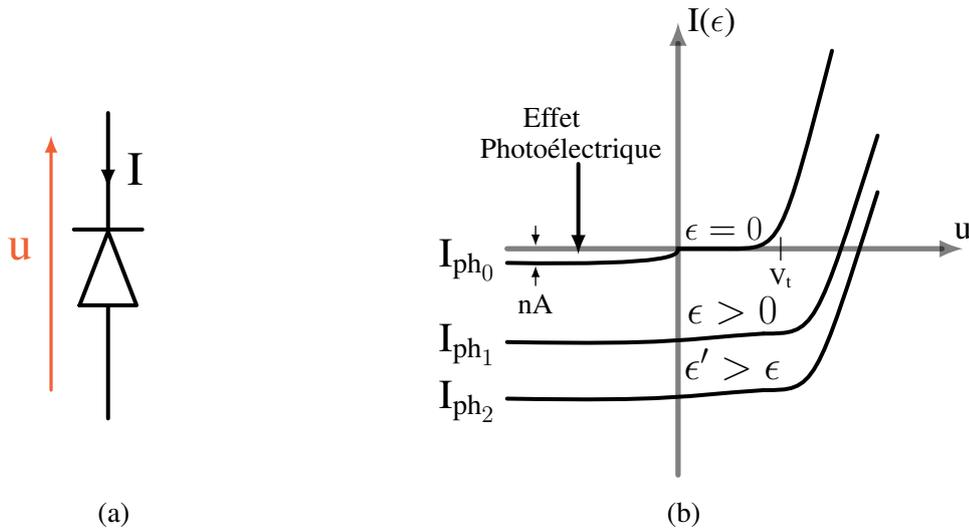
Une photodiode est à la base une diode polarisée en inverse et capable de générer un courant électrique photogénéré. En fait, le photon pénétrant dans le silicium crée une paire électron-trou qui est collectée par la photodiode avec un certain rendement connu comme étant le rendement quantique de la photodiode. Alors, la photodiode bénéficie de l'effet des photons incidents à la surface du matériau afin de générer un courant proportionnel à l'intensité de la lumière incidente. Soumise une lumière incidente, la photodiode génère un courant  $I_{ph}$ . Suivant l'équation (2.1) du courant photogénéré,  $I_{ph}$  dépend de la longueur d'onde ( $\lambda$ ), de la puissance ( $P_{ph}$ ) de la lumière incidente et de l'efficacité quantique du semi-conducteur ( $\eta$ ) (soit la capacité du semi-conducteur à absorber ces photons). On note que l'efficacité quantique de la photodiode est le rapport

entre le nombre des photons incidents à la surface de la photodiode et le nombre de charges électriques photogénérés.

$$I_{ph}(\epsilon) = \frac{|q|\eta\lambda P_{ph}}{hc} \tag{2.1}$$

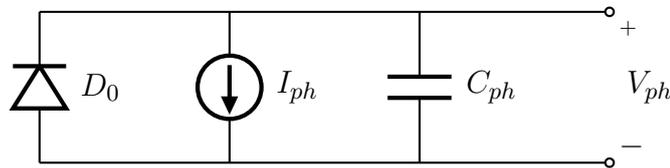
$$\eta = \frac{\text{Nombre d'électrons photogénérés}}{\text{Nombre de photons incidents}} \tag{2.2}$$

Selon la figure 2.4, on peut observer l'augmentation en valeur absolue du courant total de la photodiode en fonction du nombre des photons incidents ( $\epsilon$ ).



**FIGURE 2.4** – Exemple d’une photodiode polarisée en inverse (a) Le courant photogénéré en fonction de la tension de polarisation  $u$  et le nombre de photons incidents  $\epsilon$  (b)

Le circuit de la figure 2.5 est un circuit équivalent à la photodiode polarisée par une tension  $V_{ph}$ . Le courant photogénéré est symbolisé par une source de courant tandis que l’effet capacitif de la photodiode est exprimé par une capacité parasite  $C_{ph}$ .



**FIGURE 2.5** – Le circuit électrique équivalent de la photodiode en mode photoconducuteur

On peut alors exprimer le courant photogénéré par la photodiode par l’équation 2.3.

$$I_{ph} = C_{ph} \cdot \int \frac{dV_{ph}}{dt} \tag{2.3}$$

## 2.3 Les pixels des capteurs d'images CMOS

Due à la progression continue des technologies de fabrication des semi conducteurs, les capteurs d'images CMOS ont pu progressivement remplacer leurs prédécesseurs : les capteurs CCD. En fait, le pixel est l'élément de base d'un capteur d'images. Il est composé d'un élément photosensible (la photodiode) et d'un ou de plusieurs transistors MOS. Chaque pixel contient une information relative à la lumière incidente représentée par le courant photogénéré par la photodiode [The07]. Dans le but d'extraire l'information lumineuse de chaque pixel, on couple un système de lecture à la matrice de pixels afin de lire la tension  $V_{ph}$ , proportionnelle au nombre de photons collectés (Figure 2.11).

En fait, selon le type de fonctionnement et la structure du pixel, on peut différencier deux catégories de capteurs d'images CMOS : les capteurs d'images passifs (*Passive Pixel Sensor : PPS*) et les capteurs d'images actifs (*Active Pixel Sensor : APS*). Cette section est dédiée à ces différentes structures de pixels d'un capteur d'images CMOS.

### 2.3.1 Les pixels passifs (PPS)

Le capteur d'images passif (*Passive Pixel Sensor : PPS*) est basé sur des pixels à structure simple. Les pixels dits passifs sont, selon la figure 2.6, basés sur l'utilisation d'une photodiode et d'un seul transistor [YPGSD91].

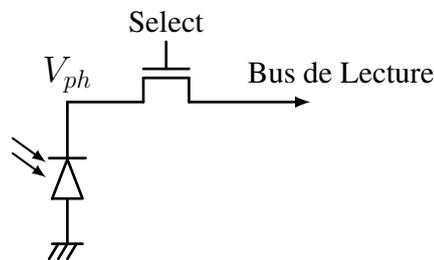


FIGURE 2.6 – Pixel Passif (*Passive Pixel Sensor : PPS*)

Le système de lecture du capteur d'images extrait séquentiellement l'information des pixels. Et, les pixels sont lus ligne par ligne. Durant chaque lecture, le transistor contrôlé par le signal *Select* devient passant dans le but de transférer le signal analogique aux bornes de la photodiode ( $V_{ph}$ ) vers le bus de lecture commun par les pixels de la même colonne.

Afin de permettre la lecture de chaque pixel, un amplificateur est ajouté au pied de chaque colonne. La simplicité de la structure à un seul transistor du pixel passif permet d'avoir des pixels à surface réduite ainsi qu'un facteur de remplissage très élevé, ainsi qu'un meilleur coefficient d'efficacité quantique. Cependant, la transmission directe du signal aux bornes de la photodiode sur des bus capacitifs augmente la charge capacitive

du nœud  $V_{ph}$  et par la suite la constante du temps  $RC$ . on comprend ainsi la vitesse de lecture limitée des capteurs passifs. De plus, la charge capacitive diminue la qualité du signal analogique transmis en augmentant le signal de bruit. Par conséquent, les capteurs passifs ont un faible rapport signal sur bruit.

Les inconvénients de ce type de capteurs d'images et les progrès technologiques ont permis l'avènement de pixels beaucoup plus performants et versatiles : les pixels actifs (*APS*) et le besoin d'intégrer des fonctions et des pré-traitements au niveau du pixel ont mis en avant les pixels actifs (*APS*).

### 2.3.2 Les pixels actifs (*APS*)

Le pixel actif est le composant élémentaire du capteur d'images actif (*Active Pixel Sensor : APS*). Dépendant de son architecture, le pixel peut générer une réponse linéaire ou non-linéaire [Fos97b]. Le pixel actif qui génère une réponse linéaire est le pixel à intégration (section 2.3.2.1), aussi appelé pixel standard car il représente la très grande majorité des pixels utilisés actuellement. Le pixel actif logarithmique fournit une réponse non-linéaire de type logarithmique (section 2.3.2.2). Les différentes architectures des pixels actifs sont détaillées dans cette section.

#### 2.3.2.1 Pixel à intégration (Pixel à réponse linéaire)

La plus simple architecture du pixel à intégration est le pixel à trois transistors (3T) et une photodiode. Selon la figure 2.7, cette structure est caractérisée par l'intégration du transconducteur d'un amplificateur suiveur au sein du pixel.

En effet, le transistor  $T_2$  permet de décorréler la photodiode du bus de lecture et résout ainsi les problèmes liés à la charge capacitive et du bruit associé du pixel passif. Il est important de noter  $T_2$  forme un amplificateur suiveur avec un transistor de charge (active) placé en pied de colonne.

Dans le but de contrôler les phases de fonctionnement du pixel, il est nécessaire d'intégrer d'autres transistors. En effet, les phases de fonctionnement élémentaires du pixel actifs sont : la phase de *Reset*, celle de l'intégration, et finalement celle de la lecture. Les transistors  $T_1$  et  $T_3$  commandent respectivement la phase de *Reset* et la phase de sélection des lignes de pixels pour la lecture.

La figure 2.8 illustre les phases de fonctionnement d'un pixel durant deux trames consécutives. Le pixel est d'abord initialisé suite à la fermeture du MOS  $T_1$ . Durant cette phase, la tension aux bornes de la photodiode est maintenue à une tension proche de  $V_{dd}$ . A la fin de la phase de *Reset*, le transistor  $T_1$  est ouvert. Cet instant est le début de la phase d'intégration. Pendant cette phase, la capacité parasite du nœud  $ph$  ( $C_{ph}$ ) commence à se décharger en fonction du courant photogénéré  $I_{ph}$ . Par conséquent, la tension aux bornes de la photodiode décroît suivant l'équation (2.4). La figure 2.8 montre aussi l'effet de la valeur de la luminosité et par la suite celle du courant photogénéré durant la phase d'intégration. En effet, la valeur du courant photogénéré ( $I_{ph_1} > I_{ph_2}$ ) affecte la

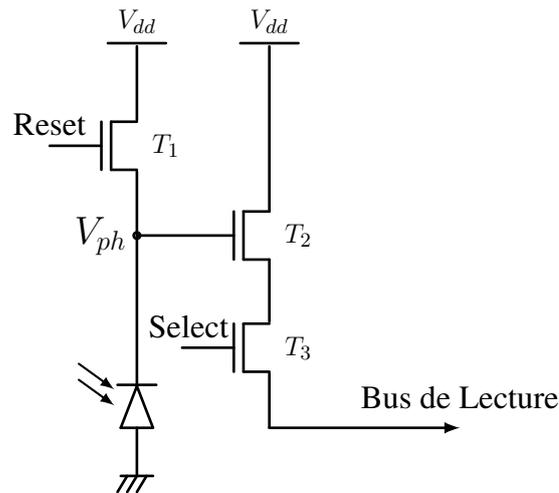


FIGURE 2.7 – Pixel Actif (*Actif Pixel Sensor : APS*)

pente de décharge de la photodiode et par la suite la tension aux bornes de la photodiode à la fin de la phase d'intégration.

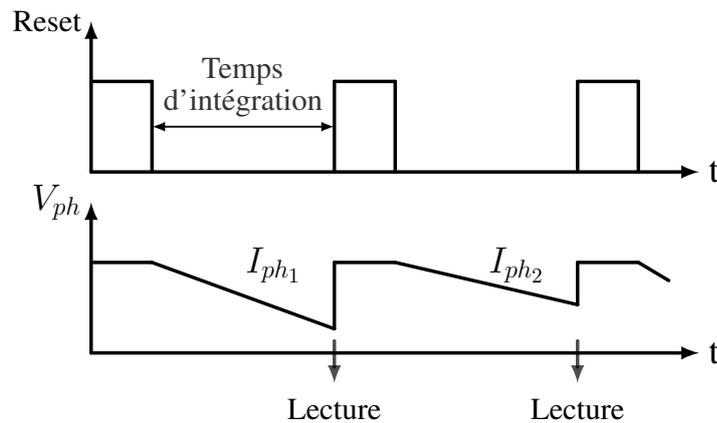


FIGURE 2.8 – Diagramme de fonctionnement d'un pixel actif standard (*APS*)

La durée de la phase d'intégration est pré-définie. Les pixels d'une même matrice suivent un temps d'intégration unique connu comme étant le temps d'intégration (*Integration Time : IT*). L'impact de la valeur du temps d'intégration est analysé dans la section suivante (section 2.4). A la fin du temps d'intégration, la tension aux bornes de la photodiode est extraite par le système de lecture.

La phase de lecture est commandée par le transistor  $T_3$ . Ce transistor, une fois fermé, connecte le transistor  $T_2$  à sa charge et ainsi fait office d'amplificateur suiveur. La tension aux bornes de la photodiode, à l'instant  $T_{int}$ , est ainsi transmise à la sortie de ce suiveur, en pied de colonne.

Grâce à la réponse linéaire en sortie du pixel, on peut déduire l'information lumineuse du pixel. Suivant l'équation (2.1),

$$\frac{dV}{dt} = \frac{I_{ph}}{C_{ph}} \quad (2.4)$$

En concevant le pixel actif avec une technologie de fabrication classique, le facteur de remplissage est limité à environ 80% de la surface total du pixel. Un facteur de remplissage réduit, diminue le courant photogénéré dans un pixel de surface donnée. Cependant la réduction de la charge capacitive grâce au transistor suiveur, diminue le bruit dans le pixel, ce qui augmente par la suite la dynamique de fonctionnement ainsi que le rapport signal sur bruit [YPEC04]. De plus, l'ajout de lentilles focalisantes au dessus de chaque pixel permet de limiter l'impact de ce facteur de remplissage limité.

### 2.3.2.2 Pixel logarithmique

Le pixel à compression logarithmique se distinguent des pixels à intégration par leur mode de fonctionnement et leur réponse en sortie [Lab07 ; Sic99 ; Fos97a]. En premier lieu, les pixels logarithmiques suivent un fonctionnement continu. En d'autres termes, ils ne subissent pas de phase d'initialisation (*Reset*) ou d'intégration. En deuxième lieu, l'information lumineuse du pixel est traduite suivant un réponse logarithmique en sortie du pixel (Figure 2.10). La réponse logarithmique du pixel est décrite par l'équation (2.5).

$$V_{ph} = V_{dd} - K \ln \frac{I_{ph}}{I_0} \quad (2.5)$$

L'architecture du pixel logarithmique comporte essentiellement trois transistors (Figure 2.9). Le transistor  $T_1$ , monté en diode et utilisé en régime de faible inversion, assure la compression logarithmique au sein du pixel. Tandis que le transistor  $T_3$ , agissant comme un interrupteur, est utilisé lors de la sélection du pixel pendant la lecture de la matrice. En raison de la réponse logarithmique, cette structure de pixel permet de convertir plusieurs ordres de grandeur de courant photogénéré en un seul ordre de grandeur de tension [Lab07]. Par conséquent, le pixel logarithmique dispose d'une grande dynamique de fonctionnement qui peut atteindre plus de 100 dB tout en gardant une architecture simple à trois transistors. Cependant, la structure pixel logarithmique manifeste une sensibilité réduite pour les scènes à basse luminosité ainsi qu'une réponse plus sensible à la température que celle du pixel à réponse logarithmique [Zim11]. Cette structure présente également un bruit spatial fixe (FPN) plus important que la structure standard [CC07]. Enfin, sa réponse non linéaire fait que cette structure est très peu utilisé.

### 2.3.2.3 Choix

Dans le cadre de cette thèse, on a adopté l'architecture du pixel à intégration, le fonctionnement et la réponse linéaire du pixel à intégration sont les plus adaptés au fonctionnement événementiel désiré. De plus, le principe et le fonctionnement du capteur

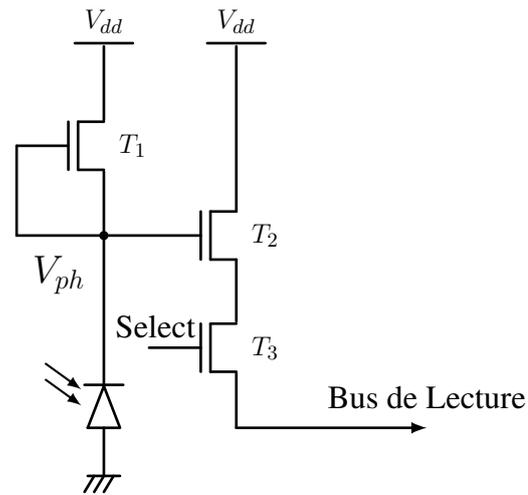


FIGURE 2.9 – Pixel logarithmique basique

d'images asynchrone événementiel compense le défaut en dynamique de fonctionnement du pixel [PMW11] comme on le verra par la suite.

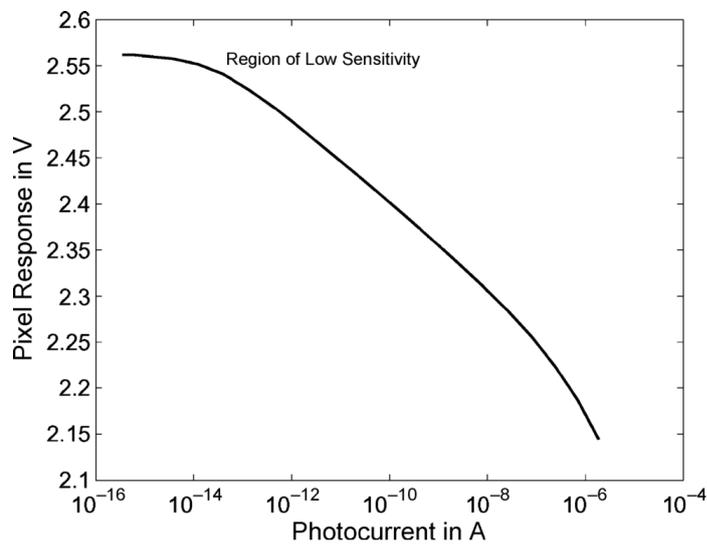


FIGURE 2.10 – Réponse typique d'un pixel logarithmique[CC07]

## 2.4 Chaîne de lecture du capteur d'images CMOS

Le circuit de lecture d'un capteur d'images CMOS est dédié à l'extraction et la numérisation de l'information lumineuse des pixels. Pour un imageur conventionnel, les pixels sont lus séquentiellement, cadencée par un signal d'horloge qui contrôle les signaux de commande du système entier. L'architecture du système de lecture d'un imageur CMOS est typiquement composé d'une matrice de pixels, des décodeurs d'adresses numériques, d'amplificateurs colonne (*Column Amplifier* : CA) et d'un ou de plusieurs convertisseurs analogique-numériques (*Analog-to-Digital Converter* : ADC) [YPEC04 ; The08]. La figure 2.11 illustre l'architecture typique d'un capteur d'images CMOS conventionnel.

### 2.4.1 Les décodeurs d'adresses

Les décodeurs d'adresses ligne et colonne donnent accès aux pixels aux bus de sortie. Le décodeur ligne est dédié à la sélection de la ligne de pixels lue et son accès aux bus de sortie colonne. Tandis que le décodeur colonne contrôle l'accès sur le bus connectant les pixels au convertisseur analogique numérique.

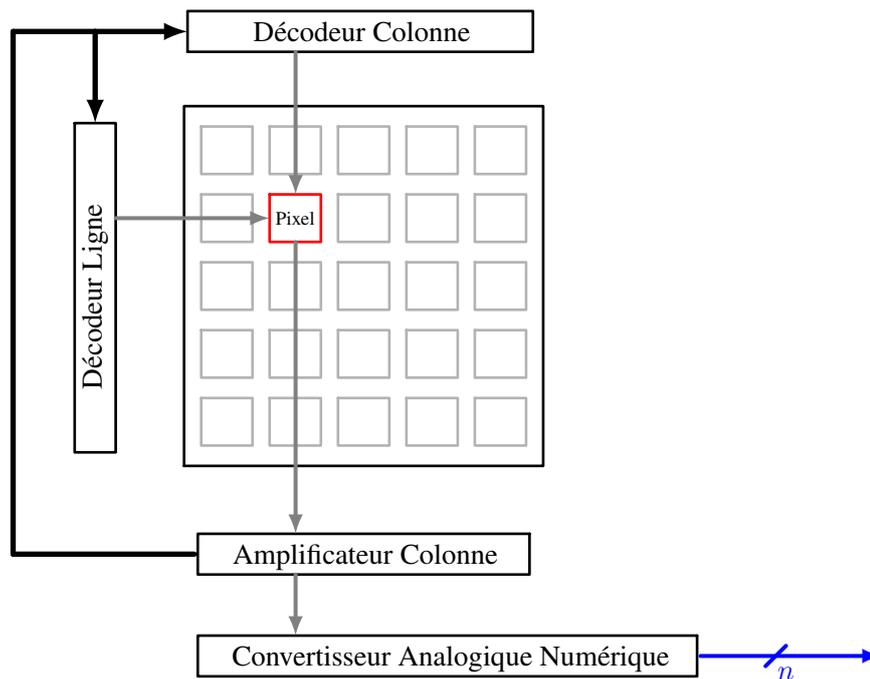
Étant des blocs numériques, les décodeurs d'adresses sont conçus à avec un langage de description matérielle de haut niveau (*VHDL*, *Verilog*), synthétisés par les outils de conception assistée par ordinateur ((*CAO*) *Computer-aided design* : CAD). Ces blocs sont intégrés dans la puce du capteur d'images.

### 2.4.2 Amplificateur Colonne

En fait, l'amplificateur colonne est composé basiquement de circuits échantillonneur-bloqueur (*Sample and Hold S/H*) et d'amplificateurs distribués par colonne. Le premier étage suiveur de ce bloc est en partie dans le pixel et les étages suivants permettent de gérer le niveau de la tension obtenue (amplification) et la compensation du bruit spatial fixe. A la sortie de l'amplificateur colonne, le signal analogique du pixel est discrétisé grâce à l'étage de conversion analogique-numérique.

### 2.4.3 Conversion analogique numérique

La conversion analogique-numérique : CAN (*Analog-to-Digital Conversion* : ADC) de l'information analogique des pixels transfère l'information lumineuse dans le domaine numérique. L'évolution des convertisseurs analogique numérique dédiés aux capteurs d'images ainsi que leurs architectures constituent un axe de recherche important dans le domaine de l'imagerie [LBFBRV14 ; Xue+14 ; OG14 ; Han+13 ; SI10 ;



**FIGURE 2.11** – Architecture d'un capteur d'images CMOS conventionnel

[Sno+07b](#)]. Le convertisseur analogique numérique peut être implémenté au niveau du pixel, au pied de chaque colonne et au niveau de la matrice des pixels.

Intégrer un seul convertisseur pour l'ensemble des pixels implique le besoin d'une vitesse de conversion identique à celle de lecture des pixels. Cette méthode d'implémentation nécessite un effort de conception afin d'avoir la vitesse de lecture désirée tout en conservant une consommation électrique raisonnable.

Dans le but de relâcher les contraintes de conception pour des matrices de grande dimension en réduisant la vitesse de conversion ainsi que la consommation électrique du système, l'intégration des convertisseurs au pied de chaque colonne a été adaptée [[OE13](#) ; [SKK12](#) ; [NAA09](#) ; [Sno+07a](#) ; [Sno07](#) ; [ST06](#)]. Cependant, la réalisation des convertisseurs par colonne peut s'avérer coûteuse en effort de conception et en surface. Cette implémentation est illustrée par l'exemple de la figure 2.11.

La conversion analogique numérique peut également être intégrée au sein de chaque pixel [[PT07](#) ; [HHK03](#) ; [YFE99](#)]. L'avantage de cette implémentation réside dans la sortie numérique robuste du pixel. Néanmoins, ce type de convertisseurs augmente fortement la consommation électrique, le nombre de transistors et la surface du pixel et diminue ainsi son facteur de remplissage. Cependant, cette approche peut s'avérer intéressante en adoptant la conception en technologie 3D, vu que dans ce cas le convertisseur par pixel n'est pas à l'intérieur mais dans une couche au dessus ou au dessous du pixel.

Actuellement, la très grande majorité des imageurs vendus dans le monde ont un convertisseur analogique numérique par colonne.

## 2.5 Caractéristiques générales du capteur d'images CMOS

### 2.5.1 Le facteur de remplissage

Le facteur de remplissage (*Fill Factor*) est défini comme étant le rapport entre la surface occupée par l'élément photosensible (la photodiode) et la surface totale du pixel. Même si le facteur de remplissage d'un pixel actif est toujours inférieur à 100%, on vise à avoir le plus grand facteur de remplissage. En fait, un grand facteur de remplissage augmente l'efficacité quantique (*Quantum Efficiency : QE*) définie comme étant le rapport entre le nombre d'électrons photogénérés et le nombre de photons attaquant la surface de la photodiode. En revanche, la diminution de la zone photosensible, et par la suite du facteur de remplissage, diminue la sensibilité du pixel, le signal électrique photogénéré et le rapport signal sur bruit (*Signal-to-Noise Ratio*) [Amh12 ; Lab07].

Afin d'améliorer le facteur de remplissage, on peut, d'une part, optimiser les tailles des transistors dans le pixel. D'autre part, on peut concevoir le capteur d'images avec un processus de fabrication récent. Néanmoins, ces solutions impliquent un bruit plus élevé dans le pixel et une dynamique de fonctionnement réduite due à la baisse de la tension d'alimentation. En outre, une solution adoptée par les industriels consiste à déposer des lentilles convergentes sur les pixels afin de concentrer les photons vers la photodiode et ainsi augmenter de manière optique le facteur de remplissage.

### 2.5.2 La sensibilité du pixel

La sensibilité du capteur d'images est principalement liée à celle de l'élément photosensible : la photodiode. En fait, La sensibilité correspond au nombre de photons incidents nécessaires pour passer d'un niveau de gris au suivant. En d'autre terme, la sensibilité de la photodiode est caractérisée par sa réponse aux photons incidents. Cette réponse est liée à plusieurs paramètres. On cite d'abord l'effet de la surface de la photodiode sur la sensibilité. En effet, en augmentant la surface photosensible, on augmente le nombre de photons reçus par unité de temps et améliore par la suite la sensibilité de la photodiode. De plus, l'efficacité quantique (*Quantum Efficiency : QE*) augmente relativement par rapport à la surface de la photodiode. Et selon l'équation (2.1), le courant photogénéré  $I_{ph}$  dépend essentiellement de l'efficacité quantique. En outre, la sensibilité est influencée par le rapport signal sur bruit (*SNR*) du pixel. Par exemple, l'augmentation du nombre de transistors dans le pixel, augmente le bruit et diminue par la suite le *SNR*.

Un autre critère qui influe la sensibilité du pixel, c'est le courant d'obscurité  $I_0$  (*Dark Current*). Ce dernier, étant le courant de diffusion généré par une diode polarisée en inverse (Reverse Bias), s'ajoute au courant photogénéré ( $I_{ph}$ ). L'effet du courant d'obscurité est néfaste sur la mesure d'un faible courant photogénéré et par la suite sur la

sensibilité du pixel. En d'autres termes, le courant d'obscurité affecte la détection des faibles luminosités. De plus, ce courant, étant fortement dépendant de la température, fluctue avec cette dernière et double de valeur chaque 6 à 8 °C [Bae06]. Alors, le courant d'obscurité est non seulement nocif à la sensibilité du pixel mais aussi difficilement contrôlable.

### 2.5.3 La dynamique de fonctionnement

En général, la dynamique de fonctionnement est la différence entre la plus petite et la plus grande valeur détectable d'un signal lors de sa transmission, son traitement et son stockage. Pour un capteur d'image, la dynamique de fonctionnement (*Dynamic Range* :  $DR$ ) est principalement limitée par le pixel. En fait, la dynamique de fonctionnement du pixel dépend de sa capacité à différencier les diverses luminosités d'une image. Cette dynamique de fonctionnement illustre la capacité d'un capteur à transmettre une image comportant à la fois des parties à forte et à basse luminosité sans perte de détails. Selon l'équation (2.6), cette grandeur, exprimée en *décibel*, est le rapport entre la plus grande valeur du signal d'entrée non saturé ( $E_{max}$ ) et la plus petite valeur du signal d'entrée détectable ( $E_{min}$ ).

$$DR_{dB} = 20 \log \left( \frac{E_{max}}{E_{min}} \right) \quad (2.6)$$

L'œil humain a une plage de dynamique de plus que 140dB, alors que les capteurs d'images conventionnels offrent une dynamique de fonctionnement limitée actuellement à 70 dB – 80 dB. En fait, cette limitation en dynamique est principalement liée à la technique de lecture séquentielle classique qui impose le temps d'intégration dans le pixel et au *SNR* de l'électronique. Pour cela, l'amélioration de la dynamique de fonctionnement des capteurs d'images est toujours un sujet de recherche à la mode [Abb14; GMC14; JJw14; Abb+13; Liu12; KGG06; XE03]. Dans cette thématique, les techniques de lecture non conventionnelles des capteurs d'images événementiels, souvent dits asynchrones, offrent une augmentation de la dynamique de fonctionnement de l'imageur [PMW11; PMW10; Gui+07; CEC04]. Ce type de capteur est étudié en détails dans la section 2.7.

### 2.5.4 La résolution

La résolution d'un capteur d'images illustre sa capacité à mesurer et distinguer les détails d'une scène. Le terme de résolution est généralement utilisé pour représenter le nombre de pixels, vu que la quantité d'information dans l'image est corrélée au nombre de pixels total. Par exemple, pour un capteur d'images à  $M$  lignes et  $N$  colonnes de pixels, on obtient une résolution spatiale du capteur d'images de  $M*N$  pixels. De plus, le capteur d'images CMOS est également caractérisé par le temps nécessaire

pour accomplir l'acquisition d'une trame. Cette durée est connue comme étant la résolution temporelle du capteur d'images. Cette résolution détermine le nombre d'images, ou de trames acquises par seconde (*Frame Per Second : FPS*). Autrement dit, la résolution temporelle impose le flux sortant ou *Frame Rate* du capteur d'images. Par conséquent, la résolution temporelle des capteurs à flux vidéos doit respecter un *frame rate* qui varie classiquement entre 24 et 60 trames (*frames*) par seconde. Tandis que pour les caméras ultra rapides, le *frame rate* peut atteindre plus de 1,000,000 trames (*frames*) par seconde [Pha].

## 2.6 Limitations des imageurs conventionnels

### 2.6.1 Flots de données en sortie

Les capteurs de vision CMOS sont composés principalement d'un vecteur de pixels à deux dimensions. Cependant, l'extraction simultanée des données de l'ensemble des pixels s'avère impossible avec la technologie de fabrication conventionnelle. Par conséquent, le mode de lecture adopté par les concepteurs des imageurs CMOS, est un mode de lecture séquentiel. Cette méthode robuste et simple à implémenter, consiste à extraire l'information lumineuse de tous les pixels durant chaque trame selon un procédé parfaitement déterminé et régulier.

Cette méthode de lecture exhaustive, implique un flot de données régulier en sortie du capteur d'images, ce qui entraîne un gaspillage d'énergie. Cette consommation électrique est essentiellement liée au bloc de conversion analogique numérique qui fonctionne à la vitesse de lecture du capteur. De ce fait, la consommation électrique d'un capteur d'images augmente proportionnellement à son *frame rate* et à sa résolution spatiale. Cette problématique persiste notamment dans les capteurs d'images récents qui comptent des millions de pixels et qui fonctionnent à une vitesse de lecture élevée. Ainsi, il faut satisfaire un compromis, lors de la conception des imageurs conventionnels, entre d'une part l'énergie électrique consommée par le système de vision et d'autre part la vitesse de lecture et la résolution spatiale du capteur.

Dans le cadre des imageurs conventionnels, des études sur la compression du flots de données ont été faites au niveau du capteur d'images et de son circuit de lecture [HPDA12 ; Amh12 ; Amh+11 ; CFC08 ; Chi+07 ; EYF99 ; JG93] ainsi qu'au niveau algorithmique (par exemple : JPEG) grâce à des blocs de traitement d'images dédiées [CSS13 ; HPDA12]. Cependant, ces blocs impliquent une consommation électrique supplémentaire à celle du capteur d'images et de son système numérique de lecture. Une solution alternative consiste en une lecture événementielle des pixels qui réduit le flot de données en sortie du capteur d'images. Dans ce cas, les pixels sont lus à la demande, une fois que l'information est considérée comme significative. Cette approche utilisée

par les concepteurs des capteurs d'images événementiels est détaillée dans la suite du manuscrit.

### 2.6.2 Conversion analogique numérique

La conversion analogique numérique limite en premier lieu la vitesse de lecture séquentielle de la matrice et participe de manière forte à la consommation électrique du système de vision. Ces limitations deviennent de plus en plus contraignantes suite à l'augmentation accrue de la résolution du capteur d'images ainsi que de l'augmentation du *frame rate* désiré.

Dans le but de réduire le temps de conversion et d'augmenter par la suite la vitesse de lecture du capteur d'images, de nouvelles architectures de convertisseur analogique numérique colonne ont été proposées ces dernières années. Dans le cadre des convertisseurs colonne, les architectures des convertisseurs à approximations successives [Mar+11 ; Mat+09 ; YFE99], celle des convertisseurs cycliques [Tan+15 ; Lim+11 ; Fur+07], à simple rampe [SI10 ; NA10 ; NAA09 ; Sno+07a] et à rampes multiples [Sno07] ont été étudiées. En fait, la consommation électrique de ce convertisseur croît proportionnellement avec la vitesse de lecture. Étant le composant électronique le plus consommant dans une chaîne de lecture conventionnelle, des convertisseurs à basse consommation dédiés aux capteurs d'images ont été présentés dans la littérature, par exemple [Che+14 ; OG14 ; Han+13]. Ce thème constitue un axe de recherche et de développement fort actuellement tant dans l'industrie que dans la recherche publique.

Ainsi, suivant la gamme d'applications visée du capteur d'images, un compromis entre la surface du système de lecture, la vitesse de conversion et la consommation électrique du convertisseur sera impérativement nécessaire lors de la conception du convertisseur analogique-numérique.

### 2.6.3 Dynamique de fonctionnement

La dynamique de fonctionnement des capteurs de vision conventionnels est limitée par la capacité de conversion photoélectrique du pixel, le gain commun à tous les pixels, le temps d'intégration fixe, le courant d'obscurité de la photodiode et le *SNR* l'électronique. Pour un temps d'intégration fixe, on peut augmenter la dynamique de fonctionnement en diminuant le bruit de lecture des pixels, en d'autres termes en augmentant le *SNR*. Une autre approche pour augmenter la dynamique de fonctionnement consiste à diminuer le courant d'obscurité afin d'augmenter la sensibilité du pixel.

Depuis de nombreuses années, des capteurs d'images à grande dynamique ont été proposés dans la littérature afin d'améliorer la qualité de l'image résultante, par exemple [Liu12 ; Spi+09 ; Dar08]. Ces capteurs d'images à grande dynamique introduisent des approches qui augmentent certes la dynamique de fonctionnement, mais qui induisent des inconvénients tels que l'augmentation du bruit spatial fixe, de la surface du pixel ou

du temps d'affichage de l'image. Une première approche consiste à utiliser des pixels à réponse logarithmique [Lab07 ; Kav+00] ou des circuits non linéaires permettant d'augmenter la dynamique (voir section 2.3.2.2). Cependant, cette approche implique une réponse non linéaire plus difficile à exploiter lors de la reconstruction de l'image. Une deuxième approche consiste à appliquer une capacité latérale de débordement pour améliorer la dynamique de fonctionnement [Ide+07 ; Aka+06 ; Sug+05]. Cette approche augmente le courant d'obscurité de la photodiode et par la suite le bruit de grenaille lié à ce courant. La troisième approche consiste à adopter des temps d'intégration à valeurs multiples [Sch+02]. Mais cette approche implique dans certains cas une discontinuité des rapports signal sur bruit (*SNR*) aux points de changement du temps d'intégration, une diminution du *Frame Rate* et des traitements d'images supplémentaires [Liu12]. De plus, cette approche introduit éventuellement des distorsions liées aux mouvements dans l'image.

## 2.7 Les capteurs d'images asynchrones

Après avoir abord les capteurs d'images conventionnels, on consacre cette section à présenter l'état de l'art des capteurs d'images événementiels. En fait, les capteurs d'images événementiels sont à l'origine des systèmes de vision bio-inspirés, conçus et développés afin d'imiter les systèmes biologiques. Les chercheurs visaient une compréhension approfondie de la Nature ce qui leur permettrait par la suite de bénéficier des avantages de ces systèmes biologiques. Ainsi, l'évolution et le développement des capteurs d'images événementiels a rendu ces derniers suffisamment performants pour commencer à être des alternatives pertinentes aux capteurs d'images conventionnels en étant capables de surmonter certaines de leurs limitations. Ainsi, on débute cette section en présentant brièvement les systèmes neuromorphiques, pour passer ensuite au principe des imageurs asynchrones, et pour finir cette section par présenter les capteurs d'images asynchrones existant dans la littérature.

### 2.7.1 Les systèmes neuromorphiques

Malgré les progrès impressionnants qui ont bouleversé les technologies, de la micro-électronique et de l'informatique, les systèmes sensoriels et d'information artificiels demeurent considérablement moins efficaces dans le domaine du traitement en temps réel que leurs homologues biologiques [PSG14]. Ainsi, la Nature s'avère être une source d'inspiration pour les ingénieurs et ce depuis l'Antiquité.

En particulier, le système nerveux est depuis longtemps le sujet de recherche de plusieurs disciplines afin d'étudier son fonctionnement, sa structure ainsi que son développement. Ainsi, une nouvelle science interdisciplinaire est née : l'ingénierie neuromorphique. Les systèmes neuromorphiques dédiés à la vision ont été originellement

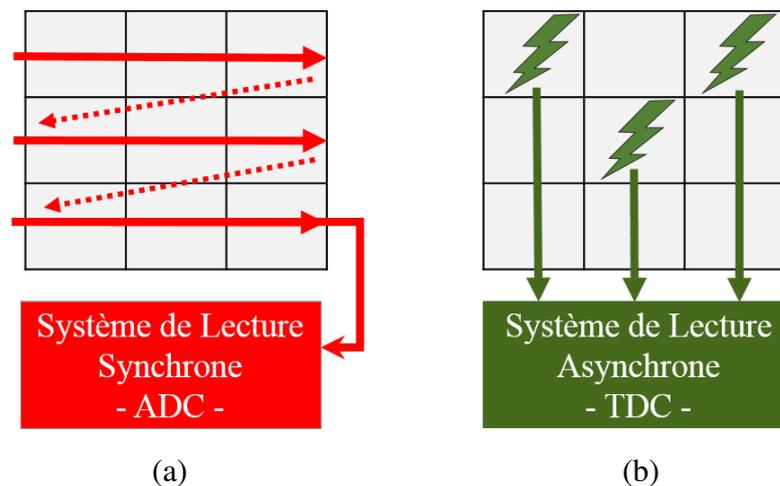
introduits par Carver Mead à la fin des années 80 [Mea89]. Ces systèmes bio-inspirés ont été conçus dans le but de bénéficier des caractéristiques avantageuses des systèmes neuronaux de la rétine. En se basant sur le fait que la puissance du cerveau est environ 10 millions de fois plus efficaces que la meilleure technologie numérique possible à l'époque [Mea90], Carver Mead a prédit un succès à long terme des systèmes bio-inspirés. Les systèmes neuromorphiques analogiques, numériques et mixtes adoptent un fonctionnement bio-inspiré en implémentant d'une part des modèles de comportements neuronaux et d'autre part des algorithmes de traitement informatiques simulant le fonctionnement neuronal biologique. Ce domaine, considéré comme étant une ouverture vers une nouvelle philosophie dans le monde de l'électronique, a été implémenté pour des applications de détection, de contrôle et de perception auditive et visuelle [Lic06b]. Ainsi la conception des systèmes neuromorphiques se situant à la limite de la biologie et inspirant les domaines de l'électronique, des mathématiques, de traitement de signal, de l'informatique et de l'ingénierie nous offrent un domaine interdisciplinaire fascinant et prometteur.

En prenant l'exemple d'un système de vision biologique, couramment connus comme étant "l'œil", d'un point de vue technique ce système profite des capacités d'adaptation aux variations lumineuses, une dynamique allant jusqu'à 160 dB ainsi qu'un pré-traitement puissant et efficace de l'information capturée au niveau de la rétine biologique. Ainsi, ces performances ont créé un nouvel axe de recherche qui vise la conception de systèmes de vision électroniques imitant le système de vision biologique.

### 2.7.2 Principe des imageurs asynchrones

Les systèmes de vision numériques conventionnels se basent sur la capture de la lumière au niveau des pixels photosensibles et l'extraction séquentielle et exhaustive de l'information lumineuse grâce à un système de lecture gouverné par un signal d'horloge. Ce type de fonctionnement nécessite un compromis entre d'une part la consommation électrique du système et d'autre part la vitesse de lecture et la résolution spatiale du capteur. En outre, le protocole de lecture des capteurs d'images conventionnels est totalement indépendant de l'occurrence des événements spatio-temporels de la scène au niveau du capteur d'images (voir 2.6). Cela veut dire qu'une partie potentiellement grande de l'énergie utilisée pour lire et numériser une image sert à extraire une information invariante par rapport à l'image précédente par exemple. On peut donc en déduire que les capteurs d'images classiques ne sont pas très efficaces d'un point de vue énergétique.

Le principe des capteurs d'images bio-inspirés repose sur la transmission, la lecture et le traitement des pixels actifs, en d'autres termes, les pixels qui expriment un nouvel événement. Une représentation symbolique des protocoles de lecture des imageurs conventionnels et asynchrones est illustrée par la figure 2.12. Contrairement à la lecture séquentielle exhaustive de l'ensemble des pixels du capteur conventionnel, le système de lecture de l'imageur asynchrone répond exclusivement aux événements au niveau



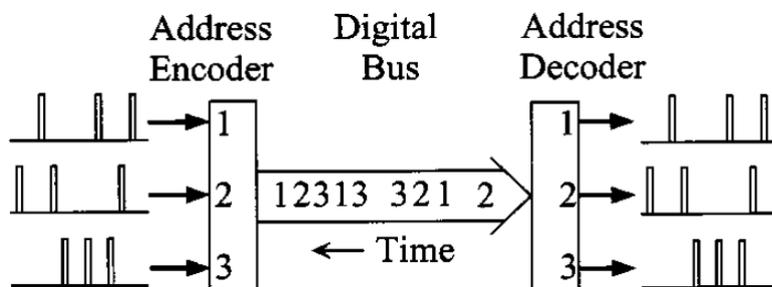
**FIGURE 2.12** – Représentation symbolique de la différence entre le protocole de lecture du système de vision conventionnel synchrone (a) et celui du système de vision événementiel asynchrone (b)

des pixels. En fait, la communication entre les pixels et le système de lecture est établie grâce à un protocole de communication bio-inspiré basé sur la représentation de l'événement par son adresse (*Address Event Representation (AER)*). Le protocole de communication *AER* ainsi que le codage des pixels sont détaillés par la suite.

### 2.7.3 Protocole de lecture asynchrone : Address Event Representation (*AER*)

*Mahowald* et *Sivilotti* ont été les premiers à proposer l'utilisation d'une représentation adresse-événement afin de transmettre des trains d'impulsions d'un réseau de neurones et établir par la suite une communication entre des puces multiples [[Boa00](#) ; [Mah94](#) ; [Mah92](#) ; [Siv91](#)]. Le protocole de communication par représentation de l'événement par adresse (*Address Event Representation : AER*) consiste à transmettre les événements consécutifs en indiquant successivement leurs adresses. La figure 2.13 illustre le fonctionnement de cette communication d'événements. Les événements codés par des impulsions de largeur et hauteur fixes sont envoyés vers un encodeur d'adresse. A son tour, l'encodeur d'adresse, situé au niveau de la puce émettrice, génère pour chaque impulsion reçue une adresse binaire unique à chaque pixel. Les adresses générées sont ensuite transmises via un bus vers la puce réceptrice. Le récepteur, couplé à un décodeur d'adresse, déchiffre le flux binaire en recréant les pulsations tout en respectant leur ordre chronologique. Ainsi, les pulsations des neurones actifs et leurs adresses sont séquentiellement transmises grâce à un bus numérique. Cependant ce protocole de communication est restreint par les limitations électroniques qui empêche le système d'atteindre

les performances du système nerveux biologique en termes de nombre de connexions parallèles possibles, de robustesse et de vitesse de fonctionnement.



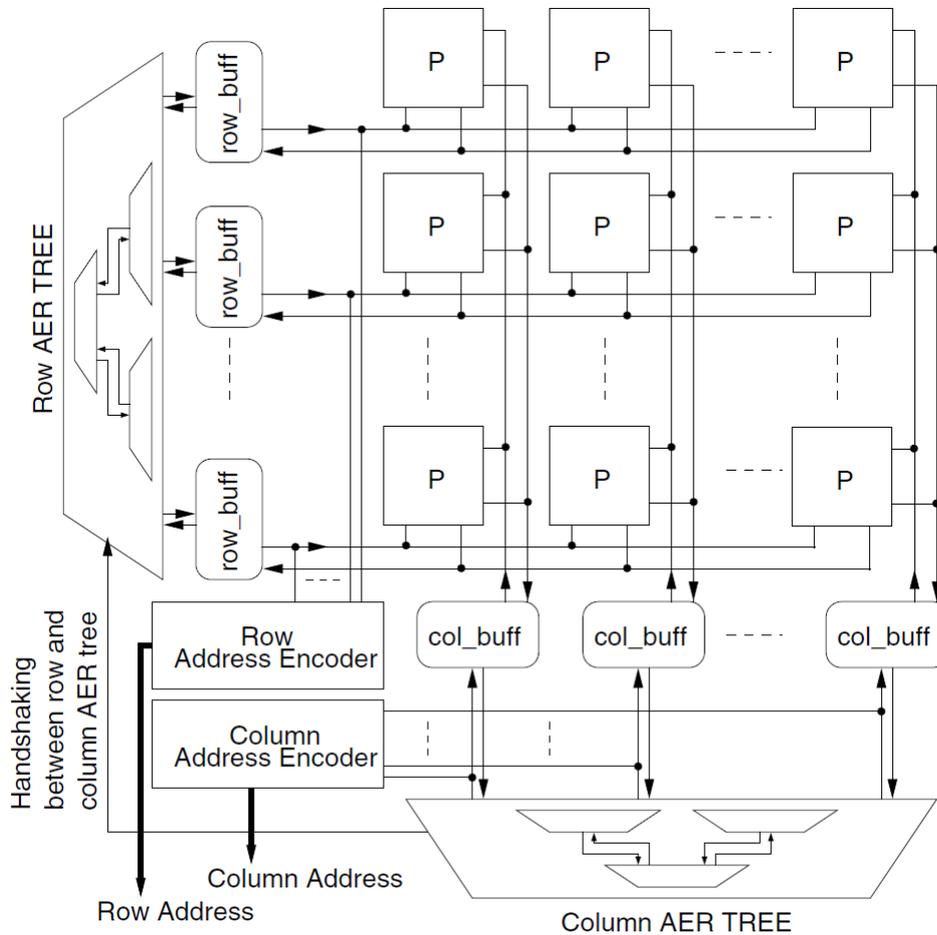
**FIGURE 2.13** – Le protocole de communication d'événement par adresse [Mah92]

Étant des systèmes neuromorphiques, les capteurs d'images bio-inspirés utilisent le même principe de transmission de l'information du système nerveux et par la suite le même protocole de communication *AER*. Dans le cas d'un capteur d'images, le pixel est chargé de détecter l'événement et d'initier le procédé de lecture. Par conséquent, la structure du pixel actif conventionnel doit être adaptée au fonctionnement événementiel et à un éventuel traitement d'images intégré.

Suite à l'occurrence d'un événement, le pixel sollicite l'accès au bus de sortie numérique à grande vitesse. Une fois l'accès au bus assuré, le pixel envoie une requête de lecture et communique simultanément son identité (en termes d'adresse), à travers le bus de sortie associé au système de lecture asynchrone. Par la suite, le système de lecture asynchrone répond en envoyant un signal d'acquiescement au pixel actif. La signalisation requête-acquiescement établie entre le pixel événementiel et le système de lecture asynchrone est identique à un protocole de communication asynchrone dit *Handshaking Protocol* employé lors de la conception des systèmes numériques asynchrones (Voir chapitre 3).

Le fonctionnement événementiel de chaque pixel implique que des requêtes de lecture simultanées peuvent être reçues par le système de lecture asynchrone, créant ainsi des conflits de lecture et éventuellement une perte de données. Afin de surmonter les complications liées à la lecture événementielle, des solutions ont été proposées dans la littérature. En fait, un circuit de lecture d'imageurs événementiels basé sur le principe d'arbitrage et a été initialement proposé par *Sivilotti* [Siv91] dans le but de gérer les requêtes de lecture des pixels simultanées. L'approche d'arbitrage proposée consiste aussi à attribuer, dans certains cas, à un nombre de pixels ou de groupe de pixels une priorité par rapport au reste de la matrice. Dans ce cas, le circuit de lecture asynchrone du capteur d'images sélectionne les requêtes prioritaires et leur permet l'accès au bus de communication [Mya+11]. Cependant, le principe d'arbitrage souffre des inconvénients liés aux erreurs de timing, de priorité fixe ainsi que de la circuiterie complexe du système d'arbitrage. En fait, la surface du circuit de l'arbitre augmente exponentiellement

avec la résolution du capteur d'images. En conséquence, l'architecture et le principe d'arbitrage ont subi des développements majeurs [Boa04a ; Boa04b ; Boa04c].



**FIGURE 2.14** – L'architecture typique d'un circuit de lecture AER pour un imageur événementiel [Mya+11]

Afin de résoudre le problème de priorité fixe, un arbitre à priorité adaptative et qui applique une priorité alternée entre les pixels a été implémenté [Sho07]. Une autre solution a été proposée par *Culurciello* : le principe du capteur d'images ALOHA [CA04]. Cette approche, qui résout aussi le problème des requêtes de lecture simultanées, consiste à supprimer les requêtes causant le conflit de lecture. Cependant, l'inconvénient de cette approche réside dans la perte de données suite à la suppression des requêtes de lecture. Des solutions ont été aussi proposées afin de résoudre la problématique des erreurs de timing. En fait, celles-ci sont dues aux délais de traitement des requêtes de lecture. Afin de diminuer ces erreurs, de nouvelles architectures pour les circuits d'arbitrage dédiés aux imageurs AER ont été proposées. Une de ces architectures proposée par [Sho07] s'appuie sur l'augmentation de la taille de la base de l'arbitre, en d'autres termes le

nombre d'entrées de l'arbitre. Une autre approche, visant la diminution de l'erreur de timing, proposée par [XLY13] se base sur le concept de correction de timing. En d'autres termes, cette approche cherche à mesurer l'erreur de timing associé à chaque pixel et à l'envoyer simultanément à la demande de lecture. Par ailleurs, la correction est appliquée au niveau du circuit de traitement et de lecture asynchrone.

Malgré les limitations et la complexité de l'arbitre, le principe d'arbitrage est adopté et implémenté par la majorité des concepteurs d'imageurs événementiels asynchrones grâce à l'ajout d'un circuit électronique qui gère les signaux de requêtes et d'acquittements au sein des pixels et qui établit la communication entre les pixels et le circuit de lecture [PSG14 ; Par+14 ; SLB11 ; PMW11 ; PMW10 ; PMW08 ; Lic08 ; SB07 ; SB06 ; Lic06b ; Lic06a ; LPD06 ; SB05 ; AAH05 ; LDK04 ; Kra02 ; Guo02]. Ainsi, l'adoption du fonctionnement événementiel et du protocole de communication *AER* réduit l'activité du capteur d'images événementiel et par conséquent son flux de données en sortie.

## 2.7.4 Codage de l'information lumineuse du pixel

### 2.7.4.1 Codage Temporel Bio-inspiré : Time-to-First-Spike (TFS)

Dans le but de comprendre la réaction des neurones à différents stimuli, le codage, le décodage et la transmission des événements à travers les neurones du système nerveux ont toujours été des problématiques fondamentales dans le domaine des neurosciences [PB68]. D'une part, le codage neuronal exprime les réponses des neurones à des stimuli bien définis. En fait, les neuroscientifiques s'appuient sur le codage neuronal afin de construire des modèles qui peuvent prédire des réponses neuronales à des stimuli. D'autre part, le décodage neuronal sert à identifier et reconstruire un stimulus en se basant sur une réponse neuronale impulsionnelle. Dans la littérature, on trouve plusieurs types de codage et de décodage neuronaux. En 1926, *Edgar Adrian* et *Yngve Zotterman* ont proposé le premier codage neuronal : le codage par taux d'impulsions neuronales. En fait, ce codage se base sur le débit des impulsions neuronales générées par les neurones actifs [AZ26]. Ce premier codage adopté dans la littérature est basé sur la présence d'une relation entre l'intensité du stimulus et le taux d'impulsions (*spike*) transmises. En d'autres termes, la fréquence des impulsions augmente avec l'intensité du stimulus [For14]. Cependant, l'efficacité de ce codage diminue face à des stimuli rapides. De plus, ce codage ne prend pas en compte l'aspect temporel d'une séquence d'impulsions : l'instant d'occurrence de la première impulsion et la différence de temps entre les impulsions d'une même séquence ne sont pas pris en compte.

Une deuxième approche de codage a été proposée dans le domaine des neurosciences et qui prend en compte l'aspect temporel des impulsions neuronales. Cette approche, contrairement au codage basé sur le débit des impulsions neuronales, prend en compte l'aspect temporel de la séquence d'impulsions. Ainsi, le codage temporel est adopté par les études où la séquence d'impulsions neuronales est à haute fréquence [But+07 ; Tho90]. Dans le domaine des stimuli visuels, il a été proposé que l'information visuelle

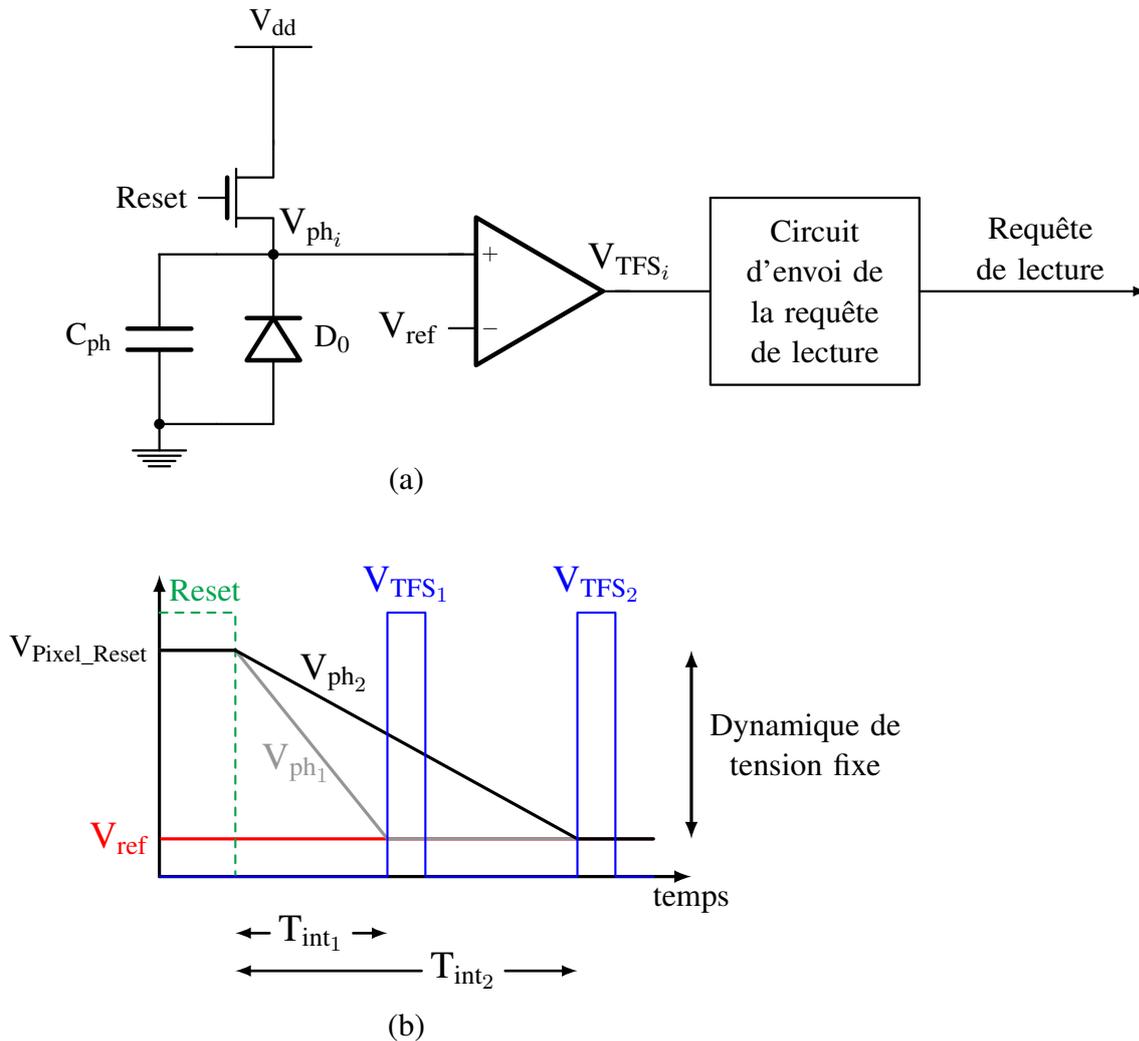
soit codée grâce à un codage temporel étudiée sur les neurones de la rétine. Plus précisément, l'information visuelle est codée par le temps écoulé entre le début du stimulus et l'occurrence de la première impulsion neuronale, également connu par le terme *Time-to-First-Spike* [GM08]. L'efficacité et l'importance du codage (*Time-to-First-Spike : TFS*) a été démontrée pour le codage d'information rétinien [GM08 ; VT01]. Ainsi, ce codage a été adapté par les concepteurs des imageurs bio-inspirés afin de représenter les événements au niveau des pixels. Depuis, les capteurs d'images bio-inspirés à base des pixels événementiels à codage *TFS* sont souvent couplés au protocole de lecture *AER* et sont considérés comme étant une alternative intéressante aux capteurs d'images conventionnels. En effet, le concept événementiel des ces imageurs mène à avoir potentiellement des capteurs d'images de haute résolution avec un fonctionnement à basse consommation et une grande dynamique de fonctionnement [Guo02].

#### 2.7.4.2 Conversion tension-temps (Time-to-Digital Conversion : TDC)

Le fonctionnement événementiel du capteur d'images asynchrone modifie en plus du protocole de lecture, le codage de l'information lumineuse du pixel. En fait, l'information lumineuse transmise est encodée par l'instant de l'occurrence de son impulsion au niveau de l'arbitre ou de l'encodeur d'adresse. Ainsi, le système de vision asynchrone code l'information de chaque pixel par l'instant de réception de sa requête de lecture au lieu de la tension aux bornes de sa photodiode. Par conséquent, le bloc de conversion analogique numérique n'est plus utile dans l'architecture de lecture de l'imageur et est remplacée par une conversion tension-temps (*Time-to-Digital Conversion : TDC*). Cette section est consacrée à l'explication de l'architecture des pixels à codage temporel *TFS* ainsi que leur fonctionnement.

La figure 2.15a illustre l'architecture typique d'un pixel événementiel *TFS*. En effet, le pixel *TFS*, composé d'un transistor d'initialisation *Reset* et d'une photodiode polarisée en inverse, assure le fonctionnement événementiel grâce à un comparateur qui se comporte comme un convertisseur analogique-numérique à 1 bit (*1-bit ADC*). En fait, la comparaison entre la tension aux bornes de la photodiode ( $V_{ph}$ ) et une tension prédéfinie par le concepteur ( $V_{ref}$ ) entraîne la détection des nouveaux événements. En d'autres termes, lorsque la tension ( $V_{ph}$ ) croise la tension ( $V_{ref}$ ), on considère que le pixel est actif et contient une nouvelle information qui doit par la suite être lu.

Afin de visualiser le fonctionnement du pixel *TFS*, on se réfère au chronogramme dans la figure 2.15b. Ce chronogramme illustre le fonctionnement de deux pixels événementiels soumis à différentes luminosités et initialisés à une tension  $V_{Pixel\_Reset}$  avec  $V_{Pixel\_Reset} = V_{dd} - V_{th}$  sachant que  $V_{th}$  est la tension de seuil du transistor d'initialisation. La lumière incidente, générant suivant l'effet photoélectrique de la photodiode un courant photogénéré, décharge la capacité parasite ( $C_{ph}$ ) et provoque par la suite la diminution de la tension aux bornes de la photodiode ( $V_{ph_i}$ ). Cependant, la différence entre les intensités lumineuses incidentes implique des multiples taux de décrémenta-



**FIGURE 2.15** – (a) L'architecture typique du pixel événementiel *TFS* (b) Chronogramme de deux pixels événementiels *TFS* soumis à des différentes valeurs de luminosité

tion. Ce dernier est proportionnel à l'intensité lumineuse et donc au courant photogénéré. Lorsque la tension  $V_{ph_i}$  atteint la tension de référence ( $V_{ref}$ ), la sortie du comparateur bascule vers l'état actif (l'état actif dans cet exemple est l'état haut). Cette impulsion indique la présence d'un nouvel événement au niveau du pixel qui se traduit par l'envoi d'une requête de lecture vers le système de lecture asynchrone. Dans ce cas, l'information du pixel est codé par l'instant d'occurrence de l'impulsion  $V_{TFS_i}$  en ayant comme référence l'instant du début de l'intégration. Dans le domaine de l'imagerie, le temps *TFS* est également connu comme étant le temps d'intégration du pixel, c'est le temps écoulé entre la phase d'initialisation et la phase de lecture du pixel. En effet, les capteurs d'images basés sur les pixels événementiels *TFS* ne fonctionnent pas suivant un temps d'intégration unique et commun à tous les pixels. En revanche, chaque pixel *TFS*

événementiel fonctionne suivant son propre temps d'intégration. Dans ce cas, le temps d'intégration du pixel événementiel dépend du courant photogénéré et donc de l'intensité de la lumière incidente. En fait, le temps d'intégration d'un pixel est inversement proportionnel à son courant photogénéré. Selon l'exemple de la figure pixel 2.15b, les deux pixels, soumis à différentes luminosités, finissent par avoir des temps d'intégration distincts. Le premier pixel ( $V_{ph_1}$ ), étant soumis à une luminosité supérieure à celle du deuxième pixel ( $V_{ph_2}$ ), son temps d'intégration ( $T_{int_1}$ ) est inférieur au temps d'intégration du deuxième pixel ( $T_{int_2}$ ).

L'équation (2.7) exprime la relation inversement proportionnelle entre le courant photogénéré et le temps d'intégration. Sachant que  $I_{ph}$  est le courant photogénéré,  $V_{ref}$  est la tension de référence et  $V_{Pixel\_Reset}$  est la tension d'initialisation de la photodiode du pixel ( $V_{Pixel\_Reset} = V_{dd} - V_{th}$ ).

$$I_{ph_i} = C_{ph} \cdot \frac{V_{Pixel\_Reset} - V_{ref}}{T_{TFS}} = C_{ph} \cdot \frac{V_{Pixel\_Reset} - V_{ref}}{T_{int_i}} \quad (2.7)$$

Le codage temporel *TFS* de l'information lumineuse des pixels a amélioré considérablement les performances des capteurs d'images événementiels. En fait, l'intégration de ce type de codage au sein des pixels limite l'activité du système de vision aux événements pertinents et diminue le flux de données à traiter, sauvegarder et par la suite visualiser [FDS15 ; DFS14 ; SB06]. Par conséquent, la consommation électrique du capteur d'images est réduite [SLB11 ; SB05]. De plus, l'absence d'un temps d'intégration unique pour toute la matrice et la dépendance des temps d'intégration avec la lumière incidente sur chaque pixel événementiel augmente la dynamique de fonctionnement de l'imageur [PMW11 ; CEC04 ; CECB01] devenant des capteurs d'images à grande dynamique (*High Dynamic Range : HDR*) natifs. En outre, les capteurs d'images numériques dont la conversion analogique numérique (*1-bit ADC*) est intégré au sein du pixel numérique (*Digital Pixel Sensor : DPS*) parviennent à atteindre un rapport signal sur bruit élevé. Par conséquent, le codage temporel de l'information des pixels événementiels réforme l'expression du rapport signal sur bruit *SNR* d'un pixel *TFS* (2.8)(2.9) [Sho07 ; Guo02].

$$SNR_{TFS \text{ Pixel}} = 10 \cdot \log \left( \frac{C_{ph} \cdot (\text{Dynamique de tension})}{q} \right) \quad (2.8)$$

$$SNR_{TFS \text{ Pixel}} = 10 \cdot \log \left( \frac{C_{ph} \cdot (V_{Pixel\_Reset} - V_{ref})}{q} \right) \quad (2.9)$$

Ainsi, les avantages de ce type de codage temporel ont poussé les concepteurs d'imageurs bio-inspirés événementiels à considérer le codage *TFS* et la conversion tension-temps comme une alternative avantageuse. Originellement, cette approche a été adoptée par [QGH04 ; LH04], mais dès lors cette approche a été développée par des différents groupes de recherche [Dar+15 ; DFS15 ; DFS14 ; PMW11 ; PMW10 ; PMW08 ; Sho07 ; SB07 ; SB05 ; CEC04].

Néanmoins, le pixel à codage temporel *TFS* présente des inconvénients liés en premier lieu à la surface du pixel. En effet, en faisant appel à un convertisseur analogique numérique sur 1 bit et en ajoutant des blocs de gestions des requêtes de lecture au sein du pixel, cela augmente la surface globale de ce dernier et diminue son facteur de remplissage. En deuxième lieu, même si les capteurs d'images événementiels bénéficient d'une grande dynamique de fonctionnement en comparaison aux capteurs d'images conventionnels, cette dynamique demeure contrôlée et limitée par le nombre de bits du convertisseur tension-temps, en d'autres termes par le nombre de bits du *timer* qui enregistre les instants d'occurrence des événements et des requêtes de lecture des pixels.

## 2.8 Conclusion

Dans ce chapitre, l'état de l'art des capteurs d'images a été présenté en commençant par une comparaison entre les imageurs CCD et CMOS. Ensuite, la chaîne de lecture et de traitement des capteurs d'images CMOS a été décrite. En premier lieu, l'élément photosensible du pixel CMOS a été détaillé, suivi par la présentation des architectures typiques des pixels logarithmiques et standards ainsi que de la chaîne de lecture principalement basée sur un convertisseur analogique-numérique (*ADC*) afin de numériser l'information lumineuse des pixels. Ensuite, on a présenté les principales caractéristiques des capteurs d'images CMOS comme le facteur de remplissage, la sensibilité du pixel, la dynamique de fonctionnement et la résolution. Par ailleurs, les limitations des capteurs d'images conventionnels ont été aussi présentées dans le but de souligner les points faibles en cours de développement.

Dans le but de présenter une alternative aux imageurs conventionnels, on a détaillé et expliqué le concept des imageurs événementiels, dits asynchrones. On a alors listé les travaux déjà existant dans la littérature tout en détaillant et expliquant leur principe et leurs différents aspects. En outre, le principe des capteurs d'images asynchrones a été expliqué, en insistant sur la représentation de l'événement par son adresse (*Address Event Representation : AER*) et du système de codage temporel (*Time-to-First-Spike : TFS*) de l'information au niveau du pixel. En adoptant ces deux approches, le pixel intègre alors un convertisseur analogique numérique sur un seul bit, ce qui fait que le pixel événementiel *TFS* est un pixel numérique (*Digital Pixel Sensor : DPS*). En fait, le pixel *TFS* se comporte comme un détecteur d'événement, et active la chaîne de lecture grâce à un envoi de requête de lecture vers le système de lecture *AER* après la détection d'un événement pertinent. Cette approche implique un fonctionnement à débit de données réduit et normalement à une plus faible consommation électrique vu que seule l'information des pixels actifs est lue et traitée. De plus, le principe de la conversion tension temps (*Time-to-Digital Conversion : TDC*) de l'information lumineuse des pixels événementiels remplace le convertisseur analogique numérique du système de lecture par un convertisseur tension-temps. Ainsi, suite au remplacement de l'*ADC*, la consommation

électrique du système de vision est diminuée.

Ce chapitre a mis en évidence l'intérêt des approches alternatives aux capteurs d'images conventionnels. On présentera dans les chapitres suivants notre approche pour les capteurs d'images asynchrones. Une référence à ce chapitre sera faite tout au long de ce manuscrit, afin de réaliser une comparaison aux travaux déjà existants.

# 3

## LA LOGIQUE ASYNCHRONE

### Sommaire

---

|       |  |           |
|-------|--|-----------|
| 3.1   | Problématique des circuits synchrones . . . . .            | <b>36</b> |
| 3.2   | Conception des circuits asynchrones . . . . .              | <b>37</b> |
| 3.2.1 | Principe . . . . .   | 37        |
| 3.2.2 | La porte Muller . . . . .                                  | 37        |
| 3.3   | Protocole de communication . . . . .                       | <b>38</b> |
| 3.3.1 | Protocole à données intégrées . . . . .                    | 39        |
| 3.3.2 | Protocole double rails à quatre phases . . . . .           | 39        |
| 3.4   | Classes des circuits asynchrones . . . . .                 | <b>42</b> |
| 3.4.1 | Insensible aux délais (Delay Inensitive (DI)) . . . . .    | 42        |
| 3.4.2 | Quasi-Insensible aux délais (Quasi-Delay Inensitive (QDI)) | 42        |
| 3.4.3 | Indépendant de la vitesse (Speed Independent (SI)) . . . . | 43        |
| 3.4.4 | Micropipeline . . . . .                                    | 43        |
| 3.5   | Caractéristiques des circuits asynchrones . . . . .        | <b>46</b> |
| 3.5.1 | Absence du signal d'horloge . . . . .                      | 46        |
| 3.5.2 | Calcul à temps minimal . . . . .                           | 46        |
| 3.5.3 | Consommation électrique . . . . .                          | 47        |
| 3.5.4 | Modularité . . . . .                                       | 47        |
| 3.5.5 | Migration technologique . . . . .                          | 47        |
| 3.6   | Conclusion . . . . .                                       | <b>48</b> |

---

La logique numérique asynchrone, et par la suite les circuits asynchrones, ont introduit une nouvelle approche de l'électronique numérique et de traitement de signal non conventionnel. Le principe de la logique asynchrone a été proposé dans le monde académique pendant les années 1950 [Ung69]. Mais il a fallu attendre la fin des années 1990 pour que cette technologie ait une certaine maturité suffisante pour concevoir des circuits VLSI. En fait, le développement de la conception asynchrone a été l'objet de nombreux sujet de recherche, principalement pendant les deux dernières décennies. Cependant l'emploi de cette méthodologie non conventionnelle demeure jusqu'à nos jours limité comparativement à la conception synchrone, même si les résultats et les progrès des recherches récentes sont prometteurs.

## 3.1 Problématique des circuits synchrones

Grâce à la simplicité de la centralisation du contrôle des circuits synchrones, ces derniers dominent l'industrie microélectronique. En effet, la conception synchrone est la méthodologie dominante depuis les années 1960 [BOF10]. Le système synchrone se compose de sous-systèmes commandés par une ou plusieurs horloges qui synchronisent les signaux et la communication entre les divers blocs. En principe, le signal d'horloge joue le rôle d'un actionneur global, ce qui implique l'ajout d'une contrainte temporelle. Cette contrainte doit être respectée par tous les composants du système synchrone. Mais si le signal d'horloge n'est pas uniformément distribué, les éléments du circuit synchrones reçoivent éventuellement le signal d'horloge déphasé. Cette imperfection est connue comme étant le gigue des horloges (*Clock Skew*). Ce phénomène influe, par la suite, le fonctionnement du circuit. En plus, si un composant du système achève le traitement de données associé rapidement, sa performance reste toujours limitée par la fréquence d'horloge lié à la vitesse du composant le plus lent. En d'autres termes, la performance de tous les composants est bornée par la performance du pire cas. En dépit du fait que plusieurs solutions sont employées afin de résoudre le problème de gigue d'horloge, ces approches correctives mènent vers une perte de performance (réduction de vitesse, augmentation de la surface) des circuits synchrones. En outre, la complexité accrue, l'augmentation de la vitesse et la diminution de la surface des circuits, rendent l'approche de la conception synchrone plus contraignante. En plus, la distribution d'horloge dans le circuit synchrone est une source de consommation majeure et augmente avec la fréquence de fonctionnement. Ce problème est critique lors de la conception des systèmes à faible consommation.

En outre, le système d'horloge globale limite la modularité qui un aspect important actuellement dans la conception des systèmes synchrones. Autrement dit, l'ajout ou le remplacement d'un sous-système ne peut pas être fait sans valider les hypothèses temporelles globales du circuit. La fréquence d'horloge doit être adaptée à la vitesse du nouveau sous-bloc, d'où la difficulté de combiner des sous-systèmes. En résumé, on peut considérer que la modularité qui est un aspect important de la conception des circuits ne peut pas être mis en place aisément dans le domaine synchrone.

Enfin, il existe de nombreuses difficultés limitant la conception des systèmes synchrones. En fait, le développement des systèmes complexes, ainsi que les hautes performances demandées ont orienté les concepteurs vers une nouvelle approche de conception basée sur la logique asynchrone. L'idée générale de l'approche asynchrone est basée sur l'utilisation d'une synchronisation locale au lieu d'une synchronisation globale gouvernée par un signal d'horloge. En effet, la logique asynchrone nous permet d'éviter les contraintes liés à l'horloge globale.

## 3.2 Conception des circuits asynchrones

### 3.2.1 Principe

Contrairement à un circuit synchrone gouverné par un ou plusieurs signaux d'horloge, un circuit asynchrone est un système localement contrôlé. On considère que le fonctionnement d'un circuit asynchrone est de type flot de données. En d'autre terme, le système est contrôlé par des signaux locaux qui assurent les échanges d'information entre les sous-blocs fonctionnels. Dans ce cas, l'activité d'un sous-bloc est liée à l'activité du sous-bloc précédent et suivant. En utilisant l'exemple de la figure 3.1, pour que le bloc  $N$  exécute le traitement de données associé, il doit recevoir un signal de demande de réception et de traitement des données du bloc précédent  $N-1$ . Ce signal est connu comme étant le signal de requête (Figure 3.1 : *Requête N-1*). Afin de libérer le bloc précédent, le sous bloc  $N$  lui envoie un signal de confirmation de la réception des données. Ce signal est connu sous le nom du signal d'acquiescement (Figure 3.1 : *Acquiescement N*). Le sous-bloc précédent ( $N-1$ ) est ainsi libéré et peut par la suite recevoir de nouveau des données. Ainsi, une fois le sous bloc  $N$  a accompli sa tâche, il envoie à son tour le signal de requête  $N$  au bloc suivant ( $N+1$ ) afin de lui envoyer par la suite le signal de données. Le bloc  $N+1$  répond alors en envoyant le signal d'acquiescement  $N+1$  afin de libérer le bloc  $N$ . En résumé, chaque bloc est en attente des signaux de contrôle et de données en provenance du bloc précédent, les traite et les envoie ensuite au bloc suivant.

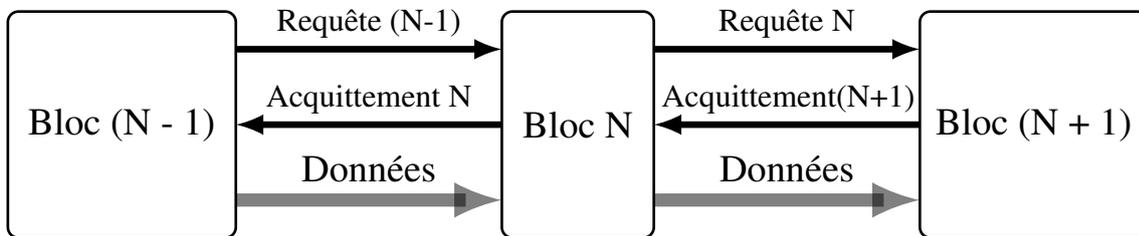


FIGURE 3.1 – Schéma fonctionnel d'un circuit asynchrone

L'exemple, dans la figure 3.1, illustre un contrôle par signalisation bidirectionnelle utilisée dans le but de synchroniser les sous-blocs du système asynchrone. Ce protocole de communication est connu sous le nom de *Handshaking Protocol*. Plusieurs protocoles de communication seront détaillés par la suite. [BOF10 ; NS15a ; NS15b ; SF01 ; Ung69]

### 3.2.2 La porte Muller

Pour un circuit synchrone, l'horloge gère la notion de temps et détermine les points temporels où les signaux sont considérés comme valides et stables. A part ces points,

l'état des signaux n'est pas pris en compte. C'est pour cela, que les aléas du signal entre deux fronts d'horloge sont non significatifs du point de vue fonctionnel. Néanmoins, pour un circuit asynchrone de contrôle, l'absence d'horloge signifie que toutes les transitions des signaux sont pertinentes et sont prises en compte. Par conséquent, tout aléas au niveau des signaux est intolérable.

Pour implémenter les protocoles de communication (*Handshaking Protocol*) des circuits asynchrones, les portes logiques élémentaires ne suffisent pas. Pour cela on fait appel à la porte de Muller proposées par David Eugene Muller [Mul63 ; MB59]. La porte Muller aussi connu sous le nom de l'élément C (*C-element*) est essentielle vu qu'elle permet de matérialiser le rendez-vous entre plusieurs signaux.

La figure 3.2 illustre le symbole et la table de vérité d'une porte Muller. Pour une porte Muller conventionnelle, la sortie  $z$  maintient son niveau logique "0" tant que toutes les entrées sont à l'état bas. Lorsque les deux entrées  $x$  et  $y$  basculent vers l'état "1", la sortie  $z$  passent au niveau logique "1". A part ces deux cas, la sortie  $z$  maintient sa dernière valeur. En d'autre terme, la sortie d'une porte de Muller copie la valeur de ses entrées lorsque celles-ci sont identiques, autrement elle mémorise son état précédent.

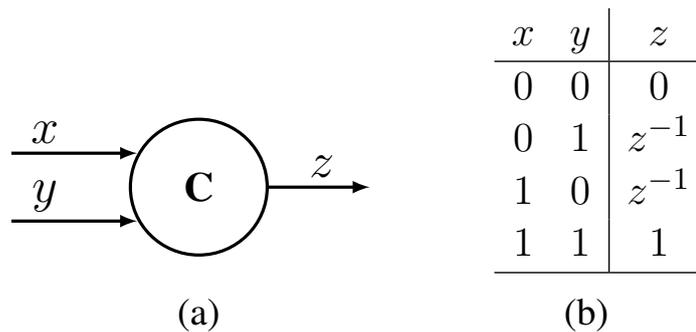


FIGURE 3.2 – La Porte Müller. Symbole (a) et table de vérité (b)

Il existe d'autres versions de la porte de Muller, on cite par exemple la porte de Muller généralisée, aussi connue sous le nom de la porte Muller dissymétrique. Les signaux d'entrée de cette porte qui font monter la sortie à "1" peuvent être distincts de ceux qui basculent son état vers "0".

Pour toutes les variétés de portes Mullers, cet élément est considéré comme étant un élément de mémorisation d'état.

### 3.3 Protocole de communication

Il existe plusieurs protocoles de communication qui assurent la synchronisation entre les différents composants d'un système asynchrone. Dans cette section, on présente

deux exemples de protocoles de communication en expliquant leur fonctionnement ainsi que le transfert de données associé.

### 3.3.1 Protocole à données intégrées

Les canaux à données intégrées se composent d'un bus de données unidirectionnel couplé avec un signal de requête et un signal d'acquiescement ou d'accusé de réception. La figure 3.3a illustre un exemple du canal à données intégrés de type *Push*. Pour ce type de canal, c'est l'expéditeur qui initie la communication et le transfert de nouvelles données valides disponibles. Les canaux à données intégrées peuvent être mis en œuvre avec un protocole à deux phases ou à quatre phases.

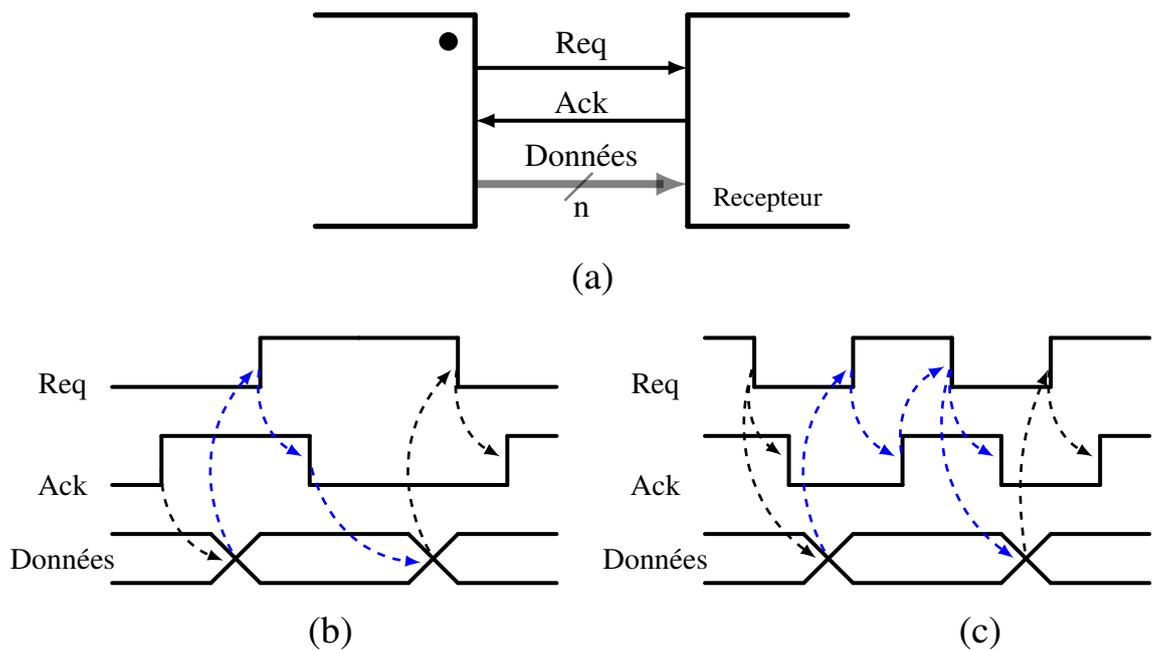
Pour un protocole à deux phases, la communication est établie grâce aux transitions des signaux de demandes de transfert de données (requêtes) et des acquiescements. Comme la figure 3.3b le montre, il n'y a pas de distinction entre les fronts montants et descendants. En d'autres termes, les deux sens de transition indiquent la disponibilité de nouvelles données valides.

Une fois que les données transmises sont absorbées par le récepteur, ce dernier bascule l'état du signal d'acquiescement. Le bloc expéditeur est par la suite libéré, peut modifier les données envoyées et bascule à son tour l'état du signal de requête. En résumé, ce sont les transitions, et non pas les états des signaux de contrôle qui initient et gèrent la communication entre les blocs du système asynchrone. Ainsi, le protocole à deux phases est souvent appelée une signalisation par transitions (*Transition Signaling*) ou une signalisation sans retour à zéro (*Non-Return-to-Zero (NRZ) Signaling*).

En revanche, pour le protocole à quatre phases, c'est l'état du signal de requête qui indique la validité et la disponibilité des nouvelles données. La figure 3.3c représente le chronogramme du protocole à données intégrées à quatre phases. En passant de la logique "0" vers la logique "1", l'état haut du signal de requête indique que des nouvelles données sont disponibles pour être transmises au bloc suivant. Idem pour le signal d'acquiescement, lorsque le récepteur consomme les données transmises, ce signal de contrôle bascule de l'état bas vers l'état haut. Ceci implique que le bloc expéditeur doit remettre le signal de requête à l'état bas et peut modifier en toute sécurité les données. Contrairement au protocole à 2 phases, les signaux de contrôle suivant le protocole à quatre phases doivent reprendre leur état bas à la fin de chaque transfert. Pour cela, quand l'expéditeur remet le signal de requête à zéro, le récepteur remet à son tour le signal d'acquiescement à zéro. Ainsi, le protocole à quatre phases est aussi connu comme étant la signalisation avec retour à zéro (*Return-to-Zero (RZ) Signaling*).

### 3.3.2 Protocole double rails à quatre phases

Le protocole double rails à quatre phases intègre les signaux de requête dans les signaux de données. Pour ce faire, on code les données en utilisant deux fils (signaux) par

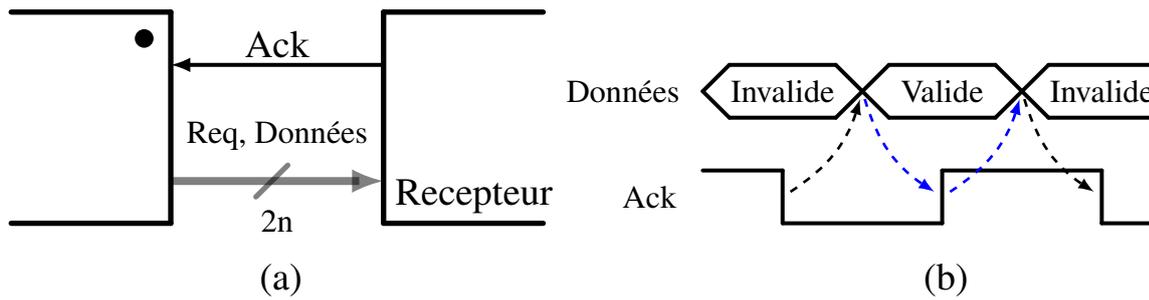


**FIGURE 3.3** – Protocole de communication à données intégrées. Illustration de l'établissement de la communication entre deux blocs (a). Chronogramme de fonctionnement du protocole de communication à données intégrées à deux phases (b) et à quatre phases (c)

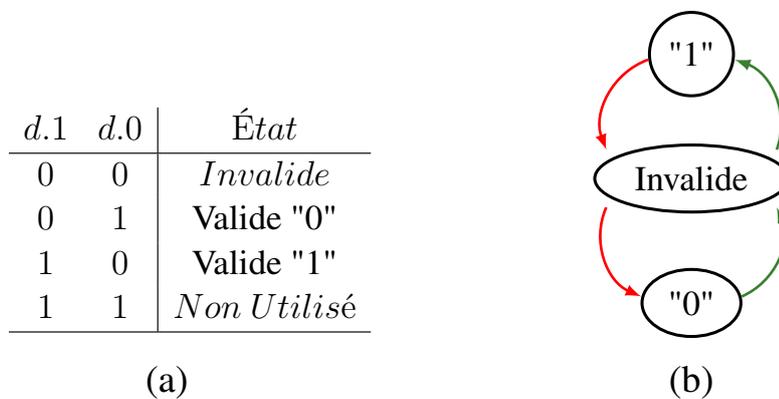
bit (Figure 3.4a). En d'autre terme, ce protocole de communication emploie deux fils (signaux) de requêtes pour chaque bit à transférer. Les signaux  $d.0$  et  $d.1$  sont utilisés pour transférer la logique "0" et "1" respectivement. Cependant, l'étage précédent est acquitté par un seul signal d'acquiescement (Figure 3.4a). Vu que les composants peuvent échanger des données indépendamment des délais des fils de connexions, ce protocole de communication manifeste une robustesse élevée vis-à-vis les variations de temps de propagation dues à la température, aux variations de tension ou des procédés de fabrication. C'est un protocole insensible aux délais.

En effet, on peut considérer la donnée  $\{d.1, d.0\}$  transférée comme étant un code ou un mot codé. Les deux valeurs possibles de ce code  $\{d.1, d.0\} = \{1, 0\}$  et  $\{d.1, d.0\} = \{0, 1\}$  illustrent les valeurs valides portant la logique "1" et "0" respectivement. La valeur  $\{d.1, d.0\} = \{0, 0\}$  représente une valeur invalide ou vide, et la paire  $\{d.1, d.0\} = \{1, 1\}$  est un code non utilisé. Le tableau 3.5a regroupe les différents états de la paire de fils  $\{d.1, d.0\}$ .

Ce codage est connu comme étant le codage à trois états. La figure 3.5b illustre le passage entre les états selon ce type de codage. On peut alors remarquer que le passage direct d'un état valide "0" à un état valide "1", ou l'inverse, est interdit. Il faut passer par l'état intermédiaire invalide, afin d'assurer un passage entre deux états valides selon le codage trois états.



**FIGURE 3.4** – Protocole de communication double rails à quatre phases. Illustration de l'établissement de la communication entre deux blocs (a). Chronogramme de fonctionnement (b)



**FIGURE 3.5** – Codage du protocole de communication double rails à quatre phases (a). Passage entre les états valides et l'état invalide (b)

En s'inspirant du fonctionnement du codage trois états, on peut déduire alors l'enchaînement des différentes phases du protocole de communication asynchrones à quatre phases (Figure 3.4b). Lorsque l'émetteur délivre une donnée valide "0" ou "1", le récepteur l'absorbe et active le signal d'acquiescement (*Ack*). Ensuite, l'émetteur délivre l'état invalide (vide). Le récepteur effectue alors une transition négative au niveau du signal d'acquiescement (*Ack*). Par la suite, l'émetteur peut transférer de nouveau une donnée valide.

Pour un canal de communication à  $N$  bits, on doit transférer  $N$  paires de fils afin de respecter le fonctionnement du protocole double rails à quatre phases. Dans ce cas, le récepteur doit détecter d'une part l'instant où toutes les paires de bits représentent des états valides et activer le signal d'acquiescement par la suite. D'autre part, il doit aussi détecter quand toutes les paires de bits représentent un état invalide, pour qu'il remette le signal d'acquiescement à zéro.

En résumé, le protocole de communication double rails à quatre phases est un canal de transfert de paires de bits en état valide séparées par des états invalides.

## 3.4 Classes des circuits asynchrones

La caractérisation de la conception des circuits asynchrones n'est pas limitée au choix du protocole à implémenter. On présente dans cette section les classes des circuits asynchrones. Ces classes manifestent une variété d'avantages et de défis associés à chaque style de conception.

### 3.4.1 Insensible aux délais (Delay Inensitive (DI))

Cette classe de circuits utilise un mode de fonctionnement purement asynchrone. Basés sur des travaux de Clark Wasley, et reformalisés par [Udd86], les circuits appartenant à cette classe n'ont aucune hypothèse temporelle à respecter. En d'autre terme, le fonctionnement de ces circuits est indépendant des délais introduits par les fils ou par les éléments logiques. Cette classe de circuit utilise le modèle de délais pour les fils et les portes logiques qui est non borné. Par conséquent, les circuits insensibles aux délais sont les plus robustes face à toutes les formes de variations parmi les circuits asynchrones. Les circuits insensibles aux délais sont supposés répondre correctement à une demande de traitement externe à condition qu'ils aient assez de temps pour le calcul et le traitement. Ceci implique l'utilisation de signaux de requêtes et d'acquittements afin de contrôler le début et la fin de la transmission et du traitement des données. Cependant, les contraintes de réalisation pratique imposées par cette classe de circuits sont très fortes. En plus, pour un circuit *DI* adoptant un modèle de délai non borné, l'utilisation des portes logiques à une seule sortie est interdite. En effet, puisque la sortie des portes logiques standards peut éventuellement alterner suite à un changement d'état d'au moins une de ses entrées, le processus d'acquittement et le temps de calcul ne sont plus respectés. En fait, la seule porte qui fonctionne comme un élément de mémorisation et respecte par la suite les temps de calcul ou de traitement c'est la porte Muller. Par contre, [Mar90] a montré que les fonctions réalisables uniquement basées sur des portes de Muller sont limitées. En conséquence, on fait appel à un modèle de circuits de type "portes complexes" pour les composants élémentaires. Les circuits insensibles aux délais sont ainsi conçus à base de composants standard complexes à plusieurs entrées et à plusieurs sorties [Hau95 ; Ebe91].

### 3.4.2 Quasi-Insensible aux délais (Quasi-Delay Inensitive (QDI))

Les circuits quasi-insensibles aux délais (*QDI*) adopte le même modèle de délai non borné pour les fils de connexions mais en ajoutant la notion de fourche isochrone (*iso-*

*chronic fork*). En effet, une "fourche" est un fil qui connecte un bloc émetteur à deux récepteurs. En assumant que les délais entre l'expéditeur et les récepteurs sont identiques, la fourche est dite isochrone. Il a été montré que l'hypothèse temporelle de la fourche isochrone est la plus faible à ajouter aux circuits insensibles aux délais (*DI*) pour les rendre réalisables avec des portes à une seule sortie [Mar93 ; Mar86]. En d'autre terme, cette hypothèse temporelle rend le modèle des circuits possible à réaliser en résolvant le problème d'utilisation des portes logiques à une seule entrée. En effet, si les fourches sont isochrones, on se permet de vérifier le signal d'acquittement d'une seule branche des branches de la fourche.

En résumé, les circuits quasi-insensibles aux délais adoptent un modèle de délais non borné pour les fils de connexion, en ajoutant en plus l'hypothèse de la fourche isochrone et un modèle de type "porte simple" pour les composants élémentaires du circuit. Cette classe de circuits est alors conçue à base des cellules standard normalement utilisées lors de la conception des circuits synchrones [DD03].

### 3.4.3 Indépendant de la vitesse (Speed Independent (SI))

Pour un circuit indépendant de la vitesse (*Speed Independent (SI)*), on considère que les délais des portes logiques sont aléatoires. Par contre, tous les délais des fils de connexion sont considérés comme étant négligeables [Mil65]. En d'autres termes, toutes les fourches sont isochrones.

Cependant, les processus de fabrication impliquent des délais sur les fils de connexion comparable mais supérieurs à ceux dans les portes. Ce modèle de délai est difficilement acceptable spécialement pour les circuits à forte intégration [BOF10].

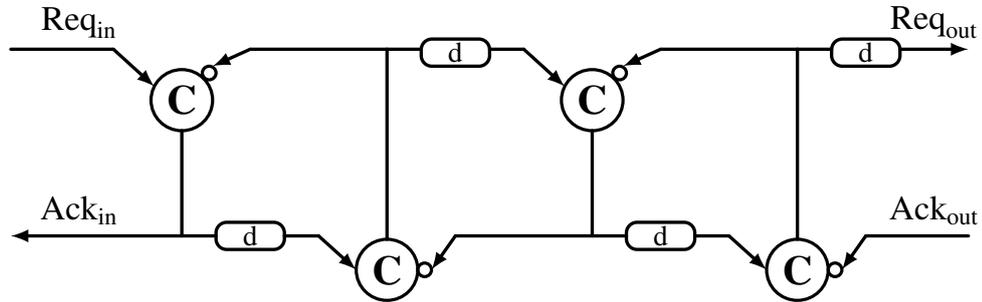
Bien qu'on essaye de toujours différencier le modèle des circuits quasi-insensibles aux délais et les circuits indépendants de la vitesse, aujourd'hui ces deux modèles produisent en synthèse des circuits souvent similaires. Par contre, l'utilisation des circuits QDI est plus avantageuse puisqu'il distingue les fourches isochrones des fourches non isochrones. Ceci implique que l'outil de vérification n'effectue que les vérifications nécessaires. On peut ainsi déduire que le modèle du circuit quasi-insensible aux délais est plus précis, plus puissant et donc plus robuste que le circuit indépendant de la vitesse [BOF10 ; Rez04].

### 3.4.4 Micropipeline

La technique micropipeline a été introduite en 1989 par Ivan Sutherland [Sut89a]. Les circuits micropipelines correspondent plutôt à une classe d'architectures de circuit asynchrone que directement à un modèle de délais de circuit asynchrone. Ces circuits sont composés d'une partie de contrôle insensible aux délais qui commande les chemins de données en utilisant le modèle de délais borné [Rez04].

La figure 3.6 illustre le circuit de contrôle d'un micropipeline. Cette structure de base se

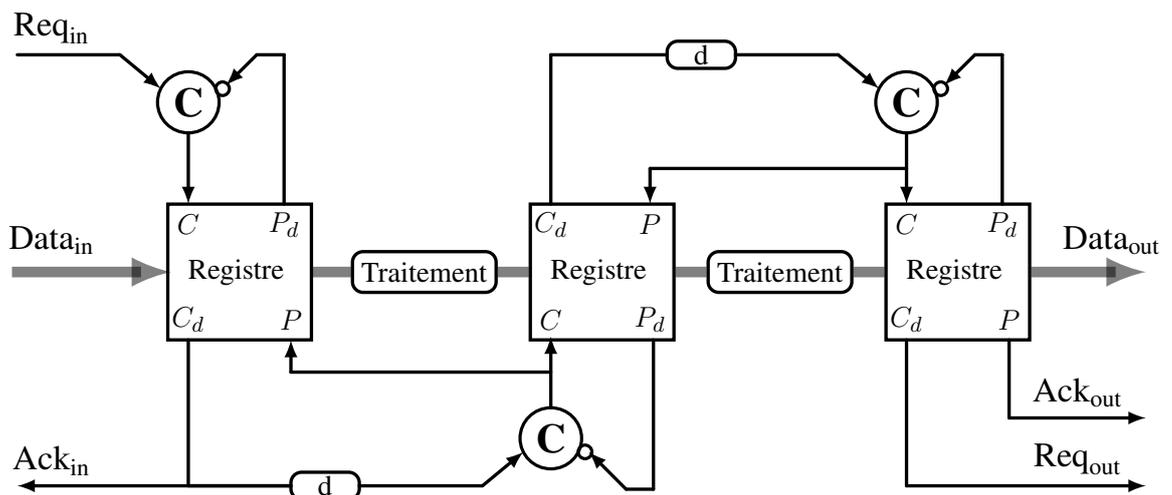
compose d'éléments identiques à base d'une porte Muller (*C-element*) connectés tête-bêche.



**FIGURE 3.6** – Structure de base du circuit de contrôle d'un micropipeline

Le fonctionnement de ce circuit asynchrone est implémenté grâce au protocole de communication à deux phases. En d'autres termes, le circuit réagit aux transitions des signaux et non pas leurs états. Dans ce cas, chaque transition est associée à un événement.

Initialement, tous les signaux sont à zéro. Ainsi, une transition positive sur  $Req_{in}$  mène une transition positive au niveau de  $Ack_{in}$ . Cette transition se propage également à l'étage suivant. Quand le deuxième étage reçoit et produit la transition positive, il la propage vers l'étage suivant et libère l'étage précédent en lui envoyant une transition positive dans le sens inverse. En d'autres termes, à la réception d'une transition positive, l'étage autorise l'étage précédent à recevoir des nouvelles transitions. De cette façon, les transitions des signaux se propagent successivement dans les étages libres. Ce fonctionnement de type FIFO est insensible aux délais.



**FIGURE 3.7** – Micropipeline avec blocs combinatoires

Cette architecture de contrôle dans un micropipeline commande les chemins de données.

Ainsi, le transfert de données entre les blocs de traitement combinatoire est contrôlé par ce type de structure. La Figure 3.7 illustre un micropipeline avec des blocs de traitement. Pour qu'on puisse transférer des données dans ce micropipeline, le premier registre doit être libre. Dans ce cas le signal  $P_d$  qui indique si le registre est libre ou occupé, est inactif. Lorsqu'une transition positive est détectée au niveau du signal  $Req_{in}$ , le signal  $C$  devient à son tour actif. Ceci implique la mémorisation des données à l'entrée du registre et une transition positive au niveau du signal  $C_d$ . En pratique,  $C_d$ , le signal  $C$  retardé, propage la requête  $Req_{in}$  vers l'étage suivant. On note que le retard ajouté lors de la propagation de  $Req_{in}$  assure la convergence de la logique combinatoire avant la capture des données résultantes par le deuxième étage. Lorsque le signal  $P$  provenant de l'étage suivant devient actif, les données mémorisées dans le registre sont transférées vers la sortie. Le premier étage est ainsi libéré et peut par la suite recevoir un nouvel événement associé à des nouvelles données. Enfin, on peut déduire que le protocole de communication asynchrone implémenté est le protocole à données intégrés à deux phases.

L'importance de ce type de circuits asynchrones réside dans le fait qu'on obtient un pipeline élastique. En d'autre terme, le nombre et la taille des données ne sont pas imposés et peuvent variés [Rig02 ; SF01], et la progression des données dans le pipeline dépend de l'état (occupé, libre) des étages. Ainsi, les données progressent dans le micropipeline tant qu'elles ne rencontrent pas de registres occupés et indépendamment des données suivantes [BOF10 ; Rez04 ; SF01]. Cependant, ce type de circuits révèle un certain nombre d'inconvénients. Par exemple, pour résoudre le problème des aléas, on doit ajouter des délais sur les signaux de contrôle. Les délais, étant fixés par une estimation du *timing* pire cas, assurent la mémorisation uniquement des données stables en sortie des parties combinatoires. Néanmoins, ces délais limitent la vitesse de propagation des données dans le pipeline, puisqu'une donnée stable, doit aussi respecter les délais fixes imposés sur les lignes de transfert. Afin d'améliorer les performances du pipeline, on utilise des registres et des opérateurs combinatoires capables de générer leur propre signal de fin de calcul sans avoir recours à des délais fixes (Figure 3.8). Dans ce cas, le signal  $C_d$  est transmis vers la partie combinatoire. Cette dernière initie à son tour le transfert des données vers l'étage suivant. Il en résulte une structure de micropipeline indépendante de la vitesse.

En outre, de part le fonctionnement type des FIFO des micropipelines asynchrones, on emploie ces structures afin de concevoir machines d'états asynchrones (*Asynchronous Finite State Machine : A-FSM*). Le micropipeline asynchrone assure, dans ce cas, l'enchaînement du fonctionnement lorsqu'un nouvel état est détecté. Les avantages de cette structure pour une machine d'état d'asynchrone ainsi que son implémentation sont revues dans la partie III.

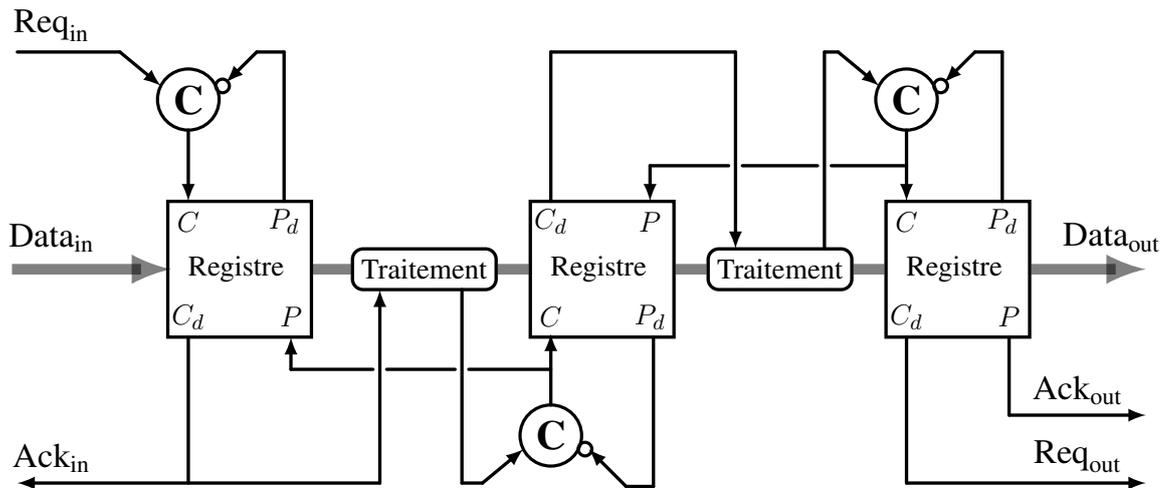


FIGURE 3.8 – Micropipeline indépendant de la vitesse

## 3.5 Caractéristiques des circuits asynchrones

### 3.5.1 Absence du signal d'horloge

L'avantage majeur des circuits asynchrones est l'absence de l'horloge et des problèmes associés à la manipulation de ce signal comme le problème de gigue des horloges (*Clock Skew*). En fait, la conception des circuits d'horloge est de plus en plus critique vu qu'ils limitent les performances des circuits synchrones. Cependant, pour les circuits asynchrones, l'absence de l'horloge globale implique l'implémentation des éléments de synchronisation et de contrôle locaux. Ils sont distribués dans l'ensemble du circuit et leur conception est ainsi beaucoup plus facile à maîtriser. En outre, en faisant appel à des technologies de conception modernes, on a le potentiel de concevoir des circuits plus complexes et plus rapides.

### 3.5.2 Calcul à temps minimal

Une des conséquences fondamentales de l'utilisation des circuits asynchrones est le fonctionnement type flots de données. Ce type de fonctionnement rend les circuits asynchrones très robustes vis-à-vis les variations des paramètres technologiques, de la température ou des tensions d'alimentation. En plus, le temps de traitement ou de transfert des données est fonctionnellement variable et borné. En effet, le temps dépend uniquement des données fournies et du chemin de données adapté. Ainsi, la fin du traitement des données est localement détectée et signalée aux étages suivants. Dans ce cas, les variations de la vitesse induites par les modifications des propriétés physiques n'influent pas le comportement fonctionnel. On peut alors conclure que le temps de traversée des

données est toujours optimal pour réaliser la fonction demandée, vis-à-vis du chemin emprunté par les données, la vitesse opérationnelle des dispositifs élémentaires et les conditions de fonctionnement de la technologie.

### 3.5.3 Consommation électrique

Même si le terme *asynchrone* ne signifie pas systématiquement la réduction de la consommation électrique, les caractéristiques du fonctionnement asynchrone améliorent les performances du circuit. En absence du signal de synchronisation global, on peut exclure la consommation du système de distribution des horloges de la consommation électrique totale du circuit. En plus, l'aspect événementiel de la logique asynchrone offre une mise en veille implicite pour les sous-blocs inactifs. En d'autre terme, seuls les sous-blocs ayant des données à traiter ou à transférer seront actifs.

Plusieurs approches de réduction de la consommation électrique ont été introduites comme la réduction de la tension d'alimentation et la modification de la fréquence d'horloge du système. Cependant, ces approches diminuent les performances du système ainsi que celles des sous-blocs. Pour un circuit synchrone, cela oblige notamment à modifier en conséquence le signal d'horloge. Pour un circuit asynchrone, le fonctionnement des sous-blocs n'étant pas lié à une contrainte de temps, la vitesse du traitement des données est imposée par la vitesse de fonctionnement. Ce qui fait que l'approche de diminution de la tension d'alimentation peut être facilement utilisée des systèmes asynchrones.

### 3.5.4 Modularité

Le contrôle local dans un circuit asynchrone et l'utilisation d'un protocole de communication commun entre les différents opérateurs permet une modularité aisée dans les systèmes asynchrones. En effet, on peut aisément construire une fonction complexe en connectant des modules préexistants via des connexions quasi "plug & play".

### 3.5.5 Migration technologique

Les circuits asynchrones se prêtent facilement aux migrations technologiques et les rendent plus « souples ». En effet, le comportement fonctionnel d'un circuit asynchrone est indépendant de la réalisation des cellules qui le constituent pourvu que le protocole de communication soit respecté. Ainsi, au niveau le plus bas, il est possible de modifier l'implémentation ou la technologie des cellules de base sans modifier la fonction. La vitesse de fonctionnement obtenue sera simplement la vitesse maximale permise par la nouvelle technologie étant donnée la structure du circuit [AII03].

## 3.6 Conclusion

Ce chapitre, on commence par une présentation brève des problèmes qui persistent lors de la conception des circuits synchrones. Ces limitations sont essentiellement liées à la synchronisation globale par un signal d'horloge. La logique asynchrone a été, par la suite, proposée comme étant une alternative à la conception synchrone. Dans ce cas, la synchronisation globale, fonctionnelle et temporelle, introduite par l'horloge est remplacée par une synchronisation locale atemporelle grâce à une signalisation bidirectionnelle qui garantit uniquement le traitement événementiel. Cette approche de conception est implémentée à travers des protocoles de communication asynchrone (*Handshaking Protocol*). Deux protocoles de communication ont été présentés : le protocole à données intégrées et le protocole double rails. Ensuite, plusieurs types des circuits asynchrones ont été exposés, classés suivant leur modèle de délais pour les portes logiques et pour les fils de connexion. Ce chapitre se termine par l'explication de l'effet de la logique asynchrone sur les performances des circuits conçus.

# 4

## ÉCHANTILLONNAGE NON-UNIFORME

### Sommaire

---

|         |  |           |
|---------|--|-----------|
| 4.1     | La théorie de l'échantillonnage . . . . .                  | <b>50</b> |
| 4.2     | Théorème de Shannon : L'échantillonnage Uniforme . . . . . | <b>51</b> |
| 4.2.1   | La Quantification . . . . .                                | <b>52</b> |
| 4.3     | Échantillonnage Non Uniforme . . . . .                     | <b>53</b> |
| 4.3.1   | Principe . . . . .   | <b>53</b> |
| 4.3.2   | Techniques de mise en œuvre . . . . .                      | <b>54</b> |
| 4.3.2.1 | Échantillonnage aléatoire additif . . . . .                | <b>54</b> |
| 4.3.2.2 | Échantillonnage uniforme à sauts aléatoires . . . . .      | <b>54</b> |
| 4.3.2.3 | Échantillonnage par croisement de signal . . . . .         | <b>56</b> |
| 4.3.2.4 | Échantillonnage par traversée de niveaux . . . . .         | <b>57</b> |
| 4.4     | Réduction du flot de données . . . . .                     | <b>58</b> |
| 4.4.1   | 1 Dimension . . . . .                                      | <b>59</b> |
| 4.4.2   | Extension à 2 Dimensions . . . . .                         | <b>60</b> |
| 4.5     | Conclusion . . . . .                                       | <b>61</b> |

---

Dans un monde, comme le nôtre, où la majorité des signaux sont de nature analogique, la numérisation de ces derniers demeure une étape indispensable. Cette numérisation permet d'interfacer le monde analogique et le monde numérique. De plus, la numérisation des signaux analogiques a aussi permis le passage du traitement analogique des signaux vers un traitement numérique plus avantageux en termes de vitesse, de consommation et de versatilité [IJ01].

La numérisation est composée de deux procédés élémentaires : l'échantillonnage et la quantification. La première étape de la numérisation, l'échantillonnage, consiste à transformer le signal analogique en capturant des valeurs à intervalles réguliers ou irréguliers. Alors que la quantification est le procédé qui permet d'adapter les valeurs capturées à des valeurs d'un ensemble discret.

Bien que dans ce chapitre, nous nous intéressons à l'échantillonnage, la quantification sera étudiée dans les chapitres suivants. La théorie de l'échantillonnage, depuis le théorème de Shannon [Sha49], a été largement abordée dans la littérature. Nous pouvons citer plusieurs exemples comme : [Beu70 ; Jer77 ; Mar87 ; BM92 ; Bil07 ; Uns00 ; Mar98 ; Fon99 ; Aes06].

Dans ce chapitre, nous décrivons brièvement la théorie d'échantillonnage classique

connue comme étant l'échantillonnage uniforme. Ensuite nous présentons l'échantillonnage non uniforme ainsi que quelques techniques de mise en œuvre figurant dans la littérature. Puisque le but de cette thèse consiste à effectuer une lecture efficace et intelligente du capteur d'images, nous insistons à travers ce chapitre sur l'importance de l'échantillonnage non uniforme dans notre travail.

## 4.1 La théorie de l'échantillonnage

L'échantillonnage d'un signal analogique est le passage du domaine analogique vers un domaine à valeurs discrètes.

Dans le domaine temporel, l'échantillonnage est le résultat de la multiplication du signal temporellement continu  $x(t)$  et la fonction d'échantillonnage  $s_F(t)$ . La dernière étant donnée sous une forme générale suivant l'équation (4.1) [Bil07]. Par la suite, le signal échantillonné  $x_s(t)$  est représenté suivant l'équation (4.2).

$$s_F(t) = \sum_{n=-\infty}^{+\infty} \delta(t - t_n) \quad (4.1)$$

$$x_s(t) = \sum_{n=-\infty}^{+\infty} x(t) \cdot \delta(t - t_n) \quad (4.2)$$

Sachant que :

$t_n$  : est l'instant d'échantillonnage  
 $\delta(t - t_n)$  : est la fonction de Dirac

Dans le domaine fréquentiel, on fait appel à la transformée de Fourier de la fonction d'échantillonnage  $s_F(t) : S_F(f)$ . Celle-ci peut être écrite suivant l'équation (4.3). Ainsi, l'échantillonnage est le produit de convolution entre le signal analogique et la fonction d'échantillonnage  $S_F(f)$ .

$$S_F(f) = \sum_{n=-\infty}^{+\infty} \exp^{-2\pi j f t_n} \quad (4.3)$$

Sachant que  $X(f)$  est le signal d'entrée analogique, le signal échantillonné peut être écrit sous la forme suivante (4.4)

$$X_s(f) = X(f) * S_F(f) \quad (4.4)$$

Dans les deux sections suivantes, on verra l'effet de la distribution uniforme ou non uniforme des instants d'échantillonnage  $t_n$ .

## 4.2 Théorème de Shannon : L'échantillonnage Uniforme

L'échantillonnage uniforme, aussi connu comme étant l'échantillonnage classique a été proposé en 1949 par Claude Elwood Shannon [Sha49]. C.E. Shannon a fourni le théorème d'échantillonnage uniforme de la conversion du signal analogique vers un ensemble de valeurs discrètes et qui est devenu plus tard la base du traitement numérique du signal.

L'échantillonnage uniforme est un procédé périodique, équidistant et théoriquement déterministe. Les instants d'échantillonnage sont uniformément distribués, ce qui fait que l'intervalle de temps entre deux échantillonnages consécutifs est fixe. Cet intervalle de temps est connu comme étant la période d'échantillonnage :  $T_s$ . L'ensemble des instants d'échantillonnage  $t_n$  est défini comme étant le multiple  $n$  (avec  $n \in \mathbb{N}$ ) de la période d'échantillonnage :

$$t_n = nT_s \quad (4.5)$$

En pratique, les instants d'échantillonnage uniforme  $T_s$  sont prédéfinis et déclenchés par un signal d'horloge.

En intégrant la relation (4.5) dans les équations vues précédemment, on obtient donc, les nouvelles relations qui définissent l'échantillonnage uniforme :

$$s_F(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) \quad (4.6)$$

$$x_s(t) = \sum_{n=-\infty}^{+\infty} x(t) \cdot \delta(t - nT_s) \quad (4.7)$$

$$S_F(f) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} \delta(f - nF_s) \quad (4.8)$$

On peut alors représenter le signal uniformément échantillonné sous la forme suivante :

$$X_s(f) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} X(f - nF_s) \quad (4.9)$$

La figure 4.1 est un simple exemple d'un signal aléatoire échantillonné avec une période  $T_s$ . Pour cet exemple, on a  $n$  points échantillonnés uniformément repartis. Sachant que  $F_s$  est la fréquence d'échantillonnage et  $f_{max}$  est la fréquence maximale du signal à échantillonner  $x(t)$ . Shannon a prouvé que  $x(t)$  peut être reconstruit en utilisant les coordonnées de la série des points espacés par  $1/(2f_{max})$ . Ce qui fait que la valeur de la fréquence d'échantillonnage  $F_s$  est soumise à la condition ci-dessous :

$$F_s \geq 2f_{max} \quad (4.10)$$

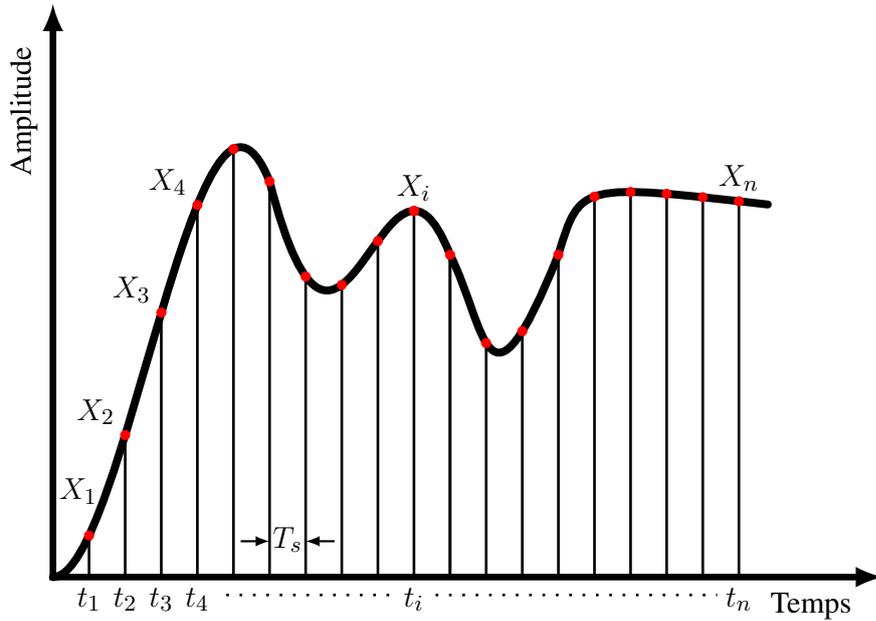


FIGURE 4.1 – Échantillonnage Uniforme d'un signal aléatoire

Par la suite, Nyquist a développé la relation établie entre  $F_s$  et  $f_{max}$  (équation (4.10)). Il a prouvé qu'on peut échantillonner, et reconstruire plus tard le signal initial sans repliement, en utilisant une fréquence d'échantillonnage exactement égale à la fréquence maximale du signal  $x(t)$ .

$$F_{Nyq} = 2f_{max} \quad (4.11)$$

En résumé, l'échantillonnage classique uniforme est une forme généralisée de l'échantillonnage et indépendant du signal analogique. Néanmoins, les conditions sur la fréquence d'échantillonnage et la redondance du flot de données résultant sont contraignantes. Certaines de ces contraintes peuvent être levées ou minimisées par l'exploitation d'un échantillonnage efficace et plus avantageux : l'échantillonnage non uniforme

### 4.2.1 La Quantification

La quantification est la deuxième étape de la conversion analogique numérique (CAN). Pour un CAN classique ayant une horloge idéale sans variation de phase (*jitter*), on assume que les instants d'échantillonnage sont connus. Dans ce cas, le bruit de quantification est la seule imprécision qui survient durant la conversion. Le rapport signal/bruit (*Signal-to-Noise Ratio* : *SNR*) est alors utilisé pour caractériser ce bruit. Le rapport signal/bruit est représenté suivant l'équation (4.12) :

$$SNR_{db} = 10 \log_{10} \left( \frac{P_{signal}}{P_{bruit}} \right) = 10 \log_{10} \left( \frac{A_{signal}}{A_{bruit}} \right)^2 \quad (4.12)$$

Avec  $P_{signal}$ ,  $A_{signal}$  et  $P_{bruit}$ ,  $A_{bruit}$  sont les puissance moyennes et les valeurs quadratiques moyennes (*Root Mean Square* : *RMS*) du signal et du bruit respectivement.

Pour un signal d'entrée sinusoïdal, on peut écrire le rapport signal/bruit sous la forme suivante (4.13) :

$$SNR_{db} = 1.76 + 6.02N \quad (4.13)$$

Avec  $N$  est le nombre de bits du convertisseur analogique numérique. On note que le rapport signal/bruit dépend uniquement du nombre de bits  $N$ .

## 4.3 Échantillonnage Non Uniforme

### 4.3.1 Principe

L'échantillonnage non uniforme consiste à échantillonner le signal analogique à des instants irréguliers et éventuellement aléatoires. L'échantillonnage non uniforme est utilisé dans le but de réduire la fréquence de traitement lorsque le flux de signal utile est irrégulier. Dans ce cas, seules les variations du signal analogiques peuvent être échantillonnées et ensuite traitées. Par conséquent, la fréquence d'échantillonnage est adaptée à la vitesse de variation du signal. De ce fait, ce type d'échantillonnage permet, d'une part, une réduction de la consommation électrique surtout si l'étude vise un signal de nature prédéfini et d'autre part, l'échantillonnage non uniforme entraîne une réduction des niveaux des signaux parasites liés au signal d'horloge.

Selon la méthode d'échantillonnage non uniforme implémentée, cet échantillonnage peut être soit déterministe, soit non déterministe. Les instants de l'échantillonnage non uniforme sont alors distribués suivant la technique de mise en œuvre utilisée. Plusieurs distributions des points et des instants d'échantillonnage, avec des caractéristiques diverses ont été proposées dans la littérature [BM92 ; Bil07 ; Mar98 ; Fon99 ; Aes06]. Par exemple, l'échantillonnage aléatoire additif (*Additive Random Sampling*) est un procédé non déterministe. On détaille par la suite quelques techniques de mise en œuvre de l'échantillonnage non uniforme.

La non uniformité peut être introduite dans le processus de l'échantillonnage d'une façon directe ou indirecte. Par la suite, on peut classer les types d'échantillonnage non uniforme en deux catégories : direct et indirect. Dans le but d'avoir une non uniformité directe, le signal d'entrée est échantillonné à des instants générés aléatoirement. C'est le cas pour l'échantillonnage aléatoire additif (*Additive Random Sampling*) [OPS48] ainsi que pour l'échantillonnage uniforme à sauts aléatoires (*Uniform Sampling with Random Skip*) [Fon99 ; Aes06].

Par contre, l'échantillonnage non uniforme de type indirect emploie une fonction de référence afin d'introduire la non uniformité dans le processus d'échantillonnage [DF01 ; VT09 ; Car+12]. On cite par exemple pour ce type, l'échantillonnage par croisement de

signal (*Signal Crossing Sampling*) ainsi que l'échantillonnage par traversée de niveaux (*Level Crossing Sampling*).

Les exemples d'échantillonnage non uniforme direct et indirect donnés précédemment sont détaillés dans les sections suivantes.

### 4.3.2 Techniques de mise en œuvre

Dans cette section on détaille quelques techniques de mise en œuvre de l'échantillonnage non uniforme direct et indirect.

#### 4.3.2.1 Échantillonnage aléatoire additif

L'échantillonnage aléatoire additif (*Additive Random Sampling : ARS*) a été proposé par *Shapiro* et *Silverman* dans le but de présenter un système d'échantillonnage non uniforme sans recouvrement spectral (*Alias-Free*) [SS60]. Les instants d'échantillonnage sont décrits par l'équation (4.14).

$$t_n = t_{n-1} + \tau_n \quad (4.14)$$

Sachant que  $t_n$  est l'instant d'échantillonnage présent et  $t_{n-1}$  est l'instant d'échantillonnage précédent. Tandis que  $\tau_n$  est un ensemble de variables aléatoires positives ayant une fonction de distribution  $p(\tau)$ , une variance  $\sigma^2$  et une moyenne  $\mu$  [OPS48]. Dans ce cas, la valeur moyenne de la fréquence d'échantillonnage est décrite par l'équation (4.15).

$$F_{moyenne} = 1/\mu \quad (4.15)$$

En fait, l'équation (4.14) impose la condition nécessaire sur les instants d'échantillonnage afin de garantir la reconstruction du signal échantillonné. L'échantillonnage aléatoire additif est illustré dans la figure 4.2. L'aspect aléatoire dans ce type d'échantillonnage est contrôlé par le ratio  $\sigma/\mu$ . Ce ratio est réglé en fonction de l'application visée de manière à ce que l'aspect aléatoire soit significatif afin d'accomplir un échantillonnage efficace. Par contre, l'augmentation de l'aspect aléatoire entraîne l'augmentation de l'erreur statique. Ce dernier point représente l'un des inconvénients de ce type d'échantillonnage [Bil07 ; OPS48].

#### 4.3.2.2 Échantillonnage uniforme à sauts aléatoires

L'échantillonnage uniforme à sauts aléatoires (*Uniform Sampling with Random Skip : USRS*) est un dérivé de l'échantillonnage uniforme classique. Ce type d'échantillonnage consiste, comme son nom l'indique, à exclure d'une façon aléatoire des échantillons. Cette forme particulière d'échantillonnage peut être le résultat d'une défaillance du système mais elle peut aussi être implémenté volontairement [Bil07 ; Aes06]. La figure

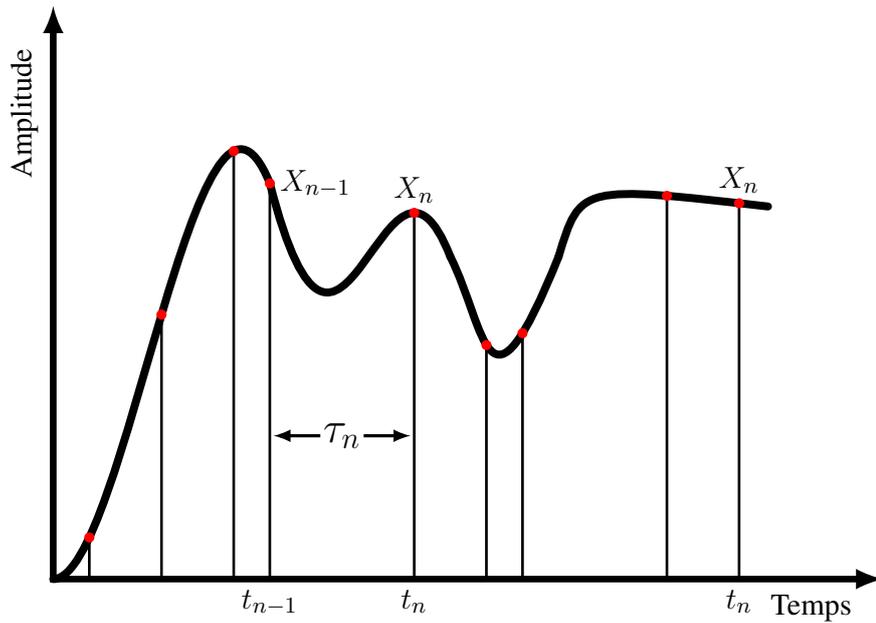


FIGURE 4.2 – Échantillonnage aléatoire additif

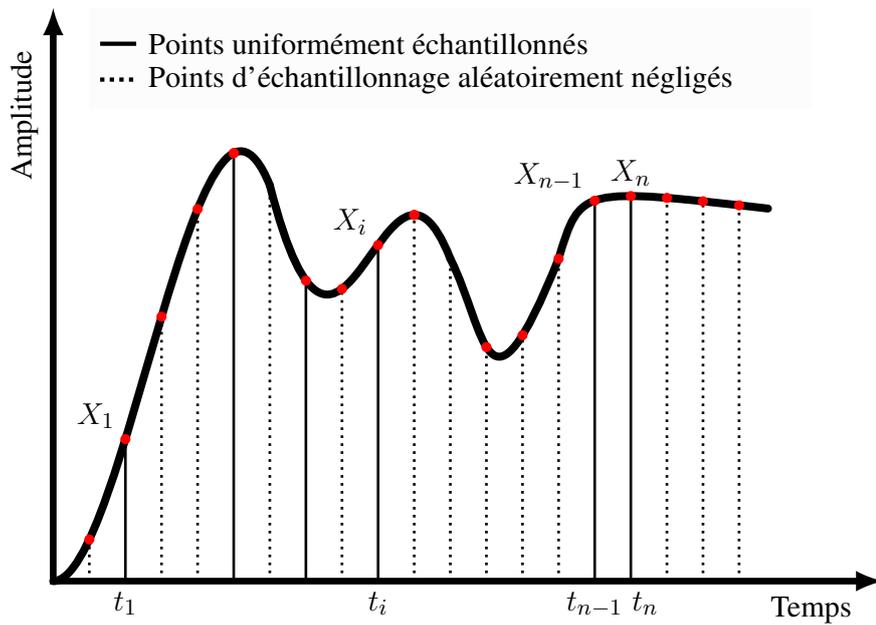


FIGURE 4.3 – Échantillonnage uniforme à sauts aléatoires

4.3 illustre l'échantillonnage uniforme à sauts aléatoires. Cet échantillonnage est basé sur un ensemble de points d'échantillonnage périodiques avec des points aléatoirement négligés. On suppose que les instants d'échantillonnage surviennent avec une probabi-

lité  $p_n$  (la probabilité pour qu'un échantillon soit négligé est de  $1 - p_n$ ). Afin d'assurer une reconstruction du signal échantillonné, les instants d'échantillonnage uniformes à sauts aléatoires sont soumis à la même condition que l'échantillonnage aléatoire additif (équation (4.14)).

#### 4.3.2.3 Échantillonnage par croisement de signal

Le concept d'échantillonnage par croisement de signal (*Signal Crossing Sampling : SCS*) se base sur la sélection d'un échantillon à chaque fois que le signal analogique croise le signal de référence  $e(t)$ . En utilisant ce type d'échantillonnage, l'efficacité de l'implémentation de ce dernier dépend du choix du signal de référence  $e(t)$ . Si le signal  $e(t)$  choisi est une fonction nulle, invariante dans le temps, on obtient un échantillonnage par passage à zéro (*zero crossing sampling*) [Car+12]. Un des signaux de référence

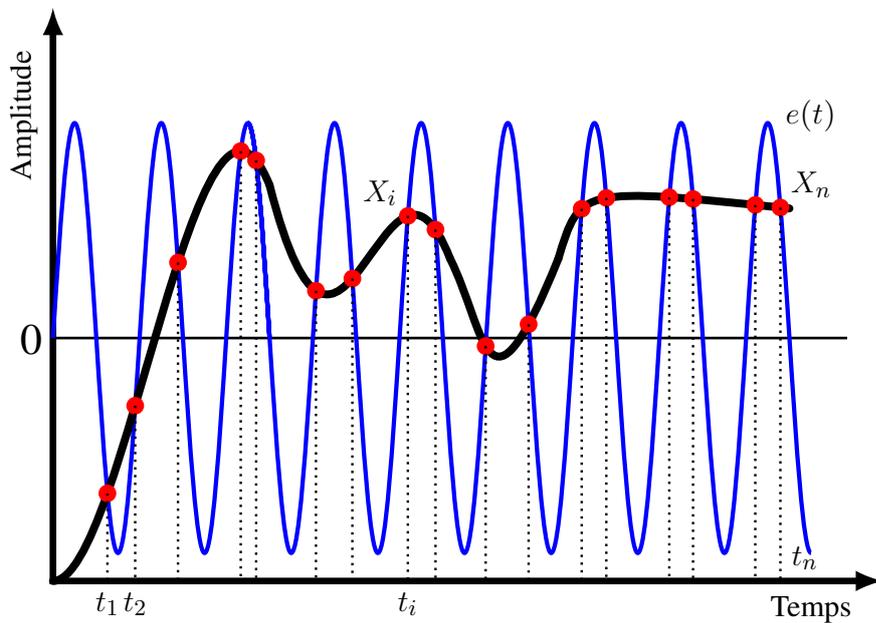


FIGURE 4.4 – Échantillonnage par croisement de signal : *Signal Crossing Sampling*

parmi les plus utilisés est le signal sinusoïdal. Le choix d'un signal sinusoïdal  $e(t)$  est justifié par le fait que ce signal est à bande étroite, stable et facilement généré [Bil07]. La figure 4.4 illustre l'échantillonnage par croisement de signal basé sur un signal de référence sinusoïdal. Pour ce type d'échantillonnage, l'intersection entre le signal analogique et le signal sinusoïdal à l'instant  $t_i$  implique la sélection d'un nouvel échantillon. Pour un signal  $e(t) = A_e \sin(2\pi f_e t)$ , la fréquence et l'amplitude de ce signal doivent respecter les deux conditions suivantes :

$$f_e \geq (f_{max}/2) \quad (4.16)$$

$$A_e \geq (\Delta_x/2) \quad (4.17)$$

$\Delta_x$  est la dynamique du signal d'entrée et  $f_{max}$  est la fréquence maximale du signal analogique. Le respect de ces conditions permet par la suite la reconstruction du signal échantillonné.

De plus, pour ce type d'échantillonnage, les instants de prise des échantillons  $\{t_1, t_2, \dots, t_i, \dots, t_n\}$  contiennent l'intégralité de l'information nécessaire pour la reconstruction du signal échantillonné. Les amplitudes des échantillons sont, par la suite, déduites à partir de la fonction du signal de référence  $e(t)$ . Dans ce cas, la reconstruction du signal échantillonné est simplifiée

#### 4.3.2.4 Échantillonnage par traversée de niveaux

L'échantillonnage par traversée ou par croisée de niveaux (*Level Crossing Sampling : LCS*) a été introduit en 1981 [MT81] dans le cadre de la compression de données. Il est également connu comme un échantillonnage basé sur événements [Le+05]. Ce concept d'échantillonnage consiste à définir un ensemble de niveaux sur l'axe d'amplitude.

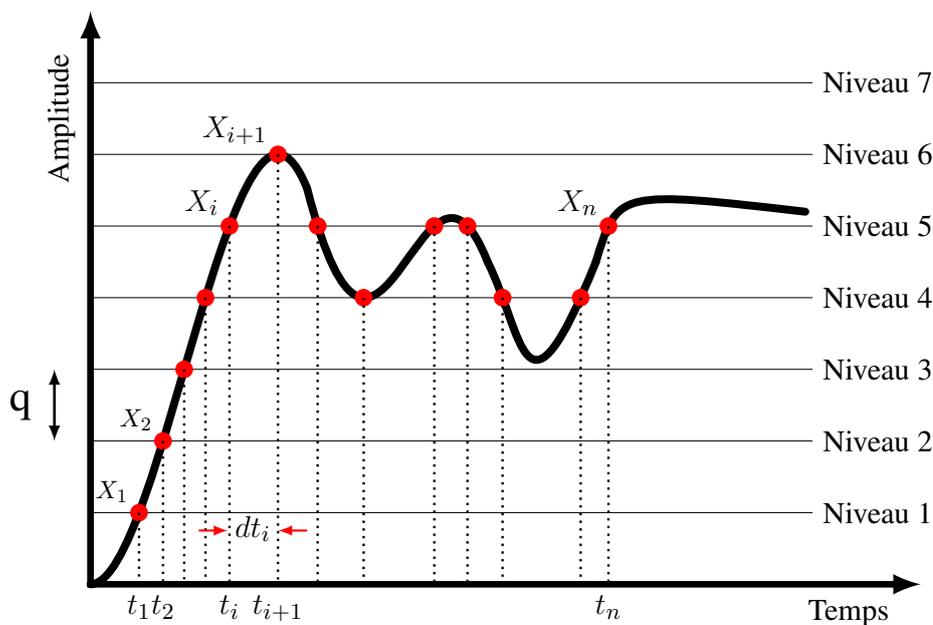


FIGURE 4.5 – Échantillonnage par traversée de niveaux : *Level Crossing Sampling*

Contrairement à d'autres types d'échantillonnage, l'échantillonnage par traversée de niveaux ne nécessite pas l'utilisation d'un signal d'horloge. Cependant un échantillon est prélevé lorsque le signal analogique traverse l'un des niveaux prédéfinis. Par conséquent, les instants d'échantillonnage ne sont plus uniformément espacés dans le temps. En outre, les variations du signal continu déterminent les instants d'échantillonnage  $t_i$  ainsi que le nombre d'échantillons pris en compte. En utilisant, le concept de l'échantillonnage par traversée de niveaux, les niveaux choisis pour les seuils doivent couvrir la

dynamique entière du signal analogique. La figure 4.5 montre une répartition éventuelle des niveaux de seuil. Dans le cas représenté, les niveaux d'échantillonnage sont équidistants et séparés par un quantum  $q$ . Pourtant, les niveaux de seuil peuvent être repartis suivant une distribution logarithmique ou suivant des distributions spécifiques et dépendantes du signal continu. En plus, la distribution de ces niveaux peuvent varier dans le temps [Bil07]. [Gar+13] présente une étude des différentes méthodes de positionnement des niveaux de discrétisation.

Avec ce type d'échantillonnage, chaque échantillon est un couple  $(X_i, t_i)$  représentant respectivement l'amplitude et l'instant d'échantillonnage. En effet, l'amplitude  $X_i$  dépend des niveaux d'échantillonnage prédéfinis, tandis que les instants d'échantillonnages sont définis par l'équation (4.18).

$$t_{i+1} = t_i + dt_i \quad (4.18)$$

Dans l'équation (4.18),  $dt_i$  est le temps écoulé entre l'instant d'échantillonnage courant  $t_{i+1}$  et l'instant d'échantillonnage précédent  $t_i$ . Ainsi, ce type d'échantillonnage exige un compteur à résolution finie avec un pas de comptage  $P_c$ , pour mesurer les intervalles de temps entre deux échantillons prélevés.  $dt_i$  peut alors s'écrire sous la forme de multiple de  $P_c$  (4.19).

$$dt_i = \alpha P_c \quad (\alpha \in \mathbb{N}^*) \quad (4.19)$$

Par conséquent, la période du compteur  $T_c$  utilisée détermine la précision et la résolution de l'échantillonnage. Contrairement à d'autres types d'échantillonnage non uniforme, l'échantillonnage par traversée de niveau ne fournit aucune connaissance anticipée des instants d'échantillonnage.

Le tableau 4.1 présente un bilan de comparaison entre les différentes techniques de mise en œuvre de l'échantillonnage non uniforme.

Dans la cadre d'un imageur à échantillonnage non uniforme, on vise une compression de données lors de l'échantillonnage du signal analogique. En d'autres termes, on cherche à compresser d'une manière efficace l'information provenant des pixels. Afin d'atteindre ce but, on adopte l'échantillonnage non uniforme par traversée des niveaux. En fait, cet échantillonnage, initialement proposé par [Gen+91], est essentiellement dédié à la compression de données.

## 4.4 Réduction du flot de données

L'adoption de l'échantillonnage non uniforme par traversée de niveaux, nous accorde la prise en compte des caractéristiques du signal d'entrée analogique lors de l'échantillonnage. Ainsi, en adaptant la fréquence d'échantillonnage aux variations du signal analogique et en utilisant une distribution efficace des niveaux de seuil, on obtient un

**TABLE 4.1** – Comparaison entre les techniques d'échantillonnage non-uniforme

| Caractéristiques            | ARS                                      | USRS                  | SCS                                | LCS                            |
|-----------------------------|--|-----------------------|------------------------------------|--------------------------------|
| Déclencheur de conversion   | Aléatoire                                | Aléatoire             | Traversée de signal                | Traversée de niveaux           |
| Fréquence d'échantillonnage | $F_{moyenne} = 1/\mu$<br>(équation 4.15) | $\geq f_{Nyquist}$    | Dépend de $f_e$<br>(équation 4.16) | Dépend de l'activité du signal |
| Signal de référence         | -  | -                     | Signal Continu                     | Niveaux discrets               |
| Fréquence du timer          | Dépend de $\tau_n$                       | Dépend de $(1 - p_n)$ | Dépend de $f_e$                    | Dépend des niveaux discrets    |
| Adaptation                  | -  | Aléatoire             | -                                  | Adapté au signal échantillonné |

taux d'échantillonnage non redondant et adapté aux variations du signal [LeP15]. L'activité et la consommation électrique de l'ensemble de la chaîne de traitement du signal est par la suite réduite.

Cette section est dédiée à la présentation d'une des applications de l'échantillonnage par traversée de niveaux en une dimension : le convertisseur analogique numérique. En outre, on présente l'extension de ce type d'échantillonnage en dimension deux, en l'intégrant dans la chaîne de traitement d'un capteur d'images CMOS [FDS15].

#### 4.4.1 1 Dimension

Les convertisseurs analogique numérique asynchrones [ATS09 ; AMA06 ; All+05 ; All03] sont considérés comme étant des alternatives efficaces aux convertisseurs classiques. L'échantillonnage non uniforme par traversée de niveaux est alors utilisé dans le but d'optimiser la chaîne de lecture de ce capteur [Roa+14]. En plus, une distribution adaptée des niveaux d'échantillonnage conduit à optimiser le processus de reconnaissance de forme pour les signaux issus de différents capteurs [LeP15 ; Qai09].

Le concept du convertisseur analogique-numérique asynchrone (*A-ADC*) se base sur l'échantillonnage par traversée de niveaux [All+05 ; All03]. En d'autres termes, la conversion est déclenchée par la traversée du signal d'entrée d'un des niveaux de seuil prédéfinis. Dans ce cas, l'amplitude est une valeur exacte et dépend des niveaux d'échantillonnage. Par contre, le temps écoulé entre deux échantillons est quantifié. Par la suite, la valeur

de  $dt_i$  contient une erreur de quantification  $\delta t$ . On présente l'erreur  $\delta t$  sous la forme d'une erreur d'amplitude  $\delta V$  :

$$\delta V = \frac{dV_{in}}{dt} \delta t \quad (4.20)$$

En considérant,  $\delta t$  et  $\frac{dV_{in}}{dt}$  comme étant deux processus aléatoires et indépendants, on peut alors écrire le bruit de quantification suivant l'équation (4.21) :

$$P(\delta V) = P\left(\frac{dV_{in}}{dt}\right) \cdot P(\delta t) \quad (4.21)$$

Pour une variable  $\delta t$  aléatoire et uniformément distribuée sur  $[0, T_c]$ , on réécrit  $P(\delta t)$  sous la forme suivante :

$$P(\delta t) = \frac{T_c^2}{3} \quad (4.22)$$

En utilisant l'équation du rapport signal/bruit (4.12) et les deux équations (4.21) et (4.22), on obtient l'équation du rapport signal/bruit pour un échantillonnage non-uniforme par traversée de niveaux [All+05 ; All03] :

$$SNR_{db} = 10 \log_{10} \left( \frac{3P(V_{in})}{P\left(\frac{dV_{in}}{dt}\right)} \right) + 20 \log_{10} \left( \frac{1}{T_c} \right) \quad (4.23)$$

En appliquant l'équation (4.23) pour un signal d'entrée sinusoïdal ayant une fréquence  $f_e$ , le rapport signal/bruit devient :

$$SNR_{db} = -11.2 + 20 \log_{10}(f_e T_c) \quad (4.24)$$

Dans le cas de l'échantillonnage par traversée de niveaux, le rapport signal/bruit ( $SNR$ ) dépend uniquement de la période du compteur. En d'autres termes, le nombre de niveaux de quantification n'affecte pas le rapport signal/bruit. Par conséquent, le  $SNR$  pour une conversion non-uniforme peut être réglé en changeant la période du compteur  $T_c$ .

Le tableau 4.2 résume les différentes caractéristiques de l'échantillonnage uniforme et de l'échantillonnage non uniforme par traversée de niveaux.

#### 4.4.2 Extension à 2 Dimensions

Les avantages de l'utilisation de l'approche événementielle persistent aussi dans le cas des capteurs d'images. En effet, l'activité de la scène peut déclencher la chaîne de lecture et de traitement du capteur d'images. Le fonctionnement événementiel pour un capteur d'images doit être premièrement intégré au niveau de l'élément photosensible :

**TABLE 4.2** – Caractéristiques de l'échantillonnage uniforme et de l'échantillonnage non uniforme par traversée de niveaux

| Caractéristiques          | Échantillonnage uniforme | Échantillonnage par traversée de niveaux |
|---------------------------|--------------------------|--|
| Déclencheur de conversion | Horloge                  | Traversée d'un niveau                    |
| Amplitude                 | Quantifiée               | Valeur exacte                            |
| Temps                     | Valeur exacte            | Quantifié                                |
| Dépendance de SNR         | Nombre de bits           | Période du compteur                      |
| Sortie                    | Amplitude                | Amplitude et Temps                       |

le pixel. Dans ce cas, l'échantillonnage non uniforme par traversée de niveaux est employé afin de détecter l'activité de la scène. De plus, le fait d'utiliser l'échantillonnage non uniforme conduit au remplacement de la conversion analogique numérique classique par une conversion *time-to-digital*. En d'autres termes, l'information du pixel est codé en temps. Ainsi, le fait de remplacer le convertisseur analogique numérique est avantageux vu que ce composant électrique est le plus consommant de tout le système de vision.

Un capteur d'images événementiel, autrement appelé 'asynchrone', est aussi caractérisé par un flot de données réduits et par la suite par un système de lecture et de traitement d'images à activité réduite. En plus, ce type d'échantillonnage non uniforme nous permet d'exécuter des traitements d'images spatio-temporels au niveau du pixel et même au niveau du système de lecture du capteur d'images [PMW10; DFS14]. L'originalité du capteurs d'images asynchrones permet potentiellement une réduction considérable au niveau de la consommation électrique.

L'importance ainsi que les avantages de ces capteurs d'images non-conventionnels asynchrones sont présentés en détails dans le chapitre 2. En outre, on détaille dans les parties II et III le capteur asynchrone proposé durant cette thèse ainsi que son principe de fonctionnement, ses caractéristiques et ses éventuelles applications.

## 4.5 Conclusion

Dans ce chapitre, on a présenté les deux étapes fondamentales de la conversion analogique numérique en insistant sur le processus de l'échantillonnage. On a ainsi développé la théorie de l'échantillonnage en introduisant le théorème de Shannon et l'échantillonnage classique uniforme. Ce type d'échantillonnage périodique gouverné par une horloge et indépendant du signal échantillonné produit un flot de données redondant et

mène à une activité de traitement de signal inutilement consommatrice. Pour cela, on a présenté l'alternative de l'échantillonnage classique : l'échantillonnage non uniforme. On a détaillé le principe et quelques techniques de mises en œuvre de ce type d'échantillonnage non conventionnel en insistant sur l'échantillonnage par traversée de niveaux. En adoptant cette approche, on obtient un taux d'échantillonnage qui s'adapte aux variations du signal échantillonné ce qui implique un flot de données réduits et limité à des informations pertinentes.

**Deuxième partie**

**Fonctionnement Général**



# 5

## VERS UN IMAGEUR ÉVÉNEMENTIEL ASYNCHRONE : APPROCHES ET CONTRIBUTIONS

### Sommaire

---

|       |  |    |
|-------|--|----|
| 5.1   | Conception des circuits numériques en logique asynchrone . . . . . | 66 |
| 5.2   | Protocole de lecture du capteur d'images . . . . .                 | 67 |
| 5.3   | Suppression des redondances spatiales et temporelles . . . . .     | 67 |
| 5.3.1 | Redondances Temporelles . . . . .                                  | 69 |
| 5.3.2 | Redondances Spatiales . . . . .                                    | 70 |
| 5.4   | Conclusion . . . . .   | 70 |

---

On débute la deuxième partie de ce manuscrit, dédiée à l'explication du fonctionnement global de notre imageur, en présentant, à travers ce chapitre, les principales contributions amenées par les travaux de cette thèse dans le domaine des imageurs événementiels. En fait, le chapitre 2 a été consacré à l'état de l'art de imageurs conventionnels et événementiels, tout en désignant les imageurs événementiels existants dans la littérature lors de l'exposition du principe des imageurs asynchrones événementiels ainsi que les différents protocoles de communication et de codage. Dans le but de mettre en évidence les différences entre le capteur d'images événementiel asynchrone proposé au cours de cette thèse et les capteurs d'images événementiels existants dans la littérature, on effectue une comparaison entre les principales caractéristiques des différents imageurs asynchrones. Ce récapitulatif servira de point de référence et d'analogie entre les approches existantes dans la littérature et celles élaborées pendant la thèse.

Le capteur d'images événementiel est originalement présenté comme étant le capteur d'images bénéficiant des avantages du fonctionnement et des protocoles de communication bio-inspirés. De plus, ces capteurs d'images sont considérés comme étant des alternatives aux imageurs conventionnels à la vue de leur avantages remarquables : vitesse de lecture, dynamique de fonctionnement, flux de données, consommation électrique. Cependant, les imageurs événementiels manifestent en premier lieu des inconvénients comme des erreurs de timing et de priorité en extrayant l'information lumineuse des pixels. En deuxième lieu, les imageurs événementiels existants dans la littérature souffrent d'un manque de traitements d'images avancés intégrés dans la chaîne de lecture susceptibles de s'adapter et bénéficier de l'aspect événementiel de ces imageurs.

Afin de montrer les travaux présentés et les approches adoptées au cours de de cette thèse, on dresse, à travers les sections suivantes, un bilan des principales caractéristiques des capteurs d'images événementiels tout en mentionnant les contributions apportées par les travaux de cette thèse.

## 5.1 Conception des circuits numériques en logique asynchrone

Le fonctionnement à basse consommation électrique des capteurs d'images asynchrones est assuré, d'une part, par le remplacement de la conversion analogique numérique par une conversion tension temps, et d'autre part, par la réduction du flux de données en sortie du capteur d'images. De plus, afin de réduire d'avantage la consommation électrique, les concepteurs ont introduit les circuits numériques asynchrones basés sur le principe des protocoles *handshake*. En fait, les circuits numériques asynchrones sont caractérisés par leur faible activité et optimisée ce qui implique une faible (ou au moins) consommation électrique ainsi qu'une vitesse de fonctionnement élevée par rapport aux circuits numériques synchrones [BOF10 ; SF01]. En outre, les circuits asynchrones sont particulièrement bien adaptés à la gestion d'un flux de données événementiel. En effet, un circuit asynchrone réagissant à des événements ou des données utilise le protocole de communication de type *handshake* afin de réaliser la synchronisation, la communication et le séquençage de données nécessaires entre ses sous blocs. Ainsi, le protocole de communication bio-inspiré *AER (Address Event Representation)* employé entre le système de lecture du capteur et les pixels événementiels a été implémenté grâce à des circuits asynchrones (par exemple l'arbitre) qui gère les requêtes de lecture des pixels à travers le protocole de communication *handshake*.

Dans le but de bénéficier des avantages des circuits asynchrones, on présente dans cette thèse, un capteur d'images événementiel et complètement asynchrone. En d'autres termes, le circuit numérique de lecture ainsi que les traitements d'images intégrés au sein de la chaîne de lecture sont conçus en logique totalement asynchrone. Par conséquent, le capteur d'images présenté est un capteur d'images événementiel (la lecture et le traitement des pixels sont uniquement exécutés à l'occurrence d'un nouvel événement) et totalement asynchrone (les circuits numériques sont entièrement conçus en logique asynchrone). De plus, les blocs de traitements d'images intégrés dans les systèmes de vision sont souvent les sources de consommation électrique importantes vu qu'ils implémentent des algorithmes complexes et répétitifs. De ce fait, le fonctionnement totalement asynchrone de notre capteur d'images implique une réduction de l'activité du circuit capteur d'images, et par la suite de la consommation électrique de l'ensemble du système de vision.

**Contributions :** Le capteur d'images événementiel proposé est conçu avec une logique totalement asynchrone. Notre imageur bénéficie du fonctionnement asynchrone tout au long de la chaîne de sa lecture.

## 5.2 Protocole de lecture du capteur d'images

Les capteurs d'images événementiels dans la littérature implémentent le protocole de lecture *AER* par l'intermédiaire d'un circuit d'arbitrage non déterministe. En fait, l'arbitre séquentialise les requêtes de lecture asynchrones. En effet, il décide l'ordre de réception et de traitement des requêtes de lecture des pixels. En fonction du non déterminisme de l'arbitre, une même scène au niveau du capteur d'images génère potentiellement des flots différents de données en sortie. Cependant, en plus des problèmes de priorité et des erreurs de timing (voir section 2.7.3), ce circuit non-déterministe est difficile à optimiser et non synthétisable.

Afin de surmonter les limitations des circuits d'arbitrage, on propose une nouvelle stratégie de lecture, basée sur le principe *AER* consistant à paralléliser la réception des requêtes de lectures des pixels. Cette méthode de lecture demeure une lecture sur événements mais contrairement à une lecture arbitrée, elle adopte un fonctionnement déterministe. Cette méthode de lecture est implémentée grâce à un circuit numérique simple : l'encodeur. Ce bloc collecte toutes les adresses des pixels actifs simultanément et enregistre leurs instants d'occurrence. Le calcul du temps d'intégration de chaque pixel se fait ultérieurement ce qui permet une lecture rapide et non différée. De plus, grâce à cette méthode spécifique de lecture, on réduit la redondance spatiale dans le flux de données en sortie du capteur de manière importante (voir chapitre 7).

**Contributions :** Une nouvelle méthode de lecture, basée sur le protocole *AER*, est capable de paralléliser la réception des requêtes de lecture. Cette approche est implémentée à travers un circuit déterministe et relativement simple à synthétiser.

## 5.3 Suppression des redondances spatiales et temporelles

Dans un souci de rendre le compromis entre consommation électrique, vitesse de lecture et résolution spatiale du capteur d'images moins contraignant, les concepteurs d'imageur cherchent à réduire le flot de données en sortie du capteur d'images. Cette approche peut être appliquée en réduisant voire supprimant les redondances d'information dans le flot de données. Il existe deux types de redondances dans un flot de données en sortie d'un capteur d'images : les redondances spatiales et les redondances temporelles. Le taux de redondances spatiales dans une scène capturée est défini par le nombre de pixels ayant la même information lumineuse et par la suite la même courant photogénéré. Dans un système de vision conventionnel, ces pixels sont lus séquentiellement et indépendamment de leur valeur. En revanche, adopter une lecture unique pour les pixels ayant la même information lumineuse, en d'autres termes une lecture par valeur, diminue le flot de données en sortie du capteur. En prenant, l'exemple de la figure 5.1, on

peut visuellement déduire la présence de redondances spatiales dans cette scène statique. On a besoin d'effectuer  $M*N$  lectures, avec  $M$  et  $N$  étant le nombre de pixels par lignes et par colonne respectivement. Cependant, la suppression des redondances spatiales lors de la lecture du capteur associé à cette image, en d'autres termes, en limitant la lecture de l'image à une lecture par valeur, sachant que cette image possède 256 niveaux de gris, diminue le nombre de lectures nécessaires à 1% du nombre de lectures initiales ( $M*N$  lectures).

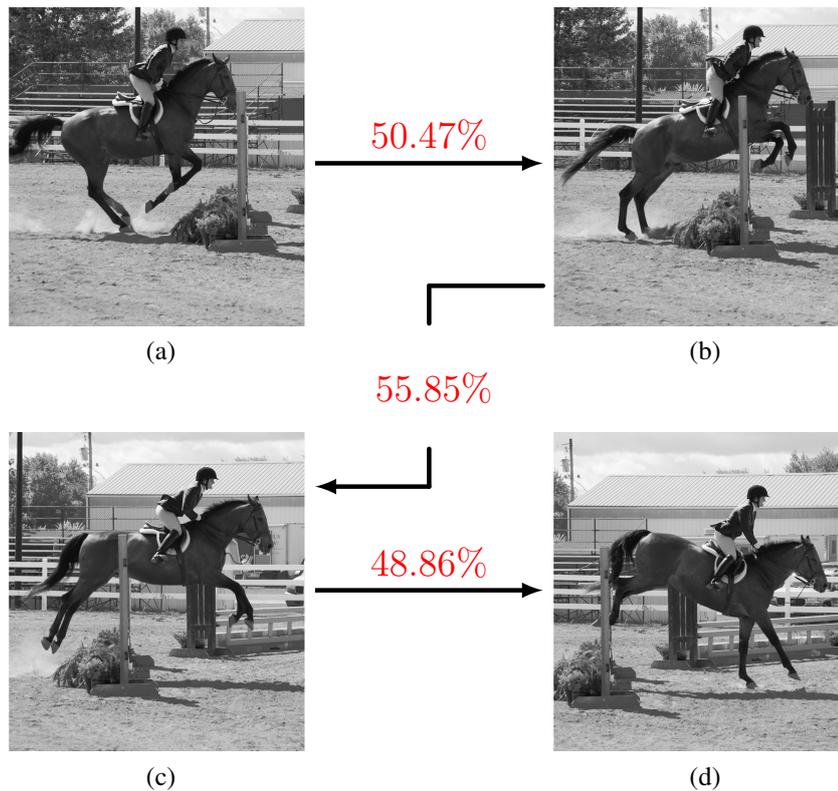


**FIGURE 5.1** – Exemple d'une image spatialement redondante

Ensuite, en examinant une séquence de vidéo, les parties statiques de la scène filmée représentent la redondance temporelle. En fait, le taux de redondance temporelle est défini par le nombre de pixels ayant une valeur invariante dans le temps pour des trames consécutives. De ce fait, limiter la lecture aux pixels ayant une nouvelle information lumineuse, diminue considérablement le flot de données d'une séquence vidéo. La figure 5.2 comporte 4 trames consécutives extraites d'une séquence vidéo. Selon l'exemple présenté, en moyenne, 51% des lectures effectuées pendant chaque trame sont temporellement redondantes.

En fait, la réduction des redondances spatiales et temporelles est une des pistes les plus examinées et développées afin de réduire le flux de données en sortie des capteurs d'images. Cette approche est exploitée pour les capteurs d'images conventionnels standards (voir section 2.6.1) ainsi que pour les capteurs événementiels. Néanmoins, l'aspect événementiel ainsi que l'adoption de la conversion tension-temps permettent aux capteurs événementiels d'être encore plus efficaces pour ce type de traitement d'images. De plus, la réduction des redondances dans un flux de données en sortie des capteurs d'images événementiels se révèle comme étant, dans certains cas, cohérente à l'aspect événementiel de ces capteurs d'images.

En principe les redondances spatiales et temporelles dans une scène capturée indiquent



**FIGURE 5.2** – Exemple d’une séquence d’images successives temporellement redondantes. Les pourcentages (en rouge) marqués entre les images indiquent le taux de redondance temporelle entre la trame précédente et la trame suivante

la présence d’une information répétitive. Ceci dit, l’extraction de l’information lumineuse exhaustive du capteur d’images implique une lecture coûteuse en termes de consommation électrique.

### 5.3.1 Redondances Temporelles

Afin de réduire le flot de données en sortie du capteur d’images, [PMW08] a introduit un capteur d’images événementiel capable de supprimer les redondances temporelles au niveau du pixel. En fait, le composant photosensible du pixel proposé mesurant l’intensité lumineuse a été originalement conçu dans le cadre des travaux sur la détection du contraste dans l’image [Lic08 ; Pos+07 ; Lic06a]. En fait, le pixel proposé dans [PMW11 ; PMW10 ; PMW08] est capable, grâce à son détecteur de variation de luminosité, de sauvegarder la dernière information lumineuse afin de la comparer à la nouvelle information occurrente. Si les deux valeurs sont identiques, la requête de lecture n’est pas envoyée et par la suite le pixel n’est pas lu. Dans ce cas, la redondance temporelle

est réduite, vu que le pixel n'est lu que lorsqu'il contient une nouvelle information.

L'inconvénient de ce type de circuit réside dans la grande surface du pixel vu que chaque pixel possède deux convertisseurs-analogique numérique 1 bit ainsi que deux photodiodes dont une seule est considérée comme étant effectivement photosensible et servant à mesurer la lumière incidente. De plus, ce capteur d'images événementiel souffre d'une lecture arbitrée non déterministe.

### 5.3.2 Redondances Spatiales

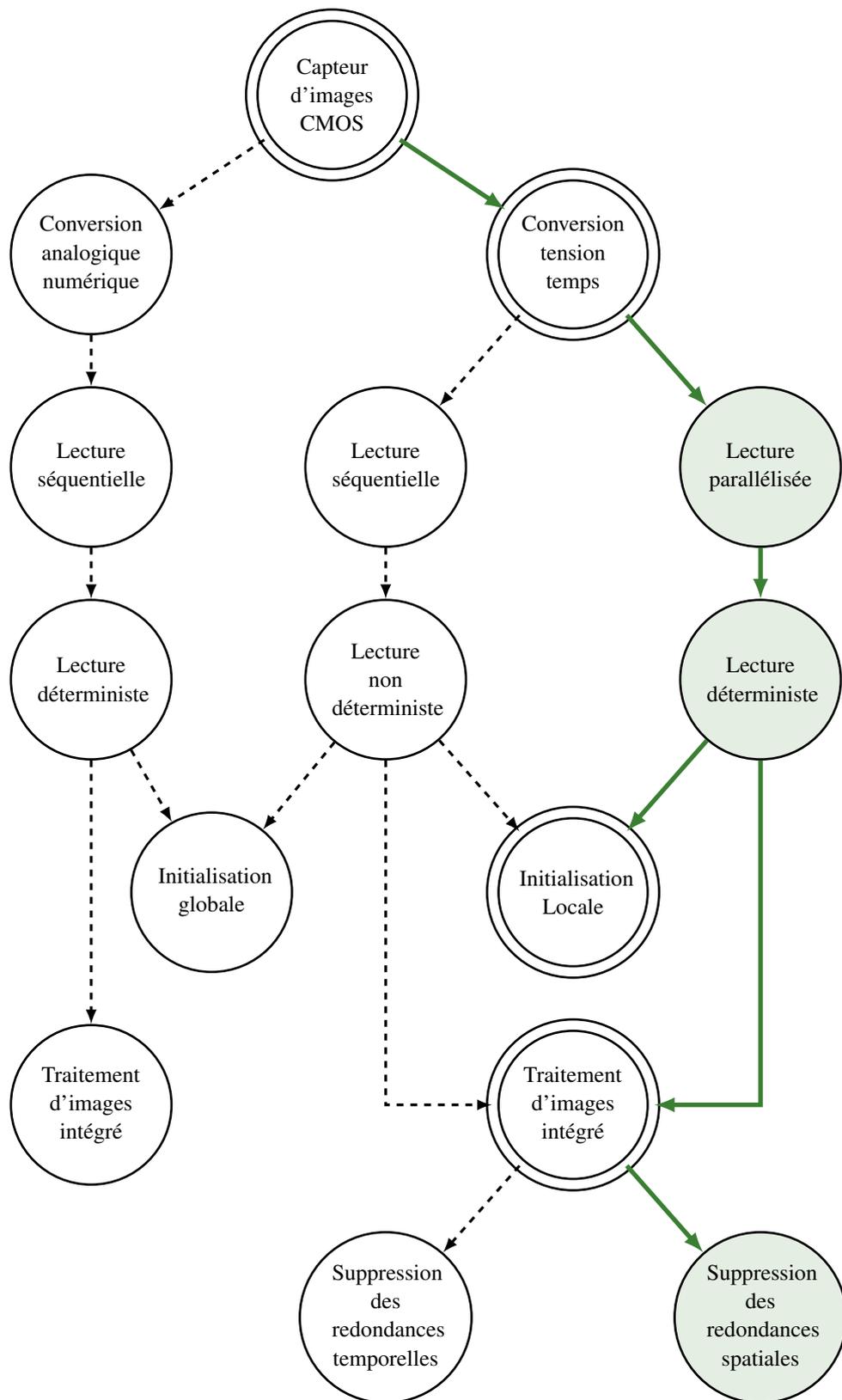
Dans le cadre de cette thèse, on s'intéresse à la réduction de la redondance spatiale. L'avantage de ce traitement d'images réside dans le fait, que les redondances spatiales existent, contrairement aux redondances temporelles, lors de la capture d'une seule image mais aussi dans les flux vidéos (elles existent dans chaque trame lue). Ainsi, le capteur d'images événementiel et asynchrone proposé bénéficie de la conversion tension-temps adaptée au pixel et du protocole de lecture spécifique de notre système de vision (voir section 5.2), afin de réduire, voire de supprimer les redondances spatiales. En fait, dans le but de supprimer les redondances spatiales, on regroupe les pixels par instant de requête de luminosité et on limite par la suite l'extraction exhaustive de l'information lumineuse. En principe, les requêtes de lecture envoyées dans la même fenêtre de temps n'impliquent pas nécessairement que les pixels associés soient soumis à une même luminosité. Si tous les pixels sont initialisés (*Reset*) puis actifs au même instant ils seront rattachés à la même valeur de luminosité. Cependant, pour les pixels initialisés à des instants différents, un bloc arithmétique a été ajouté afin de calculer leurs temps d'intégration propres. En adoptant cette méthode, les pixels sont groupés par les instants de requête, et la lecture des pixels est effectuée par valeur de pixel au lieu de lire chaque pixel individuellement. Par la suite, les redondances spatiales dans le flux de données en sortie du capteur d'images sont réduites considérablement comme nos résultats le montrent. Cette approche ainsi que les résultats obtenus sont détaillés dans les chapitres qui suivent.

**Contributions** : Le système de vision proposé réduit le flux de données en sortie du capteur d'images en réduisant, voire en supprimant, les redondances spatiales tout en profitant de l'aspect événementiel du capteur ainsi que de son protocole de lecture déterministe.

## 5.4 Conclusion

Dans ce chapitre, on a présenté les principales approches adoptées dans le cadre de cette thèse. La figure 5.3 illustre un organigramme récapitulant les contributions proposées au cours de cette thèse. Au cours de ce chapitre, on a effectué une comparaison

rapide entre le travail de la thèse et des travaux similaires déjà existants dans la littérature. Ce chapitre sert comme une introduction vers les chapitres suivants qui expliquent en détails les approches adoptées.



**FIGURE 5.3** – Récapitulatif de contributions proposées au cours de cette thèse. Les doubles cercles représentent les approches adoptées en commun avec l'état de l'art tandis que les cercles en couleur verte représentent les contributions apportées durant cette thèse

# 6

## FONCTIONNEMENT DE L'IMAGEUR ÉVÉNEMENTIEL ASYNCHRONE

### Sommaire

---

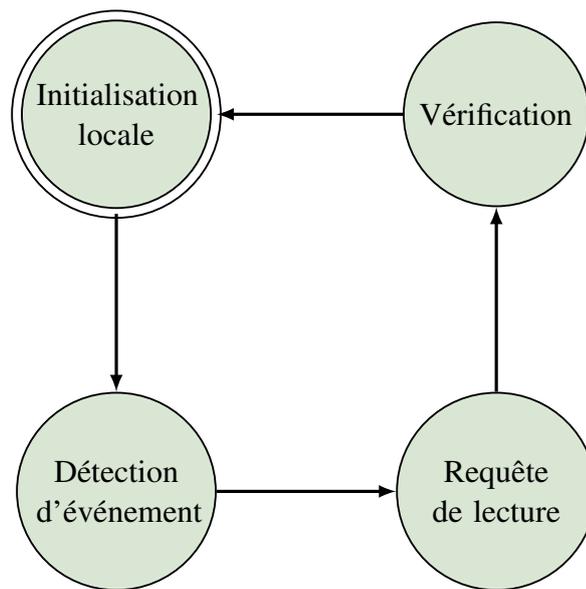
|         |   |           |
|---------|---|-----------|
| 6.1     | Chaîne de lecture de l'imageur événementiel asynchrone . . . . .      | <b>73</b> |
| 6.2     | Système de lecture asynchrone . . . . .                               | <b>76</b> |
| 6.2.1   | Architecture globale du système de lecture . . . . .                  | 76        |
| 6.2.2   | Bloc d'horodatage (Time Stamping Bloc) . . . . .                      | 78        |
| 6.2.3   | Architecture du bloc de traitement des requêtes de lecture            | 79        |
| 6.2.3.1 | Encodeur - La phase de réception des requêtes<br>de lecture . . . . . | 79        |
| 6.2.3.2 | Mémoire tampon : FIFO . . . . .                                       | 80        |
| 6.2.3.3 | Bloc de vérification - La phase de vérification .                     | 80        |
| 6.3     | Réduction des redondances spatiales . . . . .                         | <b>82</b> |
| 6.4     | Le pixel événementiel TFS . . . . .                                   | <b>84</b> |
| 6.4.1   | Fonctionnement du pixel événementiel . . . . .                        | 85        |
| 6.4.2   | Niveau de détection des événements . . . . .                          | 85        |
| 6.4.3   | Initialisation locale . . . . .                                       | 87        |
| 6.4.4   | Temps mort du pixel . . . . .   | 87        |
| 6.5     | Conclusion . . . . .  | <b>88</b> |

---

Cette partie du manuscrit introduit le fonctionnement général de notre capteur d'images événementiel asynchrone ainsi que les résultats de simulations haut niveau. Le deuxième chapitre de cette partie est dédié à l'exposition et l'explication du fonctionnement général du système de lecture ainsi que celui du pixel événementiel. L'architecture globale du système de lecture asynchrone est aussi présentée au cours de ce chapitre.

### 6.1 Chaîne de lecture de l'imageur événementiel asynchrone

Cette section présente les principales fonctionnalités intégrées dans la chaîne de lecture de notre imageur événementiel asynchrone. Il est important de rappeler que la lecture de notre système est événementielle et parallélisée, contrairement à lecture séquentielle des imageurs conventionnels. De plus, le fonctionnement de notre système de vision ne suit pas la notion de trame par seconde. Cette particularité est due au fonctionnement événementiel et autonome du pixel. Les phases principales du fonctionnement



**FIGURE 6.1** – Les phases principales de fonctionnement du pixel événementiel

de notre pixel événementiel sont présentées par la figure 6.1. Après l'initialisation du système de lecture et du pixel, la séquence de lecture est amorcée par le pixel qui, suite à une détection d'événement, envoie une requête de lecture. Cette dernière déclenche la séquence de lecture du pixel. Une vision synthétique de la chaîne de lecture du pixel est présentée par la figure 6.2. A cet état, le système de lecture est en phase de réception des requêtes de lecture, débutant une nouvelle itération de lecture. Au début de l'itération, le système d'une part, enregistre l'instant d'occurrence de la première requête reçue et d'autre part, il demeure capable de recevoir simultanément les demandes de lecture de tous les pixels actifs. Cette lecture parallélisée nous permet également d'effectuer une suppression des redondances spatiales existantes dans le flot de données en sortie du capteur d'images. A la fin de la phase de réduction des redondances spatiales, le système de lecture et le pixel passent en phase de vérification. Cette phase est indispensable dans le cas de la lecture simultanée des pixels actifs et est expliquée dans la section suivante (section 6.2). Suite à la phase de vérification, le pixel est capable d'exécuter une réinitialisation locale (interne). Par conséquent, chaque pixel du système fonctionne de façon autonome et est capable de déclencher une lecture mais aussi sa propre réinitialisation. Ensuite, le système de lecture passe en phase de calcul du temps d'intégration de chaque pixel en se basant sur l'instant d'occurrence de la première requête de cette itération de lecture. On note que la phase de vérification et de réception des requêtes de lecture sont décorrélées. En d'autres termes, le système est capable d'une part, de recevoir des nouvelles requêtes de lecture, et d'autre part, d'établir une communication avec les pixels pour effectuer la vérification nécessaire.

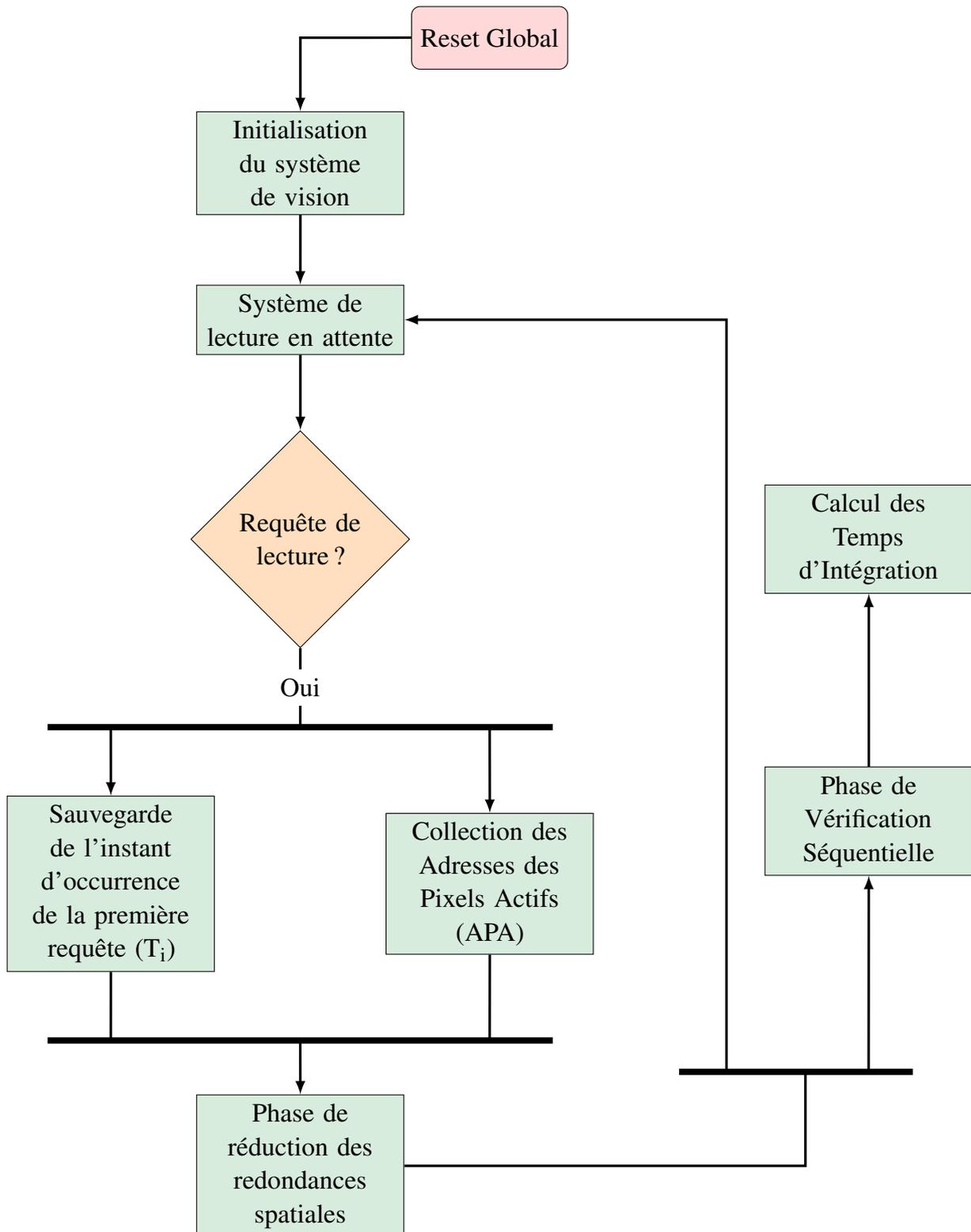


FIGURE 6.2 – Organigramme du fonctionnement du système de lecture

## 6.2 Système de lecture asynchrone

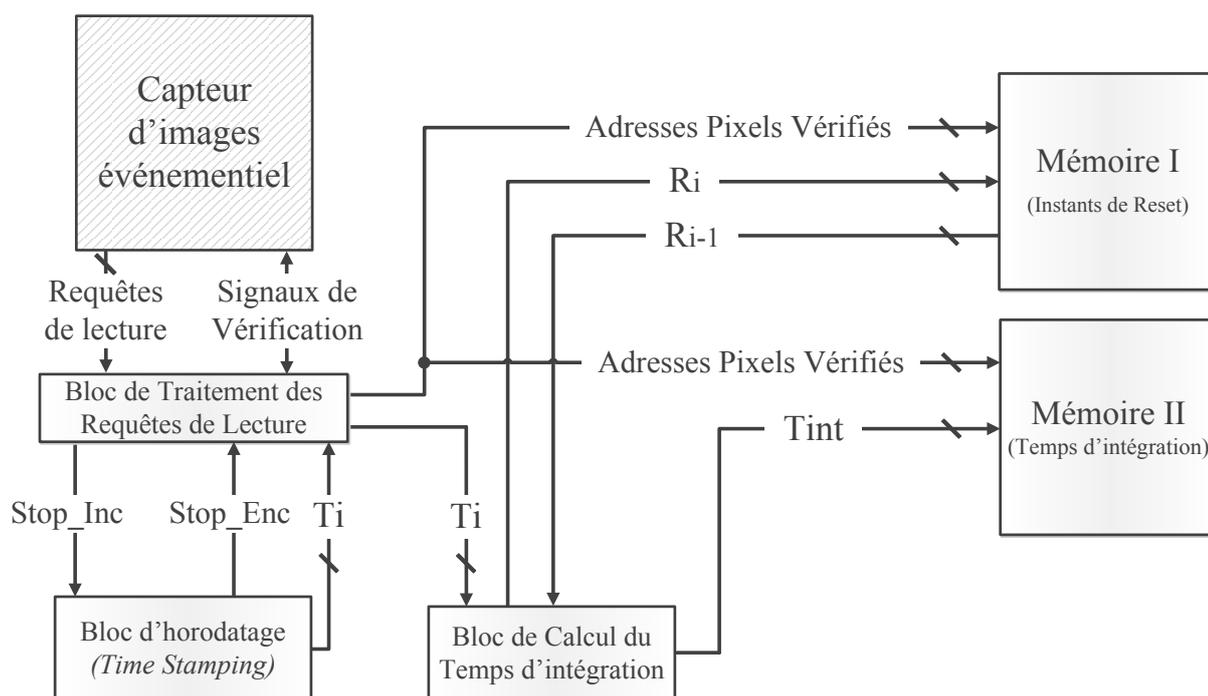
Le système de lecture est un système numérique conçu en logique asynchrone. Ce système est chargé de la capture de l'information lumineuse du capteur d'images. En d'autres termes, il est essentiellement responsable de l'établissement de la communication entre les pixels événementiels et ses composants, de l'encodage de l'information lumineuse des pixels et de la réduction des redondances spatiales. Dans ce qui suit, on détaille le fonctionnement du système de lecture, en se basant sur la figure 6.2, à travers son architecture présentée par les figures 6.3, 6.4 et 6.5.

### 6.2.1 Architecture globale du système de lecture

Le système de lecture du capteur d'images, proposé dans le cadre de cette thèse, est un système numérique complètement asynchrone. De ce fait, les blocs de ce système fonctionnent suivant un protocole de communication asynchrone. De plus, l'activité du système de lecture est adaptée aux événements détectés au niveau des pixels. On note qu'au cours de cette section, qui explique le système de lecture, on considère le pixel événementiel comme étant un détecteur d'événement capable d'établir une communication asynchrone avec le système de lecture. Le fonctionnement détaillé du pixel événementiel est expliqué par la suite (section 6.4). Selon l'organigramme de la figure 6.2, le système de vision est initialisé par un signal de reset global. Ce dernier initialise le système de lecture ainsi que le capteur d'images.

L'initialisation du système de lecture implique la mise à zéro du bloc d'horodatage plus précisément du compteur synchrone et des mémoires. Suite à l'initialisation du système de lecture, ce dernier lance le compteur synchrone du bloc d'horodatage (figure 6.3 et 6.4) et demeure en attente de la première requête de lecture. L'information temporelle fournie par le compteur est nécessaire pour le codage temporel du pixel événementiel *Time-to-First-Spike* : *TFS*. En d'autres termes, le bloc d'horodatage permet la conversion tension temps de l'information lumineuse du pixel.

Après la réception d'une première requête, une itération de lecture est alors lancée. Le bloc de traitement des requêtes de lecture ordonne, à travers le signal *Stop\_Inc* au bloc d'horodatage d'arrêter l'incrémementation du compteur synchrone. Le bloc d'horodatage envoie l'instant d'occurrence de la première requête de lecture  $T_i$ . Pendant cette itération, le bloc de traitement des requêtes de lecture continue la réception des nouvelles requêtes des pixels pour une durée de temps spécifique  $\Delta_t$ . L'intervalle de temps  $\Delta_t$  est précisé via le bloc d'horodatage et contrôle alors la durée de la phase de lecture. Une fois que  $\Delta_t$  est écoulé, le bloc d'horodatage signale la fin de la phase de lecture en cours à travers son signal *Stop\_Enc*. Le bloc de traitement arrête alors la réception des requêtes de lecture et se remet de nouveau en phase d'attente d'une nouvelle requête de lecture. Ainsi, à la fin d'une itération de lecture, le système de traitement des requêtes fournit un groupe d'adresses des pixels actifs ainsi que l'instant d'occurrence de la première



**FIGURE 6.3** – Architecture globale du système de lecture numérique asynchrone

requête de lecture  $T_i$ . Cette information temporelle est associée à l'ensemble des pixels de la même itération, effectuant ainsi une suppression des redondances spatiales existantes dans le flot de données en sortie du capteur d'images. L'approche de réduction des redondances spatiales est expliquée plus précisément dans la section 6.3.

A la fin de l'itération de lecture, les pixels actifs se réinitialisent et se remettent de nouveau en phase de détection des événements. Par ailleurs, le système de lecture asynchrone calcule le temps d'intégration  $T_{int}$  de chaque pixel actif, durant la dernière itération de lecture, en se référant à l'information temporelle  $T_i$  du groupe en cours de traitement et le dernier instant d'initialisation du pixel  $R_{i-1}$  déjà sauvegardé dans la mémoire I. Sachant que  $T_{int}$  est le temps écoulé entre les deux instants  $T_i$  et  $R_{i-1}$ . Après la fin de calcul de chaque temps d'intégration, la mémoire II est alors actualisée par les nouvelles valeurs de  $T_{int}$ . La mémoire I est également mise à jour, en sauvegardant les nouveaux instants d'initialisation des pixels actifs pendant cette itération.

Selon la figure 6.5, suite à la fin de l'itération de lecture, les adresses des pixels actifs (APA) ainsi que  $T_i$  sont transférés vers une mémoire tampon (FIFO) (Figure 6.5). Le système de lecture passe ensuite en phase de vérification séquentielle. Ces deux phases de fonctionnement élémentaires sont implémentées au niveau du bloc de traitements des requêtes et elles seront détaillées au cours de la présentation de ce bloc.

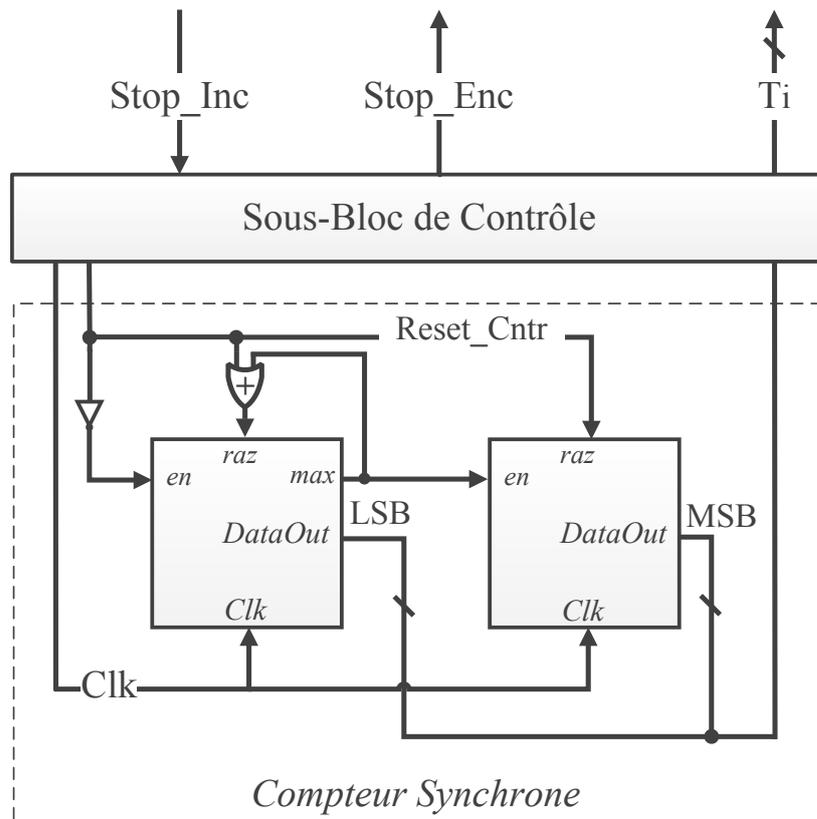


FIGURE 6.4 – Architecture du bloc d'horodatage (*Time Stamping Bloc*)

### 6.2.2 Bloc d'horodatage (Time Stamping Bloc)

Le bloc d'horodatage est responsable d'une part du codage de l'information lumineuse du pixel et d'autre part de contrôler la durée de l'itération de lecture en précisant l'intervalle de temps  $\Delta_t$  qui permet de gérer l'aspect temporel de la phase de réduction des redondances spatiales.

Le bloc d'horodatage est composé de deux sous blocs élémentaires : le compteur synchrone et le sous-bloc de contrôle. En premier lieu, le sous-bloc de contrôle est chargé d'établir la communication avec le bloc de traitement des requêtes de lecture. Pour cela, le signal *Stop\_Inc* et le signal *Stop\_Enc* sont reçus et envoyés par le sous-bloc de contrôle respectivement.

En deuxième lieu, le compteur synchrone du bloc d'horodatage permet la conversion tension temps de l'information lumineuse du pixel en fournissant l'instant d'occurrence  $T_i$  de la première requête d'une itération.

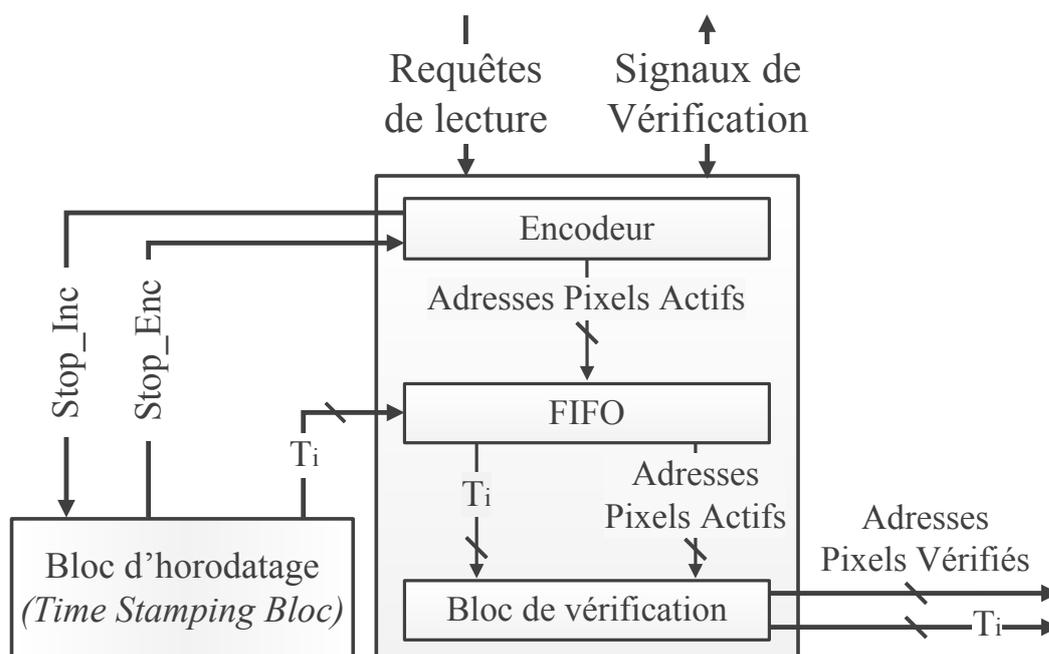


FIGURE 6.5 – Architecture du bloc de traitement asynchrone des requêtes de lecture

### 6.2.3 Architecture du bloc de traitement des requêtes de lecture

La réception et le traitement des requêtes de lecture sont réalisés au niveau du bloc de traitement des requêtes de lecture. Ce dernier est composé de trois sous-blocs : l'encodeur, la mémoire tampon FIFO et le bloc de vérification. La figure 6.5 illustre le bloc de traitement des requêtes de lecture. Ce bloc est chargé en premier lieu, via l'encodeur, de grouper les requêtes de lecture et par la suite les adresses des pixels actifs. En deuxième lieu, par l'intermédiaire de son sous-bloc de vérification, il vérifie l'authenticité des requêtes de lecture. Le terme "vérification" est employé pour une phase de fonctionnement attribuée au système de lecture. Cette phase est détaillée ultérieurement (voir section 6.2.3.3).

#### 6.2.3.1 Encodeur - La phase de réception des requêtes de lecture

L'encodeur est le bloc responsable de la gestion de la phase de réception des requêtes de lecture. Ces requêtes de lecture sont reçues par deux composantes. Par conséquent, la lecture événementielle adaptée à notre capteur d'images implique le besoin d'un bloc qui encode les adresses des pixels actifs en se basant sur les composantes  $x$  et  $y$  des requêtes de lecture reçues. D'où l'implémentation de l'encodeur au sein du bloc de traitement des requêtes de lecture.

Comme on l'a mentionné précédemment, notre capteur d'images asynchrone fonctionne sans trame mais par itération de lecture. La phase de réception des requêtes de

lecture est implémentée au niveau de l'encodeur et du bloc d'horodatage. En fait, la durée de la phase de réception des requêtes de lecture est temporellement contrôlée par l'encodeur et le bloc d'horodatage à travers les signaux *Stop\_Inc* et le signal *Stop\_Enc* échangés entre ces deux blocs. A la fin de cette phase de fonctionnement, l'encodeur et le bloc d'horodatage envoient à la mémoire intermédiaire (FIFO) le groupe des adresses des pixels actifs (*Adresses Pixels Actifs : APA*) et l'information temporelle  $T_i$  associée.

### 6.2.3.2 Mémoire tampon : FIFO

La mémoire intermédiaire, située entre l'encodeur et le bloc de vérification, permet au système de lecture de décorréliser les deux phases de fonctionnement : la phase de réception des requêtes et la phase de vérification. Par conséquent, le bloc de mémoire tampon de type First-In-First-Out (FIFO) maintient d'une part une lecture simultanée et continue des pixels actifs, et d'autre part le fonctionnement sans trame de notre système de vision. De plus, la mémoire FIFO diminue l'effet de latence causée par la vérification séquentielle. On note que la profondeur (taille) de la mémoire tampon dépend basiquement de la résolution spatiale du capteur d'images et de l'activité de la scène capturée.

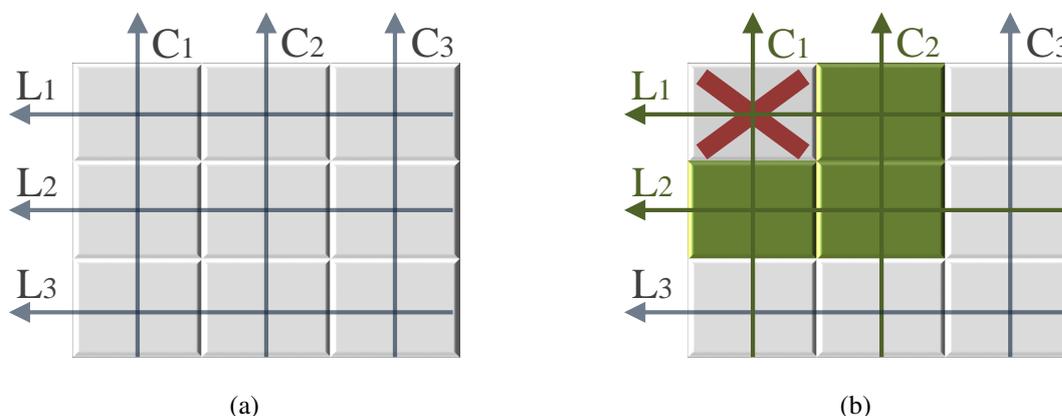
### 6.2.3.3 Bloc de vérification - La phase de vérification

Le bloc de vérification exécute une fonctionnalité essentielle pendant l'extraction et le codage de l'information lumineuse des pixels. Afin de mettre en avant le rôle de ce bloc dans l'architecture du système de lecture, on commence par expliquer le fonctionnement au cours de la phase de vérification et la raison de son besoin.

La communication entre les pixels événementiels et le système de lecture est établie par des signaux numériques. Idéalement, on pourrait attribuer à chaque pixel son propre signal de requête et d'acquiescement. Cependant, cette approche s'avère désavantageuse car elle ajoute de très nombreux signaux supplémentaires (2 fils par pixel), ce qui diminue considérablement la résolution spatiale du capteur. Par conséquent, on fait appel à une signalisation partagée entre les pixels. Ainsi, l'adresse du pixel est obtenue par le croisement des deux signaux de requête ligne et colonne. En d'autres termes, chaque adresse est représentée par un premier signal ligne et un deuxième signal colonne. Par exemple, un signal  $S_{xy}$  est un signal associé au pixel  $P_{xy}$  situé au croisement de la ligne  $x$  et de la colonne  $y$ . La figure 6.6a illustre une matrice (3x3) de pixels inactifs où chaque pixel a accès à deux signaux orthogonaux. Ainsi, une requête de lecture envoyée par le pixel  $P_{xy}$  rend les signaux requête ligne  $L_x$  et requête colonne  $P_y$  actifs, ce qui n'est pas souhaité.

La figure 6.6b, montre un exemple dans lequel les pixels  $\{P_{21}, P_{12}, P_{22}\}$  sont actifs. Ils envoient leurs requêtes de lecture au système numérique en activant les signaux de requêtes  $\{L_1, L_2, C_1, C_2\}$ . Dans ce cas, lors de la réception des requêtes de lecture des trois pixels actifs, le pixel  $P_{11}$  étant sur le croisement des signaux de la première ligne et de la première colonne, semble également être actif. En adoptant le système de lecture

simultanée proposé, cette fausse requête est prise en compte par le système numérique lors de la lecture des pixels réellement actifs.



**FIGURE 6.6** – La signalisation partagée entre les pixels d’un capteur d’images. Illustration de deux matrices (3x3) de pixels où les pixels de la première sont inactifs dans (a). Trois événements sont détectés au niveau de la deuxième matrice (b) rendant par la suite les trois pixels  $P_{21}$ ,  $P_{12}$  et  $P_{22}$  actifs. Par conséquent, le pixel  $P_{11}$ , situé au croisement des signaux de requête actifs, semble également être actif. La couleur grise, verte et rouge représente respectivement les pixels inactifs, actifs et faussement actifs.

Afin d’éviter l’ajout de fils de requêtes propres pour chaque pixel du capteur d’images ainsi que la prise en compte des fausses requêtes de lecture, une phase de vérification est essentielle à la chaîne de lecture de notre système de vision et suit la phase de la réception des requêtes de lecture. Cette phase de fonctionnement examine séquentiellement les pixels collectés afin de vérifier leur état.

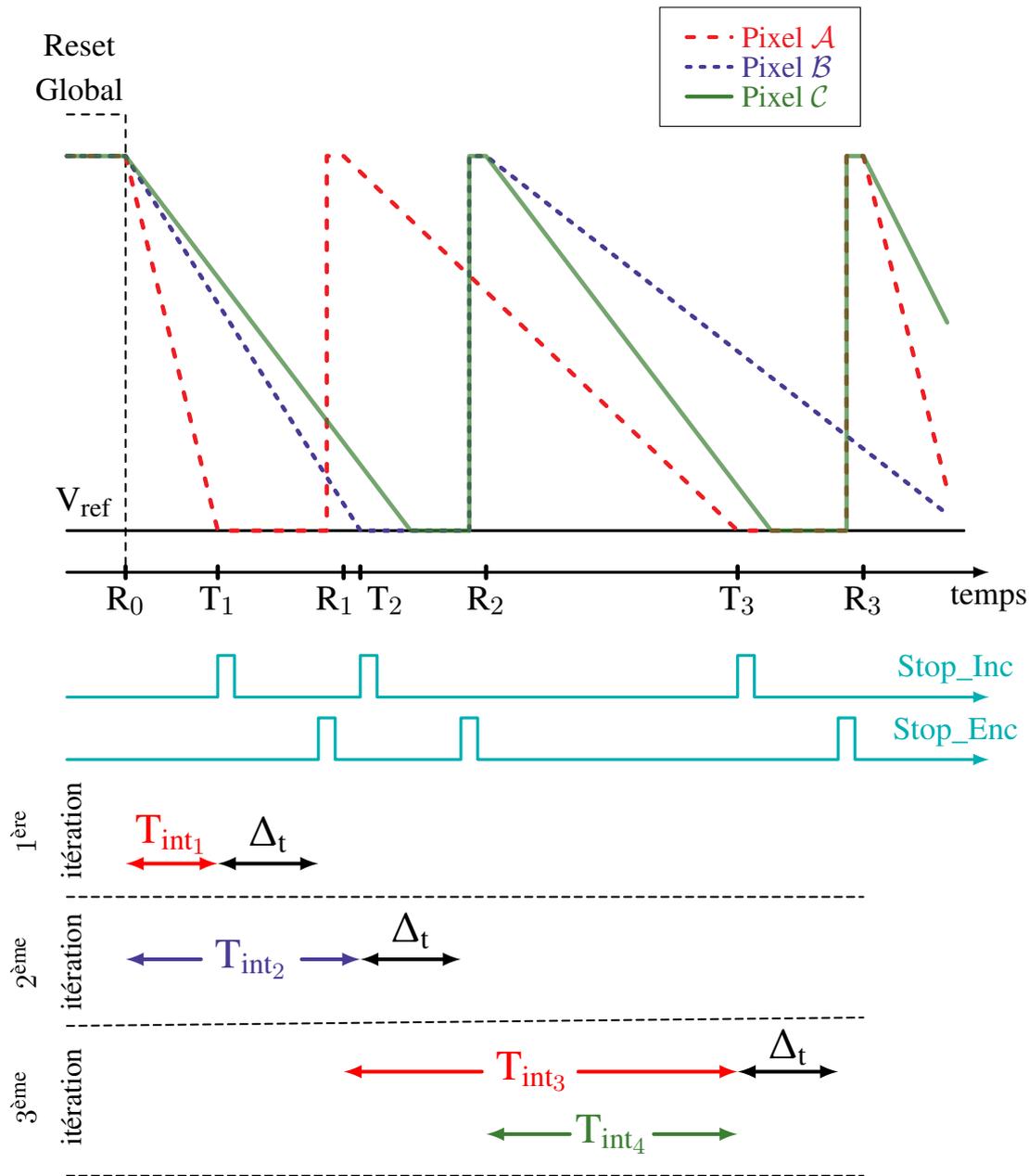
Du point de vue architectural, le bloc de vérification fonctionne par groupe de pixels. Il peut détecter la présence d’une nouvelle donnée dans la mémoire tampon et reçoit par la suite à son entrée le groupe des adresses des pixels actifs (*Adresses Pixels Actifs : APA*) ainsi que leur information temporelle associée  $T_i$  provenant de la FIFO. Ensuite, le bloc de vérification interroge séquentiellement, à travers les signaux de vérification, les pixels actifs de ce groupe en leur envoyant un signal qui examine leurs états. Suite à chaque vérification, le pixel interrogé envoie sa réponse au bloc de vérification. Selon cette dernière, les pixels qui s’avèrent inactifs ne sont pas pris en compte. A la fin de chaque phase de vérification, seules les adresses des pixels réellement actifs sont conservées. Par la suite, le bloc de vérification envoie les adresses des pixels vérifiés (*Adresses Pixels Vérifiés : APV*) ainsi que l’information temporelle ( $T_i$ ) associée au bloc chargé du calcul des temps d’intégration.

### 6.3 Réduction des redondances spatiales

Notre système de lecture effectue une réduction des redondances spatiales lors de la réception des requêtes de lecture. Cette section introduit un exemple, en utilisant 3 pixels, de la méthode de réduction des redondances spatiales. Dans ce qui suit, les instants  $R_i$  font référence à la fin de la phase d'initialisation (début de la phase d'intégration) des pixels, tandis que les instants  $T_i$  indiquent la fin de la phase d'intégration. Ce traitement de données est appliqué grâce à la lecture simultanée des pixels actifs. En fait, le procédé mis en œuvre consiste à rassembler les adresses des pixels actifs entre les instants d'occurrence de requêtes de lecture  $T_i$  et  $T'_i$  avec  $T'_i = T_i + \Delta_t$ . Sachant que  $\Delta_t$  est un intervalle de temps spécifique qui gère la durée de la phase de réception des requêtes de lecture. Son effet sur la qualité de l'image résultante est discuté ultérieurement (voir chapitre 7). En effet, les pixels actifs d'un groupe  $G_i$  partagent une seule et même information temporelle  $T_i$  qui précise la fin de la phase d'intégration du premier pixel actif de ce groupe. Alors, les pixels qui appartiennent au groupe  $G_i$  associé à l'instant  $T_i$  et qui partagent la même information temporelle  $R_i$  ont le même temps d'intégration  $T_{int_i}$  et par conséquent la même luminance, s'ils ont été initialisés au même instant.

Une seule lecture est donc effectuée pour un ensemble de pixels qui partagent une information temporelle liée à la fin de leurs phases d'intégration. En d'autres termes, la redondance existante dans le flot de données en sortie du capteur d'images est réduite. Cette redondance est une redondance spatiale vu que les informations temporelles associées à chaque pixel encodent l'intensité lumineuse incidente. Cependant, la lecture parallélisée et la notion d'initialisation (Reset) locale après chaque itération attribuée aux pixels événementiels impliquent la possibilité que les pixels d'un groupe  $G_i$  qui partagent l'information temporelle  $T_i$  soient éventuellement initialisés à différents instants. En d'autres termes, des pixels soumis à des intensités lumineuses différentes peuvent appartenir à un même groupe de pixels actifs. Cette possibilité ne s'avère pas pénalisante vu que l'information temporelle  $R_i$  de chaque pixel est mémorisée ce qui fait du calcul du temps d'intégration une simple formalité en se référant à  $R_{i-1}$  du pixel et en utilisant la nouvelle information temporelle  $T_i$ .

Afin de visualiser le procédé de suppression des redondances spatiales, on prend l'exemple de la figure 6.7. Cette figure comporte le chronogramme de trois pixels événementiels  $\mathcal{A}$ ,  $\mathcal{B}$  et  $\mathcal{C}$  originellement initialisés au même instant  $R_0$ . Cette figure montre aussi l'activité des deux signaux *Stop\_Inc* et *Stop\_Enc* qui précisent le début et la fin de chaque itération de lecture sont échangés entre deux blocs de l'architecture du système de lecture (section 6.2). En raison des différentes intensités lumineuses incidentes sur chaque pixel, les pixels atteignent le niveau de détection d'événement à des instants différents. Le pixel  $\mathcal{A}$  est le premier à croiser ce niveau à l'instant  $T_1$ . En conséquence d'une détection d'événement, le pixel  $\mathcal{A}$  envoie une requête de lecture qui déclenche le procédé de réception des requêtes de lecture de tous les pixels actifs entre les instants  $T_1$  et  $T'_1$  avec  $T'_1 = T_1 + \Delta_t$ . A l'instant  $T'_1$ , le circuit de lecture arrête la réception des



**FIGURE 6.7** – Chronogramme de trois pixels événementiels illustrant la réception des requêtes de lecture et la suppression des redondances spatiales. Les instants  $R_i$  font référence à la fin de la phase d’initialisation (début de la phase d’intégration) des pixels. Tandis que les instants  $T_i$  indiquent la fin de la phase d’intégration. Les signaux  $Stop\_Inc$  et  $Stop\_Enc$ , échangés entre le bloc d’horodatage et le bloc de traitement des requêtes de lecture de l’architecture du système de lecture, annoncent le début et la fin de chaque itération de lecture.

demandes de lecture et envoie les adresses des pixels actifs afin de calculer leurs temps d'intégration. Dans cet exemple, pendant la première itération, seul le pixel  $\mathcal{A}$  est actif. Le temps d'intégration du pixel est subséquemment calculé en se référant aux instants qui indiquent le début  $R_0$  et la fin  $T_1$  de sa phase d'intégration :  $T_{int_1} = T_1 - R_0$ . Le pixel  $\mathcal{A}$  est ensuite réinitialisé ce qui lui permet de nouveau de se comporter comme détecteur d'événement.

La deuxième itération est déclenchée par le pixel  $\mathcal{B}$  qui atteint le niveau de détection à  $T_2$  et envoie une requête de lecture au système de lecture. Ce dernier sauvegarde l'instant d'occurrence  $T_2$  de la requête. Pendant l'intervalle de temps  $\Delta_t$ , le pixel  $\mathcal{C}$ , devient actif et envoie sa requête de lecture. Son adresse est alors prise en compte et associée à l'instant de la première requête de lecture de cette itération. Pendant cette itération, les deux pixels  $\mathcal{B}$  et  $\mathcal{C}$  sont associés au même temps d'intégration  $T_{int_2}$  vu qu'ils partagent les mêmes références de temps  $R_0$  et  $T_2$ . La redondance spatiale existante est alors supprimée en lisant simultanément deux pixels et en leurs attribuant une seule information lumineuse.

La troisième itération est déclenchée par le pixel  $\mathcal{A}$  qui atteint le niveau de détection à l'instant  $T_3$  avec un temps d'intégration :  $T_{int_3} = T_3 - R_1$ . Le pixel  $\mathcal{C}$  est aussi pris en compte pendant cette itération. Le groupe de pixels de la troisième itération comporte alors deux pixels actifs mais initialisés à des instants différents. Les pixels  $\mathcal{A}$  et  $\mathcal{C}$  sont initialisés respectivement aux instants  $R_1$  et  $R_2$  ce qui implique que ces deux pixels n'ont pas le même temps d'intégration. Le temps d'intégration du pixel  $\mathcal{C}$  est ensuite calculé :  $T_{int_4} = T_3 - R_2$ . Par conséquent, on peut dire que cette approche n'est pas limitée qu'à la réduction des redondances spatiales dans le flot de données en sortie du capteur d'images. Cette approche diminue d'avantage le flot de données sortant en recevant et traitant simultanément toutes les requêtes des pixels actifs, même les pixels qui sont initialisés à des instants différents. En d'autres termes, les pixels qui n'ont pas la même luminance. De plus, la lecture événementielle du capteur d'images est assurée, même pendant le calcul du temps d'intégration et la vérification.

Enfin, l'approche de réduction des redondances spatiales expliquée au cours de cette section diminue considérablement le flot de données en sortie de notre capteur d'images. Les résultats de simulations de cette approche sont présentés au chapitre 7.

## 6.4 Le pixel événementiel TFS

Cette section est dédiée à l'explication du fonctionnement haut niveau du pixel. Le pixel événementiel asynchrone proposé dispose de 4 phases de fonctionnement élémentaires : la phase d'initialisation (globale ou locale) la phase d'intégration, la phase de requête et d'acquiescement et la phase de vérification. La signalisation entre le pixel événementiel et le système de lecture est assurée par un protocole de communication asynchrone *handshake* à 4 phases (voir chapitre 3). L'information lumineuse du pixel est

temporellement codée suivant le codage *Time-to-first Spike* (voir section 2.7.4)

L'organigramme de la figure 6.8 récapitule le fonctionnement du pixel événementiel. Les pixels de notre capteur d'images sont simultanément initialisés lors de la première mise en marche du système de vision. Par la suite, les pixels se réinitialisent et déclenchent seuls leurs phases d'intégration. Afin d'expliquer le fonctionnement de nos pixels événementiels, on se propose de décrire un unique pixel.

### 6.4.1 Fonctionnement du pixel événementiel

Suite à son initialisation, la durée de la phase d'intégration d'un pixel, et par la suite son temps d'intégration, varient en fonction de la lumière incidente sur le pixel. Pendant cette phase, le pixel est chargé de la détection d'un nouvel événement. Afin de détecter un événement, le pixel compare constamment la tension aux bornes de l'élément photosensible à une tension de référence imposée et prédéfinie. On considère qu'un nouvel événement est détecté lorsque ces deux tensions se croisent. L'impact du choix de la valeur de la tension de référence est expliqué ultérieurement (section 6.4.2).

Selon l'organigramme de la figure 6.8, la requête de lecture est envoyée suite à la détection de l'événement. Dans cet état, le pixel mémorise l'événement. Dans ce manuscrit, le terme requête locale fait référence à la mémorisation d'un événement dans le pixel. On note que la présence de la requête locale dans le pixel indique que ce dernier a été effectivement actif pendant la dernière itération de lecture. Suite à la réception de la requête de lecture, le système, et non pas le pixel, sauvegarde l'instant d'occurrence de cette dernière. Cette valeur de temps est utilisée lors du calcul du temps d'intégration du pixel. Selon le protocole de communication asynchrone (*Handshake Protocol*), un bloc ayant envoyé une requête doit être impérativement acquittée. Par conséquent, le pixel actif demeure en attente d'acquiescement. Provenant du circuit de lecture, l'acquiescement informe le pixel de la prise en compte de sa requête de lecture et lui permet par la suite la remise à zéro de son signal de requête. L'état du pixel actif est maintenu en attendant la phase de vérification. En fait, le début de cette dernière est contrôlé par les signaux de vérification envoyés par le système de lecture. Une fois que le pixel passe en phase de vérification, il répond positivement au système de lecture et exécute une réinitialisation locale (*Reset Local*). Cette réinitialisation locale lui permet de passer à une nouvelle phase d'intégration, et donc de détecter un nouvel événement. On note que suite à la vérification, le système de lecture calcule le temps d'intégration du pixel en se référant à l'instant d'occurrence de la requête de lecture et le dernier instant d'initialisation de ce pixel.

### 6.4.2 Niveau de détection des événements

L'information lumineuse du pixel est échantillonnée grâce à un bloc de conversion analogique numérique à 1 bit intégré dans le pixel. L'échantillonnage est effectué sui-

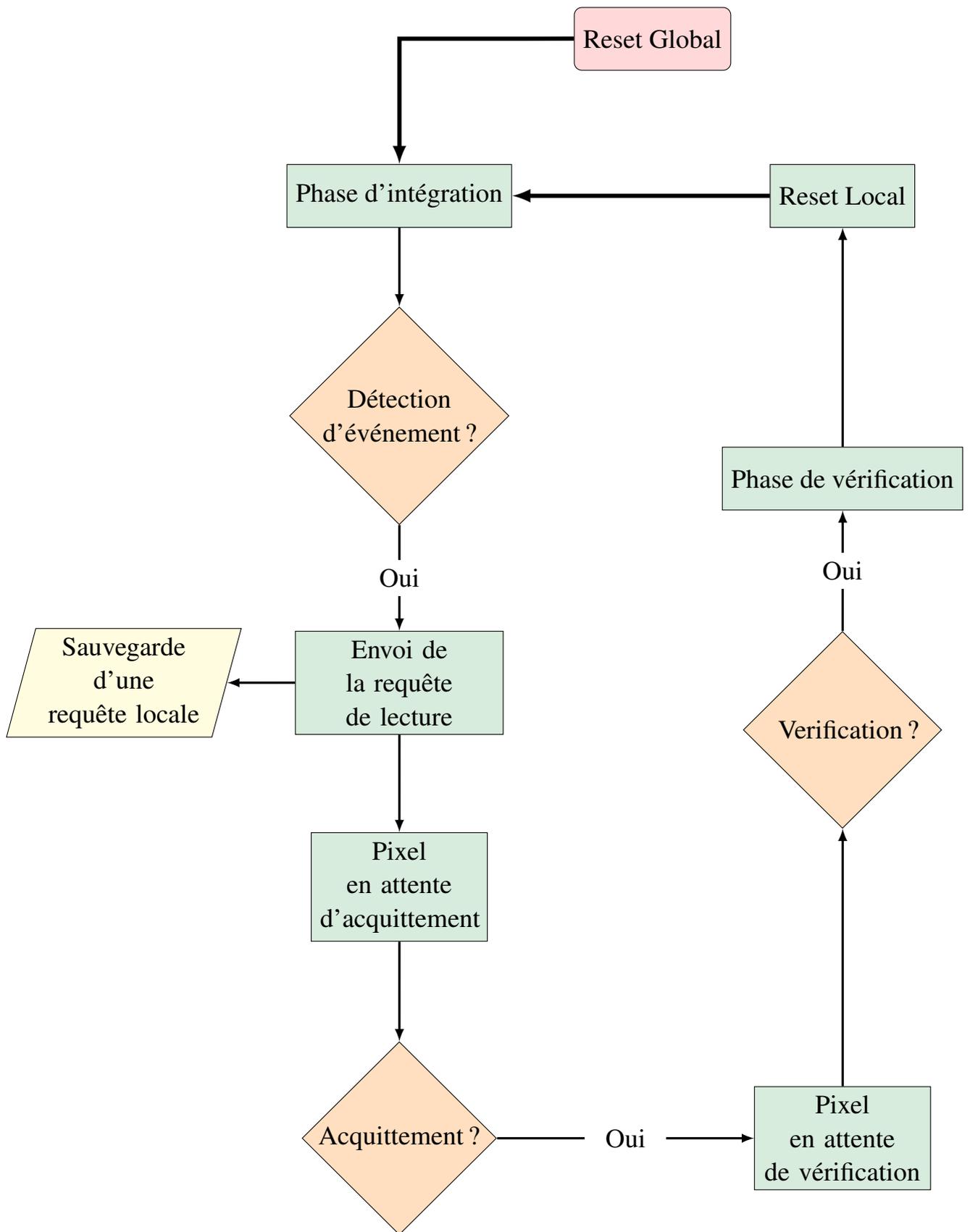


FIGURE 6.8 – Organigramme du fonctionnement du pixel événementiel

vant un niveau de détection. Ce dernier doit être réglé d'une manière à respecter la réponse des pixels soumis à une luminosité maximale, dits pixels blancs, ainsi que celle des pixels à une intensité lumineuse nulle, dits pixels noirs. Par exemple, avec un seuil haut d'échantillonnage, proche du niveau d'initialisation du pixel, on risque de ne pas différencier les réponses des pixels ayant de faibles intensités lumineuses. En revanche, un seuil bas d'échantillonnage augmente le risque de saturation du compteur du bloc d'horodatage et diminue la vitesse de réponse et donc celle de lecture des pixels.

Idéalement, on pourrait attribuer un niveau d'échantillonnage propre à chaque pixel qui varie suivant les intensités lumineuses successivement détectées afin de s'adapter aux environnements à éclairage vivement variable. Néanmoins, d'un point de vue conception en technologie conventionnelle (2D), cette approche s'avère fortement désavantageuse et pénalisante, en terme de surface de silicium, vue qu'elle nécessite l'affectation d'un signal de référence pour chaque pixel. Par ailleurs, du point de vue système, un niveau de détection variable et propre à chaque pixel nécessite un bloc de mémoire qui suit et sauvegarde les différentes valeurs des niveaux d'échantillonnage vu que cette information est nécessaire lors du calcul du temps d'intégration du pixel. En conséquence et dans le but de faciliter le design du capteur d'images, on attribue un seul niveau d'échantillonnage pour l'ensemble des pixels événementiels.

### 6.4.3 Initialisation locale

Le fonctionnement de notre pixel événementiel, illustré par l'organigramme de la figure 6.8 commence par une initialisation globale (Reset Global). L'initialisation globale déclenche l'ensemble du système de vision (matrice de pixels et système de lecture). Au-delà de cette initialisation, le pixel est capable de se réinitialiser, en d'autres termes le pixel exécute une initialisation locale (Reset Local : *Self Reset*) après chaque itération. Les pixels de notre capteur d'images fonctionnent donc indépendamment des uns des autres. Cette approche permet théoriquement l'obtention d'une dynamique de fonctionnement (*Dynamic Range*) optimale, vu que chaque pixel possède son propre temps d'intégration associé à l'intensité lumineuse incidente. De plus, une initialisation locale peut éventuellement augmenter le rapport signal sur bruit (*SNR*) comme cela est montré dans [PRJ07].

La lecture parallélisée du capteur d'images et l'initialisation locale du pixel implique le besoin d'une mémoire qui sauvegarde les instants de reset de chaque pixel vu que cette information est nécessaire lors du calcul du temps d'intégration. Cette mémoire est, comme expliqué précédemment, externe à la matrice photosensible.

### 6.4.4 Temps mort du pixel

Le temps écoulé entre de l'envoi la requête de lecture et l'initialisation locale est considéré comme étant le temps mort du pixel. En fait, c'est le temps nécessaire pour

traiter une ou plusieurs requêtes de lecture. Pendant ce temps, le pixel est aveugle au changement de sa lumière incidente. Pour cela, en concevant ce système de vision, on a intérêt à réduire cette valeur de temps afin d'avoir un fonctionnement efficace. Dans le but de limiter ce temps mort, on a exploité une solution qui consiste à partialement paralléliser la seule phase qui suscite un fonctionnement séquentiel : la phase de vérification. En d'autres termes, les pixels actifs appartenant au même groupe subissent une vérification exhaustive et non parallèle. Par conséquent, la durée de cette phase augmente proportionnellement au nombre de pixels actifs d'un groupe. Afin de réduire cette durée, on divise la matrice de pixels en sous-blocs, de sorte que les pixels appartenant à ces différents sous blocs soient vérifiés simultanément.

## 6.5 Conclusion

Dans ce chapitre, on a d'abord présenté la chaîne de lecture complète du pixel événementiel proposé. Ensuite, on a détaillé le fonctionnement du système de lecture asynchrone au travers de l'architecture de ses blocs et sous-blocs et de son organigramme de fonctionnement. Notre protocole de lecture est caractérisé par une lecture parallélisée des pixels actifs. En d'autres termes, les requêtes de lecture sont simultanément reçues par le système de lecture, évitant par la suite des erreurs de timing naturellement induites par une lecture exploitant un arbitrage non déterministe. Cette approche de lecture spécifique nous permet de réduire, voire de supprimer les redondances spatiales existantes dans le flot en sortie du capteur d'images événementiel. La réduction des redondances spatiales est expliquée et détaillée par l'intermédiaire d'un exemple. La phase de vérification, essentielle à notre protocole de lecture, est aussi expliquée. Enfin, le fonctionnement du pixel événementiel est détaillé et illustré par un organigramme.

# 7

## SIMULATION HAUT NIVEAU

### Sommaire

---

|       |                                       |     |
|-------|---------------------------------------|-----|
| 7.1   | Images numériques . . . . .           | 89  |
| 7.2   | Simulation Matlab . . . . .           | 90  |
| 7.2.1 | Protocole de simulation . . . . .     | 90  |
| 7.2.2 | Modélisation de $\Delta_t$ . . . . .  | 91  |
| 7.2.3 | Choix des images numériques . . . . . | 92  |
| 7.3   | Discussion des résultats . . . . .    | 97  |
| 7.4   | Conclusion . . . . .                  | 103 |

---

Ce chapitre présente les résultats de simulations haut-niveau du protocole de lecture sous MATLAB. La stratégie de lecture et la réduction des redondances spatiales sont évaluées au cours des simulations présentées ci-dessous. Pour appliquer le traitement d'images proposé au cours de cette thèse, on a utilisé des images numériques prises par un capteur d'images conventionnels. A la fin de chaque simulation, les images résultantes sont évaluées grâce à des paramètres et à des facteurs de qualité usuels.

### 7.1 Images numériques

On traite une image numérique monochromatique en niveaux de gris (*Grayscale Image*) comme étant une matrice de dimension  $M \times N$ , en notant que  $M$  et  $N$  représentent le nombre de ligne et de colonnes respectivement. Chaque élément de cette matrice est un pixel qui stocke une valeur, comprise entre 0 et 255, proportionnelle à l'intensité lumineuse incidente lors de la capture d'images. La valeur 0 correspond au noir (absence de photons incidents) tandis que la valeur 255 correspond au blanc. Dans ce cas, le pixel est soumis à la luminosité maximale. Les valeurs intermédiaires, entre 0 et 255, correspondent toutes à des niveaux de gris allant du noir au blanc. La figure 7.1 est un exemple d'une image, selon laquelle on peut voir les valeurs d'un nombre de pixels. On peut remarquer le changement de ces valeurs par rapport à la nuance de gris. On note que les valeurs des pixels varient entre le 0 et le 255, parce que chaque pixel est codé sur 8 bits. Le nombre de bits consacré pour le codage ainsi que le nombre de niveaux de gris ne sont pas choisis de façon aléatoire. En fait, 8 bits est le nombre de bits minimal nécessaire pour coder une image non pixélisée par rapport à la vision humaine et sans perte d'informations.



**FIGURE 7.1** – Exemple d’une image numérique

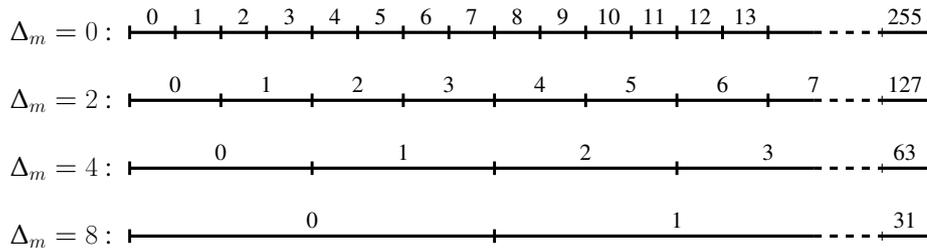
Au cours des simulations présentées, les images numériques utilisées sont capturées par des imageurs conventionnels et elles sont codées sur 8 bits.

## 7.2 Simulation Matlab

Dans le but de simuler le fonctionnement du système de lecture, on a utilisé le logiciel de programmation MATLAB. Ce choix est justifié par la simplicité de la gestion des images dans MATLAB qui les traite comme étant des matrices mathématiques.

### 7.2.1 Protocole de simulation

L’algorithme de simulation, sous MATLAB, suit fidèlement le chemin de la donnée de l’architecture réelle du système de lecture proposé. Il consiste en premier lieu à lire une image, déjà prise par un capteur d’images conventionnel. En deuxième lieu, le programme, en respectant les valeurs des pixels, crée un flot de données événementiel normalement généré en sortie d’un capteur d’images événementiel ainsi gérant l’ensemble des pixels de la matrice. En principe, les pixels ayant une grande valeur (pixels proche du blanc) seront les premiers à envoyer les requêtes de lecture et par la suite les premiers à être lus. Ceci dit, l’algorithme est chargé de créer l’ordre d’occurrence des requêtes de lecture et d’appliquer par la suite la méthode de réduction des redondances spatiales. Cependant, pour un langage de programmation de haut niveau comme MATLAB, la notion de temps physique n’existe pas. Pour cela, les grandeurs temporelles (instant de requête, instant d’initialisation,  $\Delta_t$ ), essentielles et nécessaires pour le bon fonctionnement de notre système de lecture doivent être discrétisées d’une façon différente. On rappelle que  $\Delta_t$  est l’intervalle de temps spécifique qui gère la durée de la phase de réception des requêtes de lecture et la phase de réduction des redondances spatiales. Au cours de ce



**FIGURE 7.2** – Fusionnement des niveaux de gris avec l’augmentation de la valeur du coefficient de réduction des redondances spatiales  $\Delta_m$

chapitre, on s’intéresse essentiellement à l’effet et à la discrétisation de  $\Delta_t$  afin que les simulations présentées valident effectivement la réduction des redondances spatiales.

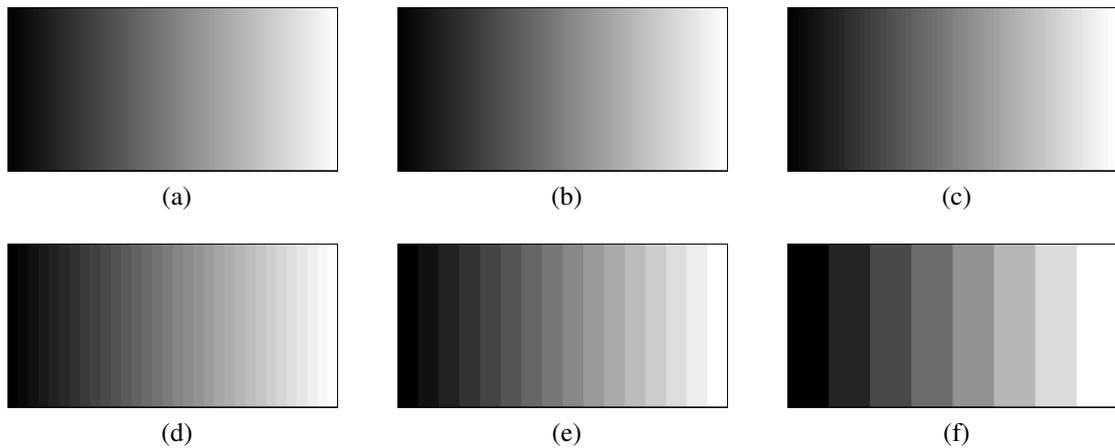
### 7.2.2 Modélisation de $\Delta_t$

Cette section est consacrée à l’explication de l’effet de  $\Delta_t$  sur la qualité de l’image résultante ainsi que la discrétisation de  $\Delta_t$  en un équivalent  $\Delta_m$  pour les simulations MATLAB.

Le système de lecture asynchrone proposé, au cours de cette thèse, utilise le paramètre  $\Delta_t$  pour contrôler le taux de réduction de redondances spatiales. Le paramètre  $\Delta_t$ , contrôlant aussi la durée de réception des requêtes de lecture, affecte la qualité de l’image résultante. En supposant que tous les pixels sont initialisés au même instant, pour un  $\Delta_t$  de valeur nulle, seuls les pixels de même niveau de gris sont associés au même groupe de pixels  $G_i$  (voir section 6.3). Néanmoins, en augmentant la valeur de  $\Delta_t$  (et en gardant l’hypothèse que tous les pixels sont initialisés au même instant) les requêtes de lecture des pixels actifs commencent à se regrouper. En d’autres termes, les pixels, qui dans le cas d’un imageur conventionnel, appartiendraient à des différents niveaux de gris, seront associés au même niveau de gris en utilisant la méthode de lecture proposée. Par conséquent, les niveaux de gris adjacents commencent à se fusionner formant par la suite un seul niveau de gris (Figure 7.2). On peut alors dire que le nombre de niveaux de gris d’une image, en sortie du système de vision événementiel et asynchrone proposé, varie d’une façon inversement proportionnelle à la valeur de  $\Delta_t$ .

Du point de vue du flot de données, le paramètre  $\Delta_t$  contrôle le nombre d’itérations de lecture effectuées pendant l’extraction de l’image. Il existe donc un compromis entre, d’une part, le nombre de niveaux de gris de l’image résultante et d’autre part, le nombre d’itérations nécessaire pour la lecture du capteur d’images proposé. Ce compromis est contrôlé et maîtrisé au travers de la valeur de  $\Delta_t$ .

Dans le but de modéliser et de simuler la méthode de réduction des redondances spatiales, on remplace le paramètre temporel  $\Delta_t$  par le paramètre  $\Delta_m$ . Ce dernier est l’image de  $\Delta_t$  dans le domaine de simulation haut niveau de MATLAB. En se basant sur un raisonnement identique, on applique le protocole de lecture événementiel en



**FIGURE 7.3** – Variation du nombre des niveaux de gris par rapport à  $\Delta_m$ . (a)  $\Delta_m = 0$ , 256 niveaux de gris. (b)  $\Delta_m = 2$ , 128 niveaux de gris. (c)  $\Delta_m = 4$ , 64 niveaux de gris. (d)  $\Delta_m = 8$ , 32 niveaux de gris. (e)  $\Delta_m = 16$ , 16 niveaux de gris. (f)  $\Delta_m = 32$ , 8 niveaux de gris.

utilisant différentes valeurs de  $\Delta_m$ , allant de 0 à 255. Afin de visualiser l'effet de la valeur de  $\Delta_m$  sur le nombre des niveaux de gris, on se réfère à la figure 7.3. On prédit, alors, que la qualité de l'image décroît inversement proportionnellement avec  $\Delta_m$ . Des simulations sont, alors, réalisées, dans le but de prouver l'efficacité de notre protocole de lecture événementiel.

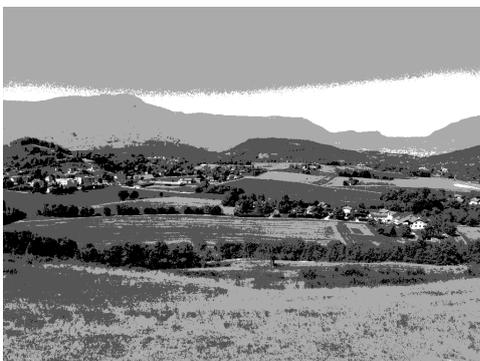
### 7.2.3 Choix des images numériques

Les images choisies pour les simulations de MATLAB sont des images prises par un appareil photo numérique conventionnel. Pour effectuer ces simulations, on a choisi quatre échantillons. La première image A qui couvre toute la dynamique et possède une structure détaillée dans son arrière plan. La deuxième image B représente les images qui couvrent un nombre limité des niveaux de gris avec un fond clair. La troisième image C est similaire à l'image B, mais avec un fond sombre couvrant quelques nuances de gris. La quatrième image D est constituée de tous les niveaux de gris possibles, avec un arrière plan et un avant plan détaillé. De plus, on a simulé le fonctionnement de notre système de lecture en utilisant des images classiques et standards dans le domaine du traitement de l'image. Les résultats de ces simulations sont présentés dans l'annexe A.

On vise grâce à ces quatre échantillons à visualiser le fonctionnement de notre système de lecture avec plusieurs types d'images. A la fin de chaque simulation, la qualité des images résultantes sont évaluées avec des paramètres qui estiment l'effet de la compression effectuée. Les figures 7.4, 7.5, 7.6 et 7.7 regroupent les images résultantes en fonction des principales valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1

(b)  $\Delta = 2$ , PSNR = 40.03 dB, MSSIM = 0.999(c)  $\Delta = 4$ , PSNR = 38.10 dB, MSSIM = 0.919(d)  $\Delta = 8$ , PSNR = 36.29 dB, MSSIM = 0.765(e)  $\Delta = 16$ , PSNR = 31.46dB, MSSIM = 0.635(f)  $\Delta = 32$ , PSNR = 22.51dB, MSSIM = 0.483(g)  $\Delta = 64$ , PSNR = 18.59dB, MSSIM = 0.273(h)  $\Delta = 128$ , PSNR = 8.52dB, MSSIM = 0.068

**FIGURE 7.4** – Résultat des simulations MATLAB : Image A. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.7.8)



(a) Image de référence, MSSIM =1



(b)  $\Delta = 2$ , PSNR = 28.70dB, MSSIM = 0.989



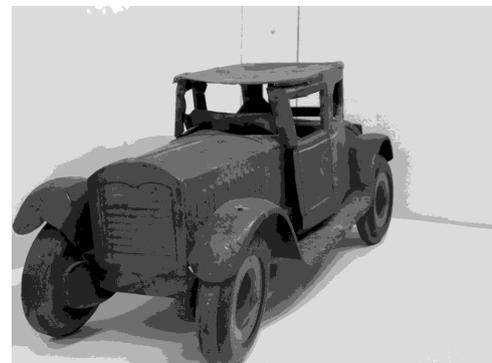
(c)  $\Delta = 4$ , PSNR = 28.02dB, MSSIM = 0.707



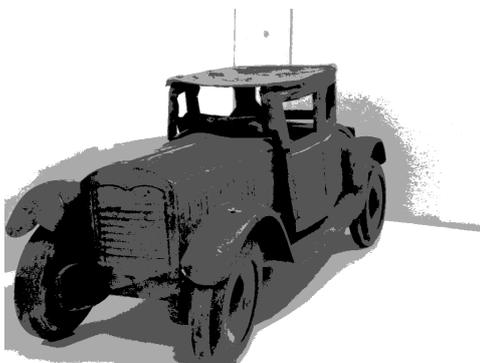
(d)  $\Delta = 8$ , PSNR = 28.25dB, MSSIM = 0.521



(e)  $\Delta = 16$ , PSNR = 28.20dB, MSSIM = 0.381



(f)  $\Delta = 32$ , PSNR = 24.44dB, MSSIM = 0.232



(g)  $\Delta = 64$ , PSNR = 16.56dB, MSSIM = 0.121

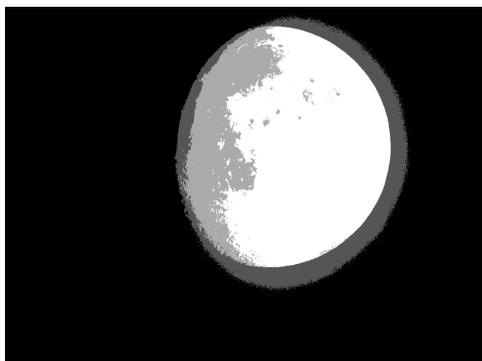
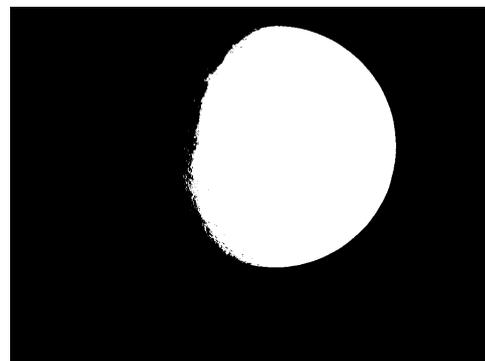


(h)  $\Delta = 128$ , PSNR = 11.4dB, MSSIM = 0.039

**FIGURE 7.5** – Résultat des simulations MATLAB : Image B. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.7.9)



(a) Image de référence, MSSIM = 1

(b)  $\Delta = 2$ , PSNR = 39.19dB, MSSIM = 0.990(c)  $\Delta = 4$ , PSNR = 37.75dB, MSSIM = 0.739(d)  $\Delta = 8$ , PSNR = 33.98dB, MSSIM = 0.419(e)  $\Delta = 16$ , PSNR = 28.72dB, MSSIM = 0.206(f)  $\Delta = 32$ , PSNR = 22.47dB, MSSIM = 0.095(g)  $\Delta = 64$ , PSNR = 18.55dB, MSSIM = 0.041(h)  $\Delta = 128$ , PSNR = 15.6dB, MSSIM = 0.013

**FIGURE 7.6** – Résultat des simulations MATLAB : Image C. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.7.10)



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 37.78 dB, MSSIM = 0.998



(c)  $\Delta = 4$ , PSNR = 35.37 dB, MSSIM = 0.953



(d)  $\Delta = 8$ , PSNR = 34.03 dB, MSSIM = 0.869



(e)  $\Delta = 16$ , PSNR = 30.02dB, MSSIM = 0.776



(f)  $\Delta = 32$ , PSNR = 21.92 dB, MSSIM = 0.647



(g)  $\Delta = 64$ , PSNR = 18.83dB, MSSIM = 0.432



(h)  $\Delta = 128$ , PSNR = 9.13dB, MSSIM = 0.136

**FIGURE 7.7** – Résultat des simulations MATLAB : Image D. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.7.11)

## 7.3 Discussion des résultats

Dans le but d'évaluer les résultats obtenus à la fin de chaque simulation, en d'autres termes, estimer la qualité des images résultantes, on s'est appuyé sur des paramètres spécifiques.

On commence par le nombre de lecture nécessaires pour une image. Ce paramètre détermine la taille du flot de données obtenu en sortie du capteur d'images événementiel proposé par rapport au flot de données en sortie du capteur d'images conventionnels. En fait, ce dernier, ayant une lecture déterministe à flot de données fixe, effectue  $M \times N$  opérations de lecture ( $M \times N$  conversions analogique-numériques) pour extraire l'information des pixels, en d'autres termes, lire une image, ou une trame dans le cas d'une vidéo. Dans le cas du protocole de lecture proposé, la lecture des pixels est événementielle, initiée par le pixels suite à la détection des événements et se fait par groupe de pixels actifs (voir section 6.2). Ce qui rend le flot de données en sortie de notre capteur, d'une part, adapté au contenu de la scène capturée, et d'autres part, compressé grâce à la méthode de réduction des redondances spatiales. De plus, l'effet de notre protocole de lecture évolue en fonction de la résolution du capteur d'images et du coefficient de réduction des redondances spatiales  $\Delta_m$ . En fait, le nombre de lecture maximal nécessaire pour une image est, dans notre cas, principalement lié à la dynamique de fonctionnement ( $d$ ) du pixel. Par exemple, en employant notre protocole de lecture, les images monochromatiques à 256 niveaux de gris (résolution =  $512 \times 384$ ), utilisées au cours des simulations présentés, nécessitent un maximum de 256 lectures pour extraire l'information de tous les pixels, contrairement aux imageurs conventionnels, où le nombre de lecture est lié à la résolution de l'imageur ( $M \times N$ ). Ceci est dû au fait que notre capteur effectue une lecture collective à tous les pixels appartenant à un même niveau de gris. Afin de visualiser l'effet de notre protocole de lecture, on estime le taux de lecture (*Asynchronous Image Sensor Reading Rate : AISRR*) nécessaire par rapport à celui de l'imageur conventionnel. La relation entre les deux taux de lecture est exprimée par l'équation (7.1)

$$\text{AISRR}(\%) = 100 \cdot \left( \frac{d}{M \cdot N} \right) \quad (7.1)$$

Ainsi, le fonctionnement de notre système de lecture et la méthode de réduction des redondances expliquent le taux de lecture inférieur à 1% obtenu à la fin des simulations des images A (Figure 7.8a), B (Figure 7.9a), C (Figure 7.10a) et D (Figure 7.11a) pour  $\Delta_m = 0$ . La valeur de ce paramètre diminue d'avantage en augmentant le coefficient de réduction de redondances spatiales ( $\Delta_m$ ).

La réduction des redondances spatiales revenant à une méthode de compression de données en sortie du capteur, il est indispensable d'estimer la qualité des images à la fin de chaque simulation. Pour cela, on fait appel à deux paramètres habituellement utilisés dans le domaine de traitement d'images. Le premier paramètre est le *Peak Signal to Noise Ratio (PSNR)* défini par l'équation (7.2) avec  $d$  la dynamique du signal (pour une

image à 8 bits,  $d = 256$ ) et MSE l'erreur quadratique moyenne (*Mean Squared Error*) entre l'image de référence ( $\text{Im}_o$ ) et l'image résultante ( $\text{Im}_r$ ) définie par l'équation (7.3). Le PSNR(dB), qui est facilement calculé, mesure les distorsions induites par la compression, en d'autres termes, il mesure la proximité de l'image compressée par rapport à l'image de référence au niveau du signal. Typiquement, une image monochromatique compressée est de bonne qualité si son PSNR varie entre 30 et 50 dB. Généralement, un PSNR de grande valeur indique que l'image résultante et reconstruction est de bonne qualité. Cependant, dans certain cas, même pour une grande valeur de PSNR, l'image résultante est de mauvaise qualité visuelle. Cela est dû au fait que le PSNR évalue uniquement la différence pixel à pixel entre l'image de référence et l'image résultante. Par conséquent, le PSNR n'est pas suffisant pour évaluer la qualité visuelle de reconstruction de l'image. Il n'est pas alors considéré comme une mesure objective de la qualité visuelle d'une image.

$$\text{PSNR}_{dB} = 10 \cdot \log \left( \frac{d^2}{\text{MSE}} \right) \quad (7.2)$$

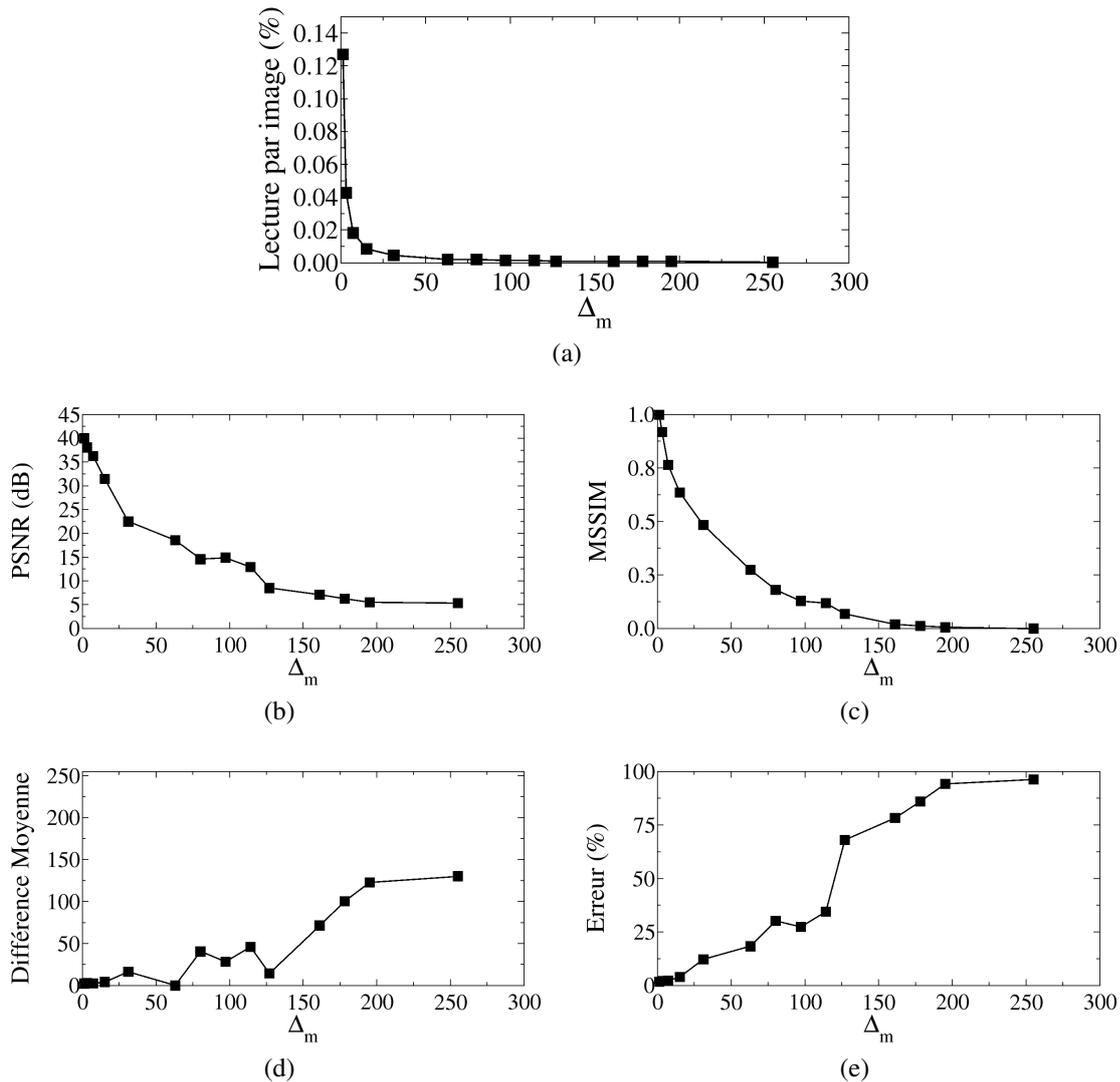
$$\text{MSE} = \frac{1}{M \cdot N} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left\| \text{Im}_o(i, j) - \text{Im}_r(i, j) \right\|^2 \quad (7.3)$$

Pour surmonter la limitation du PSNR, on utilise un deuxième paramètre, aussi employé lors de l'évaluation des images après compression : le *Structural SIMilarity (SSIM)* [Wan+04 ; ZB02]. Contrairement au PSNR, le SSIM est un paramètre non linéaire qui évalue l'effet de la compression en prenant en compte la perception visuelle humaine en prenant en compte la luminance, le contraste et la structure de l'image résultante [RH08]. Ce paramètre évalue l'effet de compression en calculant la moyenne et la variance d'un bloc de 8x8 qui se déplace de pixel par pixel sur l'image de référence et l'image compressée. SSIM est exprimé par l'équation (7.4) avec  $x$  et  $y$  sont deux signaux provenant des deux blocs (8x8). Sachant que  $\mu_x$  et  $\mu_y$  sont la moyenne,  $\sigma_x^2$  et  $\sigma_y^2$  sont la variance de  $x$  et  $y$  respectivement. Et  $\sigma_{xy}$  est la covariance entre  $x$  et  $y$  tandis que  $c_1$  et  $c_2$  sont des constantes fixées par [Wan+04].

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (7.4)$$

A la fin du calcul, on obtient un SSIM pour chaque bloc de l'image. Dans le but d'avoir un seul paramètre d'évaluation pour chaque image compressée, le paramètre MSSIM (*Mean Structural SIMilarity*) (7.5), qui comme son nom l'indique est la moyenne de tous les SSIM d'une image compressée. Le MSSIM d'une image compressée est compris entre 0 et 1. Pour des images compressées de bonne qualité, on vise un MSSIM proche de la valeur 1.

$$\text{MSSIM}(\text{Im}_o, \text{Im}_r) = \frac{1}{K} \sum_{k=1}^K \text{SSIM}(x_k, y_k) \quad (7.5)$$

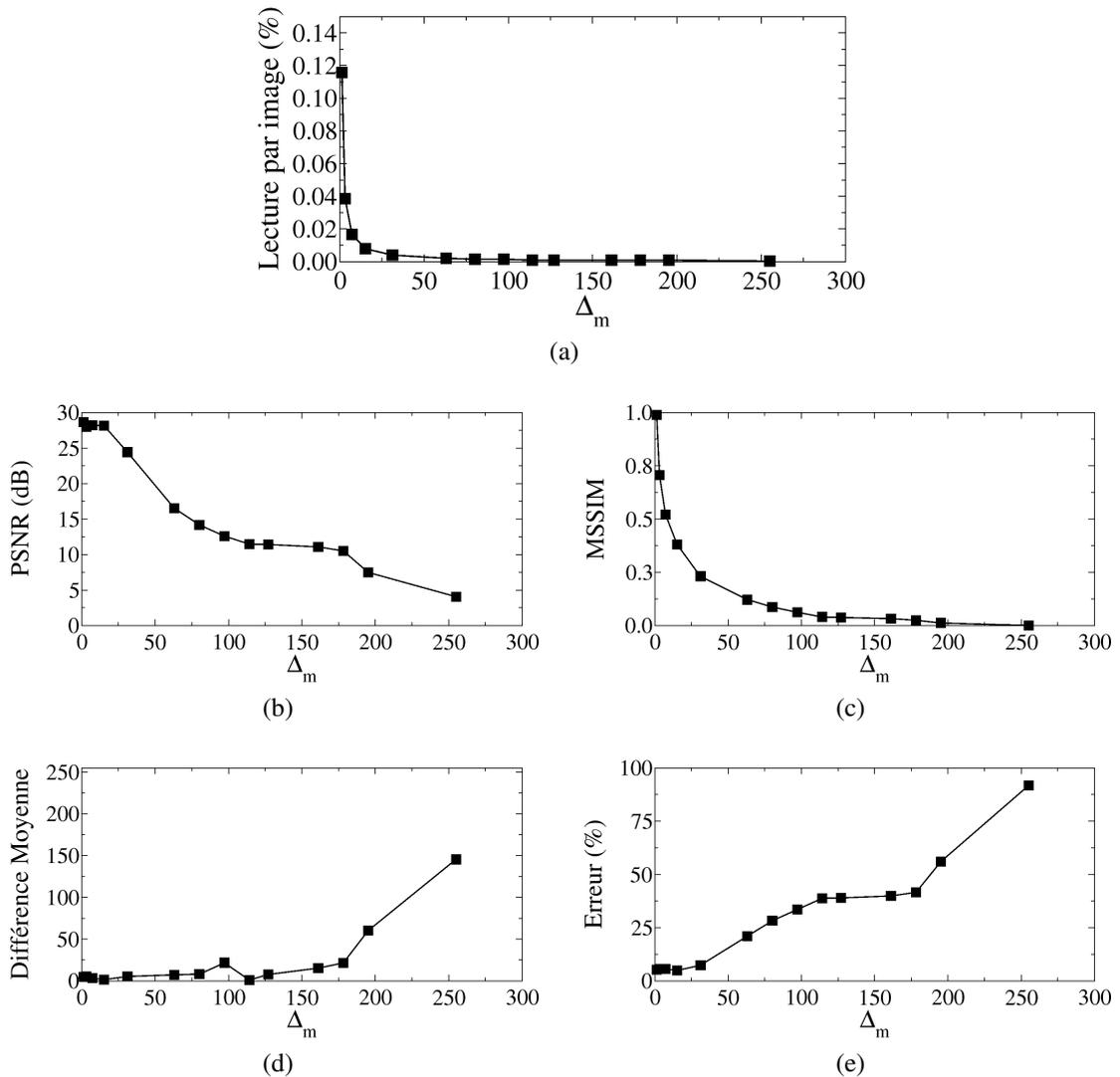


**FIGURE 7.8** – Évaluation de la qualité des images résultantes de simulations de l'image A (Fig.7.4) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .

On considère que le MSSIM, utilisé simultanément avec les approches d'évaluation traditionnelles comme le PSNR et le MSE, apporte une bonne analyse de la méthode de compression et une évaluation de images compressées.

A la fin des simulations, on obtient le MSSIM de chaque image (A, B, C et D) en fonction du coefficient de compression  $\Delta_m$ .

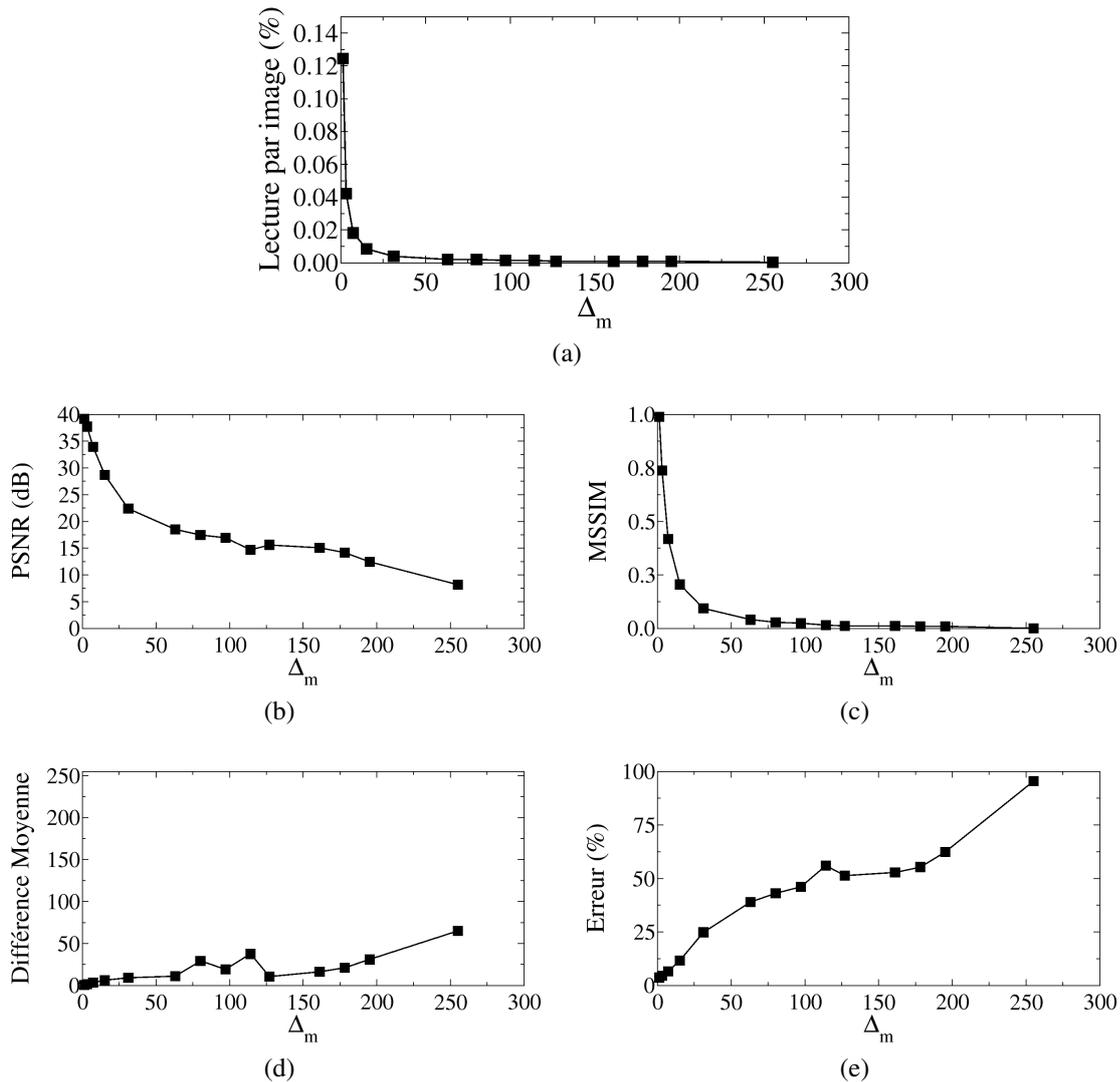
Les images compressées de l'image A ont maintenu un MSSIM supérieur à 0.7 (AISRR =



**FIGURE 7.9** – Évaluation de la qualité des images résultantes de simulations de l'image B (Fig.7.5) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .

0.02%, PSNR = 36.29 dB) pour  $\Delta_m = 8$ . Pour cette image (Figure 7.4d), même si une partie (le ciel) de l'image commence à montrer une distorsion, les détails de l'image demeurent visibles et distincts.

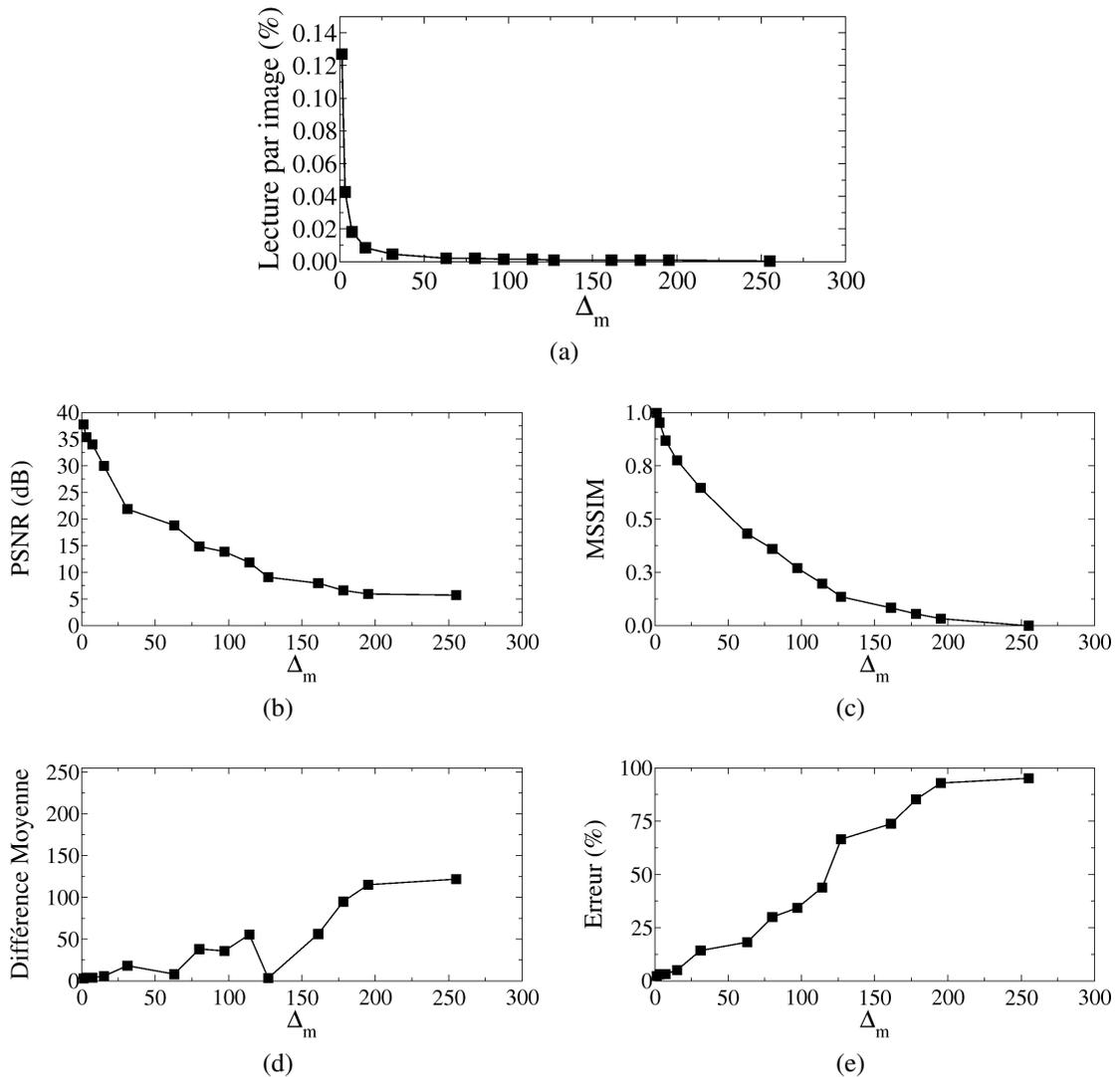
Cependant, le MSSIM des images associées à l'image de référence B diminuent en fonction de  $\Delta_m$  (Figure 7.9c) d'une manière plus rapide que ceux de l'image A. Le MSSIM atteint 0.707 pour  $\Delta_m = 4$  (AISRR = 0.03%, PSNR = 28.02 dB) (Figure



**FIGURE 7.10** – Évaluation de la qualité des images résultantes de simulations de l'image C (Fig.7.6) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural SIMilarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .

7.5c). Mais même pour l'image compressée avec un  $\Delta_m = 8$  (Figure 7.5d), qui a MSSIM égal à 0.521 et un PSNR de 28.25 dB, les détails de l'image sont clairs et elle a une bonne qualité visuelle .

Les images compressées associées à la troisième image de référence C, maintiennent une bonne qualité visuelle jusqu'à  $\Delta_m = 8$  même avec un MSSIM d'une valeur relativement basse de 0.419 et un PSNR de 33.98 dB. La décrémentation rapide des valeurs



**FIGURE 7.11** – Évaluation de la qualité des images résultantes de simulations de l'image D (Fig.7.7) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural SIMilarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .

MSSIM en fonction de  $\Delta_m$  (Figure 7.10c) est justifiée par le fait que le paramètre SSIM est fortement sensible aux contours [RH08]. En effet, comme on l'a déjà mentionné, l'augmentation de la valeur de  $\Delta_m$ , fusionne les niveaux de gris adjacents ce qui introduit des distorsions au niveau de l'image résultante et cause la création des nouveaux contours dans l'image comme les deux images 7.6d, 7.6e et 7.6f le montre. Alors pour l'image de référence C, une compression avec un  $\Delta_m = 2$  et  $\Delta_m = 4$ , nous fournissent

des images de bonne qualité, avec un PSNR supérieur à 37 dB et un AISSR inférieur à 0.13% pour les deux images résultantes et un MSSIM de 0.99 et 0.739 respectivement.

Les images compressées issues de l'image de référence D maintiennent une bonne qualité visuelle et un MSSIM supérieur à 0.75 ainsi qu'un PSNR supérieur à 30 dB jusqu'à  $\Delta_m = 16$ . Pour cette valeur de  $\Delta_m$ , les détails de l'image résultante (Figure 7.7e) demeurent clairs et intacts. En fait, MSSIM (Figure 7.11c) diminue, en fonction du  $\Delta_m$ , d'une manière plus lente que pour les autres images résultantes. Cela est dû au fait que la majorité des parties de l'image D ont un contraste élevé. Ces parties sont les moins affectées par la méthode de réduction de redondances spatiales vu que leurs pixels n'appartiennent pas à des niveaux de gris adjacents ce qui évitent la création des distorsions causées par le fusionnement des niveaux de gris. Alors pour l'image de référence D, une compression avec un  $\Delta_m = 2$ ,  $\Delta_m = 4$  et  $\Delta_m = 8$ , nous fournissent des images de bonne qualité, avec un PSNR supérieur à 34 dB et un AISSR inférieur à 0.13% pour les deux images résultantes et un MSSIM de 0.998, 0.953 et 0.869 respectivement.

Finalement, pour évaluer l'effet de notre méthode de réduction des redondances spatiales, on observe la différence moyenne et l'erreur (%) entre l'image de référence et chaque image compressée. Les valeurs de ces deux paramètres augmentent proportionnellement à  $\Delta_m$ . La différence moyenne qui varie entre 0 (cas idéal) et 255 (pire cas), est calculée pour chaque image résultante en fonction de  $\Delta_m$ . De même pour l'erreur introduite par la compression qui garde une valeur raisonnable inférieure à 10% pour des valeurs  $\Delta_m$  allant jusqu'à 16 pour l'image A (figure 7.8e) et D (figure 7.11e) et jusqu'à 32 pour l'image B (figure 7.9e) et et un  $\Delta_m$  allant jusqu'à 8 pour l'image C (figure 7.10e).

## 7.4 Conclusion

Dans ce chapitre, on a présenté les simulations de haut niveau effectuées dans le but de valider le protocole de lecture et la méthode de réduction des redondances spatiales proposés dans cette thèse. Les simulations proposées utilisent des images numériques originalement capturées par des imageurs numériques conventionnels, ainsi que des images numériques standards dans le domaine de l'imagerie (Annexe A), afin de créer un flot de données événementiel et d'émuler le fonctionnement du système de lecture.

En premier lieu, l'effet du protocole de lecture événementiel est observé à travers le taux de lecture nécessaire pour l'extraction de l'information de chaque image. Ce dernier est calculé en considérant que le taux de lecture de l'image de référence est le pire cas. En effet, le taux de lecture nécessaire pour le système de lecture proposé variant en fonction du coefficient de compression  $\Delta_m$  a maintenu une valeur inférieure à 1% du taux de lecture des imageurs conventionnels. Ce résultat valide l'efficacité de notre méthode de lecture événementielle pour les différentes scènes capturées. Ensuite, l'im-

Le pact de la réduction des redondances spatiales sur les images résultantes est évalué au travers des paramètres d'évaluation de qualité, habituellement utilisés pour l'évaluation de images après une compression de données comme le PSNR, le MSSIM, la différence moyenne et l'erreur entre les images résultantes et l'image de référence. Les images de référence utilisées proposent le même type de résultats. De plus, les valeurs de ces paramètres sont indiqués pour des faibles valeurs de  $\Delta_m$  (en moyenne inférieur à 25), la méthode de compression de données proposée a généré des images nécessitant un faible taux de lecture mais d'une qualité proche de celle des images de références. La tendance globale des résultats obtenus valide le protocole de lecture proposé ainsi que la méthode de réduction des redondances spatiales.

# **Troisième partie**

## **Design**



# 8

## CONCEPTION DU CIRCUIT DE LECTURE ASYNCHRONE

### Sommaire

---

|         |   |            |
|---------|---|------------|
| 8.1     | Système de lecture . . . . .  | <b>107</b> |
| 8.1.1   | Encodeur - Bloc de vérification . . . . .                           | 107        |
| 8.1.1.1 | Automate à états finis . . . . .                                    | 109        |
| 8.1.1.2 | Automate à états finis asynchrone . . . . .                         | 110        |
| 8.1.2   | Logique combinatoire . . . . .                                      | 112        |
| 8.1.3   | Mémoire tampon : FIFO . . . . .                                     | 112        |
| 8.1.4   | Simulation VHDL . . . . .   | 113        |
| 8.2     | Simulation au niveau netlist . . . . .                              | <b>115</b> |
| 8.2.1   | Simulation MATLAB : Création des requêtes de lecture . . . . .      | 117        |
| 8.2.2   | Simulation VHDL niveau netlist . . . . .                            | 119        |
| 8.2.3   | Simulation MATLAB : Reconstruction des images résultantes . . . . . | 119        |
| 8.2.4   | Résultats . . . . .   | 119        |
| 8.3     | Architecture du pixel événementiel . . . . .                        | <b>122</b> |
| 8.4     | Conclusion . . . . .  | <b>123</b> |

---

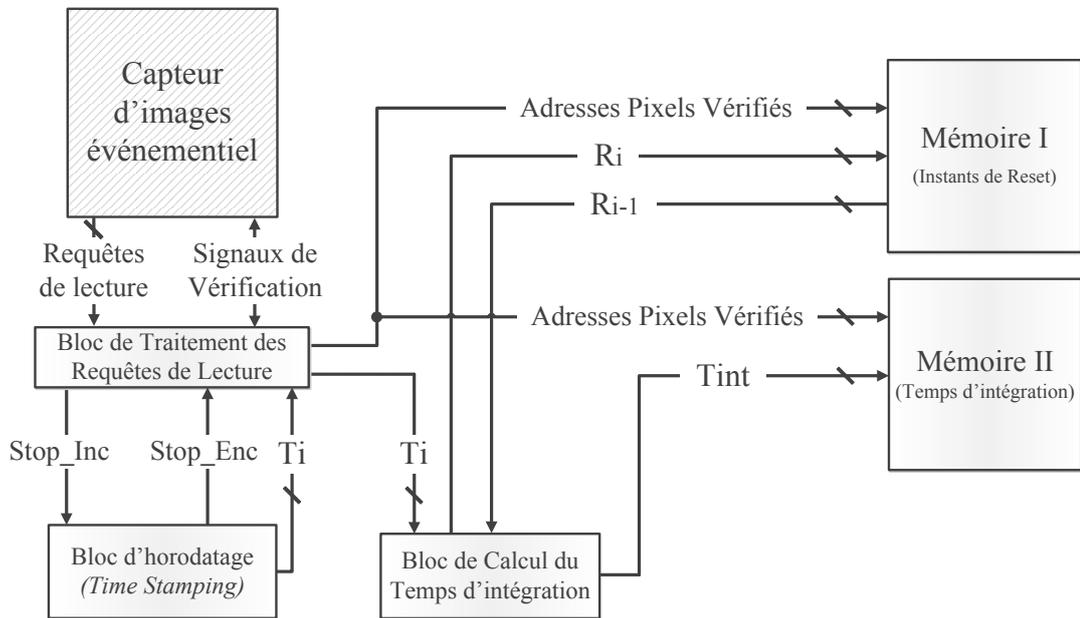
Dans ce chapitre, on reprend l'architecture du système de lecture du capteur d'images présenté au cours du chapitre 6 dans le but d'expliquer la conception et la synthèse en logique asynchrone de ce dernier.

## 8.1 Système de lecture

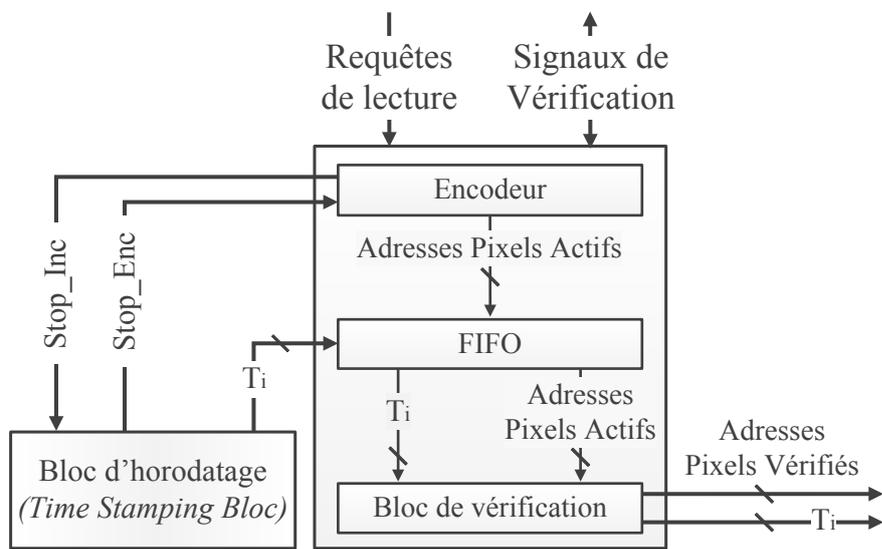
Le système de lecture présenté au cours du chapitre 6 (Figure 8.1) est modélisé grâce au langage de description de matériel : VHDL, puis synthétisé. Cependant, dans ce manuscrit, on se limite à l'explication de la conception du bloc de traitement des requêtes de lecture.

### 8.1.1 Encodeur - Bloc de vérification

Le fonctionnement de l'encodeur et du bloc de vérification du bloc de traitement des requêtes de lecture sont décrits comme étant des automates à états finis synchrones. En



(a)



(b)

**FIGURE 8.1** – Architecture du système de lecture. (a) L'image de référence. (b) Bloc de traitement des requêtes de lecture

principe, le modèle d'un automate à états finis est utilisé afin de traduire la séquentialité de fonctionnement. Dans le cas de notre architecture, le fonctionnement séquentiel est présent au niveau de l'encodeur (la phase de réception des requêtes de lecture et la phase de suppression des redondances spatiales) et au niveau du bloc de vérification (la phase de vérification). Cette approche de description par machine à états est alors adoptée pour la conception de ces deux sous-blocs, vu qu'elle est une méthode d'implémentation simple et facilement ajustable pour les circuits asynchrones. De plus, le modèle d'un automate à états finis est facile à synthétiser.

### 8.1.1.1 Automate à états finis

Un automate à états finis, également connu sous le nom de machine à états finis (*Finite State Machine : FSM*), est un modèle de calcul utilisé pour représenter un algorithme ou un système électronique. Dans notre cas, des automates à états finis sont employés pour modéliser le fonctionnement séquentiel de certains blocs du système de lecture numérique. Une machine à états finis est caractérisée par son nombre d'états ( $S_i$ ) qui est déterminé selon le fonctionnement désiré. Le passage d'un état actif, dit état courant (*Current State*) et l'état suivant (*Next State*) est contrôlé par des transitions ( $T_i$ ) calculées selon l'état des entrées ( $X_i$ ) de la machine. On distingue deux types principaux d'automate à états finis : la machine de Mealy et la machine de Moore. Dans le cas d'une machine de Mealy, les transitions ( $T_i$ ) et les sorties ( $Y_i$ ) dépendent de l'état courant et des entrées ( $X_i$ ). En d'autres termes, l'état de la sortie est déterminé suivant les transitions. En revanche, les sorties d'une machine de Moore dépendent uniquement de l'état courant de la machine. En fait, toutes les sorties de la machine sont affectées pour chaque état courant. En comparaison, la machine de Moore a l'avantage de la simplicité et la facilité de compréhension mais l'utilisation de la machine de Mealy implique la réduction du nombre d'états nécessaire.

De point de vue électronique, une machine à états finis est composée, comme le montre la figure 8.2, d'un bloc combinatoire (*Combinational Logic*) qui effectue le calcul et le traitement de données nécessaire et qui détermine l'état suivant de la machine et les sorties selon les entrées, et d'un bloc séquentiel qui assure la séquentialité de la machine et la transition entre les états conditionnée par la valeur des entrées de la machine. Pour, une machine synchrone, la partie séquentielle est basée sur un pipeline synchrone composé de cellules de mémorisation et contrôlé par un signal d'horloge (Figure 8.3). Pour ce pipeline synchrone, l'arrivée du front d'horloge provoque le déplacement des données d'un registre au suivant. Étant contrôlé par un signal d'horloge, ce pipeline souffre des mêmes limitations des circuits synchrones comme la vitesse limitée et la consommation électrique qui augmente proportionnellement à la fréquence de fonctionnement.

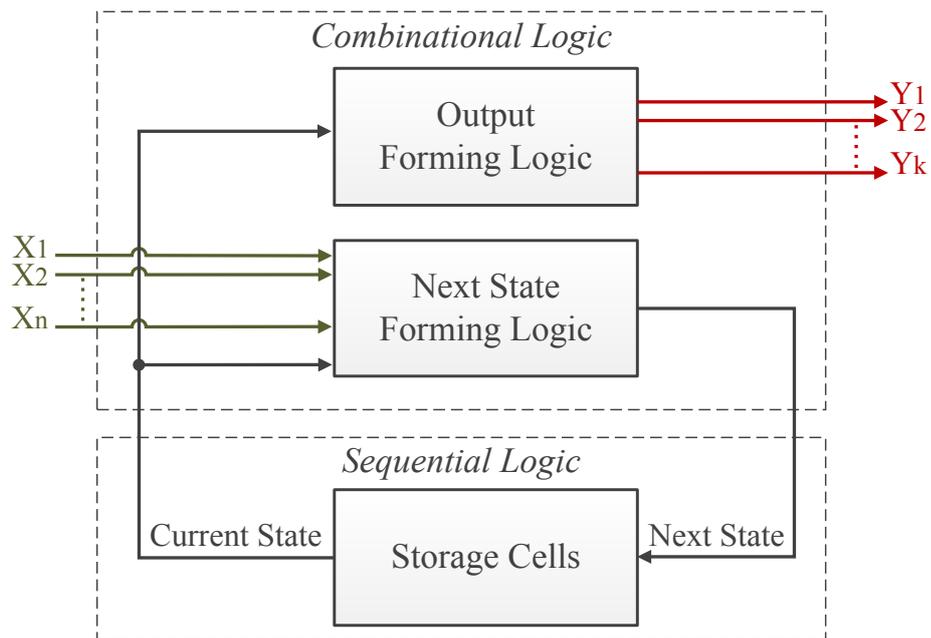


FIGURE 8.2 – Modèle d'une machine à états finis de type Moore

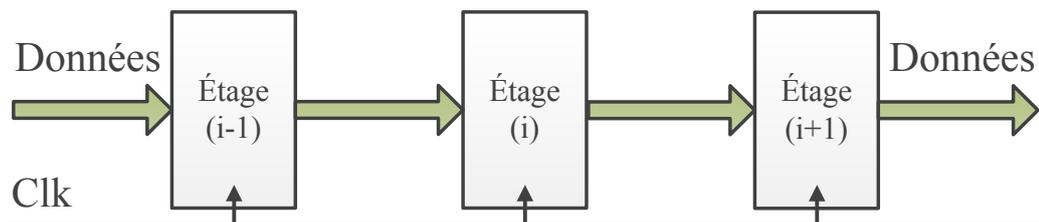


FIGURE 8.3 – Pipeline synchrone à trois étages

### 8.1.1.2 Automate à états finis asynchrone

Notre capteur d'images étant événementiel et totalement asynchrone, les blocs du système de lecture modélisés en des automates conservent leur aspect asynchrone en faisant appel à des machines à états finis asynchrones (*Asynchronous Finite-State Machine : A-FSM*). L'automate asynchrone conserve le même modèle de la machine synchrone de type Moore (Figure 8.2). Cependant, le pipeline séquentiel synchrone de la machine est remplacé par un circuit micropipeline asynchrone linéaire [Sut89b] (une entrée, une sortie) illustré par la figure 8.4. Les signaux de requêtes indiquent la disponibilité de données valides à un étage du pipeline tandis que l'état des signaux d'acquittement détermine si l'étage est libre et capable de recevoir de nouvelles données. Ce type de synchronisation qui dépend fortement des données accorde à ce circuit micropipeline

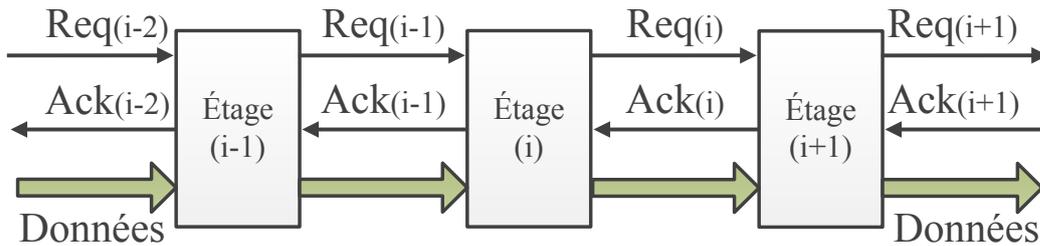


FIGURE 8.4 – Pipeline asynchrone à trois étages

l'avantage d'un fonctionnement adapté aux données reçues. Ceci assure une faible activité et une faible consommation électrique. Un pipeline asynchrone linéaire est caractérisé par son nombre d'étages et le nombre de jetons et de bulles. On note qu'un jeton représente une donnée valide tandis qu'une bulle représente une ancienne donnée, déjà propagée, et qui peut être remplacée. On considère, alors, que les bulles permettent aux jetons de se déplacer dans le pipeline asynchrone. Le pipeline asynchrone doit avoir une ou plusieurs bulles, sinon il sera dans un état bloqué (*Locked state*). Ceci implique que des bulles doivent être introduites lors de l'initialisation du circuit.

Les étages, en d'autres termes, les blocs élémentaires, du pipeline asynchrone sont essentiellement composés d'un élément de stockage de données (un latch) qui assure aussi le déplacement des données d'un étage au suivant et d'un sous-bloc de contrôle qui établit la synchronisation locale et la communication entre les différents étages. Au cours de la conception de notre circuit, on a utilisé le protocole WCHB (*Weak Condition Half-Buffer*). L'étage de micropipeline de type WCHB, illustré par la figure 8.5, est dit *half-buffer* parce qu'il ne peut avoir deux jetons distincts sur son entrée et sa sortie. Au départ, après l'initialisation, les signaux de requêtes sont au niveau bas et les signaux d'acquiescement sont au niveau haut. Suite à la réception d'une requête (ReqIn passe au niveau haut), la sortie de la porte de Müller passe au niveau actif afin de déclencher le fonctionnement du latch. Ce signal, retardé par un délai équivalent au temps nécessaire de déplacement des données de l'entrée vers la sortie du latch, est le signal de requête en sortie de cet étage envoyé à l'étage suivant. Simultanément à l'envoi de la requête en sortie, le signal d'acquiescement AckIn passe à son état actif afin de libérer l'étage précédent.

Un des codages des différents états d'un automate est le codage 1 parmi  $N$  (*One-Hot Encoding*). En utilisant ce codage, chaque état différent est représenté par un seul bit des signaux de l'état courant (*Current State*) et de l'état suivant (*Next State*). Ce codage rend la logique de calcul de l'état suivant simple et diminue les aléas normalement introduits par d'autre type d'encodage. Ce type de codage peut s'avérer désavantageux pour des automates complexes ayant un nombre d'états élevé, ce qui augmente la taille des signaux de l'état courant et de l'état suivant. Cependant, les automates de l'encodeur et

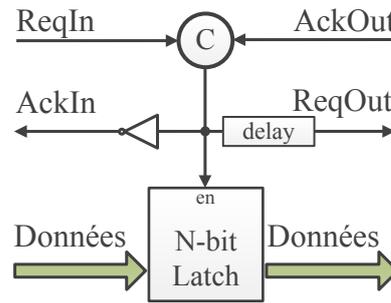


FIGURE 8.5 – Étage d'un micropipeline WCHB linéaire

du bloc de vérification étant simples avec un nombre d'états modeste, le codage *one-hot* s'avère conforme à nos besoins.

### 8.1.2 Logique combinatoire

La logique combinatoire de chacun des deux automates finis dépend essentiellement de leur fonctionnement, détaillé au cours des chapitres précédents. La logique combinatoire suit donc exactement le fonctionnement ainsi que les conditions selon laquelle, l'automate passe d'un état au suivant.

La partie combinatoire de la machine à états finis est synthétisée en utilisant les outils de conception assistée par ordinateur ou en convertissant manuellement le modèle décrit au niveau RTL (Register Transfer Level) en une *netlist* associée au circuit.

### 8.1.3 Mémoire tampon : FIFO

La mémoire tampon utilisée est une mémoire de type FIFO Dual nécessaire pour sauvegarder les adresses des pixels actifs ainsi que l'information temporelle associée. En fait, la FIFO asynchrone est un pipeline asynchrone linéaire similaire à celui des automates décrits précédemment. Le transfert des données dans la FIFO asynchrone se fait par une signalisation requête/acquittement entre les étages. Si l'étage courant n'émet aucun signal de requête en aval, un blocage des données en amont est implicitement imposé. Ceci est assuré par l'attente du signal d'acquittement généré par l'étage courant, qui permet de libérer et d'informer l'étage en amont de la réception des données. Par conséquent, une bulle est créée entre les étages. Aucune nouvelle donnée ne peut donc être transférée par l'étage en amont tant qu'il n'a pas reçu l'acquittement de l'étage courant. De même, l'étage en aval reste bloqué tant que l'acquittement qu'il a envoyé n'a pas été reçu par l'étage courant. En d'autres termes, la requête de l'étage courant est toujours en mode actif. Ainsi, les données progressent dans la FIFO indépendamment

des données qui les suivent et tant qu'elles ne rencontrent pas un étage occupé. Ce verrouillage implicite en amont et en aval fournit la propriété intéressante de blocage local du pipeline asynchrone. L'avantage de ces pipelines asynchrones est la faible variation du courant lors des commutations et de la possibilité d'utiliser des blocages transparents sans réduire la vitesse du flot de données [Rez04]. De plus, les performances de la FIFO peuvent être améliorées en augmentant le nombre de latches dans la FIFO, ce qui implique l'augmentation des bulles et par la suite le nombre de données transférées simultanément.

L'architecture de la mémoire tampon implémentée dans notre circuit comportent deux sous-blocs essentiels en dehors de la FIFO. Le premier sous-bloc (*Write Protection*) permet de respecter l'ordre d'écriture implicitement imposé par le fonctionnement proposé et assure, par la suite, une écriture alternée entre les deux FIFOs, celle des adresses des pixels actifs et celle des instants d'occurrence des requêtes de lecture. Ainsi, les adresses des pixels actifs (APA) et l'information temporelle ( $T_i$ ) se propagent dans les mémoires tampons et se trouvent alors au même niveau dans les FIFOs. Par contre, le sous-bloc (*Read Protection*), qui reçoit le requête de lecture (*Read*), permet la lecture des FIFOs que si les deux données (APA,  $T_i$ ) se trouvent au même étage de la mémoire. Une fois que les mémoires passent en mode lecture, ce sous-bloc assure une lecture simultanée, envoyant au bloc de vérification, les adresses des pixels actifs ainsi que l'information temporelle associée.

#### 8.1.4 Simulation VHDL

Afin de valider le circuit conçu en logique asynchrone et de visualiser le fonctionnement au niveau circuit, on le simule en utilisant un testbench (VHDL) avec le simulateur de MODELSIM de Mentor. Le testbench est modélisée dans le but de se comporter comme le pixel événementiel analogique. En d'autres termes, le testbench réagit avec le circuit de lecture au niveau de l'envoi et la réception des signaux du pixel événementiel. On présente ici le chronogramme qui illustre la signalisation entre les composants du bloc de traitement des requêtes de lecture. Le chronogramme de la figure 8.7a montre la réception des requêtes, les deux demandes d'écriture dans la mémoire tampon et de la phase de vérification. La simulation est lancée par un signal d'initialisation global. Ensuite, une éventuelle détection d'événements sur des pixels se traduit par une activité sur les deux signaux de requêtes :  $RowReq = "1110"$  et  $ColReq = "1100"$ . Ceci veut dire qu'il existe six pixels qui apparaissent comme étant actifs. Suite à la réception des requêtes, le bloc chargé du codage des adresses des pixels actifs envoie le signal  $PxReq$  (c'est le signal *StopInc* dans la figure 8.1). A la réception de ce signal, le bloc d'horodatage est chargé d'exécuter trois fonctions. La première fonction consiste à arrêter l'incréméntation du compteur de la conversion tension-temps. Tandis que la deuxième fonction consiste à assurer l'intervalle de temps associé à la méthode de réduction de redondances spatiales ( $\Delta_t$ ). La dernière fonction est l'écriture de l'instant d'occurrence

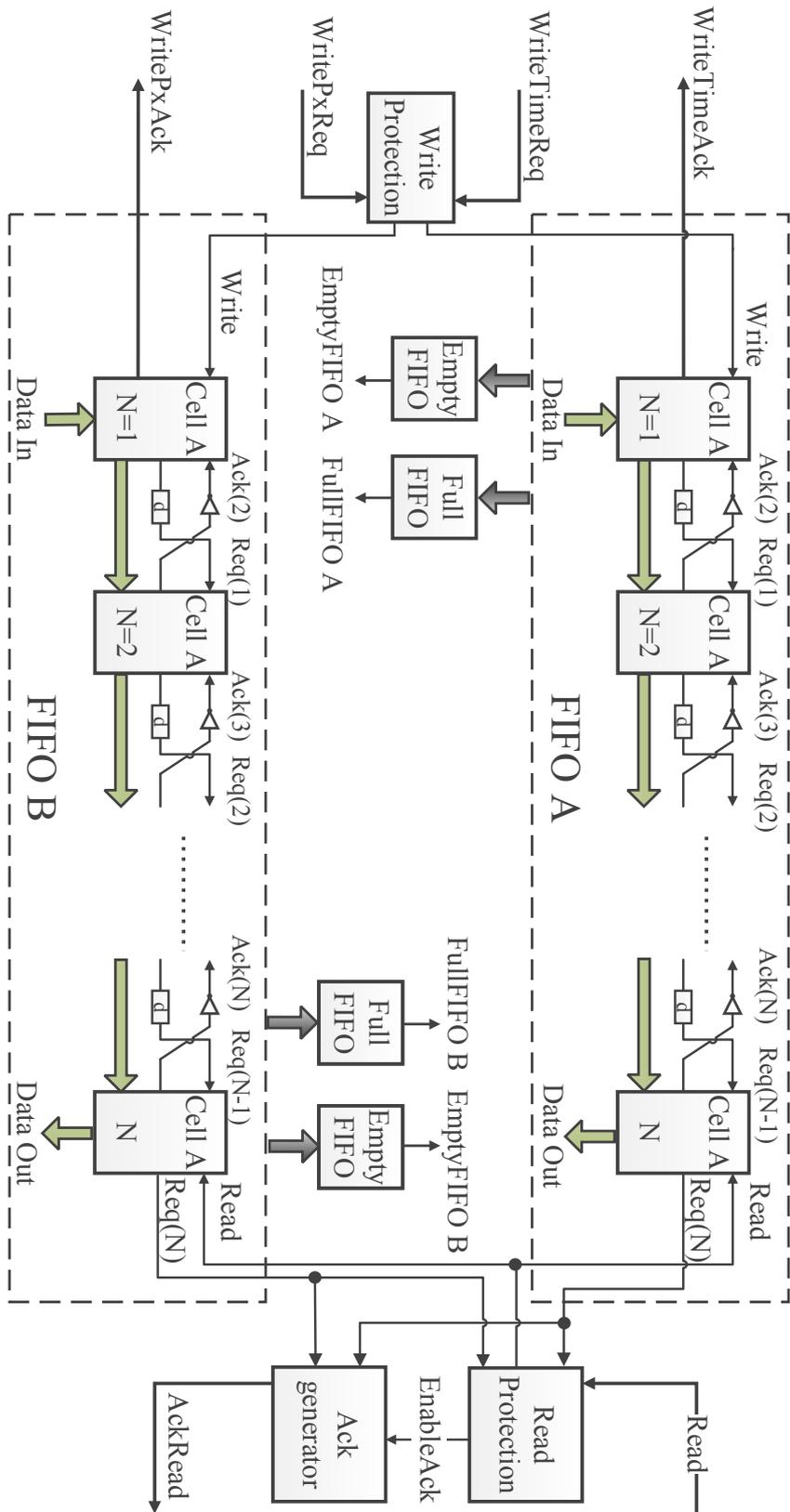


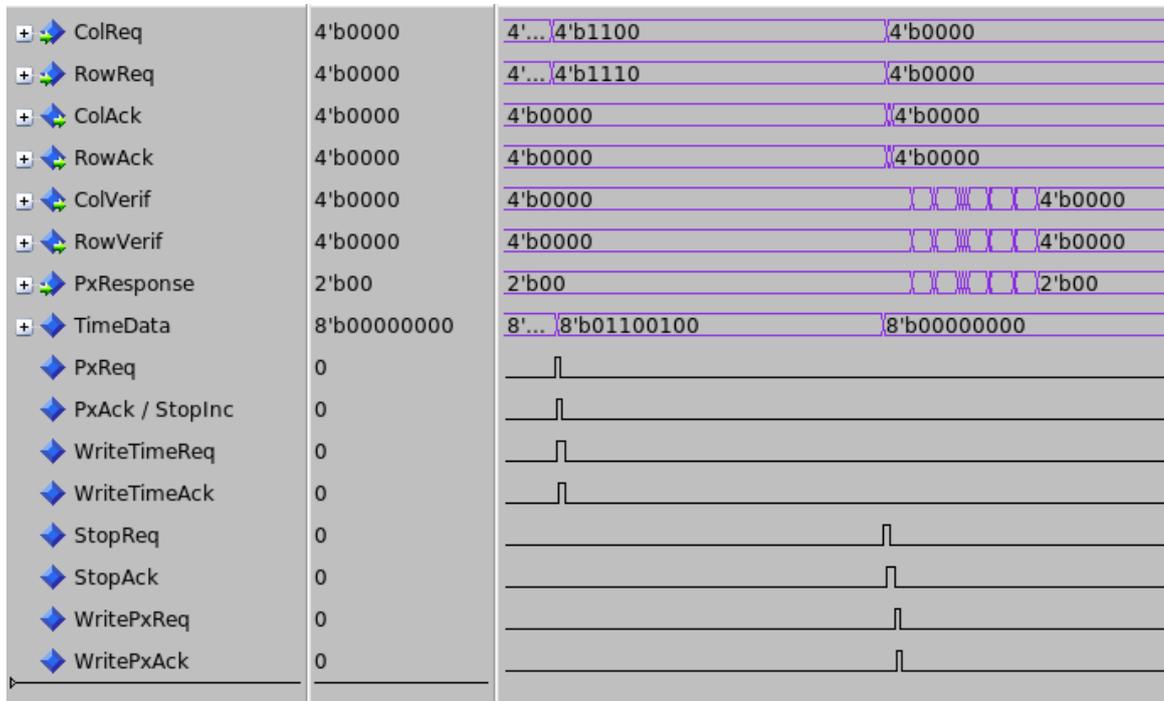
FIGURE 8.6 – Architecture de la mémoire tampon de type FIFO

de la première requête dans la mémoire tampon (FIFO). La signalisation entre le bloc d'horodatage et la mémoire tampon est établie à travers les deux signaux *WriteTimeReq* et *WriteTimeAck*. La fin de la phase de réduction des redondances spatiales ( $\Delta_t$  est écoulé) est marqué par l'envoi du signal *StopReq* (c'est le signal *StopEnc* dans la figure 8.1) à l'encodeur, qui, par la suite, arrête la réception des requêtes de lecture, acquitte les pixels actifs (*RowReq* et *ColAck*) et demande la permission de lecture des adresses des pixels actifs (APA) dans la mémoire tampon au travers des signaux *WritePxReq* et *WritePxAck*. A cet instant, l'encodeur est libéré et est capable de recevoir de nouveau des requêtes de lecture tandis que le bloc de vérification commence la vérification séquentielle des pixels actifs. Le chronogramme de la figure 8.7b est une vue élargie de la phase de vérification de l'exemple décrit ci dessus. Le circuit de lecture envoie séquentiellement aux pixels la demande de vérification au travers des signaux *RowVerif* et *ColVerif*. Suite à la vérification, le pixel événementiel répond selon la valeur de sa requête locale. L'état du pixel est transféré au système de lecture par le signal *PxRespns* codé sur 2 bits, avec "10" et "01" étant les valeurs qui indiquent respectivement si le pixel est actif ou pas.

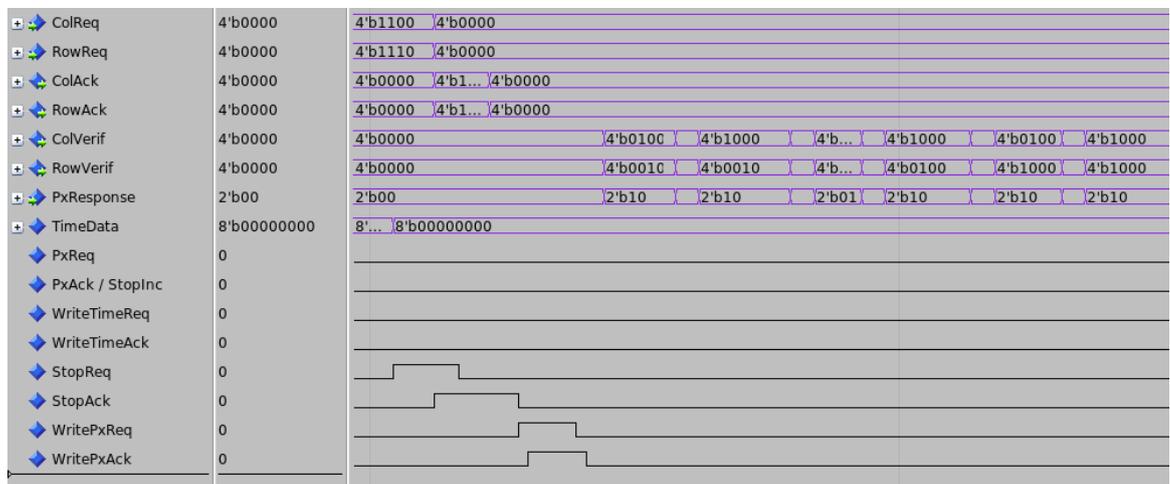
Ce testbench montre que le circuit de lecture suit fidèlement le fonctionnement du système expliqué et proposé dans le chapitre 6.

## 8.2 Simulation au niveau netlist

Dans le but de créer un stimuli similaire au flot de données en sortie du capteur d'images événementiels, et par la suite, de simuler le circuit numérique de lecture, on utilise des images prises par des appareils photos numériques conventionnels. Pareillement, à la méthode du chapitre 7, ces images sont considérées comme étant des images de référence (Figure 8.9). Afin de générer le stimuli convenable pour le circuit de lecture, on fait appel à MATLAB, pour extraire l'information d'une image de référence et créer un flot de données événementiel identique à celui en sortie du capteur d'images événementiel proposé et en se référant aux valeurs des pixels de l'image de référence. Les requêtes de lecture générées par MATLAB sont traitées par le modèle décrit au niveau *netlist* du circuit de lecture sous MODELSIM, le simulateur de MENTOR GRAPHICS, suivant le débit de traitement décrit dans le chapitre 6. A la fin de chaque itération, le modèle du circuit de lecture renvoie la mémoire qui contient les temps d'intégration de tous les pixels (ITVM) contenant les différentes valeurs de temps d'intégration des pixels actifs à MATLAB qui reconstruit l'image compressée. La figure 8.8 illustre le flot de simulation du circuit de lecture asynchrone par une image numérique de référence. Dans ce qui suit, on explique les différentes étapes de simulation.

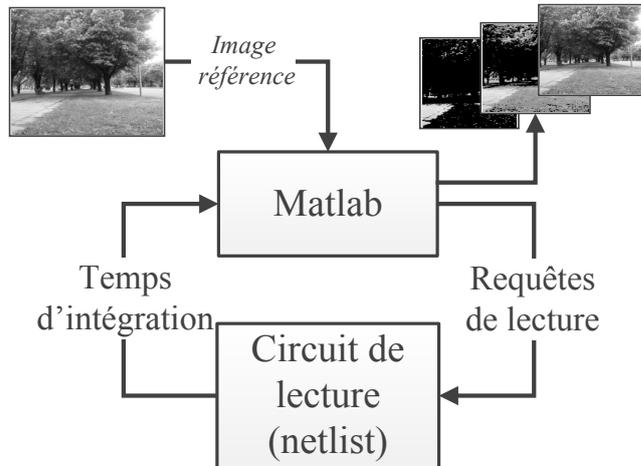


(a)



(b)

**FIGURE 8.7** – Chronogramme du système de lecture. (a) La signalisation entre les composantes du bloc de traitement des requêtes de lecture et le bloc d’horodatage (b) Vue élargie de la phase de vérification



**FIGURE 8.8** – Représentation du flot de simulation du circuit de lecture asynchrone par des images numériques

### 8.2.1 Simulation MATLAB : Création des requêtes de lecture

Pour expliquer le protocole de simulation, on considère l'image 8.9h comme l'image numérique de référence. Dans le but de générer les requêtes de lecture nécessaire pour la simulation du circuit de lecture, on se base sur le fait que l'intégration de notre pixel événementiel est linéaire. Sous MATLAB, cela nous permet de traduire l'intégration du pixel en une décrémentation à un pas uniforme (IntStep). Pendant cette décrémentation, la valeur de chaque pixel est comparée à une valeur de seuil ( $P_{th}$ ). Cet algorithme de génération de requêtes crée un set de stimuli, non seulement pour le circuit de lecture concernant la réception et le traitement des demandes de pixels, mais également pour la méthode de réduction des redondances spatiales.

#### 1. Initialisation

- a) Lire l'image de référence : RefIm
- b) Choisir le pas d'intégration des pixels : IntStep

#### 2. Itérations : $1 \rightarrow K$

- a)  $PicSimu = PicSimu - IntStep$
- b) Vérifier si  $PicSimu(i,j) \Big|_{\substack{i = 1, \dots, m \\ j = 1, \dots, n}} = P_{th} + \Delta_m$

Au cours de l'étape 2b de l'algorithme décrit ci-dessus, chaque valeur de pixel est comparée à une valeur seuil qui définit le moment où la requête de lecture doit être envoyée. En conséquence, les pixels atteignant  $P_{th}$  pendant la même itération, sont regroupés et sont associés à la même itération. En d'autres termes, les pixels dont la valeur initiale identique seront envoyées et traitées ultérieurement par le système de lecture pendant



**FIGURE 8.9** – Les images de référence utilisées pour les simulations du circuit de lecture niveau *netlist*

la même itération. Par conséquent, avec cet algorithme on suit fidèlement le fonctionnement de notre capteur d'images en limitant l'activité du système de lecture non seulement aux pixels actifs, mais aussi à une lecture pour groupe de pixels actifs.

### 8.2.2 Simulation VHDL niveau netlist

A ce niveau, le circuit de lecture traite toutes les requêtes de lecture suivant leur ordre d'envoi. En absence de requêtes de lecture, le système de lecture demeure en mode inactifs. En raison de la taille et de la complexité des sorties de la simulation VHDL, le chronogramme associé à ces simulations n'est pas présenté. Après le traitement de chaque groupe de requêtes de lecture, la valeur calculée du temps d'intégration de chaque pixel, est sauvegardée dans la mémoire ITVM. A la fin de chaque itération, le contenu de cette mémoire est transférée à MATLAB afin de reconstruire l'image résultante.

### 8.2.3 Simulation MATLAB : Reconstruction des images résultantes

Au cours de l'étape de reconstruction des images, MATLAB reçoit une mise à jour de la mémoire ITVM après chaque itération de lecture. En d'autres termes, MATLAB traite les groupes des pixels actifs au fur et à mesure de leur réception. Ce fonctionnement événementiel est visualisé par la reconstruction des images, dites intermédiaires, qui représentent la mémoire ITVM contenant les différents temps d'intégration. On considère que ces images sont intermédiaires vu qu'elles illustrent l'ordre d'envoi des requêtes de lecture. En d'autres termes, les pixels apparaissent séquentiellement dans l'image reconstruite en fonction de leurs valeurs dans l'image de référence. Les pixels ayant un court temps d'intégration (les pixels proche du blanc) sont donc les premiers à apparaître dans l'image reconstruite. Suivant les résultats de simulation de la section 8.2.4 et ceux de l'annexe B, ces pixels ont été les premiers à envoyer des requêtes de lecture, et donc les premiers à être traités par le circuit numérique. Bien évidemment, les pixels ayant la même valeur dans l'image de référence (à  $\Delta_m$  près), sont traités dans le même groupe de pixels actifs et par la suite apparaissent dans les images intermédiaires au cours de la même itération (voir figure B.8). Ceci prouve que notre méthode de réduction des redondances spatiales est efficace. La simulation du circuit numérique (en VHDL) montre des résultats identiques à ceux observés avec les simulations haut niveau présentées au cours du chapitre 7.

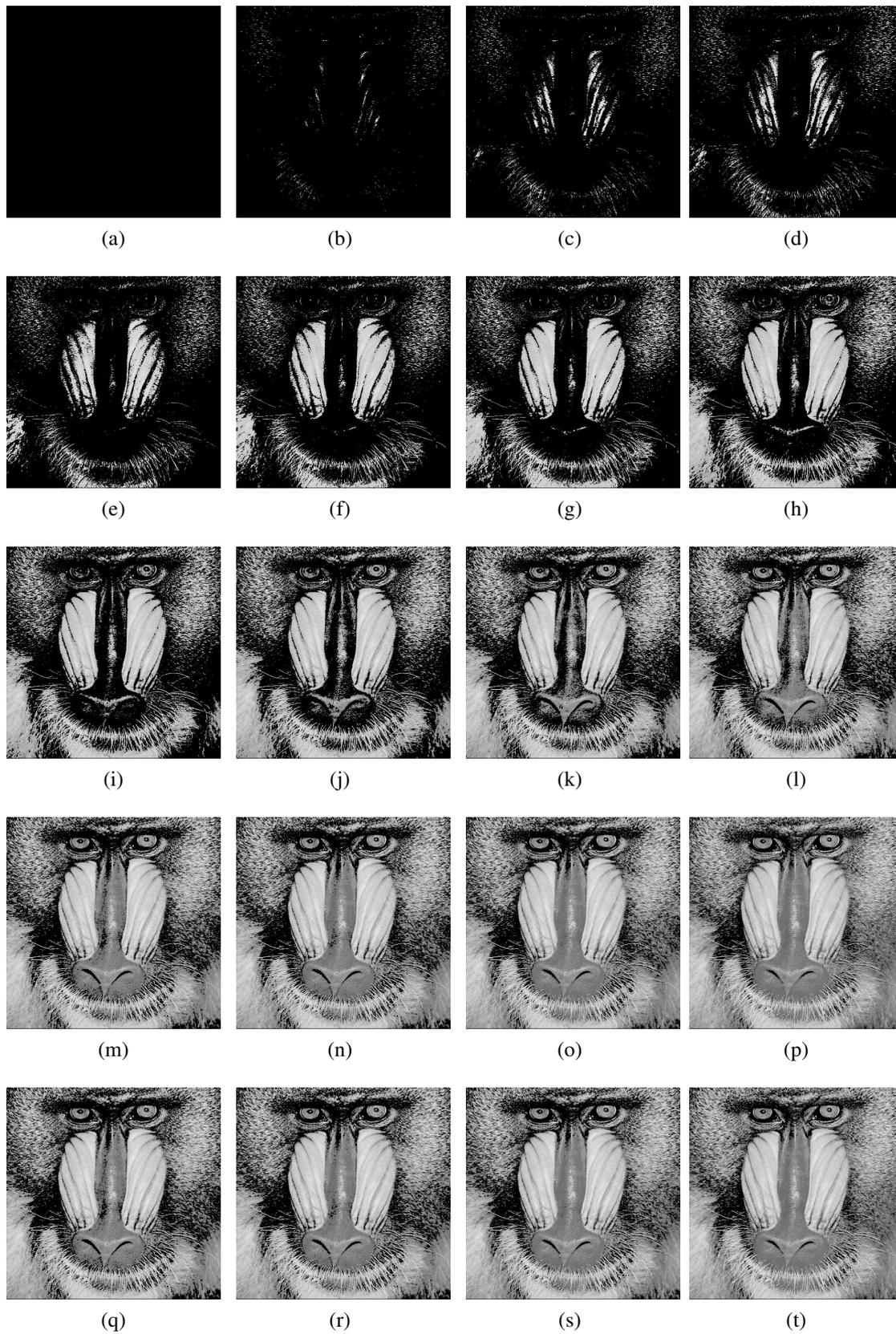
### 8.2.4 Résultats

Le tableau 8.1 regroupe les résultats de simulation des images de référence (Figure 8.9). On note que les images intermédiaires résultantes issues du reste des images de référence sont présentés dans l'annexe B. Afin d'évaluer la qualité de l'image résultante, on utilise les mêmes paramètres de qualité usuels que pour les simulations du

chapitre 7. Ces paramètres, notamment le taux de lecture par rapport au taux de lecture de l'image référence, le PSNR, le MSSIM et l'erreur entre l'image compressée et l'image de référence, nous permettent d'estimer l'effet d'une part de la lecture événementielle simultanée des pixels actifs, et d'autre part de l'effet de la réduction des redondances spatiales. Pour les simulations au niveau netlist, on exige une erreur inférieure à 10 % pour un  $\Delta_m = 2$  fixe pendant ces simulations. Les résultats obtenus pour les images compressées indiquent que notre méthode de compression n'introduit pas de distorsions majeures sur les images résultantes qui conservent un PSNR et un MSSIM supérieur à 20 dB et 0.85 respectivement. De plus, la lecture simultanée des pixels actifs a maintenu un taux de pixels lus inférieur à 0.1 % pour des images de résolution 512 x 512. En comparaison avec les résultats obtenus avec les simulations haut niveau entièrement sous MATLAB, les valeurs des paramètres de qualité obtenues restent du même ordre de grandeur que ceux obtenus avec les simulations du chapitre 7. Par conséquent, on considère que ces résultats valident le fonctionnement événementiel et le circuit asynchrone de notre système de vision.

**TABLE 8.1** – Résultats des simulations niveau *netlist* des images de référence 8.9 pour  $\Delta_m = 2$

| Référence | Taux de lecture | PSNR (dB) | MSSIM | Erreur (%) |
|-----------|-----------------|-----------|-------|------------|
| 8.9a      | 0.076 %         | 25.20 dB  | 0.855 | 5.86 %     |
| 8.9b      | 0.059 %         | 25.04 dB  | 0.878 | 5.94 %     |
| 8.9c      | 0.038 %         | 24.21 dB  | 0.887 | 6.59 %     |
| 8.9d      | 0.11 %          | 27.37 dB  | 0.957 | 9.23 %     |
| 8.9e      | 0.098 %         | 28.32 dB  | 0.905 | 8.16 %     |
| 8.9f      | 0.0975 %        | 27.62 dB  | 0.961 | 7.89 %     |
| 8.9g      | 0.078 %         | 26.07 dB  | 0.906 | 5.17 %     |
| 8.9h      | 0.0887 %        | 27.31 dB  | 0.945 | 3.76 %     |
| 8.9i      | 0.045 %         | 23.05 dB  | 0.886 | 5.98 %     |



**FIGURE 8.10** – Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9h

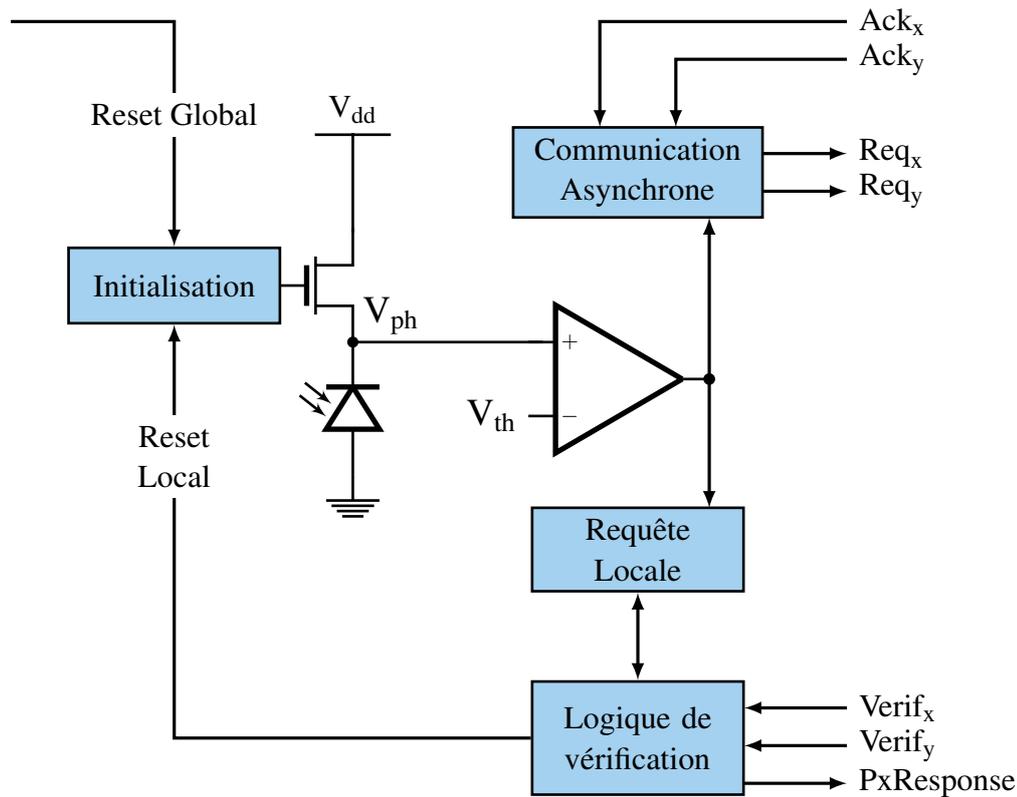


FIGURE 8.11 – Architecture du pixel événementiel

### 8.3 Architecture du pixel événementiel

Le pixel événementiel proposé conserve la structure de base du pixel codé en *Time-to-First-Spike* présenté dans le chapitre 2. En d'autres termes, le pixel comporte, d'une part, un comparateur analogique qui compare l'état de la tension aux bornes de la photodiode par rapport à une tension de référence et, d'autre part, un bloc chargé de la communication asynchrone à 4 phases entre le pixel et le système de lecture (envoi de requête de lecture, réception d'acquittement). Comme on l'a déjà mentionné, le pixel événementiel emploie une signalisation à deux dimensions (x,y). En d'autres termes, il envoie un signal par colonne et un autre par ligne. Les deux composantes x et y d'un signal sont reçues par le système de lecture qui en déduit, par la suite, l'adresse du pixel actif. On peut alors considérer que, suite à une détection d'événement, le pixel communique son identité, en termes d'adresse, au travers des signaux de requêtes.

Afin d'adapter cette architecture à la chaîne de lecture de notre capteur d'images, on développe sur la structure initiale du pixel TFS, en premier lieu, un bloc qui sauvegarde une requête locale suite à la détection d'un événement. En deuxième lieu, on ajoute un bloc chargé de la réception des signaux de vérification ( $Verif_x$ ,  $Verif_y$ ), envoyés par le système de lecture, et capable de répondre à ce dernier suivant l'état du pixel (actif ou

pas) au travers du signal (PxResponse). Ces trois signaux, échangés entre le bloc de traitement des requêtes de lecture et le pixel, sont représentés dans l'architecture du système de lecture (Figure 8.1) par les signaux de vérification. Enfin, pour que notre pixel événementiel soit capable d'exécuter un reset local, on lui ajoute un composant chargé de l'initialisation suite à la réception du signal d'initialisation global (Reset Global) ou du signal d'initialisation local. On note que le signal de reset local est généré par le bloc de vérification à la fin de la phase de fonctionnement de ce dernier. Cette initialisation annonce le début d'une nouvelle itération pour ce pixel. L'architecture de notre pixel événementiel est illustrée par la figure 8.11.

L'architecture du pixel événementiel proposé se distingue par la présence d'une majorité de signaux numériques entrants et sortants du pixel. En fait, 7 signaux numériques sont échangés entre le système de lecture et le pixel afin d'en assurer le bon fonctionnement plus un signal d'initialisation global. Ces signaux numériques sont moins sensibles au bruit ce qui implique théoriquement un rapport signal sur bruit (SNR) plus élevé. En outre, les signaux numériques sont moins affectés par l'effet capacitif des bus que les signaux analogiques. Ceci les rend plus avantageux en terme de vitesse de lecture. Enfin, on considère que l'utilisation des signaux numériques rend le pixel événementiel plus robuste et, donc, compatible avec des technologies de fabrication récente.

Le seul signal analogique provenant du périphérique de la matrice de pixels est le signal de la tension de référence. Mais ce signal, étant constant et sans changement pendant tout le fonctionnement du capteur d'images, il n'affecte pas les performances du système de vision en termes de vitesse et de bruit.

## 8.4 Conclusion

Dans ce chapitre, on a présenté le circuit du système de lecture. Ce circuit, décrit en langage VHDL au niveau *netlist*, est simulé par un testbench qui imite, génère et réagit avec le système de lecture de la même manière que le pixel événementiel décrit au chapitre 6. Ce testbench nous a permis de visualiser le fonctionnement et les échanges des signaux entre les sous-blocs du circuit. Les résultats obtenus par ce testbench, étant conforme au fonctionnement initialement présenté, nous avons développé un testbench qui se sert des images de référence afin d'évaluer le fonctionnement du circuit avec des stimuli événementiels. Pour cela, on a fait appel à MATLAB qui a généré à partir d'une image de référence un flot données événementiel imitant celui en sortie du capteur d'images. Ce flot de données, incluant les requêtes de lecture des pixels, est traité par le circuit décrit en VHDL au niveau *netlist*. Ce dernier génère et calcule à son tour le temps d'intégration de chaque pixel qui sert à reconstruire les images. A la fin de chaque simulation, l'image compressée est évaluée avec les mêmes paramètres utilisés pour les simulation de haut niveau. Les résultats obtenus à la fin de ces deux types de simulation (haut niveau et niveau *netlist*), d'une part, maintiennent le même ordre de grandeur, et

d'autre part, valide le fonctionnement et l'architecture du système de lecture numérique asynchrone au niveau fonctionnel.

## 9.1 Conclusion

Ce manuscrit synthétise mes travaux de thèse effectués dans le domaine de la conception des capteurs d'images événementiels et asynchrones. Dans cette thèse, on a présenté une approche originale qui consiste à fusionner l'aspect événementiel et la logique asynchrone, dans le but d'obtenir un imageur événementiel et véritablement asynchrone. De plus, le capteur d'images présenté réduit drastiquement l'activité du capteur et par conséquent participe grandement à son efficacité énergétique. En d'autres termes, l'activité du capteur et sa consommation sont uniquement dictées par l'apparition d'évènements qui dépendent de la scène capturée.

La chaîne de lecture de notre capteur d'images commence avec le pixel événementiel, dit *TFS*. Ce dernier est un pixel à intégration adapté pour qu'il puisse se comporter comme un détecteur d'événement. Son information lumineuse est codée par le temps selon le principe bio-inspiré *Time-to-First-Spike*. Ce codage du temps remplace la conversion tension-valeur numérique comme c'est le cas avec les imageurs conventionnels. Cette approche nous permet de palier à la consommation inhérente à la conversion analogique-numériques intégrée aux capteurs d'images. Le pixel événementiel, suite à la détection d'un nouvel événement, devient actif et déclenche la chaîne de lecture en envoyant au système de lecture asynchrone une requête de lecture. Le système de lecture, comportant un bloc d'horodatage, enregistre ensuite l'instant de réception de la requête de lecture. Cette information temporelle est essentielle lors du calcul du temps d'intégration du pixel.

Le protocole de lecture événementiel optimisé, proposé dans cette thèse, consiste à recevoir et traiter les requêtes de lecture simultanées. En d'autres termes, les pixels actifs dans la même fenêtre de temps, sont simultanément lus, créant ainsi une lecture déterministe et efficace en énergie. Par conséquent, la lecture de notre imageur se fait par groupe de pixels actifs. Cette nouvelle approche de lecture et le principe de réception simultanée des requêtes de lecture nous permet de réduire les redondances spatiales existantes dans le flot en sortie du capteur d'images événementiel. Cette gestion du flot de données au niveau du capteur implique la réduction du nombre de lecture demandée et la réduction des traitements associés à chaque lecture. De plus, la lecture simultanée nous permet d'éviter les problèmes d'arbitrage non déterministe qui sont considérés comme des limitations majeures dans les imageurs événementiels existant dans la littérature. Suite à la phase de réception des requêtes, le système calcule le temps d'intégration de chaque pixel actif, libérant par la suite ce dernier par une relance de l'intégration qui

met fin à l'itération de lecture. Le système de lecture devient alors capable de recevoir de nouvelles requêtes en provenance des pixels.

En résumé, notre capteur d'images possède les caractéristiques suivantes :

- Protocole de lecture événementiel, simultanée et déterministe.
- Réduction des redondances spatiales dans le flot de données contrôlée par la fenêtre de temps ouverte pendant la réception des requêtes de lecture. Cela correspond à une technique intégrée de compression des données sans perte.
- Le capteur d'images ne fonctionne plus selon le concept de lecture de trames. De même, il n'est plus possible de parler de la vitesse en trames par seconde. Il vaut mieux indiquer dans ce cas le nombre d'itérations de lecture par seconde.
- Les pixels fonctionnent indépendamment les uns des autres et chacun d'entre eux possède son propre temps d'intégration. Il est ainsi possible de tirer parti de cette indépendance des temps d'intégration pour rendre de façon native notre capteur d'images événementiel HDR (*High Dynamic Range*).

Dans le but d'évaluer la méthode de compression, nous avons effectué (chapitre 7) des simulations de haut niveau sous MATLAB en utilisant différentes images numériques et estimant l'impact sur les images si elles avaient été capturées par un imageur événementiel. La qualité des images résultantes a été évaluée grâce à des paramètres et à des facteurs de qualité usuels comme le PSNR, le MSSIM et l'erreur entre l'image de référence et l'image résultante. Les résultats obtenus ont montré que le taux de compression par image est drastique en diminuant le nombre de lectures nécessaires par image à moins de 1% par rapport à une lecture exhaustive d'un imageur conventionnel et en conservant une image de bonne qualité (MSSIM > 0.9 - PSNR > 25 dB).

Ensuite, nous avons présenté la méthode d'implémentation de notre système de lecture événementiel en logique asynchrone. En effet, tous les composants de son architecture fonctionnent sur événement. Cette architecture, modélisée en VHDL, a été validée au niveau fonctionnel en la simulant avec plusieurs flots de données issues d'images de référence. Les résultats obtenus au niveau du nombre de lectures nécessaires par image et la qualité des images résultantes confirment l'efficacité de notre approche. De plus, ces résultats maintiennent le même ordre de grandeur sur les résultats obtenus avec les simulations de haut niveau effectuées précédemment.

Enfin, le pixel événementiel analogique est présenté au niveau bloc et au niveau transistor. Ce pixel à 48 transistors, dont 41 sont numériques à taille quasi minimale, maintient un facteur de remplissage raisonnable pour ce type de circuit. De plus, avec des sorties uniquement numériques, on s'attend, suite à sa fabrication, à un pixel robuste côté circuit de lecture.

**TABLE 9.1** – Comparaison entre notre capteur d’images événementiel et asynchrone et les principaux capteurs d’images événementiels de la littérature

| Références   | Protocole de lecture |                     | Arbitre    | Taux de lecture            | Compression de données       |
|--|----------------------|---------------------|------------|----------------------------|------------------------------|
| [YLD15 ; Lic08 ; Lic06b]<br>[LPD06 ; LDK04]                                  | Séquentiel           | Non-déterministe    | Oui        | <i>Pixel Rate</i>          | Metadata                     |
| [PMW11 ; PMW08]  | Séquentiel           | Non-déterministe    | Oui        | <i>Pixel Rate</i>          | Redondances temporelles      |
| [SB07 ; SB06 ; KB04]   | Séquentiel           | Non-déterministe    | Oui        | <i>Pixel Rate</i>          | Pas de compression           |
| <b>Notre capteur</b><br>[DFS15 ; Dar+15 ; FDS15]<br>[DSF14 ; DFS14 ; Fes+14] | <b>Parallèle</b>     | <b>Déterministe</b> | <b>Non</b> | <b><i>Reading Rate</i></b> | <b>Redondances spatiales</b> |

On achève notre conclusion par une comparaison entre notre capteur d’images événementiel et asynchrone proposé et les principales contributions citées dans l’état de l’art (Tableau 9.1). Dans le cas de toutes les références citées, le protocole de lecture utilisé étant séquentiel, cela implique le besoin d’un circuit d’arbitrage afin de résoudre le conflit entre les requêtes de lecture simultanée. Ceci dit, ces protocoles de lecture demeurent non-déterministes. En revanche, le parallélisme de notre protocole de lecture rend tout type de circuit d’arbitrage inutile à notre architecture. De plus, les capteurs d’images cités étant événementiels, ils possèdent un taux de lecture défini par le nombre de pixel actif (séquentiellement lu) pendant un intervalle de temps (*Pixel Rate*). Cependant, notre système de lecture, qui prend en compte tous les pixels actifs au même instant, a un taux de lecture qui dépend du nombre d’itération de lecture nécessaire (*Reading Rate*). Il est important de noter que ce dernier est plafonné par le nombre de niveau de gris de la scène capturée. De ce fait, et même dans le cas d’une scène extrêmement complexe, le nombre d’itération (notamment au démarrage) sera limité aux nombres de niveaux de luminances détectables. Enfin, on compare la méthode de compression de données employées dans les capteurs d’images événementiels. Dans le cas de [Lic08 ; Lic06b ; LPD06 ; LDK04], la compression de données est effectuée en envoyant seulement la metadata. En d’autres termes, seule l’information à propos de l’image est envoyée sans envoyer l’image complète. Cependant, [PMW11 ; PMW08] a employé une suppression des redondances temporelles, contrairement à notre cas où une suppression de redondances spatiales a été employée, afin de réduire le taux de données en sortie du capteur.

Ainsi, en adaptant l'activité de notre capteur d'images à la scène capturée, en réduisant le flot de données en sortie et en utilisant de la logique asynchrone connue pour sa faible consommation par rapport à la logique synchrone, le capteur d'images événementiel asynchrone proposé se montre théoriquement efficace énergétiquement.

## 9.2 Perspectives

Personnellement, je vais avoir le privilège de continuer mes travaux de recherche dans le cadre du projet *Event-Based Control, Circuits and Processing towards Ultra-Low Power Consumption* (e-BaCCuSS<sup>1</sup>) monté par mes directeurs de thèse en collaboration avec plusieurs laboratoires grenoblois. L'objectif général de ce projet consiste à étudier et développer des techniques d'échantillonnage non-uniformes dans plusieurs domaines d'applications.

Dans le cadre de ce projet, le travail sur la thématique des imageurs prévoit, en premier lieu, l'optimisation de l'architecture du pixel événementiel et notamment la conception du comparateur analogique. Le but sera de diminuer autant que possible les dispersions de caractéristiques de ce comparateur. Cette étape sera suivie par la conception d'un imageur jusqu'au dessin des masques, par sa fabrication et par toutes les étapes de caractérisation et de test du circuit conçu. Une étude approfondie de la consommation électrique de notre capteur d'images est notamment prévue une fois que le fonctionnement aura été validé.

Enfin, l'étude de traitement d'images intégrée est incluse dans le cadre de ce projet. Les données en sortie de notre capteur d'images, grâce à la présence des blocs de mémoire, peuvent subir un traitement d'images classique similaire à ceux utilisés avec les capteurs d'images conventionnels. Cependant, afin de profiter de l'aspect événementiel, on cherchera à exploiter le flux de données issu du bloc horodateur pour développer des techniques de traitement d'images événementielles. Dans les années à venir, il sera sans doute intéressant de combiner le traitement d'images (détection de contour, filtrage...) et l'aspect événementiel et asynchrone de notre imageur. Un traitement d'images numérique intégré (en dehors de la matrice des pixels) est prévu dans le cadre du projet e-BaCCuSS. Il sera implémenté dans un FPGA en dehors du prototype de l'imageur. En outre, on pourra développer des méthodes de réduction des redondances temporelles afin d'avoir un flot de données efficient en sortie du capteur d'images qui intègre des techniques de compression spatiales et temporelles.

A long terme, je crois que le développement et l'utilisation de la technologie de fabrication 3D dans le domaine de la microélectronique servira certainement les capteurs d'images en termes de contrôle du flot de données et de gestion de la consommation électrique tout en conservant une petite taille du pixel. En effet, il sera possible d'utiliser les

---

1. <https://persyval-lab.org/en/sites/e-baccuss>

différentes couches pour y ajouter de la logique et préserver la couche photosensible sur le dessus. En particulier pour les capteurs événementiels, la technologie 3D facilitera la lecture simultanée et événementielle et permettra l'implémentation de nouvelles fonctionnalités de traitement d'images avec un effet minimal sur la surface photosensible du capteur. Ceci ouvrira la voie aux capteurs d'images événementiels et asynchrones vers l'industrie avec de nombreuses applications à la clé (automobiles, surveillance, détection de mouvement, applications grand public, ...).



# A

## SUITE DES SIMULATIONS DU CHAPITRE 7

Cet annexe comporte les images résultantes et compressées des simulations effectuées dans le chapitre 7. La qualité des images compressées est évaluées grâce à des paramètres de qualité : PSNR, MSSIM, l'erreur introduite par la compression ainsi que la taux de lecture nécessaire suivant le protocole de lecture proposé au cours de cette thèse. Les résultats obtenus en termes de réduction de nombre de lecture par image et de qualité des images compressées valident le fonctionnement haut niveau présenté au cours du chapitre 6.



(a) Image de référence, MSSIM = 1



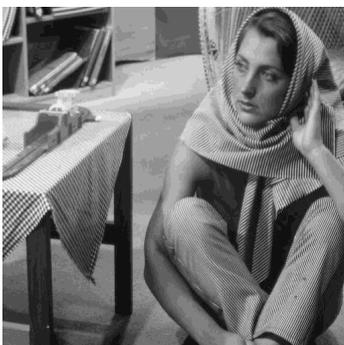
(b)  $\Delta = 2$ , PSNR = 43.2dB, MSSIM = 0.999



(c)  $\Delta = 4$ , PSNR = 39.03dB, MSSIM = 0.970



(d)  $\Delta = 8$ , PSNR = 33.90dB, MSSIM = 0.897



(e)  $\Delta = 16$ , PSNR = 31.64dB, MSSIM = 0.768



(f)  $\Delta = 32$ , PSNR = 23.01dB, MSSIM = 0.603

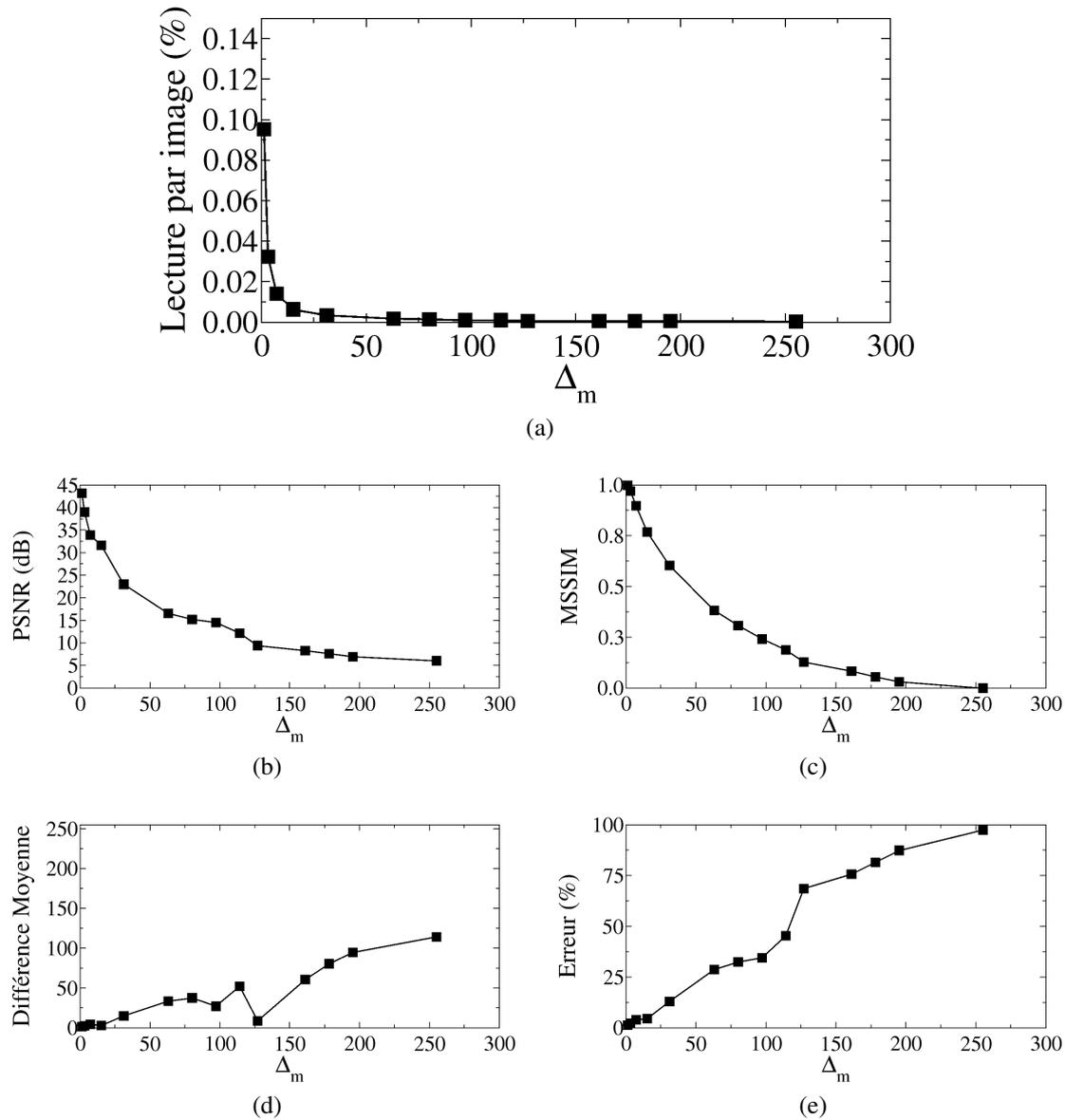


(g)  $\Delta = 64$ , PSNR = 16.58dB, MSSIM = 0.383



(h)  $\Delta = 128$ , PSNR = 9.44dB, MSSIM = 0.129

**FIGURE A.1** – Résultat des simulations MATLAB : Image E. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.2)



**FIGURE A.2** – Évaluation de la qualité des images résultantes de simulations de l'image E (Fig.A.1) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 53.50dB, MSSIM = 0.999



(c)  $\Delta = 4$ , PSNR = 45.91dB, MSSIM = 0.985



(d)  $\Delta = 8$ , PSNR = 39.89dB, MSSIM = 0.935



(e)  $\Delta = 16$ , PSNR = 29.81dB, MSSIM = 0.808



(f)  $\Delta = 32$ , PSNR = 24.47dB, MSSIM = 0.584

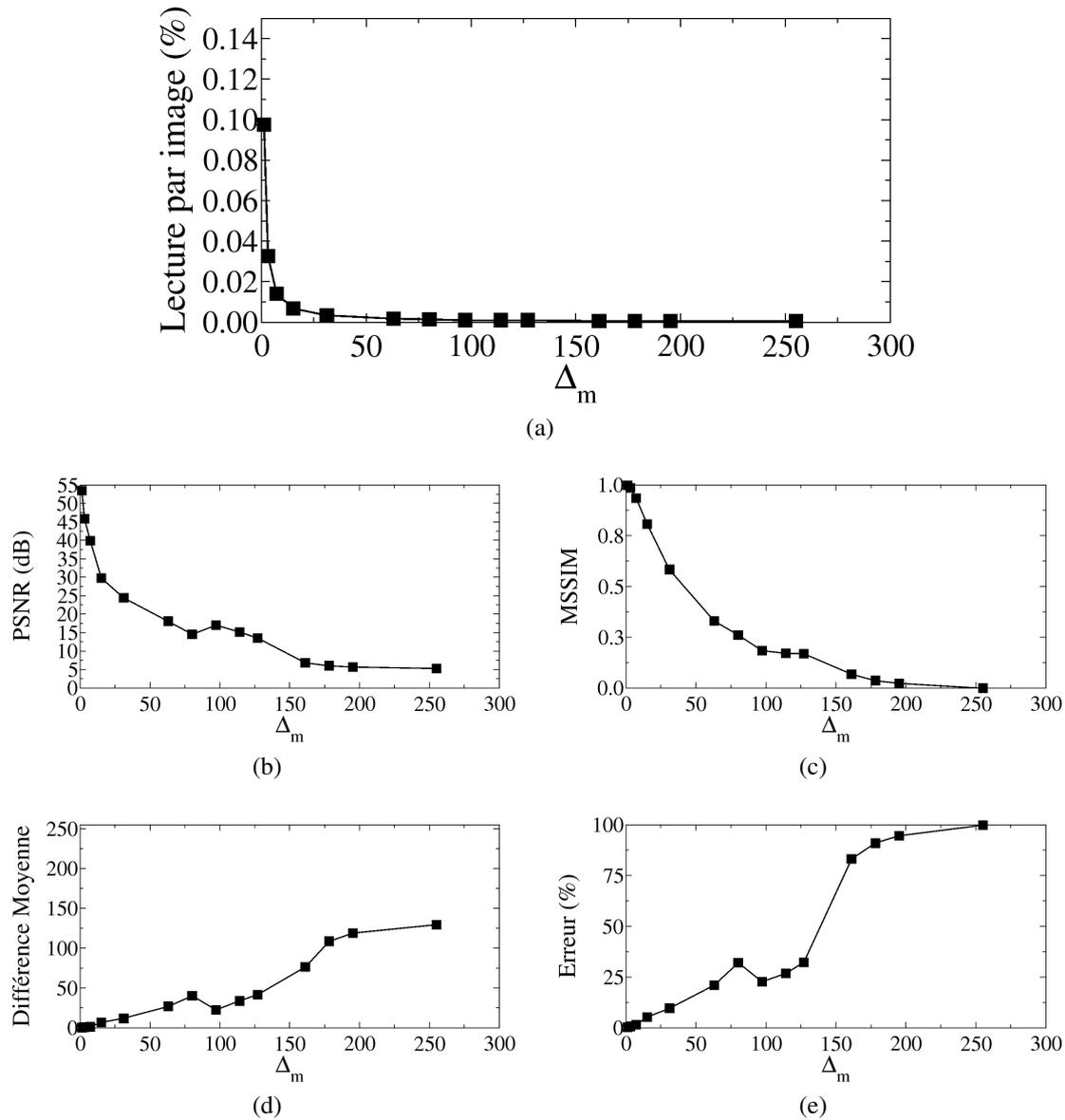


(g)  $\Delta = 64$ , PSNR = 18.12dB, MSSIM = 0.331



(h)  $\Delta = 128$ , PSNR = 13.58dB, MSSIM = 0.169

**FIGURE A.3** – Résultat des simulations MATLAB : Image F. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.4)



**FIGURE A.4** – Évaluation de la qualité des images résultantes de simulations de l'image F (Fig.A.3) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 29.65dB, MSSIM = 0.977



(c)  $\Delta = 4$ , PSNR = 29.09dB, MSSIM = 0.955



(d)  $\Delta = 8$ , PSNR = 27.91dB, MSSIM = 0.893



(e)  $\Delta = 16$ , PSNR = 25.48dB, MSSIM = 0.766



(f)  $\Delta = 32$ , PSNR = 20.75dB, MSSIM = 0.551

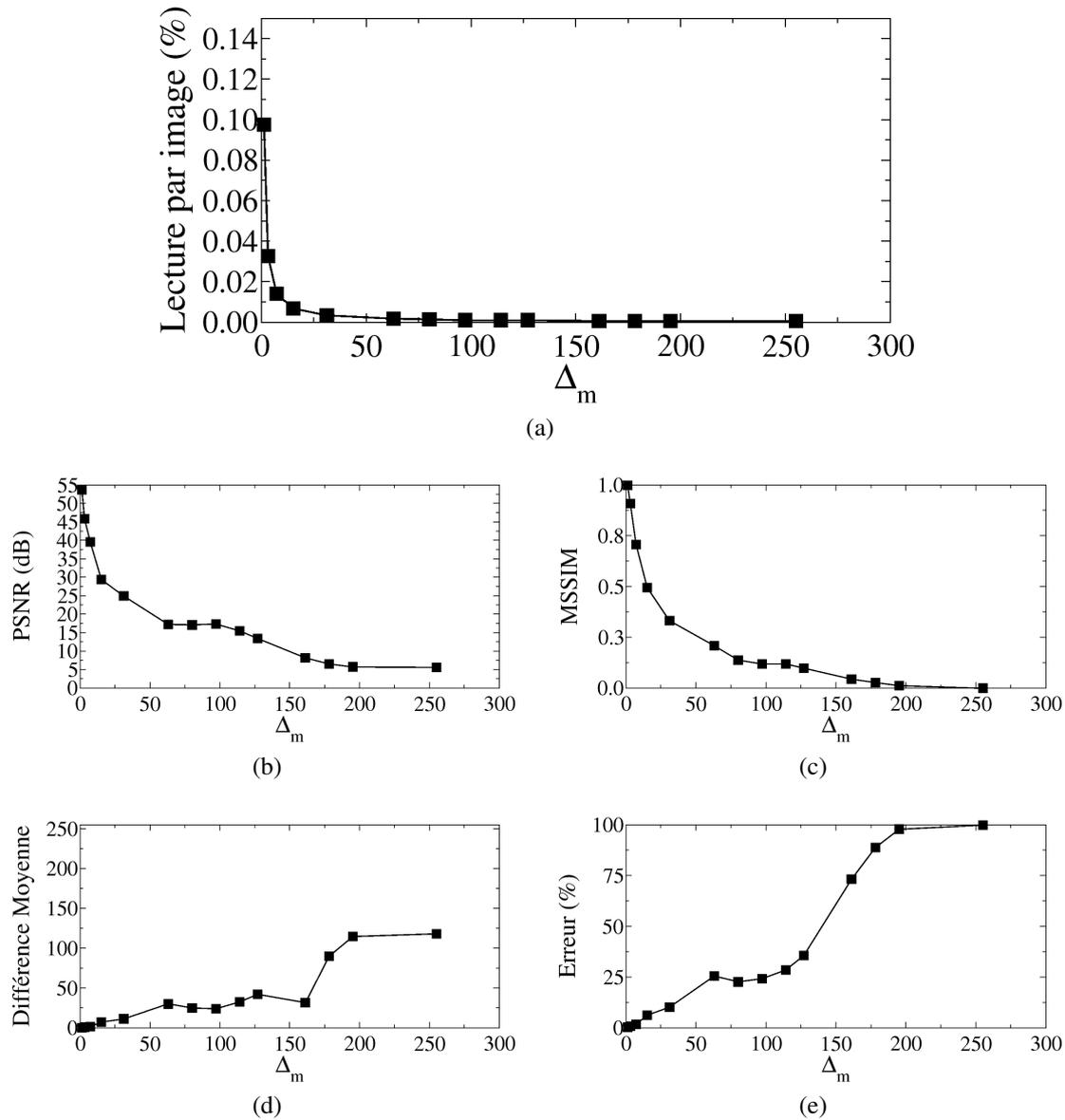


(g)  $\Delta = 64$ , PSNR = 16.03dB, MSSIM = 0.247



(h)  $\Delta = 128$ , PSNR = 8.85dB, MSSIM = 0.044

**FIGURE A.5** – Résultat des simulations MATLAB : Image G. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.6)



**FIGURE A.6** – Évaluation de la qualité des images résultantes de simulations de l'image G (Fig.A.5) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 27.57dB, MSSIM = 0.963



(c)  $\Delta = 4$ , PSNR = 27.26dB, MSSIM = 0.924



(d)  $\Delta = 8$ , PSNR = 26.27dB, MSSIM = 0.834



(e)  $\Delta = 16$ , PSNR = 24.41dB, MSSIM = 0.719



(f)  $\Delta = 32$ , PSNR = 20.17dB, MSSIM = 0.497

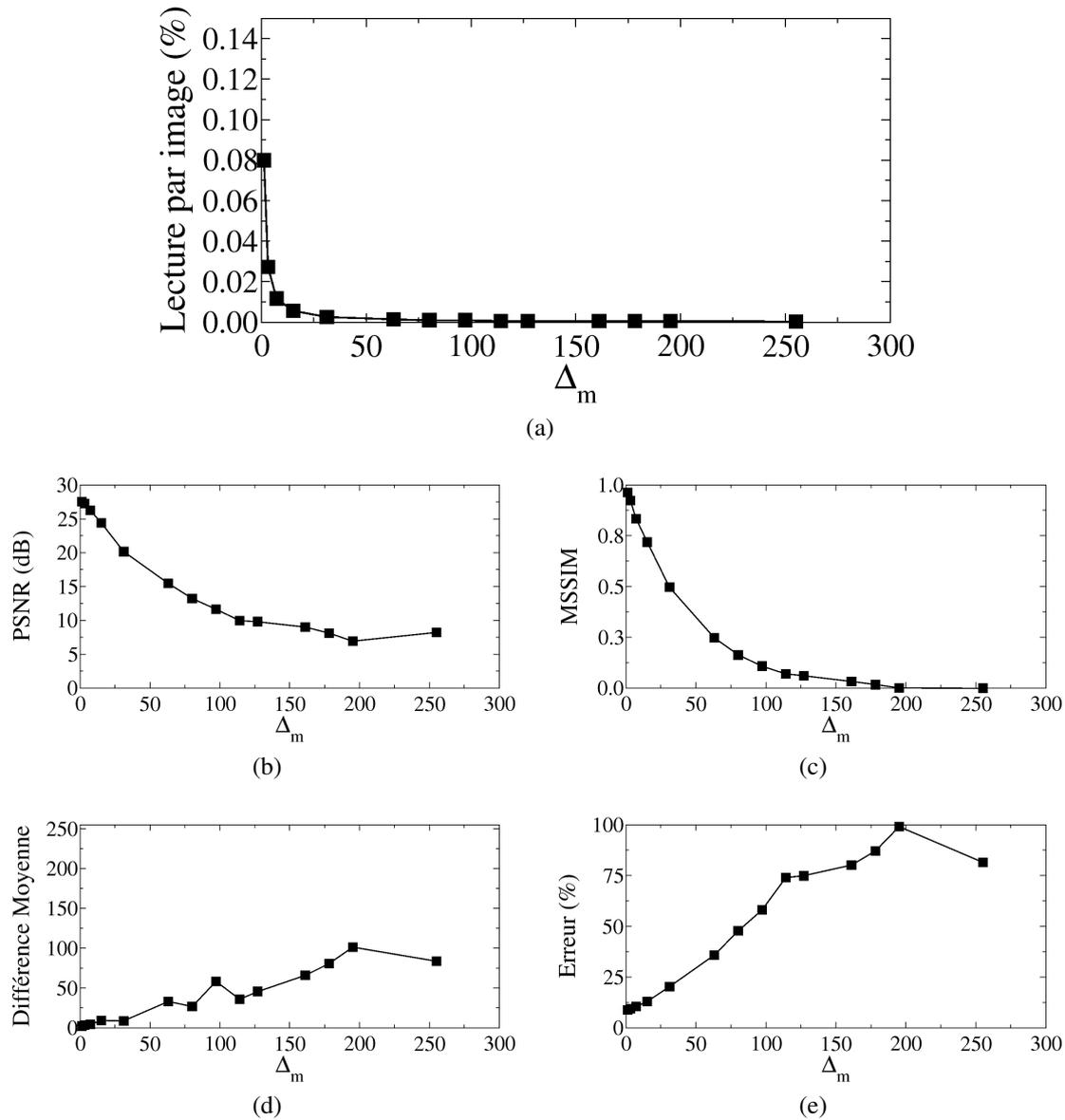


(g)  $\Delta = 64$ , PSNR = 15.46dB, MSSIM = 0.248

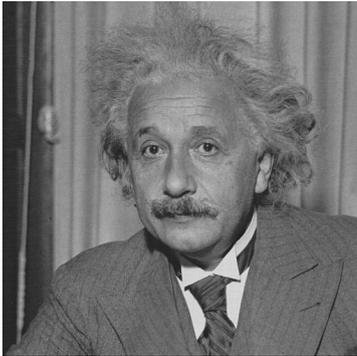


(h)  $\Delta = 128$ , PSNR = 9.83dB, MSSIM = 0.060

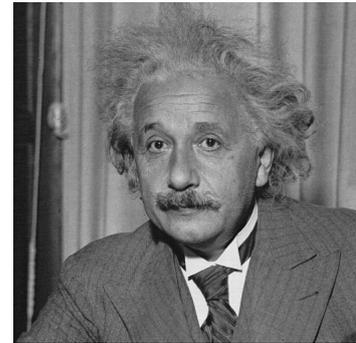
**FIGURE A.7** – Résultat des simulations MATLAB : Image H. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.8)



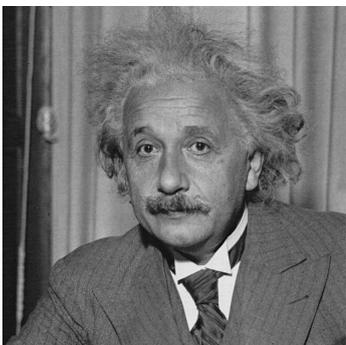
**FIGURE A.8** – Évaluation de la qualité des images résultantes de simulations de l'image H (Fig.A.7) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



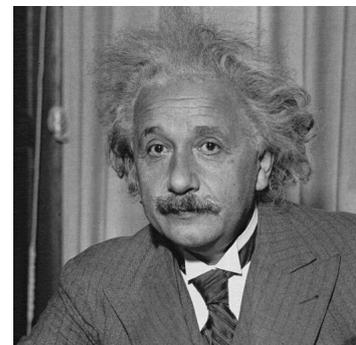
(a) Image de référence, MSSIM = 1



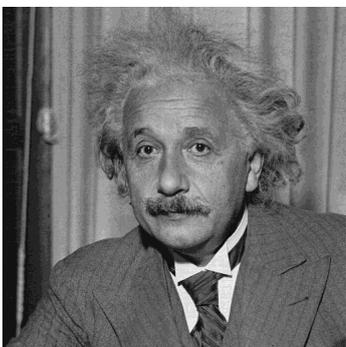
(b)  $\Delta = 2$ , PSNR = 32.58dB, MSSIM = 0.983



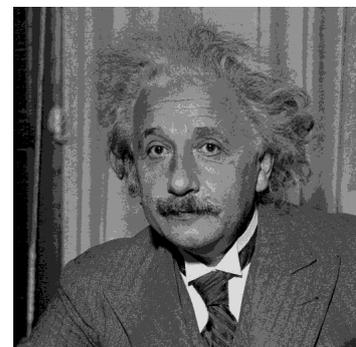
(c)  $\Delta = 4$ , PSNR = 31.70dB, MSSIM = 0.967



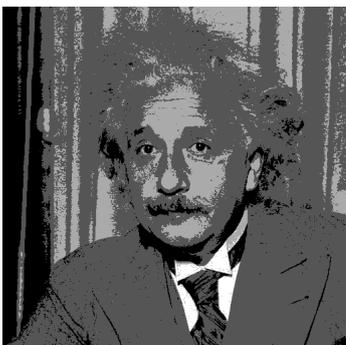
(d)  $\Delta = 8$ , PSNR = 30.06dB, MSSIM = 0.907



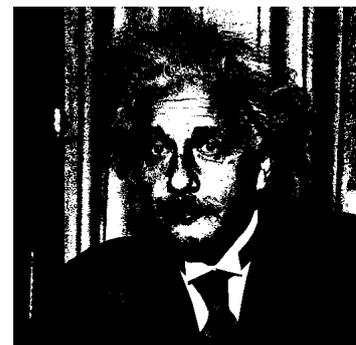
(e)  $\Delta = 16$ , PSNR = 27.31dB, MSSIM = 0.751



(f)  $\Delta = 32$ , PSNR = 20.72dB, MSSIM = 0.494

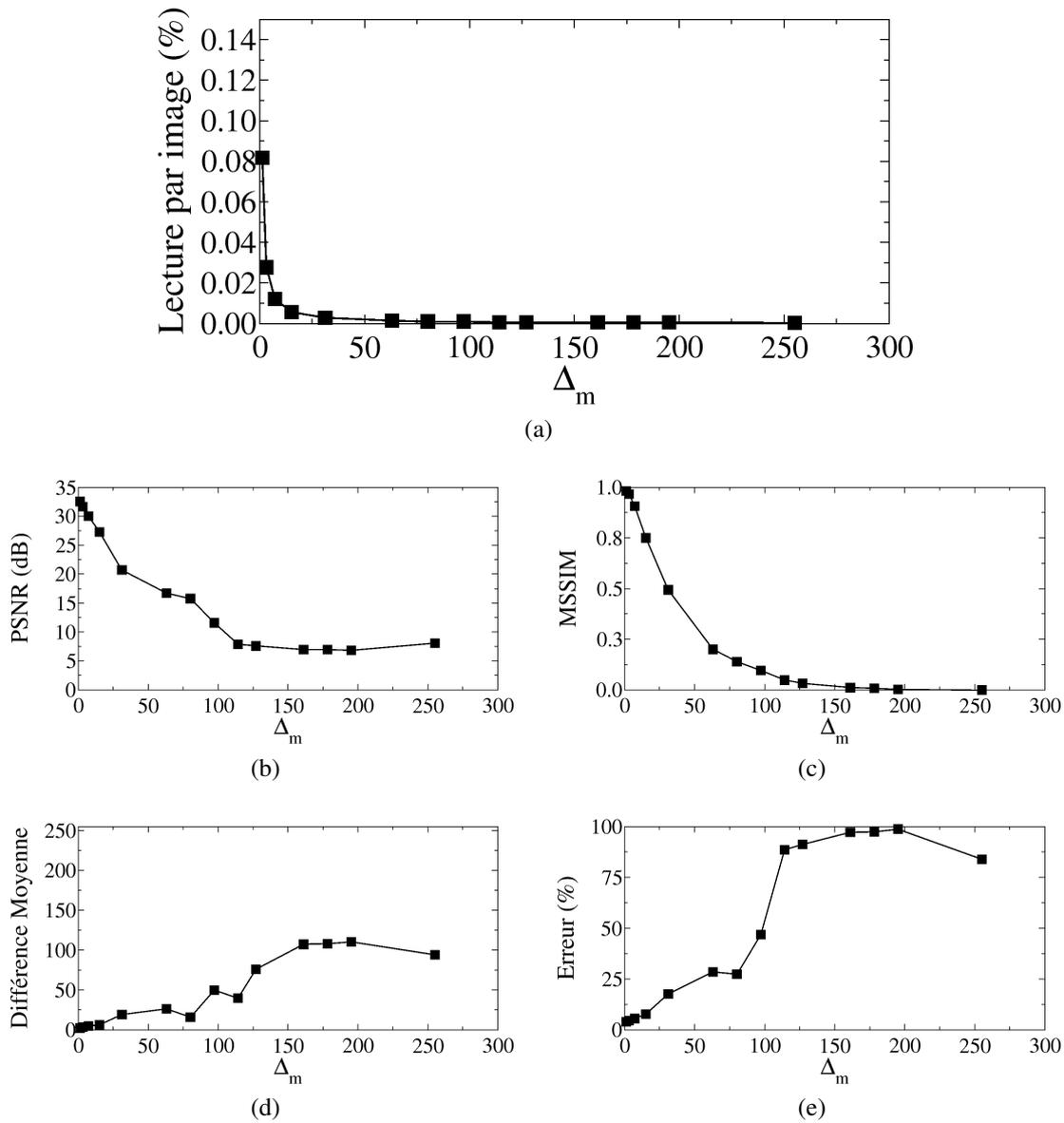


(g)  $\Delta = 64$ , PSNR = 16.77dB, MSSIM = 0.200



(h)  $\Delta = 128$ , PSNR = 7.61dB, MSSIM = 0.033

**FIGURE A.9** – Résultat des simulations MATLAB : Image I. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.10)



**FIGURE A.10** – Évaluation de la qualité des images résultantes de simulations de l'image I (Fig.A.9) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 39.19dB, MSSIM = 0.990



(c)  $\Delta = 4$ , PSNR = 37.75dB, MSSIM = 0.739



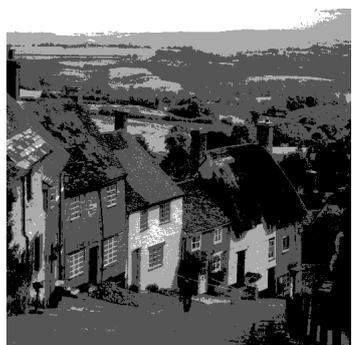
(d)  $\Delta = 8$ , PSNR = 33.98dB, MSSIM = 0.419



(e)  $\Delta = 16$ , PSNR = 28.72dB, MSSIM = 0.206



(f)  $\Delta = 32$ , PSNR = 22.47dB, MSSIM = 0.095

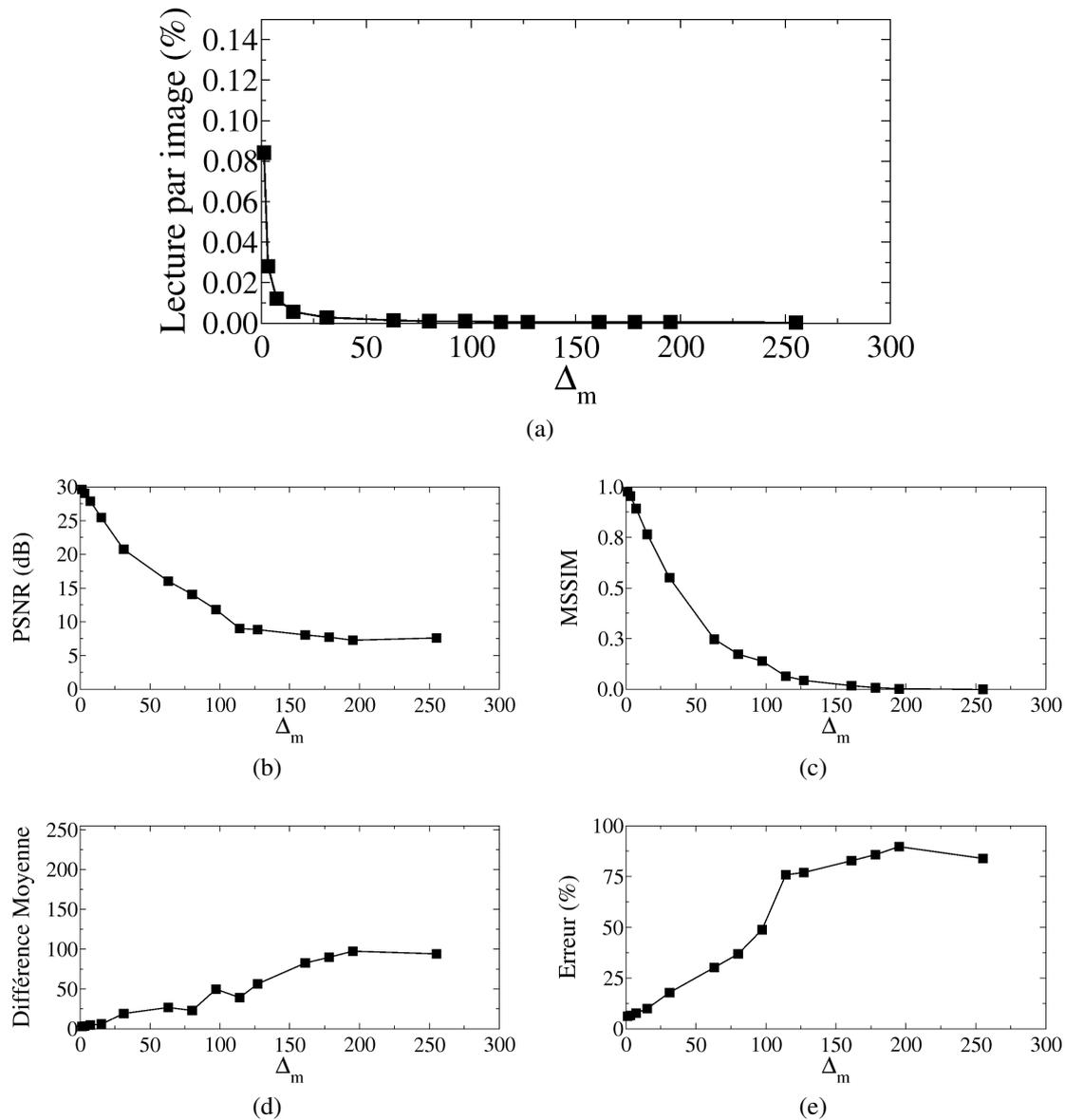


(g)  $\Delta = 64$ , PSNR = 18.55dB, MSSIM = 0.041



(h)  $\Delta = 128$ , PSNR = 15.6dB, MSSIM = 0.013

**FIGURE A.11** – Résultat des simulations MATLAB : Image J. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.12)



**FIGURE A.12** – Évaluation de la qualité des images résultantes de simulations de l’image J (Fig.A.11) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l’image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l’image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 31.10dB, MSSIM = 0.987



(c)  $\Delta = 4$ , PSNR = 30.34dB, MSSIM = 0.942



(d)  $\Delta = 8$ , PSNR = 28.36dB, MSSIM = 0.813



(e)  $\Delta = 16$ , PSNR = 26.24dB, MSSIM = 0.605



(f)  $\Delta = 32$ , PSNR = 22.13dB, MSSIM = 0.392

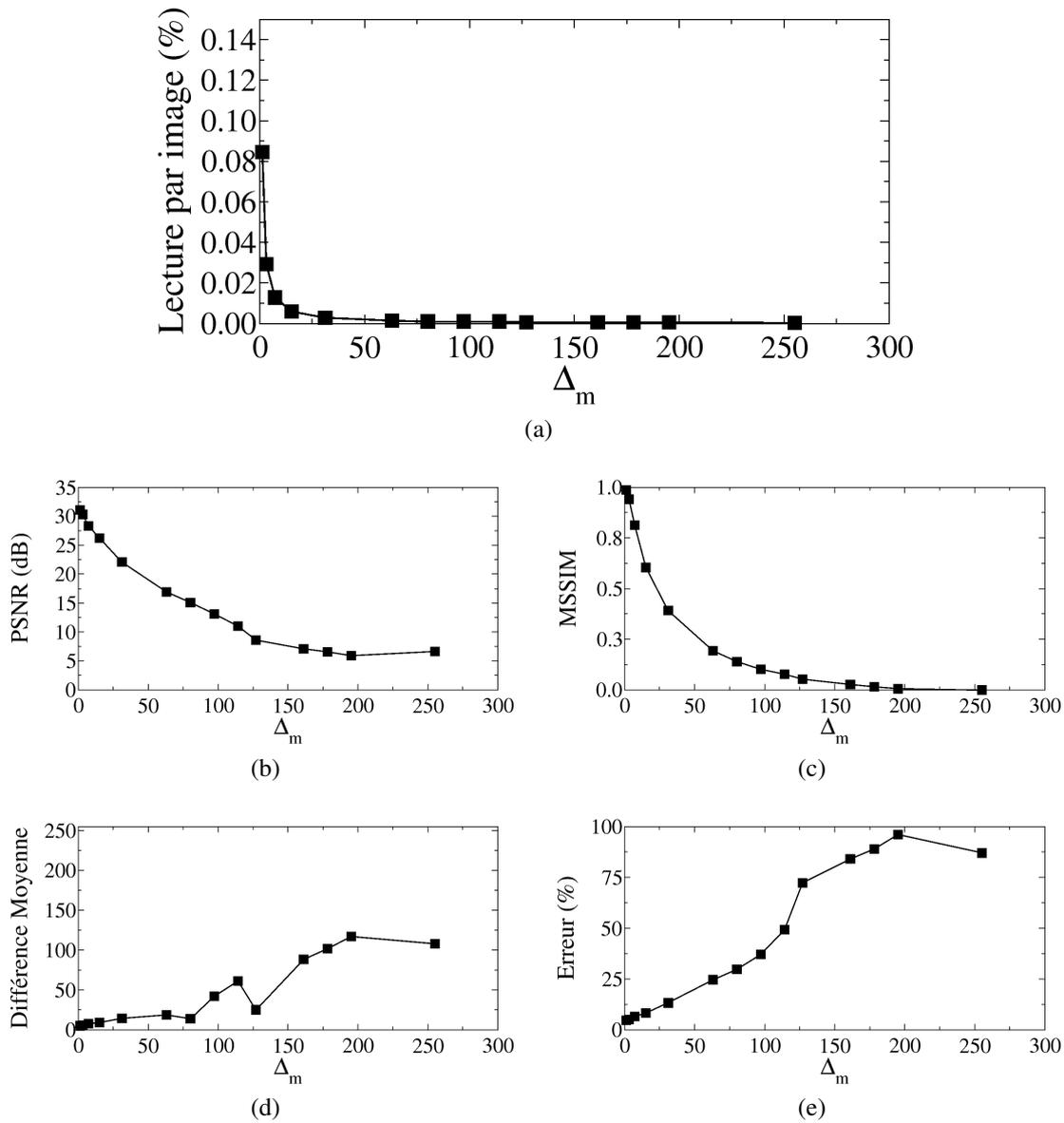


(g)  $\Delta = 64$ , PSNR = 16.95dB, MSSIM = 0.194

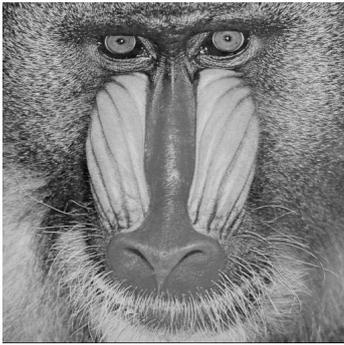


(h)  $\Delta = 128$ , PSNR = 8.61dB, MSSIM = 0.053

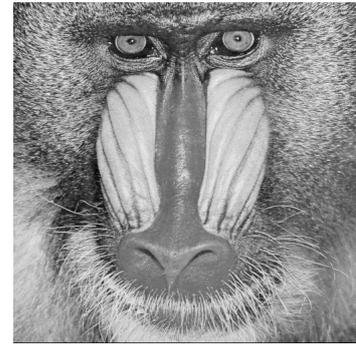
**FIGURE A.13** – Résultat des simulations MATLAB : Image K. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.14)



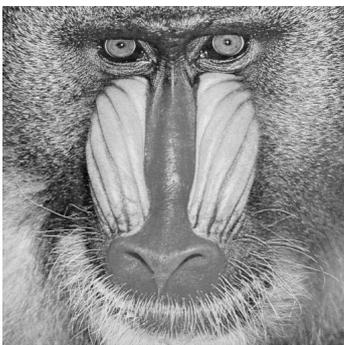
**FIGURE A.14** – Évaluation de la qualité des images résultantes de simulations de l'image K (Fig.A.13) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



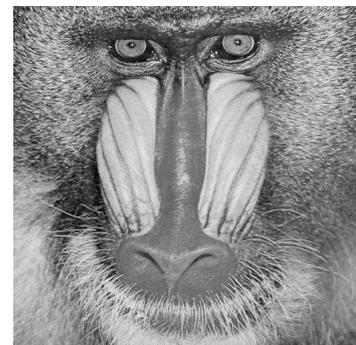
(a) Image de référence, MSSIM = 1



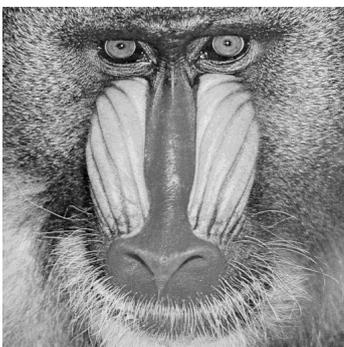
(b)  $\Delta = 2$ , PSNR = 25.09dB, MSSIM = 0.990



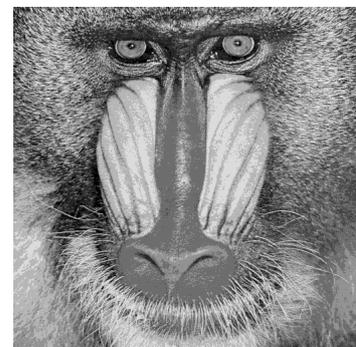
(c)  $\Delta = 4$ , PSNR = 25.73dB, MSSIM = 0.988



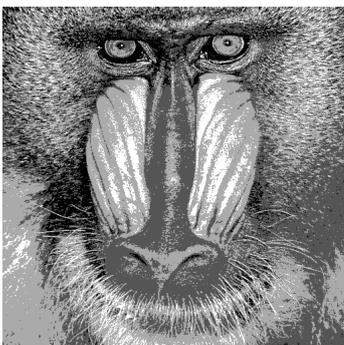
(d)  $\Delta = 8$ , PSNR = 27.05dB, MSSIM = 0.975



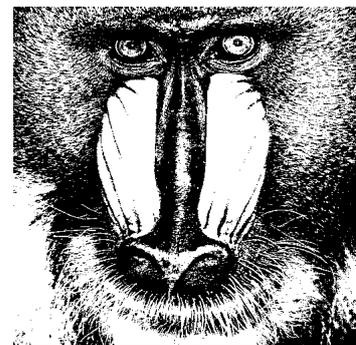
(e)  $\Delta = 16$ , PSNR = 26.53dB, MSSIM = 0.925



(f)  $\Delta = 32$ , PSNR = 25.21dB, MSSIM = 0.801

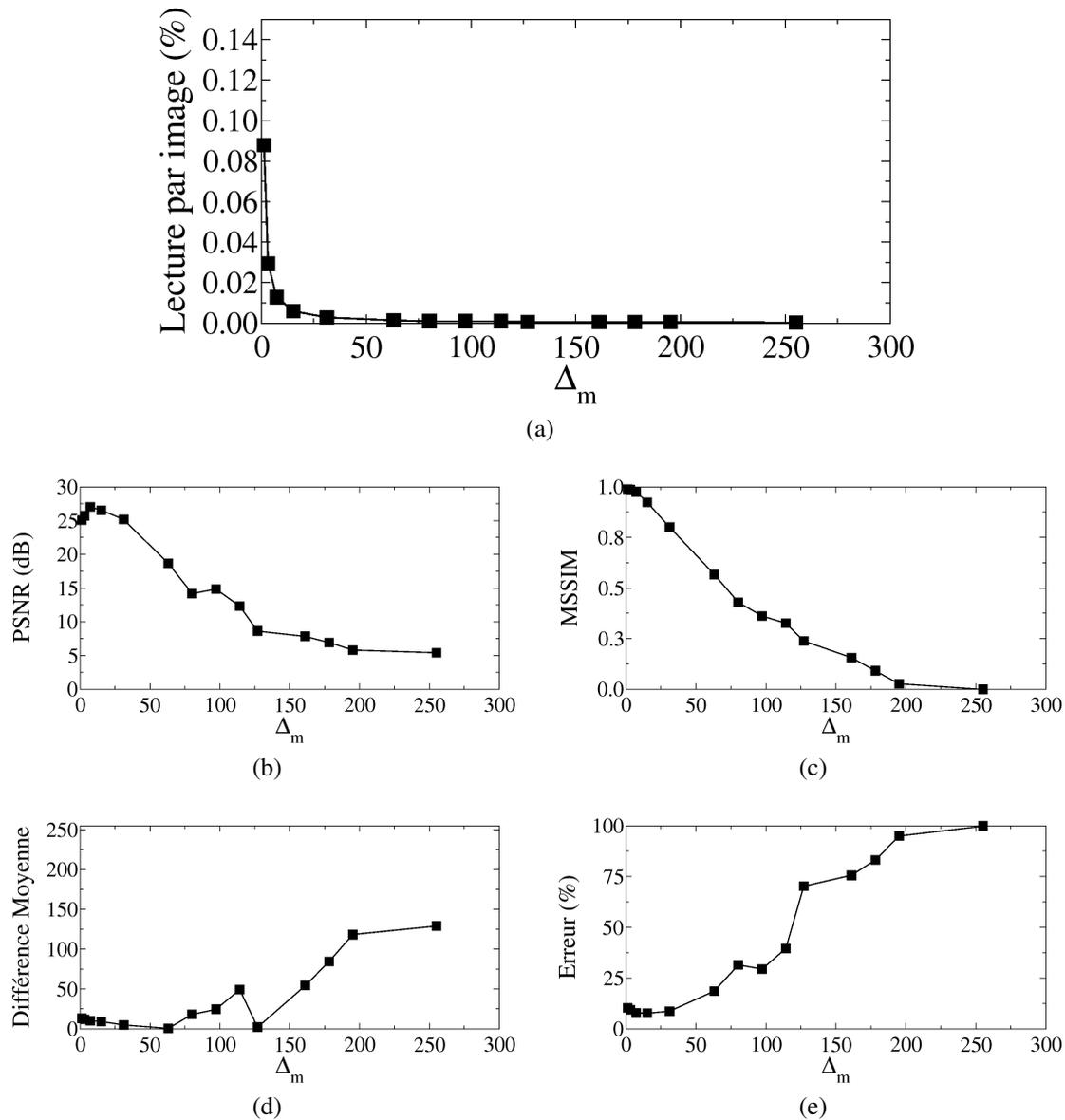


(g)  $\Delta = 64$ , PSNR = 18.69dB, MSSIM = 0.567



(h)  $\Delta = 128$ , PSNR = 8.56dB, MSSIM = 0.238

**FIGURE A.15** – Résultat des simulations MATLAB : Image L. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.16)



**FIGURE A.16** – Évaluation de la qualité des images résultantes de simulations de l’image L (Fig.A.15) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l’image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l’image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 34.40dB, MSSIM = 0.997



(c)  $\Delta = 4$ , PSNR = 34.08dB, MSSIM = 0.983



(d)  $\Delta = 8$ , PSNR = 35.58dB, MSSIM = 0.925



(e)  $\Delta = 16$ , PSNR = 29.69dB, MSSIM = 0.746



(f)  $\Delta = 32$ , PSNR = 24.66dB, MSSIM = 0.440

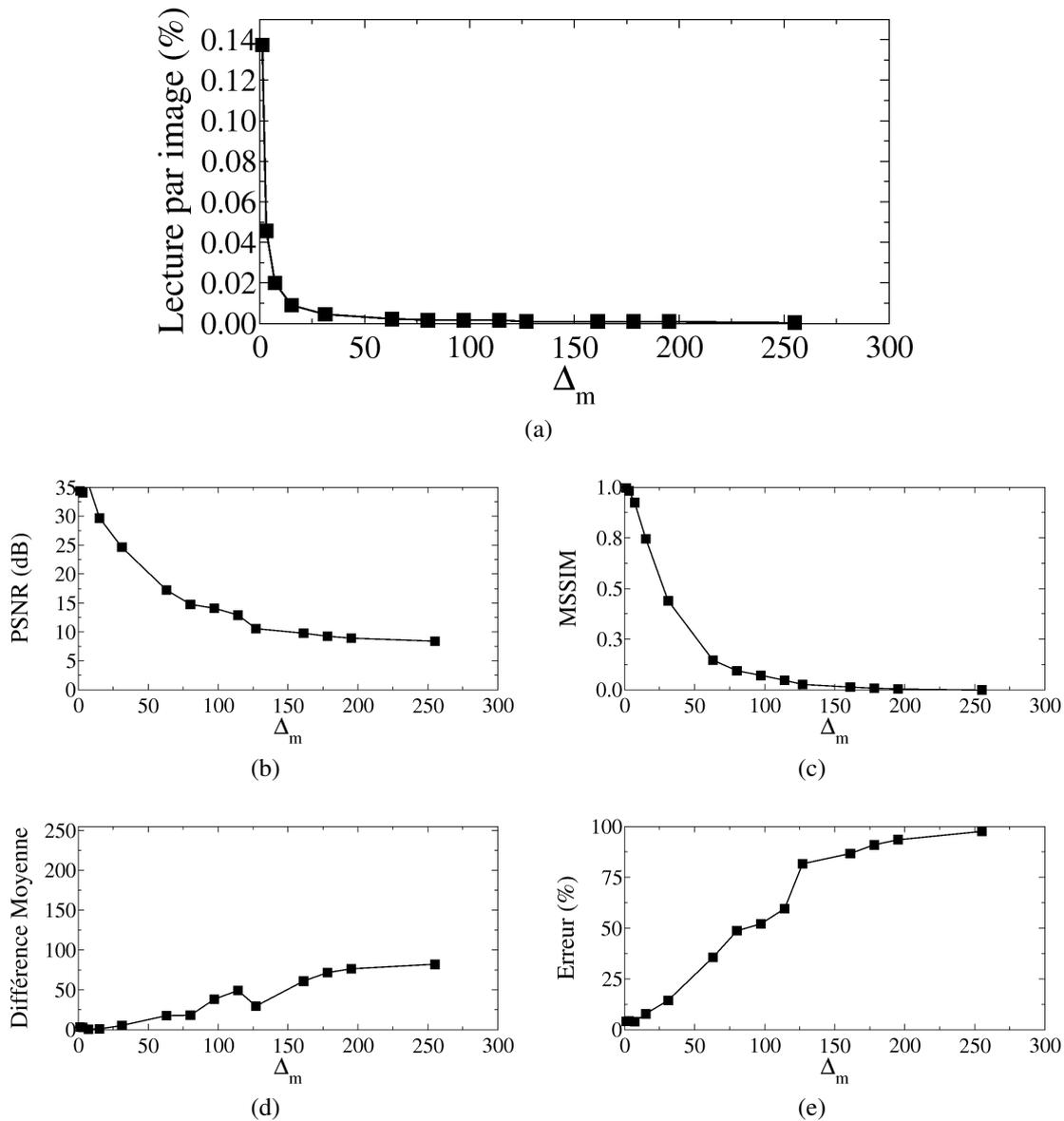


(g)  $\Delta = 64$ , PSNR = 17.30dB, MSSIM = 0.146



(h)  $\Delta = 128$ , PSNR = 10.57dB, MSSIM = 0.027

**FIGURE A.17** – Résultat des simulations MATLAB : Image M. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.18)



**FIGURE A.18** – Évaluation de la qualité des images résultantes de simulations de l'image M (Fig.A.17) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



(a) Image de référence, MSSIM = 1



(b)  $\Delta = 2$ , PSNR = 37.02dB, MSSIM = 0.996



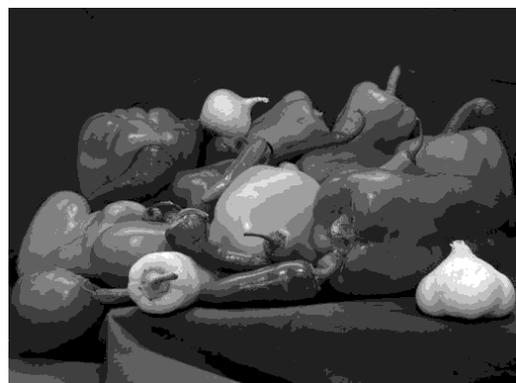
(c)  $\Delta = 4$ , PSNR = 35.24dB, MSSIM = 0.909



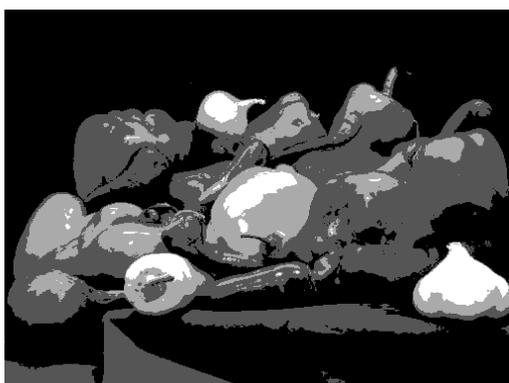
(d)  $\Delta = 8$ , PSNR = 33.04dB, MSSIM = 0.723



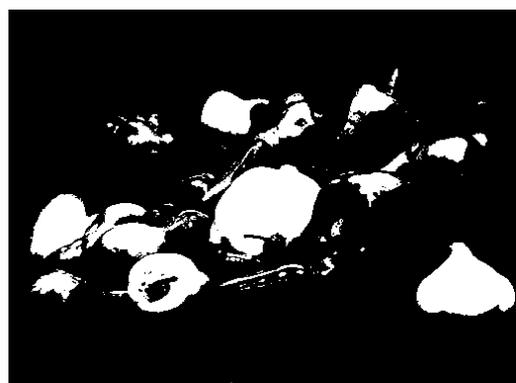
(e)  $\Delta = 16$ , PSNR = 28.59dB, MSSIM = 0.517



(f)  $\Delta = 32$ , PSNR = 22.28dB, MSSIM = 0.297

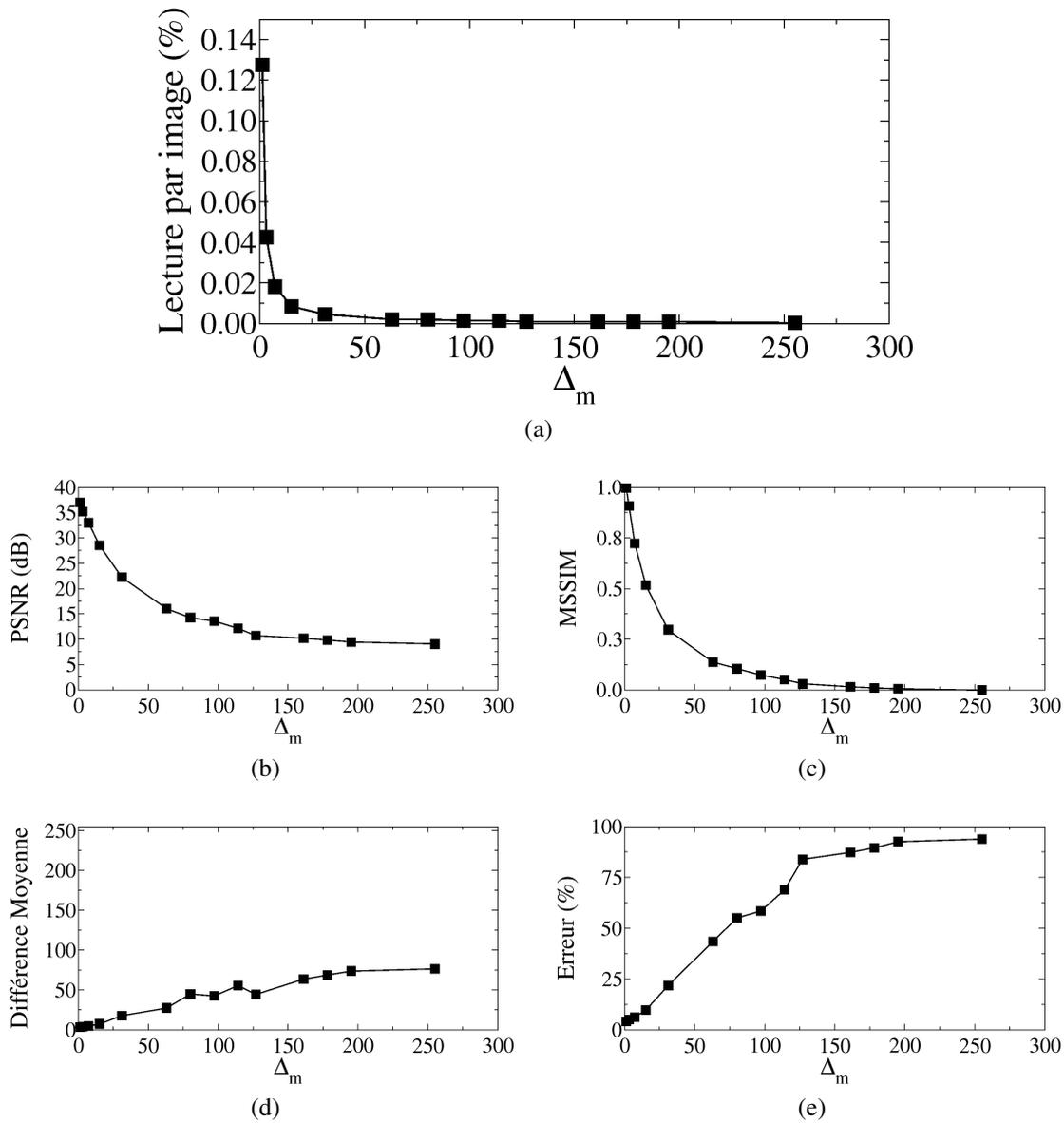


(g)  $\Delta = 64$ , PSNR = 16.05dB, MSSIM = 0.137



(h)  $\Delta = 128$ , PSNR = 10.73dB, MSSIM = 0.030

**FIGURE A.19** – Résultat des simulations MATLAB : Image N. (a) L'image de référence. (b-h) Les images résultantes des simulations suivant les valeurs de  $\Delta_m$ . (Évaluation des images résultantes Fig.A.20)



**FIGURE A.20** – Évaluation de la qualité des images résultantes de simulations de l'image N (Fig.A.19) en fonction de  $\Delta_m$ . (a) Taux de lecture par image par rapport au flot de données de lecture de l'image de référence. Les images résultantes sont évaluées en utilisant les paramètres : (b) Peak signal-to-noise ratio : PSNR (dB), (c) Mean Structural Similarity : MSSIM, (d) Différence moyenne et (e) Erreur (%) entre l'image de référence et les images résultantes suivant les valeurs de  $\Delta_m$ .



# B | SUITE DES SIMULATIONS DU CHAPITRE 8

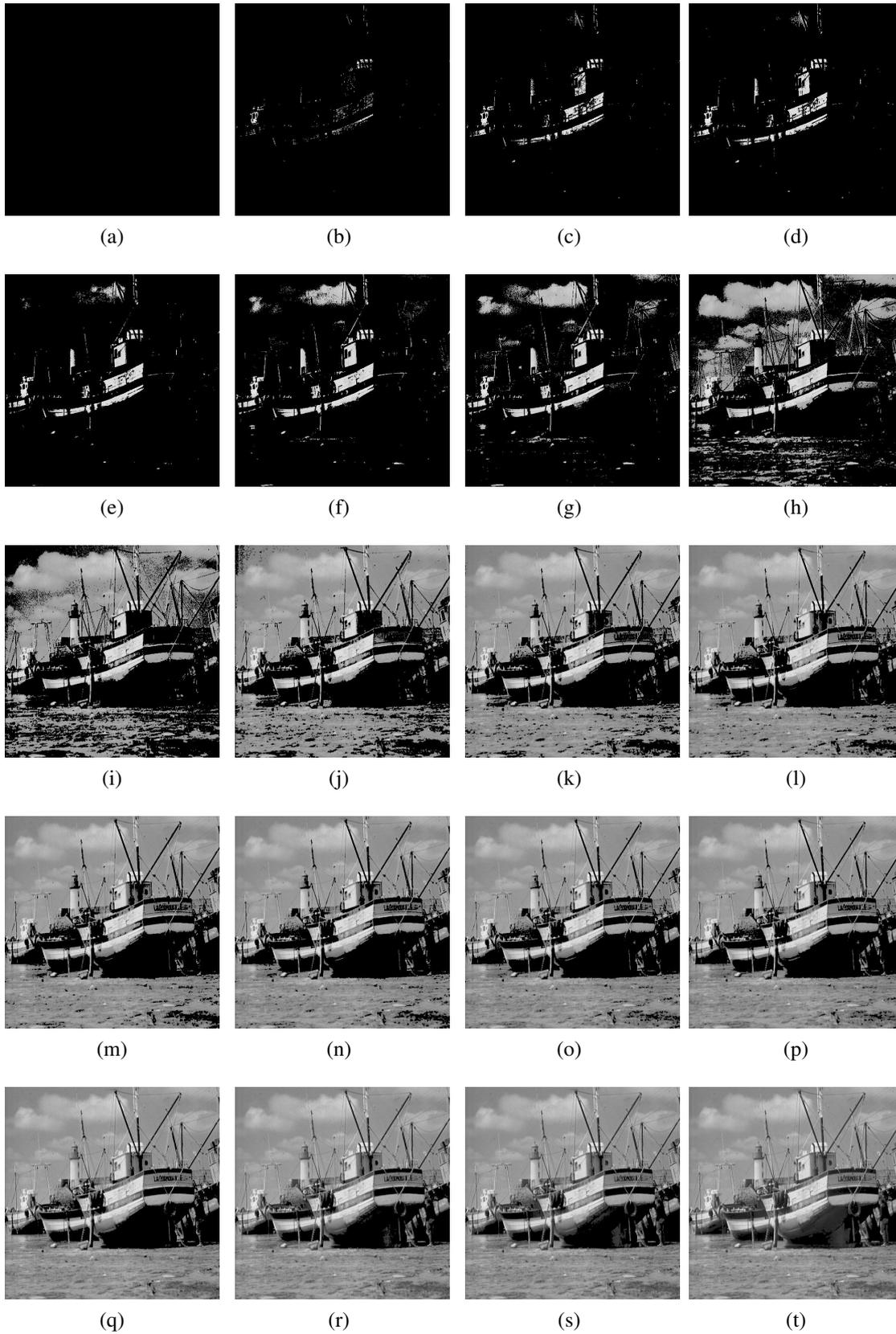
Cet annexe comporte les images intermédiaires résultantes des simulations niveau *netlist* effectuées dans le chapitre 8. La qualité des images compressées est évaluée grâce à des paramètres de qualité : PSNR, MSSIM, l'erreur introduite par la compression ainsi que la taux de lecture nécessaire suivant le protocole de lecture proposé au cours de cette thèse. Les valeurs des paramètres de qualité sont groupés dans le tableau B.1.

**TABLE B.1** – Résultats des simulations niveau *netlist* des images de référence 8.9 pour  $\Delta_m = 2$

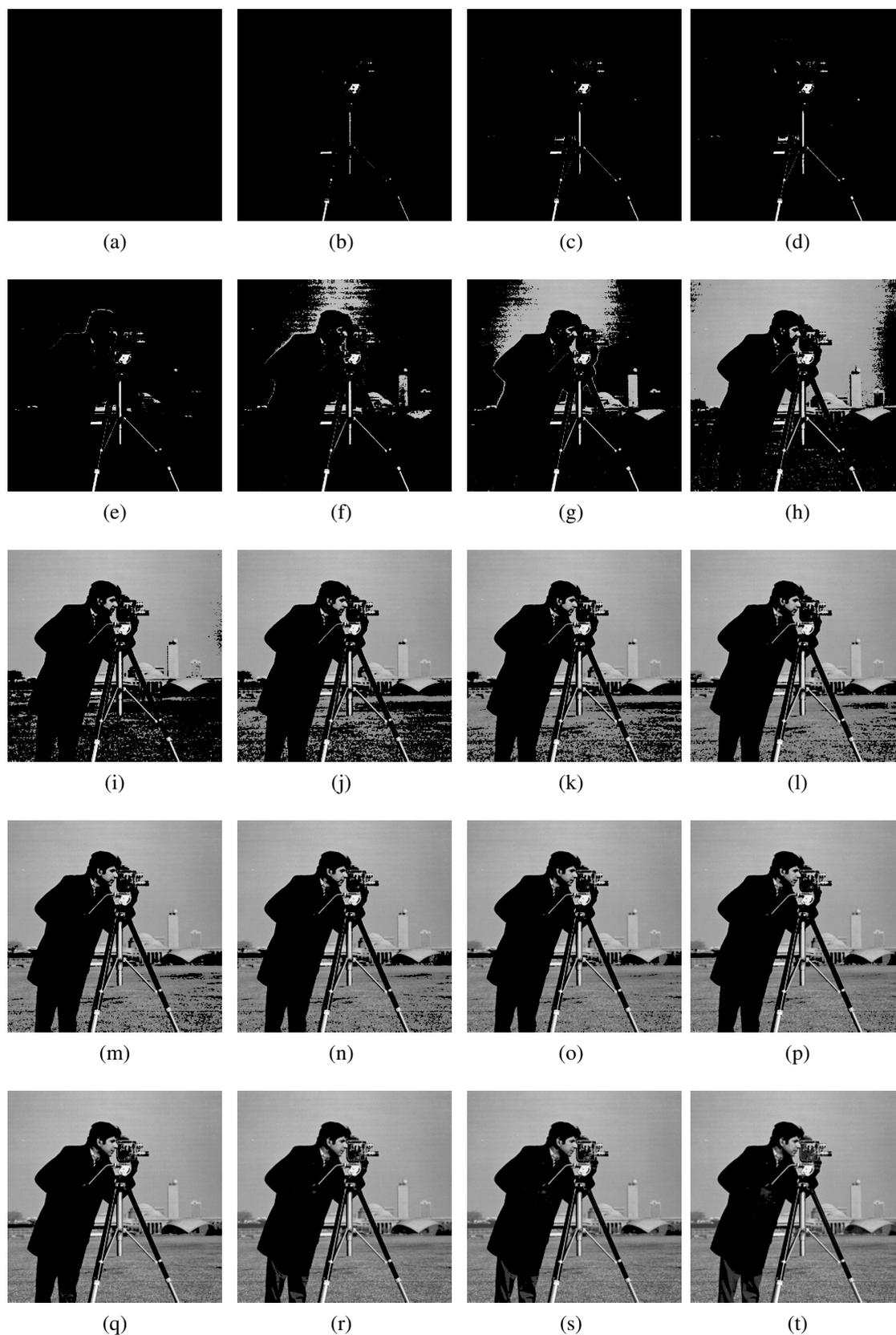
| Référence | Taux de lecture | PSNR (dB) | MSSIM | Erreur (%) |
|-----------|-----------------|-----------|-------|------------|
| 8.9a      | 0.076 %         | 25.20 dB  | 0.855 | 5.86 %     |
| 8.9b      | 0.059 %         | 25.04 dB  | 0.878 | 5.94 %     |
| 8.9c      | 0.038 %         | 24.21 dB  | 0.887 | 6.59 %     |
| 8.9d      | 0.11 %          | 27.37 dB  | 0.957 | 9.23 %     |
| 8.9e      | 0.098 %         | 28.32 dB  | 0.905 | 8.16 %     |
| 8.9f      | 0.0975 %        | 27.62 dB  | 0.961 | 7.89 %     |
| 8.9g      | 0.078 %         | 26.07 dB  | 0.906 | 5.17 %     |
| 8.9h      | 0.0887 %        | 27.31 dB  | 0.945 | 3.76 %     |
| 8.9i      | 0.045 %         | 23.05 dB  | 0.886 | 5.98 %     |



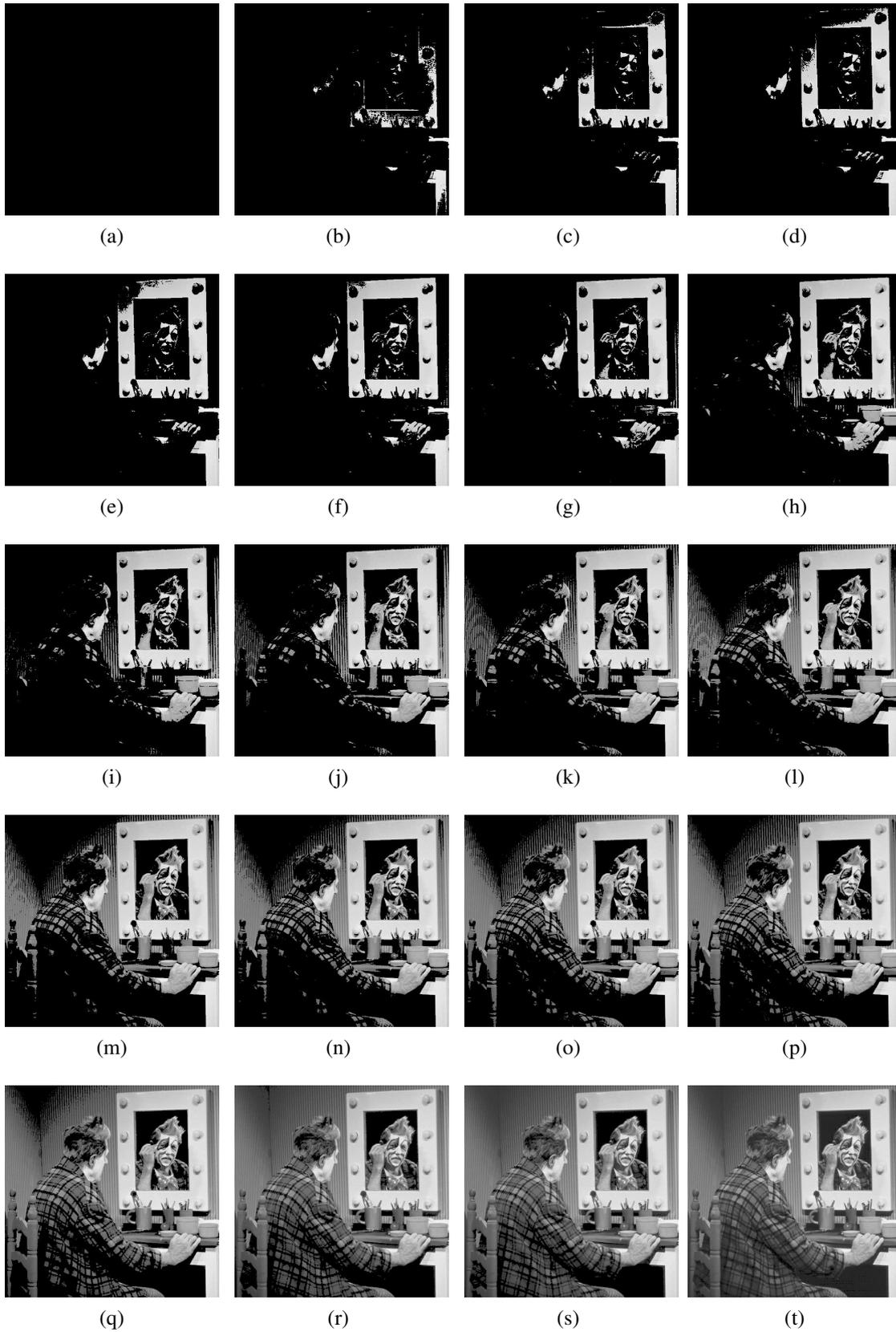
**FIGURE B.1** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9a



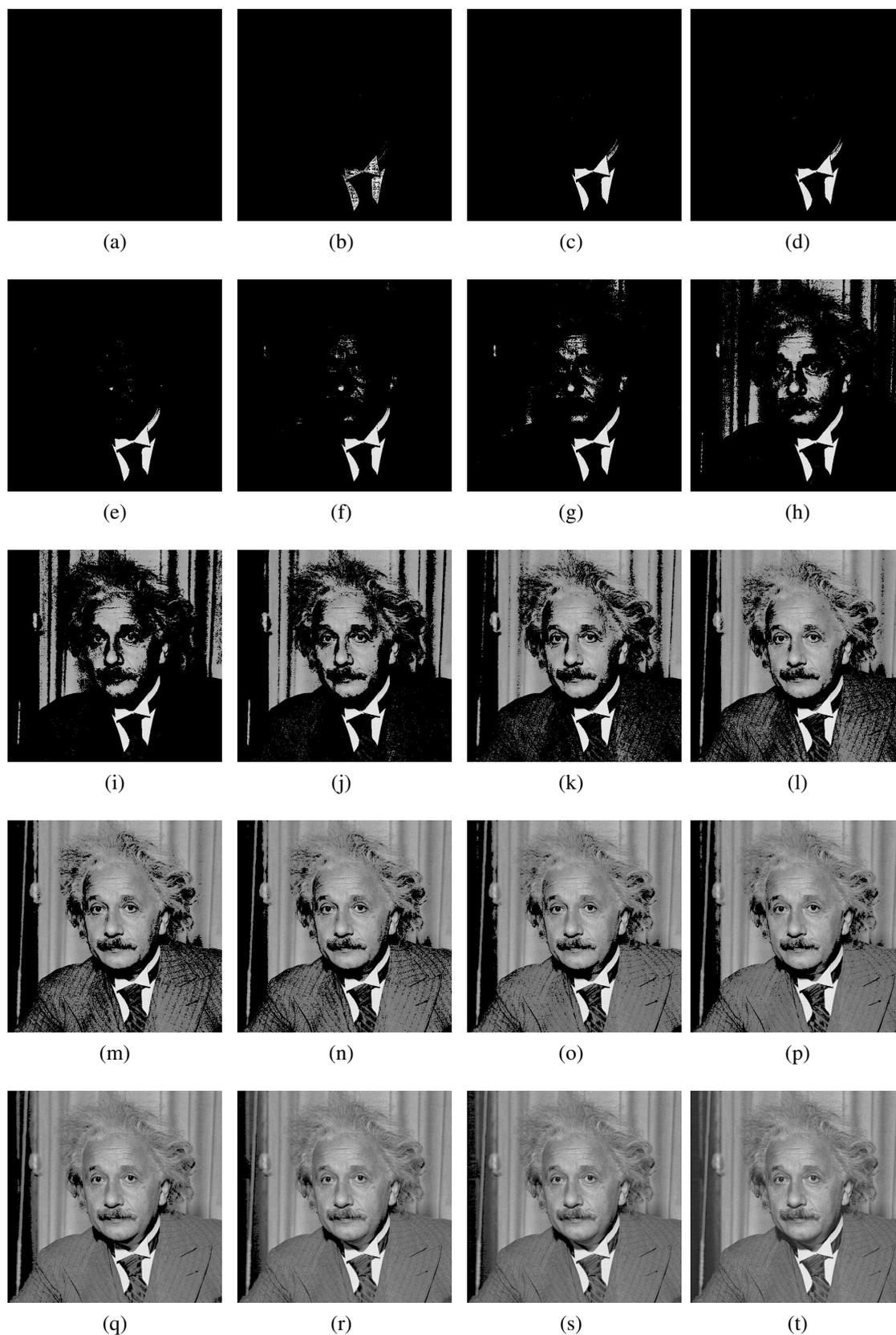
**FIGURE B.2** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9b



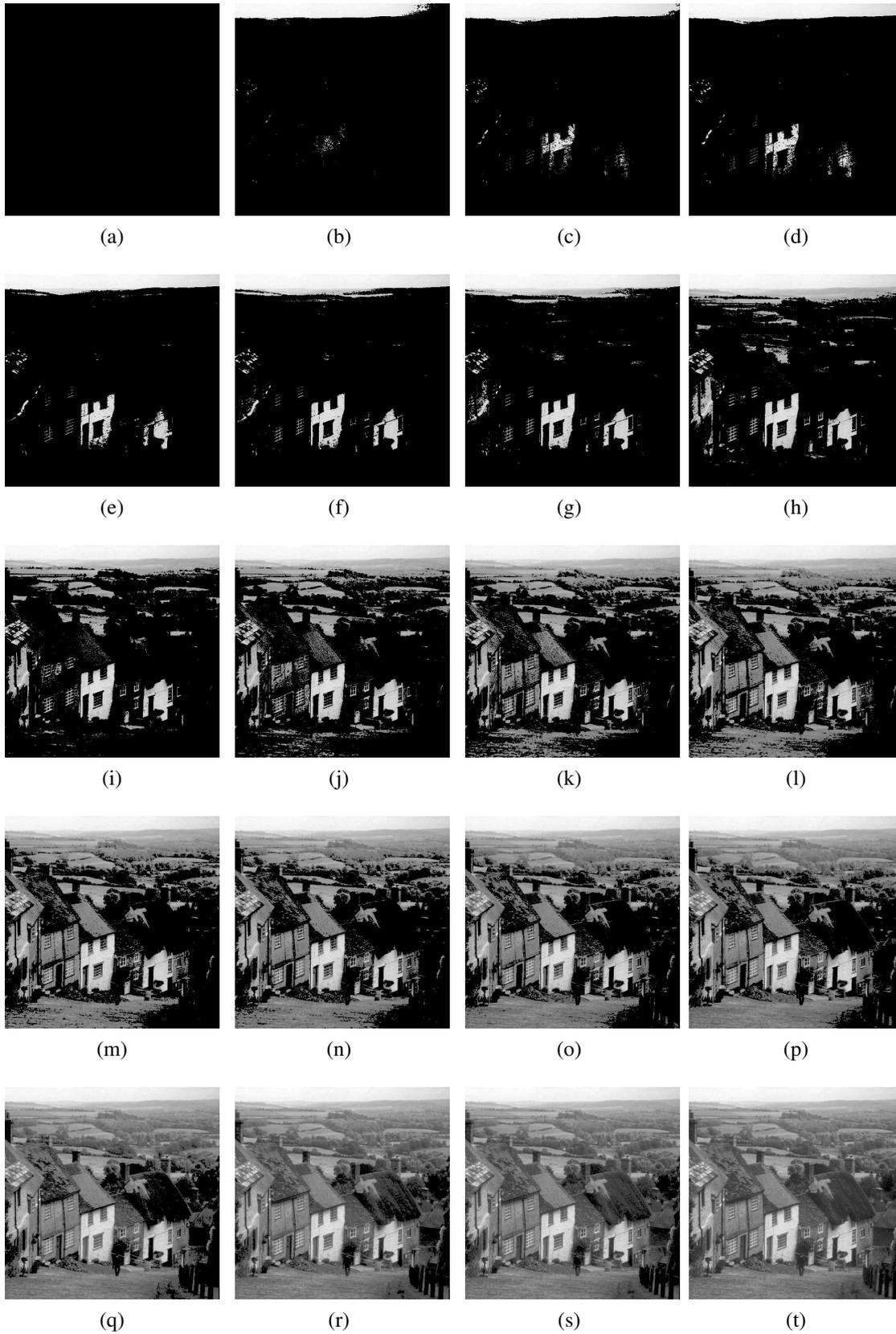
**FIGURE B.3** – Les images intermédiaires reconstruites par MATLAB associées à l'image de référence 8.9c



**FIGURE B.4** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9d



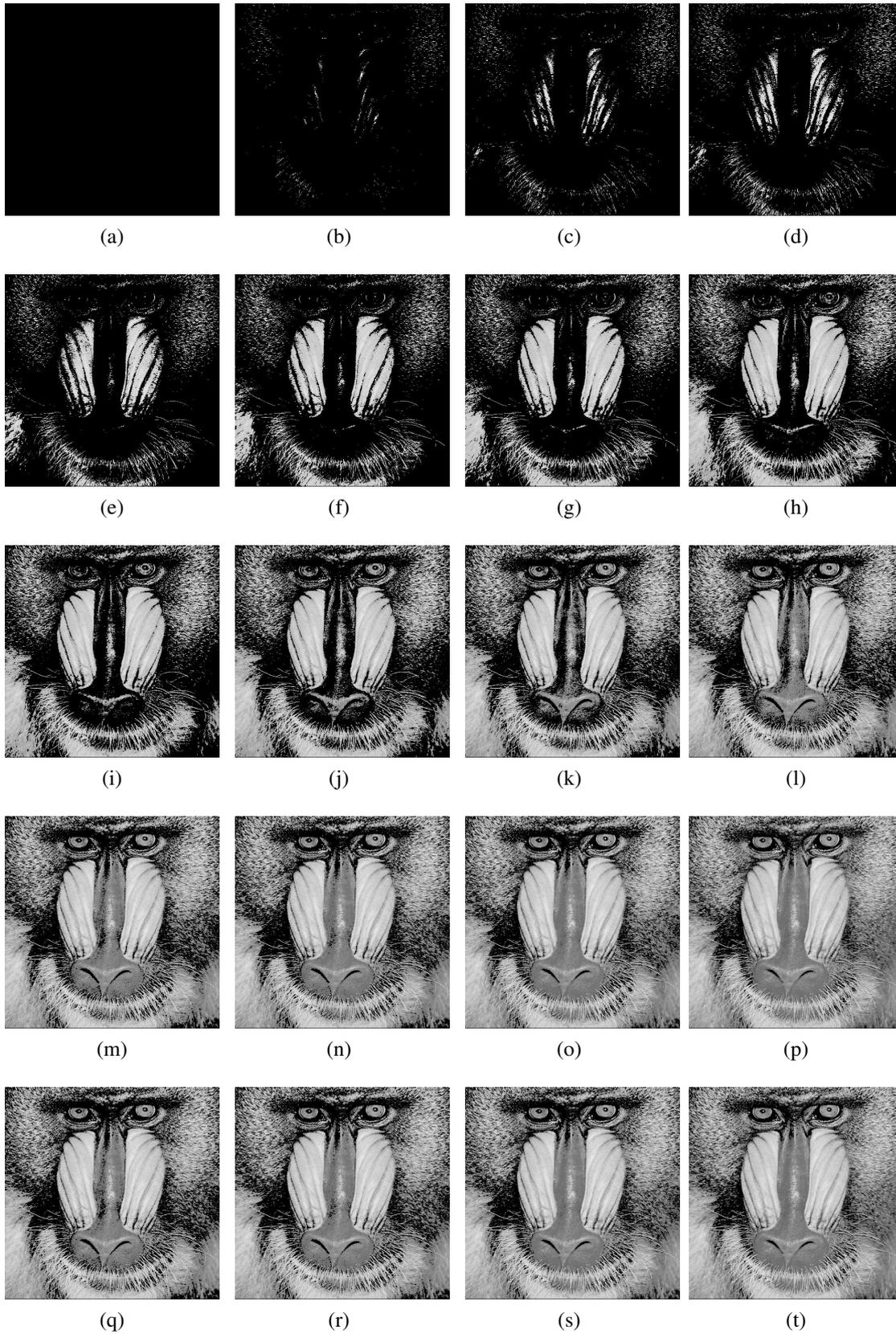
**FIGURE B.5** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9e



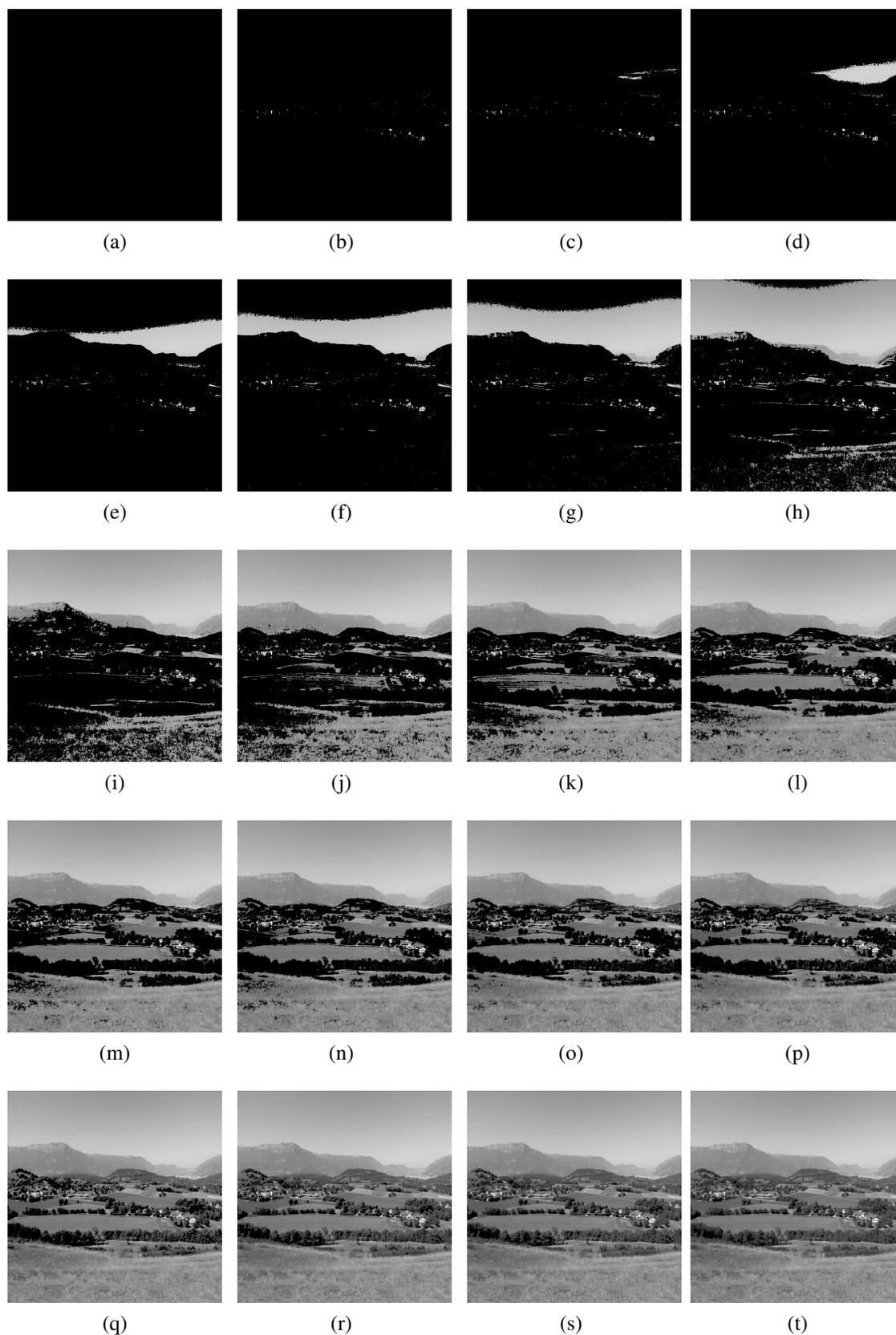
**FIGURE B.6** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9f



**FIGURE B.7** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9g



**FIGURE B.8** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9h



**FIGURE B.9** – Les images intermédiaires reconstruites par MATLAB associées à l’image de référence 8.9i



## C.1 Conférences Internationales

### 1. Design of a Fully Asynchronous Image Sensor Reading System

Amani Darwish, Leandro M. G. Rocha, Laurent Fesquet, Gilles Sicard.

The xxx Conference on Design of Circuits and Integrated Systems, (DCIS'2015)

### 2. RTL Simulation of an Asynchronous Reading Architecture for an Event-driven Image Sensor

Amani Darwish, Laurent Fesquet, Gilles Sicard.

First IEEE International Conference on Event-Based Control, Communication, and Signal Processing, (EBCCSP'2015)

### 3. Sampling Circuits for 1D and 2D Sensors for Low-Power Purpose

Laurent Fesquet, Amani Darwish, Gilles Sicard.

The 11th International Conference on Sampling Theory and Applications, (SampTA'2015)

### 4. 1-level Crossing Sampling Scheme for Low Data Rate Image Sensors

Amani Darwish, Laurent Fesquet, Gilles Sicard.

The 12th IEEE International New Circuits and Systems Conference (NEWCAS'14), 2014

### 5. Low data rate architecture for smart image sensor

Amani Darwish, Gilles Sicard, Laurent Fesquet.

Image Sensors and Imaging Systems, San Francisco, California, USA, 2014

## C.2 Conférences Invitées

### 1. Low-power Event-driven Image Sensor

Laurent Fesquet, Amani Darwish, Gilles Sicard

Keynote talk, First International Conference on Advances in Signal, Image and Video Processing (SIGNAL'16), Lisbon, Portugal, 2016

**2. Event-based design for mitigating energy in electronic systems**

Laurent Fesquet, Jean Simatic, Amani Darwish, Abdelkarim Cherkaoui

Keynote talk, OAGM & ARW Joint Workshop on "Computer Vision and Robotics", Wels, AUSTRIA, 2016

**3. Mitigating the data-deluge by an adequate sampling for low-power systems**

Laurent Fesquet, Tugdual Le Pelleter, Amani Darwish, Taha Beyrouthy, Brigitte Bidegaray-Fesquet

The 5th International Conference on Computational Harmonic Analysis, 2014

## **C.3 Conférences Nationales**

**1. Capteur d'image CMOS asynchrone événementiel**

Amani Darwish, Laurent Fesquet, Gilles Sicard.

Journées Nationales du Réseau Doctoral en Micro-nanoélectronique, (JNRDM'2015)

## BIBLIOGRAPHIE

- [AAH05] M AZADMEHR, J P ABRAHAMSEN et P HAFLIGER. “A foveated AER imager chip”. In : *Circuits and Systems, ...* 1 (2005), p. 2–5. URL : <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=1465196> (cf. p. 29).
- [Abb+13] Hassan ABBASS et al. “Novel mixed design of tone mapping technique for HDR CMOS image sensor”. In : *2013 25th International Conference on Microelectronics (ICM)*. IEEE, 2013, p. 1–4. ISBN : 978-1-4799-3570-3. DOI : [10.1109/ICM.2013.6735016](https://doi.org/10.1109/ICM.2013.6735016). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6735016> (cf. p. 21).
- [Abb14] Hassan ABBASS. “Contrôle Adaptatif Local dans un Capteur de Vision CMOS”. Thèse de doct. Université de Grenoble, 2014 (cf. p. 21).
- [Aes06] Fabien AESCHLIMANN. “Non uniformly sampled signal processing : algorithm and architecture”. Thèse de doct. Institut National Polytechnique de Grenoble, 2006. URL : <http://tel.archives-ouvertes.fr/tel-00011758/> (cf. p. 49, 53, 54).
- [Aka+06] N. AKAHANE et al. “A 200dB Dynamic Range Iris-less CMOS Image Sensor with Lateral Overflow Integration Capacitor using Hybrid Voltage and Current Readout Operation”. In : *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*. IEEE, 2006, p. 1161–1170. ISBN : 1-4244-0079-1. DOI : [10.1109/ISSCC.2006.1696161](https://doi.org/10.1109/ISSCC.2006.1696161). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1696161> (cf. p. 24).
- [All+05] Emmanuel ALLIER et al. “Asynchronous level crossing analog to digital converters”. In : *Measurement* 37.4 (2005), p. 296–309. ISSN : 02632241. DOI : [10.1016/j.measurement.2005.03.002](https://doi.org/10.1016/j.measurement.2005.03.002) (cf. p. 59, 60).
- [All03] Emmanuel ALLIER. “Interface Analogique Numérique Asynchrone : Une Nouvelle Classe de Convertisseurs Basés sur la Quantification du temps”. Thèse de doct. Institut national polytechnique de grenoble, 2003. URL : <http://hal.inria.fr/docs/00/04/58/61/PDF/tel-00004027.pdf> (cf. p. 47, 59, 60).

- [AMA06] F. AKOPYAN, R. MANOHAR et A.B. APSEL. “A Level-Crossing Flash Asynchronous Analog-to-Digital Converter”. In : *12th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'06)*. IEEE, 2006, p. 12–22. ISBN : 0-7695-2498-2. DOI : [10.1109/ASYNC.2006.5](https://doi.org/10.1109/ASYNC.2006.5). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1595684> (cf. p. 59).
- [Amh+11] Hawraa AMHAZ et al. “An improved smart readout technique based on temporal redundancies suppression designed for logarithmic CMOS image sensor”. In : *2011 18th IEEE International Conference on Electronics, Circuits, and Systems (2011)*, p. 472–475. DOI : [10.1109/ICECS.2011.6122315](https://doi.org/10.1109/ICECS.2011.6122315). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6122315> (cf. p. 22).
- [Amh12] Hawraa AMHAZ. “Traitement d’images bas niveau intégré dans un capteur de vision CMOS”. Thèse de doct. Université de Grenoble, 2012. URL : <https://tel.archives-ouvertes.fr/tel-00838399/> (cf. p. 20, 22).
- [ATS09] R. AGARWAL, M. TRAKIMAS et S. SONKUSALE. “Adaptive asynchronous analog to digital conversion for compressed biomedical sensing”. In : *2009 IEEE Biomedical Circuits and Systems Conference*. IEEE, 2009, p. 69–72. ISBN : 978-1-4244-4917-0. DOI : [10.1109/BIOCAS.2009.5372083](https://doi.org/10.1109/BIOCAS.2009.5372083). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5372083> (cf. p. 59).
- [AZ26] E. D. ADRIAN et Yngve ZOTTERMAN. “The impulses produced by sensory nerve-endings”. In : *The Journal of physiology* 61.1 (1926). DOI : [16993776](https://doi.org/10.1111/j.16993776.1926.tb01478.x). URL : <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1514782/> (cf. p. 29).
- [Bae06] Richard L. BAER. “A model for dark current characterization and simulation”. In : *Proceedings of SPIE*. T. 6068. 2006, p. 606805–606805–12. ISBN : 0819461083. DOI : [10.1117/12.639844](https://doi.org/10.1117/12.639844). URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=729134> (cf. p. 21).
- [Beu70] F. BEUTLER. “Alias-free randomly timed sampling of stochastic processes”. In : *IEEE Transactions on Information Theory* 16.2 (1970), p. 147–152. ISSN : 0018-9448. DOI : [10.1109/TIT.1970.1054435](https://doi.org/10.1109/TIT.1970.1054435). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1054435> (cf. p. 49).
- [Bil07] Ivars BILINSKIS. *Digital Alias-free Signal Processing*. John Wiley & Sons, 2007, p. 454. ISBN : 0470511621 (cf. p. 49, 50, 53, 54, 56, 58).

- [BM92] Ivars BILINSKIS et Arnolds MIKELSONS. *Randomized Signal Processing*. 1992, p. 665. ISBN : 978-0137510740 (cf. p. 49, 53).
- [Boa00] Kwabena A. BOAHEN. “Point-to-point connectivity between neuromorphic chips using address events”. In : *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 47.5 (2000), p. 416–434. ISSN : 10577130. DOI : [10.1109/82.842110](https://doi.org/10.1109/82.842110). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=842110> (cf. p. 26).
- [Boa04a] Kwabena A. BOAHEN. “A Burst-Mode Word-Serial Address-Event Link - I : Transmitter Design”. In : *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.7 (2004), p. 1269–1280. ISSN : 1549-8328. DOI : [10.1109/TCSI.2004.830703](https://doi.org/10.1109/TCSI.2004.830703). URL : <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1310498> (cf. p. 28).
- [Boa04b] Kwabena A. BOAHEN. “A Burst-Mode Word-Serial Address-Event Link - II : Receiver Design”. In : *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.7 (2004), p. 1281–1291. ISSN : 1057-7122. DOI : [10.1109/TCSI.2004.830702](https://doi.org/10.1109/TCSI.2004.830702). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1310499> (cf. p. 28).
- [Boa04c] Kwabena A. BOAHEN. “A Burst-Mode Word-Serial Address-Event Link - III: Analysis and Test Results”. In : *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.7 (2004), p. 1292–1300. ISSN : 1057-7122. DOI : [10.1109/TCSI.2004.830701](https://doi.org/10.1109/TCSI.2004.830701). URL : <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1310500> (cf. p. 28).
- [BOF10] Peter A. BEEREL, RO OZDAG et M FERRETTI. *A designer’s guide to asynchronous VLSI*. 2010. ISBN : 9780521872447. URL : <http://www.cambridge.org/us/academic/subjects/engineering/circuits-and-systems/designers-guide-asynchronous-vlsi?format=HB> (cf. p. 36, 37, 43, 45, 66).
- [But+07] Daniel A BUTTS et al. “Temporal precision in the neural code and the timescales of natural vision”. In : *Nature* 449.7158 (2007), p. 92–95. ISSN : 0028-0836. DOI : [10.1038/nature06105](https://doi.org/10.1038/nature06105). URL : <http://www.nature.com/doifinder/10.1038/nature06105> (cf. p. 29).
- [CA04] Eugenio CULURCIELLO et A.G. ANDREE. “ALOHA CMOS imager”. In : *2004 IEEE International Symposium on Circuits and Systems*. T. 4. IEEE, 2004, p. IV–956–9. ISBN : 0-7803-8251-X. DOI : [10.1109/](https://doi.org/10.1109/)

- ISCAS.2004.1329164. URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1329164> (cf. p. 28).
- [Car+12] Tony Chan CARUSONE et al. *Analog Integrated Circuits Design 2nd*. John Wiley & Sons, Inc., 2012, p. 822. ISBN : 9780470770108. URL : <http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract> (cf. p. 53, 56).
- [CC07] Bhaskar CHOUBEY et Steve COLLINS. “Models for Pixels With Wide-Dynamic-Range Combined Linear and Logarithmic Response”. In : *IEEE Sensors Journal* 7.7 (2007), p. 1066–1072. ISSN : 1530-437X. DOI : 10.1109/JSEN.2007.895959. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4207409> (cf. p. 16, 17).
- [CEC04] Eugenio CULURCIELLO et Ralph ETIENNE-CUMMINGS. “Second generation of high dynamic range, arbitrated digital imager”. In : *2004 IEEE International Symposium on Circuits and Systems* (2004), p. IV–828–31. DOI : 10.1109/ISCAS.2004.1329132. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1329132> (cf. p. 21, 32).
- [CECB01] Eugenio CULURCIELLO, R ETIENNE-CUMMINGS et Kwabena A. BOAHEN. “High dynamic range, arbitrated address event representation digital imager”. In : *ISCAS 2001 The 2001 IEEE International Symposium on Circuits and Systems*. T. 2. IEEE, 2001, p. 505–508. ISBN : 0-7803-6685-9. DOI : 10.1109/ISCAS.2001.921358. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=921358> (cf. p. 32).
- [CFC08] Eugenio CULURCIELLO, Zhengming FU et Eugenio CULURCIELLO. “A 1.2mW CMOS temporal-difference image sensor for sensor networks”. In : *2008 IEEE International Symposium on Circuits and Systems*. IEEE, 2008, p. 1064–1067. ISBN : 978-1-4244-1683-7. DOI : 10.1109/ISCAS.2008.4541605. URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4541605> (cf. p. 22).
- [Che+14] DG Denis Guangyin CHEN et al. “Low-Power CMOS Laser Doppler Imaging Using Non-CDS Pixel Readout and 13.6-bit SAR ADC”. In : *IEEE transactions on biomedical circuits and systems* (2014), p. 1–14. ISSN : 1940-9990. DOI : 10.1109/TBCAS.2014.2365515. URL : <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=6985687> (cf. p. 23).

- [Chi+07] Yu M. CHI et al. “CMOS Camera With In-Pixel Temporal Change Detection and ADC”. In : *IEEE Journal of Solid-State Circuits* 42.10 (2007), p. 2187–2196. ISSN : 0018-9200. DOI : [10.1109/JSSC.2007.905295](https://doi.org/10.1109/JSSC.2007.905295). URL : <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=4317686> (cf. p. 22).
- [CSS13] Ahmed CHEFI, Adel SOUDANI et Gilles SICARD. “Contribution to the design of a CMOS image sensor with low-complexity video compression for wireless sensor networks”. In : *Journal of Systems Architecture* 59.10 (2013), p. 818–825. ISSN : 13837621. DOI : [10.1016/j.sysarc.2013.07.010](https://doi.org/10.1016/j.sysarc.2013.07.010). URL : <http://linkinghub.elsevier.com/retrieve/pii/S1383762113001458> (cf. p. 22).
- [Dar+15] Amani DARWISH et al. “Design of a fully asynchronous image sensor reading system”. In : *2015 Conference on Design of Circuits and Integrated Systems (DCIS)*. IEEE, 2015, p. 1–5. ISBN : 978-1-4673-7228-2. DOI : [10.1109/DCIS.2015.7388583](https://doi.org/10.1109/DCIS.2015.7388583). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7388583> (cf. p. 32, 127).
- [Dar08] Arnaud DARMONT. “Methods to extend the dynamic range of snapshot active pixels sensors”. In : *Electronic Imaging 2008*. T. 6816. 2008, p. 681603–681603–11. DOI : [10.1117/12.761600](https://doi.org/10.1117/12.761600). URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=812309> (cf. p. 23).
- [DD03] Anh-Vu DINH-DUC. “Synthese Automatique de Circuits Asynchrones QDI”. Thèse de doct. Institut National Polytechnique de Grenoble, 2003. URL : <http://hal.inria.fr/docs/00/04/53/78/PDF/tel-00002937.pdf> (cf. p. 43).
- [DF01] C. DONOVAN et M.P. FLYNN. “A ‘digital’ 6-bit ADC in 0.25  $\mu\text{m}$  CMOS”. In : *Proceedings of the IEEE 2001 Custom Integrated Circuits Conference*. IEEE, 2001, p. 145–148. ISBN : 0-7803-6591-7. DOI : [10.1109/CICC.2001.929743](https://doi.org/10.1109/CICC.2001.929743). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=929743> (cf. p. 53).
- [DFS14] Amani DARWISH, Laurent FESQUET et Gilles SICARD. “1-Level crossing sampling scheme for low data rate image sensors”. In : *New Circuits and Systems ...* IEEE Computer Society (2014), p. 289–292. URL : <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=6934039> (cf. p. 32, 61, 127).

- [DFS15] Amani DARWISH, Laurent FESQUET et Gilles SICARD. “RTL simulation of an asynchronous reading architecture for an event-driven image sensor”. In : *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*. IEEE, 2015, p. 1–4. ISBN : 978-1-4673-7888-8. DOI : [10.1109/EBCCSP.2015.7300666](https://doi.org/10.1109/EBCCSP.2015.7300666). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7300666> (cf. p. 32, 127).
- [DSF14] Amani DARWISH, Gilles SICARD et Laurent FESQUET. “Low data rate architecture for smart image sensor”. In : ... *Imaging* (2014), p. 902206. DOI : [10.1117/12.2036851](https://doi.org/10.1117/12.2036851). URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1840905> (cf. p. 127).
- [Ebe91] Jo C. EBERGEN. “A formal approach to designing delay-insensitive circuits”. In : *Distributed Computing* 5.3 (1991), p. 107–119. ISSN : 0178-2770. DOI : [10.1007/BF02252954](https://doi.org/10.1007/BF02252954). URL : <http://link.springer.com/10.1007/BF02252954> (cf. p. 42).
- [EYF99] Abbas EL GAMAL, David X. D. YANG et Boyd A. FOWLER. “Pixel Level Processing — Why, What, and How?” In : *Electronic Imaging’99*. 1999, p. 2–13. ISBN : 6507238473. DOI : [10.1117/12.342849](https://doi.org/10.1117/12.342849). URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=979328> (cf. p. 22).
- [FDS15] Laurent FESQUET, Amani DARWISH et Gilles SICARD. “Sampling circuits for 1D and 2D sensors for low-power purpose”. In : *2015 International Conference on Sampling Theory and Applications (SampTA)*. IEEE, 2015, p. 430–434. ISBN : 978-1-4673-7353-1. DOI : [10.1109/SAMP TA.2015.7148927](https://doi.org/10.1109/SAMP TA.2015.7148927). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7148927> (cf. p. 32, 59, 127).
- [Fes+14] Laurent FESQUET et al. “Mitigating the data-deluge by an adequate sampling for low-power systems”. In : *5th International Conference on Computational Harmonic Analysis* (2014) (cf. p. 127).
- [FH14] Eric FOSSUM et Donald HONDONGWA. “A Review of the Pinned Photodiode for CCD and CMOS Image Sensors”. In : *IEEE Journal of the Electron Devices Society* 6734.c (2014), p. 1–1. DOI : [10.1109/JEDS.2014.2306412](https://doi.org/10.1109/JEDS.2014.2306412). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6742594> (cf. p. 11).
- [Fon99] Ludovic FONTAINE. “Traitement des signaux à échantillonnage irrégulier Application au suivi temporel de paramètres cardiaques”. Thèse de doct. Institut National Polytechnique de Lorraine, 1999 (cf. p. 49, 53).

- [For14] Michael David FORREST. “Intracellular Calcium Dynamics Permit a Purkinje Neuron Model to Perform Toggle and Gain Computations Upon its Inputs”. In : *Frontiers in Computational Neuroscience* 8.86 (2014). DOI : [10.3389/fncom.2014.00086](https://doi.org/10.3389/fncom.2014.00086). URL : [http://www.frontiersin.org/computational{\\\\_}neuroscience/10.3389/fncom.2014.00086/abstract](http://www.frontiersin.org/computational{\\_}neuroscience/10.3389/fncom.2014.00086/abstract) (cf. p. 29).
- [Fos93] Eric R. FOSSUM. “Active pixel sensors: are CCDs dinosaurs?” In : *Proceedings of SPIE*. T. 1900. 1993, p. 2–14. DOI : [10.1117/12.148585](https://doi.org/10.1117/12.148585). URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1008573> (cf. p. 10, 11).
- [Fos97a] E.R. FOSSUM. “Image Capture Circuits in CMOS”. In : *Proceedings of Technical Papers. International Symposium on VLSI Technology, Systems, and Applications* 12.3 (1997). DOI : [10.1109/VTSA.1997.614726](https://doi.org/10.1109/VTSA.1997.614726) (cf. p. 16).
- [Fos97b] Eric R. FOSSUM. “CMOS active pixel image sensors”. In : *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 395.3 (1997), p. 291–297. ISSN : 01689002. DOI : [10.1016/S0168-9002\(97\)00812-7](https://doi.org/10.1016/S0168-9002(97)00812-7). URL : <http://www.sciencedirect.com/science/article/pii/S0168900297008127> (cf. p. 14).
- [Fur+07] Masanori FURUTA et al. “A High-Speed, High-Sensitivity Digital CMOS Image Sensor With a Global Shutter and 12-bit Column-Parallel Cyclic A/D Converters”. In : *IEEE Journal of Solid-State Circuits* 42.4 (2007), p. 766–774. ISSN : 0018-9200. DOI : [10.1109/JSSC.2007.891655](https://doi.org/10.1109/JSSC.2007.891655). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4140584> (cf. p. 23).
- [Gar+13] Salvador GARCIA et al. “A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning”. In : *IEEE Transactions on Knowledge and Data Engineering* 25.4 (2013), p. 734–750. ISSN : 1041-4347. DOI : [10.1109/TKDE.2012.35](https://doi.org/10.1109/TKDE.2012.35). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6152258> (cf. p. 58).
- [Gen+91] Y. GENDAI et al. “An 8b 500MHz ADC”. In : *1991 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. IEEE, 1991, p. 172–311. ISBN : 0-87942-644-6. DOI : [10.1109/ISSCC.1991.689114](https://doi.org/10.1109/ISSCC.1991.689114). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=689114> (cf. p. 58).

- [GM08] Tim GOLLISCH et Markus MEISTER. “Rapid Neural Coding in the Retina with Relative Spike Latencies”. In : *Science* 319.5866 (2008), p. 1108–1111. ISSN : 0036-8075. DOI : [10.1126/science.1149639](https://doi.org/10.1126/science.1149639). URL : <http://science.sciencemag.org/content/319/5866/1108> (cf. p. 30).
- [GMC14] Luiz Carlos GOUVEIA, Waqas MUGHAL et Bhaskar CHOUBEY. “A reconfigurable CMOS pixel for applying tone mapping on high dynamic range images”. In : *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings 1* (2014), p. 1098–1101. DOI : [10.1109/I2MTC.2014.6860912](https://doi.org/10.1109/I2MTC.2014.6860912). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6860912> (cf. p. 21).
- [Gui+07] Alexandre GUILVARD et al. “A High Dynamic Range digital LinLog CMOS image sensor architecture based on Event Readout of pixels and suitable for low voltage operation”. In : (2007), p. 34–37. URL : <http://oatao.univ-toulouse.fr/312/> (cf. p. 21).
- [Guo02] Xiaochuan GUO. “A Time-Based Asynchronous Readout CMOS Image Sensor”. Thèse de doct. University of Florida, 2002. URL : [http://etd.fcla.edu/UF/UFE0000540/guo{\\\_}x.pdf](http://etd.fcla.edu/UF/UFE0000540/guo{\_}x.pdf) (cf. p. 29, 30, 32).
- [Han+13] Ye HAN et al. “A low-power column-parallel ADC for high-speed CMOS image sensor”. In : *ISPD 2013 - Fifth International Symposium on Photoelectronic Detection and Imaging*. International Society for Optics et Photonics, 2013, 89082E. DOI : [10.1117/12.2034924](https://doi.org/10.1117/12.2034924). URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1731630> (cf. p. 18, 23).
- [Hau95] Scott HAUCK. “Asynchronous design methodologies: An overview”. In : *Proceedings of the IEEE* 83.1 (1995), p. 69–93. ISSN : 00189219. DOI : [10.1109/5.362752](https://doi.org/10.1109/5.362752). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=362752>[http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=362752](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=362752) (cf. p. 42).
- [HHK03] F HASHEMI, KH HADIDI et A KHOEI. “Design of a CMOS image sensor with pixel-level ADC in 0.35  $\mu\text{m}$  process”. In : *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03. T. 2*. IEEE, 2003, p. II–600–II–603. ISBN : 0-7803-7761-3. DOI : [10.1109/ISCAS.2003.1206045](https://doi.org/10.1109/ISCAS.2003.1206045). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1206045> (cf. p. 19).

- [HPDA12] Tien HO-PHUOC, Antoine DUPRET et Laurent ALACOQUE. “Saliency-based data compression for image sensors”. In : *2012 IEEE Sensors*. IEEE, 2012, p. 1–4. ISBN : 978-1-4577-1767-3. DOI : [10.1109/ICSENS.2012.6411255](https://doi.org/10.1109/ICSENS.2012.6411255). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6411255> (cf. p. 22).
- [Ide+07] N. IDE et al. “A Wide DR and linear response CMOS image sensor with three photocurrent integrations in photodiodes, lateral overflow capacitors and column capacitors”. In : *ESSCIRC 2007 - 33rd European Solid-State Circuits Conference*. IEEE, 2007, p. 336–339. ISBN : 978-1-4244-1125-2. DOI : [10.1109/ESSCIRC.2007.4430312](https://doi.org/10.1109/ESSCIRC.2007.4430312). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4430312> (cf. p. 24).
- [IJ01] Emmanuel IFEACHOR et Barrie JERVIS. *Digital Signal Processing: A Practical Approach (2nd Edition)*. Pearson Education, 2001, p. 933. ISBN : 0201596199 (cf. p. 49).
- [Jer77] A.J. JERRI. “The Shannon sampling theorem - Its various extensions and applications: A tutorial review”. In : *Proceedings of the IEEE* 65.11 (1977), p. 1565–1596. ISSN : 0018-9219. DOI : [10.1109/PROC.1977.10771](https://doi.org/10.1109/PROC.1977.10771). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1455040> (cf. p. 49).
- [JG93] B JÄHNE et Gerhard GOOS. *Spatio-temporal image processing: Theory and scientific applications*. Sous la dir. de Bernd JÄHNE. T. 751. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 1993. ISBN : 978-3-540-57418-7. DOI : [10.1007/3-540-57418-2](https://doi.org/10.1007/3-540-57418-2). URL : <http://link.springer.com/10.1007/3-540-57418-2> (cf. p. 22).
- [JJw14] Blake C. JACQUOT et Nathan JOHNSON-WILLIAMS. “Multiple-samples-method enabling high dynamic range imaging for high frame rate CMOS image sensor by FPGA and co-processor”. In : *Proc. SPIE* 9216 (2014), p. 921609–921609–12. DOI : [10.1117/12.2062284](https://doi.org/10.1117/12.2062284). URL : <http://dx.doi.org/10.1117/12.2062284> (cf. p. 21).
- [Kav+00] S. KAVADIAS et al. “A logarithmic response CMOS image sensor with on-chip calibration”. In : *IEEE Journal of Solid-State Circuits* 35.8 (2000), p. 1146–1152. ISSN : 0018-9200. DOI : [10.1109/4.859503](https://doi.org/10.1109/4.859503). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=859503> (cf. p. 24).
- [KB04] Alistair KITCHEN et Amine BERMAK. “PWM digital pixel sensor based on asynchronous self-resetting scheme”. In : *Electron Device Letters*, ... 25.7 (2004), p. 471–473. URL : <http://ieeexplore.org>.

- [ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=130842301308423](http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=130842301308423)  
(cf. p. 127).
- [KGG06] S. KAVUSI, K. GHOSH et Abbas El Gamal Abbas El GAMAL. “Architectures for High Dynamic Range, High Speed Image Sensor Readout Circuits”. In : *2006 IFIP International Conference on Very Large Scale Integration* (2006), p. 36–41. DOI : [10.1109/VLSISOC.2006.313269](https://doi.org/10.1109/VLSISOC.2006.313269) (cf. p. 21).
- [Kra02] J KRAMER. “An on/off transient imager with event-driven, asynchronous read-out”. In : *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*. IEEE, 2002, p. II–165–II–168. ISBN : 0-7803-7448-7. DOI : [10.1109/ISCAS.2002.1010950](https://doi.org/10.1109/ISCAS.2002.1010950). URL : <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=1010950> (cf. p. 29).
- [Lab07] Estelle LABONNE. “Contribution à la conception de capteurs de vision CMOS à grande dynamique”. Thèse de doct. Grenoble National Institute of Technology, 2007. ISBN : 9782848131047. URL : <https://tel.archives-ouvertes.fr/tel-00178598/document> (cf. p. 16, 20, 24).
- [LBFBRV14] Juan A. LEÑERO-BARDALLO, Jorge FERNÁNDEZ-BERNI et Angel RODRIGUEZ-VAZQUEZ. “Review of ADCs for imaging”. In : *IS&T/SPIE Electronic Imaging*. 2014. ISBN : 9780819499394. DOI : [10.1117/12.2041682](https://doi.org/10.1117/12.2041682). URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1840915> (cf. p. 18).
- [LDK04] Patrick LICHTSTEINER, Tobi DELBRÜCK et J KRAMER. “Improved on/off temporally differentiating address-event imager”. In : *Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems, 2004. ICECS 2004*. T. 4. IEEE, 2004, p. 211–214. ISBN : 0-7803-8715-5. DOI : [10.1109/ICECS.2004.1399652](https://doi.org/10.1109/ICECS.2004.1399652). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1399652> (cf. p. 29, 127).
- [Le+05] H.P. LE et al. “Ultra-low-power variable-resolution successive approximation ADC for biomedical application”. In : *Electronics Letters* 41.11 (2005), p. 634. ISSN : 00135194. DOI : [10.1049/el:20050490](https://doi.org/10.1049/el:20050490). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1437867> (cf. p. 57).
- [LeP15] Tugdual LEPELLETER. “Méthode de discrétisation adaptée à une logique événementielle pour l’ultra-faible consommation : application à la reconnaissance de signaux physiologiques”. Thèse de doct. Université De Grenoble, 2015 (cf. p. 59).

- [LH04] Qiang LUO et J.G. HARRIS. “A time-based CMOS image sensor”. In : *2004 IEEE International Symposium on Circuits and Systems*. T. 4. IEEE, 2004, p. IV–840–3. ISBN : 0-7803-8251-X. DOI : [10.1109/ISCAS.2004.1329135](https://doi.org/10.1109/ISCAS.2004.1329135). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1329135> (cf. p. 32).
- [Lic06a] Patrick LICHTSTEINER. “A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change”. In : *Solid-State Circuits ...* 51.7 (2006), p. 844–847. URL : [http://ieeexplore.ieee.org/xpls/abs/\\_all.jsp?arnumber=1696265](http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=1696265) (cf. p. 29, 69).
- [Lic06b] Patrick LICHTSTEINER. “An AER Temporal Contrast Vision Sensor”. Thèse de doct. Eidgenössischen Technischen Hochschule Zürich, 2006. URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.9301{\&}rep=rep1{\&}type=pdf> (cf. p. 25, 29, 127).
- [Lic08] Patrick LICHTSTEINER. “A 128 x 128 120 dB 15  $\mu$ s latency asynchronous temporal contrast vision sensor”. In : *Solid-State Circuits, IEEE ...* 43.2 (2008), p. 566–576. URL : [http://ieeexplore.ieee.org/xpls/abs/\\_all.jsp?arnumber=4444573](http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=4444573) (cf. p. 29, 69, 127).
- [Lim+11] Seunghyun LIM et al. “A 240-frames/s 2.1-Mpixel CMOS Image Sensor With Column-Shared Cyclic ADCs”. In : *IEEE Journal of Solid-State Circuits* 46.9 (2011), p. 2073–2083. ISSN : 0018-9200. DOI : [10.1109/JSSC.2011.2144010](https://doi.org/10.1109/JSSC.2011.2144010). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5771069> (cf. p. 23).
- [Liu12] Yang LIU. “The Design of a High Dynamic Range CMOS Image Sensor in 110nm Technology”. Thèse de doct. Delft University of Technology, 2012. URL : <http://repository.tudelft.nl/view/ir/uuid:3d4832fb-53f3-4997-84b5-5c5d6bd265b9/> (cf. p. 21, 23, 24).
- [LPD06] Patrick LICHTSTEINER, Christoph POSCH et Tobi DELBRÜCK. “A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change”. In : *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*. T. 20. IEEE, 2006, p. 2060–2069. ISBN : 1-4244-0079-1. DOI : [10.1109/ISSCC.2006.1696265](https://doi.org/10.1109/ISSCC.2006.1696265). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1696265> (cf. p. 29, 127).

- [Mah92] M. MAHOWALD. “VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function”. Thèse de doct. California Institute of Technology, 1992, p. 236. URL : <http://caltechcstr.library.caltech.edu/591/> (cf. p. 26, 27).
- [Mah94] Misha MAHOWALD. *An Analog VLSI System for Stereoscopic Vision*. Springer US, 1994. ISBN : 978-1-4613-6174-9. DOI : 10.1007/978-1-4615-2724-4. URL : <http://link.springer.com/10.1007/978-1-4615-2724-4> (cf. p. 26).
- [Mar+11] Malika Alami MARKTANI et al. “A successive-approximation ADC for CMOS image sensors”. In : *2011 International Conference on Multimedia Computing and Systems*. IEEE, 2011, p. 1–6. ISBN : 978-1-61284-730-6. DOI : 10.1109/ICMCS.2011.5945569. URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5945569> (cf. p. 23).
- [Mar86] Alain J. MARTIN. “Compiling communicating processes into delay-insensitive VLSI circuits”. In : *Distributed Computing* 1.4 (1986), p. 226–234. ISSN : 0178-2770. DOI : 10.1007/BF01660034. URL : <http://link.springer.com/10.1007/BF01660034> (cf. p. 43).
- [Mar87] Farokh A. MARVASTI. *A unified approach to zero-crossings and nonuniform sampling of single and multidimensional signals and systems*. 1987, p. 180. ISBN : 0961816708 (cf. p. 49).
- [Mar90] Alain J MARTIN. “The Limitations to Delay-Insensitivity in Asynchronous Circuits”. In : *Proceedings of the 6th MIT Conference on Advanced Research in VLSI* September (1990). DOI : 10.1007/978-1-4612-4476-9. URL : <http://link.springer.com/10.1007/978-1-4612-4476-9> (cf. p. 42).
- [Mar93] Alain J. MARTIN. *Synthesis of Asynchronous VLSI Circuits*. 1993. URL : <http://authors.library.caltech.edu/26746/2/postscript.pdf> (cf. p. 43).
- [Mar98] Richard James MARTIN. “Irregularly sampled signals: theories and techniques for analysis”. In : January (1998). URL : <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.1679> (cf. p. 49, 53).
- [Mat+09] Shinichiro MATSUO et al. “8.9-Megapixel Video Image Sensor With 14-b Column-Parallel SA-ADC”. In : *IEEE Transactions on Electron Devices* 56.11 (2009), p. 2380–2389. ISSN : 0018-9383. DOI : 10.1109/TED.2009.2030649. URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5280329> (cf. p. 23).

- [MB59] D E MULLER et W S BARTKY. “A Theory of Asynchronous Circuits”. In : *International Symposium on Theory of Switching*. Harvard University Press, 1959, p. 204–243 (cf. p. 38).
- [Mea89] C.A. MEAD. *Analog VLSI and neural systems*. Addison Wesley Publishing Company, 1989, p. 257–278. ISBN : 978-0201059922 (cf. p. 25).
- [Mea90] C. MEAD. “Neuromorphic electronic systems”. In : *Proceedings of the IEEE* 78.10 (1990), p. 1629–1636. DOI : 10.1109/5.58356. URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=58356> (cf. p. 25).
- [Mil65] R. E. MILLER. *Switching Theory, Volume II: Sequential Circuits and Machines*. Wiley, 1965 (cf. p. 43).
- [MT81] J. MARK et T. TODD. “A Nonuniform Sampling Approach to Data Compression”. In : *IEEE Transactions on Communications* 29.1 (1981), p. 24–32. ISSN : 0096-2244. DOI : 10.1109/TCOM.1981.1094872. URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1094872> (cf. p. 57).
- [Mul63] David Eugene MULLER. “Asynchronous logics and application to information processing”. In : *Proc. Symp. on Application of Switching Theory in Space Technology*. Sous la dir. de H. AIKEN et W. F. MAIN. Stanford University Press, 1963, p. 289–297 (cf. p. 38).
- [Mya+11] Aung MYAT et al. “Adaptive priority toggle asynchronous tree arbiter for AER-based image sensor”. In : *2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip 2* (2011), p. 66–71. DOI : 10.1109/VLSISoC.2011.6081631. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6081631> (cf. p. 27, 28).
- [NA10] Fan Z. NELSON et Suat U. AY. “Integration of a New Column-Parallel ADC Technology on CMOS Image Sensor”. In : *2010 IEEE Workshop on Microelectronics and Electron Devices* (2010), p. 1–4. DOI : 10.1109/WMED.2010.5453748. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5453748> (cf. p. 23).
- [NAA09] Fan Z. NELSON, Mustafa N. ALAM et Suat U. AY. “A Single-Slope Look-Ahead Ramp (SSLAR) ADC for Column Parallel CMOS Image Sensors”. In : *2009 IEEE Workshop on Microelectronics and Electron Devices* (2009), p. 1–4. DOI : 10.1109/WMED.2009.4816152. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4816152> (cf. p. 19, 23).

- [NS15a] Steven M NOWICK et Montek SINGH. “Asynchronous Design (Part 1): Overview and Recent Advances”. In : *IEEE Design & Test* March (2015), p. 1–1. ISSN : 2168-2356. DOI : [10.1109/MDAT.2015.2413759](https://doi.org/10.1109/MDAT.2015.2413759). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7061401> (cf. p. 37).
- [NS15b] Steven M NOWICK et Montek SINGH. “Asynchronous Design (Part 2): Systems and Methodologies”. In : *IEEE Design & Test* 32.3 (2015), p. 19–28. ISSN : 2168-2356. DOI : [10.1109/MDAT.2015.2413757](https://doi.org/10.1109/MDAT.2015.2413757). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7061445> (cf. p. 37).
- [OE13] Yusuke OIKE et Abbas EL GAMAL. “CMOS Image Sensor With Per-Column  $\Sigma\Delta$  ADC and Programmable Compressed Sensing”. In : *IEEE Journal of Solid-State Circuits* 48.1 (2013), p. 318–328. ISSN : 0018-9200. DOI : [10.1109/JSSC.2012.2214851](https://doi.org/10.1109/JSSC.2012.2214851). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6323048> (cf. p. 19).
- [OG14] Piotr OTFINOWSKI et Pawel GRYBOS. “A 7-bit 500kS/s 1V micro-power SAR A/D converter for pixel systems”. In : *Microelectronics Journal* (2014), p. 1–5. ISSN : 00262692. DOI : [10.1016/j.mejo.2014.04.026](https://doi.org/10.1016/j.mejo.2014.04.026). URL : <http://linkinghub.elsevier.com/retrieve/pii/S0026269214001281> (cf. p. 18, 23).
- [OPS48] B.M. OLIVER, J.R. PIERCE et C.E. SHANNON. “The Philosophy of PCM”. In : *Proceedings of the IRE* 36.11 (1948), p. 1324–1331. ISSN : 0096-8390. DOI : [10.1109/JRPROC.1948.231941](https://doi.org/10.1109/JRPROC.1948.231941). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1697556> (cf. p. 53, 54).
- [Par+14] Jongkil PARK et al. “A 12.6 mW 8.3 Mevents/s contrast detection 128 x 128 imager with 75 dB intra-scene DR asynchronous random-access digital readout”. In : ... *Circuits and Systems* ... (2014), p. 6–9. DOI : [10.1109/BioCAS.2014.6981788](https://doi.org/10.1109/BioCAS.2014.6981788). URL : <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6981788> (cf. p. 29).
- [PB68] Donald H. PERKEL et Theodore Holmes BULLOCK. “Neural coding”. In : *Neurosciences Research Program Bulletin* Vol 6 (3) (1968), p. 221–348 (cf. p. 29).
- [Pha] *Phantom Camera Products*. URL : <http://www.visionresearch.com/Products/High-Speed-Cameras/> (cf. p. 22).

- [PMW08] Christoph POSCH, Daniel MATOLIN et Rainer WOHLGENANNT. “An asynchronous time-based image sensor”. In : *2008 IEEE International Symposium on Circuits and Systems*. IEEE, 2008, p. 2130–2133. ISBN : 978-1-4244-1683-7. DOI : [10.1109/ISCAS.2008.4541871](https://doi.org/10.1109/ISCAS.2008.4541871). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4541871> (cf. p. 29, 32, 69, 127).
- [PMW10] Christoph POSCH, Daniel MATOLIN et Rainer WOHLGENANNT. “High-DR frame-free PWM imaging with asynchronous AER intensity encoding and focal-plane temporal redundancy suppression”. In : *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (2010)*, p. 2430–2433. DOI : [10.1109/ISCAS.2010.5537150](https://doi.org/10.1109/ISCAS.2010.5537150). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5537150> (cf. p. 21, 29, 32, 61, 69).
- [PMW11] Christoph POSCH, Daniel MATOLIN et Rainer WOHLGENANNT. “A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS”. In : *IEEE Journal of Solid-State Circuits* 46.1 (2011), p. 259–275. ISSN : 0018-9200. DOI : [10.1109/JSSC.2010.2085952](https://doi.org/10.1109/JSSC.2010.2085952). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5648367> (cf. p. 17, 21, 29, 32, 69, 127).
- [Pos+07] Christoph POSCH et al. “A Dual-Line Optical Transient Sensor with On-Chip Precision Time-Stamp Generation”. In : *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. T. 13. 2. IEEE, 2007, p. 500–618. ISBN : 1-4244-0852-0. DOI : [10.1109/ISSCC.2007.373513](https://doi.org/10.1109/ISSCC.2007.373513). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4242484> (cf. p. 69).
- [PRJ07] Dongwon PARK, Jehyuk RHEE et Youngjoong JOO. “A wide dynamic-range CMOS image sensor using self-reset technique”. In : *IEEE Electron Device Letters* 28.10 (2007), p. 890–892. ISSN : 07413106. DOI : [10.1109/LED.2007.905396](https://doi.org/10.1109/LED.2007.905396) (cf. p. 87).
- [PSG14] Christoph POSCH et Teresa SERRANO-GOTARREDONA. “Retinomorph Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output”. In : *ieeexplore.ieee.org* 102.10 (2014), p. 1470–1484. ISSN : 0018-9219. URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6887319> (cf. p. 24, 29).
- [PT07] Arnaud PEIZERAT et Michael TCHAGASPANIAN. “An analog counter architecture for pixel-level ADC”. In : *2007 International ...* (2007) (cf. p. 19).

- [Qai09] Saeed MIAN QAISAR. “Echantillonnage et Traitement Conditionnes par le Signal : Une Approche Prometteuse Pour des Traitements Efficace à Pas Adaptatifs”. Thèse de doct. Institut National Polytechnique de Grenoble, 2009. ISBN : 9782848131320 (cf. p. 59).
- [QGH04] Xin QI, Xiaochuan GUO et JG G John G HARRIS. “A time-to-first spike CMOS imager”. In : *Circuits and Systems, 2004. ISCAS'04. ...* (2004), p. 8–11. URL : [http://ieeexplore.ieee.org/xpls/abs/{\\\_}all.jsp?arnumber=1329131](http://ieeexplore.ieee.org/xpls/abs/{\_}all.jsp?arnumber=1329131) (cf. p. 32).
- [Rez04] A REZZAG. “Synthèse logique de circuits asynchrones micropipeline”. Thèse de doct. Institut National Polytechnique de Grenoble, 2004. URL : <http://tel.archives-ouvertes.fr/tel-00008398/> (cf. p. 43, 45, 113).
- [RH08] David M. ROUSE et Sheila S. HEMAMI. “Understanding and simplifying the structural similarity metric”. In : *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, p. 1188–1191. ISBN : 978-1-4244-1765-0. DOI : 10.1109/ICIP.2008.4711973. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4711973> (cf. p. 98, 102).
- [Rig02] Jean-Baptiste RIGAUD. “Specification de Bibliothèques pour la Synthèse de Circuits Asynchrones”. Thèse de doct. 2002 (cf. p. 45).
- [Roa+14] Giuseppe ROA et al. “Designing ultra-low power systems with non-uniform sampling and event-driven logic”. In : *Proceedings of the 27th Symposium on Integrated Circuits and Systems Design - SBCCI '14*. New York, New York, USA : ACM Press, 2014, p. 1–6. ISBN : 9781450331562. DOI : 10.1145/2660540.2660973 (cf. p. 59).
- [SB05] Chen SHOUSHUN et A. BERMAK. “A Low Power CMOS Imager based on Time-to-First-Spike encoding and Fair AER”. In : *2005 IEEE International Symposium on Circuits and Systems* (2005), p. 5306–5309. DOI : 10.1109/ISCAS.2005.1465833. URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1465833> (cf. p. 29, 32).
- [SB06] Chen SHOUSHUN et A BERMAK. “A second generation time-to-first-spike pixel with asynchronous self power-off”. In : *Circuits and Systems, 2006. ISCAS ...* (2006). URL : [http://ieeexplore.ieee.org/xpls/abs/{\\\_}all.jsp?arnumber=1693078](http://ieeexplore.ieee.org/xpls/abs/{\_}all.jsp?arnumber=1693078) (cf. p. 29, 32, 127).

- [SB07] Chen SHOUSHUN et Amine BERMAK. “Arbitrated time-to-first spike CMOS image sensor with on-chip histogram equalization”. In : *Very Large Scale Integration (VLSI) ...* 15.3 (2007), p. 346–357. ISSN : 10638210. DOI : [10.1109/TVLSI.2007.893624](https://doi.org/10.1109/TVLSI.2007.893624). URL : [http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=4154777](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=4154777) (cf. p. 29, 32, 127).
- [Sch+02] Olaf SCHREY et al. “A 1 K x 1 K high dynamic range CMOS image sensor with on-chip programmable region-of-interest readout”. In : *IEEE Journal of Solid-State Circuits* 37.7 (2002), p. 911–915. ISSN : 0018-9200. DOI : [10.1109/JSSC.2002.1015690](https://doi.org/10.1109/JSSC.2002.1015690). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1015690> (cf. p. 24).
- [SF01] Jens SPARSØ et Steve FURBER. “Principles of Asynchronous Circuit Design— A Systems Perspective”. In : *Kluwer Academic Publishers, ...* (2001). URL : <http://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:PRINCIPLES+OF+ASYNCHRONOUS+CIRCUIT+DESIGN+?+A+Systems+Perspective{\#}2> (cf. p. 37, 45, 66).
- [Sha49] C.E. SHANNON. “Communication in the Presence of Noise”. In : *Proceedings of the IRE* 37.1 (1949), p. 10–21. ISSN : 0096-8390. DOI : [10.1109/JRPROC.1949.232969](https://doi.org/10.1109/JRPROC.1949.232969). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1697831> (cf. p. 49, 51).
- [Sho07] Chen SHOUSHUN. “Time domain CMOS image sensor: From photodetection to on-chip image processing”. In : *ProQuest Dissertations and Theses* (2007), 145–145 p. URL : <http://repository.ust.hk/ir/Record/1783.1-2844> (cf. p. 28, 32).
- [SI10] Muung SHIN et M IKEBE. “Column parallel single-slope ADC with time to digital converter for CMOS imager”. In : *Electronics, Circuits, and ...* (2010), p. 863–866. URL : [http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=5724649](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=5724649) (cf. p. 18, 23).
- [Sic99] Gilles SICARD. “De la biologie au silicium : une rétine bioinspirée analogique pour un capteur de vision "intelligent" adaptatif”. Thèse de doct. Institut Polytechnique de Grenoble, 1999. URL : <http://www.theses.fr/1999INPG0020> (cf. p. 16).
- [Siv91] Massimo Antonio SIVILOTTI. “Wiring Considerations in Analog VLSI Systems, with Application to Field-Programmable Networks”. Thèse de doct. California Institute of Technology, 1991, p. 147 (cf. p. 26, 27).

- [SKK12] MS SHIN, JB KIM et MK KIM. “A 1.92-megapixel CMOS image sensor with column-parallel low-power and area-efficient SA-ADCs”. In : *Electron Devices, IEEE* ... 59.6 (2012), p. 1693–1700. URL : [http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=6178785](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=6178785) (cf. p. 19).
- [SLB11] Chao SHI, MK LAW et Amine BERMAK. “A novel asynchronous pixel for an energy harvesting CMOS image sensor”. In : *Very Large Scale Integration (VLSI)* ... 19.1 (2011), p. 118–129. URL : [http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=5280193](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=5280193) (cf. p. 29, 32).
- [Sno+07a] M. F. MR SNOEIJ et al. “A CMOS image sensor with a column-level multiple-ramp single-slope ADC”. In : *Solid-State Circuits* ... (2007), p. 506–508. ISSN : 01936530. DOI : [10.1109/ISSCC.2007.373516](https://doi.org/10.1109/ISSCC.2007.373516). URL : [http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=4242487](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=4242487) (cf. p. 19, 23).
- [Sno+07b] Martijn F. SNOEIJ et al. “Power and Area Efficient Column-Parallel ADC Architectures for CMOS Image Sensors”. In : *2007 IEEE Sensors* (2007), p. 523–526. DOI : [10.1109/ICSENS.2007.4388451](https://doi.org/10.1109/ICSENS.2007.4388451). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4388451> (cf. p. 19).
- [Sno07] MF SNOEIJ. “Multiple-ramp column-parallel ADC architectures for CMOS image sensors”. In : *Solid-State Circuits*, ... 42.12 (2007), p. 2968–2977. URL : [http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=4381464](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=4381464) (cf. p. 19, 23).
- [Spi+09] Arthur SPIVAK et al. “Wide-Dynamic-Range CMOS Image Sensors - Comparative Performance Analysis”. In : *IEEE Transactions on Electron Devices* 56.11 (2009), p. 2446–2461. ISSN : 0018-9383. DOI : [10.1109/TED.2009.2030599](https://doi.org/10.1109/TED.2009.2030599). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5272465> (cf. p. 23).
- [SS60] Harold S. SHAPIRO et Richard A. SILVERMAN. “Alias-Free Sampling of Random Noise”. In : *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), p. 225–248. ISSN : 0368-4245. DOI : [10.1137/0108013](https://doi.org/10.1137/0108013). URL : <http://epubs.siam.org/doi/abs/10.1137/0108013> (cf. p. 54).
- [ST06] MF F SNOEIJ et Albert THEUWISSEN. “A CMOS imager with column-level ADC using dynamic column FPN reduction”. In : *Solid-State Circuits* ... 41.12 (2006), p. 76–79. URL : [http://ieeexplore.ieee.org/xpls/abs{\\\_}all.jsp?arnumber=1696260](http://ieeexplore.ieee.org/xpls/abs{\_}all.jsp?arnumber=1696260) (cf. p. 19).

- [Sug+05] S. SUGAWA et al. “A 100dB dynamic range cmos image sensor using a lateral overflow integration capacitor”. In : *ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005*. T. 44. 10. IEEE, 2005, p. 352–354. ISBN : 0-7803-8904-2. DOI : [10.1109/ISSCC.2005.1494014](https://doi.org/10.1109/ISSCC.2005.1494014). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1494014> (cf. p. 24).
- [Sut89a] I. E. SUTHERLAND. “Micropipelines”. In : *Communications of the ACM* 32.6 (1989), p. 720–738. ISSN : 0001-0782. DOI : [10.1145/63526.63532](https://doi.org/10.1145/63526.63532). URL : <http://doi.acm.org/10.1145/63526.63532> (cf. p. 43).
- [Sut89b] Ivan E SUTHERLAND. “Micropipelines”. In : *Communications of the ACM* 32.6 (1989), p. 720–738. ISSN : 00010782. DOI : [10.1145/63526.63532](https://doi.org/10.1145/63526.63532) (cf. p. 110).
- [Tan+15] Fang TANG et al. “A Column-Parallel Inverter-Based Cyclic ADC for CMOS Image Sensor With Capacitance and Clock Scaling”. In : *IEEE Transactions on Electron Devices* (2015), p. 1–1. ISSN : 0018-9383. DOI : [10.1109/TED.2015.2450253](https://doi.org/10.1109/TED.2015.2450253). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7163317> (cf. p. 23).
- [The07] Albert THEUWISSEN. “CMOS image sensors: State-of-the-art and future perspectives”. In : *ESSCIRC 2007 - 33rd European Solid-State Circuits Conference*. IEEE, 2007, p. 21–27. ISBN : 978-1-4244-1125-2. DOI : [10.1109/ESSCIRC.2007.4430242](https://doi.org/10.1109/ESSCIRC.2007.4430242). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4430242> (cf. p. 13).
- [The08] Albert THEUWISSEN. “CMOS image sensors: State-of-the-art”. In : *Solid-State Electronics* 52.9 (2008), p. 1401–1406. ISSN : 00381101. DOI : [10.1016/j.sse.2008.04.012](https://doi.org/10.1016/j.sse.2008.04.012). URL : <http://linkinghub.elsevier.com/retrieve/pii/S0038110108001317> (cf. p. 18).
- [Tho90] Simon J THORPE. “Spike arrival times: A highly efficient coding scheme for neural networks”. In : *Parallel processing in neural systems and computers*. North-Holland Elsevier, 1990, p. 91–94. ISBN : 978-0444883902 (cf. p. 29).
- [Udd86] Jan Tijmen UDDING. “A formal model for defining and classifying delay-insensitive circuits and systems”. In : *Distributed Computing* 1.4 (1986), p. 197–204. ISSN : 0178-2770. DOI : [10.1007/BF01660032](https://doi.org/10.1007/BF01660032). URL : <http://link.springer.com/10.1007/BF01660032> (cf. p. 42).

- [Ung69] Stephen H. UNGER. *Asynchronous Sequential Switching Circuits*. T. John Wiley. Wiley-Interscience, 1969, 1969, p. 290 (cf. p. 35, 37).
- [Uns00] M. UNSER. “Sampling-50 years after Shannon”. In : *Proceedings of the IEEE* 88.4 (2000), p. 569–587. ISSN : 0018-9219. DOI : [10.1109/5.843002](https://doi.org/10.1109/5.843002). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=843002> (cf. p. 49).
- [VT01] Rufin VANRULLEN et Simon J THORPE. “Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex.” In : *Neural computation* 13.6 (2001), p. 1255–83. ISSN : 0899-7667. URL : <http://www.ncbi.nlm.nih.gov/pubmed/11387046> (cf. p. 30).
- [VT09] Christos VEZYRTZIS et Yannis TSIVIDIS. “Processing of signals using level-crossing sampling”. In : *2009 IEEE International Symposium on Circuits and Systems*. IEEE, 2009, p. 2293–2296. ISBN : 978-1-4244-3827-3. DOI : [10.1109/ISCAS.2009.5118257](https://doi.org/10.1109/ISCAS.2009.5118257). URL : <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5118257> (cf. p. 53).
- [Wan+04] Zhou WANG et al. “Image quality assessment: from error visibility to structural similarity.” In : *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 13.4 (2004), p. 600–12. ISSN : 1057-7149. URL : <http://www.ncbi.nlm.nih.gov/pubmed/15376593> (cf. p. 98).
- [XE03] XINQIAO LIU et A. EL GAMAL. “Synthesis of high dynamic range motion blur free image from multiple captures”. In : *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 50.4 (2003), p. 530–539. ISSN : 1057-7122. DOI : [10.1109/TCSI.2003.809815](https://doi.org/10.1109/TCSI.2003.809815). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1196451> (cf. p. 21).
- [XLY13] Jiangtao XU, Dongsheng LI et Suying YAO. “A Time Error Correction Method Applied to High-Precision AER Asynchronous CMOS Image Sensor”. In : *Journal of Signal Processing Systems* 75.1 (2013), p. 1–13. ISSN : 1939-8018. DOI : [10.1007/s11265-013-0800-0](https://doi.org/10.1007/s11265-013-0800-0). URL : <http://link.springer.com/10.1007/s11265-013-0800-0> (cf. p. 29).
- [Xue+14] Feifei XUE et al. “Design of a 10-bit 50MSPS pipeline ADC for CMOS image sensor”. In : *2014 9th IEEE Conference on Industrial Electronics and Applications*. IEEE, 2014, p. 891–895. ISBN : 978-1-4799-4315-9. DOI : [10.1109/ICIEA.2014.6931289](https://doi.org/10.1109/ICIEA.2014.6931289). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6931289> (cf. p. 18).

- [YFE99] D.X.D. YANG, B. FOWLER et A. EL GAMAL. “A Nyquist-rate pixel-level ADC for CMOS image sensors”. In : *IEEE Journal of Solid-State Circuits* 34.3 (1999), p. 348–356. ISSN : 00189200. DOI : [10.1109/4.748186](https://doi.org/10.1109/4.748186). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=748186> (cf. p. 19, 23).
- [YLD15] Minhao YANG, Shih-chii LIU et Tobi DELBRÜCK. “A Dynamic Vision Sensor With 1 % Temporal Contrast Sensitivity and In-Pixel Asynchronous Delta Modulator for Event Encoding”. In : *IEEE Journal of Solid-State Circuits* 50.9 (2015), p. 2149–2160. ISSN : 0018-9200. DOI : [10.1109/JSSC.2015.2425886](https://doi.org/10.1109/JSSC.2015.2425886). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7128412> (cf. p. 127).
- [YPEC04] O YADID-PECHT et R ETIENNE-CUMMINGS. *CMOS imagers: from phototransduction to image processing*. 2004. ISBN : 1402079621. URL : <http://www.springer.com/us/book/9781402079610> (cf. p. 16, 18).
- [YPGSD91] Orly YADID-PECHT, Ran GINOSAR et Yosi SHACHAM-DIAMAND. “A random access photodiode array for intelligent image capture”. In : *IEEE Transactions on Electron Devices* 38.8 (1991), p. 1772–1780. ISSN : 00189383. DOI : [10.1109/16.119013](https://doi.org/10.1109/16.119013). URL : <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=119013> (cf. p. 13).
- [ZB02] Wang ZHOU et a C BOVIK. “A universal image quality index”. In : *Signal Processing Letters, IEEE* 9.3 (2002), p. 81–84. ISSN : 1070-9908. DOI : [10.1109/97.995823](https://doi.org/10.1109/97.995823). URL : <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=995823> (cf. p. 98).
- [Zim11] Hakim ZIMOUCHE. “Capteur d’images CMOS à réponse insensible aux variations de température”. Thèse de doct. Université de Grenoble, 2011. URL : <https://tel.archives-ouvertes.fr/tel-00656381> (cf. p. 16).





## Résumé

---

Face aux défis actuels liés à la conception de capteurs d'images à forte résolution comme la limitation de la consommation électrique, l'augmentation du flux de données ainsi que le traitement de données associé, on propose, à travers cette thèse, un capteur d'image novateur asynchrone à échantillonnage non uniforme. Ce capteur d'images asynchrone est basé sur une matrice de pixels événementiels qui intègrent un échantillonnage non uniforme par traversée de niveaux. Contrairement aux imageurs conventionnels, où les pixels sont lus systématiquement lors de chaque trame, les pixels événementiels proposés sont consultés que lorsqu'ils contiennent une information pertinente. Cela induit un flux de données réduit et dépendant de l'image.

Pour compléter la chaîne de traitement des pixels, on présente également une architecture numérique de lecture dédiée conçue en utilisant de la logique asynchrone et destinée à contrôler et à gérer le flux de données des pixels événementiels. Ce circuit de lecture numérique permet de surmonter les difficultés classiques rencontrées lors de la gestion des demandes simultanées des pixels événementiels sans dégrader la résolution et le facteur de remplissage du capteur d'images. En outre, le circuit de lecture proposé permet de réduire considérablement les redondances spatiales dans une image ce qui diminue encore le flux de données.

Enfin, en combinant l'aspect échantillonnage par traversée de niveau et la technique de lecture proposée, on a pu remplacer la conversion analogique numérique classique de la chaîne de traitement des pixels par une conversion temps-numérique (*Time-to-Digital Conversion*). En d'autres termes, l'information du pixel est codée par le temps. Il en résulte une diminution accrue de la consommation électrique du système de vision, le convertisseur analogique-numérique étant un des composants les plus consommant du système de lecture des capteurs d'images conventionnels.

**Mots Clés** – *Capteur d'images CMOS, Échantillonnage Non Uniforme, Événementiel, Logique asynchrone, Réduction du flux de données, , Conversion Tension-Temps*

---

## Abstract

---

In order to overcome the challenges associated with the design of high resolution image sensors, we propose, through this thesis, an innovative asynchronous event-driven image sensor based on non-uniform sampling. The proposed image sensor aims the reduction of the data flow and its associated data processing by limiting the activity of our image sensor to the new captured information.

The proposed asynchronous image sensor is based on an event-driven pixels that incorporate a non-uniform sampling crossing levels. Unlike conventional imagers, where the pixels are read systematically at each frame, the proposed event-driven pixels are only read when they hold new and relevant information. This induces a reduced and scene dependent data flow.

In this thesis, we introduce a complete pixel reading sequence. Beside the event-driven pixel, the proposed reading system is designed using asynchronous logic and adapted to control and manage the flow of data from event pixels. This digital reading system overcomes the traditional difficulties encountered in the management of simultaneous requests for event pixels without degrading the resolution and fill factor of the image sensor. In addition, the proposed reading circuit significantly reduces the spatial redundancy in an image which further reduces the data flow.

Finally, by combining the aspect of level crossing sampling and the proposed reading technique, we replaced the conventional analog to digital conversion of the pixel processing chain by a time-to-digital Conversion (TDC). In other words, the pixel information is coded by time. This results in an increased reduction in power consumption of the vision system, the analog-digital converter being one of the most consuming reading system of conventional image sensors components.

**Keywords** – *Image Sensor, CMOS, Asynchronous, Non Uniform Sampling, Reduced Data Flow, Level Sampling Scheme, Time-to-Digital Conversion*

---