



**HAL**  
open science

# Formalization and Simulation of Bio-Inspired On-chip Dynamic Attractors for Low Power Computer Vision

Louise Sarrabezolles

► **To cite this version:**

Louise Sarrabezolles. Formalization and Simulation of Bio-Inspired On-chip Dynamic Attractors for Low Power Computer Vision. Artificial Intelligence [cs.AI]. Institut Polytechnique de Paris, 2020. English. NNT : 2020IPPAE001 . tel-02887189

**HAL Id: tel-02887189**

**<https://theses.hal.science/tel-02887189v1>**

Submitted on 2 Jul 2020

**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.



INSTITUT  
POLYTECHNIQUE  
DE PARIS



NNT : 2020IPPAE001

# Formalisation et simulation des Attracteurs Dynamiques sur composant bio-inspiré pour une vision artificielle à faible consommation énergétique

Thèse de doctorat de l'Institut Polytechnique de Paris  
préparée à l'École nationale supérieure des techniques avancées (ENSTA) et  
l'Institut Franco-Allemand de Recherche de Saint-Louis (ISL)

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP  
Paris)  
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Saint-Louis (Haut-Rhin), le 8 janvier 2020, par

**LOUISE SARRABEZOLLES**

Composition du Jury :

Alain Dieterlen Professeur, HDR, Université de Haute-Alsace (MIPS)	Président
Michel Paindavoine Professeur, HDR, Université de Bourgogne (LEAD)	Rapporteur
Eva Dokladalova Professeur associé, HDR, Université Paris-Est, ESIEE (A3IS)	Rapporteur
Titus Zaharia Professeur, HDR, Institut Polytechnique de Paris (ARTEMIS)	Examineur
Antoine Manzanera Professeur, HDR, Institut Polytechnique de Paris, ENSTA Paris (U2IS)	Directeur de thèse
Pierre Raymond Directeur de recherche, HDR, Institut France-Allemand de Saint-Louis (ELSI)	Co-directeur de thèse
Nicolas Hueber Chargé de recherche, Institut France-Allemand de Saint-Louis (ELSI)	Co-encadrant de thèse
Patrick Pirim Ingénieur, AnotherBrain	Invité

Thèse de doctorat



## Declaration of Authorship

I, Louise SARRABEZOLLES, declare that this thesis titled, “Formalization and Simulation of Bio-Inspired On-chip Dynamic Attractors for Low Power Computer Vision” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 13/02/2020



**“Del Spooner:** *So, Dr. Calvin, what exactly do you do around here?*

**Dr. Susan Calvin:** *My general fields are advanced robotics and psychiatry. Although, I specialize in hardware-to-wetware interfaces in an effort to advance U.S.R.'s robotic anthropomorphization program.*

**Del Spooner:** *So, what exactly do you do around here?*

**Dr. Susan Calvin:** *I make the robots seem more human.*

**Del Spooner:** *Now wasn't that easier to say?*

**Dr. Susan Calvin:** *Not really. No.”*

*I, Robot*, directed by Alex Proyas, 2004.



## Acknowledgements

Le doctorat est un peu comme une longue, trop longue, traversée du désert. On est toujours heureux d'y croiser une oasis, elle donne une motivation, un objectif et nous fait aller de l'avant. Certaines oasis sont grandes, d'autres petites, certaines sont riches, d'autres pauvres, enfin certaines ne sont en fait que des mirages. Aucune ne s'étend sur tout le parcours, certaines nous entraînent même dans de mauvaises directions. Et à chaque fois qu'elles s'arrêtent ou qu'elles disparaissent, le retour au désert est encore plus difficile. Mais je me dois et je vais m'efforcer ici de ne garder en mémoire que les bons côtés de l'oasis et de remercier tous ceux qui ont pu un jour en être une, quelle que soit sa forme, même les mirages.

Je remercie tout d'abord les entités sans qui cette thèse n'aurait pas eu lieu : **L'ISL** pour m'avoir financé pendant quatre années, m'avoir permis d'aller en conférence et en école d'été, m'avoir donné un cadre de travail agréable et une facilité à gérer les déplacements entre Paris et Saint-Louis ; **l'ENSTA** pour m'avoir accueilli un tiers du temps et m'avoir donné l'opportunité de donner des TDs.

Je remercie tous les acteurs de cette thèse qui en ont permis le commencement, la continuité et la finalisation :

**Patrick Pirim**, qui a donné à plusieurs reprises de son temps afin de me permettre de mieux comprendre le fonctionnement de son dispositif, le composant BIPS, et du concept bio-inspiré qui lui est associé ; **Christophe Visine** qui a poussé à l'étude du composant BIPS et par conséquent à l'ouverture de ce poste de doctorante ; mes encadrants : **Antoine Manzanera** pour sa disponibilité et sa réactivité, **Pierre Raymond** pour sa franchise et son efficacité, **Nicolas Hueber** pour sa simplicité et sa bonne humeur. Leurs trois profils très différents m'ont donné un équilibre, pas toujours stable mais formateur tout de même, entre la liberté de gestion de mon projet de recherche et le maintien d'une ligne directrice ponctuée d'avancées scientifiques et médiatiques.

Je remercie également tous les membres du jury : **Michel Paindavoine**, **Eva Dokladalova**, **Alain Dieterlen** et **Titus Zaharia**, qui ont accepté leurs rôles de rapporteurs et examinateurs malgré des délais serrés, des grèves fortement perturbantes et une vidéo-transmission compliquée. Vos retours sur mon manuscrit et nos échanges lors de ma soutenance m'ont permis de mieux prendre conscience de l'importance et de l'intérêt de mes travaux et de leur place dans le domaine de la vision artificielle embarquée.

Je remercie enfin **Adriana Tapus** et **Emmanuel Fullenwarth** qui m'ont aidé à surmonter rapidement les problématiques administratives liées au changement de date de la soutenance et au changement d'université de rattachement de l'ENSTA.

Je souhaite également remercier tous ceux qui m'ont aidé sans être directement impliqués dans la thèse :

**Sylvain** pour m'avoir donné le bon coup de pouce, **Adrian (and the LAL team)** for making me discover and love research, **Daniel (et l'équipe de CS Toulon)** pour m'avoir permis d'emboîter le pas vers la recherche en vision embarquée ; **mes collègues de l'ISL et de l'U2IS** (tous ceux qui ont eu le courage d'aller jusqu'au fond du couloir) : partenaires de jeux, de pauses cafés, et de ménages de printemps ;) ; **Adina și Roxana** pentru toate pauzele și discuțiile de pe culoar ; **Daniela** pelas longas conversas no escritório (espero que você não fique sozinha por muito tempo em seu palácio ;) ) ; **PH, Adrien, Clément, Pinard** pour toutes



ces soirées qui ont rendus les jours à Paris bien meilleurs ; **Antoine** pour avoir été le meilleur des collègues et l'élément motivateur de notre petite équipe Vision (même bien après avoir quitté le labo !) ; mes amis de toujours qui m'ont permis de sortir la tête du guidon de temps en temps et garder un peu de vie sociale : **Anne-Lise, Coco, Lulu, Pumpum, Titi, Nicolas, PA, Julien, le BG3D et pièces rapportées, Élise, Vincent, Riri, Fifi et Flo** ; **Olivier** pour m'avoir fait grandir et avoir supporté mes aller-retours et mes déprimés aussi longtemps que possible ; **Dom** pour avoir organisé toutes ces soirées pour me remonter le moral ; et enfin ma famille : **mes cousins** pour les retrouvailles dans les Alpes et dans les Pyrénées qui m'ont redonné un peu d'air, **Manou** pour ta présence et les discussions toujours passionnantes, **Mamy** pour m'avoir toujours gardé une place à Paris et pour tous tes conseils yoga (j'ai fini par les suivre :) ), **Marine** et **Alix** pour les coups de fils motivateurs depuis le Texas, **Pauline** pour les conseils de thésardes, **Constance** et **Charlie** pour les sorties running et ski et surtout **mes parents** sans qui je n'aurais jamais eu de pot à ma soutenance !

Ceux qui me connaissent bien savent à quel point le sport m'a permis de survivre à cette thèse et je souhaite remercier tous ceux qui ont été là pour me rappeler et me remotiver à sortir me défouler :

**les collègues sportifs du CSX, de l'ASA et des Dauphins de Saint-Louis** pour les séances de fractionnés, les exercices sur la dure montée de l'X, la découverte des trails, de l'escalade, et surtout pour les barbecues et les restaurants post-entraînements (c'est toujours le meilleur moyen de ramener les gens dans l'eau, sur la piste ou sur le mur ;) ) ; **Marion et Nathalie** pour votre constance à la gym et à tous mes pots de départs ^^ ; **Alex et Michel** pour toutes les sorties Géocaching, les montées d'arbres à 30m et rampements dans les tunnels boueux qui m'ont permis de retrouver confiance en moi et de me libérer de mes peurs.

Et pour terminer, je souhaite remercier ceux qui ont été envoyés par mon ange-gardien et qui principalement les derniers mois de thèse ont éclairé mes journées, m'ont permis de surmonter les difficultés extérieures, de tenir bon et de voir la fin de ce long tunnel :

**Lucas** pour toutes les pauses cafés, les tours de parc, les discussions et le travail essentiel de relecture de mes slides ; **Julie** pour me suivre dans tous mes délires sportifs, pour tenir les longues sessions nécessaires à mon retour au calme, pour m'écouter raconter mes soucis pendant des heures et pour éternellement garder le sourire et la bonne humeur ; **Marvin** für alle die Schwimmtrainings und Text-nachrichten, dass da war, wenn ich es brauchte, insbesondere in den letzten Wochen und für die Ermutigung, dass er, ohne es zu wissen, nach jeder der harten Schlägen der letzten Monate, mir gebracht hast ; **Caro** pour ce mot de soutien qui pendant deux ans, posé sur mon bureau, m'a rappelé que je batterai ces montagnes russes, pour être la meilleure des colloqs et avoir couru m'aider jusqu'à ma dernière nuit blanche ; **Régine et Juluan** pour m'avoir accueilli comme un membre de la famille dans un moment où la solitude n'était plus possible ; et **Philippe** pour s'être occupé de moi sur toute cette fin de thèse, pour m'avoir logé, pour m'avoir encadré et poussé à mettre de vraies deadlines, pour m'avoir fait des retours constructifs sur mon travail, pour m'avoir permise de retrouver un bon rythme de vie et pour m'avoir permise d'arriver au bout de la thèse.

# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Acknowledgements</b>	<b>v</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Abbreviations</b>	<b>xv</b>
<b>List of Symbols</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Embedded Computer Vision constraints . . . . .	2
1.2 Objective . . . . .	4
1.3 Contributions . . . . .	5
1.4 Thesis outline . . . . .	6
<b>2 The Bio-inspired Approach</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Approach motivations . . . . .	9
2.2.1 Adequacy between theoretical and engineer approaches . . .	9
2.2.2 Algorithms limitations . . . . .	10
2.3 Human vision . . . . .	11
2.3.1 Nerve cells . . . . .	12
2.3.2 The retina . . . . .	13
2.3.3 The Lateral Geniculate Nucleus (LGN) . . . . .	14
2.3.4 The visual cortex . . . . .	15
2.3.5 The brain systems . . . . .	16
2.3.6 Conclusion . . . . .	17
2.4 The BIPS on-chip . . . . .	17
2.4.1 Related bio-inspired developments . . . . .	17
2.4.2 Light perception . . . . .	18
2.4.3 Feature extraction . . . . .	18
2.4.4 the Bio-Inspired Perception Sensor . . . . .	18
<b>3 BIPS formalization</b>	<b>21</b>
3.1 Introduction . . . . .	21
3.2 Feature extraction . . . . .	23
3.2.1 Tonal features . . . . .	24
3.2.2 Dynamic features . . . . .	25
3.2.3 Structural features . . . . .	27
3.2.4 Formalization . . . . .	29
3.2.5 Conclusion . . . . .	30

3.3	Description and Localization: DA module	31
3.3.1	Feature Mode Detector (FMD)	32
3.3.2	Dynamic Attractor (DA)	35
3.3.3	DA process illustration	39
3.3.4	Conclusion	43
3.4	Multiple detection	44
3.4.1	Connections of DAs	44
3.4.2	Observations	45
3.4.3	Parameters	48
3.4.4	Conclusion	49
3.5	Discussion	49
<b>4</b>	<b>The BIPS simulator</b>	<b>51</b>
4.1	Feature	52
4.1.1	The Feature class	52
4.1.2	Simulator implemented features	53
4.2	Feature Mode Detector	55
4.2.1	The FeatModeDetector class	55
4.2.2	FMD pixel validation	57
4.2.3	Histogram construction	58
4.2.4	Update boundaries	59
4.3	Dynamic Attractor	60
4.3.1	The DynamicAttractor class	60
4.3.2	Feature value	61
4.3.3	Pixel validation	62
4.3.4	Histogram computation	63
4.3.5	Update	64
4.3.6	Iterative process	65
4.3.7	Time integration	68
4.4	Application and Visualization	69
4.4.1	Process	69
4.4.2	Images class	70
4.5	Conclusion	71
<b>5</b>	<b>Application to road lane detection</b>	<b>73</b>
5.1	Introduction	73
5.2	Lane detection	73
5.2.1	Application context	74
5.2.2	Methods and Models	75
5.2.3	Databases	77
5.2.4	Conclusion	77
5.3	BIPS adaptation	77
5.3.1	Features space	77
5.3.2	Architecture	83
5.3.3	Conclusion	84
5.4	Experimental results	84
5.4.1	The <i>caltech-lanes</i> dataset	85
5.4.2	Number of DAs and object size impact	87
5.4.3	Histogram feature thresholds impact	93
5.4.4	Conclusion	95
5.5	Discussion	96

<b>6</b>	<b>Extension proposals</b>	<b>97</b>
6.1	Introduction . . . . .	97
6.2	Time integration proposal . . . . .	98
6.2.1	Time integration overview . . . . .	98
6.2.2	Method proposal . . . . .	99
6.2.3	Illustration . . . . .	101
6.2.4	Conclusion . . . . .	111
6.3	Feature space reorientation . . . . .	111
6.3.1	Reorientation proposal . . . . .	112
6.3.2	Illustration . . . . .	115
6.3.3	Conclusion . . . . .	117
6.4	Discussion . . . . .	117
<b>7</b>	<b>Discussion and Perspectives</b>	<b>119</b>
7.1	Thesis contributions . . . . .	119
7.1.1	BIPS formalization . . . . .	119
7.1.2	BIPS simulator . . . . .	120
7.1.3	First parametric and performance analysis . . . . .	120
7.1.4	Extension proposals . . . . .	121
7.2	Dynamic Attractors benefits and limits . . . . .	121
7.2.1	Advantages . . . . .	121
7.2.2	Parametric limitations . . . . .	122
7.3	Conclusions and Perspectives . . . . .	123
7.3.1	Conclusions . . . . .	123
7.3.2	Short term perspectives . . . . .	124
7.3.3	Long term perspectives . . . . .	124
	<b>Bibliography</b>	<b>125</b>
<b>A</b>	<b>The BIPS proofs and alternative equations</b>	<b>133</b>
A.1	Feature Mode Detector . . . . .	133
A.1.1	Histogram computation . . . . .	133
A.1.2	Boundaries update . . . . .	133
A.1.3	Alternative equations . . . . .	134
A.2	Dynamic Attractor . . . . .	135
A.2.1	Convergence . . . . .	135



# List of Figures

1.1	B- <small>SAVED</small>	1
1.2	Highly constrained applications	2
1.3	BIPS	4
2.1	KITTI road results	11
2.2	Human brain	12
2.3	Neuron	12
2.4	Retina	13
2.5	Cones and Rods	14
2.6	Visual cortex	15
2.7	Ventral and dorsal streams	16
2.8	BIPcam and BIPeye	18
2.9	BIPS examples	20
3.1	BIPS representation by P. Pirm	21
3.2	BIPS diagram	22
3.3	Description and Localization part diagram	23
3.4	Color cylindrical representation	24
3.5	Tonal feature extraction	25
3.6	Dynamic feature extraction	26
3.7	Gradient coordinates representation	27
3.8	Structural feature extraction	28
3.9	Cyclic case: Boundaries selection	29
3.10	Dynamic Attractor representation by P. Pirm	31
3.11	Feature Mode Detector	32
3.12	Histogram computation and Boundaries update	34
3.13	Dynamic Attractor diagram	35
3.14	Iterative process diagram	37
3.15	Illustration: DA steps - 1	39
3.15	Illustration: DA steps - 2	40
3.15	Illustration: DA steps - 3	41
3.16	Illustration: the two modes - 1	42
3.16	Illustration: the two modes - 2	42
3.17	Illustration: the histogram threshold - 1	43
3.18	Illustration: the histogram threshold - 2	43
3.19	Colored chips image	46
3.20	Illustration: Multiple DAs	46
3.21	Illustration: Image transformation	47
3.22	Road image	47
3.23	Illustration: Cartesian space	48
4.1	Feature class	52
4.2	toYCbCr	53

4.3	toTonal	53
4.4	toDynamic	54
4.5	toStructural	54
4.6	toGrad	55
4.7	FeatModeDetector class	56
4.8	getValidFMD	57
4.9	addToMap	58
4.10	updateBoundaries	59
4.11	DA class	60
4.12	getKey	61
4.13	isValid	63
4.14	addToHisto	63
4.15	updateRotParam	64
4.16	DA iterative process	67
4.17	prediction	68
4.18	updateTime	68
4.19	process	69
4.20	MyWindow	70
5.1	Road conditions - 1	74
5.1	Road conditions - 2	75
5.2	Hillel et al. generic method	75
5.3	Gradient angle extraction on road images	78
5.4	Rotation adaptation: In the chip	80
5.5	Road image	81
5.5	Rotation adaptation: Successive compared to Iterative computation	81
5.6	Road image	82
5.6	Rotation adaptation: Selection compared to Estimation	82
5.7	Modified DA iterative process	83
5.8	<i>caltech-lanes</i> dataset	85
5.9	ROC curves with fixed number of DAs	88
5.10	Detected object per each DA	89
5.11	Number of sought lanes per objects size threshold values	90
5.12	Number of necessary DAs	91
5.13	ROC curves with automatic setting	92
5.14	Impact of $\tau_F$	94
5.15	Impact of $\tau_X$	95
6.1	BIPS diagram with Kalman	101
6.2	Illustration: No tracking	102
6.3	Illustration: Only enlargement - 1	103
6.4	Illustration: Only enlargement - 2	104
6.5	Illustration: Tracking with medians	105
6.6	Illustration: Tracking with Kalman - 1	107
6.7	Illustration: Tracking with Kalman - 2	108
6.8	Illustration: Tracking with Kalman - 3 iterations	110
6.9	Iterative process: extension	113
6.10	Illustration: Reorientation proposal - 1	115
6.10	Illustration: Reorientation proposal - 2	115
6.10	Illustration: Reorientation proposal - 3	116
6.10	Illustration: Reorientation proposal - 4	116

# List of Tables

2.1	Hardware platform . . . . .	10
3.1	Feature extraction computation time . . . . .	30
3.2	DA computation cost . . . . .	39
5.1	Performance comparison . . . . .	93
6.1	Number of reinitializations . . . . .	106
6.2	Mean number of iterations . . . . .	109
6.3	Percentage of convergence . . . . .	109
6.4	Number of reinitializations - 3 iterations . . . . .	109
7.1	Parameters of influence . . . . .	123





# List of Abbreviations

<b>ACC</b>	<b>A</b> daptive <b>C</b> ruise <b>C</b> ontrol
<b>AI</b>	<b>A</b> rtificial <b>I</b> ntelligence
<b>BIPS</b>	<b>B</b> io- <b>I</b> nspired <b>P</b> erception <b>S</b> ensor
<b>BVS</b>	<b>B</b> rain <b>V</b> ision <b>S</b> ystems
<b>DA</b>	<b>D</b> ynamic <b>A</b> tttractor
<b>DGA</b>	<b>D</b> irection <b>G</b> énérale de l' <b>A</b> rmement
<b>ENSTA</b>	<b>É</b> cole <b>N</b> ationale <b>S</b> upérieure de <b>T</b> echniques <b>A</b> vancées
<b>FMD</b>	<b>F</b> eature <b>M</b> ode <b>D</b> etector
<b>FN</b>	<b>F</b> alse <b>N</b> egative
<b>FP</b>	<b>F</b> alse <b>P</b> ositive
<b>FPR</b>	<b>F</b> alse <b>P</b> ositive <b>R</b> ate
<b>ISL</b>	<b>F</b> rench- <b>G</b> erman <b>R</b> esearch <b>I</b> nstitute of <b>S</b> aint- <b>L</b> ouis
<b>LDI</b>	<b>L</b> ane <b>D</b> eparture <b>I</b> dentification
<b>LDW</b>	<b>L</b> ane <b>D</b> eparture <b>W</b> arning
<b>LKA</b>	<b>L</b> ane <b>K</b> eeping <b>A</b> ssist
<b>LGN</b>	<b>L</b> ateral <b>G</b> eniculate <b>N</b> ucleus
<b>RT</b>	<b>R</b> eal <b>T</b> ime
<b>SWaP</b>	<b>S</b> ize, <b>W</b> eight and <b>P</b> ower
<b>TN</b>	<b>T</b> rue <b>N</b> egative
<b>TP</b>	<b>T</b> rue <b>P</b> ositive
<b>TPR</b>	<b>T</b> rue <b>P</b> ositive <b>R</b> ate



# List of Symbols

## Formalization

$W$	image width
$H$	image height
$\mathbb{I} \times \mathbb{J}$	image pixel space
$p, (i, j), (y, x)$	pixel
$t$	time
$k$	DA iteration
$M, M_q$	number of DA features
$n_{DA}$	number of DAs
$\tau_{sth}$	threshold
$val_{sth}$	validation
$\mathbb{P}$	feature application domain
$F, F_m$	feature extraction
$f, f_m$	feature value
$\Omega, \Omega_m$	feature values set
$H, H_m$	conditioned feature histogram
$(q_{max}, f_{max})$	histogram mode
$A, A_m$	lower feature boundary
$B, B_m$	upper feature boundary
$N$	number of valid pixels
$P$	framing
$\mathcal{O}$	valid pixels set

## Features

$L$	luminance
$Sa$	saturation
$Hu$	hue
$\hat{L}$	background estimation
$D$	temporal variation
$V$	optical flow velocity
$\theta$	optical flow direction
$W$	derivative 2D Gaussian steerable filter
$An$	gradient angle
$Mo$	gradient module

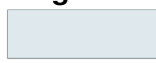
## Rotation

$\alpha$	spatial axis rotation angle
$\Delta_\alpha$	geometrical variation of the angle
$\langle \cdot \rangle$	scalar product
$\mathbf{u}, \mathbf{u}_m$	unit vector axis
$M_c$	number of rotated features
$\hat{\mu}, \hat{\mu}_m$	histogram median

$\mu, \mu_m$	histogram mean
$\Sigma$	covariance matrix
$e, e_m$	eigen vector

**Kalman filter**

F	state-transition model
P	error covariance
Q	process noise covariance
H	observation model
S	innovation covariance
R	observation noise covariance
K	gain

**Diagrams**

system or function block



function block



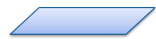
test block



input



output



internal variables used as input or output of function blocks

*Pour mes grand-parents*



## Chapter 1

# Introduction

The autonomous surveillance agent concept B-SAVED (Hueber et al., 2015), which was at the origin of the thesis research project, intends to help the soldiers in difficult surveillance tasks. In such military context, the area covered are often on sensitive sites and large area. Any soldier present on the site is thus exposed to an elevated level of danger. Moreover, the communication of any device is strongly limited to avoid all risk of message interception. The autonomous surveillance agent would thus have to be discrete in its form and its information communication.

As in several embedded vision applications (e.g. drones, autonomous cars, unmanned robots), the system should be minimized in Size, Weight and Power consumption (SWaP), while performing real-time detection, recognition and identification of events of interest (e.g. locations, actions, behaviors, situations of danger). This can involve different types of objects (humans, animals, cars, drones, etc.) and different sceneries (public places, mountains, rainy days, nights, etc.).



FIGURE 1.1 – The autonomous surveillance agent concept B-SAVED 1<sup>st</sup> generation developed by (Hueber et al., 2015) is at the origin of the thesis research project.

Building efficient, versatile and evolving algorithms under embedded constraints is one of today's challenges of Computer Vision. In such context, the bio-inspired and close to hardware approaches have been chosen as a guide for new and improved computer vision processes. The Bio-inspired Perception Sensor (BIPS) component and its Dynamic Attractor (DA) processing modules more specifically interested us and are at the center of this thesis work.



In this introductory Chapter, we present the scope and the objective of this study in respectively Section 1.1 and Section 1.2, before resuming our contributions in Section 1.3 and presenting the thesis outline in Section 1.4.

## 1.1 Embedded Computer Vision constraints

The significant progresses in computer vision at the beginning of this decade conducted to a growing interest in the artificial intelligence (AI) field. The proliferation of public researches on the subject and quick progression in the methods and results obtained led to the integration of the field as one of the eleven axes in the late innovation for Defense orientation document of the French government (*Document d'orientation de l'innovation de Défense (DOID) 2019*). The use of AI for the Defense, however, presents different specifications and constraints, which are far from being solved. The ministry of the Armies, thus, points out several research challenges, among which: (1) the robustness, (2) the simplification, (3) the versatility, (4) the embedded implementation and (5) the performance. Those challenges are also present in several computer vision applications, like the three following examples: surveillance, flying drone and autonomous driving (Fig 1.2).



(A)



(B)



(C)

FIGURE 1.2 – Examples of embedded computer vision applications implying multiple constraints on the mechanic, hardware and software developments as well as on the high-level process they must be able to perform. (A) Surveillance cameras (Hustvedt, 2008); (B) Drone Phantom (Sorenson, 2018); (C) Waymo autonomous car (Dilu, 2017)

## **Surveillance and Intelligence**

Surveillance is mostly performed online by human operators on video streams and sometimes on recorded video data (e.g. for forensic uses). Such conditions of analysis require high concentrations from the operators, which have to detect and react quickly to potential threats after hours, even days of observation.

The most recent generation of surveillance systems (3GSSs) integrates computer vision algorithms in order to reduce the data redundancy overload on the operators and the storage devices. The survey of (Kim et al., 2010) and the thesis introduction of (Ghorayeb, 2007) present the objectives of this new generation. The new systems tend on using more and more multimodal sensors to obtain higher level of information for threat detection and forensic analysis. Thus, the computer vision systems have to manage a large amount of information, to correlate extracted data from different sensors, to transmit the alert in real-time for online surveillance or to give forensic analysis results in good time. Moreover, the transmitted data have to be reduced to only high level information to avoid overloading the network bandwidth. On edge computing is then necessary. Meanwhile, the sensors are expected to be minimal in size, weight and power consumption for its discretion, portability and long autonomy.

## **Unmanned Aerial Vehicles**

Unmanned Aerial Vehicles (UAV) are mostly used in observation tasks during flight, having to face changeable environment and unforeseeable obstacles. Thus, their crash risk is growing up. Any error coming from the inertial measure unit (IMU), the control or the image analysis can lead to an unpredictable and uncontrolled behaviours, which eventually end in crash.

The survey (Kanellakis and Nikolakopoulos, 2017) underlines the need of real-time, further, highly sophisticated and robust control schemes in close loop with the perception information. The perception is, in fact, performed by numerous sensors (e.g. single, stereo, IR or RGB-D cameras, IMU sensors, GPS), and implies a very large amount of data to merge and to process, while the dynamics of the UAV requires real-time action. Moreover, the drone mechanics is size and weight limited. Thus, highly limiting the energy and hardware modules that can be embedded.

## **Autonomous driving**

In autonomous driving and driving assistance, the implemented systems are confronted to multiple situations (e.g. lightning, pedestrian, sudden crossing of animals, unpredictable driver behaviours), multiple scene contexts (e.g. highway, country road, rain, snow, fuzz). The high-level information extracted must then be relevant for multiple tasks, thus highly increasing the computational cost and complexity of the computer vision algorithms.

As we can see through the different surveys on the subject -e.g. (Horgan et al., 2015), (Ranft and Stiller, 2016)-, the autonomous driving task is confronted to similar issues as the UAVs: data fusion, high-level information extraction, robustness, real-time and low power consumption, but with a more important need in the multi-tasking capacity and long autonomy. The implemented algorithms must also be able to evolve along the scenery changes, which can be abrupt like in raining cases or crowded places.

## 1.2 Objective

### Close to hardware development

The Computer vision community proposes today elaborate visual processing algorithms that reach an interesting level of efficiency and robustness. However, as it is pointed out in the surveys cited above, the majority of those algorithms can not be integrated directly in the systems due to the limitations imposed by their architecture and their processing power.

The research project that encompasses this thesis aims to conceive, to validate and to optimize a hardware-software design improving the detection and recognition processes for embedded computer vision under highly constrained conditions. As shown by the three application examples, such design development will try to reach the five following points:

- Size, Weight and Power (SWaP) optimization
- Real-time (RT)
- Efficiency and Robustness
- Versatility
- Evolutivity

In this thesis project, we choose to follow a close to hardware approach for our design. As explained in (Ehsan and McDonald-Maier, 2015), in the context of UAV's computer vision developments, the pure algorithmic approach is confronted to many implementation difficulties and almost never reach the hardware implementation level. The choice of a close to hardware approach thus favors the RT and SWaP mitigation of the developed design.

### A peculiar processor

In the overview of existing computer vision components, the processor BIPS (Bio-inspired Perception Sensor) (Pirim, 2015) caught our intention with its biologically inspired aspect, its embedded properties and its promising computer vision results.

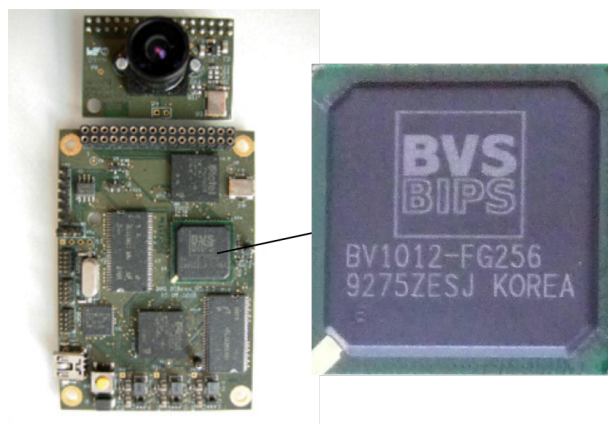


FIGURE 1.3 – BIPS processor developed by BVS-Tech (Pirim, 2015).

The processor results of a peculiar development process, which was purely hardware. Thus, no formalization or software simulator of the method behind it were available. The objective of this thesis became then to highlight the community on the functioning of the BIPS processor, to validate its properties and to evaluate its performance on academic conditions.

## 1.3 Contributions

This thesis contributes to the formalization and simulation of the original bio-inspired on-chip concept: the BIPS (Bio-inspired Perception Sensor), which presents interesting results on different industrial vision applications (e.g. traffic analysis, driving assistance, object tracking). The BIPS is a small and low-power component, and proposes interesting and uncommon bio-inspired ways to perform multiple perception tasks. The thesis contributes also in the experimental analysis of the formalized BIPS-based method, which helps in the understanding of its behavior and its parameters setting. Finally, the thesis proposes two extensions of the implemented method to improve its efficiency: one by temporal prediction and the other by adaptation of the detection process to the orientation of the input.

This work led to several publications, oral presentations and poster presentations in French and international conferences and workshops.

## Publications

Sarrabezolles, Louise, Antoine Manzanera, and Nicolas Hueber (2019). “Bio-Inspired Perception Sensor (BIPS) Concept Analysis for Embedded Applications”. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by Ruben Vera-Rodriguez, Julian Fierrez, and Aythami Morales. Lecture Notes in Computer Science. Springer International Publishing, pp. 428–435. ISBN: 978-3-030-13469-3.

Sarrabezolles, Louise et al. (2017). “Dual field combination for unmanned video surveillance”. In: vol. 10223. Anaheim, California, USA: International Society for Optics and Photonics, 102230A. DOI: [10 . 1117 / 12 . 2262696](https://doi.org/10.1117/12.2262696). (Visited on 12/12/2017).

## Oral Presentation

Sarrabezolles, Louise (2017a). *Bio-inspired and multi-modal smart sensor for real-time detection and recognition of visual events*. ISL, French-German Research Institute of Saint-Louis, Saint-Louis, France.

— (2017b). *Dual field combination for unmanned video surveillance*. Anaheim, California, USA.

— (2018a). *Bio-Inspired Perception Sensor (BIPS) concept analysis and application to real-time detection*. ISL, French-German Research Institute of Saint-Louis, Saint-Louis, France.

— (2018b). *Bio-inspired Perception Sensor (BIPS) Concept Analysis for Embedded Applications*. Madrid, Spain.

## Poster Presentation

- Sarrabezolles, Louise (2016). *Bio-Inspired and Multi-Modal Smart Sensor for Real-Time Detection and Recognition of Visual Events*. ISL, Saint-Louis, France.
- (2017). *Bio-inspired object representation method for real-time detection and recognition of visual events*. Lincoln, UK.
  - (2018a). *Bio-inspired perception sensor (BIPS) concept analysis and application to real-time object detection*. Bordeaux, France.
  - (2018b). *Bio-inspired Perception Sensor (BIPS) Concept Analysis for Embedded Applications*. Madrid, Spain.
  - (2019a). *An extended bio-inspired perception sensor (BIPS) concept for embedded tracking applications*. ISL, French-German Research Institute of Saint-Louis, Saint-Louis, France.
  - (2019b). *Bio-inspired perception sensor (BIPS) extensions for real-time detection and tracking*. Lille, France.

## 1.4 Thesis outline

The thesis manuscript is composed of 7 chapters following linearly the steps of our research project, at the exception of the simulator presented in Chapter 4, which has been developed and improved along the whole study.

**Chapter 2** presents the motivations of the BIPS (Bio-inspired Perception Sensor) component study. It firstly highlights on today's embedded computer vision bottleneck, which led us to follow close to hardware and bio-inspired approaches. Then, this Chapter gives an overview on the principal parts of the biological visual pathway, which have been a source of inspiration in many computer vision developments and are used by the BIPS component. This overview tries to highlight on the possible use of such biological mechanisms, but also on the limitation of their reproduction in software and hardware implementations. Finally, this Chapter introduces the BIPS component, its attractive properties and its lack of formalization and academic analysis, which this thesis tries to fill in.

**Chapter 3** presents the mathematical formalization of the BIPS component and its processing modules. A first part performs local feature extractions of tonal, structural and dynamic types, but can be formalized as generic perceptive features. The second part performs detection of object of interests, using the particular modules called Dynamic Attractors (DAs). The formalization brings forward several parameters whose setting is not automatized. Experimental studies of their influence is then necessary.

**Chapter 4** presents the simulator, which I developed during this thesis work. It permits to illustrate the BIPS module behaviors, to easily adapt its parameters to the different set of experiments and finally it allows to modify and extend the method for our improvement proposals. The simulator has been developed along the whole thesis work and thus integrates the different steps of formalization and extensions of the method.

**Chapter 5** presents the experimental study of the BIPS in the context of the road lane detection application. This first academical experiment permits to evaluate its

performance and the influence of several of its internal parameters. The chosen application allows to reduce the parameters of influence to only four. Using the *caltech-lanes* dataset, we have been able to propose an automatic setting of two of those parameters, which led us to results at the level of the reference (Aly, 2008).

Then **Chapter 6** proposes two extensions of the BIPS in order to improve its detection performances and widen its field of applications. The first one integrates time prediction with a Kalman filter to obtain better framings of the objects in a reduced computational time. The second one automatically reorients the input features space to obtain a better framing.

Finally, **Chapter 7** summarizes the contributions of this thesis, then discusses the different advantages and drawbacks of the studied method, and presents the expected future works.

At the end, the **Appendix A** gives mathematical proofs of the BIPS expected behaviour expressed in Chapter 3.



## Chapter 2

# The Bio-inspired Approach

### 2.1 Introduction

The Bio-Inspired Perception Sensor (BIPS), on which this thesis focuses, results from a peculiar development process. It has been created directly from hardware blocks implementation trying to follow several bio-inspired principles. The lack of mathematical formalization of its processes makes it difficult to understand the concept and mechanisms behind it. But its embedded characteristics and industrial results make it an interesting and potential solution for the embedded computer vision systems.

This Chapter introduces the hardware/hybrid and bio-inspired approach that led us to the study of this peculiar component. Section 2.2 presents the hardware and algorithmic limitations that oriented part of the bio-inspired embedded vision community toward close-to-hardware developments rather than pure algorithmic developments. Then Section 2.3 gives an overview of today's knowledge on the human visual systems, which is at the origin of many computer vision methods and permits to better understand the implementation proposed by the BIPS. Finally, Section 2.4 introduces the component, its history and its industrial results.

### 2.2 Approach motivations

The increase in precision and quality of the sensors as well as the acceleration of the computation capacities enable the development of new detection and recognition algorithms with remarkable performances and thereby open new horizon for artificial intelligence. Unfortunately, in the current state, the most promising computer vision algorithms are still using too much computational resources and power to achieve the RT and SWaP constraints of embedded and autonomous visual systems.

To overcome this issue, the bio-inspired computer vision community follows a strategy which leans on the hardware limitation knowledge to build adapted bio-inspired methods. In this Section, we explain the deep gap between the high-level computer vision concepts and their specific hardware integration, which induces us in the choice of bio-inspired and close-to-hardware approaches.

#### 2.2.1 Adequacy between theoretical and engineer approaches

During the last fifty years, the computer vision community relied on Moore's law for a rapid computational performance improvement of their implemented algorithms. However, the size and power consumption reduction have now reached its limitation and new approaches should be followed (Markov, 2014).



Some approaches will concentrate on new hardware development as presented in (Schuman et al., 2017). However, associated software support is not ensured and results for our constrained applications may never come. Another set of approaches will focus on new and improved computer vision algorithms compatible with the current hardware platforms.

A current state of hardware platforms can be found in (Shi et al., 2017) and (Velez and Otaegui, 2016). They describe the advantages and drawbacks of the *Central Processing Unit* (CPU), the *Graphics Processing Unit* (GPU), the *Field-Programmable Gate Array* (FPGA), the *Application-Specific Integrated Circuit* (ASIC) and the *Digital Signal Processor* (DSP) for embedded driving assistance applications. We summarized it in Table 2.1.

Hardware platform	Power consumption	Process level	Flexibility
CPU	high	high	modifiable
GPU	very high	high	modifiable
FPGA	very low	mid	modifiable
ASIC	low	mid	specific
DSP	low	low	specific

TABLE 2.1 – Hardware platform principal advantages and drawbacks.

The description given in the surveys highlights the difficulty of developing high level processing for limited computational resources. Low power components are often not suited for intensive computing as required by high-level processes and versatile algorithms. Moreover good knowledge on their internal functioning is necessary for an optimized implementation. This digs the gap between mathematical concepts and low level hardware developments with the solution providers having to adapt the most promising emerging concepts at the hardware level. As it is explained in (Ehsan and McDonald-Maier, 2015), an hardware/hybrid solution approach -meaning hardware-software co-design- has therefore better potential than pure software approach to fit in the limitations of small UAVs, and by extension of any highly constrained embedded computer vision applications.

## 2.2.2 Algorithms limitations

The current most efficient algorithms for computer vision tasks are, according to the academic datasets and challenges -e.g. (*KITTI*), (*PASCAL VOC*)-, deep learning methods. These methods appeared in the last decades as a universal solution in the domain. However, as G. Marcus points out in his critical appraisal (Marcus, 2018), one must not be fooled. The method excels at solving close-end classification problems, but is soon limited with generalizing challenges dealing with limited amount of training data, with novelty, or that can not be expressed as a classification problem. Moreover, most of the deep learning algorithms are tested on multi-GPU environments and yet, few of them are close to real-time. The four best algorithms on road detection according to KITTI challenge (*KITTI*) are all running in between 5 and 10 frame/s, only the fifth is able to perform in real-time (Fig. 2.1). The required energy remains the main drawback for embedded application.



	Method	Setting	Code	MaxF	AP	PRE	REC	FPR	FNR	Runtime	Environment
1	<a href="#">PLARD</a>			97.05 %	93.53 %	97.18 %	96.92 %	1.28 %	3.08 %	0.16 s	GPU @ 2.5 Ghz (Python)
Z. Chen, J. Zhang and D. Tao: <i>Progressive LiDAR adaptation for road detection</i> . IEEE/CAA Journal of Automatica Sinica 2019.											
2	<a href="#">DH-OCR</a>			96.84 %	91.10 %	97.08 %	96.60 %	1.32 %	3.40 %	0.2 s	GPU @ 2.5 Ghz (C/C++)
3	<a href="#">ZongNet</a>			96.70 %	90.12 %	96.00 %	97.41 %	1.85 %	2.59 %	0.1 s	1 core @ 2.5 Ghz (C/C++)
4	<a href="#">RockyNet</a>			96.65 %	90.09 %	95.97 %	97.35 %	1.86 %	2.65 %	0.1 s	1 core @ 2.5 Ghz (C/C++)
5	<a href="#">NF2CNN</a>			96.09 %	88.40 %	94.11 %	98.16 %	2.80 %	1.84 %	.006 s	GPU @ 3.5 Ghz (Python)

FIGURE 2.1 – Five best algorithms on road detection according to KITTI challenge (*KITTI*). They are all deep learning based methods. They run on highly consuming platforms, but still only one is able to perform in more than 10 frames/s

There are several attempts to integrate simplified deep networks on low power system-on-chip, but these are at their infancy and present much loss in the method efficiency, evolving and real-time capacity (Ota et al., 2017). Thus, one must look at different kinds of solutions. Among them, the neuromimetic solution, drawn on cognitive and psychological studies, has been of constant interest in the computer vision community and still today we have much to learn from the brain and how it processes and extracts information from the noisy and challenging world of our environment (Cristóbal, Perrinet, and Keil, 2015).

Estimation of the brain energy consumption, when the body is at rest is of about 20W (*Power of a Human Brain - The Physics Factbook*). This energy is sufficient for all cognitive tasks among which the visual attention tasks. However, human made system still cannot perform such tasks even with the most efficient hardware platforms. Even if the comparison between hardware system and biological system can not be done that easily, this energy argument encourages to a better understanding on how visual mechanisms can be so energy frugal and how we could reproduce them on hardware platforms.

## 2.3 Human vision

Since the works of Hermann von Helmholtz at the end of 19th century showing that the eyes only couldn't make the vision possible (Cahan, 1993), psychophysicists, neurophysicists and physiologists have continually improved the knowledge on the primate vision. This makes it today's most understood biological visual system and then the most used for bio-inspired computer vision.

In this Section, we present a short overview of the current knowledge on Human Vision to have a better understanding on the existing bio-inspired solutions and their possible use in embedded context. This overview is concentrated on the biological functioning of the visual system. The bio-inspiration that emerged from it will be discussed in Section 2.4. The following description is based on several books of Biology, among which: the *Introduction to the VISUAL SYSTEM* of (Tovée, 2008), the *Eye, Brain and Vision* book of Nobel prized David Hubel (Hubel, 1995), the online book of Kolb et al. (*Webvision – The Organization of the Retina and Visual System*) and the *Neuroscience* book of (Purves et al., 2018).

Schematically, the human visual system can be decomposed into three processing parts, which form the visual pathway as represented in Figure 2.2: the retina at the bottom of the eyes; the lateral geniculate nucleus (LGN) placed in the thalamus; and the visual cortex. Through electrical and chemical actions of nerve cells, the retina translates the incoming light into neural signals, which are processed a first time in the LGN and then sent into the visual cortex for higher level processing.

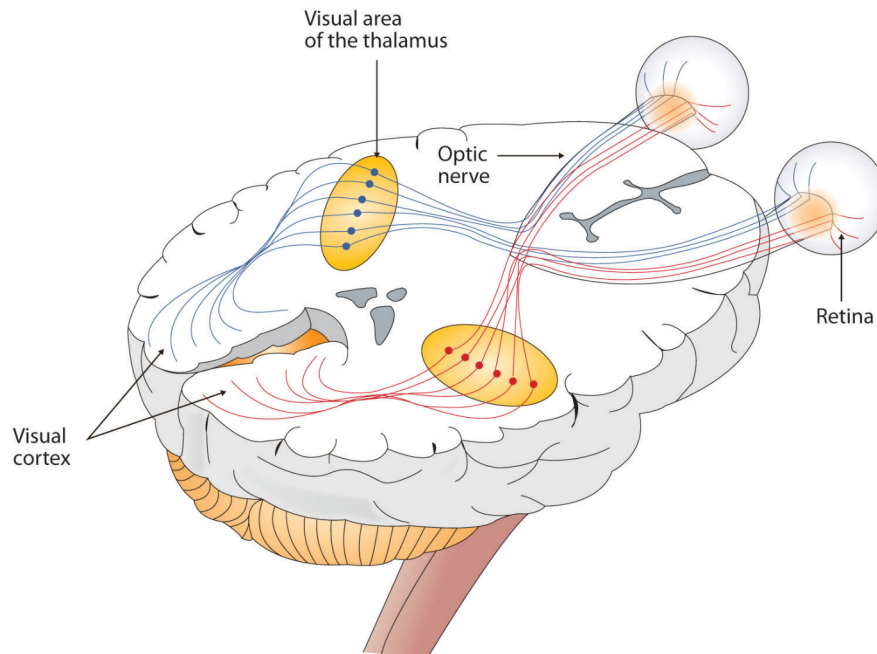


FIGURE 2.2 – Schema representing the principal parts of the human visual pathway: the retina, the LGN (visual area of the thalamus) and the visual cortex (*Visual pathway*).

### 2.3.1 Nerve cells

The nerve cells, also called neurons, are the main elements of the brain. They enable the generation of action potential and the transmission of nerve impulses (Gautam, 2017). They are composed of a body called the soma; dendrites and axon attached to the soma; and myelin sheath covering the axon, secreted by Schwann cells and separated by nodes of Ranvier as shown in Figure 2.3.

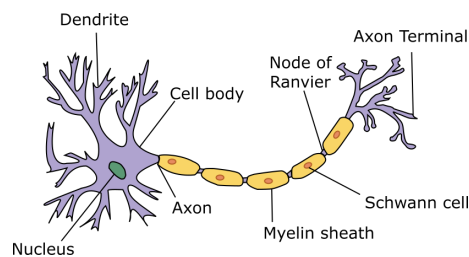


FIGURE 2.3 – Simple schema representing the neuron (Dhp1080, 2019)

There is between one hundred and one thousand types of nerve cells in the human brain. It is difficult for biologist and neuroscientific to put straight limits between the different types of cells: their constitutions can differ a lot and their functions in the brain can be totally different as it can be observed in the three parts of the visual pathway.

### 2.3.2 The retina

The eyes have the function to capture and focus light on to the retina. They manage the pupil aperture and the lens size, while maintaining good state and correct orientation of the whole. Moreover, they are controlled by six extra-ocular muscles, which orient the eyeball and trigger microsaccades whose function might also be for the perception of motion and spatial information (Martinez-Conde, Macknik, and Hubel, 2000) (Yablonski et al., 2017).

The retina takes place at the bottom of the eye and corresponds to the first neuronal layers, it is even often considered as a part of the brain (*Webvision – The Organization of the Retina and Visual System*). The retina is composed of three layers of nerve cells. The outer layer composed of the cones and rods photoreceptors; the inner layer composed of amacrine, bipolar and horizontal cells; and the ganglion cells layer as shown in Figure 2.4.

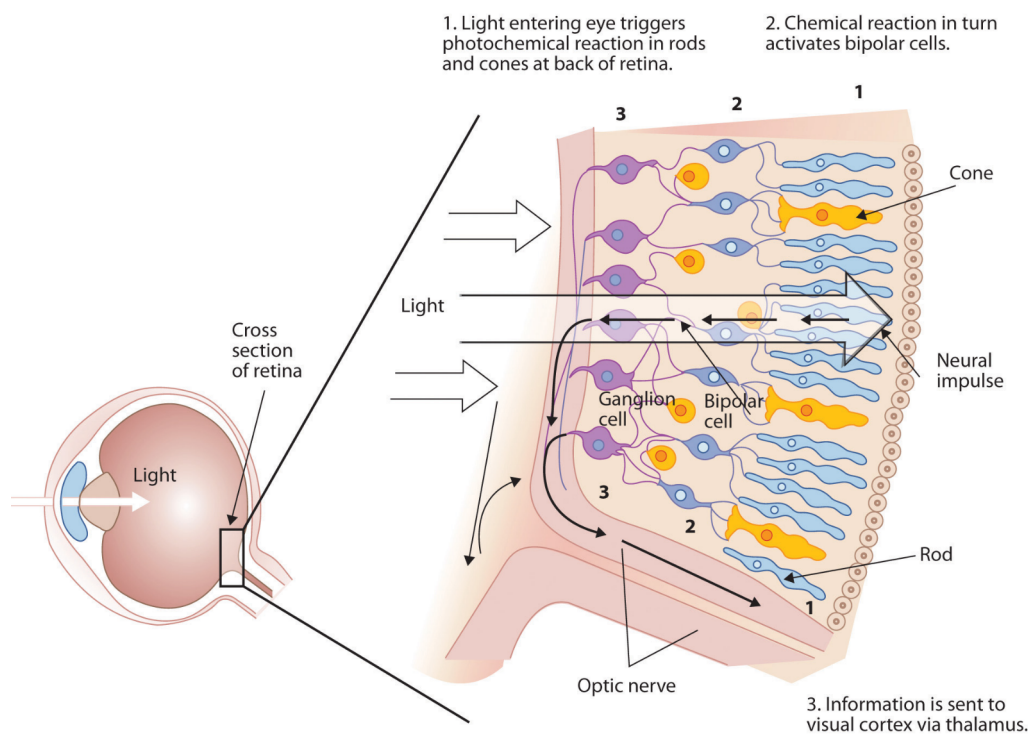


FIGURE 2.4 – Diagram representation of the retina and its three layers of cells (*The Retina with Its Specialized Cells*).

The photoreceptors are the first cells of the process: they receive the light and transform it into a first neural signal. The rods are sensitive to the brightness, whereas the cones are sensitive to the colors. It has been observed that there are three types of cones in the primate retinas: one reacts more to the red, one to the green and one to the blue. Their distribution on the retina is unequal: most of the cones are placed in the fovea, whereas rods are more in the close periphery as shown in Figure 2.5.

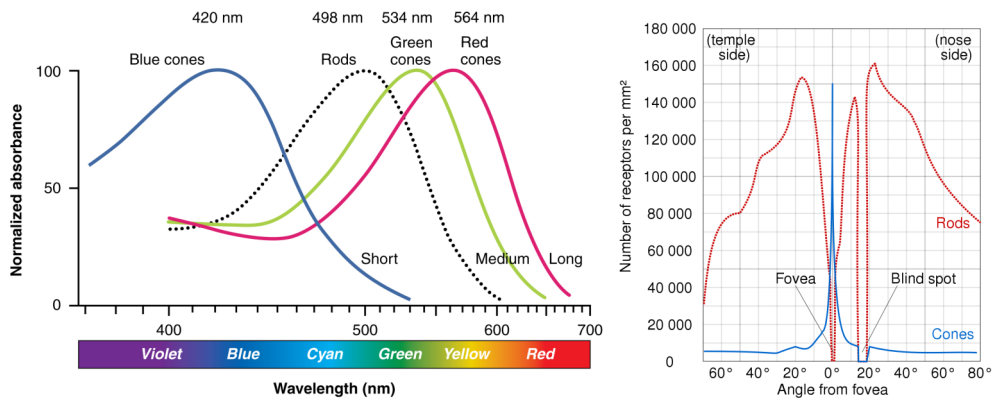


FIGURE 2.5 – Left: Wavelength sensitivity of the cones and rods cells (*English*). Right: Distribution of the cones and rods cells on the retina (*Color Sensitivity*).

The other cells perform the first combinations and higher level extraction of the image information. Amacrine cells help inhibiting other neurons; horizontal cells help regulate by region the photoreceptor inputs; and bipolar cells help in transmitting the signals to the ganglion cells. Finally, the ganglion cells are at 99% composed of M-cells and P-cells. The P-cells are selective for wavelength and high spatial frequencies and have tonic responses -slow and possibly graded-. The M-cells sensitive to low spatial frequencies and have phasic responses -fast and discontinuous-. Those ganglion cells axons are directly linked to the visual part of the thalamus, called the Lateral Geniculate Nucleus (or Body), and thus transmit the early processed image to the brain. It is important to notice that the retinal cells activity is performed asynchronously.

### 2.3.3 The Lateral Geniculate Nucleus (LGN)

The Lateral Geniculate Nucleus (LGN) can be found on each side of the brain inside the thalamus. It consists of six layers of cells: layers 2, 3 and 5 receive signals from the same side as the LGN and layers 1, 4 and 6 from the opposite side. The two first layers, called magnocellular layers, receive input from the M-cells; the four others, called parvocellular layers, receive input from the P-cells. Between those layers, there are koniocellular cells. It has been observed that the topographic disposition of the retinal ganglion cells were preserved within the LGN, letting suppose that the retina map was globally maintained.

The LGN receives 90% of the signals projected from the retina, the majority of inputs (90%) is coming from other parts of the brain, including the first layers of the visual cortex V1 (30%) It has been suggested that the LGN did not only play modulation and filter roles in the visual pathway, but that it had a broader role in cognition and perception (Saalman and Kastner, 2011).

### 2.3.4 The visual cortex

Most of the knowledge on the human visual cortex derives from experiments on monkeys, cats or rats and from psychophysical studies. They help constructing a schematic organisation of the visual cortex, decomposed into several areas as shown in Figure 2.6. Some areas have been identified as working mostly on motion, form or color information.

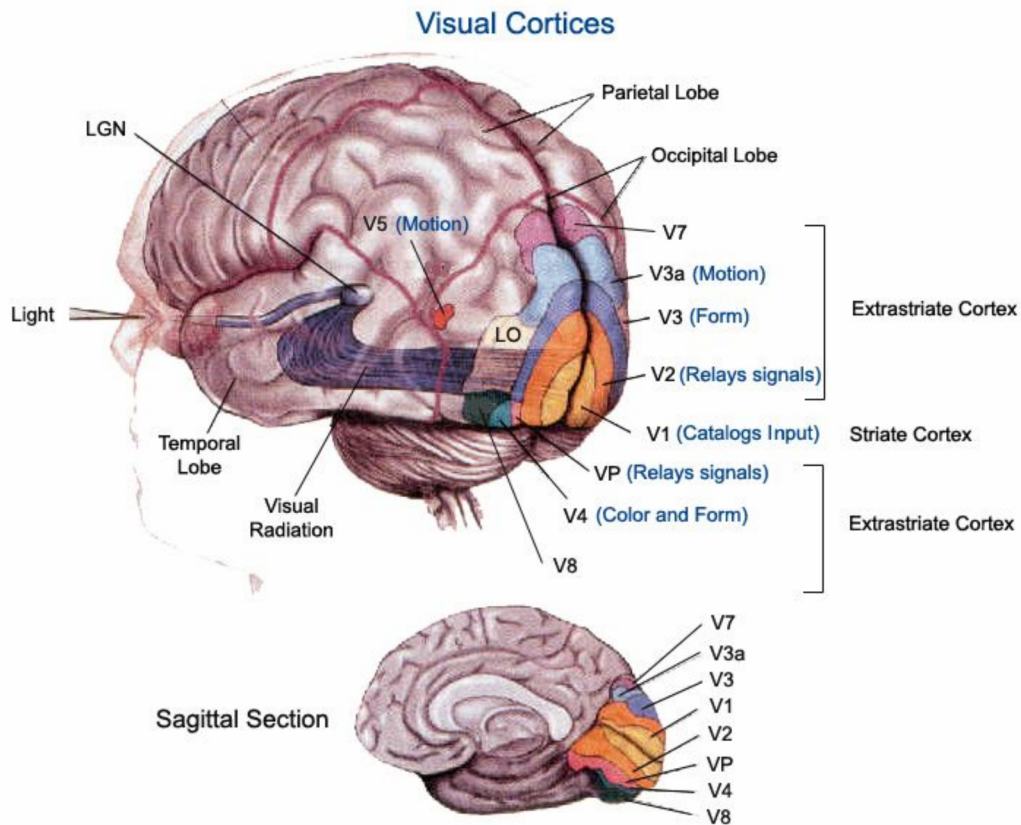


FIGURE 2.6 – Schema representative of the human visual cortex and its different areas (*Visual cortices*)

However, today's understanding of the visual processing does not permit to give a full description of these areas and their inner signal processes. We concentrate here in two principal elements, on which the BIPS component based its bio-inspired model: the primary visual cortex V1 and the dorsal and ventral streams.

#### The primary visual cortex

The primary visual cortex, also known as the striate cortex, corresponds to the area V1. It is the principal receptor of the LGN cells and mostly sends computed information to the V2 area. The work of Hubel and Wiesel in the late 70s, awarded with the Nobel prize in 1981 (Hubel, 1982)(Wiesel, 1982), permits to reveal the columnial organization of V1 and to identify that neurons from this area were responding to edges and light-dark bars and that they were associated to specific orientations. Further work have shown that other stimulus features (e.g. color, motion direction,

spatial frequency) are also distributed iteratively in the columns. It is thought today that the primary visual cortex performs simple filtering to enhance contours and edges information, which are then organized such that they reach more easily the appropriate areas of the cortex (e.g. MT for the movement, V4 for the color).

### The dorsal and ventral streams

The neuron responses in some regions of the cortex suggest that they are highly specialized in some visual processes. For example the middle temporal area (MT) is known to react to motion stimuli, whereas the area V4 is known to react to color stimuli. The existence of two pathways of information processes have been hypothesized for the primate brains: the ventral pathway ("What"), which processes identification of an object and the dorsal pathway ("Where"), which processes the spatial localization of the object (Weller, 1988).

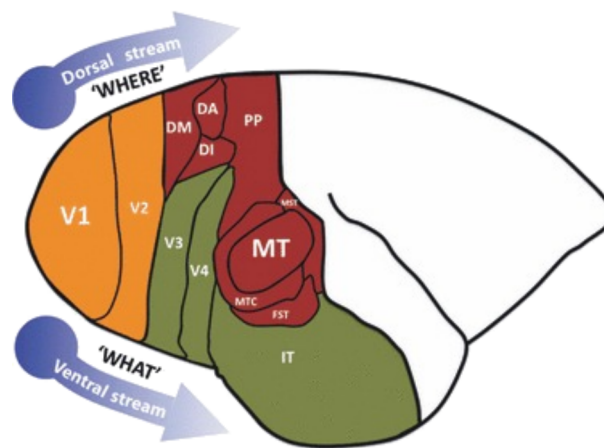


FIGURE 2.7 – Schema representing the dorsal and ventral streams (Bourne, 2010).

The M and P pathways in the cortex, coming from the LGN magno- and parvocellular layers, are thought to be at the origin of these two streams. As the first type of cells are more concerned with the motion, stereopsis and form linked to motion, the second type of cells are more concerned with the colour and shape (Livingstone and Hubel, 1987). The two pathways are often considered as working independently to each others, but several elements tend to indicate that they are working together on the object visualisation (Wang et al., 1999).

### 2.3.5 The brain systems

The BIPS component is also inspired by two features of the brain system: the plasticity of the cortex and the attractor network.

#### The cortex plasticity

For decades, researchers assumed that each part of the brain acquired its specific task in early childhood, but recent works showed that they could adapt to new tasks even in the adult age (Gilbert and Li, 2012). Experimentation done on subjects having neural lesion, showed that other parts of the cortex were able, after a short time,

to replace the deficient neuronal region. Also, experiments done by (Rita, Tyler, and Kaczmarek, 2003) showed that regions associated to a type of sensory input could adapt to another type of sensory input. It is then thought that the mechanisms used in the visual pathway are similar to the mechanisms of other sensors pathway (auditory, touch, etc.), and could be used to do high level processing of the perception inputs.

### Attractor networks

One model of the brain systems sees it as attractor networks (Knierim and Zhang, 2012). The attractor is a mathematical concept describing a state, or a collection of states, which attract all their neighboring states. This concept is especially thought to exist in the hippocampus, for patterns memory (Rennó-Costa, Lisman, and Verschure, 2014), but yet its has been difficult for the biologist to prove the existence of neural circuits performing that task.

### 2.3.6 Conclusion

This Section presented several biological model of the brain vision and systems. Part of those models are well known today (e.g. retina, LGN), but most of them are still at the state of hypothetical models (e.g. visual cortex, plasticity, attractor networks). The understanding of those biological models helps to understand the bio-inspired computer vision algorithms and to foresee the difficulties they might encounter. In the specific case of the BIPS component, this biological overview helps to formalize the concept hidden behind the existing hardware implementation.

## 2.4 The BIPS on-chip

### 2.4.1 Related bio-inspired developments

David Marr's book *Vision* (Marr, Ullman, and Poggio, 2010) is one of the most referenced work in the bio-inspired computer vision domain. In fact, Marr did a major work trying to unify neurobiology, psychophysics and computer vision domains. Unfortunately, as (Medathati et al., 2016) points it out, the study of biological and artificial systems have, since then, drifted apart. We have to admit that the transcription of biological systems can not be straight forward to the hardware, as the former works asynchronously on complex chemical actions in 3D wiring networks, whereas the latter is limited by its clock frequency, its 2D structure and its elementary silicon logic.

The questions are: where does the bio-inspiration start and where does it stop? There are plenty of computer vision developments considering themselves as bio-inspired. Some are trying to build new hardware components closer to biological neuronal systems. Others use the visual pathway model as a working step reference, but don't really care of the real capacity of each internal process.

In order, not to disperse ourselves we will discuss only of two bio-inspired developments: the light perception and the feature extraction process, as they are both based on the most known parts of the visual pathway and they are both relevant for the BIPS method.



## 2.4.2 Light perception

The cameras are the first bio-inspired systems. The use of a shutter, a lens and a sensor is inspired from the mechanics of the eye and the first retina layer: the photoreceptors. The CMOS sensor, most used nowadays, tries to reproduce the Red, Green and Blue sensitivities of the cones and compute the brightness sensitivity of the rods from the intensity of each color. These form the RGB input. Other colorimetric systems have also been developed to better represent the actual incoming light, like the CIE standard system, the Yuv system and the different types of HSV (Hue, Saturation, Value) systems (Ohta and Robertson, 2006).

The incoming light is captured on a matrix of receptive cells, which are evenly distributed. This does not correspond well to the biological system, which perceive the light asynchronously, with a very specific distribution of cones and rods. Some research development are looking into new asynchronous event-based cameras, like (Posch et al., 2014). Although their design is not new (Delbruck, 1993), their development and use are still at their beginning. Mechanical bio-inspired organisations have also been developed like dual vision systems (Hueber et al., 2015)(Hengstler et al., 2007). In the BIPS, the camera input used is a classic Yuv system based on the BT601 norm (*BT.601*).

## 2.4.3 Feature extraction

The extraction of information from the incoming light, called feature extraction, corresponds to the different layers of process in respectively the retina, the LGN, and the primary visual cortex. The feature extraction is commonly used in classical computer vision methods before the higher level processing (detection, tracking, recognition). The process performed in the retina and the LGN could be associated to Laplacian of Gaussian filtering (Marr, Ullman, and Poggio, 2010), showing that low level features could already have an impact on the vision interpretation. The organisation of the V1 area also showed that there were three main types of features: those related to the motion, those related to the contours and shapes and those related to the brightness and color. This organisation is represented in the BIPS also.

## 2.4.4 the Bio-Inspired Perception Sensor

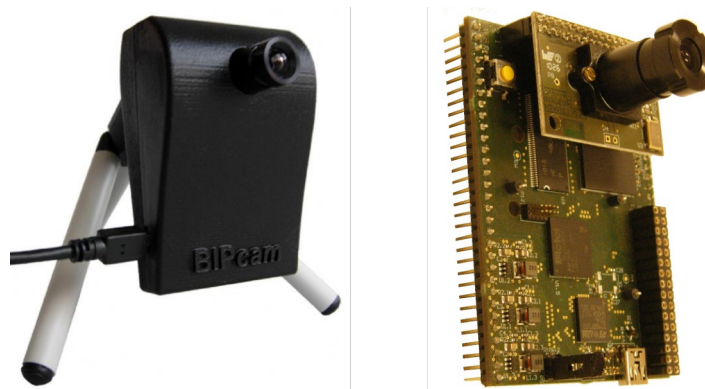
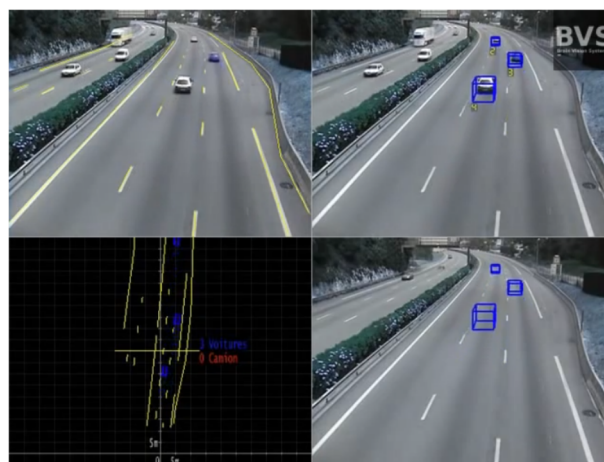
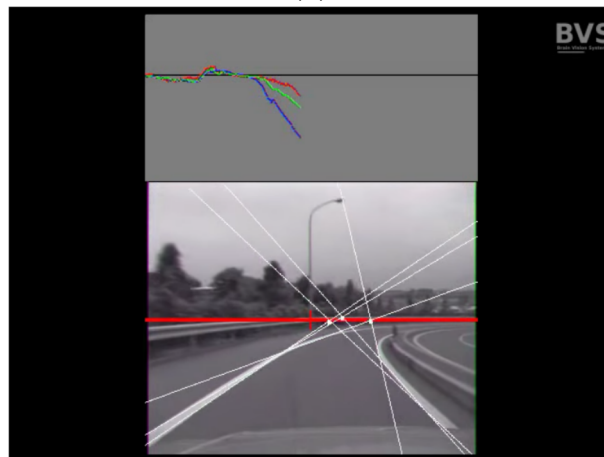


FIGURE 2.8 – Products of BVS-Tech based on the BIPS processor. Left: BIPcam (*BIPcam*); Right: BIPeye (*BIPeye*).

During our research on existing bio-inspired visual processing systems, the processor developed by BVS-Tech (Pirim, 2015) drew our attention. Its development began thirty years ago with the first implementation made by its inventor P. Pirim. Since then, it went up following a bio-inspired approach using the last technological and biological discoveries. Among them, the vision functions described by Hubel and Wiesel (Hubel, 1995), the two streams completion (Wang et al., 1999), the principle of cortex plasticity demonstrated by Bach-y-Rita (Rita, Tyler, and Kaczmarek, 2003) and the attractor networks studied by Rennó-Costa et al. (Rennó-Costa, Lisman, and Verschure, 2014). The last one inspired the peculiar module called "Dynamic Attractor", which permits the convergence to the combined information of the "what" and "where" streams. The module, however, is not directly related to the mathematical attractor concept, except for its expected convergence behavior. Its implementation, formalized in the next Chapter (Chap. 3), uses histograms on local features to converge to description and localization of objects in the scene.



(A)



(B)



(c)

FIGURE 2.9 – Examples of BIPS industrial applications: (A) traffic management, (B) road lanes detection, (C) tracking with robotic eyes (*BVS-Tech website*).

The processor has been used in different industrial applications, like road lane tracking, robotic binocular vision, traffic management (Fig. 2.9) This diversity demonstrated a certain versatility of the system, which is low-power (3W). The combination of these properties widens the application range. However, there is still no formalization of the system enabling an efficient integration within a full visual system. This formalization will help the understanding and therefore the integration in our processing model, while keeping its interesting computational performances. This mathematical formalisation is part of the challenge of this PhD work. The second challenge is the identification of the parameters having to be tuned to get a correct response.

## Chapter 3

# BIPS formalization

### 3.1 Introduction

As introduced in the previous chapter (Sec. 2.4), the Bio-Inspired Perception Sensor (BIPS) developed by Patrick Pirim (Fig. 3.1), is expected to have interesting real-time performances on several detection, tracking and recognition applications and is well optimized for autonomous embedded systems (real-time, low power). Moreover this component is presented as a bio-inspired and generic component that could be adapted to a large set of computer vision applications (Pirim, 2015). The chip has caught the attention of the bio-inspired and hardware computer vision community (*BioComp 2016 – GDR BioComp*) (*Journées NeuroSTIC 2017 | ISIS | NeuroSTIC*) and several entities like the French Directorate General of Armaments (DGA) and the French-German Research Institute of Saint-Louis (ISL), which decided to initiate a PhD work on this subject in order to understand how it works and to investigate its true potential for Defense and security projects. The lack of documentation and scientific explanation about this concept and the absence of a software simulator hinder the use of this method for academic computer vision applications.

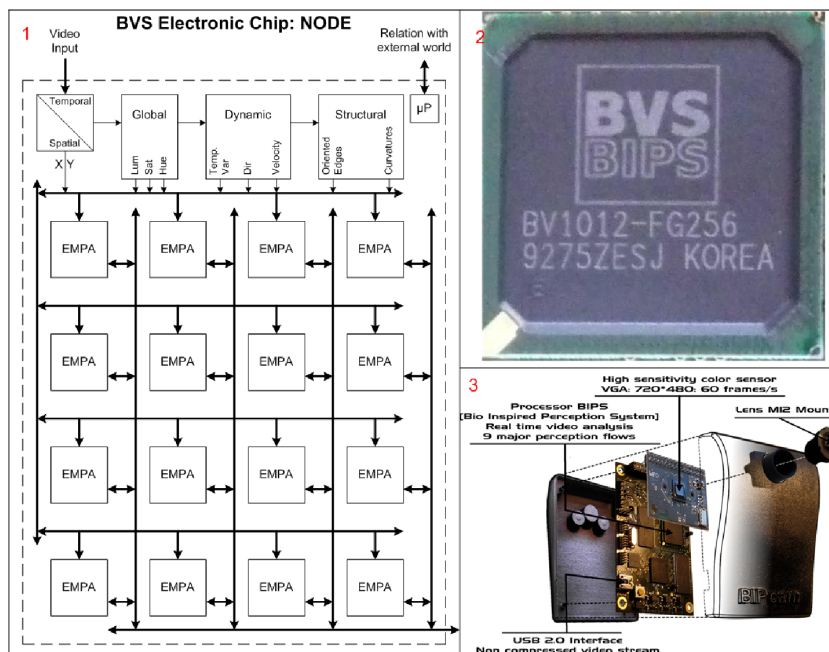


FIGURE 3.1 – Image representing the BIPS component in a BVS-Tech embedded system (Pirim, 2013). 1- Block description of the BIPS; 2- 2013 chip; 3- Complete BVS product.

The development history of the component can explain this situation. The BIPS component has been developed without prior academic mathematical formalization. Its first implementation was made in 1987 (Thuries and Pirim, 1988), then improved along the years with bio-inspired mechanisms directly transcribed into hardware implementation by P. Pirim. The existing literature on the BIPS (Pirim, 2013) (Pirim, 2015)(Pirim, 2016) focuses on the bio-inspired aspects of the component but does not provide any formalization. My discussions with Patrick Pirim and my work on his articles and patents allowed me to establish a mathematical formalization of the methods used, to highlight their properties and to reproduce the BIPS behavior through my own simulator.

The given representations of the BIPS component and its concept -also called USER (Universal SEMantic Representation) concept in (Pirim, 2016)- show its structure into two main parts as presented in Figure 3.2: 1- a set of feature extractions, 2- a simultaneous description and localization of objects of interest. The second process can itself be split into multiple similar modules called Dynamic Attractors (DAs) -also called EMPAs (Electronic Models of Population Activities)-. Each DA processes the description and the localization of one object in the scene. The multiplication of this module with different inputs and combination rules establishes the complete BIPS second part as represented in Figure 3.3.

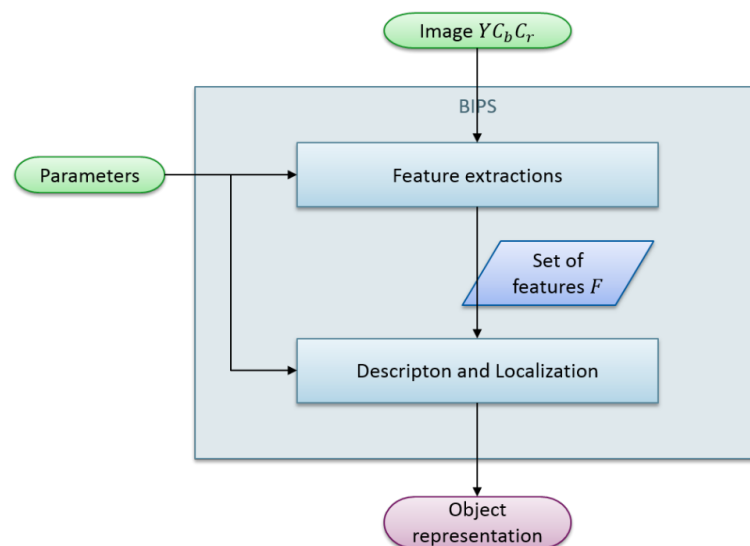


FIGURE 3.2 – Diagram representative of the BIPS's two main processes. Firstly, the feature extractions, which produce a set of local features from the input image. Secondly, the description and localization part, which detects objects of interest in the given feature space and provides a representation of them.

This Chapter investigates the mathematical formalization of the BIPS parts and modules with the goal of bringing to the academic community a complete reusable expression of the method and proofs on its capacities and limits. The chapter is organized as follows. Section 3.2 focuses on the current main feature extractions used by the component and their generic formalization. Section 3.3 presents the basic functions used for the description and localization of one object. Then Section 3.4 investigates the complete description and localization process. Finally, Section 3.5 discusses the highlighted properties and parameters of the BIPS concept.

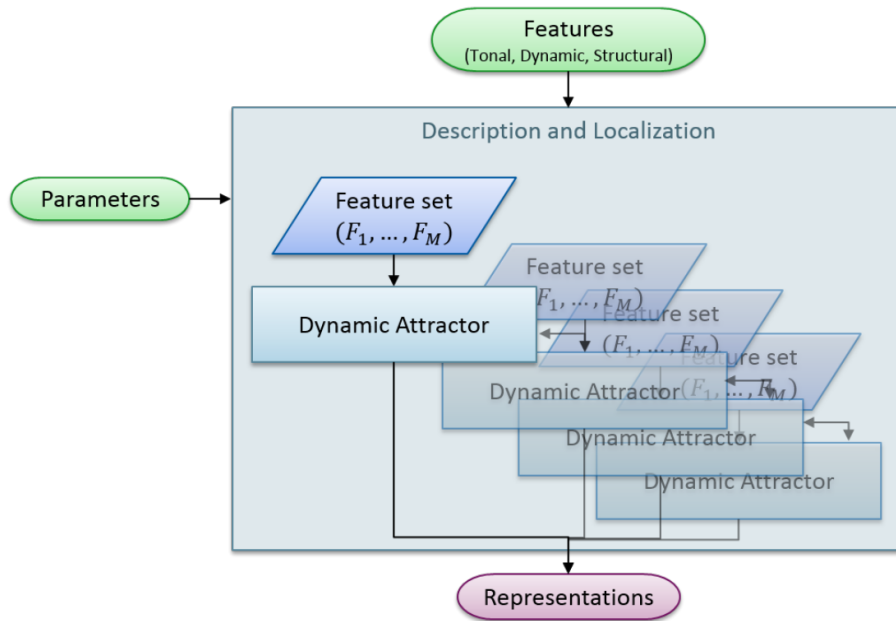


FIGURE 3.3 – Diagram representative of the Description and Localization part and its modules: the Dynamic Attractors (DAs).

## 3.2 Feature extraction

This Section focuses on the mathematical formalization of the first BIPS processing part, which performs in parallel several feature extractions. A generic description of this part is presented in (Pirim, 2013), (Pirim, 2015), and more details can be found in the patents (Pirim, 1998), (Pirim, 2005).

The BIPS extracts nine local features. They are gathered, depending on their nature, into three groups of extractors (see Fig. 3.1). These extractions correspond to early visual features used to construct pre-attentive saliency maps, like in the work of (Itti and Koch, 2000). The first group, deceptively called 'Global', includes the luminance, the saturation and the hue features. To avoid any misunderstanding we change its name to 'Tonal' features. The second group, called 'Dynamic', includes a peculiar estimation of the luminance temporal variation and its corresponding optic flow. Finally, the third group, called 'Structural', includes the gradient and the curvature features. However, in the concept proposed by the inventor this part may process all kinds of perception: vision, audio, touch and other senses. Its formalization is thus not restricted to its current implementation, but needs to embrace a larger panel of feature extractors.

The Sections 3.2.1, 3.2.2 and 3.2.3 investigates respectively the Tonal, the Dynamic and the Structural feature extractions proposed by the BIPS on-chip implementation. A more generic formalization of the extractors is then discussed in Section 3.2.4, before concluding in Section 3.2.5.

### 3.2.1 Tonal features

In the BIPS, the implemented Tonal features correspond to one of the cylindrical representation of the luminance and chroma (Fig. 3.4). It is based on the  $(Y, C_b, C_r)$  Cartesian representation and the polar transformation of its chroma coordinates to find the corresponding hue  $Hu$  and saturation  $Sa$  (Eq. 3.1, 3.2, 3.3). A threshold  $\tau_{sat}$  conditions the hue and allows to associate a group of validated pixels  $\mathcal{O}_{hue}$  with the feature (Eq. 3.4).

**Definition.** Luminance

$$L = Y \quad (3.1)$$

**Definition.** Saturation

$$Sa = \sqrt{C_b^2 + C_r^2} \quad (3.2)$$

**Definition.** Hue

$$Hu = \begin{cases} 2 \arctan \left( \frac{C_r}{C_b + \sqrt{C_b^2 + C_r^2}} \right) & , \text{ if } Sa > \tau_{sat} \\ \text{undefined} & , \text{ if } Sa \leq \tau_{sat} \end{cases} \quad (3.3)$$

**Definition.** Hue validated pixels

$$\begin{aligned} val_{hue} : \mathbb{I} \times \mathbb{J} &\longrightarrow \{0, 1\} \\ p &\longmapsto val_{hue}(p) = \begin{cases} 1 & , \text{ if } Sa > \tau_{sat} \\ 0 & , \text{ if } Sa \leq \tau_{sat} \end{cases} \end{aligned} \quad (3.4)$$

$$\mathcal{O}_{hue} = val_{hue}^{-1}(1)$$

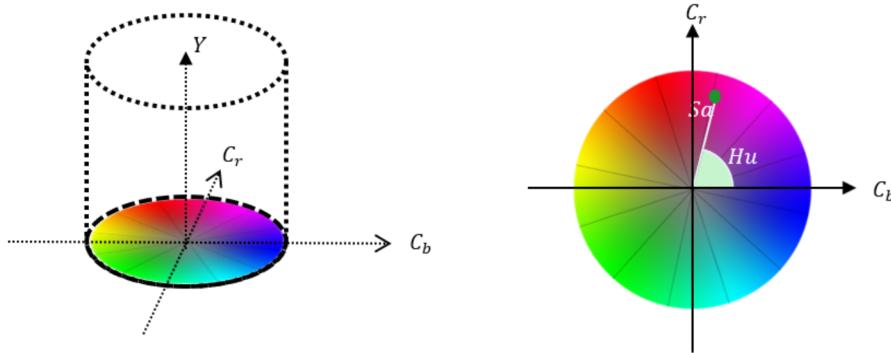


FIGURE 3.4 – Representation of the Tonal values, based on the  $Y, C_b, C_r$  color space.

The light and tone are the most common features used. A multitude of brightness and color spaces have been modeled, a description of them can be found in (Malacara, 2011). Even though the computation of these color spaces comes from the signal analysis domain, their foundation lies on the cones and rods perception in the retina, which plays a major role in the visual perception (Sec. 2.3.2).

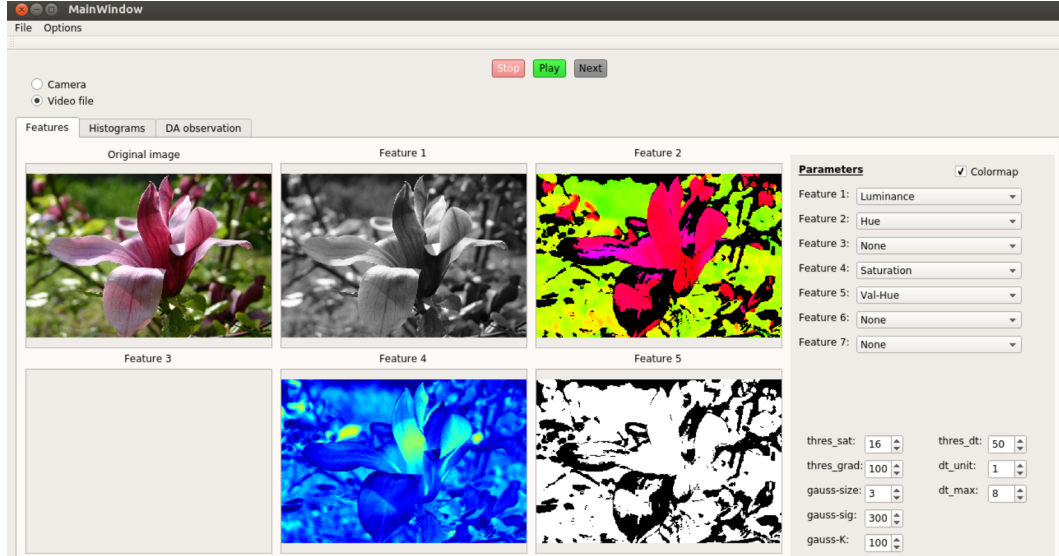


FIGURE 3.5 – Extraction of the tonal features on a Magnolia image (Wikimedia, 2006). Screen-shot of our simulator, the threshold  $\tau_{sat}$  is set at 20. Upper-left: original RGB image; Upper-middle: luminance; Upper-right: hue; Bottom-middle: saturation; Bottom-right: group of validated pixels associated with the hue.

### 3.2.2 Dynamic features

In the BIPS, the implemented Dynamic features correspond to a peculiar background estimation and an associated temporal variation extraction, followed by an unusual optical flow extraction. The background estimation  $\hat{L}$  is based on an exponential smoothing of the luminance using an evolving smoothing factor  $\alpha = 1/2^{D(t)}$ . Thus, the temporal variation  $dt = 2^{D(t)}$  locally adapts to the level of luminance variation (Eq. 3.5, 3.6). A threshold  $\tau_{lum}$  on this variation allows to associate a group of validated pixels  $\mathcal{O}_{D(t)}$  with the feature  $D$  (Eq. 3.7) and to increase (resp. decrease) its value by a unity  $u$  up to its maximum (resp. minimum) value  $D_{\max}$  (resp. 0).

**Definition.** Background estimation

$$\hat{L}(t) = \hat{L}(t-1) + \frac{L(t) - \hat{L}(t-1)}{2^{D(t-1)}} \quad (3.5)$$

**Definition.** Temporal variation

$$D(t) = \begin{cases} D_{\max} & \\ \max(D(t-1) - u, 0) & , \text{ if } |\hat{L}(t) - L(t)| > \tau_{lum} \\ \min(D(t-1) + u, D_{\max}) & , \text{ if } |\hat{L}(t) - L(t)| \leq \tau_{lum} \end{cases} \quad (3.6)$$



**Definition.** Temporal variation validated pixels

$$\begin{aligned}
 val_{D(0)} : \mathbb{I} \times \mathbb{J} &\longrightarrow \{0, 1\} \\
 p &\longmapsto val_{D(0)}(p) = 0 \\
 \\
 val_{D(t)} : \mathbb{I} \times \mathbb{J} &\longrightarrow \{0, 1\} \\
 p &\longmapsto val_{D(t)}(p) = \begin{cases} 1 & , \text{ if } |\hat{L}(t) - L(t)| > \tau_{lum} \\ 0 & , \text{ if } |\hat{L}(t) - L(t)| \leq \tau_{lum} \end{cases} \quad (3.7)
 \end{aligned}$$

$$\mathcal{O}_{D(t)} = val_{D(t)}^{-1}(1)$$

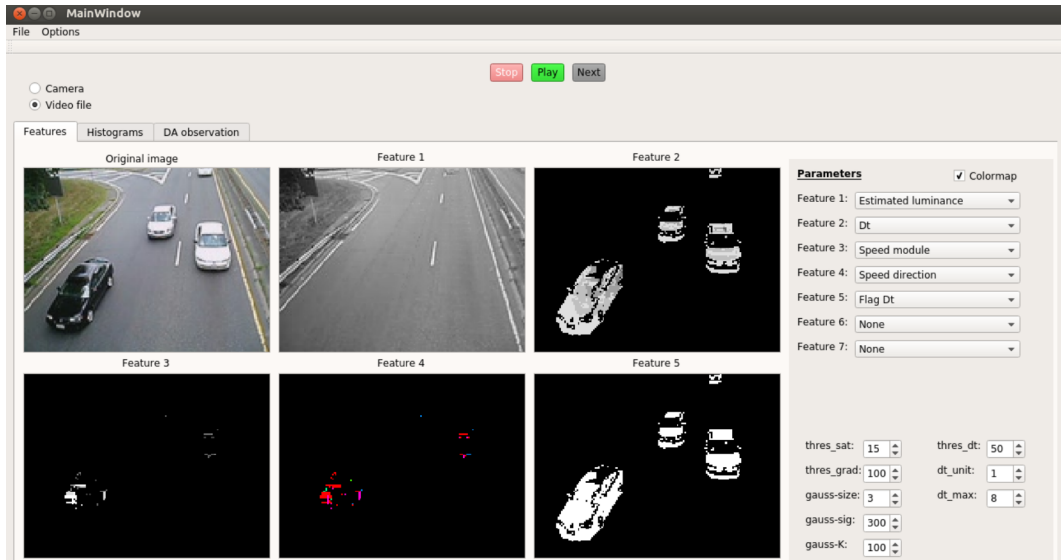


FIGURE 3.6 – Extraction of the dynamic features on the "traffic.mj2" video (*MATLAB and Image Processing Toolbox Release R2018a*). Screen-shot of our simulator, the threshold  $\tau_{lum}$  is set at 50, the step unit  $u$  at 1 and the maximum temporal variation at  $D_{max} = 8$ . Upper-left: original RGB image; Upper-middle: estimated luminance; Upper-right: temporal variation; Bottom-left: velocity module; Bottom-middle: velocity direction; Bottom-right: group of validated pixels associated with the temporal variation.

In the biological point of view, even though studies showed which parts of the brain play an important role in processing motion information, it is not known today which exact form of features they use. The choice of specific dynamic extraction is just one of many possibilities helping in motion processing (Burr and Thompson, 2011). Among them, the optical flow is a well-known feature in computer vision (Agarwal, Gupta, and Singh, 2016), but the chosen implementation of it in the BIPS does not correspond to the classical and academic approaches. The proposed method is quite empirical and limited by the peculiar movement model, which is based on the spatial variation of the specific time parameter  $D$ . Moreover the Dynamic feature extracted have a small range of possible values -represented by 3bits only-, which led us to put aside these features in our future experiments.

### 3.2.3 Structural features

In the BIPS, the implemented Structural features correspond to the gradient module  $Mo$  and angle  $An$  resulting from the convolution of the luminance and the chosen edge filter kernel  $W$  followed by a polar transformation (Eq. 3.8, 3.9, 3.10). A threshold  $\tau_{mo}$  conditions the angle and allows to associate a group of validated pixels  $\mathcal{O}_{ang}$  to the feature (Eq. 3.11). An empirical extraction of the angle variation is also proposed and roughly associated to the curvature.

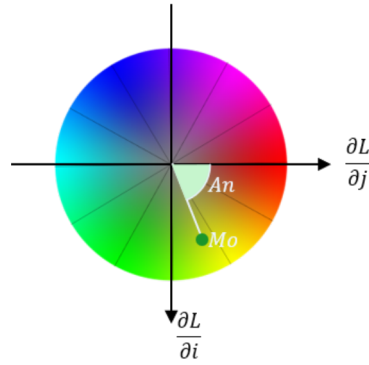


FIGURE 3.7 – Representation of the gradient coordinates. For a better visualization, the values are associated with colors.

**Definition.** Gradient

$$\frac{\partial L}{\partial j} = conv(W, L) \text{ and } \frac{\partial L}{\partial i} = conv(W^t, L) \quad (3.8)$$

**Definition.** Gradient module

$$Mo = \sqrt{\left(\frac{\partial L}{\partial j}\right)^2 + \left(\frac{\partial L}{\partial i}\right)^2} \quad (3.9)$$

**Definition.** Gradient angle

$$An = \begin{cases} 2 \arctan \left( \frac{\frac{\partial L}{\partial i}}{\frac{\partial L}{\partial j} + \sqrt{\left(\frac{\partial L}{\partial j}\right)^2 + \left(\frac{\partial L}{\partial i}\right)^2}} \right) & , \text{ if } Mo > \tau_{mo} \\ \text{undefined} & , \text{ if } Mo \leq \tau_{mo} \end{cases} \quad (3.10)$$

**Definition.** Gradient angle validated pixels

$$\begin{aligned} val_{ang} : \mathbb{I} \times \mathbb{J} &\longrightarrow \{0, 1\} \\ p &\longmapsto val_{ang}(p) = \begin{cases} 1 & , \text{ if } Mo > \tau_{mo} \\ 0 & , \text{ if } Mo \leq \tau_{mo} \end{cases} \end{aligned} \quad (3.11)$$

$$\mathcal{O}_{ang} = val_{ang}^{-1}(1)$$

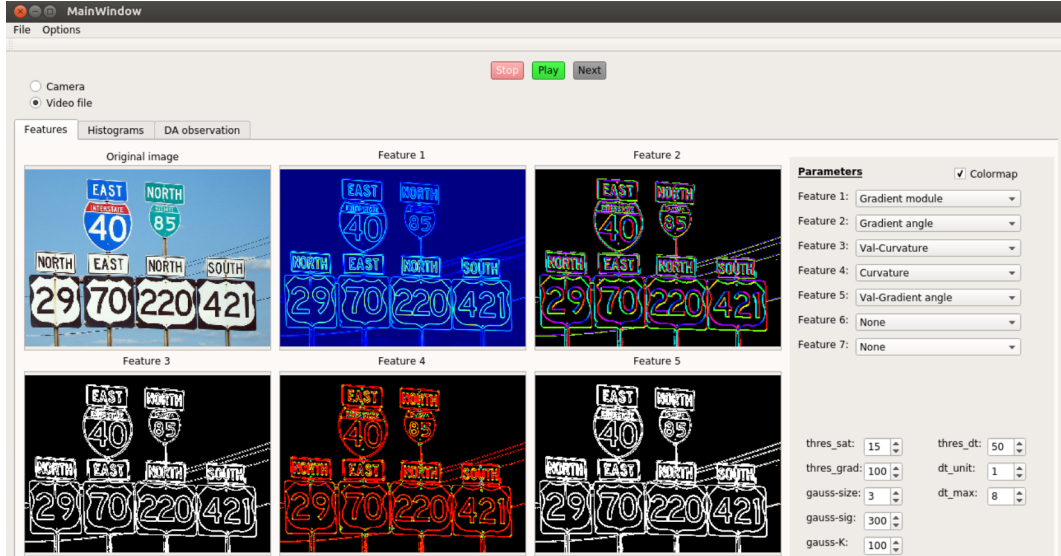


FIGURE 3.8 – Extraction of the structural features on a road signs image (Commons, 2012). Screen-shot of our simulator, the threshold  $\tau_{mo}$  is set at 4% of the maximum possible value. Upper-left: original RGB image; Upper-middle: gradient module; Upper-right: gradient angle; Bottom-left: pixel validation associated with the angle variation; Bottom-middle: angle variation; Bottom-right: validated pixels associated with the gradient angle.

The edge extraction is performed in the visual cortex by cells of the first layer, which have been identified by (Wiesel, 1982) and (Hubel, 1982). Even though the work of (Marçelja, 1980) and (Daugman, 1980) showed that the Gabor kernels gives a better reproduction of the cells activities, the Gaussian kernels model has long being used. Just as the Canny edge detector (Canny, 1986), the BIPS edge extraction uses the first derivative of a 2D Gaussian steerable filter (Eq. 3.12). However, the given matrix in the patent (Eq. 3.13) does not exactly correspond to the value of such filter. With a comparative analysis of the given values and the filter model values while varying its parameters, we are able to give its closest representation, which corresponds to  $n = 3$ ,  $K = 15.1$  and  $\sigma = 1.44$ .

**Definition.** First derivative of a 2D Gaussian steerable filter of size  $2n + 1$

$$\forall i \in [-n, n], \forall j \in [-n, n],$$

$$W(i, j) = \frac{\partial G}{\partial j}(i, j) = K \frac{-j}{2\pi\sigma^4} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right) \quad (3.12)$$

**Definition.** Fixed kernel from the patent (Pirim, 2001)

$$W = \begin{pmatrix} 0.022 & 0.049 & 0.050 & 0.0 & -0.050 & -0.049 & -0.022 \\ 0.074 & 0.163 & 0.167 & 0.0 & -0.167 & -0.163 & -0.074 \\ 0.0151 & 0.334 & 0.342 & 0.0 & -0.342 & -0.334 & -0.151 \\ 0.192 & 0.424 & 0.434 & 0.0 & -0.434 & -0.424 & -0.192 \\ 0.0151 & 0.334 & 0.342 & 0.0 & -0.342 & -0.334 & -0.151 \\ 0.074 & 0.163 & 0.167 & 0.0 & -0.167 & -0.163 & -0.074 \\ 0.022 & 0.049 & 0.050 & 0.0 & -0.050 & -0.049 & -0.022 \end{pmatrix} \quad (3.13)$$

For our experiments, we chose to use the generic kernel construction rather than using the fix kernel of the chip, as this gives us more flexibility on the choice of the matrix size and variance used.

### 3.2.4 Formalization

The BIPS first processing part, according to (Pirim, 2015), should be able to provide a wider panel of features (e.g. visual, sound) However it is not said which conditions they must fulfill to maintain the BIPS properties.

Based on our BIPS concept study (see Sec. 3.3, 3.4), we are able to propose a formalized definition of the extracted features (Eq. 3.14), such that the BIPS computation and detection properties are preserved. The feature is represented by an application from a domain  $\mathbb{P}$  to a feature space  $\Omega$ .

**Definition.** Feature

$$\begin{aligned} F: \mathbb{P} &\longrightarrow \Omega \\ p &\longmapsto f = F(p), \end{aligned} \quad (3.14)$$

To ensure the histogram computation and convergence performed by the following processing part of the BIPS, the definition must respect the following conditions:

- The set  $\mathbb{P}$  must be the same for all connected features in the Dynamic Attractor (DA) modules
- The space  $\Omega$  must be represented by a finite and totally ordered set of values

#### Visual feature example

In the case of visual local features, the set  $\mathbb{P}$  can be the pixel set  $\mathbb{I} \times \mathbb{J} = \llbracket 0; H \rrbracket \times \llbracket 0; W \rrbracket$ , where  $H$  and  $W$  are the height and width of the input image. For the luminance, the set representing  $\Omega$  can be  $\{16, 17, \dots, 235\}$  and the corresponding application  $L(i, j) = \text{floor}(Y(i, j))$ . For more precision on the luminance the chosen set could be  $\{16, 16.25, \dots, 234.75, 235\}$  and the corresponding application  $L(i, j) = \frac{\text{floor}(Y(i, j) * 4)}{4}$ .

#### Cyclic space case

The feature space can be cyclic as for the hue or the gradient angle, in this case if the representing set is  $G = \{g_0, \dots, g_{n-1}\}$  the selection performed by the Feature Mode Detectors (FMD), presented in the following Section (3.2) would separate  $g_0$  and above from  $g_{n-1}$  and beneath whereas they correspond to the same object.

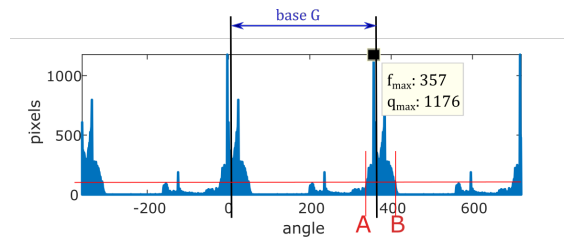


FIGURE 3.9 – Boundaries selection in the cyclic case, when the selection is around the upper values of the set.

To avoid such cut we propose an extension of the BIPS: the chosen representing set for  $\Omega$  is  $\{g_{-(n-1)}, \dots, g_{2n+1}\}$ . The points of the initial set  $G$  are repeated three times allowing to detect the right boundaries around a peak of the feature histogram without changing the algorithm as shown in Figure 3.9. Only the histogram mode  $f_{\max}$  is still chosen in  $G$ .

### 3.2.5 Conclusion

The BIPS first part presented in this Section tries to mimic the perception and information extraction of the human senses. In its current implementation, several local visual feature extractions can be performed in parallel and on the fly.

The luminance is directly extracted from the input  $(Y, C_b, C_r)$ . The hue, the saturation and the temporal variation are extracted pixel wise and without delay as the polar transform -computed with the CORDIC method (Volder, 1959)- and the background estimation can be obtained on the fly. The gradient module and angle extractions depends on the size of the convolution kernel, as well as the velocity and the angle variation extractions depend on the size of the surrounding matrix used for their computation. In the BIPS the convolution kernel is a  $7 \times 7$  matrix, the velocity uses a  $17 \times 17$  matrix, and the angle variation uses a  $3 \times 3$  matrix, it implies respectively a 3 lines delay, and 8 lines delay and a single line delay for their computation. This is summarized in Table 3.1.

Extraction	Computation	Dependency
Tonal	no delay	input pixel $(Y, C_b, C_r)$
Temporal variation	no delay	luminance pixel $L = Y$
Velocity	8 lines delay	temporal variation $17 \times 17$ surrounding matrix
Gradient	3 lines delay	luminance $7 \times 7$ surrounding matrix
Angle variation	1 line delay	gradient angle $3 \times 3$ surrounding matrix

TABLE 3.1 – Implemented feature extractions computation time and dependency to other pixels and features.

Some of the experimental features proposed by the BIPS inventor P. Pirim can be linked to well-known extractions like the HLS and the gradient extractions. But, others like the dynamic features are unique and peculiar. Moreover, these peculiar features present a lack of dynamic -only 3bit-, which does not allow much segmentation possibilities. The proposed concept, though, allows to implement other features.

We have been able to build, based on our formalization of the BIPS detection process presented in the following Section (Sec. 3.3), a generic model of the feature extraction usable in the BIPS. This model can be filled by any kind of sensory features as long as they are represented by a finite and totally ordered set of values. The extractions performed in this first part of the BIPS build a set of features, which will become the inputs of the second part of the BIPS.

### 3.3 Description and Localization: DA module

This Section focuses on the formalization and simulation of the Dynamic Attractor (DA), which is the main module of the BIPS second part. It performs, in the visual case, the simultaneous description and localization of an object in the image, which is in the generic case of sensory features the result of a detection in the DA associated feature space.

Its first description appears in the patent (Thuries and Pirim, 1988), with a more detailed description in the patent (Pirim, 2001). Then several similar definitions have been presented: in (Pirim, 2013) under the name EMPA (Electronic Models of Population Activities); in (Pirim, 2015) and (Pirim, 2016) under its current name Dynamic Attractor (DA). All these descriptions show its decomposition into three identical sub-modules (see Fig. 3.10). We called them Feature Mode Detectors (FMDs), based on their actual function.

Firstly, the Feature Mode Detector (FMD) is presented in Section 3.3.1, then these equations are extended for the Dynamic Attractor (DA) formalization in Section 3.3.2 with an analysis of its computational capacities. The expected behavior of the method is then illustrated on synthetic images in Section 3.3.3, before concluding in Section 3.3.4.

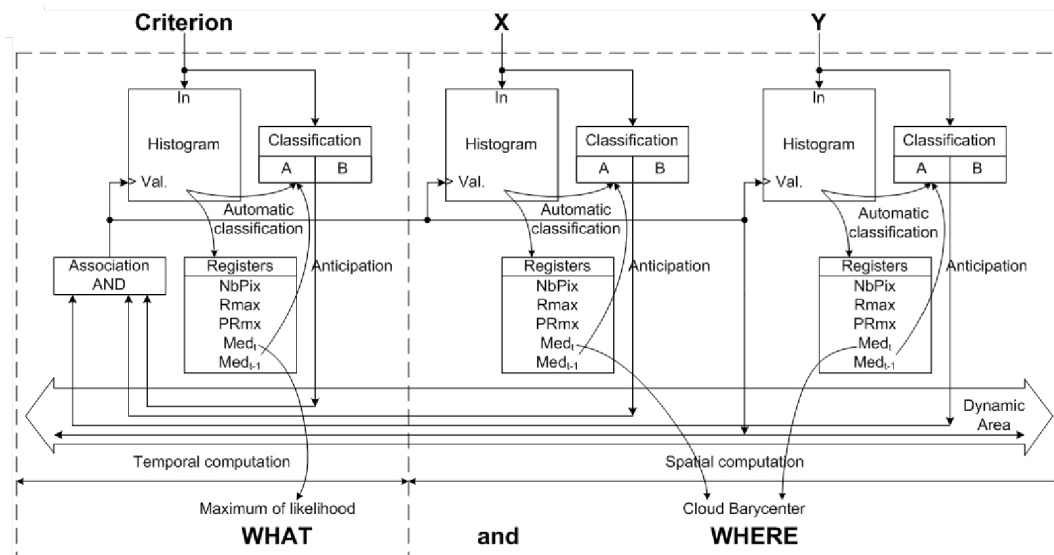


FIGURE 3.10 – Image representing the Dynamic Attractor (DA) and its decomposition into three identical sub-modules (Pirim, 2013).

### 3.3.1 Feature Mode Detector (FMD)

#### Module description

A description of the Feature Mode Detector (FMD) is presented in (Pirim, 2013) and summarized in the article by the Figure 3.11-Left. But we propose a simplified model in Figure 3.11-Right based on our understanding of its operation.

The FMD receives two synchronized data flows: one coming from the data feature extractions (referred to as "Criterion"), the other coming from a bus connected with other FMDs, which validated or not the computation for the incoming pixel ("Val."). This validation forms a mask on the analyzed data. Then the FMD performs two principal operations: firstly, the selected feature histogram is computed under the mask condition ("Histogram" block); then, boundaries selecting the maximum or all maximum peaks are computed ("Automatic classification"), to which a value of shift prediction can be added ("Anticipation"). A third operation is performed in parallel: the median of the histogram computation. The FMD has two distinctive sets of outputs: first the validation (binary value per pixel) obtained through the limitation of its boundaries ("Classification" block), then the registered data ("Registers" block).

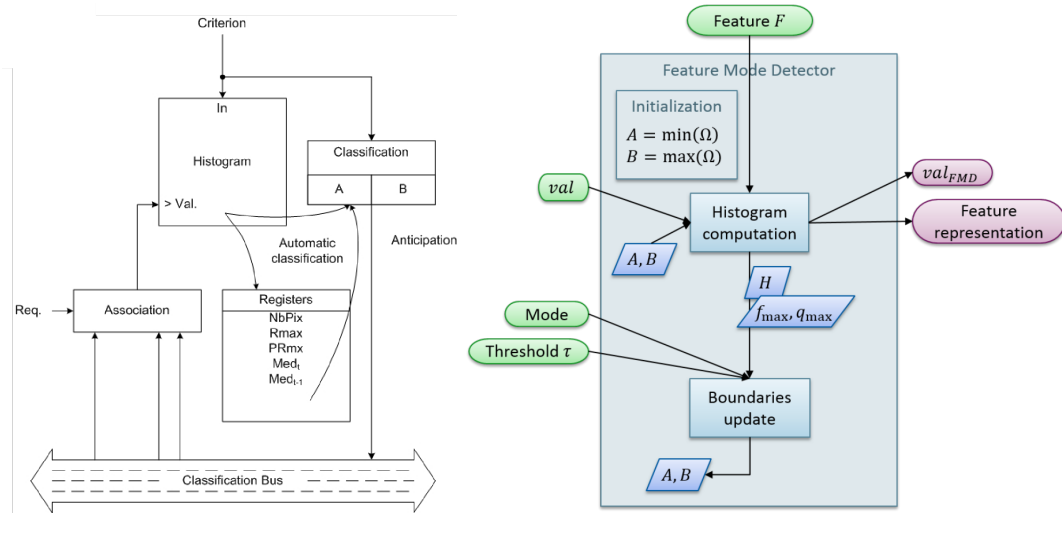


FIGURE 3.11 – Image representing the FMD. Left: Schema from (Pirim, 2013); Right: Simplified diagram.

#### Inputs

The feature input can be formalized by the generic application  $F$  defined in Section 3.2.4 (Eq. 3.14) and the validation can be formalized by the following application  $val$  (Eq. 3.15).

**Definition.** Validation

$$\begin{aligned} val : \mathbb{I} \times \mathbb{J} &\longrightarrow \{0, 1\} \\ p &\longmapsto val(p) \end{aligned} \quad (3.15)$$

### Histogram computation

The FMD operation begins with the construction of the conditioned feature histogram  $H$ . It is computed on the fly and conditioned by the validation coming from the other FMDs and the two already existing boundaries of the feature values,  $A$  the minimum and  $B$  the maximum (Eq. 3.16). The histogram mode can be stored while doing this computation, its value  $q_{\max}$  and its corresponding feature value  $f_{\max}$  are then used in the boundary update computation. As several  $f \in \Omega$  can be  $f_{\max}$ , we chose the same definition as in the chip, that is the minimal one. But we must keep in mind that the choice could be made different to improve the mode detection.

**Definition.** Conditioned feature histogram

$$H: \Omega \longrightarrow \mathbb{N}$$

$$f \longmapsto H(f) = \mathbf{card} \left\{ p \in \mathbb{I} \times \mathbb{J} \left/ \begin{array}{l} val(p) = 1, \\ F(p) = f, \\ A \leq f \leq B \end{array} \right. \right\} \quad (3.16)$$

**Definition.** Histogram maximum

$$q_{\max} = \max_{f \in \Omega} (H(f)) \quad (3.17)$$

**Definition.** Maximum associated feature

$$f_{\max} = \min(\operatorname{argmax}_{f \in \Omega} (H(f))) = \min(H^{-1}(q_{\max})) \quad (3.18)$$

### Boundaries update

Once the feature histogram and its maximum are computed, the boundaries  $A$  and  $B$  can be updated. There are two ways of computing the new boundaries. One way, the Mode 1, gives a surrounding of the maximal peak of the histogram (Eq. 3.19). The other way, the Mode 2, gives a surrounding of all histogram peaks above a threshold (Eq. 3.20).

In Mode 1, the research of the new boundaries starts from the maximum associated feature  $f_{\max}$  then looks for them by increasing (resp. decreasing) the feature value as long as the associated number of pixels per feature  $H(f)$  stays above the chosen threshold  $\tau$ .

In Mode 2, the research starts from the minimum (resp. the maximum) feature value and looks for the boundaries by increasing (resp. decreasing) the feature value as long as the associated number of pixels per feature  $H(f)$  is under the chosen threshold  $\tau$ . The Mode 2 definition, however, does not work with cyclic features.

There are two definitions of the updated boundaries that can be chosen, one is inclusive, the other exclusive. We choose to work with the first one, but for information the second one will be given in Section A.1.3 and reminded for the definitions and proofs for which this choice has an impact.

**Definition.** Mode 1

$$A_{new} = \min \left\{ f \in \Omega, f \leq f_{\max} \left/ \forall f' \in [f; f_{\max}], H(f') > \tau \right. \right\}$$

$$B_{new} = \max \left\{ f \in \Omega, f \geq f_{\max} \left/ \forall f' \in [f_{\max}; f], H(f') > \tau \right. \right\} \quad (3.19)$$



**Definition. Mode 2**

$$\begin{aligned} A_{new} &= \max \left\{ f \in \Omega \mid \forall f' \in [\min(\Omega); f], H(f') \leq \tau \right\} \\ B_{new} &= \min \left\{ f \in \Omega \mid \forall f' \in ]f; \max(\Omega)], H(f') \leq \tau \right\} \end{aligned} \quad (3.20)$$

*Remark.* For the new boundaries to exist, the threshold must verify the following constraints:  $q_{\max} > \tau \geq 0$ . This is always verified in the chip as the chosen threshold is a fraction of  $q_{\max}$ , which can not be null.

## Outputs

The FMD has two kinds of outputs: firstly, the validation of each pixel (Eq. 3.21), then a set of data representing the feature mode. It is composed of the computed histogram values  $q_{\max}$  and  $f_{\max}$  (Eq. 3.17 and Eq. 3.18), as well as the number of histogram pixels  $N$  (Eq. 3.23), and the updated boundaries  $A_{new}$  and  $B_{new}$ . This selection can be summarized by the Figure 3.12. In the chip, the median is also computed and used in the boundaries update to anticipate the distribution variation of the histogram when the dynamic process is done on a video stream and not only on a static image, this aspect of the method will be explained in Section 6.2.

**Definition. FMD validation**

$$\begin{aligned} val_{FMD}(p) : \mathbb{P} &\longrightarrow \{0, 1\} \\ p &\longmapsto val_{FMD}(p) = (A \leq F(p) \leq B) \end{aligned} \quad (3.21)$$

**Definition. Validated pixel set**

$$\mathcal{O} = F^{-1}([A, B]) \cap val^{-1}(1) \quad (3.22)$$

**Definition. Number of pixels**

$$N = \sum_{f \in \Omega} H(f) = \mathbf{card}(\mathcal{O}) \quad (3.23)$$

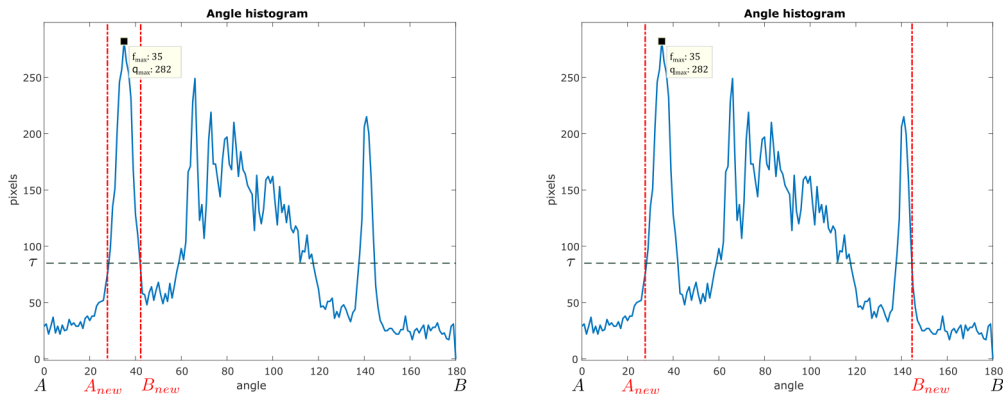


FIGURE 3.12 – Histogram computation and new boundaries selection by a FMD on the gradient angle of a road image. Left: the Mode 1 selects the maximum peak ( $f_{\max}, q_{\max}$ ); Right: the Mode 2 selects all peaks above the threshold  $\tau$ .

### 3.3.2 Dynamic Attractor (DA)

#### Module description

In the chip, the Dynamic Attractor (DA) corresponds to a group of three FMDs (Fig. 3.10). Their validation outputs ("Classification" blocks) and information coming from other DAs ("Dynamic Area" block) are associated ("Association AND" block) and re-injected as the validation inputs ("Val."). The feature inputs are chosen as follows: two of them are spatial features ("X" and "Y") and one is a descriptive feature ("Criterion") (luminance, hue, etc.). This association is inspired from the "What" and "Where" cortex processes and the hypothesis implies that both streams are interconnected and work simultaneously (Sec. 2.3.4). The "What" stream is computed by the FMDs working on descriptive features, the "Where" stream is computed by the FMDs working on spatial features. And each of them is dependent of the other outputs. A simplified representation of the module can be seen in Figure 3.13.

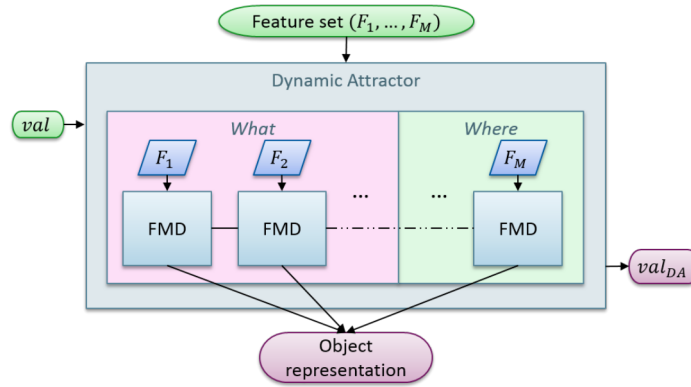


FIGURE 3.13 – Diagram representative of the Dynamic Attractor (DA). Its current implementation is composed of three FMDs, one on a descriptive feature ("What") and two on the spatial features ("Where").

#### Inputs

The choice of only three features inputs is arbitrary and could be done differently. The operation of the DA leads to the detection of a group of points in the multidimensional feature input space. This detection is the result of the iteration of the FMD processes on the same scene. The input set of features can be formalized by the vector of applications  $F$  (Eq. 3.24) selected from the feature extraction part ( $X$  and  $Y$  included). The validation input coming from other DAs and/or external conditions can be formalized like the application  $val$  (Eq. 3.15).

**Definition.** Features set input

$$\begin{aligned}
 F = (F_1, \dots, F_M) : \mathbb{I} \times \mathbb{J} &\longrightarrow \Omega = (\Omega_1, \dots, \Omega_M) \\
 (i, j) &\longmapsto f = F(i, j), \\
 (f_1, \dots, f_M) &= (F_1(i, j), \dots, F_M(i, j))
 \end{aligned} \tag{3.24}$$

### Iterative process

The sequential and cyclic application of the FMDs operations allows to converge to a stable framing  $P \subset \Omega$  (Eq. 3.29). This framing gives a representation in the multidimensional space of one of the object in the input image. The iterative process implies small changes in the Histogram computation and the Boundaries update formalization (Eq. 3.25, 3.26, 3.27 and 3.28).

**Algorithm step.** Initialization  $k = 0$

$$\forall m \in \llbracket 1, M \rrbracket, \begin{cases} A_m^0 = \min(\Omega_m) \\ B_m^0 = \max(\Omega_m) \end{cases} \quad (3.25)$$

**Algorithm step.** Histogram computation,  $n \in \llbracket 1, M \rrbracket, k \equiv n \pmod{M}$

$$\begin{aligned} H_n^k : \Omega_n &\longrightarrow \mathbb{N} \\ f &\longmapsto H_n^k(f) = \mathbf{card} \left\{ p \in \mathbb{I} \times \mathbb{J} \left/ \begin{array}{l} \text{val}(p) = 1, \\ F_n(p) = f, \\ \forall m \in \llbracket 1, M \rrbracket, \\ A_m^k \leq F_m(p) \leq B_m^k \end{array} \right. \right\} \\ q_{\max}^{n,k} &= \max_{f \in \Omega_n} (H_n^k(f)) \\ f_{\max}^{n,k} &= \min(\mathop{\text{argmax}}_{f \in \Omega_n} (H_n^k(f))) = \min((H_n^k)^{-1}(q_{\max}^{n,k})) \end{aligned} \quad (3.26)$$

**Algorithm step.** Boundaries update: Mode 1,  $n \in \llbracket 1, M \rrbracket, k \equiv n \pmod{M}$

$$\begin{aligned} A_n^{k+1} &= \min \left\{ f \in \Omega_n, f \leq f_{\max}^{n,k} \left/ \begin{array}{l} \forall f' \in [f; f_{\max}^{n,k}], \\ H_n^k(f') > \tau_n^k \end{array} \right. \right\} \\ B_n^{k+1} &= \max \left\{ f \in \Omega_n, f \geq f_{\max}^{n,k} \left/ \begin{array}{l} \forall f' \in [f_{\max}^{n,k}; f], \\ H_n^k(f') > \tau_n^k \end{array} \right. \right\} \\ \text{and } \forall m \in \llbracket 1, M \rrbracket, m \neq n, \\ A_m^{k+1} &= A_m^k \\ B_m^{k+1} &= B_m^k \end{aligned} \quad (3.27)$$

**Algorithm step.** Boundaries update: Mode 2,  $n \in \llbracket 1, M \rrbracket, k \equiv n \pmod{M}$

$$\begin{aligned} A_n^{k+1} &= \max \left\{ f \in \Omega_n \left/ \begin{array}{l} \forall f' \in [\min(\Omega_n); f], \\ H_n^k(f') \leq \tau_n^k \end{array} \right. \right\} \\ B_n^{k+1} &= \min \left\{ f \in \Omega_n \left/ \begin{array}{l} \forall f' \in ]f; \max(\Omega_n)], \\ H_n^k(f') \leq \tau_n^k \end{array} \right. \right\} \\ \forall m \in \llbracket 1, M \rrbracket, m \neq n, \\ A_m^{k+1} &= A_m^k \\ B_m^{k+1} &= B_m^k \end{aligned} \quad (3.28)$$

**Definition.** Framing

$$P = \lim_{k \rightarrow +\infty} \prod_{m=1}^M [A_m^k, B_m^k] \quad (3.29)$$

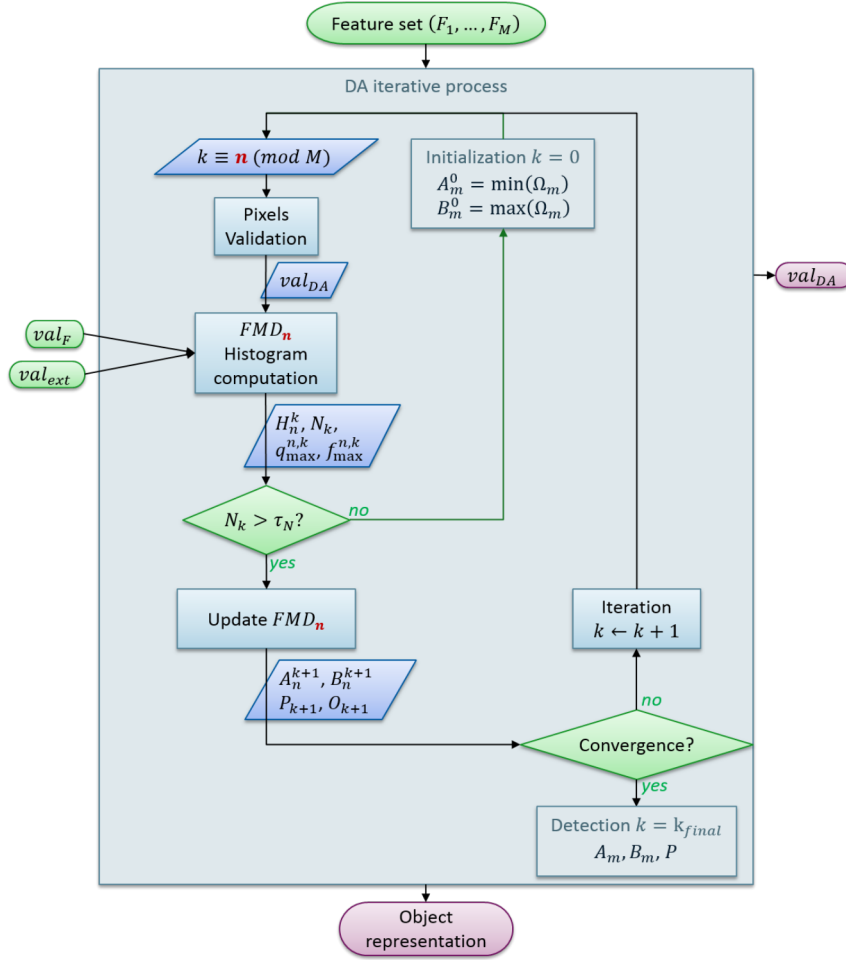


FIGURE 3.14 – Diagram representing the steps of the DA iterative process and the evolution of its elements.

### DA breakpoint

The framing  $P_k = \prod_{m=1}^M [A_m^k, B_m^k]$  is decreasing and converges to  $P$  in finite time. However in the chip, the convergence breakpoint does not necessarily corresponds to the DA breakpoint. In fact, when there is nothing to detect but noise the DA could be running for a long time before detecting a noisy object. Consequently, a threshold  $\tau_N$  on the object size is added. The number of selected pixels  $N_k$  (Eq. 3.32) is used to avoid these cases. While  $N_k > \tau_N$ , the detected object is considered interesting, but once  $N_k \leq \tau_N$  the object is considered to be irrelevant and the DA is re-initialized (Eq. 3.25) for a run in another part of the feature space.

**Definition.** DA validation

$$\begin{aligned}
 val_{DA}^k : \mathbb{P} &\longrightarrow \{0, 1\} \\
 p &\longmapsto val_{DA}^k(p) = \bigwedge_{m=1}^M (A_m^k \leq F_m(p) \leq B_m^k)
 \end{aligned} \tag{3.30}$$

**Definition.** Validated pixel set

$$\mathcal{O}_k = F^{-1} \left( \prod_{m=1}^M [A_m^k, B_m^k] \right) \cap \text{val}^{-1}(1) \quad (3.31)$$

$$\begin{aligned} \forall m \in \llbracket 1, M \rrbracket, \forall k \in \mathbb{N}, \\ N_k &= \sum_{f \in \Omega_m} H_m^k(f) \\ &= \mathbf{card}(\mathcal{O}_k) \end{aligned} \quad (3.32)$$

## Outputs

The DA outputs at iteration  $k$  can be decomposed in two parts: first the association of the FMD validation outputs  $\text{val}_{DA}^k$  (Eq. 3.30), then the data outputs of each FMD ( $N_k, f_{\max}^{m,k}, q_{\max}^{m,k}, A_m^k, B_m^k$ ).

An additional output can be taken into account: the validation of the convergence, which is reached in finite time as proved in Appendix A.2. To validate the convergence the comparison of the framing before and after going through all  $M$  FMDs can be used (Eq. 3.33).

**Definition.** Convergence validation

$$\bigwedge_{m=1}^M (A_m^k == A_m^{k-M}) \wedge (B_m^k == B_m^{k-M}). \quad (3.33)$$

## Computational costs

The mathematical formalization of the DA processes allows us to give an evaluation of their computational costs.

The DA iterative process consists of a sequential use of the FMD processes. Therefore, as the first step needs to go all over the image, the following step can not be working on the fly and the feature images must be stored. The computational cost of the DA is the multiplication of one FMD computational cost by the number of iterations before reaching the DA breakpoint. This is summarized in Table 3.2.

The time cost for each Histogram computation is of complexity  $\mathcal{O}(H \times W)$  (image size) and the memory cost is  $r_F \times H \times W$  bits, where  $r_F$  is the chosen resolution of the feature (8 or 10bits in the chip). The boundaries update needs to go through all the histogram values implying a computational cost of complexity  $\mathcal{O}(\mathbf{card}\{\Omega\})$  and a memory use of  $r_H \times \mathbf{card}\{\Omega\}$  bits, where  $r_H$  is the chosen resolution of the histogram values (8, 10, 16 or even 32bits can be needed). Other elements like  $f_{\max}, q_{\max}, N, A$  and  $B$  also need to be stored for each FMD.

Finally, the number of iterations  $k_{final}$  is finite but depends on the description of the object in the feature space. In the best case scenario  $k_{final} = M$ , only one FMD detection per feature is enough to obtain the convergence point. In the worst case  $k_{final}$  would be under  $M \times H \times W/2$  -at least two FMDs must change their boundaries and reduce the number of validated pixels when going through all  $M$  FMDs-, but such case is not realistic when looking for an object described by the chosen features and with a threshold proportional to the maximum value of the histogram. In most cases, multiple runs over all the FMD are necessary to separate objects sharing similar feature values.

Function	Computation cost
Histogram computation	$\mathcal{O}(\text{image size})$
Boundaries update	$\mathcal{O}(\text{histograms bins})$
DA complete process	$\mathcal{O}(k_{final} * (\text{image size} + \text{histogram bins}))$

TABLE 3.2 – DA computation cost per function

### 3.3.3 DA process illustration

In order to illustrate the behavior of one Dynamic Attractor, we tested it on synthetic images with different configurations (type of features, update modes and threshold rules). Three aspects of the DA process are illustrated here. First the DAs iterative steps, then the influence of the update mode and finally, the influence of the histogram threshold. The experiments have been performed with our BIPS simulator, presented in Chapter 4. A specified script has been made to run the following examples using the classes and tools we developed and the display was made with Matlab (*MATLAB and Image Processing Toolbox Release R2018a*). The parameters used in those examples have been chosen manually.

#### Experimentation 1: DA steps

The first test is done on the synthetic image of a green disk. The DA works on the hue feature, with a chosen threshold  $\tau_{sat} = 15$ , and on the spatial coordinates  $X$  and  $Y$ . The histogram thresholds are set to  $\tau_m^k = q_{max}^{m,k} / 2$ . Figure 3.15 shows the different steps of the DA process from  $k = 0$  to  $k = 3$ , its convergence state. We can observe the evolution of the valid pixels, the framing and the conditioned feature histograms.

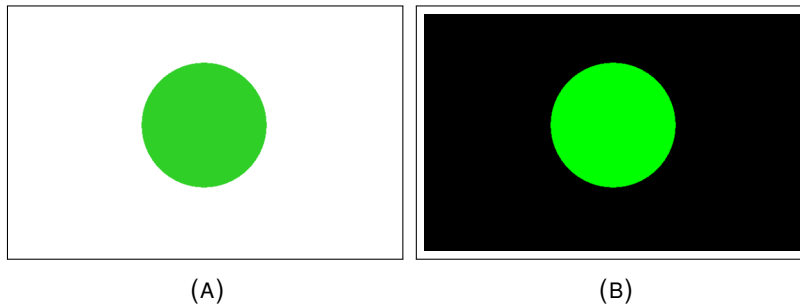


FIGURE 3.15 – (A) Green spot image. (B) Hue extraction.

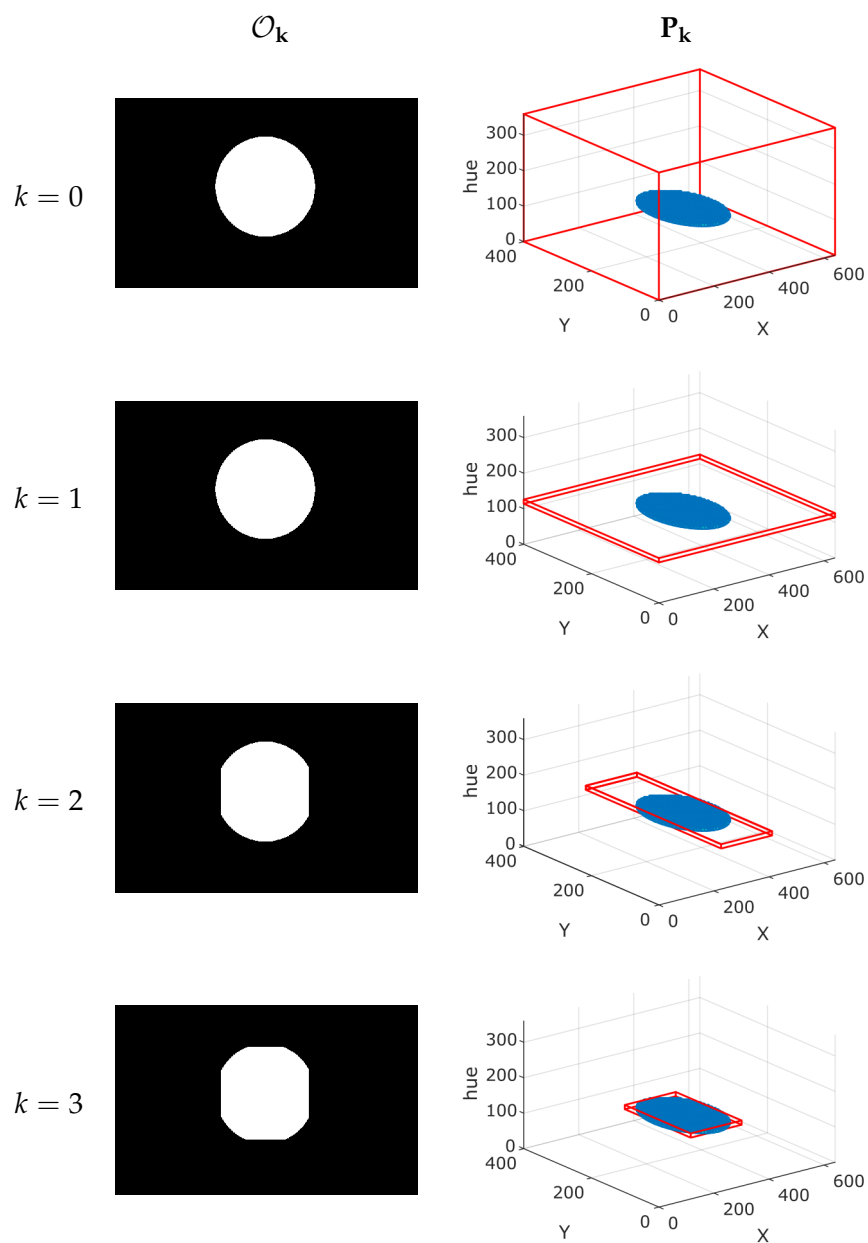


FIGURE 3.15 – Validated pixels set  $\mathcal{O}_k$  and framing  $P_k$  evolution along the iteration steps  $k$ .

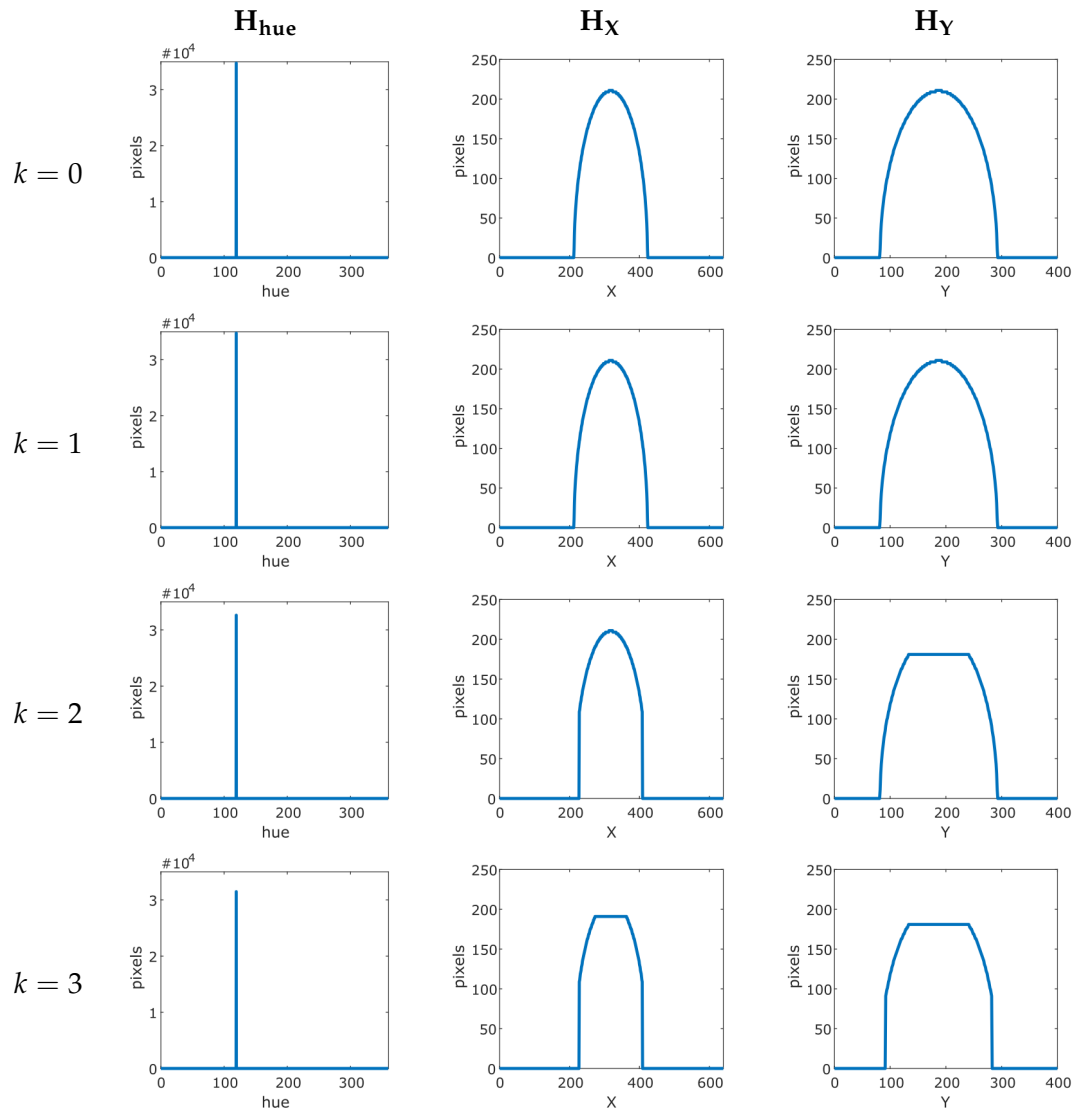


FIGURE 3.15 – Feature histograms evolution along the iteration steps  $k$ .

### Illustration 2: the two modes

The second test is done on the synthetic image of dashed lanes. The DA works on the gradient angle extracted with a derivative Gaussian kernel chosen of size 9, of standard value  $\sigma = 1.5$  and with a chosen threshold  $\tau_{mo} = 4\%$  of the possible maximum; and on the coordinates  $X$  and  $Y$ . The histogram threshold is still set to  $\tau_m^k = q_{\max}^{m,k}/2$ . Figure 3.16 shows the results obtained when using Mode 1 or Mode 2 on the  $X$  axis. The former selects only one of the dashes, whereas the latter selects the complete line.



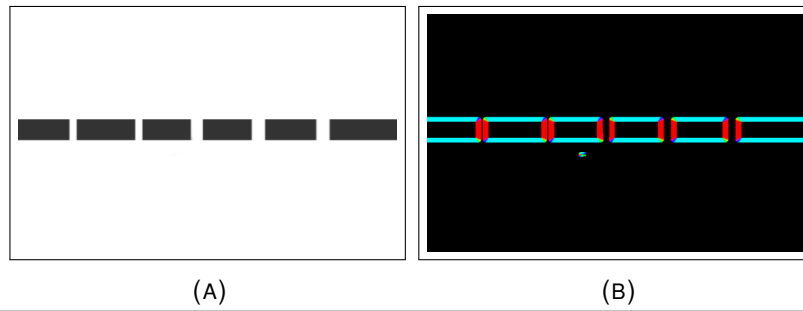


FIGURE 3.16 – (A) Dashed lanes image. (B) Gradient angle extraction.

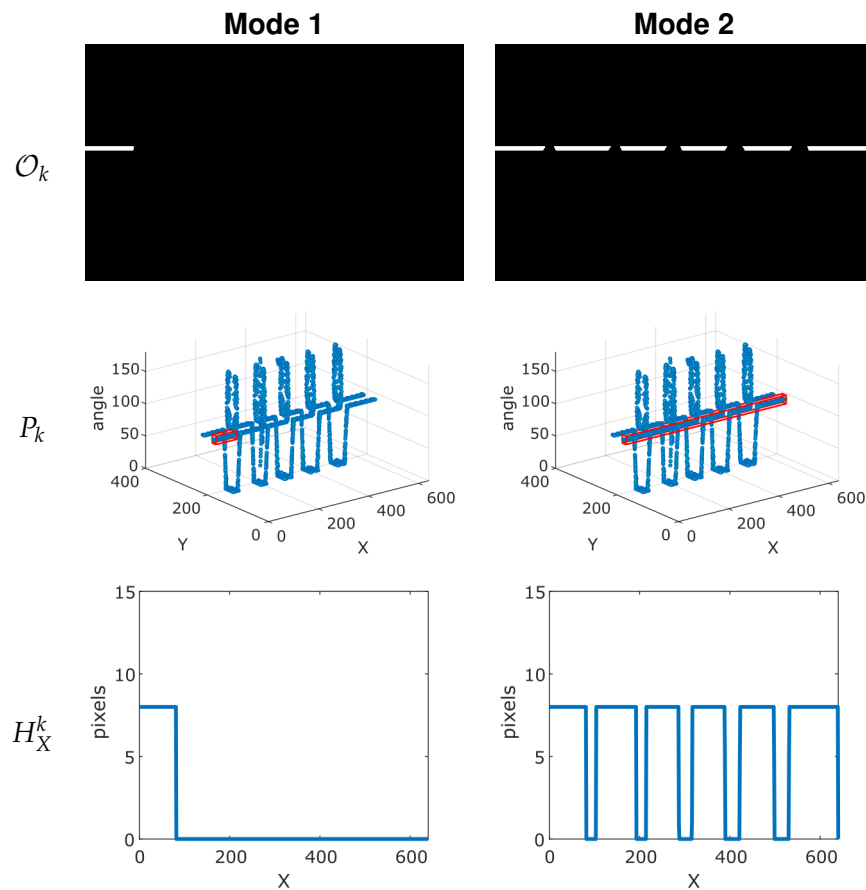


FIGURE 3.16 – Results obtained with Mode 1 or Mode 2 on the feature  $X$ .

### Illustration 3: the histogram threshold

The third test is done on the synthetic image of green sun draw. The DA works on the hue, with a chosen threshold  $\tau_{sat} = 15$ , and on the coordinates  $X$  and  $Y$ . Figure 3.17 shows the results obtained with different thresholds  $\tau_m^k$ . This puts forward one of the complexity of the Cartesian framing: if the threshold is too high, part of the object might be cut and excluded, if it is too low it might include too much noise.

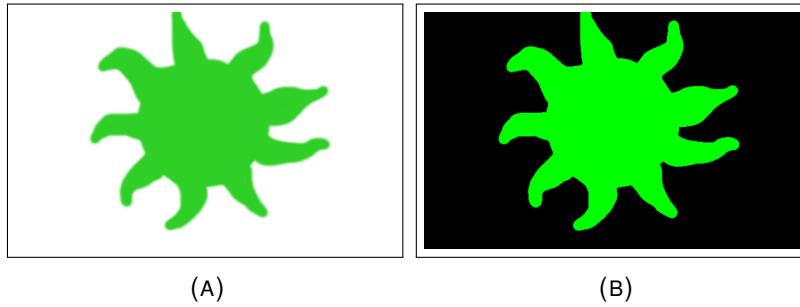


FIGURE 3.17 – (A) Green sun draw image. (B) Hue extraction.

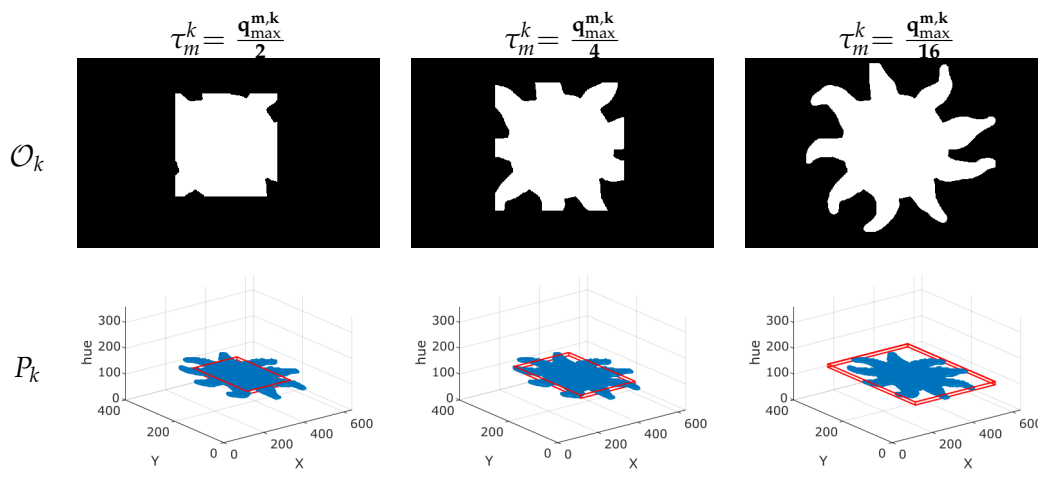


FIGURE 3.18 – Results obtained on a green sun draw with different histogram thresholds:  $1/2$ ,  $1/4$  and  $1/16$  of the histogram maximum.

### 3.3.4 Conclusion

The formalization of the Dynamic Attractor (DA) shows that this detection method is based on an iterative process of histogram computation and their peak selection. It is proved in Appendix A that the method, used on static images, converges in limited time to a high density set of pixels in the defined feature space. The combination of descriptive and spatial features, following the "What" and "Where" biologic hypothesis (Sec. 2.3.4), allows to obtain the simultaneous description and localization of elements of interest.

The real-time and low power property of the BIPS seems to be confirmed by the functions used in the FMDs, which have very low computational and memory costs. The sequential use of these FMDs might, however, hinder this property when a high number of steps are needed before converging to the object.

The first experiments confirm the detection of specific object by the DA and the real-time property on the chosen synthetic images, but also show the strong influence of several parameters, like the update selection mode or the histogram threshold.

### 3.4 Multiple detection

This Section completes the formalization and analysis of the second part of the BIPS: the Description and Localization part (Fig. 3.3), which is the connected composition of multiple DAs for multiple and/or more specific objects. The connections are presented in (Pirim, 2015) as association or inhibition of DAs. Based on our formalization and the observation of the industrial chip results, we established that the connections are logical associations or inhibitions of FMDs pixel validation outputs, which provide the pixel validation inputs  $val_{ext}$  of the DAs.

The different connections and their limitations are presented in Section 3.4.1. Then the behavior of the method is illustrated on static images in Section 3.4.2. This bring forward several complexities in the use of the BIPS method such as the influence of its architecture and its parameter settings on the detected objects and thus on its real-time, generic and efficient properties. The list of these parameters is summarized in Section 3.4.3, before concluding in Section 3.4.4.

#### 3.4.1 Connections of DAs

##### Architecture

The Description and Localization part corresponds to different sets of connected DAs. To formalize it and understand the influence of those connections we focus on the  $q^{th}$  DA of one set  $(DA_1, \dots, DA_Q)$ . The  $M_q$  features input of the  $q^{th}$  DA is a subset of the extracted features  $(F_1, \dots, F_M)$ . To differentiate them, we introduce the application  $s_q$  which associates the DA input feature indices to their indices in the global set of features as in Eq. 3.34. The validation input, in another hand, is the association of the feature masks and a combination of the validation outputs from FMDs of the other DAs. This combination corresponds to associated and inhibited connections of the FMDs validation outputs.

**Definition.** Feature set input of the  $q^{th}$  DA

$$F_q = \left( F_{s_q(m)} \right)_{m \in \llbracket 1, M_{n_{DA}} \rrbracket} \quad (3.34)$$

##### Association

The association consists in the restriction of the  $q^{th}$  DA on a part of the feature space already selected by another DA, or only by a set of FMDs. For example, a DA that detected a tennis ground using the hue feature can restrict another DA with only the spatial pixel validation  $(A_X \leq X \leq B_X) \cap (A_Y \leq Y \leq B_Y)$  to let a second DA find a lane within this ground.

In the case of association, the boundaries updates of the connected DAs must not be done in parallel with the risk of two of them converging to separate objects, which would lead to an empty set of pixels. Thus, such connection must be done on DAs working sequentially.

## Inhibition

The inhibition consists in the suppression for the  $q^{\text{th}}$  DA of a part of the feature space selected by another DA, or a set of FMDs. For example, to detect multiple objects of the same nature, like tennis balls on a training field, one needs to use identical DAs and inhibit the spatial framing the other DAs converged to.

In the case of inhibition, the boundaries update can be done in parallel, but an order of priority of the DAs must be imposed. This order must be updated in case one of the prioritized DA is reinitialized  $P_{q,0} = \Omega$ , to avoid it to empty the following DAs. The boundaries update can also be done sequentially. In this case, if one of the DAs converge to an object smaller than the chosen threshold  $\tau_N$ , it must be reinitialized with a different  $val_{ext}$ , otherwise it will loop indefinitely to the same element.

### 3.4.2 Observations

In order to illustrate the behavior of the method, we test it on several static images with different configurations: first to show the influence of the chosen configuration on the detection of multiple objects; then to describe the effects of transformation on an image like rotation, noise addition or change of resolution; finally, to highlight the influence of the Cartesian framing and thus the importance of the chosen features and the DAs combinations.

The different experiments have been done with our BIPS simulator presented in Chapter 4. A specified script has been developed to run the following examples using the **Feature**, **FeatModeDetector**, **DynamicAttractor** classes and the display was made with Matlab (*MATLAB and Image Processing Toolbox Release R2018a*). The parameters have been chosen based on our knowledge of the input image and the behavior to illustrate.

#### Illustration 1: Multiple detection

The first experiment is done on the colored chips image of Matlab. We use 30 DAs working on the hue feature with a chosen threshold  $\tau_{sat} = 30$ , and the two Cartesian coordinates  $X$  and  $Y$  (Figure 3.19). The histogram thresholds are set to  $\tau_m^k = 0.30 \times q_{\max}^{m,k}$  and Mode 1 is used for all FMDs. The object size threshold is set to  $\tau_N = 0$ , allowing all objects even pixel size objects to be detected.

The DAs are linked to each other by inhibition allowing the detection of multiple objects. The results presented in Figure 3.20 are obtained with (A) a sequential and (B) a parallel running of the DAs. We observe that the detected objects are mostly the same, even when it corresponds to just a border of a colored chip, but the order of detection differs. The small framing detection -22, 23, 24, 26 in (A) or 4, 13, 22, 24, 26, 29 in (B)- are due to the null size object threshold. Some detected objects are cut -11, 17 in (A) or 14, 25 in (B)-, this might result from two things: firstly the threshold on the features is too high, secondly the histogram maximum regroupes aligned objects leading to a higher threshold. In order to correct such problems, it would be interesting to find an automatic setting of the feature threshold. An extension of the boundaries during each step of the iterative process could also be used, but the convergence property of the DAs might no more be ensured.

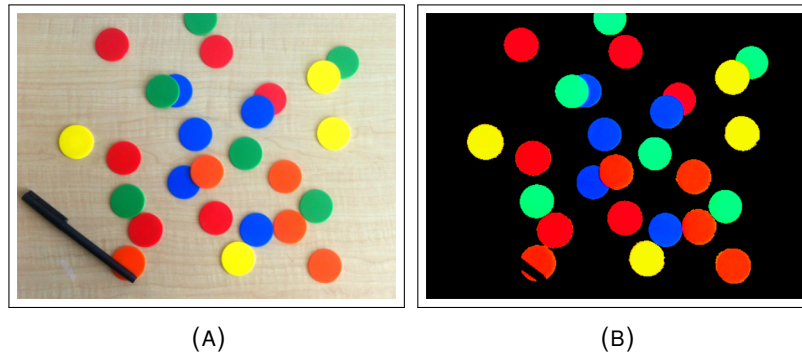


FIGURE 3.19 – (A) Colored chips image (*MATLAB and Image Processing Toolbox Release R2018a*). (B) Hue extraction.

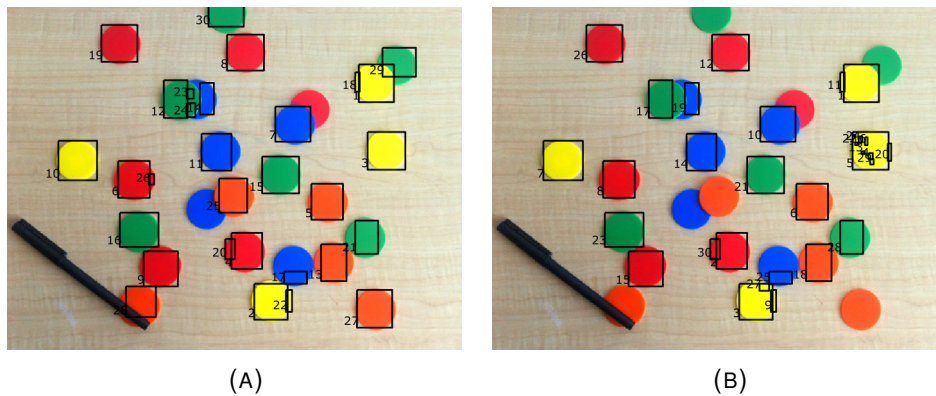
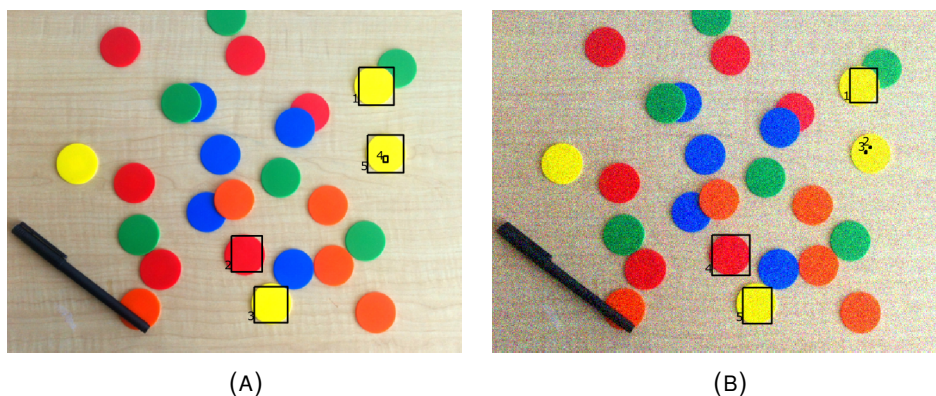


FIGURE 3.20 – (A) Results with 30 DAs running sequentially. (B) Results with 30 DAs running in parallel.

### Illustration 2: Image transformation impact

The second experiment is also done on the colored chips image of Matlab. We observe here, using only 5 DAs working in parallel, the impact of image transformation to the BIPS output. The DAs parameters are the same:  $\tau_{sat} = 30$ ,  $\tau_m^k = 0.30 \times q_{max}^{m,k}$  and  $\tau_N = 0$ . The experiment has been conducted on the original image (A), with white noise added to the image using the Matlab *imnoise* function (B), after rotating the image of  $90^\circ$  (C) and on a reduced resolution at 1/5 of the image using the Matlab *imresize* function (D). The results are presented in Figure 3.21. The convergence to the colored chips is maintained during those transformations. However, the order of detection differs.



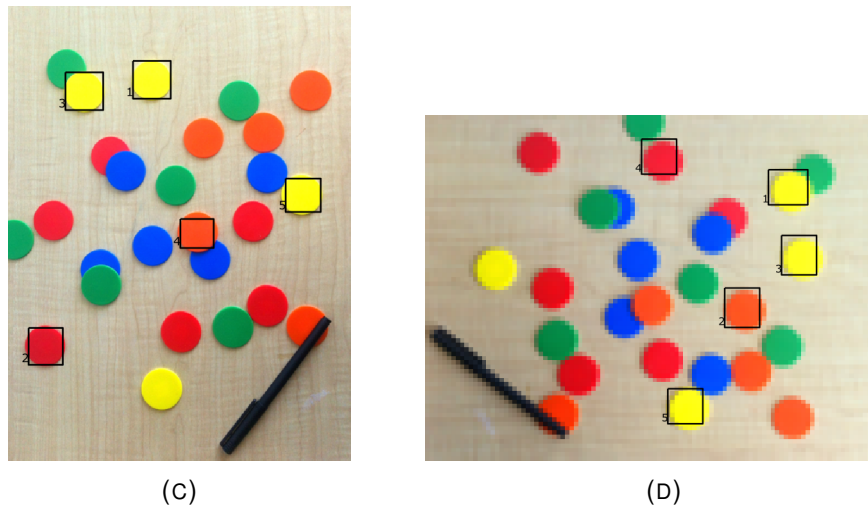


FIGURE 3.21 – (A) Results on the original image. (B) Results on the noisy image. (C) Results on the rotated image. (D) Results on the low resolution image.

### Illustration 3: Detection of transversal objects

The third experiment is done on a road image (*Road*). We run 5 DAs working on the gradient angle feature, with a chosen threshold  $\tau_{mo} = 887$ , and the two Cartesian coordinates  $X$  and  $Y$  (Figure 3.22). The histogram thresholds are set to  $\tau_m^k = 0.30 \times q_{\max}$  and Mode 1 is used for all FMDs. The object threshold is set to  $\tau_N = 0$ , allowing to detect all objects, even single pixel sized.

In Figure 3.23, the results obtained by the second and third DAs are shown. They spotlight the limitation of the Cartesian coordinates features: for the second DA, the object is aligned with  $X$  so the spatial framing is well suited, but for the third DAs, the object is not aligned with neither  $X$  or  $Y$ , which gives a large spatial framing. Even though the selected pixels correspond to the lanes, the associated bounding box does not fit well the sought object. Moreover, in case of a combination using inhibition on the spatial features, such framing will suppress too much pixels for the other DAs. For his chip, P. Pirim proposes a way to keep a close framing of object by computing several rotated coordinates (Pirim, 2005). This extension of the method will be discussed in Chapter 5 and a more generic extension will be proposed in Chapter 6.

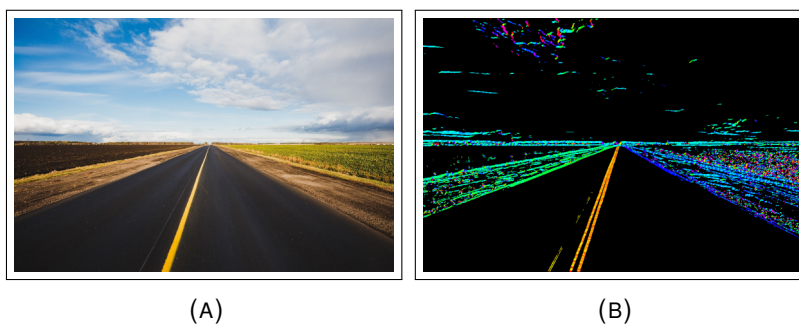


FIGURE 3.22 – (A) Road image (*Road*). (B) Angle extraction.

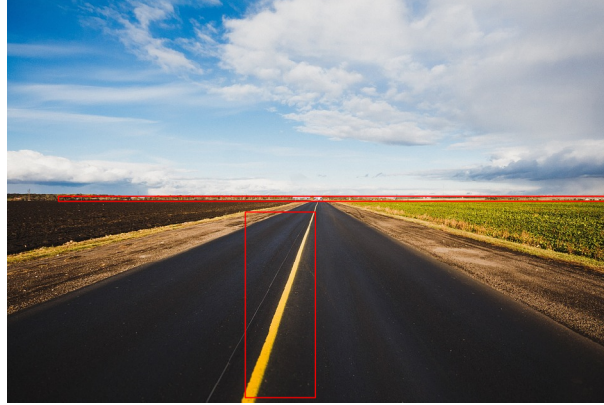


FIGURE 3.23 – Results showing the limits of the Cartesian coordinates feature for an object framing.

### 3.4.3 Parameters

The study of the BIPS concept brought forward several parameters that can have a strong impact on its results. Here I remind each of these parameters and present our analysis of their impact.

#### Features parameters

The input features  $(F_1, \dots, F_M)$  comes from extraction applications whose results depend on the input image and the application parameters -for example the threshold used for the hue or the gradient angle computation-. The objects representation in the feature space is then influenced by the input features selection and their parameters.

#### Architecture

The chosen sequences of DAs  $(DA_1, \dots, DA_Q)$ , their chosen input features  $(F_{s_q(m)})_{m \in \llbracket 1, M_q \rrbracket}$  and their connections can give different outputs when they are changed. The architecture of the BIPS might be application-specific. Our current knowledge of object representations in computer vision and our formalization of the method does not allow us to establish a new BIPS architecture which could adapt to different applications.

#### Object size threshold

The object size threshold  $\tau_N$  determines whether or not a group of pixels selected by a DA corresponds to a relevant object. If its value is too high, some objects of the scene might not be detected, whereas if it is too small too much DAs might converge to noise in the image. The choice of its value can be established based on the application and the object representation by the chosen features, however for a more generic use of the BIPS it would be interesting to find an automatic setting of this value.

### Histogram threshold

The histogram threshold  $\tau_m^k$  defines the selection of the histogram peak(s). If it is too high, it might cut parts of the detected object, if it is too small it might not separate different objects close to each other in the histogram representation or even integrate noise to the selection. This parameter is often chosen as a percentage of the maximum histogram value  $q_{\max}$ . However, depending on the application and the step of the iterative process the adequate percentage may differ.

### Mode

The Mode of selection 1 or 2 may be taken as part of the Architecture section. If the Mode 1 is chosen, only the principal peak is selected and any obstruction, diminution of values in the histogram may result in the detection of only a part of an object. If the Mode 2 is chosen, all the peaks of the histogram are selected and two different objects may be detected as one. The choice of this Mode is application dependent.

#### 3.4.4 Conclusion

The formalization of the Description and Localization part shows how the BIPS method allows to detect multiple objects in a scene. The combination of DAs through pixel validation association or inhibition brings an extension of its detection property, but also new complexity issues to the method.

Through the associations, the method can sequentially perform hierarchical detection and through the inhibitions, the method can perform multiple detections. But, the architecture is not easily adapted for generic applications. The choice of features, the sets of DAs, their combinations, their object size thresholds, their FMDs Modes and histogram thresholds, all have an impact on the detection result. Moreover, the computation time can be strongly increased as the association can not be run in parallel and a bad choice in the parameters can lead to longer time of convergence for the DAs.

The first illustrations conducted on static images with application dedicated architecture show constancy of the detection method at the exception of the order of object detection.

## 3.5 Discussion

In this Chapter we have presented our formalization of the Bio-Inspired Perception Sensor (BIPS) concept, whose on-chip implementation showed interesting real-time and low power performances on several computer vision applications, but whose mathematical model was missing. The method can be decomposed into two main parts: a Feature extractions part and a Description and Localization part.

The first one currently corresponds to ten implemented local and visual feature extractions with real-time and low consumption properties. It could perform any kind of perceptive feature extractions as long as their domain can be shared with other features and their codomain can be represented by a finite and ordered set, to ensure the functioning of the BIPS method. The latter performs a simultaneous description and localization of objects in a scene based on the combination of several original modules called Dynamic Attractors (DAs), which are inspired from the "What" and "Where" visual cortex processes. It looks for combined neuronal responses on



descriptive features and spatial features through the use of joined histograms projections.

Our formalization of the method permits henceforth to reproduce and evaluate the BIPS behavior and confirms its convergence to multiple objects of interest, but whose characteristics might be influenced by our choice of parameters. Their impact on the result is difficult to establish in a generic way mainly for two reasons: firstly, the relation between a real object and its generic feature representation is still an open computer vision problematic; secondly, in an application-specific analysis, the number of BIPS parameters to set brings a high level of complexity in the analysis. The realization of experiments on academic datasets will permit us to visualize the method behavior, to evaluate its detection performances and to propose different automatic setting methods of the parameters.

## Chapter 4

# The BIPS simulator

As mentioned in the Section 3.1 no mathematical formalization and no simulator of the method exist, but the interesting results obtained by the on-chip implementation motivates us into filling those gaps. On the basis of my mathematical formalization, I have been able to develop tools and classes simulating the method in the image analysis case. This simulator allows to modify several parameters, to adapt and run the method on different academic applications and to easily test foreseen extended versions of the method.

The simulator has been developed using the C++ language with (*Qt 5.7 2016*) and (*OpenCV 2.4 release*). The implementation is object oriented. A user interface has also been developed using the Qt frameworks allowing to visualize the outputs. However, for the analysis of the BIPS behavior, the dataset results obtained through the simulator have been evaluated with Matlab (*MATLAB and Image Processing Toolbox Release R2018a*).

This Chapter is structured as follows. Section 4.1 presents the **Feature** class developed to keep a unique format input for the FMD while being able to change the type of feature used. The feature images are obtained using extractors reproduced from the BIPS on-chip, their implementation is detailed in the same Section. Then the **FeatModeDetector** class representing the FMD attributes and internal processes is presented in Section 4.2. The DAs are represented by the class **DynamicAttractor** managing several FMDs and input Features. Their attributes and functions, but also their adaptations for different BIPS extensions are presented in Section 4.3. Finally, Section 4.4 introduces the complete process management and the user interface developed in order to observe the BIPS blocks behavior.

This simulator has been upgraded along my thesis work. Some of the functions implemented and presented in this Chapter play a role which will be introduced in the following chapters.

## 4.1 Feature

### 4.1.1 The Feature class

The **Feature** class has been created to uniformize the outputs of each feature extractor, which are used as input for different FMDs. The class represents the outputs of the application  $F$ , defined in Section 3.2.4, as well as the necessary parameters for its use in the Dynamic Attractor (DA) process.

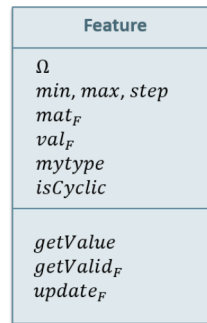


FIGURE 4.1 – The **Feature** class.

The class is composed of seven attributes:

- the set of possible feature values (i.e. the domain, denoted  $\Omega$ ), which is used to build the feature histogram
- the  $min, max$  and  $step$  values representing  $\Omega$
- the feature extraction output values ( $mat_F$ ), given by the extractor
- the associated validation mask ( $val_F$ ), also given by the extractor
- the attribute  $mytype$ , which differentiate the descriptive features from the spatial features, as the latter do not have a corresponding extractor
- the attribute  $isCyclic$ , which indicates if the features values are in a cyclic set

The class has also three main functions:

- $getValue$ , which obtains the feature value corresponding to the pixel  $p = (i, j)$ . It is  $mat_F(i, j)$  for a descriptive feature,  $j$  for the  $X$  feature and  $i$  for the  $Y$  feature
- $getValid_F$ , which gives the validation output  $val_F(i, j)$  associated to the feature
- $update_F$ , which actualizes the feature values and the associated validation mask

Each Feature is initialized at the beginning of the process and corresponds to one video flow and one feature extraction. Only the parameters  $mat_F$  and  $val_F$  can be updated at each new video frame.

### 4.1.2 Simulator implemented features

The feature values and the associated mask ( $mat_F, val_F$ ) are obtained through feature extraction that must be implemented and called at each update. Several of them have already been developed for the BIPS and reproduced in our simulator. Diagram representations of the different groups of features (Tonal, Dynamic and Structural) are given below. The corresponding equations were presented in Section 3.2.

To reproduce at best the chip process, the  $BGR$  images obtained with OpenCV from the camera stream or the input files are first transformed into  $YC_bC_r$  images following the BT601 digital norm (BT.601) (Fig. 4.2).

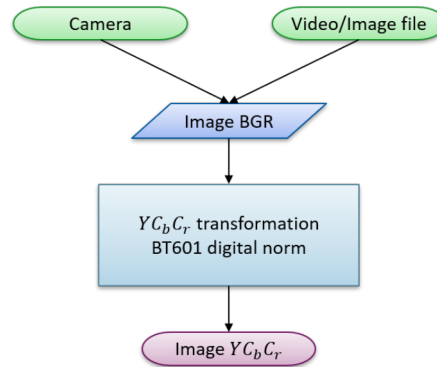


FIGURE 4.2 – Transformation of the input into  $Y, C_b, C_r$  image.

### Tonal features

The Tonal features extraction, presented in Section 3.2.1, works directly on the  $YC_bC_r$  image and produces the luminance  $L$ , the saturation  $S$ , the hue  $H$  and its associated mask  $val_{hue}$ . Their extraction process is summarized in Figure 4.3.

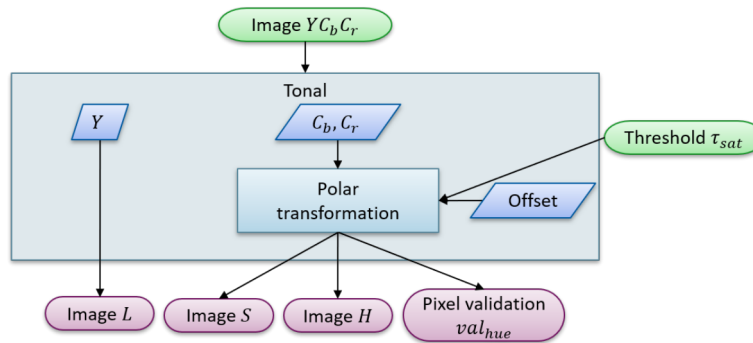


FIGURE 4.3 – Tonal features extraction diagram.

### Dynamic features

The Dynamic features extraction, presented in Section 3.2.2, works on the extracted luminance  $L$ , the memorized values of the background estimation  $\hat{L}$  and the temporal variation  $D$ , which are computed along with the associated mask  $val_D$ . Using the computed temporal variation, the application also extracts the module  $V$  and direction  $\theta$  of the peculiar optical flow developed by P. Pirim. However, the optical flow used in the time integration proposal of Chapter 6 is based on the *calcOpticalFlowFarneback* function of (*OpenCV 2.4 release*). Their extraction process is summarized in Figure 4.4.

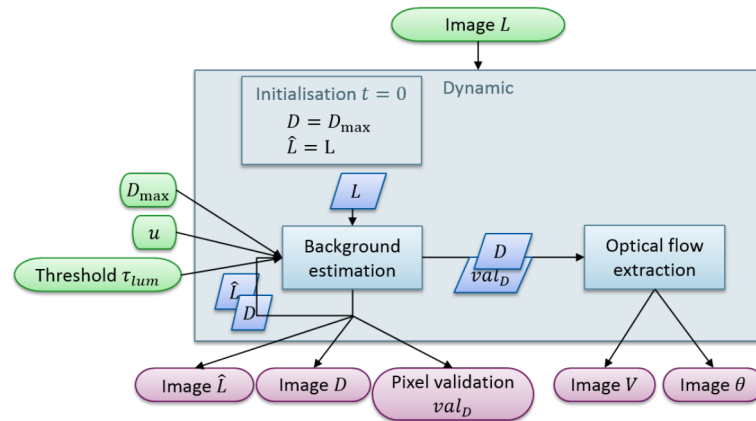


FIGURE 4.4 – Dynamic features extraction diagram.

### Structural features

The Structural features extraction, presented in Section 3.2.3, also works on the extracted luminance  $L$  and the predefined filter kernel  $W$ . They produce the module  $Mo$  and angle  $An$  of the gradient along with the associated mask  $val_{ang}$ . Using the extracted angle, its variation  $\Delta An$  and associated mask  $val_{\Delta An}$  are also produced. Their extraction process is summarized in Figure 4.5. The diagram Figure 4.6 gives more details on the gradient extraction, which was used for our experiments in Chapter 5 and 6.

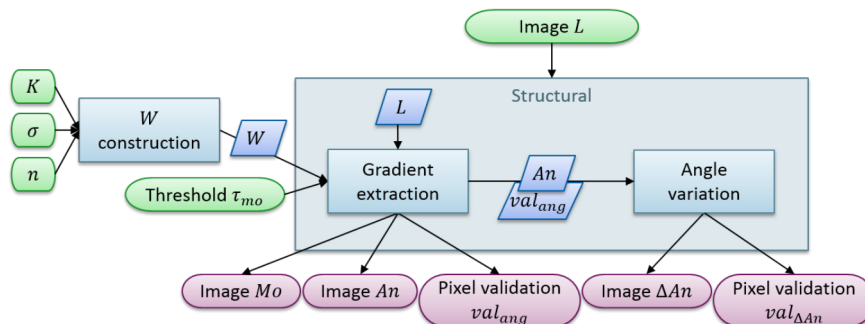


FIGURE 4.5 – Structural features extraction diagram.

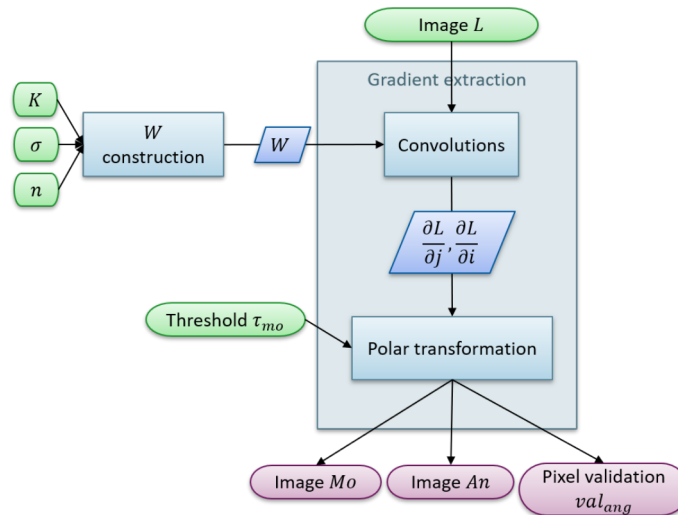


FIGURE 4.6 – Gradient features extraction diagram.

## 4.2 Feature Mode Detector

### 4.2.1 The `FeatModeDetector` class

The `FeatModeDetector` class allows to regroup the variables and functions managed by one FMD and described in Section 3.3.1.

The class is composed of seven main attributes:

- the map structure representing the histogram  $(\Omega, H)$
- the histogram maximum  $q_{\max}$
- the histogram mode  $f_{\max}$
- the boundaries  $A$  and  $B$
- the threshold  $\tau$
- the chosen update Mode

Six supplementary attributes are given to the class corresponding to several proposed extensions:

- the angle  $\alpha_{FMD}$ , which is memorized in the spatial axes rotation case in Chapter 5
- the vector  $u_{FMD}$ , which is memorized in the feature space reorientation case in Chapter 6
- the imposed minimum and maximum limits  $A_{\min}, B_{\max}$ , which can be used for the initialization of the FMD
- the cyclic value  $\delta$ , which corresponds to the cyclic order of the feature. It is null when the corresponding feature is not cyclic
- the percentage  $\tau_{per}$  used to compute the histogram threshold, which is negative when the chosen threshold is a fixed value

FeatModeDetector		
$H$	$A$	$\alpha_{FMD}, \mathbf{u}_{FMD}$
$f_{max}$	$B$	$A_{min}, B_{max}$
$q_{max}$	$\tau$	$\delta, \tau_{per}$
	$Mode$	
$getValid_{FMD}$ $addToMap$ $updateBoundaries$ $computeThreshold$ $cleanMap$ $reinit_{FMD}$ $computeMeanNMedian$ $computeEpsilon$		

FIGURE 4.7 – Representation of the **FeatModeDetector** class.

Each FMD is initialized by a DA at the beginning of the process. The map keys are fixed and correspond to the set  $\Omega$  representing the feature values, the map values are set to 0, also is  $q_{max}$ , the boundaries are set to the minimum and the maximum keys and  $f_{max}$  is set to the minimum key, as it has been defined as the minimum in Eq. 3.18. The map keys stay unchanged, but the other parameters are updated along the process.

*Remark.* In the case of the alternative equations (See App. A.1.3), the boundaries are set to the minimum minus one and the maximum plus one.

The FMD has three principal methods:

- $getValid_{FMD}$  validates or not the given key using the pixel validation  $val_{FMD}$  (Eq. 3.21)
- $addToMap$  constructs the histogram with the validated pixels
- $updateBoundaries$  updates  $A$  and  $B$  for their next use

The validation and the histogram computation are processed along the pixel flow, but the update parts is performed only when the histogram computation is complete.

The FMD has also three important methods:

- $computeThreshold$  allows to adapt the threshold to the obtained histogram maximum
- $cleanMap$  empties the map before the histogram computation
- $reinitFMD$  brings back the FMD to its initial values when the DA itself is being reinitialized

Finally two supplementary functions can be used for our tests:  $computeMeanNMedian$  and  $computeEpsilon$ . Those functions are explained in the following Sections.

### 4.2.2 FMD pixel validation

The FMD pixel validation is computed using the boundaries  $A$  and  $B$  and the given feature value. For a better framing of the object, a parameter  $\varepsilon$  obtained with the function *computeEpsilon* can be used to enlarge the set of validated pixels.

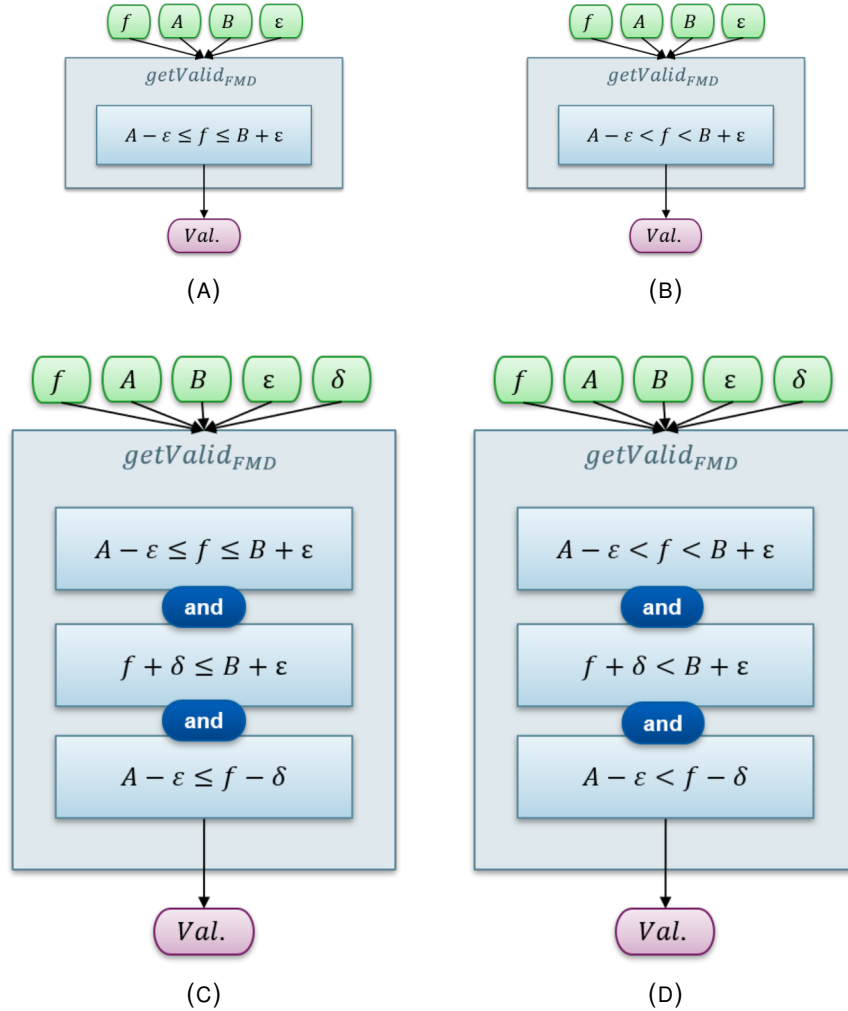


FIGURE 4.8 – Pixel validation (A) (resp. C) and its alternate version (B) (resp. D) in the classic case (resp. in the cyclic case without any rotation).

The specific case of cyclic features implies to do this test on several features values. In the classic version of the method, the test is performed on three feature values:  $f$ ,  $f - \delta$  and  $f + \delta$  where  $\delta$  is the order of the cyclic group. The validation is obtained if the test is positive for at least one of the three values. I don't treat the cyclic case in the multidimensional rotation extension as the number of tests to perform quickly gets high and too complicated to be represented.



### 4.2.3 Histogram construction

When the pixel is validated, the feature value is added to the histogram map. The histogram construction uses the given feature value to increment the map values  $H$  and updates  $f_{\max}$  and  $q_{\max}$ . The cyclic case in the classic version implies to add two other feature values  $f + \delta$  and  $f - \delta$  to the histogram as proposed in Section 3.2.4.

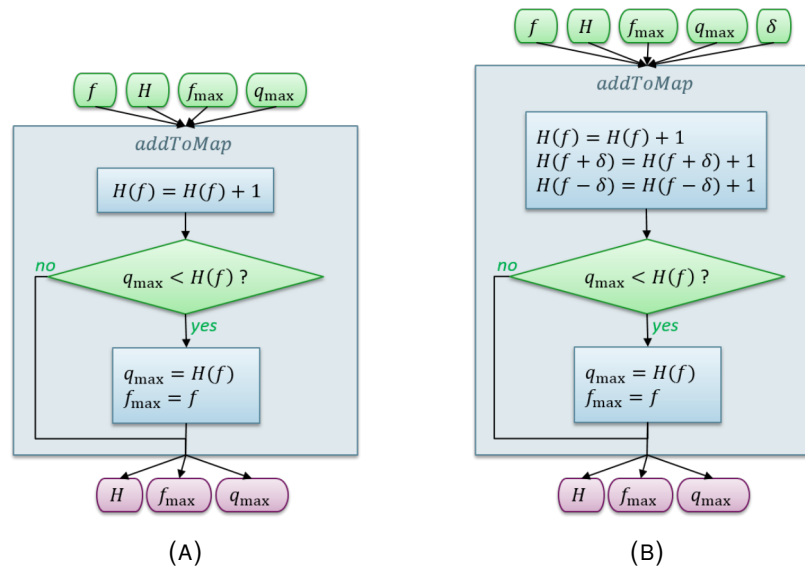


FIGURE 4.9 – Histogram computation diagram (A) and its version in the cyclic case without any rotation (B).

### 4.2.4 Update boundaries

The boundaries update is based on the chosen *Mode*, the obtained histogram map  $(\Omega, H)$ , the feature mode  $f_{\max}$ , and the given threshold  $\tau$ , which must be under the maximum  $q_{\max}$ . The threshold can be chosen fixed or a percentage of the histogram maximum. To do so two attributes can be added isPercent and percent, such that if isPercent is true, then  $\tau = (q_{\max} * \text{percent})/100$ .

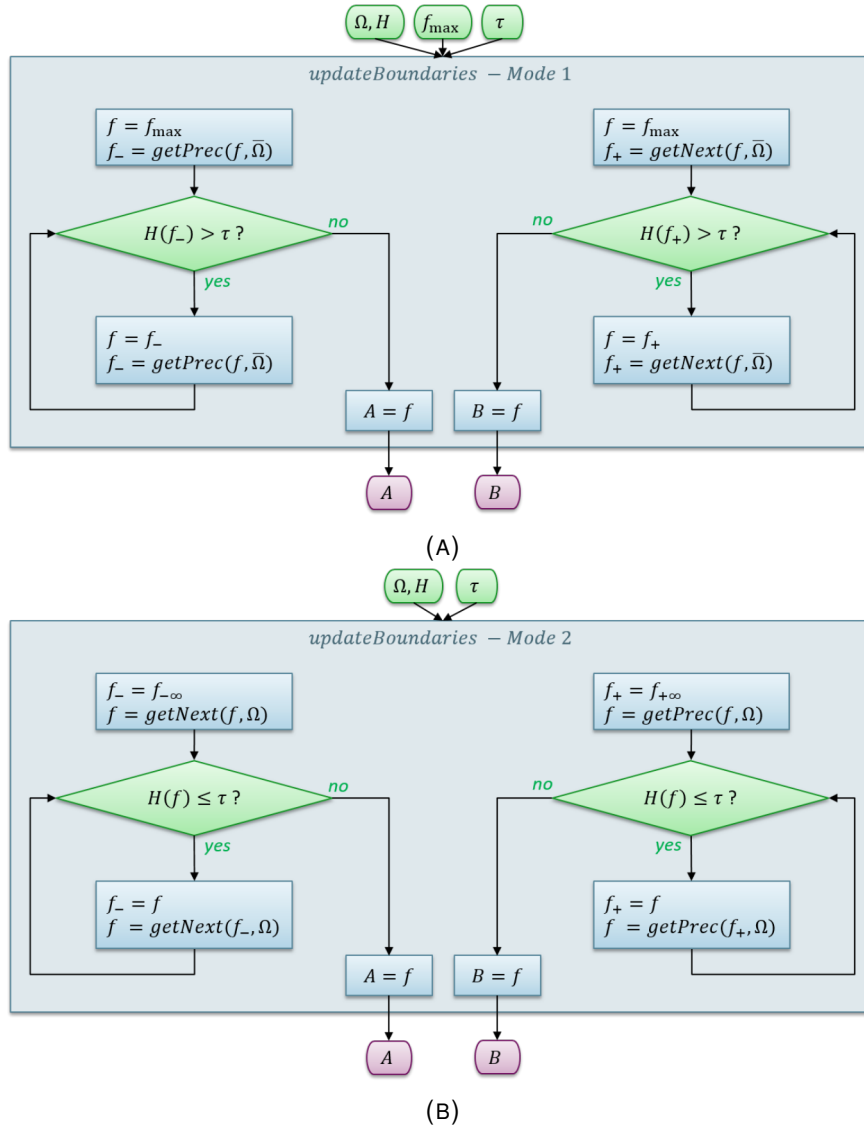


FIGURE 4.10 – Boundaries update Mode 1 (A) and Mode 2 (B). In the alternate version the outputs selected are  $f_-$  and  $f_+$ .

## 4.3 Dynamic Attractor

### 4.3.1 The DynamicAttractor class

DynamicAttractor	
$F_1, \dots, F_M$	$\alpha, \Delta_\alpha$
$FMD_1, \dots, FMD_M$	$\mathbf{u}_1, \dots, \mathbf{u}_M$
$k$	$M_c, \mu, \Sigma$
$N$	$K, \hat{\mu}$
$\tau_N$	
<i>getKey</i>	<i>hasConverged</i>
<i>ValidNHisto</i>	<i>prediction</i>
<i>update<sub>DA</sub></i>	<i>timeUpdate</i>
<i>reinit<sub>DA</sub></i>	
<i>clean<sub>DA</sub></i>	

FIGURE 4.11 – Representation of the DA class

The master class **DynamicAttractor** has been created to generate and manipulate more easily the associations of FMDs. This class is composed of five main attributes:

- a set of **Features**  $((F_m)_{m \in \llbracket 1, M \rrbracket})$
- a set of ordered **FeatModeDetectors**  $((FMD_m)_{m \in \llbracket 1, M \rrbracket})$
- an iteration step counter  $k$
- the number of validated pixels  $N$
- the threshold  $\tau_N$  on the detected object size

The extensions proposed in Chapter 5 and Chapter 6 imply the use of eight other attributes:

- the rotation angle  $\alpha$
- the corresponding geometric variation  $\Delta_\alpha$
- the FMDs axes  $(\mathbf{u}_m)_{m \in \llbracket 1, M \rrbracket}$
- the number of features to be rotated  $M_c$
- the corresponding mean  $\mu$
- the covariance matrix  $\Sigma$
- the medians  $\hat{\mu}$
- the Kalman filter  $K$

The DA performs its iterative process and manages the FMDs through eight methods:

- *getKey*, which computes the feature values
- *ValidNHisto*, which computes the validation of the pixel and constructs the histogram on the fly
- *update<sub>DA</sub>*, which updates the boundaries of the FMDs and the different parameters used for the rotation
- *hasConverged*, which validates the convergence of the DA
- *reinit<sub>DA</sub>*, which is used when the DA converged to something too small
- *clean<sub>DA</sub>*, which calls the *cleanMap* function of the FMDs
- *prediction* and *timeUpdate*, which performs the time integration for the tracking proposal in Chapter 6

### 4.3.2 Feature value

The Feature value for each pixel corresponds in the classic version of the method to the associated **Feature** *getValue*. But in the rotated version of Chapter 5, this value is dependent of the parameter  $\alpha$  and in the extended version of Chapter 6 it is dependent of the unit vector axis  $\mathbf{u}$  associated to the FMD or given by the DA. The **DynamicAttractor** class integrates the different ways of computing this value.

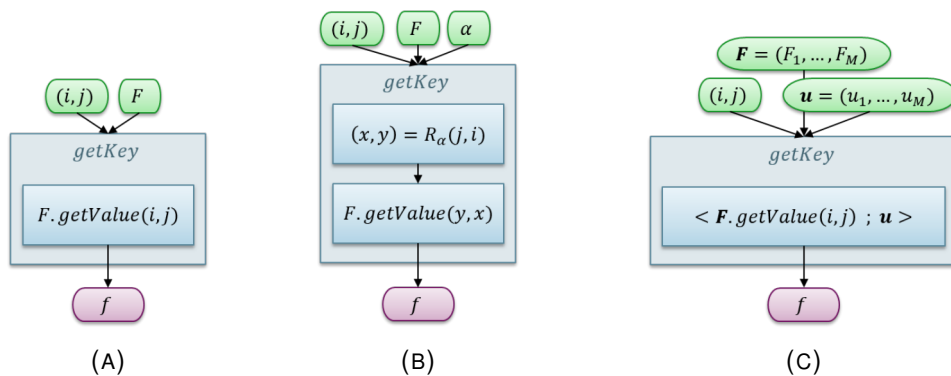
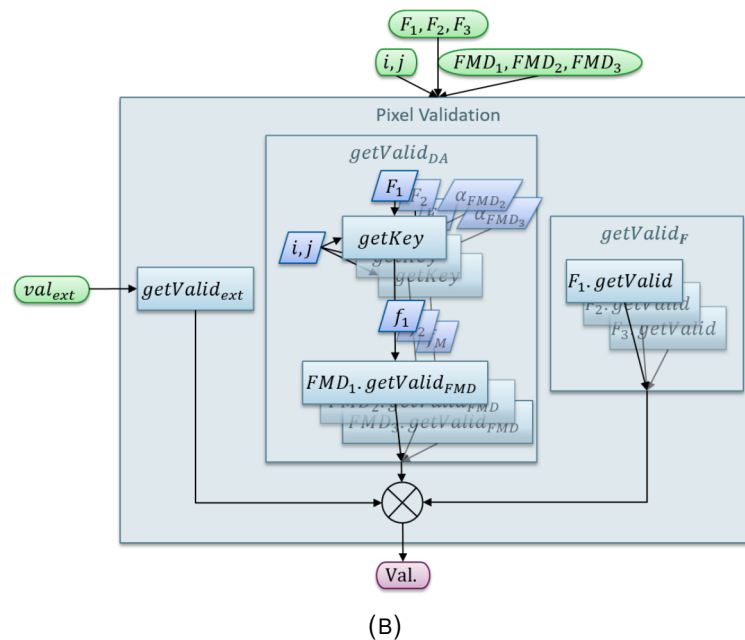
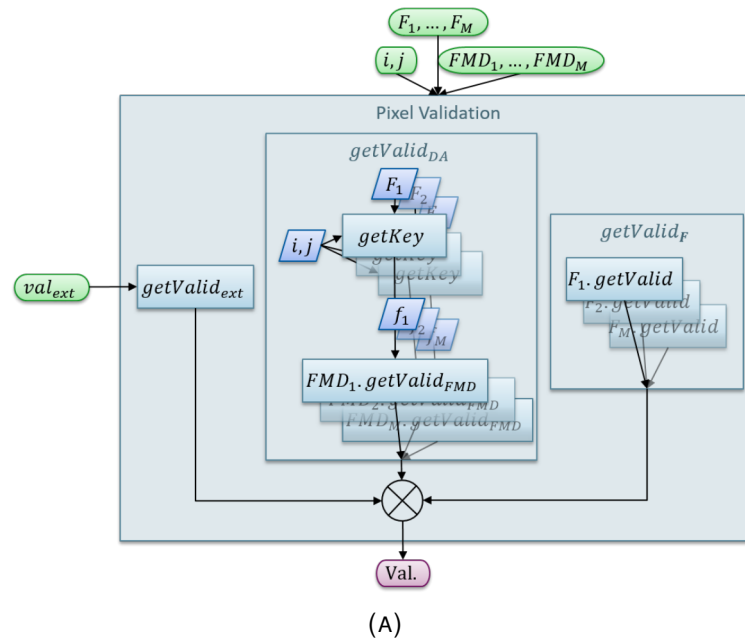


FIGURE 4.12 – Selection of the feature values. (A) In the classic version. (B) With the  $\alpha$  rotation of the spatial axes. (C) With the generic feature space rotation.

### 4.3.3 Pixel validation

The pixel validation is in fact the result of the association of three functions:  $getValid_F$ , which combines the **Features** validation maps;  $getValid_{ext}$ , which recovers the validation information coming from external DAs and FMDs; and  $getValid_{DA}$ , which combines the FMD pixel validations (Fig. 4.13).



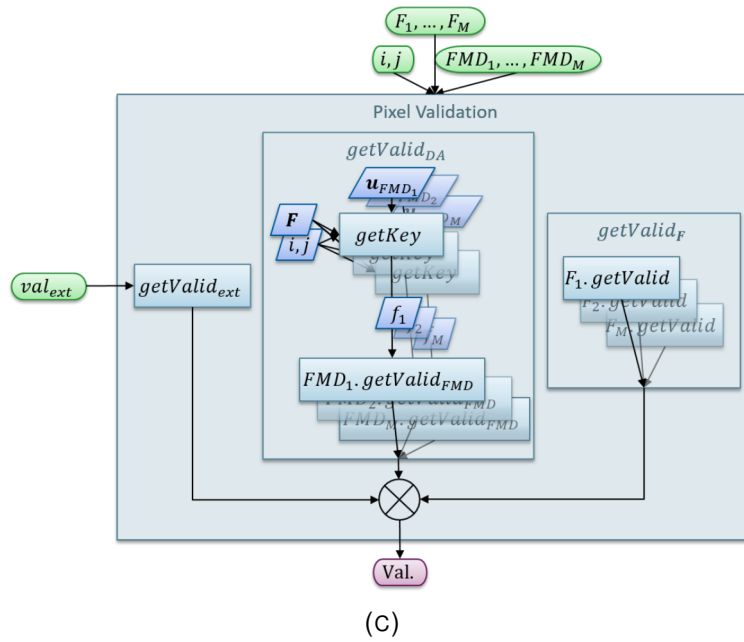


FIGURE 4.13 – Pixel validation. (A) In the classic version. (B) With the  $\alpha$  rotation of the spatial axes. (C) With the generic feature space rotation.

### 4.3.4 Histogram computation

The histogram computation consists in the use of the *addToMap* function in the concerned FMDs for the validated pixels. Depending on the version of the method the computation performed differs.

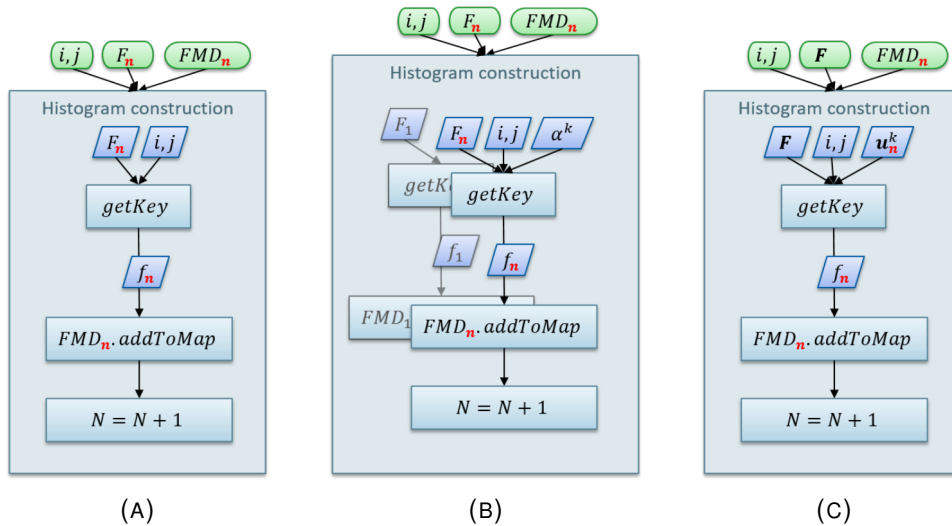


FIGURE 4.14 – Histogram computation. (A) In the classic version. (B) With the rotation of the spatial axes. (C) With the generic rotation.

### 4.3.5 Update

The update of the DA is a combination between the update of the FMDs boundaries and the rotation parameters update.

Firstly, the FMD boundaries update is decomposed into two steps:

1. the computation of the threshold  $\tau$
2. the FMD *updateBoundaries* function

Secondly, the rotation parameters update consists in two steps:

1. the FMD parameter memorization  $\alpha_{FMD}$  or  $\mathbf{u}_{FMD}$  is updated to the current parameter value in the DA
2. the DA parameter is updated for the next iteration

Depending on the rotation chosen, the parameters are updated differently: in the spatial rotation used in Chapter 5, the angle mode is used to update the rotation angle  $\alpha$ ; in the generic rotation presented in Chapter 6, the covariance matrix, computed on the fly, is used to find the principal axes of the detected object  $\mathbf{u}_{m_0}, \dots, \mathbf{u}_M$ .

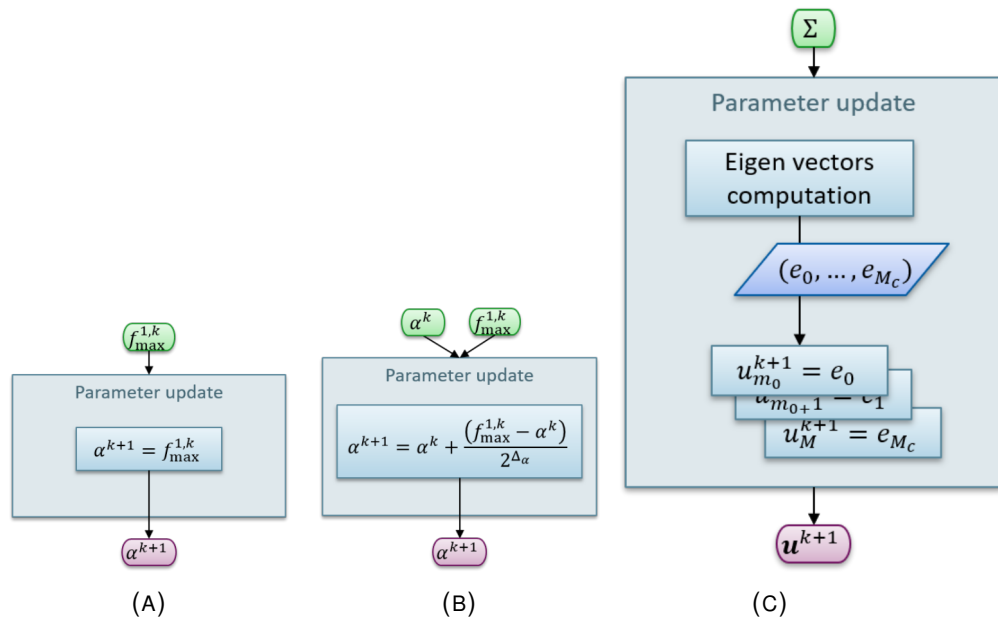
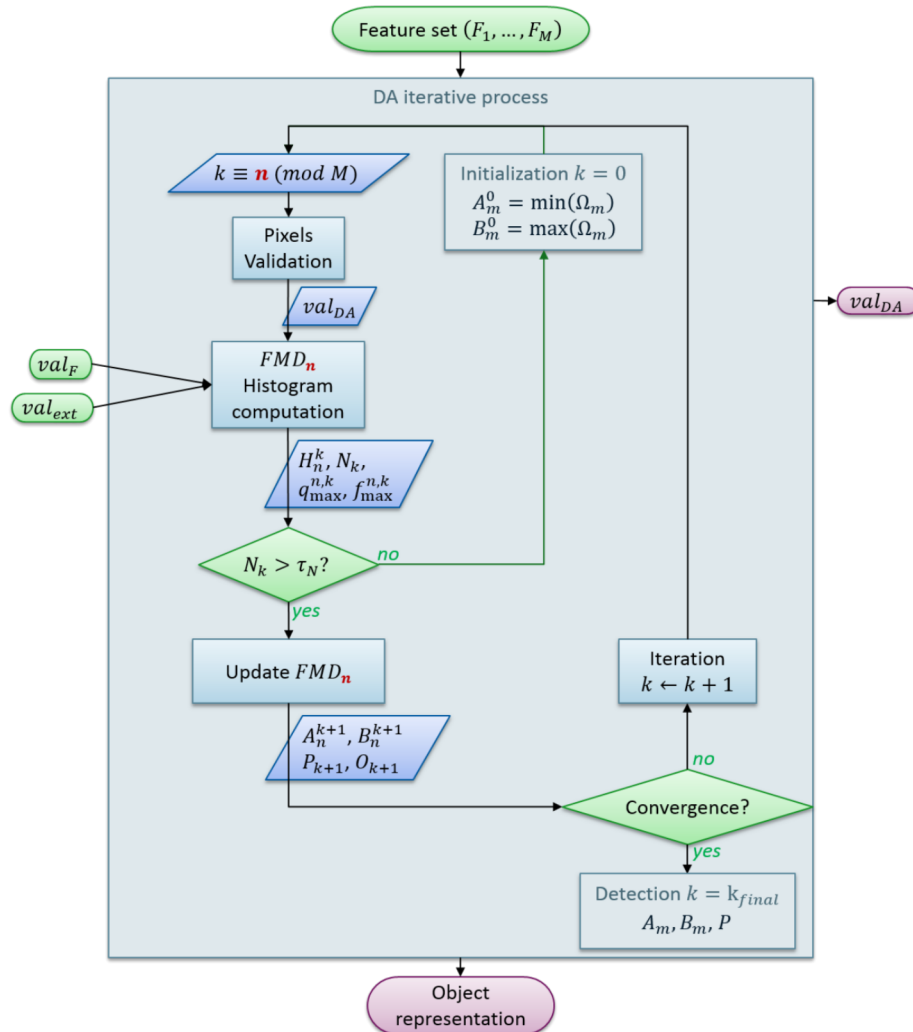


FIGURE 4.15 – Update of the rotation parameters. (A) Proposition of P. Pirm. (B) Smoothed version. (C) Generic feature space rotation.

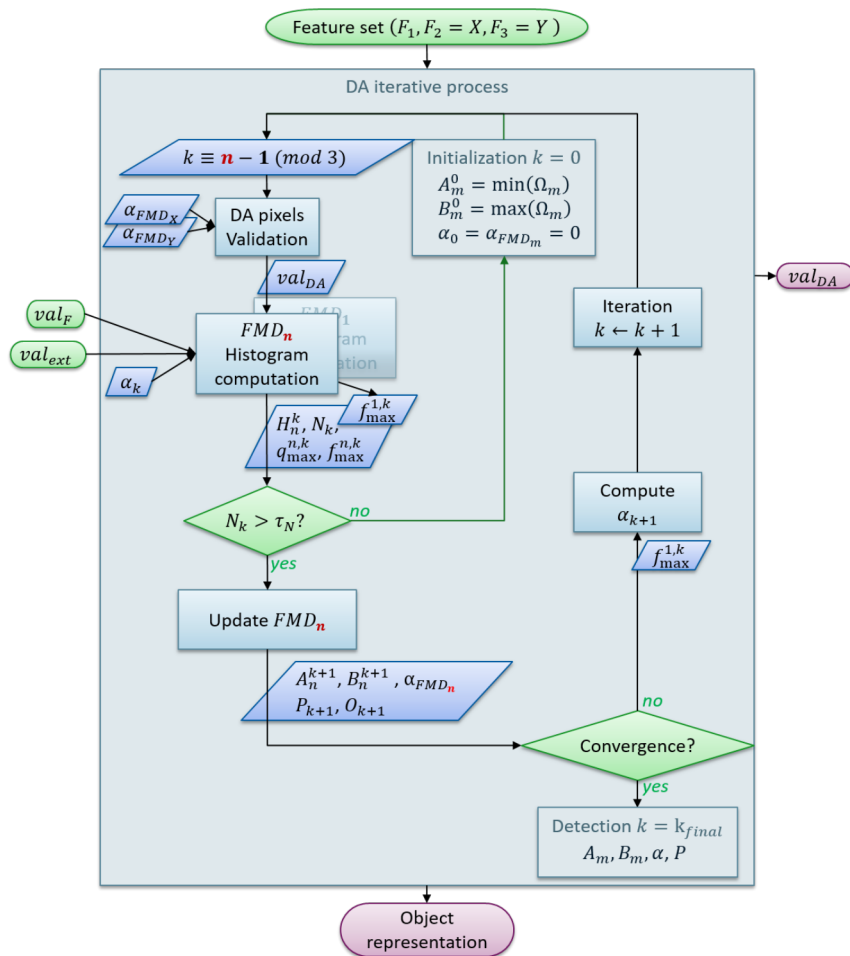
### 4.3.6 Iterative process

The iterative process of the DA permits to converge iteratively to one object of the image. Three different versions of this process have been implemented. The first one corresponds to the classical formalization of the BIPS, the second to the spatial axes rotation proposed in Chapter 5 and the third to the features space reorientation proposed in Chapter 6. The different versions of the iterative process are extensively represented in the following pages (Figure 4.16).



(A)





(B)

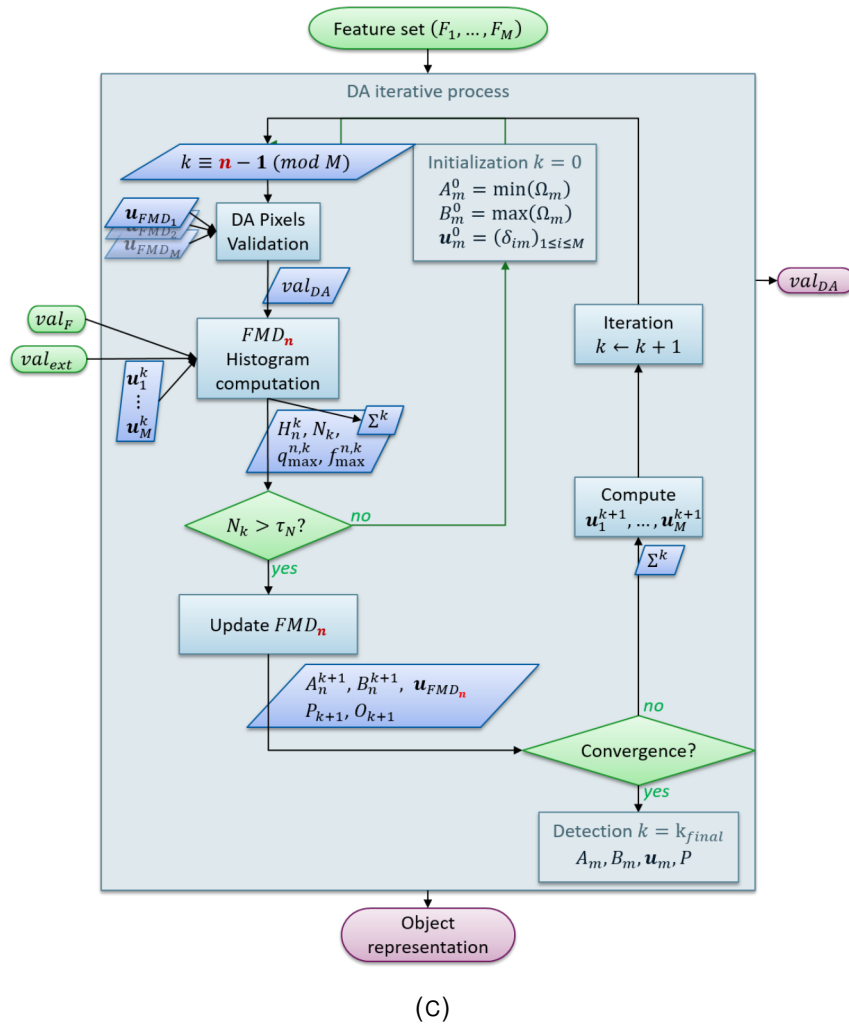


FIGURE 4.16 – DA iterative process. (A) In the classic version. (B) With the spatial rotation. (C) With the generic feature space rotation.

### 4.3.7 Time integration

The time integration proposed in Chapter 6 uses two functions: *prediction*, which performs the prediction of the boundaries for the next image; and *updateTime*, which corrects the result obtained with the DA before the prediction in the case of the Kalman filter use. The *KalmanFilter* class of (*OpenCV 2.4 release*) is used for a simplified code. The rules used to initiate the prediction and reinitialize it or not are not fixed yet and will depend on the chosen application.

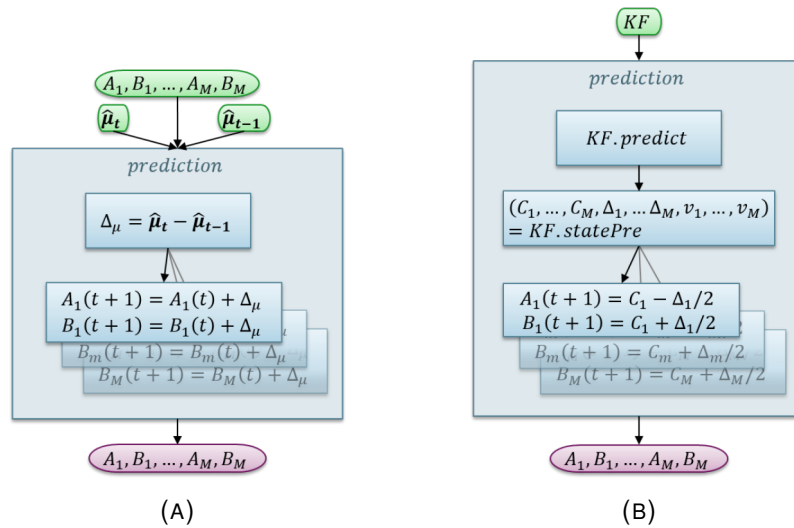


FIGURE 4.17 – Prediction implementation performed with the median (A) and with the Kalman filter (B).

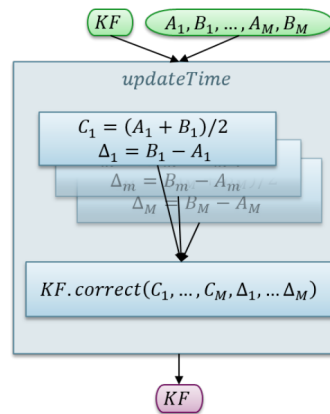


FIGURE 4.18 – Update of the Kalman filter parameters as used in the simulator.

## 4.4 Application and Visualization

### 4.4.1 Process

The complete process is presented in Figure 4.19. The inhibition rules have to be defined in the process script. The multiple parameters of the DAs are defined during the initialization step: the features used  $F$ ; their support  $\Omega_m$ ; the parameters of their extraction; the number of DAs  $n_{DA}$ ; the associated features to each DA; the threshold on object size  $\tau_N$ ; the threshold on each FMD histogram  $\tau_F$ ; the *Mode* of each FMD; the enlargement for each FMD  $\varepsilon_F$ ; the number of rotated features  $M_c$ ; the rotation rule; the prediction rule; and the limit number of iterations.

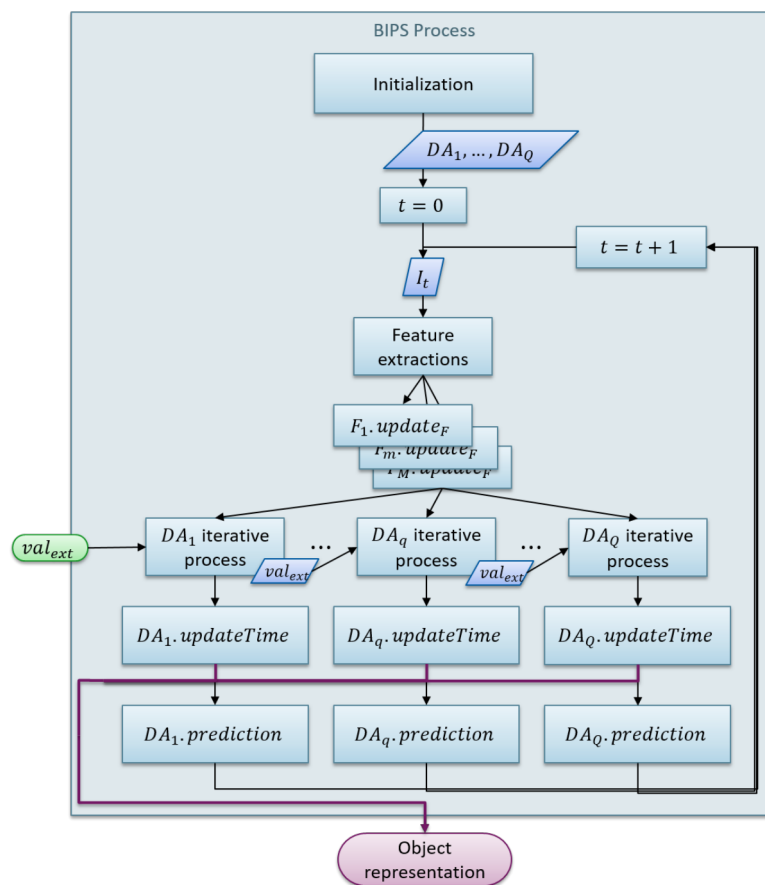


FIGURE 4.19 – Diagram of the complete process.

## 4.4.2 Images class

An **Images** class has been created to control the complete process, the different DAs and the change we might impose to the parameters through the User Interface. The User Interface itself is controlled by the class **MainWindow**, descendent of the Qt class **QMainWindow**.

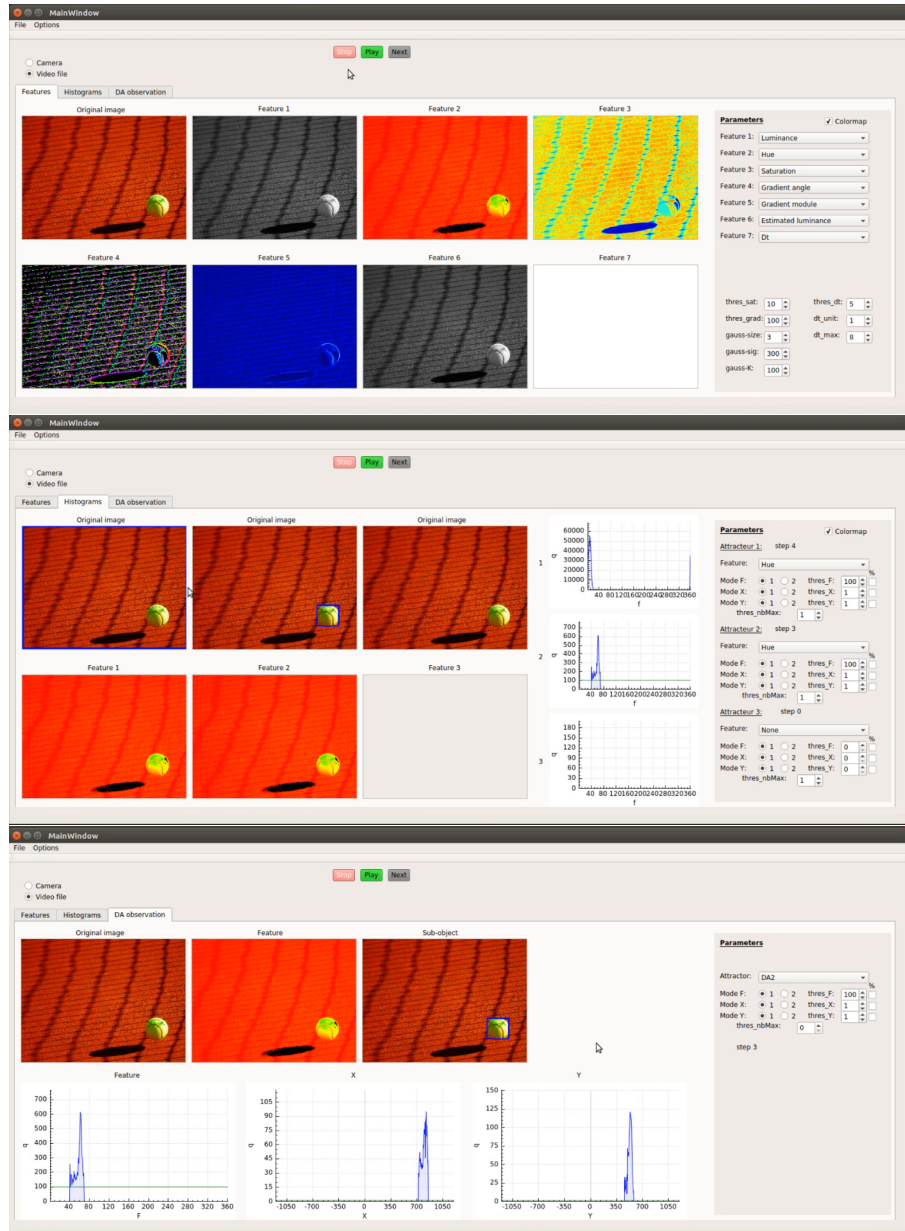


FIGURE 4.20 – Screenshots of the three visualization tabs of the User Interface.

## 4.5 Conclusion

In this Chapter we presented the simulator developed to reproduce the BIPS mechanisms that we formalized (Chap. 3). To do so, three main classes have been implemented: the **Feature** class, the **FeatModeDetector** class and the **DynamicAttractor** class. All three permit to manage the different parameters of the method. Most of the parameters of influence are thus integrated inside the classes and their setting can be changed without reimplementing the entire process. Only the architecture of the multiple DAs system could not be generalized. Thus, specific scripts have to be implemented for each new application.

The simulator integrates also the different versions of the method: the strict and non strict versions; the different rotation and reorientation of the feature space versions; and the different prediction versions. This tool allows to more easily implement the BIPS method in different applications and with different settings. It finally allows to improve and extend the functionalities of the method.



## Chapter 5

# Application to road lane detection

### 5.1 Introduction

The formalization of the Bio-Inspired Perception Sensor (BIPS) concept presented in Chapter 3 has highlighted several parameters of the method that can have a strong influence on its results: the input features chosen, the connections between Dynamic Attractors (DAs), the chosen detection mode, on one peak or on a group of peaks, and the different thresholds on features, on object sizes and on feature histograms. Neither the choice of these parameters for a practical application nor their level of influence on the whole concept is made explicit by the inventor. The formalization, although it provides many answers to the understanding of the concept, does not give a direct answer to this problematic. Thus, we decided to illustrate the behavior of the method and the possible influence of its parameters, using a specific computer vision application.

The road lane detection application, which was one of the industrial application of the BIPS (Pirim, 2013), is appropriate for a first analysis and demonstration of the BIPS concept behavior, since the lanes are distinguished by their gradient angle and their position in the image. It allows to set the architecture of the DAs, i.e. the choice of the features, the DAs connections and the mode of detection and thus to focus on the threshold parameters. Moreover it enforces the axes rotation extension of the BIPS concept -introduced in Section. 3.4.2-, which permits to improve the spatial framing.

This Chapter investigates the BIPS behavior and performances in the context of the road lane detection application. Section 5.2 gives an overview of the chosen application, its related issues and the existing solutions. Section 5.3 establishes the adapted architecture of the BIPS based on the two straight lines model of the road lane and permits to develop the spatial axes rotation extension. Using the *caltech-lanes* database (Aly, 2008), we were able to compare the performances of the method and to illustrate the influence of some of its parameters in Section 5.4. The promising results consolidate the focusing of our research on this method and its possible improvements, as well as its potential adaptation to different computer vision applications, as discussed in Section 5.5.

### 5.2 Lane detection

The road lane detection is a specific application related to advanced driver assistance systems (ADAS) and autonomous driving fields of research. Both fields are driven by the need of reducing the high mortality rate on road and the willing of improving the driver comfort, which makes them a hot research topic nowadays.



This Section gives an understanding of the road lane detection application and its current issues, in order to suitably adapt the BIPS method. The context and difficult conditions of the application are presented in Section 5.2.1. Then, the existing methods and models are discussed in Section 5.2.2. Finally, Section 5.2.3 presents the available databases for evaluation, before concluding in Section 5.2.4.

### 5.2.1 Application context

The road lane detection is part of several applications like the Lane Departure Identification (LDI) and Lane Departure Warning (LDW), the Lane Keeping Assist (LKA) and the Adaptive Cruise Control (ACC). The purpose can be to wake up a falling asleep or distracted driver; to control the vehicle direction; and also to help the control system in its road scene understanding. Such purposes go with a high reliability demand from the vehicle manufacturers under multiple conditions.

In fact, the aspect of the lanes and the road strongly differ from a time and place to another. Rainy, snowy or foggy weather lowers the visibility. Night strongly reduces the visibility. Day light adds multiple artifacts like shadows, sun facing, change of colors and luminance. Moreover, the road type does not follow a unique road model: on highway there are few artifacts and the lanes are easily described by pair of lines or curves, but the detection must work at long distance as the speed of the vehicle is high. On rural environment lanes and signs can even not exist at all. On urban environment multiple disturbing objects can obscure the road and the lane model must take into account a wider range of existing signs on the road like pedestrian lanes, merging and splitting lanes, bike lane, writings, stop lane, etc. The differences existing from one country to another also add difficulties to the task. Figure 5.1 shows different aspects of the road the vehicle might be confronted to.



FIGURE 5.1 – Road images under different conditions presented in (Bar Hillel et al., 2014). The red circles and lines are markers from the initial database.



FIGURE 5.1 – Road images under different conditions presented in (McCall and Trivedi, 2006).

The bibliographic references survey (Bar Hillel et al., 2014) and (McCall and Trivedi, 2006) show that researchers have been using different kinds of sensors to address this task: monocular visible camera, stereo cameras, LiDAR, GPS, internal vehicle state sensors and radar. The three latter are used to enhance the results obtained with the others. Even though there is obvious benefit from the fusion of those sensor's data, the monocular visible camera, robust for a cheap price and the closer to the actual human driver main modality, is the most used sensor.

### 5.2.2 Methods and Models

In their survey (Bar Hillel et al., 2014) propose a generic road lane detection method based on the state-of-the-art of the task. The method is decomposed as follows: 1-Image pre-processing, 2-Feature extraction, 3-Model fitting, 4-Time integration, 5-Image to world correspondence, as shown in Figure 5.2. A similar framework composed of three blocks was also proposed in the survey of (McCall and Trivedi, 2006): the first block (road feature extraction) is a combination of 1 and 2, the second block (post-processing/outlier removal) corresponds to 3 and the third block (tracking/filtering/data fusion) corresponds to 4 and 5. The numerous feedback arrows coming from the last processes bring forward the duality between the bottom-up extraction of the lane and the top-down influence of the model and real-world measures in this method.

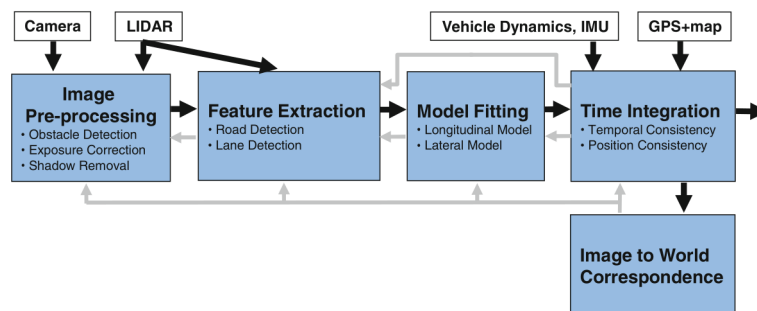


FIGURE 5.2 – Generic method for road lane detection proposed by (Bar Hillel et al., 2014).

### Pre-processing

The pre-processing part consists of cleaning the input image from all misleading artifacts like illumination variation -as in (Son et al., 2015)- or shadows -as in (Hoang et al., 2017)- and of removing irrelevant parts. The latter can correspond to the sky or the road border, which are discarded using a region of interest (ROI) selection. Most of the time a fixed mask is used like in (Aly, 2008), but some authors also use an adaptive ROI, as for example (Cáceres Hernández et al., 2016) who propose a geometrical adaptation of the ROI. Other vehicles or pedestrians present on the road can also be considered as irrelevant parts of the image and removed as for example in (Jung and Bae, 2018), where the LiDAR information is used to distinguish them. The pre-processing part must be chosen in a good balance between the brought improvements and the consequent computational costs.

### Feature extraction

The bottom-up feature extraction part is predominantly done using steerable filters to obtain the gradient and curve information (McCall and Trivedi, 2006). Inverse-perspective as well as other features like colors, texture, road boundaries or vanishing points have also been used to enhance the results (Andrade et al., 2019)(Felisa and Zani, 2010)(Borkar, Hayes, and Smith, 2009)(Audibert and Ponce, 2009)(Aly, 2008).

### Model fitting

The top-down model fitting part uses associated lane models divided by (Bar Hillel et al., 2014) into three categories: the parametric, the semi-parametric and the non-parametric models. The first one corresponds to the research of straight lines, which is mostly done using a Hough transform or a least square optimization (Gaikwad and Lokhande, 2015). The second model corresponds to splines and poly-lines, which can be found using piece-wise Hough transform results or RANSAC results (Andrade et al., 2019)(Son, Lee, and Kum, 2019). Finally, the third model corresponds to continuous, but not necessarily smooth boundaries, which have been addressed using Ant Colony optimization, Hierarchical Bayesian network or particle filters (Broggi, 1995)(Nefian and Bradski, 2006). This last model essentially corresponds to off-road boundaries search and is rarely chosen. Recently, some authors also try to enhance the obtained results in complex scenes, where the lines or splines models strongly failed, with a trained neural networks (Zhang et al., 2018).

### Time integration

The time integration part permits the readjustment and the robustness improvement of the method and model chosen, while theoretically reducing the computational cost per frame. The methods mostly use the Kalman filter or the Particle filters (Borkar, Hayes, and Smith, 2009)(Kim et al., 2007), but interesting propositions integrate also the time directly in the feature extraction, searching the line in a spatiotemporal representation space (Jung, Youn, and Sull, 2016).

### 5.2.3 Databases

The evaluation of the different methods is quite difficult as (Bar Hillel et al., 2014) raise in their conclusive remarks. In fact, most of the proposed methods are only qualitatively evaluated due to the lack of annotated datasets and generalized evaluation metrics. Several databases showing images of road lane exist (*KITTI*) (Veit et al., 2008) (Wu and Ranganathan, 2012) and tools for manually labelling the data have been provided by (Borkar, Hayes, and Smith, 2012) and (Aly, 2008), but it seems that only the *caltech-lanes* dataset of Aly provides a joint ground truth and metrics with which tests and comparison can directly be made. However, as for example in (Hoang et al., 2017), the authors often prefer to relabel themselves the database, without making their labeled data public.

### 5.2.4 Conclusion

The overview of the road lane detection task brings forward the complexity and variety of the road scene to analyze, but also the multitude of models and methods used over the last twenty years. The generic method formalization proposed by Hillel et al. gives however a tool for the adaptation and the comparison of the BIPS method, which in its current state corresponds to the second and third block of the generic method: the feature extraction and the model fitting. It helps us in the choice of the lane model to follow and in the method extension possibilities using pre-processing or time integration. A comparative analysis of the BIPS results is possible using the *caltech-lanes* dataset and its ground truth. A broader study comparing the method on multiple datasets and multiple existing methods would, however, require much more workforce due to the lack of unified ground-truths and metrics for the road lane detection task.

## 5.3 BIPS adaptation

In this study, we choose to adapt the BIPS method to the road lane detection application using the visible camera input and the straight lines model of the lane. This Section demonstrates how such adaptation can be made and how several of the unset parameters can be automatically chosen.

Section 5.3.1 develops the choice of the descriptive and spatial features in this context, especially the use of rotated spatial coordinates. Then Section 5.3.2 defines the FMDs parameters adapted for one line detection and the adapted connections between DAs to ensure the detection of all the road lanes and only them, before concluding in Section 5.3.3.

### 5.3.1 Features space

#### Gradient angle

The straight lines model of the road lane allows to distinguish each lane by their gradient angle. The proposed extraction in the chip (Sec. 3.2.3) could be used for this application. However, we prefer its generalized version, using the derivative of a 2D Gaussian steerable filter (Eq. 5.1), for which the standard deviation  $\sigma$  can be chosen depending on the input image. The kernel size  $(2n + 1)$  is often chosen depending on  $\sigma$  to restrict the kernel to the significant values of the Gaussian. In this study we always choose  $n = \lceil 3\sigma \rceil$ , and more specifically  $\sigma = 1.5$  and  $n = 4$ .

The threshold  $\tau_{mo}$  (Eq. 3.11) can then be chosen depending on the maximum possible value of the gradient module allowing to automatize and normalize the gradient angle extraction.

**Definition.** Gaussian kernel  $W$

$$W = \frac{\partial G}{\partial j} \propto \frac{-j}{2\pi\sigma^4} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right), \forall (i, j) \in \llbracket -n, n \rrbracket^2 \quad (5.1)$$

In Figure 5.3, we can visualize the concordance between a road lane and a gradient angle. The road images angle extraction has been performed with a threshold for the angle validation at 5.5% of the possible maximum. We can observe on the two images (Aly, 2008)(*Road*) that the lanes are distinguishable from the rest. Each of them corresponds to a thin peak of the gradient angle histogram.

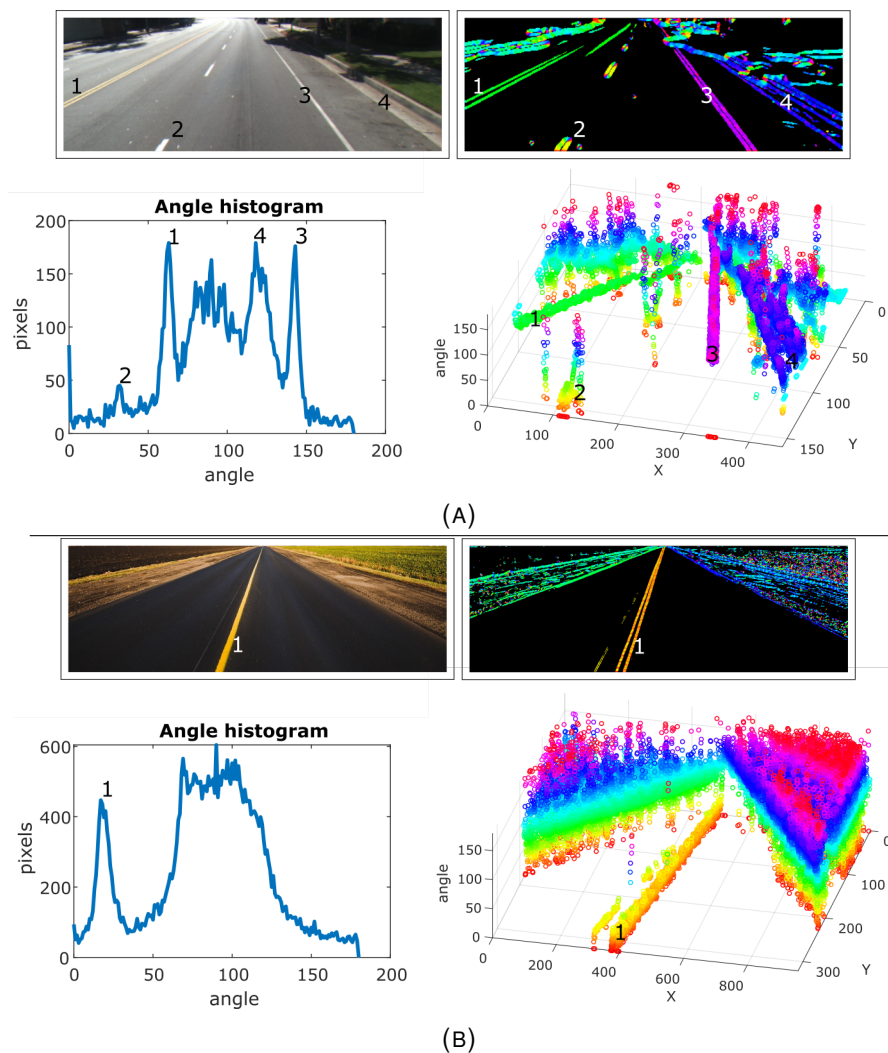


FIGURE 5.3 – Extraction of the gradient angle on road images from (A) (Aly, 2008), (B) (*Road*) showing how the lanes are distinguished by this feature. The numbering of the lanes gives the correspondence in the four representations. Upper-left: the original image; Upper-right: gradient angle extraction; Bottom-left: gradient angle histogram; Bottom-right: 3D feature space representation.

### Spatial features

As shown in 3.4.2, the Cartesian spatial coordinates can give a poor framing of the lines, whereas the use of rotated axes  $X$  and  $Y$  in the direction of the detected line would fit better. The rotation equation of the clockwise oriented axes is reminded in Eq. 5.2. The difficulty of such change is that the rotation angle  $\alpha$  is not initially known. Several solutions have been proposed in (Pirim, 2001) and (Pirim, 2005), implying adaptation of the DA iterative process and possible mismatch of its convergence property. Here we present our study of those propositions and our extension for a better implementation in our simulator (App. 4.3).

**Definition.** Rotated spatial axes

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} j \\ i \end{pmatrix} \quad (5.2)$$

**Proposition 1.** Multiple framing computation

The first and early proposition was to run simultaneously several pairs of FMDs, each of them on different spatial axes corresponding to a different rotation, then to select the axes obtaining the highest histogram peak on  $X$ . This can be done also by using identical DAs running simultaneously on the same set of pixels, but with different spatial axes inputs. This adaptation brings numerous unnecessary computations due to the multitude of FMDs. Moreover, the selected rotation is not precise if not enough pairs of FMDs are used.

Figure 5.4 presents the obtained results with this method on the road image (*Road*). In this example, four sets of FMDs are running simultaneously using respectively the rotated axis at  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ . The comparison of the  $X$  coordinate histograms shows that the chosen framing could be either at  $0^\circ$  or  $45^\circ$  as they have the highest peaks (43 pixels compare to 15 and 24 pixels). However none fits well the lane. It shows the limitation of the method, which needs a highest number of FMDs to be able to find a suitable rotation of the spatial axes.

**Proposition 2.** Successive framing computation

A first adaptation of the DA process keeps only a pair of FMDs for the spatial coordinates and computes the rotation  $\alpha$  during the DA iterative process. After a few steps with the Cartesian axes  $(J, I)$ , the rotated axes  $X$  and  $Y$  are defined using the gradient angle histogram mode  $f_{\max}^{ang}$  as the estimated rotation angle value. In (Pirim, 2005), it is also proposed to use the median of the gradient angle histogram. Then, the input features of the second and third FMDs which were the pixel coordinates  $j$  and  $i$ , become the rotated coordinates  $x$  and  $y$ . This can also be done using two DAs running successively, the first one working on the Cartesian axes, the second working on the rotated axes and conditioned by the framing of the first DA. This adaptation, however, might select an incorrect rotation angle, as the Cartesian framing can be corrupted by noise or close lines.

Figure 5.5 compares the results obtained using the successive framing computation (**Proposition 2**) and the iterative angle selection (**Proposition 3**). The gradient extraction is made with a threshold at 4% of the possible maximum and 40 DAs are running successively. The object size threshold is set at  $\tau_N = 30$  pixels. The DAs inhibit each other using their gradient angle selection  $val_{FMD_{ang}}$  (Eq. 3.21). We can observe that the successive framing computation results are often deviated when two lanes are close to each other: the DAs number 11, 17 and 26 for example are all corrected when using the iterative angle selection.

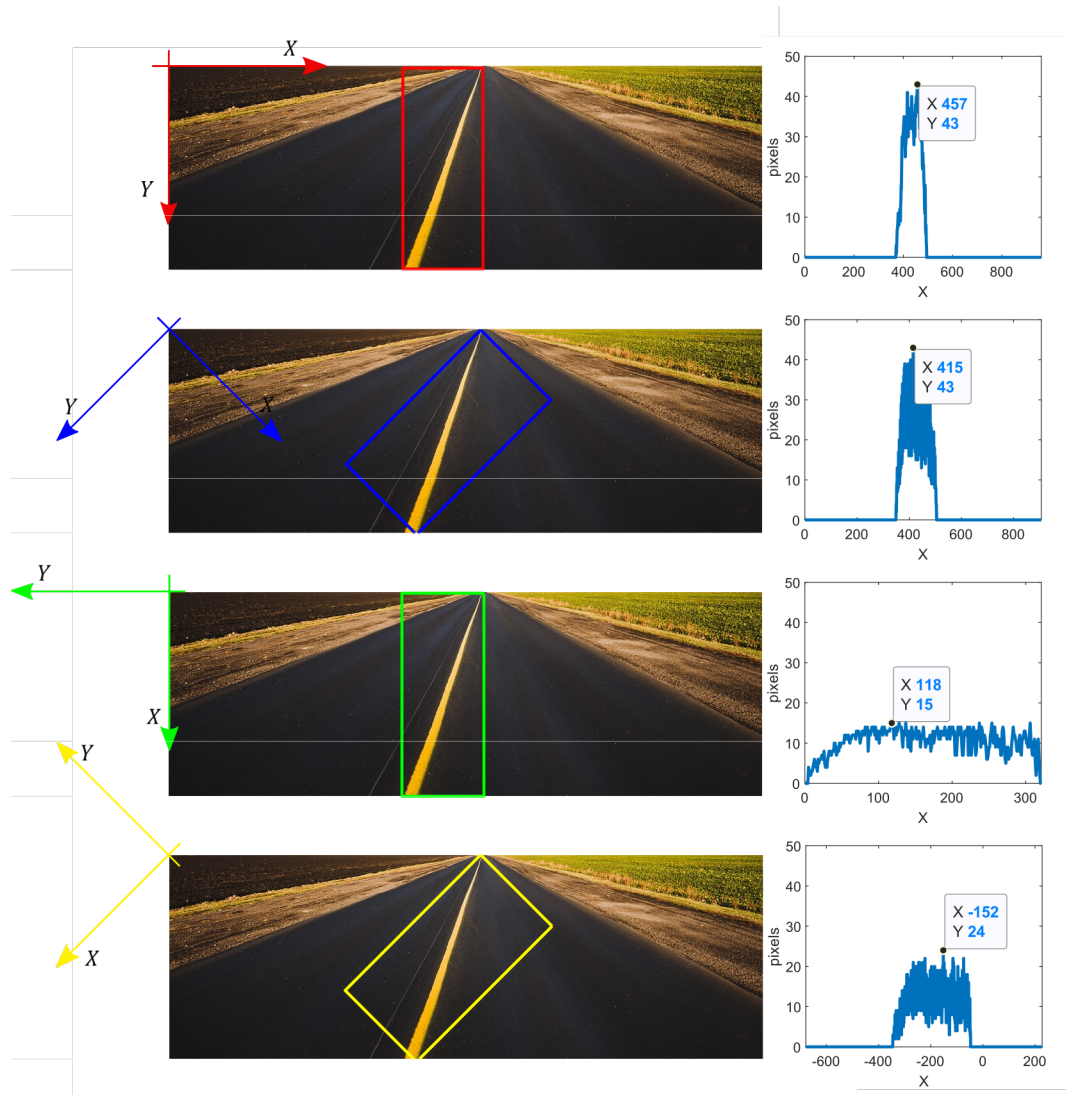


FIGURE 5.4 – Representation of the obtained framing on a road image (*Road*) for a set of 4 rotation angles as proposed in (Pirim, 2005) and the corresponding histograms on the spatial coordinate  $X$ . This brings forward: (1) the variation of the axes histograms depending on the orientation and (2) the difficulty of finding a suitable orientation with the multiple framing computation (**Proposition 1**). From up to bottom:  $\alpha = 0^\circ$ ,  $\alpha = 45^\circ$ ,  $\alpha = 90^\circ$ ,  $\alpha = 135^\circ$ .

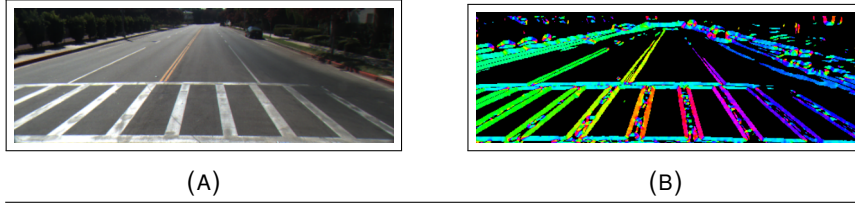
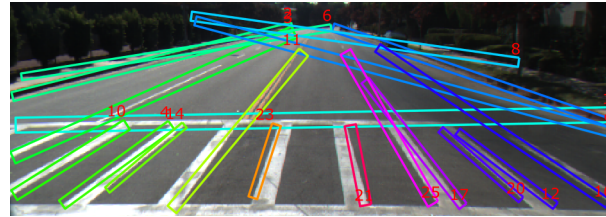
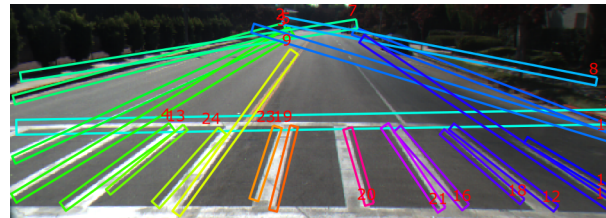


FIGURE 5.5 – (A) Road image (Aly, 2008). (B) Angle extraction.



(C) Successive framing computation (**Proposition 2**)



(D) Iterative angle selection (**Proposition 3**)

FIGURE 5.5 – Comparison of the results obtained with (C) the successive framing computation (**Proposition 2**) and (D) the iterative angle selection (**Proposition 3**). It shows the deviated framings -11, 17, 26- obtained with Proposition 2 and corrected by Proposition 3. 40 DAs are running successively and linked by inhibition of their gradient angle selection.

### Proposition 3. Iterative angle selection

In this proposition, the gradient angle histogram mode  $f_{\max}^{ang,k}$  is used at each step of the DA iterative process to better estimate the rotation angle  $\alpha_k$  and avoid the incorrect selection observed with the **Proposition 2**. Then, the new coordinates  $X$  and  $Y$  evolve at each step of the process. Therefore, the boundaries of the second FMDs -framing the  $X$  coordinate and computed at step  $k \equiv 2 \pmod{3}$ - and the boundaries of the third FMDs -framing the  $Y$  coordinates and computed at step  $k \equiv 3 \pmod{3}$ - does not necessarily correspond to the same pair of rotated coordinates.

Supplementary parameters  $(\alpha_{FMD_m})_{m=\{2,3\}}$  are then necessary to compute the pixel validation of the FMDs 2 and 3. They correspond to a memorization of the  $\alpha_k$  parameter used at their last boundaries update. This proposition implies a change in the iterative process represented in Figure 5.7.

This new iterative process does not guarantee the decreasing of the framing  $P_k$ . Nevertheless, we could observe a conservation of the objects detection during our experiments, along with, in some cases, an oscillation around the best rotation angle value. Such oscillation prevents the detection to be validated in our algorithm (Eq. 3.33). Figure 5.6 shows an example of the oscillation obtained when using the



iterative angle selection (**Proposition 3**) in comparison with the iterative angle estimation (**Proposition 4**), which smooths the rotation for a better framing adapted to our algorithm.

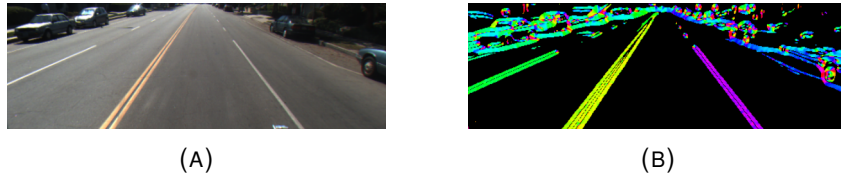
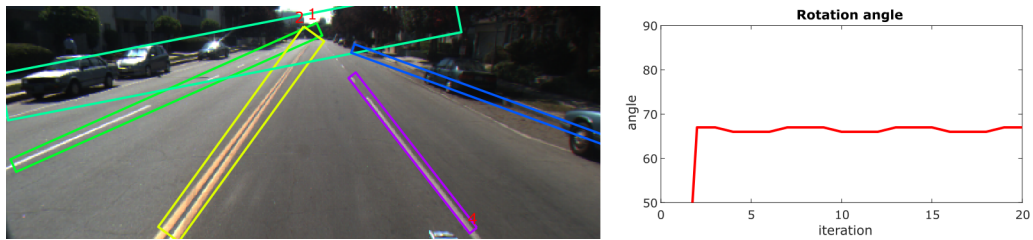


FIGURE 5.6 – (A) Road image (Aly, 2008). (B) Gradient angle extraction.



(C) Iterative angle selection (**Proposition 3**)



(D) Iterative angle estimation (**Proposition 4**)

FIGURE 5.6 – Comparison of the results obtained with (C) the iterative selection (**Proposition 3**) and (D) the iterative estimation (**Proposition 3**). Although the obtained framing are similar and close to the lanes, the iterative selection of lane 1 (green) oscillates and does not validate the convergence test.

#### Proposition 4. Iterative angle estimation

In this last proposition, we modified the iterative angle selection, such that the oscillation observed would be slowed down by smoothing the rotation angle estimation with a geometrical approximation (Eq. 5.3). By doing so, the estimated  $\alpha_k$  is influenced by its last values and can be stabilized for several steps at the same value. It allows the algorithm to validate the convergence of the DA.

**Definition.** Geometrical approximation of  $\alpha_k$

$$\alpha_{k+1} = \alpha_k + \frac{f_{\max}^{n,k} - \alpha_k}{2\Delta_k} \quad (5.3)$$

$$\Delta_{k+1} = \begin{cases} \max(\Delta_k - 1, 0) & , \text{ if } |\alpha_k - f_{\max}^{n,k}| > \tau_\alpha \\ \min(\Delta_k + 1, 8) & , \text{ if } |\alpha_k - f_{\max}^{n,k}| \leq \tau_\alpha \end{cases}$$

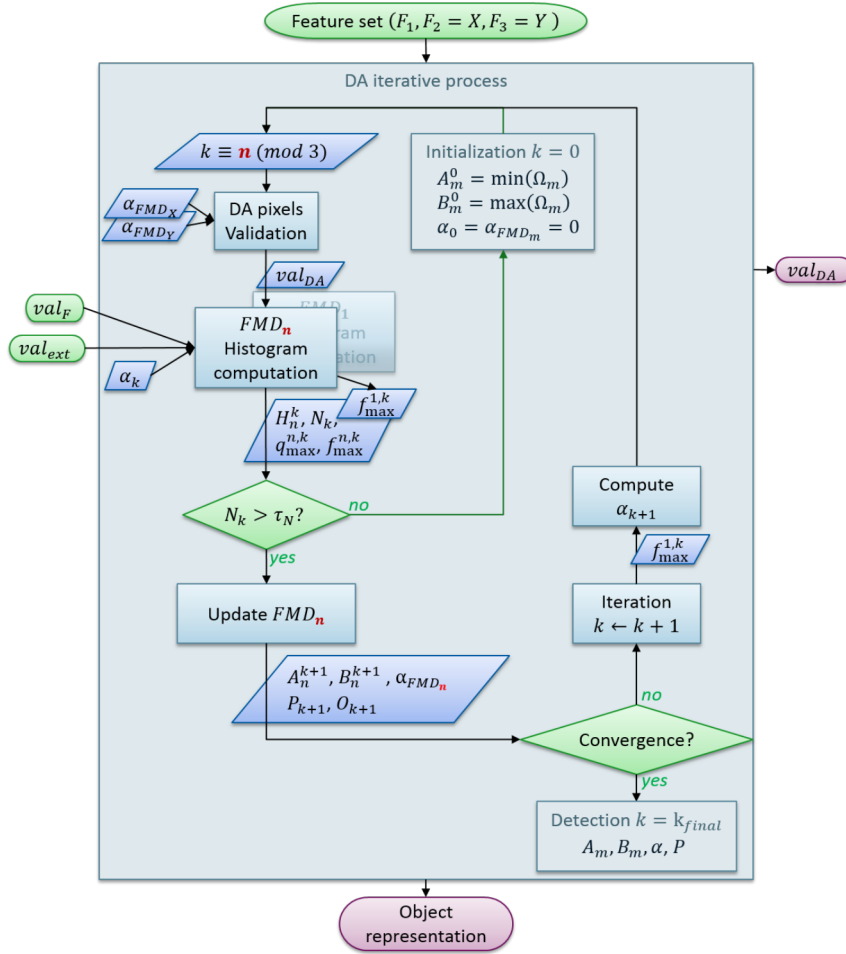


FIGURE 5.7 – Representation of the DA iterative process integrating the axes rotation. This diagram is valid for **Proposition 3** and **Proposition 4**.

### 5.3.2 Architecture

#### DA parameters

The architecture of DA we choose concentrates on detecting a line. In the chosen feature space ( $F_1 = ang$ ,  $F_2 = X$ ,  $F_3 = Y$ ), a line corresponds to one mode of the gradient angle and a position on the rotated axis  $X$ . The parameters setting of the DA is, thus, facilitated by the chosen feature space. As it has already been shown in the examples, the DA is composed of three FMDs, one working on the gradient angle feature, one on the rotated  $X$  axis, and one on the rotated  $Y$  axis. Mode 1 is used for both the gradient angle and the position  $X$  to find the line defining values and Mode 2 is used on the axis  $Y$  to regroup the aligned dashed lanes into one detection. The threshold on the  $Y$  axis is chosen null  $\tau_Y = 0$  to regroup all the aligned pixels of the lines. The boundaries  $A_Y$  and  $B_Y$  gives information on the lane endpoints, especially if it is shorter than the view of the road. The threshold on the gradient angle  $\tau_{ang}$ , the position  $\tau_X$  and the size of the line  $\tau_N$  are left unset and their influence will be analyzed in Section 5.4.3.

### Multiple lanes detection

A series of such DAs, connected by inhibition should allow to detect all the lines in the frame. However, the classic inhibition performed based on the  $val_{DA}$  applications (Eq. 3.30) only inhibits the pixels in the three features framing of the DAs. As we already observed in Section 3.4.2, this framing can cut the detected lines and then leaves several pixels corresponding to the same line uninhibited. Moreover, the context of road lane detection allows to define a spatial region around each detected lane where no other lanes should be looked for and to associate only one gradient angle per detected lane. The spatial region can be defined by the combined spatial FMD pixel validations with an extension of their boundaries as proposed in Eq. 5.4 and the inhibition of the selected gradient angle can be defined as in Eq. 5.5. The extension represented by the parameter  $\varepsilon$  can be chosen manually or proportional to the width of the selected histogram mode  $|B - A|$ , allowing a more adapted setting of this parameter.

**Definition.** Spatial inhibition coming from one DA

$$\begin{aligned} val_{ext}^k : \mathbb{P} &\longrightarrow \{0, 1\} \\ p &\longmapsto NOT \left( val_{FMD_X}^k(p, \varepsilon_X) \wedge val_{FMD_Y}^k(p, \varepsilon_Y) \right) \end{aligned} \quad (5.4)$$

**Definition.** Gradient angle inhibition coming from one DA

$$\begin{aligned} val_{ext}^k : \mathbb{P} &\longrightarrow \{0, 1\} \\ p &\longmapsto NOT \left( val_{FMD_{ang}}^k(p, \varepsilon_{ang}) \right) \end{aligned} \quad (5.5)$$

**Definition.** FMD pixel validation extension

$$val_{FMD}(p, \varepsilon) = (A - \varepsilon \leq F(p) \leq B + \varepsilon)$$

### 5.3.3 Conclusion

Through the road lane detection application, we have been able to set and automatize multiple BIPS parameters: the gradient angle extraction can be adapted to the input image through the selection of the Gaussian variance; the spatial axes are rotated to always fit the same alignment with the detected line; the DAs used are identical and linked by spatial inhibition; and the update modes of their three FMDs are defined.

This setting process allowed to develop the first extension of the BIPS method, the iterative rotation of the axes and to reduce the number of parameters of influence to four: the threshold on the descriptive feature, the position and the line size, as well as the number of DAs.

## 5.4 Experimental results

This Section presents the results obtained on the *caltech-lanes* dataset (Aly, 2008). The experimentation allows to compare our BIPS-based approach to a known academic method and to illustrate the impact of the unset parameters on its performances.

The chosen dataset is described in Section 5.4.1, along with the evaluation method of the BIPS results. Then in Section 5.4.2, we use the connection between the object size and the number of DAs to reduce the number of parameters and to improve the performance of the method. Finally, in Section 5.4.3, we illustrate the impact of the two left threshold parameters, before concluding in Section 5.4.4.

### 5.4.1 The *caltech-lanes* dataset

The *caltech-lanes* dataset, developed by Aly et al., is composed of four sets of videos taken from a car. The clips contain several difficulties: curve lanes, writing on the street, sun facing, strong shadows, passing cars, street crossing, etc. The associated ground truth concentrates only on visible road lanes from the car and provides a 3<sup>rd</sup> degree Bezier spline description of them as shown in Figure 5.8.

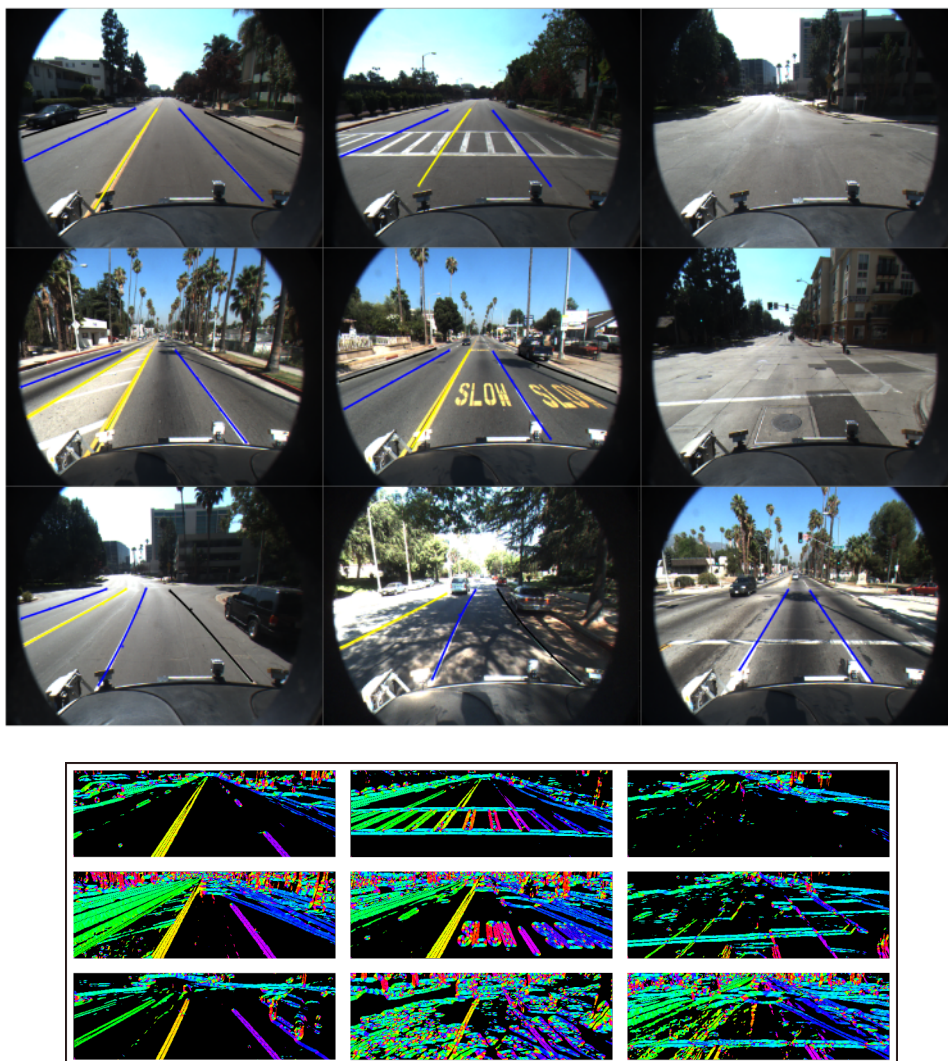


FIGURE 5.8 – (A) Images and Ground truths from the *caltech-lanes* dataset showing the encountered conditions. (B) Associated gradient angle.

In (Aly, 2008), the RANSAC method is used to fit the lanes. In a pre-processing step, a fixed mask selects the road part of the image and an inverse perspective mapping (IPM) provides a nadir ("birdeye") view of the road, assumed as being flat. The descriptive features of the lanes are extracted from the IPM images using a 2D second degree filter. Its convolution with the image allows to enhance points corresponding to the lanes which are then extracted using a threshold. From this set of points the detection of the lanes is first performed using the histogram on the number of extracted pixels per longitudinal coordinate. The local maxima of the histogram gives the positions of the lanes. The RANSAC method is used a first time here to fit the lines. From this detection Aly et al. extract bounding boxes of the lanes, which are used to perform the RANSAC method again, but this time to fit the 3<sup>rd</sup> degree Bezier splines.

The metric used for the validation of their detection is the minimum distance between the detected spline  $s$  and each ground truth spline  $s_{GT}$  (Eq. 5.6). The validation of the detection corresponds to  $d_{mean} < 20$  pixels and  $d_{median} < 15$  pixels.

**Definition.** Detection validation metric

$$\begin{aligned} d_{mean} &= \min \left( \text{mean}_{p \in s} (dist(p, s_{GT})) , \text{mean}_{p \in s_{GT}} (dist(p, s)) \right) \\ d_{median} &= \min \left( \text{median}_{p \in s} (dist(p, s_{GT})) , \text{median}_{p \in s_{GT}} (dist(p, s)) \right) \end{aligned} \quad (5.6)$$

### BIPS settings

The BIPS settings proposed in Section 5.3 are still valid for this experiment and this dataset. The Gaussian standard deviation for the gradient extraction is set to  $\sigma = 1.5$ , the threshold for the angle validation is set at  $\tau_{mo} = 4\%$  of the maximum possible value and the angle minimum and maximum values have been restricted to avoid the horizon detection. As the dashed lane are represented by one spline in the ground, the mode 2 is used on the axis  $Y$  (see Sec. 3.3.3 for an illustration of the mode 2 effect). The inhibition connection corresponds to the spatial framing exclusion on one hand and the feature exclusion on the other hand, as, thanks to the perspective, every lane corresponds to a different gradient angle and lies in a different area of the image (Eq. 5.4 and 5.5). The validation extensions are set to  $\varepsilon_X = 3/2|B_X - A_X|$ ,  $\varepsilon_Y = 0$  and  $\varepsilon_{ang} = 2|B_{ang} - A_{ang}|$ , from empirical observations made on the *cordova1* clip. In this experiment, the DAs are running successively and not pipelined as for our first implementation of the process. When a DA converged to a small size object (noise), our algorithm does not reinitialize this DA and uses its gradient angle mode to inhibit the following DAs (Eq. 5.7).

**Definition.** Feature mode inhibition

$$\begin{aligned} val_{ext} : \mathbb{P} &\longrightarrow \{0, 1\} \\ p &\longmapsto \left( F(p) \neq f_{\max}^{ang, k_{final}} \right) \end{aligned} \quad (5.7)$$

A pre-processing step, selecting a fixed ROI concentrated on the visible road lanes, is also done under the form of an input mask selecting this part of the image and used as pixel validation input (Eq. 5.8).

**Definition.** Mask input

$$val_{mask}(i, j) = \{54 < j < 575\} \wedge \{170 < i < 343\} \quad (5.8)$$

### Evaluation metrics

In order to evaluate our BIPS-based method and compare its results to the reference, we added a post-processing step transforming the DAs output into a 1<sup>st</sup> degree spline  $s$ . Two knots describe this spline  $p_1$  and  $p_2$ , they correspond to the detected line extremities (Eq. 5.9). Therefore, the same metric as Aly can be used to validate the lane detection. The Matlab functions `ccvCheckMergeSplines` and `ccvEvalBezSpline` (with a 0.01 step for the splines points), which are part of the code of (Aly, 2008), have been used to perform the validation.

**Definition.** Spline knots

$$\begin{cases} p_1 = R_{\alpha_{DA}} \begin{pmatrix} x_{\max} \\ A_Y \end{pmatrix} \\ p_2 = R_{\alpha_{DA}} \begin{pmatrix} x_{\max} \\ B_Y \end{pmatrix} \end{cases} \quad (5.9)$$

with

$$R_{\alpha} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

Only one DA per lane is validated. The first DA converging on the lane is considered as true positive (TP), but the following DAs close enough to represent the lane are considered as false positives (FP), as well as the converging DAs not associated with any lanes. The DAs that did not converge are considered as true negative (TN). Finally, the lane that haven't been detected by any DAs are considered as false negative (FN). The false positive rate (FPR) Eq. 5.10 and the true positive rate (TPR) Eq. 5.11 are used to compare the results.

$$FPR = \frac{FP}{TN + FP} \quad (5.10)$$

$$TPR = \frac{TP}{TP + FN} \quad (5.11)$$

#### 5.4.2 Number of DAs and object size impact

In the first step of the experimentation, we set the values of the feature thresholds to  $\tau_{ang} = 30\%$  and  $\tau_X = 50\%$ , as we estimate that the noise in both features is very low and the objects well separated in the joint histogram projection. The threshold on the object size is kept variable to form the ROC curve. In fact, a lower value of this threshold implies an augmentation of the number of DAs converging, thus an augmentation of true and false positives, whereas a higher value of the threshold restrain the convergence of the DAs until none of them converges at all. There is one parameter left free: the number of DAs used.

##### Fixed number of DAs

The first set of experiments have been done using a fixed number of DAs on the four videos of *caltech-lanes*. Three values have been used:  $n_{DA} = 10$ ,  $n_{DA} = 25$  and  $n_{DA} = 40$ . We chose such high number of DAs, because by using them sequentially without reinitialization, in this experimentation, many can be attracted by noise or small insignificant lines before all the lanes are detected.

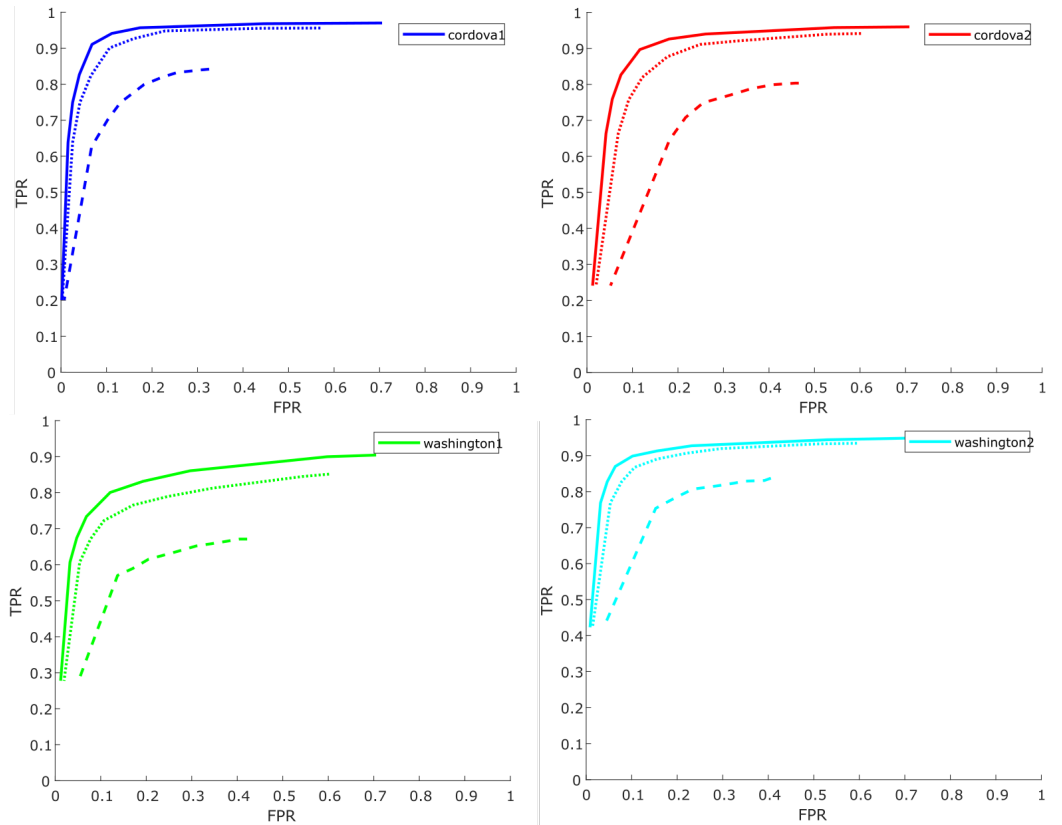


FIGURE 5.9 – ROC curves obtained with 10 DAs (dashed lines), 25 DAs (dotted lines), 40 DAs (plain lines) on each video set.

Figure 5.9 shows the ROC curves obtained. The experiments have been run with  $\tau_N = 0, 1, 3, 5, 10, 25, 50, 100,$  and  $500$  pixels. For a large object size threshold the true positive rate (TPR) and the false positive rate (FPR) are low and both increase when the object size threshold is reduced. Higher is the number of DAs used, better are the results as we can observe by comparing the area under the curve (AUC). This can be explained by the fact that a high number of DAs significantly increases the number of TN (DAs that did not converge), whereas the number of FP (DAs that converged, but are not validated) is limited and stabilized. The number of TP is also increased until all the lanes are detected (FN is null). Thus a high number of DAs reduces the FPR and increases the TPR.

However, in the context of our research, it is required to use the smallest number of not converging DAs to reduce the computational needs of the system. We choose to use the relation between the numbers of DAs  $n_{DA}$  and the object size threshold  $\tau_N$  to find the best amount of DAs for a better performance with less unused DAs. Indeed, the two parameters are linked by the number of sought objects in the image  $n_{obj}$ : when  $\tau_N$  is high, the number of sought objects is low and thus is the number of DAs needed; whereas when  $\tau_N$  is low, the number of sought objects is high and therefore the number of needed DAs too.

In order to find a relation that could rule our parameters for all the four clips of *caltech-lanes*, we proceed in two steps: first we looked for the relation between the objects size threshold and the number of sought objects in the image; then we linked this number to the number of DAs needed to detect them all.

### Number of object per image

In order to quickly estimate the relation between the number of sought objects and the objects size threshold, we use the results obtained by running sequentially 100 DAs over the video set *cordova1* with a threshold  $\tau_N = 0$ , so each of them detects an object in each image. The mean size of the objects detected per each DA, shown in Figure 5.10, gives a representation of the sought lanes in *caltech-lanes* images.

The objects of size inferior to 5 pixels can be considered as noise and ignored in the following analysis. The representation permits to roughly estimate the size of the lanes and thus the number of sought lanes depending on the objects size threshold.

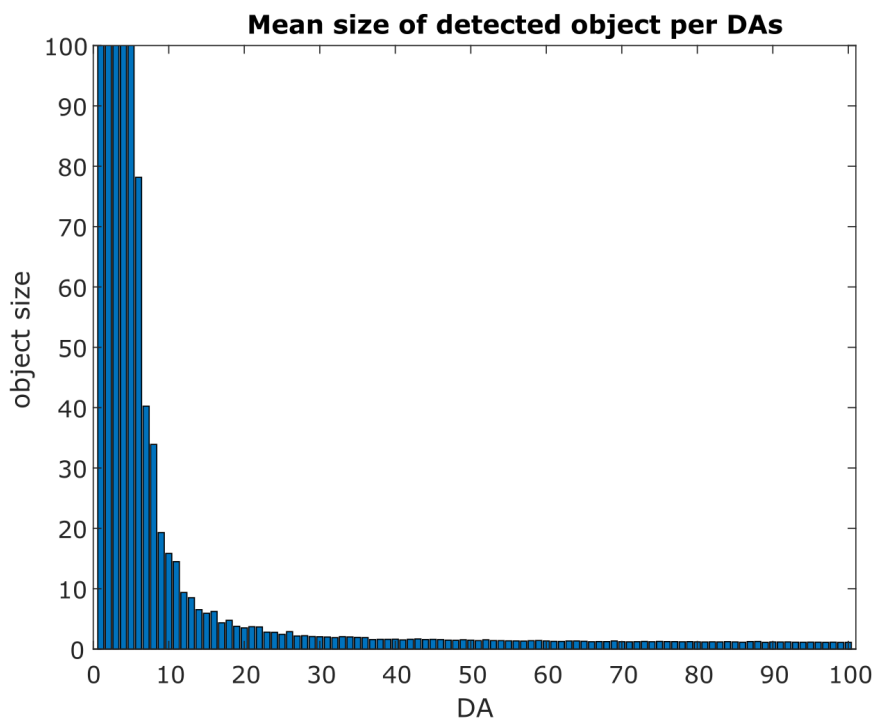


FIGURE 5.10 – Mean size of detected objects per each DA on the *cordova1* images set. This gives a representation of the lane sizes and number per image: there is about five to six big lanes per images (size above 50 pixels) and about 15 elements that could be part of a lane or a small lane (size above 5 pixels).

We observe that the number of sought lanes is proportional to a negative power of the objects size threshold. The regression on several power values  $\tau_N^{-2}$ ,  $\tau_N^{-1}$ ,  $\tau_N^{-1/2}$  and  $\tau_N^{-1/3}$ , visible on Figure 5.11, allows to estimate the following relation:

$$n_{obj} = \frac{1}{0.02\sqrt{\tau_N}}. \quad (5.12)$$



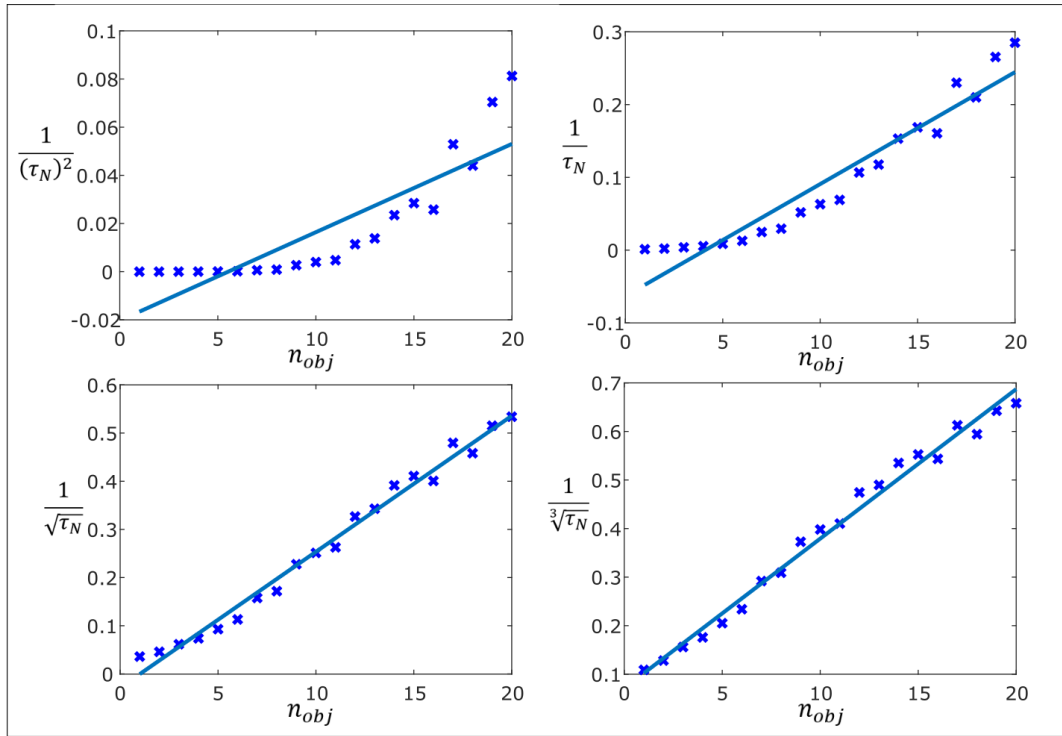


FIGURE 5.11 – Relation between the number of sought lanes and the objects size threshold for different power values.

The same experiment has been run on the other videos (*cordova2*, *washington1* and *washington2*) and confirmed the estimated relation. The four sets had their best regression value on  $\tau_N^{-1/2}$ , and an experimental coefficient between 0.022 and 0.029.

### Necessary number of DAs

The number of DAs needed is not necessarily equal to the number of sought objects. Even though the representation of the detected object for each DA (Fig. 5.10) shows that the first DAs have a tendency to detect bigger objects first, we could observe through each image result that this property is not always verified.

Figure 5.12 shows the necessary number of DAs to detect all sought lanes for different values of  $\tau_N$  -10, 20, 30, 50, 100 and 200pixels- and for each image of *cordova1*. This is based on the results obtained with 100 DAs and looking in each image for the last DA detecting an object of size superior to  $\tau_N$ . It permits to conclude that the number of necessary DAs is between one time and 2.5 times the number of sought objects.

The same results have been obtained when running this experiment on the other videos, which validates the estimated relation for the *caltech-lanes* datasets. The 2.5 factor can seem very high, but it is a consequence of our implementation using a sequential running of the DAs with no reinitialization. To obtain the best results, we chose the upper factor:

$$n_{DA} = 2.5 \times n_{obj} \quad (5.13)$$

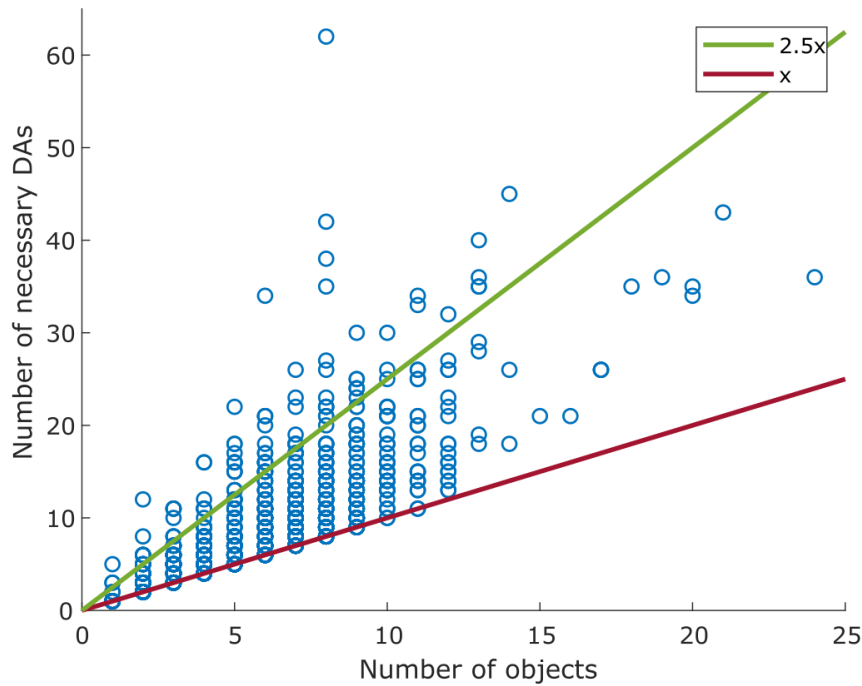


FIGURE 5.12 – Number of necessary DAs to detect all the sought lanes depending on the number of sought lanes in the image. Each circle corresponds to the result on one of the 250 images of *cordova1* and one object size threshold value (10, 20, 30, 50, 100 or 200).

This analysis permits to obtain the final relation estimation, which is presented in Equation 5.14, based only on one experiment with the following settings: 100 DAs run successively with a threshold at  $\tau_N = 0$ . The estimation is a little bit rough, but running only one scenario allowed us to optimize the computational time, which is not yet optimized for our CPU platform.

**Definition.** Relation between  $n_{DA}$  and  $\tau_N$

$$n_{DA} = \frac{2.5}{0.02\sqrt{\tau_N}} \quad (5.14)$$

### Comparison to the fixed number of DAs

The third set of experiments has been done using the automatic setting of  $n_{DA}$ , based on the estimated relation, on the four videos of *caltech-lanes*. Figure 5.13 shows the new ROC curves obtained in comparison to the previous curves. There is an improvement of the results, as it reduces the FPR and increases the TPR for small thresholds, while keeping stabilized results for high thresholds.

A shifting of the curves can be observed for the high values of the threshold. This can be explained by the small number of DAs used for high threshold. In fact, in images like cross road or pedestrian crossing lanes, the DAs might easily converge to one of the lane or road border which are not considered by the ground truth. Thus, the number of false positive is important compare to the total number of DAs.

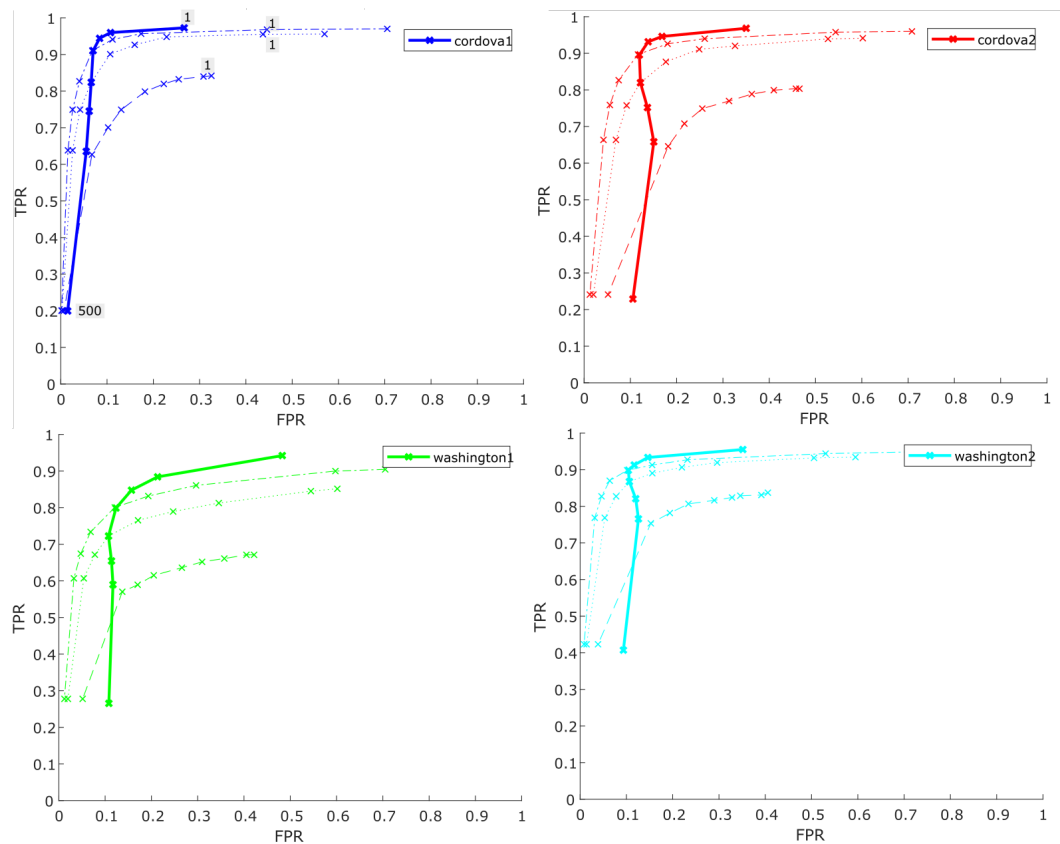


FIGURE 5.13 – ROC curves obtained with the automatic setting in comparison to the curves obtained with a fixed number of DAs on each video. Plain lines: Automatic setting; Dashed lines:  $n_{DA} = 10$ ; Dotted lines:  $n_{DA} = 25$ ; Dashed and dotted lines:  $n_{DA} = 40$ .  $\tau_N = 500, 100, 50, 25, 10, 5, 3, 1$  and 0 (the null value is not used for the automatic setting).

### Comparison to the reference

The inflexion point of each curve corresponds to the same setting:  $\tau_N = 10$  and  $n_{DA} = 39$ . Using those selected values, we compare the performances of our BIPS-based approach to the method used by Aly.

As shown in Table 5.1, the BIPS reaches the level of the reference. We can observe that the method performed better for the *cordova2* set, where there is sun facing and road work conditions, but performed worse for the *washington* sets, where shadows and street markings are more important.

Clip	correct rate		false pos. rate	
	[Aly]	[BIPS]	[Aly]	[BIPS]
cordova1	91.62%	91.06%	5.66%	6.99%
cordova2	85.5%	89.55%	40.64%	11.95%
washington1	92.78%	79.89%	13.11%	12.35%
washington2	93.66%	89.86%	8.59%	10.42%

TABLE 5.1 – Performance comparison between the method used by (Aly, 2008) and our BIPS-based method.

#### 5.4.3 Histogram feature thresholds impact

The experiments were all performed using fixed values of the histogram threshold parameters  $\tau_{ang} = 30\%$  and  $\tau_X = 50\%$ , that we empirically chose based on gradient images from *cordova1*. In the second step of our experimentation, we evaluate separately the impact of these two thresholds.

Figure 5.14 and Figure 5.15 show the variation of the FPR, the TPR and the number of detections depending on the value of the threshold  $\tau_{ang}$  first, then the threshold  $\tau_X$  for each video of the *caltech-lanes* dataset. The results have been obtained for  $\tau_{ang} = 0, 20, 30, 40, 50, 60, 80$  and 99 percent and for  $\tau_X = 0, 10, 20, 30, 40, 50, 60, 70, 80, 90$  and 99 percent. We observe that the results are not much influenced by the threshold in a large domain of values: 0.2 – 0.5 for  $\tau_{ang}$  and 0.4 – 0.7 for  $\tau_X$ .

As expected, a low value of the threshold allows noise or close lanes (either spatially or by angle) to be in the detection framing, thus reducing the number of detections and then the TPR and the FPR. And a high value of the threshold loses too much pixels from the detected lane leading to an insufficient object size for the detection, thus reducing the TPR and the FPR.

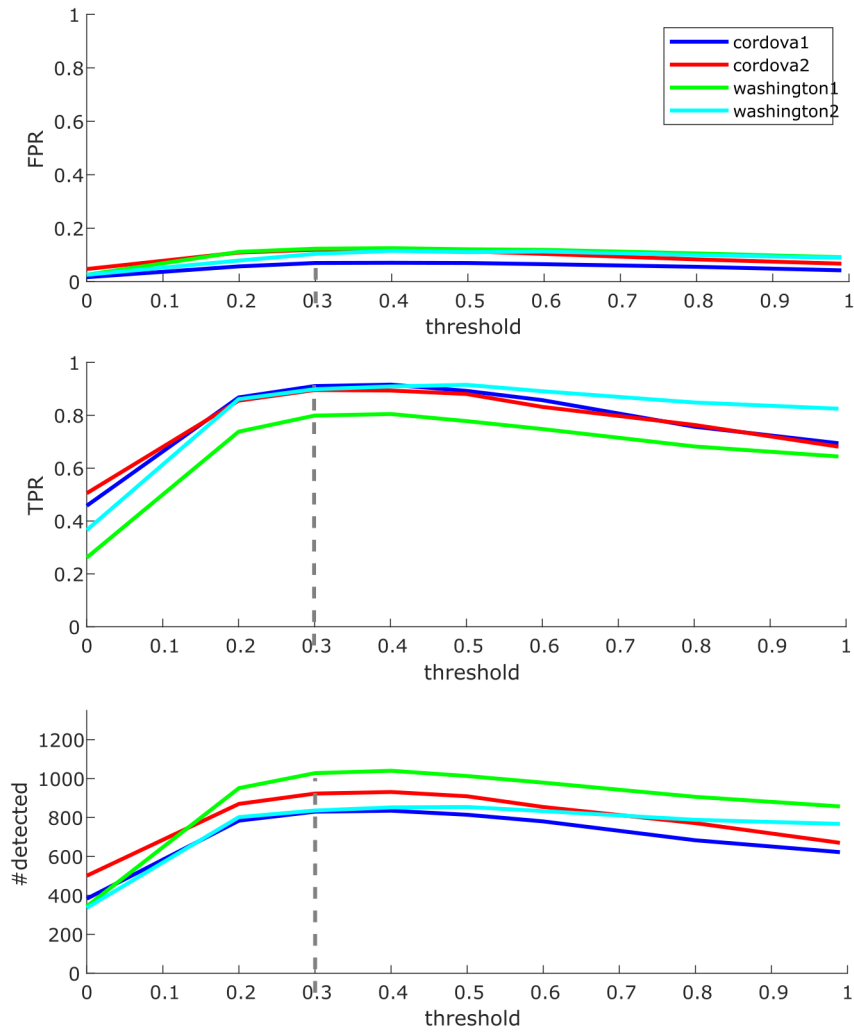


FIGURE 5.14 – Variation of the FPR, the TPR and the number of detected objects as a function of the threshold  $\tau_{ang}$ . The dashed line represents the fixed value used in previous settings.

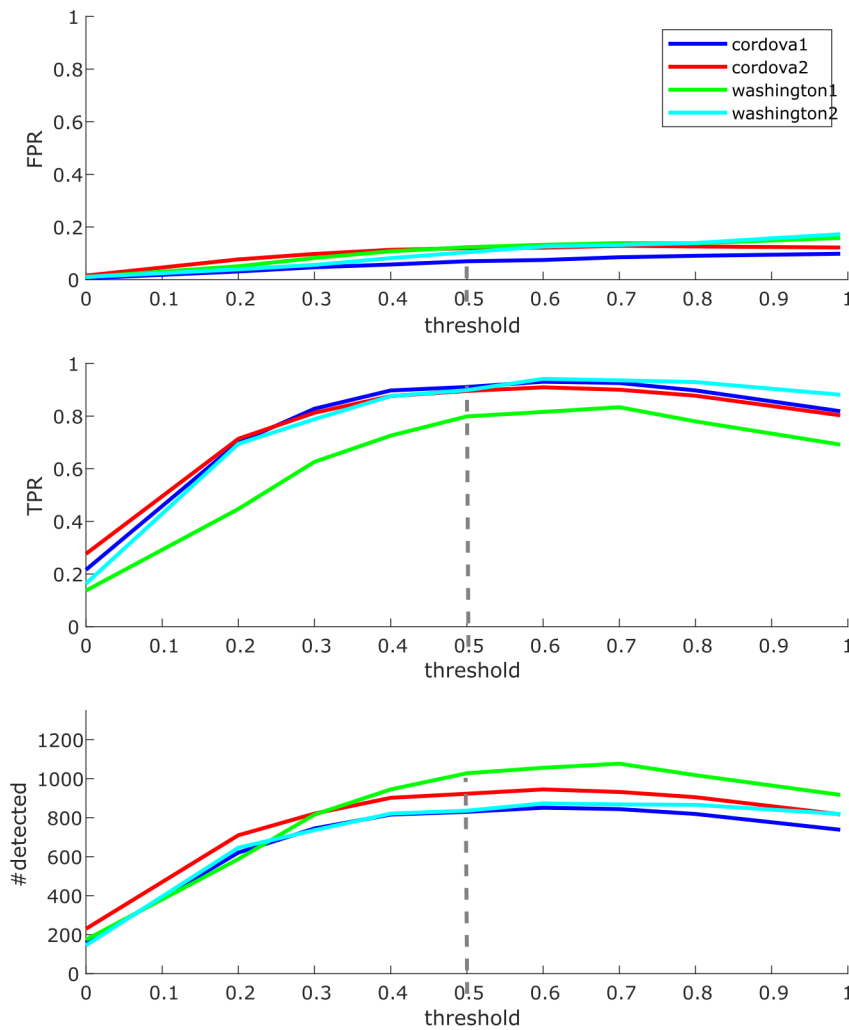


FIGURE 5.15 – Variation of the FPR, the TPR and the number of detected objects as a function of the threshold  $\tau_X$ . The dashed line represents the fixed value used in previous settings.

#### 5.4.4 Conclusion

The *caltech-lanes* dataset conditions allow to set several parameters of the method, leaving only four free: the number of Dynamic Attractors (DAs), the thresholds on the object size, the gradient angle histogram and the  $X$  coordinate histogram. The behavior and performance analysis of the method has been conducted in two steps. First, the impact of the number of DAs used has been evaluated, leading to the implementation of a size-dependent-selection of this parameter. The defined relation both automates the parametrization and improves the obtained results on the *caltech-lanes* database. This relation also optimizes the number of DAs engaged, thus minimizing the computation time and resources. In a second step, it has been verified that the estimated thresholds on the gradient angle and  $X$  histogram had not a strong impact on the results.

The BIPS results are at the level of the reference (Aly, 2008), however the rate of false positives remains high. In fact, in the ground truth the centered two lanes are considered as only one lane but they are mostly detected by two DAs, the ground

writings and pedestrian lanes are not taken into account but are often detected and the road borders are sometimes selected as lanes in the ground truth, sometimes not, but often correspond to the lane model chosen for the DA detection. The configuration of our architecture also has its impact on the true and false positive rate. The chosen model is focused on straight lanes leading to partial, multiple or nonexistent lanes detection. And the choice of a sequential running without reinitialization of the DAs reduces the number of detections and increases the false positive rate when a small number of DAs are used.

## 5.5 Discussion

In this Chapter we have adapted the BIPS approach to a specific application in order to evaluate its behavior and to compare its performance to known academic methods. All these proof of concept have been established with the BIPS SDK specifically developed for this PhD work and described in Chapter 4. It enables to understand the BIPS behavior and to evaluate the influence of all the parameters on its response and performance analysis. The road lane detection application has been chosen, as it corresponds to one of the industrial application of the BIPS on-chip and it is part of the autonomous driving hot research topic.

For this application most of the DAs architecture settings can be generalized and automatized, allowing to reduce this study to only four of the BIPS parameters: the number of DAs used and the object size, the gradient angle histogram and the  $X$  coordinate histogram thresholds. An extension of the BIPS method is however necessary, consisting in rotating the spatial axes during the iterative DA process to obtain a better framing of the detected lanes. The experimentation performed on the *caltech-lanes* datasets allowed to fit our BIPS-based approach for the straight lines model and to validate the weak influence of the histograms thresholds on the road lane detection application. Moreover, it permitted us to establish an automatic setting of the number of DAs allowing computation and resources optimization. The obtained results are promising as they reached the level of the reference (Aly, 2008), while relying only on the BIPS method without supplementary pre- or post-processings.

Our first implementation of the BIPS approach has, however, its flaws, that could be improved. Firstly, the DAs sequential running without reinitialization causes a high increase of  $n_{DA}$ , thereby increasing the computational cost. Secondly, the multi-modal capacity of the BIPS, which we could formalize through the input definition (Sec. 3.2.4), has not yet be taken into account. The method could be adapted for more complex models than the straight lines model and could integrate more perceptual inputs (like LIDAR, IMU). The rotated spatial features selection could even be done in a more generic way and on a larger set of features. Thirdly, the time information could be integrated into the process to improve the detection results, to track the detected objects and to reduce the computational time of the BIPS by pre-selecting the region of interest where each DA is likely to converge. The ground-truth of the *caltech-lane* dataset has its flaws too, as the team of Aly focused more on the right and left first lanes leaving the labelization of the other lanes inconsistent. It would be then interesting to establish a well defined labelization of this database completed by other datasets with different road conditions and to compare the performances of the BIPS-approach to more known road lane methods tuned for the chosen labelization rules.

## Chapter 6

# Extension proposals

### 6.1 Introduction

The study of the BIPS method in the specific application of road lane detection presented in Chapter 5 allowed to confirm its convergence behavior toward objects of interests and its at-the-reference-level performances when its settings are adapted in a top-down manner (i.e. based on a model of the objects). However, the study failed to bring out an answer concerning its generic and evolutive capacities, neither its efficiency in other applications. The real-time computation property might be broken by the multiple sequential processes of the method. Moreover, it seems complicated to automatically tune all the parameters to get the expected BIPS result. Mainly for this reason, it was decided to develop a software simulator. Even if this approach is far from the BIPS real time aspects, it enabled a precise comprehension of the different processing units of the BIPS and their cross-interaction. It can now also help in the improvement and extension of its functionalities.

The biological model, on which the BIPS concept is based, supposes that the association of the same process working on the few sets of first and second order features (e.g. brightness, color, edges) computed in the first layers of the visual pathway is enough to perform detection and recognition tasks in the brain (see Chapter 2). It implies that the right combination of FMD processes on a set of first and second order features should allow us to perform different and more complex visual tasks. The experimentation and evaluation of this hypothesis is not straightforward as our understanding of the FMD processes combination effect must be further investigated. As a first step toward it, we can already propose some extensions of the DA process, which will open the method to a larger set of applications.

In this Chapter, we present two DA extension proposals. The first one, presented in Section 6.2 permits to reduce computational time, to improve detection results and to perform object tracking by integrating time into the process. We suggest an improvement of the method using a Kalman filter to perform a more robust and smooth object tracking while reducing the computational time of the DA process. The second proposal, presented in Section 6.3 permits to obtain a better framing of the objects and opens the method to automatic and evolutive selection of the input features. The proposal is inspired from the spatial axes rotation used in the road lane detection application for improvement of the framing. Same issue happens in cases where the gradient angle can not be used to find the new axes. Thus, we propose an integration of the principal directions of the sought objects into the DAs to permit a reorientation of their input feature space. Finally, the contribution of these first proposals and their limitations will be discussed in Section 6.4.



## 6.2 Time integration proposal

In (Pirim, 2013), the presentation of the Dynamic Attractor (DA) process mentioned the use of the median value of features in previous images to estimate the displacement of the object framing  $P$  and predict it for the next image. In this Section, we study how time can, in fact, be integrated into the BIPS method and which improvements this extension brings in terms of computation and applications.

Section 6.2.1 presents a quick overview of the existing tracking methods and the advantages time integration could bring to the classical BIPS method. Then, our time integration proposal, using the Kalman filter, is presented in Section 6.2.2. Illustration and first analysis of the BIPS method integrating our extension has been run on the *view001* video from PETS 2009 benchmark (*PETS 2009*) and is presented in Section 6.2.3. The obtained results comfort us in the BIPS capacity to obtain better detection and tracking information in a limited number of iterations, thanks to the smoothed and robust process of the Kalman filter. But on the other hand, as discussed in Section 6.2.4, it also increases number of parameters, which complicates the generic and evolutive implementation of the BIPS concept.

### 6.2.1 Time integration overview

#### Time integration in the BIPS

The proposed road lane detection framework of (Bar Hillel et al., 2014), followed in Section 5.2.2, presents a fourth block named "Time integration". Its purpose is to correct the detection using the previous frames information. In our formalization of the BIPS, we did not integrate this information yet, but this has been considered by the inventor P. Pirim, when he proposed to use the median displacement to shift the new boundaries  $(A_{new}, B_{new})$  (Pirim, 2013). The idea is to predict the sought object framing  $P_{t+1}$  based on the framing  $P_{t,k_{final}}$  (Eq. 3.29) obtained at the end of the iterative process (or when it is stopped) and uses this predicted framing to accelerate the DA convergence process on the next image.

Such time information can improve the BIPS in several ways. Firstly, as it was observed in Section 3.4.2, the correspondence between a DA and an object is not consistent from one image to another in the classical method, but time integration could help maintain a DA on the same object, thus enlarging the domain of application of the BIPS to tracking applications. Secondly, the DAs linked by inhibition can not completely work in a parallel way, as each of them have to wait for the preceding DAs to leave them some pixels in the frame to work on. Moreover, the order of detection is strongly influenced by the feature order and does not necessarily correspond to the order of importance of the object, as we could observe in the experiments Section 5.4.2. But, with time integration, it can be expected that the DAs will be directly launched on different well chosen areas of the feature space, based on the previous detection. Computation is thus also improved as the number of iterations in the DA process can be strongly reduced.

#### Existing methods

There is a multitude of tracking methods that could be used to perform the new framing  $P_{t+1,0}$  prediction. The survey of Yilmaz et al. (Yilmaz, Javed, and Shah, 2006) gives an overview and a useful classification of the tracking methods, even though most of them are more concentrated in finding the path of the object rather than

predicting its next position. The tracking implementation of the framing belongs to the *point tracker* family, as its position is computed at each frame by the DA process and its representation corresponds to a set of points. Among the gathered methods for this class, the most appropriated are the statistical methods (Kalman filter and Particle filter), which are more robust to occlusions and model uncertainties than the deterministic methods. They are less complex than the probabilistic methods. Bar Hillel et al. also pointed out those two methods as the two most used methods for time integration in road lane detection.

For our time integration extension, we choose the Kalman filter as it showed several interesting results in robotic application (Chen, 2012), it is well-known and easily implemented in its linear version.

## 6.2.2 Method proposal

### Validation framing enlargement

First and foremost, we use an enlargement of the detected framing  $P$  for the pixel validation (Eq. 3.21) allowing the DAs to look for bigger objects when their predicted framing initialization is too small. The enlargement is integrated to the iterative process and allows also to reconstruct framing that had been cut due to the inhibition and the parallel running of DAs, as mentioned in Section 3.4.2. However, this enlargement might alter the DA convergence property (see App. ??). The principle is to set manually or automatically an offset  $\varepsilon_F$  for each feature  $F$ , such that the pixels validated for the histogram computation are in the enlarged framing  $P_{t,k}^\varepsilon$  as defined below.

**Definition.** Enlarged framing

$$P_{t,k}^\varepsilon = \prod_{m=1}^M \left[ A_m^{t,k} - \varepsilon_m, B_m^{t,k} + \varepsilon_m \right] \quad (6.1)$$

### Chosen model

The Kalman filter is used in each DA to update and estimate the framing  $P$  described by its center  $\mathbf{C} = (C_m)_{m \in \llbracket 1, M \rrbracket}$  and its widths  $(\Delta_m)_{m \in \llbracket 1, M \rrbracket}$ .

For this first extension trial of the time integration into the BIPS, the chosen model of the framing displacement is the translation at constant speed  $(v_m)_{m \in \llbracket 1, M \rrbracket}$ , and constant widths. The corresponding transition matrix is defined by Eq. 6.4. The observation of the state is based on the obtained boundaries at the end of the iterative process (or when it is stopped), the corresponding observation matrix is defined by Eq. 6.5. The Kalman filter linear equations used are reminded in Eq. 6.6 for the definition part, Eq. 6.7 for the update part and Eq. 6.8 for the prediction part. Finally, the update framing is used to re-calibrate the DA output and the predicted framing is used as initialization of the DA iterative process on the next image. A flowchart representing the integration of the Kalman filters into the BIPS process is presented in Figure 6.1.

**Definition.** State vector

$$\mathbf{x} = (C_1, \dots, C_M, \Delta_1, \dots, \Delta_M, v_1, \dots, v_M) \quad (6.2)$$

**Definition.** Measure vector

$$z_t = \left( A_1^{t,k_{final}}, \dots, B_M^{t,k_{final}}, B_1^{t,k_{final}}, \dots, B_M^{t,k_{final}} \right) \quad (6.3)$$

**Definition.** State transition matrix

$$F = \begin{pmatrix} I_M & 0 & I_M \\ 0 & I_M & 0 \\ 0 & 0 & I_M \end{pmatrix} \quad (6.4)$$

**Definition.** Observation matrix

$$H = \begin{pmatrix} I_M & -\frac{1}{2}I_M & 0 \\ I_M & +\frac{1}{2}I_M & 0 \end{pmatrix} \quad (6.5)$$

**Definition.** Kalman definitions

$$\begin{aligned} \hat{x}_{t|t} &: \text{a posteriori state estimate} \\ \hat{x}_{t+1|t} &: \text{a priori state estimate} \\ z_t &: \text{measurement} \\ P_{t|t} &: \text{a posteriori error covariance matrix} \\ P_{t+1|t} &: \text{a priori error covariance matrix} \\ F_t &: \text{state transition model} \\ H_t &: \text{observation model} \\ Q_t &: \text{covariance of the transition noise} \\ R_t &: \text{covariance of the observation noise} \\ K_t &: \text{gain} \\ I &: \text{identity matrix} \end{aligned} \quad (6.6)$$

**Definition.** Kalman update

$$\begin{aligned} K_t &= P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \\ \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t (z_t - H_t \hat{x}_{t|t-1}) \\ P_{t|t} &= (I - K_t H_t) P_{t|t-1} \end{aligned} \quad (6.7)$$

**Definition.** Kalman prediction

$$\begin{aligned} \hat{x}_{t+1|t} &= F_t \hat{x}_{t|t} \\ P_{t+1|t} &= F_t P_{t|t} F_t^T + Q_t \end{aligned} \quad (6.8)$$

**Algorithm step.** Update  $t, k_{final}$

$$\begin{aligned} \forall m \in \llbracket 1, M \rrbracket \quad A_m^t &= \hat{C}_{t|t} - \frac{\hat{\Delta}_{t|t}}{2} \\ B_m^t &= \hat{C}_{t|t} + \frac{\hat{\Delta}_{t|t}}{2} \end{aligned} \quad (6.9)$$

**Algorithm step.** Prediction  $t \rightarrow t+1, k=0$

$$\begin{aligned} \forall m \in \llbracket 1, M \rrbracket \quad A_m^{t+1,0} &= \hat{C}_{t+1|t} - \frac{\hat{\Delta}_{t+1|t}}{2} \\ B_m^{t+1,0} &= \hat{C}_{t+1|t} + \frac{\hat{\Delta}_{t+1|t}}{2} \end{aligned} \quad (6.10)$$

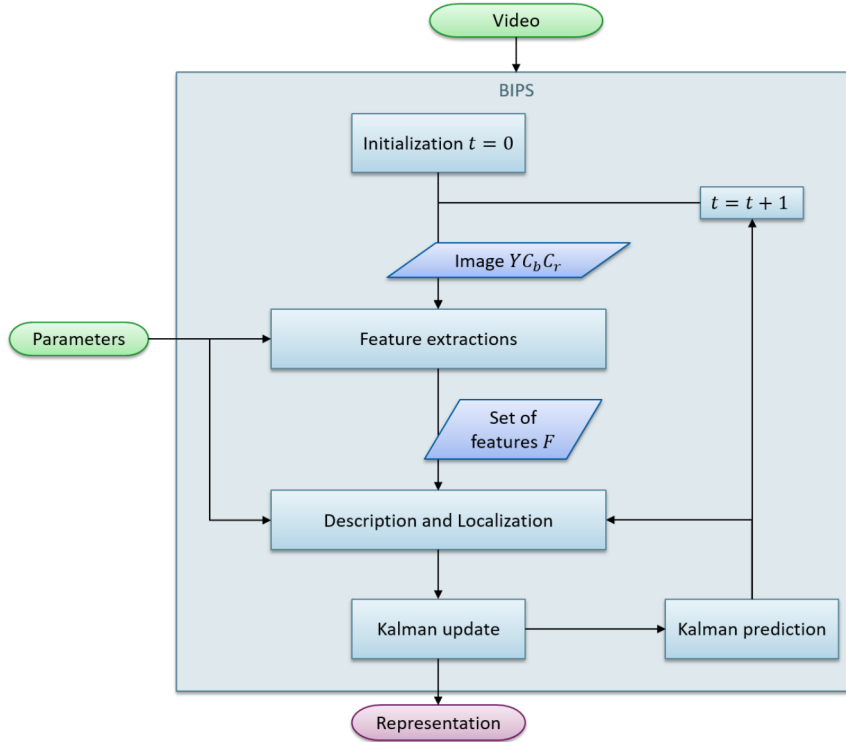


FIGURE 6.1 – Representation of the BIPS process when using a Kalman filter.

### 6.2.3 Illustration

We illustrated the proposed method and initiated a comparison to the medians displacement shifting proposed by the author on the video *view001* from the PETS 2009 benchmark (*PETS 2009*). The video presents moving pedestrians with a static point of view. The experiments have been done with our BIPS simulator presented in Chapter 4 and the display was made with Matlab (*MATLAB and Image Processing Toolbox Release R2018a*).

We ran only one DA to not be disturbed by the chosen inhibition rules. The optical flow direction is used as the descriptive feature and the Cartesian spatial axes  $X$  and  $Y$  as space features. We use the optical flow obtained with the Farnebäck method (Farnebäck, 2003) and the function developed for OpenCV (*OpenCV 2.4 release*), with a threshold  $\tau_{velo} = 2$  pixels/frame for the validation of the direction value. The histogram thresholds are set to  $\tau_{dir} = 0.10 \times q_{max}^{dir}$ ,  $\tau_X = 0.20 \times q_{max}^X$  and  $\tau_Y = 0.10 \times q_{max}^Y$ . The mode 1 is used for all FMDs and the object size threshold is set to  $\tau_N = 100$  pixels. Those values have been chosen based on the observation of the features extraction results. The threshold on  $Y$  is kept low to avoid the cut of the heads and legs of the tracked pedestrian.

Four configurations are illustrated: (1) no tracking is used and the DA is reinitialized for each frame; (2) the obtained framing is reused with enlargement; (3) the medians displacement is used to predict movement of the framing between two images; (4) our extension proposal: the Kalman filter updates and predicts the framing between two images.

### 1. Without tracking

In this first configuration, the DA is reinitialized for each frame and the classical BIPS method is used. In Figure 6.2, we can observe that from one image to another the DA does not detect the same person.

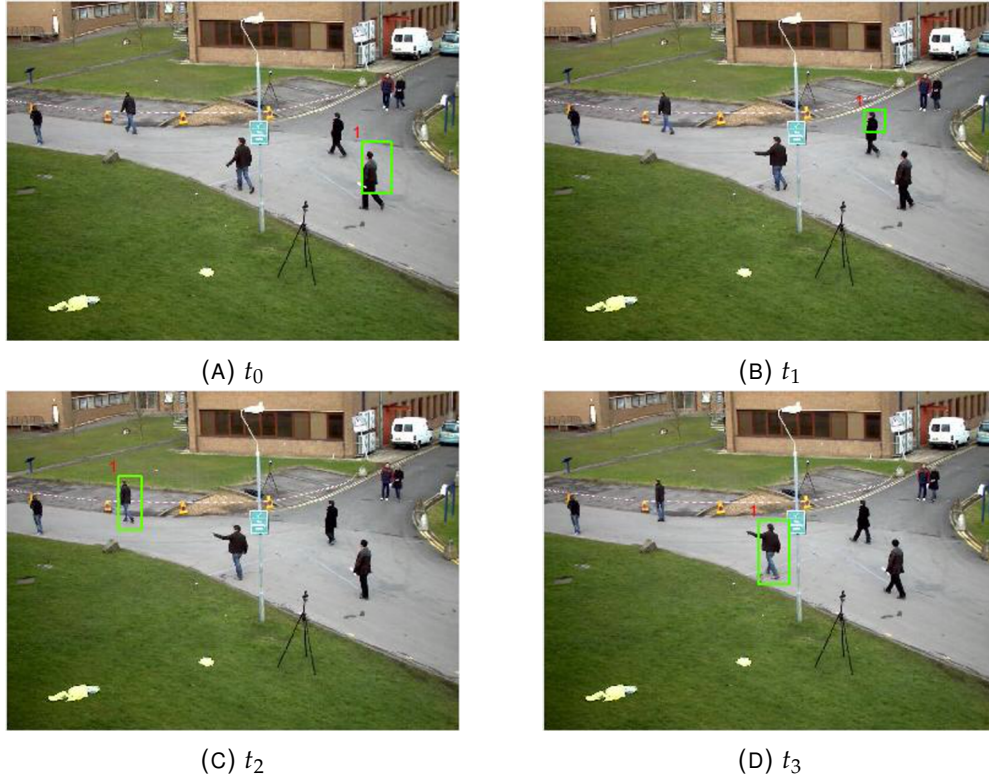


FIGURE 6.2 – DA detection on successive images without tracking. The detection performed by the first DA (direction-related-colored framing) is not maintained on the same person along the four successive frames.

## 2. Using the enlarged framing

In this configuration, the boundaries are directly reused for the next image with an enlarged framing of  $\varepsilon_{dir} = 10^\circ$ ,  $\varepsilon_X = 10$  pixels and  $\varepsilon_Y = 10$  pixels. If we look at the same four frames sequence, we can observe that the enlargement helps maintaining the DA on the same person (Fig. 6.3). But also, as shown in Figure 6.4, this enlargement can cause a change of detection, when two persons are close to each other.

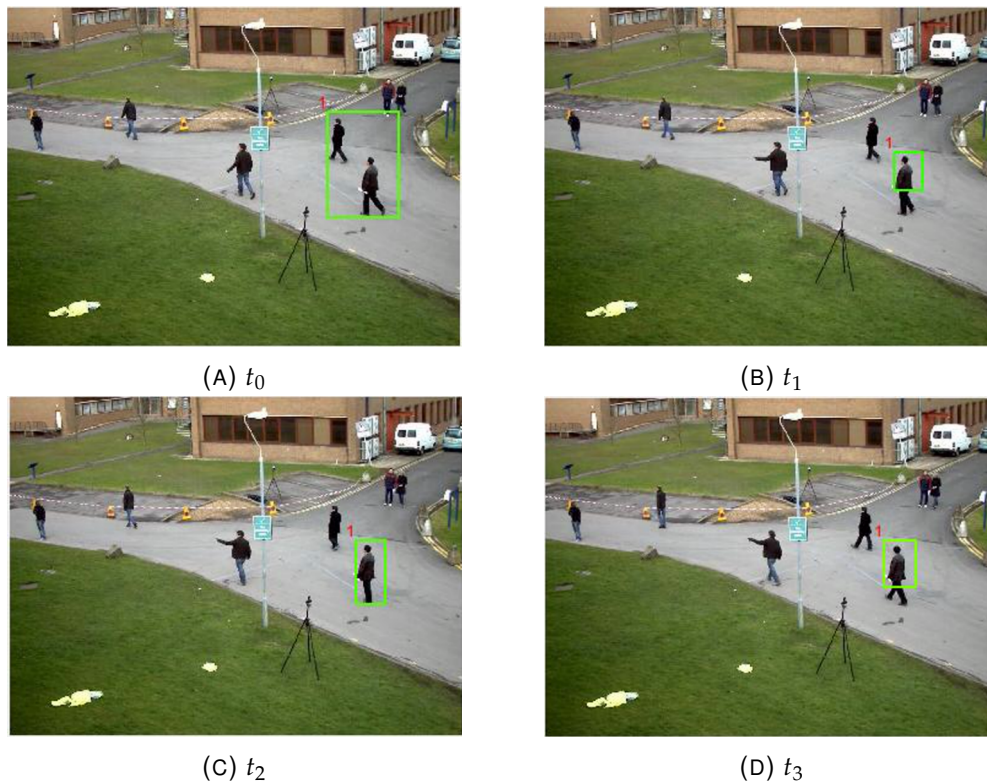


FIGURE 6.3 – DA detection on successive images with an enlarged framing tracking. The detection performed by the first DA (direction-related-colored framing) is now maintained on the same person along the four successive frames.

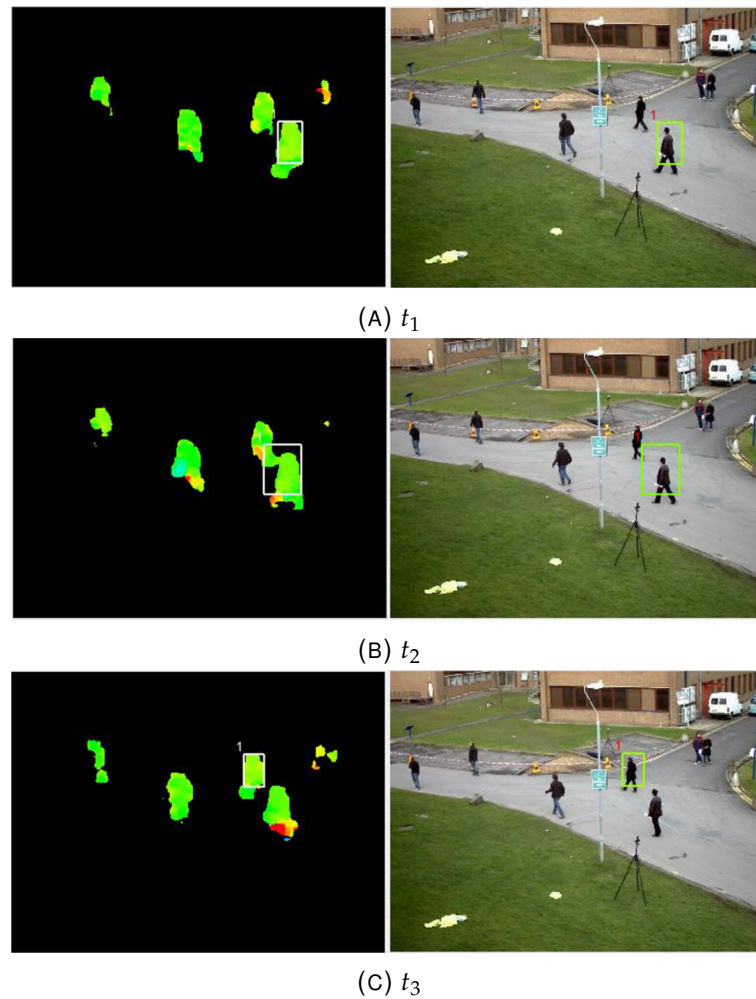


FIGURE 6.4 – Switched detection observed when two pedestrians are close on three successive images with an enlarged framing tracking. Left: direction extraction - white framing for detection; Right: image - direction-related-colored framing for detection.

### 3. Using the Median

This configuration, based on the proposal of P. Pirim (Pirim, 2013), uses the displacement of the different feature medians to shift the boundaries. The enlarge framing defined above is still used to avoid a reinitialization of the DA due to a too small predicted framing. The displacement is initialized to zero after the first detection of the DA, then it corresponds to the difference between the current detection medians and the preceding images detection medians.

The method permits to maintain the DA on the same person, but it can be observed that the predicted framing is often too much shifted. (Fig. 6.5).

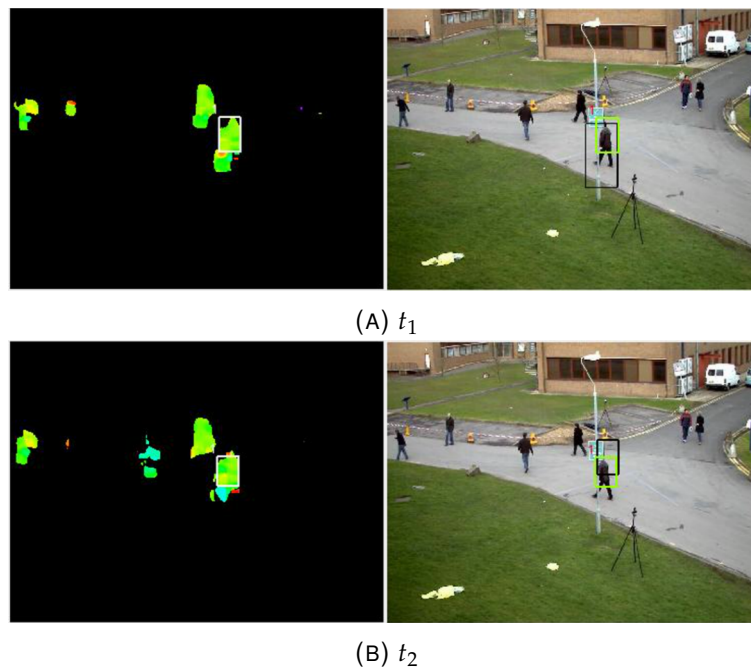


FIGURE 6.5 – Shifted predicted framing observed on two successive images with a tracking using the medians. It is due to the strong variation of the features. Left: direction extraction - white framing for detection; Right: image - direction-related-colored framing for detection - black framing for prediction.



#### 4. Using the Kalman filter

This configuration, based on our proposal, uses the Kalman filter. The covariance matrices of noise are to be set according to the chosen application and video. The model noises are supposed Gaussian and independent with a standard-deviation of  $1^\circ$  for the angle center, width and displacement between two images; and 1 pixel for the position center, framing widths and axes displacements between two images. The measure noises are also supposed Gaussian and with a standard deviation of  $10^\circ$ . The enlarge framing defined above is still used to avoid a reinitialization of the DA due to a too small predicted framing. Initialization of the values is performed at the first detection of the DA, then corresponds to the Kalman outputs.

The method maintains the detection of one person even when it passes behind the pole, which was not the case for the two previous configurations (Fig. 6.6). The detection seems smoother and more consistent, as we expected. The number of reinitialization of the DA is also reduced (Tab. 6.1). However it can still be observed that two close persons can deviate the detection (Fig 6.7).

Method	Number of DA reinitialization
No tracking	91
Enlarged framing	3
Using the medians	8
Using Kalman	2

TABLE 6.1 – Number of DA reinitialization along the sequence of 91 images from PETS 2009 video *view001*.

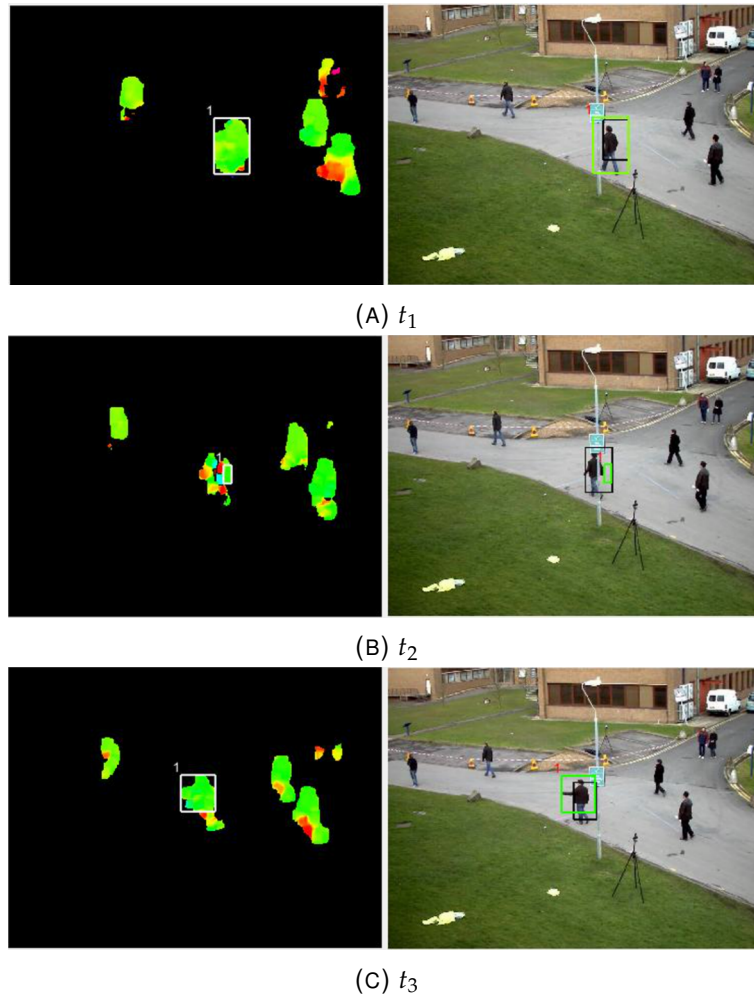


FIGURE 6.6 – Smoothed tracking of a person walking behind the pole on three successive images with a tracking using the Kalman filter. Left: direction extraction - white framing for detection; Right: image - direction-related-colored framing for detection - black framing for prediction.

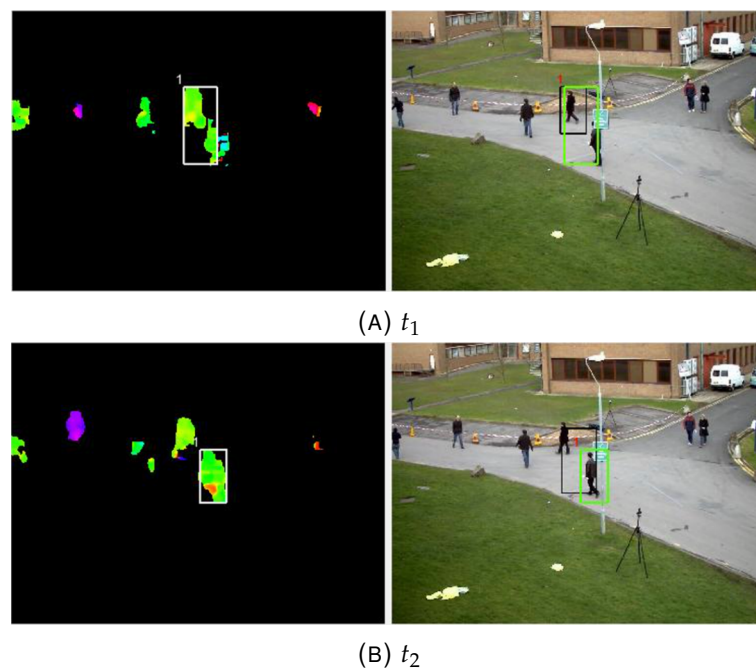


FIGURE 6.7 – Switched detection observed when two persons are close on two successive images with a tracking using the Kalman filter. Left: direction extraction - white framing for detection; Right: image - direction-related-colored framing for detection - black framing for prediction.

### Number of iterations

A look at the number of iterations for a DA to converge shows also how the smoothed tracking of the Kalman filter accelerates the convergence time compared to only the enlarge framing used, whereas the medians tracking slows it down (Tab. 6.2 and 6.3). Moreover, when limiting the number of iterations per image to only three iterations (i.e. one iteration per feature), whereas the medians methods can not keep track of the target at all, the Kalman filter gives similar results as when waiting for the convergence of the DA (Fig. 6.8 and Table 6.4). This allows to strongly reduce the number of iterations per image and per DA, thus the computational cost of the method.

Method	Mean number of iterations
No tracking	9.87
Enlarged framing	11.82
Using the medians	12.93
Using Kalman	10.91

TABLE 6.2 – Mean number of iterations of the DA on a sequence of the video *view001* from (*PETS 2009*).

Method	7 iterations	10 iterations	13 iterations	16 iterations
No tracking	0%	40.7%	91.2%	100%
Enlarged framing	4.4%	57.1%	73.6%	76.9%
Using the medians	2.2%	44.0%	63.7%	71.4%
Using Kalman	1.1%	49.4%	76.9%	84.6%

TABLE 6.3 – Percentage of convergence per number of iterations. Results obtained when running one DA in a sequence of the video *view001* from (*PETS 2009*).

Method	Number of DA reinitialization
No tracking	91
Enlarged framing	4
Using the medians	58
Using Kalman	2

TABLE 6.4 – Number of DA reinitializations along the sequence of 91 images from *PETS 2009* video *view001* when stopping the iterative process at three iterations.

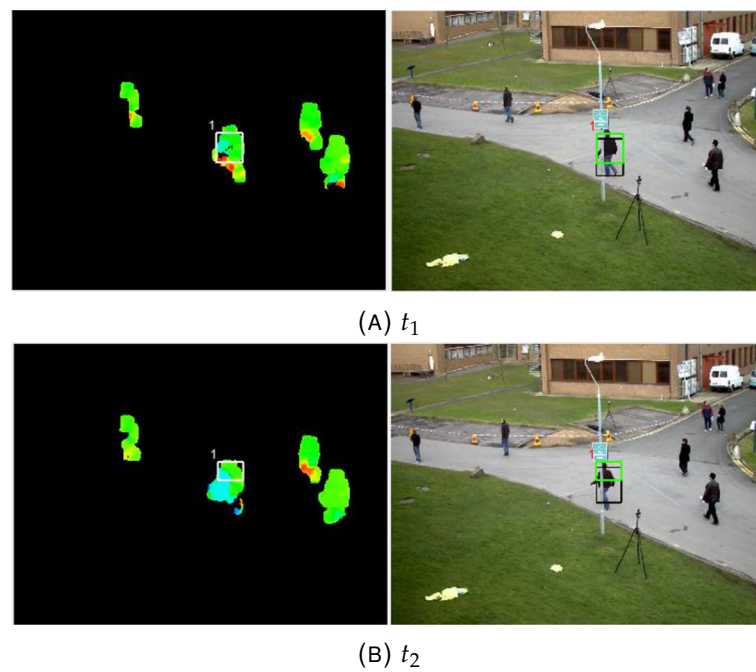


FIGURE 6.8 – Tracking results on successive images obtained with the Kalman filter when stopping the iterative process after three iterations. Left: direction extraction - white framing for detection; Right: image - direction-related-colored framing for detection - black framing for prediction.

### 6.2.4 Conclusion

In this Section, we gave a first look at the possible time integration extensions of the BIPS. The purposes are to maintain the association DA-sought object, to reduce the number of iterations specially when using multiple DAs and to enlarge the BIPS range of applications. An implementation of the Kalman filter into the BIPS process has been proposed. It allows to correct and predict the sought object framing detected by the DA. Thus, the detection results are improved, each DA keeps track of its detected object and the computational cost can be reduced.

The preliminary experiments made on the video *view001* from (*PETS 2009*) encourages our choice as the obtained results give a smoothed track of the sought objects even with limited number of iterations, which is maintained even in the occlusion test. It also encourages us in keeping a statistical tracking method, as the strategy proposed by the author P. Pirim seems strongly affected by the descriptive feature artifacts and even more by the limited number of iterations. However, the proposed time integration brings even more parameters into the BIPS method: the enlargement values, the number of iterations per image and the noise covariance matrices of the Kalman filter. Their settings, which have been made manually for our experiment, is dependent to the chosen application and the chosen dataset. The automation of their value for a more generic implementation of the BIPS concept implies therefore supplementary internal processes, which could increase the computational needs. Moreover, it is not certain that such automation is possible.

## 6.3 Feature space reorientation

In the context of road lane detection, improvement of the results were obtained using rotation of the spatial axes during the DA iterative process. This extension of the method allowed to find a better spatial framing of the sought objects (Sec. 5.3.1). The issue was that, even if the selection of pixels in the feature space corresponds well to the object, the projection of the framing into the spatial coordinates could give an inadequate localization of the object bounding box. Moreover, in the case where connections between DAs were only based on the spatial framing, this would inhibit other lanes to be detected. The same issue exists for objects which are not described by a unique gradient angle, in those cases the same method for rotating the axes can not work.

This Section presents our second extension proposal, which extends the axes rotation solution to generic cases. The objective is to rotate the DA input features during its iterative process so that the obtained framing better fits the object it is converging to. The proposal is based on the covariance of the validated pixels data. As the number of features used should not be much increased (3 features currently in our tests), the computational needs of such extension should respects the constraints of autonomous embedded systems. Section 6.3.1 presents the algorithm and modifications of the DA iterative process for the computation of the axes and the reorientation of the feature space. Then, Section 6.3.2 gives a visualization of the extension effect by experimenting it on an example image. The obtained results tends to confirm the improvement of the detections with a closer framing. Our proposed rotation hence allows the BIPS method to perform in a wider set of applications and gives a tool for a more versatile and evolving implementation of the BIPS method as discussed in Section 6.3.3.

### 6.3.1 Reorientation proposal

#### Algorithm

In a DA working on the feature set  $F_1, \dots, F_M$ , we define a first set of objects defining features  $F_1, \dots, F_{m_0-1}$  and the reorientation will be performed in a subset  $F_{m_0}, \dots, F_M$ . A vector unit  $\mathbf{u}_m$  is associated to each FMD and updated at each iteration in order to change and adapt the features on which the DA is working. The pixel validation (Eq. 6.11) and the conditioned histogram computation (Eq. 6.12) are slightly modified as the feature value used becomes the projected value of the feature vector  $F$  on the vector unit  $\mathbf{u}_m$ . Similarly to the space rotation, in order to keep consistency in the set of pixels surrounded, the pixel validation is performed using the memorized vector ( $\mathbf{u}_{FMD_m}$ ) corresponding to the last boundary update, whereas the one used for the histogram is the new one  $\mathbf{u}_m^{t,k}$ .

The principal component analysis is performed on the validated pixels data, more specifically on their value in the feature sub-space formed by  $(F_{m_0}, \dots, F_M)$ . The covariance matrix is estimated at each iterative step using the feature values of the validated pixels (Eq. 6.13, 6.14, 6.15, 6.16, 6.17). This can be performed on the fly during the histogram computation. The eigen vectors  $(\mathbf{e}_m)_{m \in \llbracket 1, M_c \rrbracket}$  of the covariance matrix give the main directions of the selected object and then the new axes of the feature space (Eq. 6.18 and 6.19). The iterative process of the DA is then modified and must integrate the computation of the covariance and the eigen vectors, then the memorization and update of the axes. The modified DA iterative process is presented in Figure 6.9.

**Definition.** FMD pixel validation

$$\begin{aligned} val_{FMD_m}^k : \mathbb{P} &\longrightarrow 0, 1 \\ p &\longmapsto val_{FMD_m}^k(p) = (A_m^k \leq \langle F(p), \mathbf{u}_{FMD_m} \rangle \leq B_m^k) \end{aligned} \quad (6.11)$$

**Definition.** Conditioned feature histogram

$$\begin{aligned} H_n^k : \Omega &\longrightarrow \mathbb{N} \\ f &\longmapsto H_n^k(f) = \mathbf{card} \left\{ p \in \mathbb{P} \left/ \begin{array}{l} val_{ext}^k(p) = 1, \\ \langle F(p), \mathbf{u}_n^k \rangle = f, \\ \forall m \in \llbracket 1, M \rrbracket, \\ val_{FMD_m}^k(p) = 1 \end{array} \right. \right\} \end{aligned} \quad (6.12)$$

**Definition.** Validated pixel set

$$\mathcal{O}_k = \left( \bigcap_{m=1}^M (val_{FMD_m}^k)^{-1}(1) \right) \cap (val_{ext}^k)^{-1}(1) \quad (6.13)$$

$$\begin{aligned} \forall m \in \llbracket 1, M \rrbracket, \forall k \in \mathbb{N}, \\ N_k &= \sum_{f \in \Omega} H_m^k(f) \\ &= \mathbf{card}(\mathcal{O}_k) \end{aligned} \quad (6.14)$$

**Definition.** Sample for mean and covariance estimation

$$s(p) = (F_{m_0}(p) \cdots F_M(p)) \quad (6.15)$$

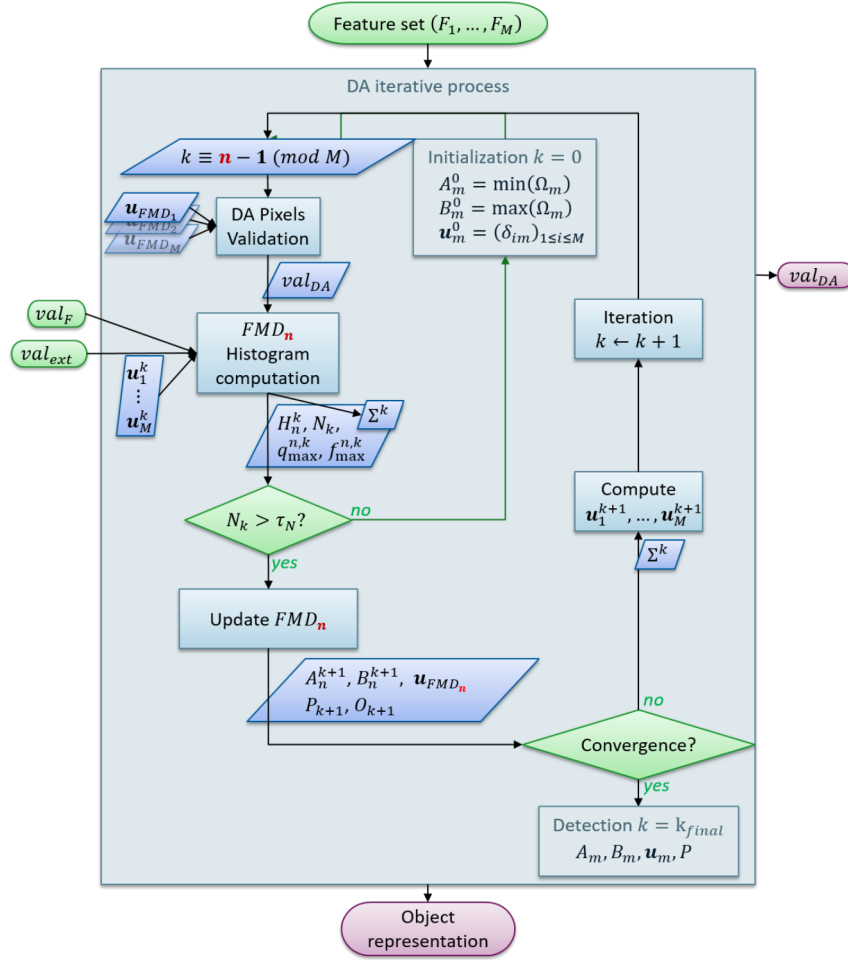


FIGURE 6.9 – Representation of the DA iterative process with the multidimensional feature space rotation extension.

**Definition.** Mean vector estimation

$$\mu_k = \frac{1}{N_k} \sum_{p \in \mathcal{O}_k} s(p) \quad (6.16)$$

**Definition.** Covariance matrix estimation

$$\Sigma_k = \frac{1}{N_k} \left( \sum_{p \in \mathcal{O}_k} s(p)^T s(p) \right) - \mu_k^T \mu_k \quad (6.17)$$

**Algorithm step.** Feature space axes: Initialization with the canonical basis

$$\forall m \in \llbracket 1, M \rrbracket, \quad \mathbf{u}_m^0 \in \mathbb{R}^M \quad (6.18)$$

$$\forall n \in \llbracket 1, M \rrbracket, \quad \mathbf{u}_m^0(n) = \delta_{nm}$$



**Algorithm step.** Feature spaces axes: Update of the rotated axes

$$\begin{aligned}
& \forall k > 0, \forall m \in \llbracket 1, m_0 - 1 \rrbracket & \mathbf{u}_m^k \in \mathbb{R}^M \\
& & \forall n \in \llbracket 1, M \rrbracket, \quad \mathbf{u}_m^k(n) = \delta_{nm} \\
& \forall k > 0, \forall m \in \llbracket m_0, M \rrbracket & \mathbf{u}_m^k \in \mathbb{R}^M \\
& & \forall n \in \llbracket 1, m_0 - 1 \rrbracket, \quad \mathbf{u}_m^k(n) = 0 \\
& & \forall n \in \llbracket m_0, M \rrbracket, \quad \mathbf{u}_m^k(n) = \mathbf{e}_m^k(n - m_0 + 1)
\end{aligned} \tag{6.19}$$

### Consequences on the parameters settings

Several parameters of the DAs are strongly dependent of the chosen input features. The new iterative computation of those inputs have therefore consequences on the setting of those parameters.

In each FMD, the feature set  $\Omega$ , the enlarge framing value  $\varepsilon$ , the histogram threshold  $\tau$ , and the boundaries update *mode* should evolve with their corresponding feature. However, implementation of these evolutive behaviors can be resources consuming. Modification of the feature set  $\Omega$  implies to change the support set of the histogram at each iteration, which generally implies a high computational cost. In our simulator, we choose to fix the support set, such that every possible new features would fit in this support. This was easily done for the two spatial axes as they have similar sets. The task becomes more complex when the rotation is done on several features of different precision and width. The chosen support set must be adapted to keep a good precision of the values without creating holes in the histogram representation of the feature. Modification of the framing enlargement  $\varepsilon$  and the histogram threshold  $\tau$  can be automatized using the values of the histogram mode and framing width. This can be done using percentage value as we did in our simulator, but we could observe that even the percentage needed to be adapted to the object the DA was converging to and thus needed to evolve along the iterative process. Finally, the boundaries update *Mode* has such impact on the object detection, it is difficult to establish when and how it must be changed. Especially when the object representation in the feature space is not *a priori* known, like in pre-attention application. In the case of the road lane detection, the axis  $Y$  was associated with the *Mode 2* as it was expected to be parallel with the sought lane. Similarly, the object principal axes can be associated with the *Mode 2* when it is expected that the object will expand on those directions.

Moreover this extension, like the rotation proposed in Chapter 5, distords enough the framing  $P_k$  to break its decreasing property. The convergence is therefore no more guaranteed, however we could observe that the detection was preserved, but oscillating. In the road lane detection, we used a smoothing of the rotation to stabilize the new axis long enough to be validated by the algorithm. Here, we simplify this by defining an accepted error for each boundary value to validate the convergence (Eq. 6.20). This, however, add supplementary parameters to the method: the accepted error  $err_m$  for each feature  $F_m$ .

**Algorithm step.** Convergence validation with accepted error

$$\bigwedge_{m=1}^M \left( \left| A_m^k - A_m^{k-M} \right| < err_m \right) \wedge \left( \left| B_m^k - B_m^{k-M} \right| < err_m \right). \tag{6.20}$$

### 6.3.2 Illustration

We illustrated the proposed method on the colored image from rawpixels.com showing multiple colored animals toys in different orientations. The experiments have been done with our BIPS simulator presented in Chapter 4 and the display was made with Matlab (*MATLAB and Image Processing Toolbox Release R2018a*).

We run 40 DAs with the hue as descriptive feature with a validation threshold  $\tau_{sat} = 2$  and the rotation only done on the spatial features  $X$  and  $Y$ . The observation of the image features helped us in the choice of the histogram thresholds:  $\tau_{hue} = 5\%$ ,  $\tau_{F_2} = 10\%$ ,  $\tau_{F_3} = 10\%$ ; the framing enlargement:  $\varepsilon_{hue} = 2^\circ$ ,  $\varepsilon_{F_2} = 10$ ,  $\varepsilon_{F_3} = 10$ ; and the object size threshold:  $\tau_N = 3000$  pixels. The update Mode = 1 for all FMDs. The DAs are launched in parallel and linked with inhibition of the three dimensional framing they detected.

We observe the DA states obtained with the classical method and the reorientation proposal at different iterative steps: 40, 80 and 160 shown in Figure 6.10. As expected the spatial framing obtained with the reorientation gives a better selection of the objects. The objects are entirely selected by the DAs and the ratio between the object size and the framing size is reduced.



FIGURE 6.10 – (A) Colored animal toys image. (B) Hue extraction.

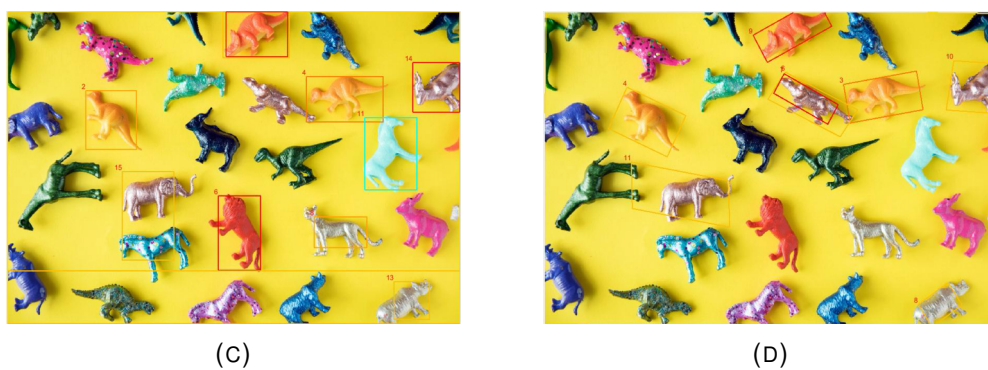


FIGURE 6.10 – DAs states at iteration 40 with (C) the classical method and (D) the reorientation of the spatial axes.

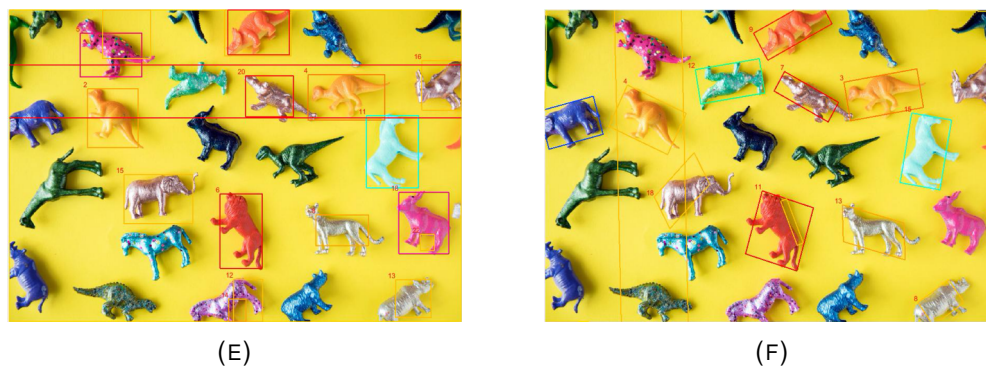


FIGURE 6.10 – DAs states at iteration 80 with (E) the classical method and (F) the reorientation of the spatial axes.

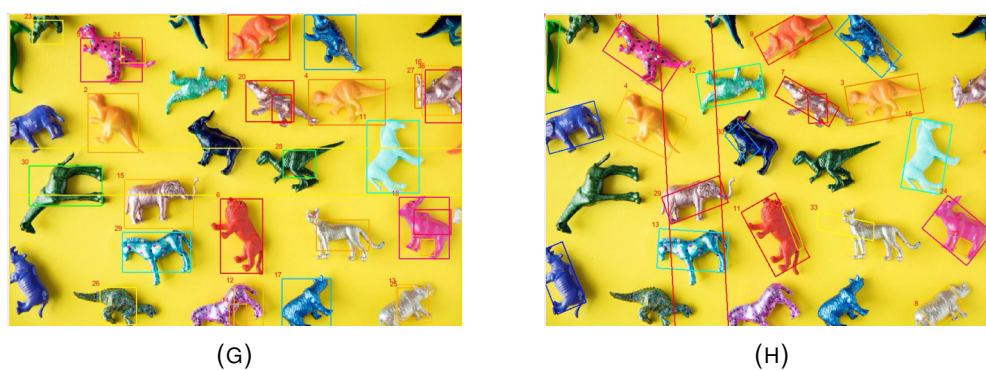


FIGURE 6.10 – DAs states at iteration 160 with (G) the classical method and (H) the reorientation of the spatial axes.

### 6.3.3 Conclusion

In this Section, we proposed a new and generic way to adapt the FMD input features, which would allow to obtain better framing of the sought objects. The proposed method calculates the covariance of the selected features at each iterative steps of the DA process and uses the obtained orthogonal basis to create the new set of input features.

The illustration of the extension tends to confirm the expected behavior of the DAs, thus extending the rotation to application without gradient angle information. This extension is a first step towards a more generic and evolving way to adapt the DA features input to the object it is attracted to.

## 6.4 Discussion

In this Chapter we proposed two extensions of our first BIPS-based implementation as first steps toward the SWaP, real-time, efficient, generic and evolutive properties expected from the BIPS bio-inspired concept. Firstly, a time integration is proposed to reduce the number of iterations, to improve the detection results and to track the detected objects. The Kalman filter is used to obtain a smooth and robust tracking of the DA framing even with a minimum number of iterations per image. Secondly, a reorientation of the input features space in each DA is proposed to improve the framing results and the inhibition connections between DAs.

The preliminary experiments used to illustrate those two proposals seem to validate the expected behaviors, however it also points out the highly parametric dependency of the method. Firstly, the time integration process implies the use of an enlarged framing and an error validation, whose enlargement  $\varepsilon_F$  and error values  $err_F$  need to be set depending on the application, the dataset and the DA architecture. Moreover, this framing can alterate the convergence property of the DA and its impact on other DAs it inhibits. The same issue happens with the reorientation of the feature space. Although we observe that the DA stays attracted to an object with a small oscillation around it, we have not yet validated that convergence to intervals. We could also observe that this oscillation, specially at the beginning of the attraction, had negative effects on the other DAs inhibited. In fact, the iterative reorientation of the feature space can create a very large parallelepiped framing of the validated pixels, which crushes all DAs already running in this area and inhibited by the master DA. This issue reminds the difficulty to set the associations and inhibitions between DAs, either for a specific application (top-down approach) or a more generic and evolving way (bottom-up approach).

A more complete analysis of the BIPS-based method with extensions is needed to estimate the influence and possible automation of each of these parameters and to evaluate the efficiency and real-time capacity of the method in different computer vision applications. Comparative experimentation on the caltech-lane dataset of the classical and new reorientation feature space, with and without time integration would give preliminary answers.



## Chapter 7

# Discussion and Perspectives

This thesis provides the formalization, simulation, evaluation on an academic dataset and extension proposals of the bio-inspired on-chip Dynamic Attractors (DAs), principal element of the Bio-Inspired Perception Sensor (BIPS) component.

The BIPS, developed by P. Pirim, is a bio-inspired component of interest for the embedded computer vision community. Its use in different industrial applications like detection and tracking, made it a good candidate for embedded computer vision solutions. However, the component, developed directly with an hardware point of view, suffers a lack of documentation and scientific publications on its underlying principles and precise method making it difficult for any academic evaluation. The thesis tries to fill in this gap.

This conclusive Chapter summarizes the contributions of our work in Section 7.1 and discusses of the benefits and limitations of the method in Section 7.2. Finally, Section 7.3 presents our conclusions on the BIPS formalized method and gives suggestions for future works.

## 7.1 Thesis contributions

### 7.1.1 BIPS formalization

Our first contribution, presented in Chapter 3, is the mathematical formalization of the method. Based on the patents and the articles of the inventor P. Pirim, we have been able to explain and to formulate mathematically the component embedded processes.

A first part of the BIPS proceeds to a selection of feature extractions, which can correspond to any kind of sensory perception (e.g. visual, audio). It has been proven that any feature represented by a finite and totally ordered set of values could be used without changing the functioning of the BIPS. The current implemented features are local and have a low computational cost. Some of them correspond to well known bio-inspired features (e.g. luminance, hue, saturation, gradient), others have an experimental and very peculiar implementation (e.g. temporal variation, velocity, angle variation).

The second part of the BIPS proceeds to the detection of objects in selected feature spaces. The principal element of this part is the Dynamic Attractor (DA). This element is inspired by several functions of the human brain: the "What" and "Where" stream computational connection (Wang et al., 1999), the attractor networks model (Knierim and Zhang, 2012) and the plasticity of the brain (Rita, Tyler, and Kaczmarek, 2003). In its concrete implementation, the module is based on the combination of three feature histogram analysis, performed in modules we called Feature Mode Detectors (FMDs). One feature is descriptive, the two others spatial. Our formalization however proves that the number and the type of features doesn't

break the convergence property of the DA. The module does an iterative process and its result is progressively attracted to a descriptive and spatial representation of one of the object of the scene. The DAs also interact with each other in parallel and in pipeline in order to either: (1) explore new areas (inhibition rule) or (2) complete the preceding detection with another DA descriptor (association rule) for enriching the object representation or for assessing the representation of an object made of multiple features.

Even if we were not able to link the DA process to known mathematical methods, this formalization now provides tools for a better understanding of the BIPS functioning and its possible use in computer vision applications.

### 7.1.2 BIPS simulator

The formalization of the DAs brings forward multiple parameters that are not automatically set and must be chosen depending on the application and the dataset. Moreover, the implementation of the method and the analysis of its behavior in different applications is not straightforward. Thus, the need for a software simulator became of major importance.

I developed the first BIPS simulator written in C++, using Qt frameworks and OpenCV. The simulator reproduces the different modules of the BIPS and its internal processes. Three main classes have been developed: the **Feature** represents an extracted feature and its attributes, the **FeatModeDetector** regroups the tools for the histogram analysis and the **DynamicAttractor** controls the FMDs and the iterative detection process. Supplementary tools have been developed: several feature extractions have been implemented, but also tools for recording or visualizing the results have been developed. The simulator leave most of the parameters free, so they could be adapted to the chosen application. Only the multiple DAs architecture has not been generalized yet and a specific script must be written for each application.

This simulator provides tools to facilitate the implementation of the BIPS method on specific application and datasets, to visualize the DAs behavior and to record the obtained results. It also permits the integration of method extensions like the reorientation of the feature space axes and the Kalman filter prediction.

### 7.1.3 First parametric and performance analysis

The two previous contributions allow to set up the first academic analysis of the BIPS method. This contribution helps in the understanding of the DAs behavior and setting methodology for a specific application.

The road lane detection application has been chosen for its straightforward choice of the DAs parameters and architecture. Using the two lines model of the road lanes, we are able to set the inputs to the gradient angle and the spatial axes. A first extension of the DAs is then applied to allow the reorientation of the spatial axes and a better framing of the sought lane. For this application, multiple similar DAs are linked to each other by inhibition to work on different areas of the image. Only four of their parameters are not set automatically: the threshold on the gradient angle and the  $X$  axis histograms, the number of DAs and the threshold on the size of the sought lane. The analysis of the BIPS behavior on the *caltech-lanes* dataset permits to link the two last parameters and to validate the weak influence of the two histogram thresholds.

The study shows that the method is able to reach detection results at the level of the reference (Aly, 2008). It permits to validate the detection behavior presented

by the formalization and gives promising results for specific application use of the method.

#### **7.1.4 Extension proposals**

The robustness and evolutive property of the method, which are expected for embedded computer vision under high constraints, are not in adequacy with the application dependency of several DA parameters. In our last research work, we propose two extension proposals of the BIPS method. The main goal is to overcome some limitations of the formalized DAs processes.

The first proposal permits the integration of the time information, allowing first to improve the detection results, then to reduce the DAs iterations and thus to decrease the computational time of the process. Moreover, this new approach is able to issue tracking mechanisms in addition to the detected object representation. The second proposal permits to adapt the feature input space to the different objects orientations along the DAs iterative process. This gives better representation of the detected objects and allows finer analysis of the processed image.

## **7.2 Dynamic Attractors benefits and limits**

The formalization and simulation of the method, as well as our experimentations on the road lane detection, and first experimentations on moving people detection and tracking highlight the advantages, but also the limitations of such a promising method.

### **7.2.1 Advantages**

#### **Size, Weight and Power optimized**

One determining advantage of this method is that it has already been implemented on an ASIC component of 50mm size for a power consumption of  $\sim 2W$  (Pirim, 2015). The formalization of the method demonstrates that it is solely based on low level local feature extraction processes and histogram computations. Those processes are power frugal. The Size, Weight and Power optimisation objective should be then validated for the new hardware implementations of the BIPS method.

#### **Real-time**

Even if the real-time capacity of the method has not been academically proved, the BIPS formalization, behavior analysis and first tracking experimentations give promising computational time results. The experiments demonstrated also that only a few iterative steps were enough to obtain a significant representation of the detected object.

#### **Versatile**

The formulation of the method demonstrates that the input feature does not necessarily have to be local visual features. The conditions on the input features consist in having a common definition domain, and a finite and totally ordered co-domain set. This construction allows to wide up the spectrum of features. Moreover, the



number of features does not necessarily have to be set to 3, neither it has to be one descriptive and two spatial features like in the BIPS component. Thus, multiple ways of detecting elements can be performed by that same DA process. This property of the DA opens it to multiple applications and seems to validate the versatile objective of the embedded computer vision systems.

### 7.2.2 Parametric limitations

The multiple ways to organize and set the DAs bring, however, limitations in their capacity to evolve and adapt. At a first glance, the BIPS seems to be a simple and powerful component for the environment perception. But, its behaviour requires to tune a set of parameters whose influence is not easy to understand. To simplify this task, the following table reminds the different parameters of the method and explains the difficulties to find an automated setting of them.

Parameters	Influence	Could be automatized with
<i>in the FMDs</i>		
Histogram threshold $\tau_{F_m}^k$	the object representation too high $\rightarrow$ information lost too low $\rightarrow$ integrate information of other objects	the histogram peak $q_{m,k}^{\max}$ , the current object size $N_k$ , the sought object size $\tau_N$
Update mode	the type of object mode 1 $\rightarrow$ one peak of the associated feature mode 2 $\rightarrow$ all the peaks	the FMD associated feature (which can evolve in the reorientation extension)
Enlargement $\epsilon_m^k$	the object representation and the convergence time too big $\rightarrow$ integrate information of other objects too small $\rightarrow$ slow down the convergence	the boundaries distance $(B_m^k - A_m^k)$ , the current object size $N_k$ the sought object size $\tau_N$
<i>in the DAs</i>		
Features input set	the type of object and its representation	<i>It depends on the application. The reorientation extension allows to adapt the features to the detected object</i>
Size threshold $\tau_N$	the object representation too high $\rightarrow$ some objects might not be detected too low $\rightarrow$ noises might be detected as object	the size of previous detection the size of other DAs detection

Number of iterative steps $k_{final}$	the object representation and the computational time too big $\rightarrow$ increases the computational time when the convergence is slow too small $\rightarrow$ does not yet represent the object	<i>The time integration extension with Kalman filter allows to reduce the influence of this number</i>
Asymptotic validation	the object representation and the computational time too big $\rightarrow$ validation before the detection too small $\rightarrow$ validation impossible because of the oscillation	<i>It depends on the application and the chosen features</i>
Kalman parameters	the object tracking	<i>The chosen model depends on the application</i>
<i>in the complete process</i>		
Number of DAs $n_{DA}$	the maximum number of detected object	the object size threshold the number of validated pixels <i>It could be changed dynamically</i>
Architecture (association and inhibition)	the type of objects and their representation the order of detection the computational time	<i>It depends on the application</i>

TABLE 7.1 – Parameters of influence: what influence do they have and which information could help automatize their setting.

## 7.3 Conclusions and Perspectives

### 7.3.1 Conclusions

At the beginning of this thesis research project, the BIPS component, which had shown interesting embedded characteristics in industrial applications, was presenting a important lack of documentation on its functioning and had never been analyzed on academic datasets.

Our work brings to community a formalization and a simulator of the BIPS. Both tools help in the understanding of the method behavior and its use in different applications. It also confirms its interesting characteristics for embedded computer vision applications. The key points of this technology lies in its SWaP optimization, its real-time running and its possible use in different kinds of applications. However, the number of parameters to set and their strong dependencies to the chosen application could fail the evolutivity and efficiency expected in highly constrained applications like autonomous driving, surveillance and drone flying. The method seems more adapted for specific application or general purpose (low-level) pre-attention tasks.

The first academic experimentation, we set on the road lane application, highlights the setting methodology of the BIPS and shows the promising performance of the method. Moreover, our work on the prediction and reorientation proposals shows ways to improve its performance in more complex applications.

### 7.3.2 Short term perspectives

In a short term perspective, the ongoing works on the BIPS extension would have to be tested and validated on academic datasets.

- **Reorientation improvements analysis:**

It is desirable to test the reorientation extension on the *caltech-lane* dataset and compare its results to the classic method with rotation and to the reference (Aly, 2008). Moreover, the framing improvement of the extension has to be validated on segmented object databases. The ratio between the object area and the framing area can be used to compare results with and without the reorientation.

- **Tracking performances analysis:**

The preliminary tracking results obtained on the PETS benchmark need to be completed. After implementing the multiple object tracking, a comparative analysis between the classical method and its extension is expected. Moreover, the detection results for different number of iterations should be analyzed.

### 7.3.3 Long term perspectives

#### Parametric study

The biological principle followed by the BIPS suggests that the method should work for different kinds of perception application similarly to the first layers of the visual pathway. Keeping this idea implies that a combination of DAs should work for multiple applications. The current study does not permit to validate this hypothesis. To do so, a complete parametric analysis of the BIPS method should be run. First, the illustration of its behavior in different conditions and applications would help propose automatic way to set the parameters. Then, comparison to academic results would permit to evaluate the method performance in different applications.

#### Hardware performances analysis

Since the beginning of this thesis work, new deep-learning hardware implementations have emerged (Ionica and Gregg, 2015), (*Xilinx AI website*). Few performance analysis have been made yet, but this technology should be taken into account for future experimentations. Firstly, it would be interesting to compare hardware performances of the BIPS and these neural networks. But, also, it could be interesting to combine the two approaches, since the deep neural network could higher the level of analysis of the BIPS and the BIPS could reduce the number of layers necessary by performing a pre-processing of the scene.

# Bibliography

- Agarwal, A., S. Gupta, and D. K. Singh (2016). “Review of optical flow technique for moving object detection”. In: *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 409–413. DOI: [10.1109/IC3I.2016.7917999](https://doi.org/10.1109/IC3I.2016.7917999).
- Aly, M. (2008). “Real time detection of lane markers in urban streets”. In: *2008 IEEE Intelligent Vehicles Symposium*, pp. 7–12. DOI: [10.1109/IVS.2008.4621152](https://doi.org/10.1109/IVS.2008.4621152).
- Andrade, D. C. et al. (2019). “A Novel Strategy for Road Lane Detection and Tracking Based on a Vehicle’s Forward Monocular Camera”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.4, pp. 1497–1507. ISSN: 1524-9050. DOI: [10.1109/TITS.2018.2856361](https://doi.org/10.1109/TITS.2018.2856361).
- Audibert, J. and J. Ponce (2009). “Vanishing point detection for road detection”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 96–103. DOI: [10.1109/CVPR.2009.5206787](https://doi.org/10.1109/CVPR.2009.5206787).
- Bar Hillel, Aharon et al. (2014). “Recent progress in road and lane detection: a survey”. en. In: *Machine Vision and Applications* 25.3, pp. 727–745. ISSN: 1432-1769. DOI: [10.1007/s00138-011-0404-2](https://doi.org/10.1007/s00138-011-0404-2).
- BioComp 2016 – GDR BioComp*. URL: <https://gdr-biocomp.fr/en/colloque-2016/> (visited on 03/16/2019).
- BIPcam*. URL: <https://www.bvs-tech.com/website/eng/page.php?id=8> (visited on 10/22/2019).
- BIPeye*. URL: <https://www.bvs-tech.com/website/eng/page.php?id=8> (visited on 10/22/2019).
- Borkar, A., M. Hayes, and M. T. Smith (2009). “Robust lane detection and tracking with ransac and Kalman filter”. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 3261–3264. DOI: [10.1109/ICIP.2009.5413980](https://doi.org/10.1109/ICIP.2009.5413980).
- (2012). “A Novel Lane Detection System With Efficient Ground Truth Generation”. In: *IEEE Transactions on Intelligent Transportation Systems* 13.1, pp. 365–374. ISSN: 1524-9050. DOI: [10.1109/TITS.2011.2173196](https://doi.org/10.1109/TITS.2011.2173196).
- Bourne, James A. (2010). “Unravelling the development of the visual cortex: implications for plasticity and repair”. eng. In: *Journal of Anatomy* 217.4, pp. 449–468. ISSN: 1469-7580. DOI: [10.1111/j.1469-7580.2010.01275.x](https://doi.org/10.1111/j.1469-7580.2010.01275.x).
- Broggi, A. (1995). “Parallel and local feature extraction: a real-time approach to road boundary detection”. In: *IEEE Transactions on Image Processing* 4.2, pp. 217–223. ISSN: 1057-7149. DOI: [10.1109/83.342193](https://doi.org/10.1109/83.342193).
- BT.601. BT.601 : Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios*. URL: <https://www.itu.int/rec/R-REC-BT.601-5-199510-S/en> (visited on 06/04/2019).
- Burr, David and Peter Thompson (2011). “Motion psychophysics: 1985–2010”. In: *Vision Research. Vision Research 50th Anniversary Issue: Part 2* 51.13, pp. 1431–1456. ISSN: 0042-6989. DOI: [10.1016/j.visres.2011.02.008](https://doi.org/10.1016/j.visres.2011.02.008).
- BVS-Tech website*. URL: <https://www.bvs-tech.com/website/eng/page.php?id=17> (visited on 10/20/2019).

- Cahan, David (1993). *Hermann von Helmholtz and the Foundations of Nineteenth-Century Science*. University of California Press. ISBN: 978-0-520-08334-9. URL: <https://www.ucpress.edu/book/9780520083349/hermann-von-helmholtz-and-the-foundations-of-nineteenth-century-science>.
- Canny, J. (1986). "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6, pp. 679–698. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767851.
- Chen, S. Y. (2012). "Kalman Filter for Robot Vision: A Survey". In: *IEEE Transactions on Industrial Electronics* 59.11, pp. 4409–4420. ISSN: 0278-0046. DOI: 10.1109/TIE.2011.2162714.
- Cmglee. *Density of rod (dotted line) and cone (solid line) photoreceptors along a line passing through the fovea and the blind spot of a human eye vs the angle measured from the fovea, based on 'Foundations of Vision' by Brian A. Wandell*. URL: [https://commons.wikimedia.org/wiki/File:Human\\_photoreceptor\\_distribution.svg](https://commons.wikimedia.org/wiki/File:Human_photoreceptor_distribution.svg) (visited on 10/04/2019).
- Commons, Wikimedia. *Color Sensitivity*. URL: [https://en.wikipedia.org/wiki/File:1416\\_Color\\_Sensitivity.jpg](https://en.wikipedia.org/wiki/File:1416_Color_Sensitivity.jpg) (visited on 10/04/2019).
- (2012). *Greensboro, North Carolina concurrency between highways*. URL: [https://commons.wikimedia.org/wiki/File:Greensboro\\_road\\_signs.jpg](https://commons.wikimedia.org/wiki/File:Greensboro_road_signs.jpg) (visited on 02/19/2019).
- Cristóbal, Gabriel, Laurent Perrinet, and Matthias S. Keil, eds. (2015). *Biologically Inspired Computer Vision: Fundamentals and Applications*. Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA. ISBN: 978-3-527-68086-3 978-3-527-41264-8.
- Cáceres Hernández, Danilo et al. (2016). "Real-Time Lane Region Detection Using a Combination of Geometrical and Image Features". In: *Sensors (Basel, Switzerland)* 16.11. ISSN: 1424-8220. DOI: 10.3390/s16111935.
- Daugman, John G. (1980). "Two-dimensional spectral analysis of cortical receptive field profiles". In: *Vision Research* 20.10, pp. 847–856. ISSN: 0042-6989. DOI: 10.1016/0042-6989(80)90065-6.
- Delbruck, T. (1993). "Silicon retina with correlation-based, velocity-tuned pixels". eng. In: *IEEE transactions on neural networks* 4.3, pp. 529–541. ISSN: 1045-9227. DOI: 10.1109/72.217194.
- Dhp1080 (2019). *Neuron*. URL: <https://commons.wikimedia.org/wiki/File:Neuron.svg> (visited on 10/04/2019).
- Dllu (2017). *Autonomous Waymo Chrysler Pacifica Hybrid minivan undergoing testing in Los Altos, California*. URL: [https://commons.wikimedia.org/wiki/File:Waymo\\_Chrysler\\_Pacifica\\_in\\_Los\\_Altos,\\_2017.jpg](https://commons.wikimedia.org/wiki/File:Waymo_Chrysler_Pacifica_in_Los_Altos,_2017.jpg) (visited on 09/16/2019).
- Document d'orientation de l'innovation de Défense (DOID) 2019. Document d'orientation de l'innovation de Défense (DOID) 2019 : les nouvelles ambitions du ministère en matière d'innovation*. fr. URL: <https://www.defense.gouv.fr/actualites/articles/document-d-orientation-de-l-innovation-de-defense-doid-2019-les-nouvelles-ambitions-du-ministere-en-matiere-d-innovation> (visited on 09/15/2019).
- Ehsan, Shoaib and Klaus D. McDonald-Maier (2015). "On-Board Vision Processing For Small UAVs: Time to Rethink Strategy". In: *arXiv:1504.07021 [cs]*. arXiv: 1504.07021.
- Farnebäck, Gunnar (2003). "Two-Frame Motion Estimation Based on Polynomial Expansion". In: *Image Analysis*. Ed. by Josef Bigun and Tomas Gustavsson. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 363–370. ISBN: 978-3-540-45103-7.

- Felisa, M. and P. Zani (2010). “Robust monocular lane detection in urban environments”. In: *2010 IEEE Intelligent Vehicles Symposium*, pp. 591–596. DOI: [10.1109/IVS.2010.5548028](https://doi.org/10.1109/IVS.2010.5548028).
- Gaikwad, V. and S. Lokhande (2015). “Lane Departure Identification for Advanced Driver Assistance”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.2, pp. 910–918. ISSN: 1524-9050. DOI: [10.1109/TITS.2014.2347400](https://doi.org/10.1109/TITS.2014.2347400).
- Gautam, Akash (2017). “Nerve Cells”. In: *Encyclopedia of Animal Cognition and Behavior*, pp. 1–3. DOI: [10.1007/978-3-319-47829-6\\_1282-1](https://doi.org/10.1007/978-3-319-47829-6_1282-1).
- Ghorayeb, Hicham (2007). “Conception et mise en oeuvre d’algorithmes de vision temps réel pour la vidéo surveillance intelligente”. PhD thesis. Paris, ENMP.
- Gilbert, Charles D. and Wu Li (2012). “Adult Visual Cortical Plasticity”. In: *Neuron* 75.2, pp. 250–264. ISSN: 0896-6273. DOI: [10.1016/j.neuron.2012.06.030](https://doi.org/10.1016/j.neuron.2012.06.030).
- Hengstler, Stephan et al. (2007). “MeshEye: A Hybrid-Resolution Smart Camera Mote for Applications in Distributed Intelligent Surveillance”. In: *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. New York, NY, USA, pp. 360–369. ISBN: 978-1-59593-638-7. DOI: [10.1109/IPSN.2007.4379696](https://doi.org/10.1109/IPSN.2007.4379696).
- Hoang, Toan Minh et al. (2017). “Road Lane Detection Robust to Shadows Based on a Fuzzy System Using a Visible Light Camera Sensor”. In: *Sensors* 17.11, p. 2475. DOI: [10.3390/s17112475](https://doi.org/10.3390/s17112475).
- Horgan, J. et al. (2015). “Vision-Based Driver Assistance Systems: Survey, Taxonomy and Advances”. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2032–2039. DOI: [10.1109/ITSC.2015.329](https://doi.org/10.1109/ITSC.2015.329).
- Hubel, David (1982). “Evolution of ideas on the primary visual cortex, 1955–1978: A biased historical account”. In: *Bioscience Reports* 2.7, pp. 435–469. ISSN: 0144-8463, 1573-4935. DOI: [10.1007/BF01115245](https://doi.org/10.1007/BF01115245).
- (1995). *Eye, Brain, and Vision*. URL: <http://hubel.med.harvard.edu/index.html> (visited on 02/17/2016).
- Hueber, Nicolas et al. (2015). “Bio-inspired approach for intelligent unattended ground sensors”. In: vol. 9494. Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation; Theory and Applications IX. Misty Blowers, Dans Popa, Muthu B.J. Wijesundara. DOI: [10.1117/12.2177400](https://doi.org/10.1117/12.2177400). URL: <http://dx.doi.org/10.1117/12.2177400>.
- Hustvedt (2008). *Three surveillance cameras on the corner of a building*. URL: [https://commons.wikimedia.org/wiki/File:Three\\_Surveillance\\_cameras.jpg](https://commons.wikimedia.org/wiki/File:Three_Surveillance_cameras.jpg) (visited on 09/16/2019).
- Ionica, M. H. and D. Gregg (2015). “The Movidius Myriad Architecture’s Potential for Scientific Computing”. In: *IEEE Micro* 35.1, pp. 6–14. ISSN: 0272-1732. DOI: [10.1109/MM.2015.4](https://doi.org/10.1109/MM.2015.4).
- Itti, Laurent and Christof Koch (2000). “A saliency-based search mechanism for overt and covert shifts of visual attention”. In: *Vision Research* 40.10–12, pp. 1489–1506. ISSN: 0042-6989. DOI: [10.1016/S0042-6989\(99\)00163-7](https://doi.org/10.1016/S0042-6989(99)00163-7).
- Journées NeuroSTIC 2017 | ISIS | NeuroSTIC*. URL: <http://www.gdr-isis.fr/neurostic/?p=363> (visited on 03/16/2019).
- Jung, Jiyoung and Sung-Ho Bae (2018). “Real-Time Road Lane Detection in Urban Areas Using LiDAR Data”. In: *Electronics* 7.11, p. 276. DOI: [10.3390/electronics7110276](https://doi.org/10.3390/electronics7110276).
- Jung, S., J. Youn, and S. Sull (2016). “Efficient Lane Detection Based on Spatiotemporal Images”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.1, pp. 289–295. ISSN: 1524-9050. DOI: [10.1109/TITS.2015.2464253](https://doi.org/10.1109/TITS.2015.2464253).

- Kanellakis, Christoforos and George Nikolakopoulos (2017). "Survey on Computer Vision for UAVs: Current Developments and Trends". In: *Journal of Intelligent & Robotic Systems* 87.1, pp. 141–168. ISSN: 1573-0409. DOI: [10.1007/s10846-017-0483-z](https://doi.org/10.1007/s10846-017-0483-z).
- Kim, Hansung et al. (2007). "Robust Foreground Extraction Technique Using Gaussian Family Model and Multiple Thresholds". In: *Computer Vision – ACCV 2007. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 758–768. ISBN: 978-3-540-76385-7 978-3-540-76386-4. DOI: [10.1007/978-3-540-76386-4\\_72](https://doi.org/10.1007/978-3-540-76386-4_72).
- Kim, In Su et al. (2010). "Intelligent visual surveillance — A survey". In: *International Journal of Control, Automation and Systems* 8.5, pp. 926–939. ISSN: 2005-4092. DOI: [10.1007/s12555-010-0501-4](https://doi.org/10.1007/s12555-010-0501-4).
- Knierim, James J. and Kechen Zhang (2012). "Attractor Dynamics of Spatially Correlated Neural Activity in the Limbic System". In: *Annual Review of Neuroscience* 35.1. PMID: 22462545, pp. 267–285. DOI: [10.1146/annurev-neuro-062111-150351](https://doi.org/10.1146/annurev-neuro-062111-150351).
- Livingstone, M. S. and D. H. Hubel (1987). "Psychophysical evidence for separate channels for the perception of form, color, movement, and depth". In: *Journal of Neuroscience* 7.11, pp. 3416–3468. ISSN: 0270-6474, 1529-2401. DOI: [10.1523/JNEUROSCI.07-11-03416.1987](https://doi.org/10.1523/JNEUROSCI.07-11-03416.1987).
- Malacara, Daniel (2011). *Color Vision and Colorimetry: Theory and Applications, Second Edition*. SPIE. ISBN: 978-0-8194-8398-0. DOI: [10.1117/3.881172](https://doi.org/10.1117/3.881172).
- Marcus, Gary (2018). "Deep Learning: A Critical Appraisal". In: *arXiv:1801.00631 [cs, stat]*. arXiv: 1801.00631.
- Markov, Igor L. (2014). "Limits on fundamental limits to computation". In: *Nature* 512.7513, pp. 147–154. ISSN: 1476-4687. DOI: [10.1038/nature13570](https://doi.org/10.1038/nature13570).
- Marr, David, Shimon Ullman, and Tomaso A. Poggio (2010). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press. ISBN: 978-0-262-28898-9.
- Martinez-Conde, Susana, Stephen L. Macknik, and David H. Hubel (2000). "Microsaccadic eye movements and firing of single cells in the striate cortex of macaque monkeys". In: *Nature Neuroscience* 3.3, pp. 251–258. ISSN: 1546-1726. DOI: [10.1038/72961](https://doi.org/10.1038/72961).
- Marçelja, S. (1980). "Mathematical description of the responses of simple cortical cells\*". In: *JOSA* 70.11, pp. 1297–1300. DOI: [10.1364/JOSA.70.001297](https://doi.org/10.1364/JOSA.70.001297).
- MATLAB and Image Processing Toolbox Release R2018a*. Natick, Massachusetts, United States. URL: <https://fr.mathworks.com/products/matlab.html>.
- McCall, J. C. and M. M. Trivedi (2006). "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation". In: *IEEE Transactions on Intelligent Transportation Systems* 7.1, pp. 20–37. ISSN: 1524-9050. DOI: [10.1109/TITS.2006.869595](https://doi.org/10.1109/TITS.2006.869595).
- Medathati, N. V. Kartheek et al. (2016). "Bio-inspired computer vision: Towards a synergistic approach of artificial and biological vision". In: *Computer Vision and Image Understanding* 150.Supplement C, pp. 1–30. ISSN: 1077-3142. DOI: [10.1016/j.cviu.2016.04.009](https://doi.org/10.1016/j.cviu.2016.04.009).
- Nefian, A. V. and G. R. Bradski (2006). "Detection of Drivable Corridors for Off-Road Autonomous Navigation". In: *2006 International Conference on Image Processing*, pp. 3025–3028. DOI: [10.1109/ICIP.2006.313004](https://doi.org/10.1109/ICIP.2006.313004).
- Ogawa, Shumpei. *Visual cortices*. URL: <https://www.pinterest.fr/pin/527765650055756445/?lp=true>.

- Ohta, Noboru and Alan Robertson (2006). *Colorimetry: Fundamentals and Applications*. John Wiley & Sons. ISBN: 978-0-470-09473-0.
- OpenCV 2.4 release. URL: <https://opencv.org/releases.html> (visited on 04/02/2019).
- Ota, Kaoru et al. (2017). "Deep Learning for Mobile Multimedia: A Survey". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 13.3s, 34:1–34:22. ISSN: 1551-6857. DOI: 10.1145/3092831.
- PETS 2009. URL: <http://www.cvg.reading.ac.uk/PETS2009/a.html> (visited on 07/28/2019).
- Pirim, Patrick (1998). "Procédé et dispositif fonctionnant en temps réel, pour le repérage et la localisation d'une zone en mouvement relatif dans une scène, ainsi que pour la détermination de la vitesse et la direction du déplacement". Pat. FR2751772A1.
- (2001). "Procédé et dispositif de perception automatique". Pat. FR2805629A1.
- (2005). "Procédé et dispositif automatisé de perception avec détermination et caractérisation de bords et de frontières d'objets d'un espace, construction de contours et applications". Pat. FR2858447A1.
- (2013). "Generic Bio-inspired Chip Model-Based on Spatio-temporal Histogram Computation: Application to Car Driving by Gaze-Like Control". In: *Biomimetic and Biohybrid Systems*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 228–239. ISBN: 978-3-642-39801-8 978-3-642-39802-5. DOI: 10.1007/978-3-642-39802-5\_20.
- (2015). "Processeur de perception bio-inspiré : une approche neuromorphique". In: *Techniques de l'ingénieur Innovations en électronique et optoélectronique*. URL: <http://www.techniques-ingenieur.fr/base-documentaire/innovations-th10/innovations-en-electronique-et-optoelectronique-42257210/processeur-de-perception-bio-inspire-une-approche-neuromorphique-in220/> (visited on 02/09/2016).
- (2016). "Perceptive Invariance and Associative Memory Between Perception and Semantic Representation USER a Universal SEMantic Representation Implemented in a System on Chip (SoC)". In: *Biomimetic and Biohybrid Systems*. Lecture Notes in Computer Science. Springer, Cham, pp. 275–287. ISBN: 978-3-319-42416-3 978-3-319-42417-0.
- Pixabay. *Road, Asphalt, Sky, Horizon*. URL: <https://pixabay.com/photos/road-asphalt-sky-horizon-direction-1031702/> (visited on 03/19/2019).
- Posch, C. et al. (2014). "Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output". In: *Proceedings of the IEEE* 102.10, pp. 1470–1484. ISSN: 0018-9219. DOI: 10.1109/JPROC.2014.2346153.
- Power of a Human Brain - The Physics Factbook*. URL: <https://hypertextbook.com/facts/2001/JacquelineLing.shtml> (visited on 04/06/2018).
- Purves, Dale et al. (2018). *Neuroscience*. 6th ed. 2018. New York: Sinauer Associates Is an Imprint of Oxford University Press. ISBN: 978-1-60535-380-7.
- Qt 5.7 (2016). URL: <https://doc.qt.io/archives/qt-5.7/> (visited on 04/02/2019).
- Ranft, B. and C. Stiller (2016). "The Role of Machine Vision for Intelligent Vehicles". In: *IEEE Transactions on Intelligent Vehicles* 1.1, pp. 8–19. DOI: 10.1109/TIV.2016.2551553.
- Rennó-Costa, César, John E. Lisman, and Paul F. M. J. Verschure (2014). "A Signature of Attractor Dynamics in the CA3 Region of the Hippocampus". In: *PLOS Computational Biology* 10.5, e1003641. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003641.



- Rita, Paul Bach-y, Mitchell E. Tyler, and Kurt A. Kaczmarek (2003). "Seeing with the Brain". In: *International Journal of Human-Computer Interaction* 15.2, pp. 285–295. ISSN: 1044-7318. DOI: [10.1207/S15327590IJHC1502\\_6](https://doi.org/10.1207/S15327590IJHC1502_6).
- Saalmann, Yuri B. and Sabine Kastner (2011). "Cognitive and perceptual functions of the visual thalamus". In: *Neuron* 71.2, pp. 209–223. ISSN: 1097-4199. DOI: [10.1016/j.neuron.2011.06.027](https://doi.org/10.1016/j.neuron.2011.06.027).
- Schuman, Catherine D. et al. (2017). "A Survey of Neuromorphic Computing and Neural Networks in Hardware". In: *arXiv:1705.06963 [cs]*. arXiv: 1705.06963.
- Shi, Weijing et al. (2017). "Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey". In: *Integration* 59, pp. 148–156. ISSN: 0167-9260. DOI: [10.1016/j.vlsi.2017.07.007](https://doi.org/10.1016/j.vlsi.2017.07.007).
- Son, Jongin et al. (2015). "Real-time illumination invariant lane detection for lane departure warning system". In: *Expert Systems with Applications* 42.4, pp. 1816–1824. ISSN: 0957-4174. DOI: [10.1016/j.eswa.2014.10.024](https://doi.org/10.1016/j.eswa.2014.10.024).
- Son, Yeongho, Elijah S. Lee, and Dongsuk Kum (2019). "Robust multi-lane detection and tracking using adaptive threshold and lane classification". In: *Machine Vision and Applications* 30.1, pp. 111–124. ISSN: 1432-1769. DOI: [10.1007/s00138-018-0977-0](https://doi.org/10.1007/s00138-018-0977-0). (Visited on 04/23/2019).
- Sorenson, Josh (2018). *A quadcopter camera drone in flight*. URL: [https://commons.wikimedia.org/wiki/File:Quadcopter\\_camera\\_drone\\_in\\_flight.jpg](https://commons.wikimedia.org/wiki/File:Quadcopter_camera_drone_in_flight.jpg) (visited on 09/16/2019).
- Stangor, Charles. *Pathway of Visual Images through the Thalamus and into the Cortex*. URL: <https://opentextbc.ca/introductiontopsychology/chapter/4-2-seeing/> (visited on 10/03/2019).
- *The Retina with Its Specialized Cells*. URL: <https://opentextbc.ca/introductiontopsychology/chapter/4-2-seeing/> (visited on 10/03/2019).
- KITTI. *The KITTI Vision Benchmark Suite*. URL: [http://www.cvlibs.net/datasets/kitti/eval\\_object.php?obj\\_benchmark](http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark) (visited on 04/06/2018).
- PASCAL VOC. *The PASCAL Visual Object Classes*. URL: <http://host.robots.ox.ac.uk/pascal/VOC/> (visited on 07/24/2018).
- Thuries, Serge and Patrick Pirim (1988). "Procédé et dispositif de traitement en temps réel d'un flot de données séquence, et application au traitement de signaux vidéo numériques représentatifs d'une image vidéo". Pat. FR2611063A1.
- Tovée, Martin J. (2008). *An Introduction to the Visual System*. Cambridge University Press. ISBN: 978-1-139-47267-8.
- Veit, T. et al. (2008). "Evaluation of Road Marking Feature Extraction". In: *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 174–181. DOI: [10.1109/ITSC.2008.4732564](https://doi.org/10.1109/ITSC.2008.4732564).
- Velez, Gorka and Oihana Otaegui (2016). "Embedding vision-based advanced driver assistance systems: a survey". In: *IET Intelligent Transport Systems* 11.3, pp. 103–112. ISSN: 1751-9578. DOI: [10.1049/iet-its.2016.0026](https://doi.org/10.1049/iet-its.2016.0026).
- Volder, J. E. (1959). "The CORDIC Trigonometric Computing Technique". In: *IRE Transactions on Electronic Computers* EC-8.3, pp. 330–334. ISSN: 0367-9950. DOI: [10.1109/TEC.1959.5222693](https://doi.org/10.1109/TEC.1959.5222693).
- Wang, Jiongjiang et al. (1999). "Relationship between ventral stream for object vision and dorsal stream for spatial vision: An fMRI+ERP study". In: *Human Brain Mapping* 8.4, pp. 170–181. ISSN: 1097-0193. DOI: [10.1002/\(SICI\)1097-0193\(1999\)8:4<170::AID-HBM2>3.0.CO;2-W](https://doi.org/10.1002/(SICI)1097-0193(1999)8:4<170::AID-HBM2>3.0.CO;2-W).
- Webvision – The Organization of the Retina and Visual System*. URL: <https://webvision.med.utah.edu/> (visited on 11/29/2018).

- Weller, R. E. (1988). "Two cortical visual systems in Old World and New World primates". In: *Progress in Brain Research*. Vision Within Extrageniculo-Striate Systems 75, pp. 293–306. ISSN: 0079-6123. DOI: [10.1016/s0079-6123\(08\)60487-2](https://doi.org/10.1016/s0079-6123(08)60487-2).
- Wiesel, Torsten N. (1982). "The postnatal development of the visual cortex and the influence of environment". In: *Bioscience Reports* 2.6, pp. 351–377. ISSN: 0144-8463, 1573-4935. DOI: [10.1007/BF01119299](https://doi.org/10.1007/BF01119299).
- Wikimedia (2006). *Magnolia*. URL: [https://commons.wikimedia.org/wiki/File:Fleur\\_de\\_magnolia2.jpg](https://commons.wikimedia.org/wiki/File:Fleur_de_magnolia2.jpg) (visited on 02/07/2019).
- Wu, T. and A. Ranganathan (2012). "A practical system for road marking detection and recognition". In: *2012 IEEE Intelligent Vehicles Symposium*, pp. 25–30. DOI: [10.1109/IVS.2012.6232144](https://doi.org/10.1109/IVS.2012.6232144).
- Xilinx. *AI Inference Acceleration*. URL: <https://www.xilinx.com/applications/megatrends/machine-learning.html> (visited on 10/16/2019).
- Yablonski, Maya et al. (2017). "Microsaccades are sensitive to word structure: A novel approach to study language processing". In: *Scientific Reports* 7. ISSN: 2045-2322. DOI: [10.1038/s41598-017-04391-4](https://doi.org/10.1038/s41598-017-04391-4).
- Yilmaz, Alper, Omar Javed, and Mubarak Shah (2006). "Object Tracking: A Survey". In: *ACM Comput. Surv.* 38.4. ISSN: 0360-0300. DOI: [10.1145/1177352.1177355](https://doi.org/10.1145/1177352.1177355).
- Zhang, Xiang et al. (2018). "A Fast Learning Method for Accurate and Robust Lane Detection Using Two-Stage Feature Extraction with YOLO v3". In: *Sensors (Basel, Switzerland)* 18.12. ISSN: 1424-8220. DOI: [10.3390/s18124308](https://doi.org/10.3390/s18124308).



# The BIPS proofs and alternative equations

In this Appendix I provide proofs of the results given in Chapter 3, Chapter 5 and Chapter 6. The values existence and the process convergence assumed in the mathematical formalization of the Bio-inspired Perceptive Sensor (BIPS) mechanisms are proven here.

Section A.1 presents the proofs linked to the FMD, whereas Section A.2 presents the proofs linked to the iterative process of the DA and the consequences of the proposed extensions.

## .1 Feature Mode Detector

### .1.1 Histogram computation

*Proof.* Existence of  $q_{\max}$

We define the codomain space  $\mathcal{H} = H(\Omega)$ , thus by definition  $q_{\max} = \max(\mathcal{H})$  (Eq. 3.17).

$$\left. \begin{array}{l} \mathcal{H} \subset \mathbb{N} \\ \Omega \text{ is a finite set} \Rightarrow \mathcal{H} \text{ is a finite set} \\ \Omega \text{ is not empty} \Rightarrow \mathcal{H} \text{ is not empty} \end{array} \right\} \Rightarrow q_{\max} \text{ exists and is unique.}$$

□

*Proof.* Existence of  $f_{\max}$

We define the subset  $\mathcal{V} = H^{-1}(q_{\max})$ , thus by definition  $f_{\max} = \min(\mathcal{V})$  (Eq. 3.18).

$$\left. \begin{array}{l} \mathcal{V} \subset \Omega \\ \mathcal{V} \neq \emptyset \end{array} \right\} \Rightarrow f_{\max} \text{ exists and is in } \Omega$$

□

### .1.2 Boundaries update

*Proof.* Existence of  $A$  and  $B$ : Mode 1

We define the spaces  $\mathcal{A} = \{f \in \Omega, f \leq f_{\max} / \forall f' \in [f; f_{\max}], H(f') > \tau\}$  and  $\mathcal{B} = \{f \in \Omega, f \geq f_{\max} / \forall f' \in [f_{\max}; f], H(f') > \tau\}$ , thus by definition  $A = \min(\mathcal{A})$  and  $B = \max(\mathcal{B})$ .

$$\left. \begin{array}{l} \mathcal{A} \subset \Omega \Rightarrow \mathcal{A} \text{ is a finite set} \\ \mathcal{A} \text{ is totally ordered} \\ q_{\max} > \tau \Rightarrow f_{\max} \in \mathcal{A} \Rightarrow \mathcal{A} \neq \emptyset \end{array} \right\} \Rightarrow A \text{ exists and is unique}$$

$$q_{\max} > \tau \Rightarrow \left. \begin{array}{l} \mathcal{B} \subset \Omega \Rightarrow \left. \begin{array}{l} \mathcal{B} \text{ is a finite set} \\ \mathcal{B} \text{ is totally ordered} \end{array} \right\} \Rightarrow B \text{ exists and is unique} \\ f_{\max} \in \mathcal{B} \Rightarrow \mathcal{B} \neq \emptyset \end{array} \right\}$$

Besides,  $\forall f \in \mathcal{A}, f \leq f_{\max}$ , so  $A \leq f_{\max}$  and  $\forall f \in \mathcal{B}, f_{\max} \leq f$ , so  $f_{\max} \leq B$ .

$$\Rightarrow A \leq B.$$

□

*Proof.* Existence of  $A$  and  $B$ : Mode 2 For this proof, we need to add two virtual values  $f_{-\infty}$  and  $f_{+\infty}$  which complete  $\Omega$  to form the extended group  $\overline{\Omega}$  and for which  $\forall f_- \in [f_{-\infty}; \min(\Omega)[\text{and } f_+ \in ]\max(\Omega); f_{+\infty}]$ ,  $H(f_-) = H(f_+) = 0$ . We admit that such values exist as  $\Omega$  is most of the time part of  $\mathbb{Z}$ .

We define the spaces  $\mathcal{A} = \{f \in \Omega / \forall f' \in [f_{-\infty}; f[ [H(f') \leq \tau]\}$  and  $\mathcal{B} = \{f \in \Omega / \forall f' \in ]f; f_{+\infty}], H(f') \leq \tau\}$ . The same demonstration is done, except  $A = \max(\mathcal{A})$ ,  $B = \min(\mathcal{B})$  and the element always in  $\mathcal{A}$  (resp.  $\mathcal{B}$ ) is  $\min(\Omega)$  (resp.  $\max(\Omega)$ ).

Moreover  $q_{\max} > \tau$  implies that  $\forall f > f_{\max}, f \notin \mathcal{A}$  and that  $\forall f < f_{\max}, f \notin \mathcal{B}$ .

$$\Rightarrow A \leq B.$$

□

### .1.3 Alternative equations

The FMD formalization can be slightly different when using exclusive "<" or inclusive " $\leq$ " comparison in the pixel validation (Eq. 3.15). The impact in the software/hardware implementation is almost nil, but if using the exclusive comparison we must adapt the other definitions to keep the convergence property. In this Section I present the alternative equations for someone who would use the exclusive comparison.

**Definition.** Initialization

$$\begin{aligned} A &= f_{-\infty} \\ B &= f_{+\infty} \end{aligned}$$

**Definition.** Conditioned feature histogram

$$\begin{aligned} H: \Omega &\longrightarrow \mathbb{N} \\ f &\longrightarrow H(f) = \mathbf{card} \left\{ p \in \mathbb{I} \times \mathbb{J} \left/ \begin{array}{l} \text{val}(p) = 1, \\ F(p) = f, \\ A < f < B \end{array} \right. \right\} \end{aligned}$$

**Definition.** Mode 1

$$\begin{aligned} A &= \min \left\{ f \in \overline{\Omega} / \forall f' \in ]f; f_{\max}], H(f') > \tau \right\} \\ B &= \max \left\{ f \in \overline{\Omega} / \forall f' \in [f_{\max}; f[, H(f') > \tau \right\} \end{aligned}$$

**Definition.** Mode 2

$$\begin{aligned} A &= \max \left\{ f \in \overline{\Omega} \mid \forall f' \in [f_{-\infty}; f], H(f') \leq \tau \right\} \\ B &= \min \left\{ f \in \overline{\Omega} \mid \forall f' \in [f; f_{+\infty}], H(f') \leq \tau \right\} \end{aligned}$$

**Definition.** FMD pixel validation

$$\begin{aligned} val_{FMD}(p) : \mathbb{P} &\longrightarrow \{0, 1\} \\ p &\longmapsto val_{FMD}(p) = (A < F(p) < B) \end{aligned}$$

**Definition.** Validated pixel set

$$\mathcal{O} = F^{-1}(\lrcorner A, B \lrcorner) \cap val^{-1}(1)$$

## .2 Dynamic Attractor

### .2.1 Convergence

The DA iterative process creates a decreasing sequence of object framing  $(P_k)_{k \in \mathbb{N}}$  and of representing group of pixels  $(O_k)_{k \in \mathbb{N}}$ .

**Definition.**  $P_k$  and  $O_k$

$$\begin{aligned} P_k &= \prod_{m \in \llbracket 1, M \rrbracket} [A_m^k, B_m^k] \\ O_k &= F^{-1}(P_k) \cap val^{-1}(1) \end{aligned}$$

*Proof.* Decrease of  $(P_k)_{k \in \mathbb{N}}$  and  $(O_k)_{k \in \mathbb{N}}$

For all  $k \in \mathbb{N}$ ,

$$\begin{aligned} P_{k+1} &= \prod_{m \in \llbracket 1, M \rrbracket} [A_m^{k+1}, B_m^{k+1}] \\ \Leftrightarrow P_{k+1} &= \left( \prod_{m \in \llbracket 1, n-1 \rrbracket} [A_m^k, B_m^k] \right) \times [A_n^{k+1}, B_n^{k+1}] \times \left( \prod_{m \in \llbracket n+1, M \rrbracket} [A_m^k, B_m^k] \right) \end{aligned}$$

where  $n$  is such that  $k = n - 1[M]$ .

Moreover,

$$\begin{aligned} \forall f \in \Omega_n, \quad f < A_n^k, \quad H_n^k(f) &= 0 \\ \forall f \in \Omega_n, \quad f > B_n^k, \quad H_n^k(f) &= 0. \end{aligned} \quad (\text{due to Eq. 3.26})$$

Thus by definitions of the Boundaries update (Eq. 3.27 and 3.28),

$$\begin{aligned} &A_n^k \leq A_n^{k+1} \leq B_n^{k+1} \leq B_n^k \\ \Rightarrow &P_{k+1} \subseteq P_k \\ \Rightarrow &F^{-1}(P_{k+1}) \subseteq F^{-1}(P_k) \\ \Rightarrow &O_{k+1} \subseteq O_k \end{aligned}$$

□

*Proof.*  $O_k$  and  $P_k$  are not empty, and  $q_{\max}^{n,k}$  is strictly positive

We suppose that the pixel validation input  $val$  is not null, otherwise the DA will not even start to work.  $P_0 = \Omega$ , thus  $O_0 = val^{-1}(1) \neq \emptyset$ .

$$\forall k \in \mathbb{N}, \forall n \in \llbracket 1, M \rrbracket,$$

$$\begin{aligned} N_k &= \mathbf{card}(\mathcal{O}_k) \\ &= \sum_{f \in \Omega_n} H_n^k(f) \end{aligned}$$

$$\forall f \in \Omega_n, H_n^k(f) = \mathbf{card} \left\{ p \in \mathbb{I} \times \mathbb{J} \left/ \begin{array}{l} p \in \mathcal{O}_k \\ F(p) = f \end{array} \right. \right\}$$

Thus, if  $\mathcal{O}_k \neq \emptyset \Rightarrow q_{\max}^{n,k} > 0$ . As defined in Section 3.3.1, the threshold can be chosen such that  $q_{\max}^{n,k} > \tau_n^k \geq 0$ .

$$\begin{aligned} q_{\max}^{n,k} > 0 &\Rightarrow \exists p, \left\{ \begin{array}{l} \exists p \in \mathcal{O}_k, F_n(p) = f_{\max}^{n,k} \quad (\text{due to Eq. 3.26}) \\ A_n^{k+1} \leq F_n(p) \leq B_n^{k+1} \quad (\text{due to Eq. 3.27 and 3.28}) \end{array} \right. \\ &\Rightarrow \exists p \in \mathcal{O}_k \\ &\Rightarrow \exists p, F_n(p) \in P_k \end{aligned}$$

□

*Proof.*  $(P_k)_{k \in \mathbb{N}}$  and  $(\mathcal{O}_k)_{k \in \mathbb{N}}$  converge

$$\left. \begin{array}{l} (\mathcal{O}_k)_{k \in \mathbb{N}} \text{ is decreasing} \\ \forall k \in \mathbb{N}, \mathcal{O}_k \neq \emptyset \end{array} \right\} \Rightarrow \mathcal{O}_k \xrightarrow[k \rightarrow +\infty]{} \mathcal{O}$$

$$\left. \begin{array}{l} (P_k)_{k \in \mathbb{N}} \text{ is decreasing} \\ \forall k \in \mathbb{N}, P_k \neq \emptyset \end{array} \right\} \Rightarrow P_k \xrightarrow[k \rightarrow +\infty]{} P = \prod_{m \in \llbracket 1, M \rrbracket} \left[ \lim_{k \rightarrow +\infty} A_m^k, \lim_{k \rightarrow +\infty} B_m^k \right]$$

□

*Remark.* The two suppositions  $q_{\max}^{n,k} > \tau_n^k$  and  $val \neq 0$  are necessary to ensure the convergence.









**Titre :** Formalisation et simulation des Attracteurs Dynamiques sur composant bio-inspiré pour une vision artificielle à faible consommation énergétique

**Mots clés :** Bio-inspiré, Vision artificielle, Embarqué, Temps-réel

**Résumé :** Il devient difficile aujourd'hui de maintenir une amélioration des performances en vision artificielle tout en gardant une capacité de traitement embarqué temps-réel. L'étude du modèle de vision de l'être humain apporte différentes inspirations pour répondre à ce problème. La méthode utilisée par le composant bio-inspiré de BVS-Tech a attiré l'attention de la communauté par ses caractéristiques matérielles et sa mise en oeuvre dans plusieurs projets industriels. Mais l'absence de formalisme mathématique et de simulation de la méthode ne permet pas une analyse des performances de la méthode sur des bases académiques.

Cette thèse s'est appliquée à approfondir, formaliser et reproduire le concept du système. La modélisation mathématique et la création d'un simulateur ont permis de valider le concept de la méthode sur un set d'images artificielles et d'analyser son comportement et ses performances dans une application spécifique. Les résultats prometteurs mettent cependant en avant la forte dépendance paramétrique de la méthode. Deux extensions de la méthode ont récemment été proposées : l'adaptation automatique des caractéristiques d'entrée d'une part, et la mise en place d'un suivi de cible.

**Title :** Formalization and Simulation of Bio-Inspired On-chip Dynamic Attractors for Low Power Computer Vision

**Keywords :** Bio-inspired, Computer Vision, Embedded, Real-time

**Abstract :** High performance in real-time is one of today's challenge of embedded Computer Vision. The study of the human vision models inspires several solutions. The bio-inspired method used by the BVS-Tech component caught the attention of the community with its hardware characteristics and its application in different industrial projects. However, the lack of mathematical formalization and simulation of the method does not allow to analyse it on academic databases. This thesis focused on the understanding,

the formalization and the reproduction of the method. The mathematical model and the simulator produced allowed to validate the concept on artificial images, to analyse its behaviour and performances in a specific application. The promising results shows, however, the strong influence of several parameters. Two extensions of the method have recently been proposed: the automatic adaptation of the feature inputs and the prediction integration.