Stage de spécialisation SCIA

Rapport de stage

Version 1.0
Du 11 juin 2004

Rédacteur :
Anthony BAILLARD
(Promotion 2004)

Période :

- Du 05 Janvier 2004 au 04 Juillet 2004

Localisation :

- Unité Terapix de l'Institut d'Astrophysique de Paris

Sujet :

Détection automatique de défauts optiques sur des images grand-champ de l'espace profond.

Remerciements

Je tiens à remercier Monsieur Emmanuel Bertin, mon maître de stage, ainsi que Monsieur Yannick Mellier, responsable de l'équipe Terapix. Je remercie également Monsieur Frédéric Magnard qui m'a accueilli dans son bureau pour la durée du stage, ainsi que tous les autres membres de l'équipe, mesdames Delphine Charbonneau et Mireille Dantel ; messieurs Laurent Domisse, Jean-Christophe Malapert, Henry Joy McCracken, Gilles Missonnier et Gérard Tissier, pour leur accueil et leur collaboration.

Un grand merci également à madame Valérie Bona qui s'est occupée des modalités administratives de mon stage avec beaucoup de volonté.

D'une manière générale, mes remerciements s'adressent à tout le personnel de l'IAP grâce à qui ce stage a pu se dérouler dans les meilleures conditions.

Remarques préliminaires et notations

Tous les mots suivis d'un * sont définis dans le glossaire (chapitre 8).

Table des matières

1 Résumé

Le présent document constitue le rapport du stage effectué de janvier à juillet 2004 au sein de l'Institut d'Astrophysique de Paris (IAP). l'IAP est un laboratoire de recherches fondamentales installé à Paris et dont les activités sont centrées autour de la recherche internationale en astrophysique. Plusieurs équipes travaillent au sein de l'Institut dont celle qui accueille le stage, TERAPIX (Traitement Élémentaire, Réduction et Analyse des PIXels de megacam), et dont le travail est de traiter des données astronomiques en provenance du télescope Canada-France-Hawaii. C'est au sein de cette équipe de 8 personnes que le stage a eu lieu.

Au cours de ces six mois, le but du stage était de produire un logiciel capable de détecter automatiquement des défauts optiques sur des images grand-champ de l'espace profond. Le logiciel devait s'inscrire dans l'ensemble du projet TERAPIX afin d'améliorer le traitement des images au niveau scientifique. Les principales techniques utilisées durant le stage sont l'Analyse en Composantes Principales et les réseaux de neurones comme système d'apprentissage supervisé.

Plusieurs modifications ont été apportées successivement et le stage a notamment été ponctué par la participation à une conférence internationale sur le traitement de données en astronomie. A l'issu du stage, le développement du logiciel était achevé et s'est avéré fonctionnel sur les défauts testés. L'extension à d'autres défauts ne devrait pas poser de problèmes.

Le stage a été une expérience positive en tous points puisqu'il m'a permis de mettre en pratique mes connaissances, c'est-à-dire, développer des technologies d'informatique avancée dans un projet professionnel, mais aussi de découvrir de nouvelles techniques et une organisation différente. En effet, l'aspect scientifique du projet et le fonctionnement du laboratoire, totalement différent d'une école d'ingénieur, ont été riches d'enseignement.

Mots-clés : TERAPIX, IAP, EPITA, SCIA, réseaux de neurones, défauts optiques, ACP

2 Introduction

2.1 Contexte du projet

A l'heure actuelle, l'astrophysique* est une science qui tire un avantage certain des progrès de l'informatique, aussi bien au niveau du matériel et des performances croissantes des machines qu'au niveau théorique et de l'informatique avancée. En effet, les relevés* effectués par les télescopes récents produisent de grandes quantités d'images du ciel correspondant aujourd'hui à plusieurs dizaines, voire plusieurs centaines de Teraoctets. Pour pouvoir traiter et exploiter scientifiquement ces données, il est donc nécessaire de disposer de grande capacité de stockage, d'un gros potentiel de calcul ainsi que d'algorithmes performants, rapides et d'une grande fiabilité.

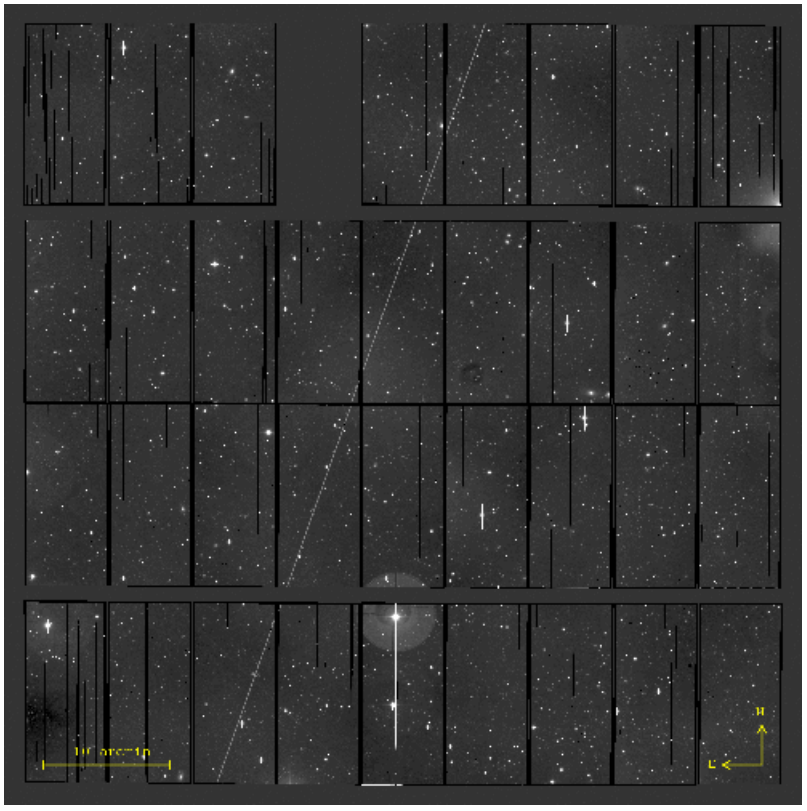


FIG. 2.1 – Image de relevé contenant aigrettes, halos et trainées de satellite

Les images sont en grande partie traitées automatiquement par des logiciels développés spécialement pour les relevés des télescopes, depuis la récupération des images CCD* jusqu'à

leur analyse scientifique.

Les relevés provenant d'instruments à grand champ souffrent de nombreux défauts optiques comme les aigrettes de diffraction et les halos, bien visibles sur l'image 2.1. Ces défauts altèrent le fond des images et contaminent le signal des étoiles et des galaxies qui engendrent des fausses détections ou des mauvaises mesures des formes des objets réels. Pour éviter ces contaminations, tous les relevés actuels produisent manuellement des masques qui suppriment les zones affectées. Dans le cas des grandes images panoramiques, comme celles traitées par Terapix, la procédure manuelle de masquage demande cinq heures par image. La suppression manuelle à ce rythme prendrait 50 000 heures pour le relevé du télescope Canada-France-Hawaii sur lequel travaille Terapix. L'intérêt du projet est donc d'automatiser la suppression des défauts à un rythme 10 à 100 fois plus élevé.

2.2 Sujet du stage

L'objectif du stage est de produire un détecteur automatique de défauts optiques sur des images grand-champ de l'espace profond capable de traiter une image toutes les cinq minutes environ. Il s'agit d'écrire un logiciel générant de manière totalement automatique un masque de pixels correspondant à l'empreinte des défauts étendus les plus courants rencontrés sur les images CCD du ciel profond. Compte-tenu du volume de pixels à analyser, le projet comprendra une étape de réduction de dimensionnalité* des données.

2.3 Contexte de l'entreprise

2.3.1 Institut d'Astrophysique de Paris

L'Institut d'Astrophysique de Paris (IAP), est un laboratoire de recherche du Centre National de la Recherche Scientifique (CNRS) associé à l'Université Pierre et Marie Curie (prochainement École Interne de l'Université Paris 6). Il est l'un des cinq laboratoires de l'*European Association for Research in Astronomy* (EARA), dont il fut l'un des fondateurs.

Laboratoire d'interface entre deux disciplines, l'IAP est constitué de deux unités, l'une (UMR 7095) regroupant des astrophysiciens du secteur des sciences de l'Univers (INSU), l'autre, le GReCO (FRE 2435) étant composée de physiciens théoriciens du secteur des sciences physiques et mathématiques (SPM).

L'Institut bénéficie, de plus, d'une plateforme technique "POLARIS" (POLe Astrophysique de Recherche en traitement de l'Information et Simulation numérique).

L'IAP compte 160 chercheurs, ingénieurs, techniciens, administratifs et doctorants, et accueille régulièrement de nombreux stagiaires et visiteurs étrangers. L'organisation générale du laboratoire est indiquée par l'organigramme de la figure 2.2.

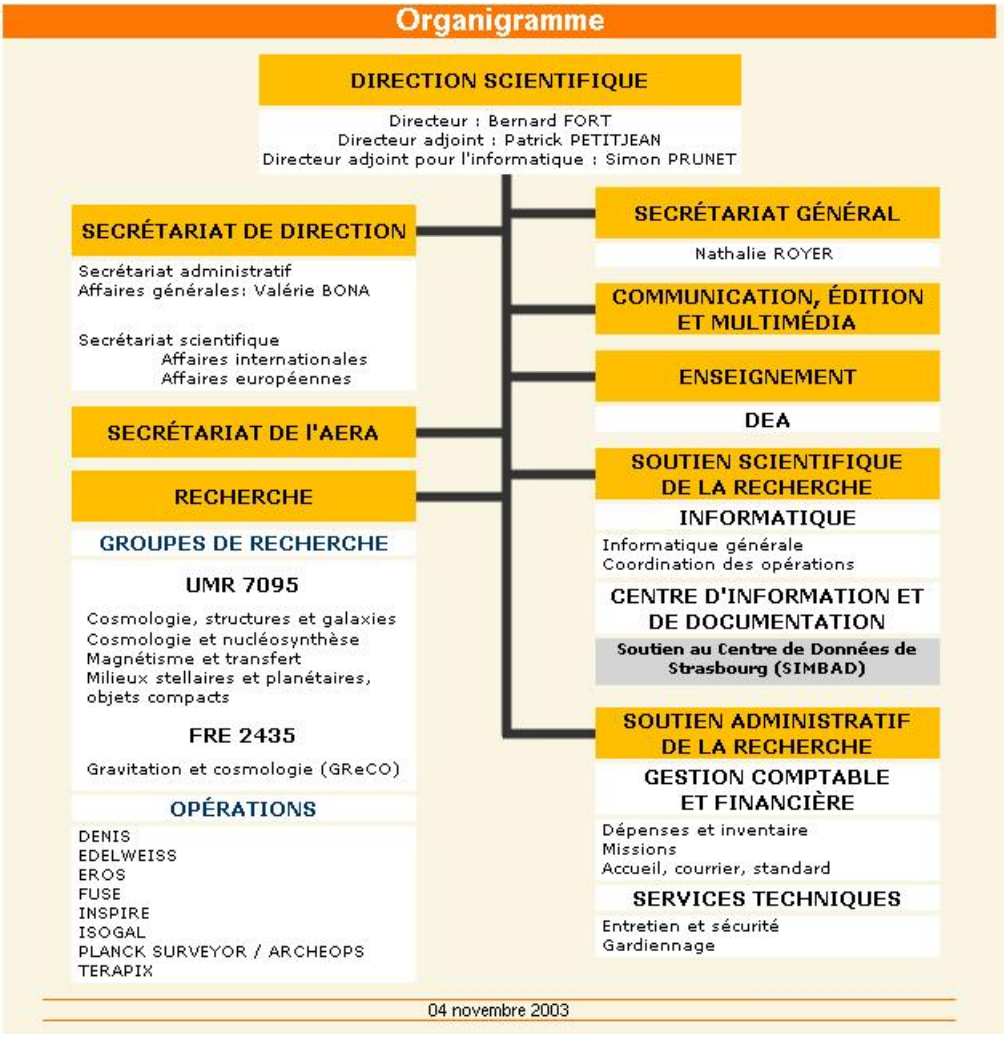


FIG. 2.2 – Organigramme de l'IAP

Parmi les projets dans lesquels est impliqué l'IAP se trouve la gestion des données des caméras CCD panoramiques MegaCam (domaine spectral visible) et WirCam (domaine spectral infrarouge proche) du *Canada-France-Hawaii Telescope* et la production d'images calibrées prêtes à l'exploitation scientifique.

Des informations complémentaires se trouvent sur le site Internet de l'IAP.

2.3.2 CFHTLS

Le *Canada-France-Hawaii Telescope* (CFHT) est un télescope de 3,6 mètres situé au sommet du volcan hawaien Mauna Kea (4200 m). Il est géré par le Canada (42% par le CNRC), la

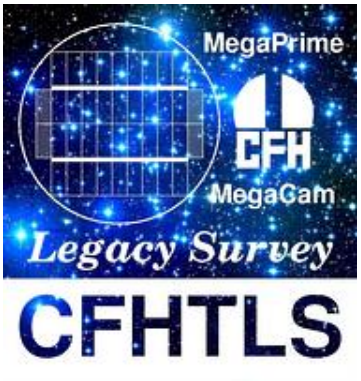
France (42% par le CNRS) et l'Université d'Hawaii.

NATIONAL RESEARCH COUNCIL CANADA
CONSEIL NATIONAL DE RECHERCHES CANADA

CNRS

cea

UNIVERSITY OF HAWAI'I



Pour rester compétitif à l'heure des télescopes de 8 mètres, le CFHT s'est doté d'une caméra CCD panoramique à grand champ de seconde génération, MegaCam. La caméra est composée de 36 CCD de 2000x4500 pixels chacun, ce qui constitue une des plus grandes caméras CCD à mosaique du monde. Avec une étendue de visualisation d'un degré carré, elle couvre une zone quatre fois grande comme la pleine lune. L'autofocus et l'autoguide assurent la meilleure qualité d'image à tout moment. Une série de filtres couvre l'étendue des longueurs d'onde entre 370nm et 1000nm.

Pour assurer le meilleur retour scientifique de cet instrument, les agences scientifiques canadiennes et françaises ont décidé de mettre en place un gigantesque programme d'observation, le *CFHT Legacy Survey* qui se compose de trois parties :

- Le CFHT-LS "shallow", couvrant 1300 degrés carrés sur l'ecliptique et concentré sur les observations trans-Neptunienne et la ceinture de Kuiper. Les temps de pose sont d'environ 5 minutes, avec seulement trois filtres.
- Le CFHT-LS "wide", couvrant 170 degrés carrés dans trois larges champs situés à de hautes latitudes galactiques, dans des zones du ciel hors des poussières absorbantes de notre galaxie ("dust-free"). Le "wide survey" se concentre sur les structures à grandes dimensions de l'Univers (distorsions gravitationnelles cosmologiques, amas de galaxies, quasars ainsi que nouvements propres d'étoiles de la galaxie). Les temps de pose sont d'environ 1 heure par filtre, pour 5 filtres.
- Le CFHT-LS "deep", couvrant quatre zones décorrélées d'un degré carré dans des zones du ciel "dust-free". Le "deep survey" est optimisé pour la détection de la mesure de l'évolution des courbes de lumière des supernovae de Type Ia et l'étude des galaxies à très

grand décalage spectral (redshift). Les temps d'exposition vont de 50 à 100 heures dans chacun des 5 filtres.

Le CFHT-LS fournira une description unique sur la physique du système solaire, l'évolution des galaxies et la courbure de l'univers. Les agences canadienne et française investiront 500 nuits sur 5 ans, à partir de février en février 2003.

Le principal défi technique du CFHT-LS est la gestion des données et le traitement des images. Environ 50 To d'images devront être traités, livrés et archivés sur les 5 ans du programme.

Afin de préparer cette mission, l'Institut National des Sciences de l'Univers et de l'Environnement (INSUE), le Commissariat à l'Energie Atomique (CEA), le Programme National de Cosmologie (PNC) et l'Institut d'Astrophysique de Paris ont mis en place un centre de traitement des images appelé TERAPIX.

2.3.3 Bref historique de l'IAP

C'est le 30 octobre 1936 que l'arrêté de Jean Zay, alors Ministre de l'Education nationale, crée le Service de recherche d'astrophysique, composé d'une station d'observation en Haute-Provence et d'un laboratoire situé Paris pour le traitement et l'étude des documents d'observation. Le Comité de direction du Service décide en janvier 1937 que le laboratoire parisien sera construit sur le site de l'Observatoire de Paris, côté boulevard Arago. En mai 1937, il organise une conférence internationale sur "l'absorption de la lumière dans l'espace interstellaire" et, en octobre, crée les Annales d'astrophysique.

Interrompue pendant la guerre, la construction du bâtiment ne sera terminée qu'en 1952, et ce laboratoire propre du CNRS depuis 1939 prendra alors le nom d'Institut d'Astrophysique de Paris ; la station d'observation, quant elle, deviendra autonome, l'Observatoire de Haute-Provence. Les premiers chercheurs de l'Institut se consacrent l'astrophysique stellaire et l'optique.

En 1946 commencent les recherches en spectrophotométrie solaire et celles sur la diffusion de la lumière . La géophysique est alors présente avec les études sur l'ozone atmosphérique, les aurores, la lumière zodiacale, le ciel nocturne. Un important bureau de calcul, où sont élaborées de nouvelles méthodes de calcul numérique, est créé, ainsi que des ateliers de mécanique, d'optique et d'aluminure de miroirs pour la fabrication de nouvelles générations d'instruments d'observation et pour le perfectionnement des instruments de dépouillement.

Les recherches théoriques se développent en parallèle : théorie des atmosphères stellaires et solaires, intérieur des étoiles et évolution. L'astrophysique théorique prend une place de plus en plus importante au fil des années avec pour objectif prioritaire la compréhension de la nature et de l'évolution des astres.

L'astrophysique fait son entrée dans les programmes universitaires, aux côtés de l'astronomie fondamentale et de la mécanique céleste, et attire de nombreux passionnés de cette science nouvelle. La réputation internationale de l'IAP va se construire, au début des années soixante, avec

les remarquables contributions scientifiques d'Evry Schatzman et de ses nombreux étudiants, qui constituent un groupe d'astrophysique théorique particulièrement vivant et productif.

Dans les années 60, les travaux de laboratoire continuent mais les développements instrumentaux sont progressivement abandonnés au profit de l'utilisation des données spatiales. Au plan théorique, l'accent est mis sur l'étude des atmosphères et de l'environnement stellaire, puis, plus tard, sur celle de notre Galaxie et des galaxies en général.

Dans les années 1990, une véritable dynamique internationale s'instaure : l'IAP obtient, en 1991, le statut de Laboratoire européen et développe les collaborations internationales, avec une politique très active d'accueil de visiteurs. Le volet enseignement prend de l'importance avec l'hébergement du DEA de l' Université Paris 6.

Les équipes se structurent autour de quatre groupes de recherche (Cosmologie, structures et galaxies ; Cosmologie et nucléosynthèse ; Magnétisme et transfert ; Milieux stellaires et planétaires, objets compacts).

Le traitement massif de données - et la simulation numérique afférente - sont renforcés avec la mise en place dune plate-forme technique POLARIS (POLe Astrophysique de Recherche en traitement de l'Information et Simulation numérique), soutenue par la Région Ile-de-France.

Une équipe de physiciens théoriciens du secteur Sciences physiques et mathématiques (SPM) rejoint le laboratoire en 2001. Il constitue un groupe de recherche, le GReCO (Gravitation et cosmologie).

En 2001, l'IAP rejoint l'Université Pierre et Marie Curie (Paris 6) et devient UMR7095.

2.3.4 l'IAP en chiffres

Budget annuel
- 5,9 millions d'euros (masse salariale comprise) avec les contributions de :
  - INSU/CNRS (Grands équipements)
  - CNES (Expériences spatiale
  - DRI/CNRS (Relations internationales)
  - Région Ile-de-France
  - Communauté européenne

Personnel
- 56 chercheurs
- 33 ingénieurs, techniciens et administratifs
- 26 chercheurs associés
- 10 post doctorants
- 18 étudiants en thèse
- 120 visiteurs/an

Publications

– 160 publications/an

Bibliothèque
– 800 périodiques
– 10 000 ouvrage

2.3.5 TERAPIX

TERAPIX (Traitement Élémentaire, Réduction et Analyse des PIXels de megacam) est un centre de réduction de données astronomiques dédié au traitement de larges flux de données provenant des relevés numériques du ciel.

Les principales tâches de TERAPIX sont :
– développer des logiciels de traitement d'images et le pipeline pour MegaCam ;
– développer et fournir des outils pour manipuler de grandes images CCD ;
– faire fonctionner le pipeline de réduction pour produire des images calibrées et des catalogues des images MegaCam acquises au cours des 5 années à venir ;
– fournir une assistance technique et des calculateurs aux utilisateurs MegaCam.

L'equipe TERAPIX est actuellement composée de 5 ingénieurs, 3 astronomes et une secrétaire. Le PI en est Yannick MELLIER.

Des infos compémentaires se trouvent sur le site Internet de TERAPIX.

TERAPIX

http://terapix.iap.fr CNRS / IAP

A suite for processing wide-field images

DOMISSE Laurent, MALAPERT Jean-Christophe, TISSIER Gérard,
BERTIN Emmanuel, MAGNARD Frédéric, MELLIER Yannick, MISSONNIER Gilles,
MORIN Bertrand.

TERAPIX (Traitement Élémentaire, Réduction et Analyse des PIXels de megacam) is an astronomical data reduction centre dedicated to the processing of extremely large data flows from digital sky surveys. Located at the IAP (Institut d'Astrophysique de Paris), its primary tasks are :

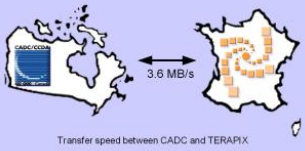
- to develop image processing and pipeline software for MegaCam (the new giant CCD camera of the CFHT telescope in Hawaii);
- to develop and provide tools for handling of large CCD images ;
- to operate the final reduction pipeline to produce calibrated images and catalogues of MegaCam images over the next 5 years ;
- to provide technical assistance and TERAPIX computing facilities to MegaCam users.

This poster presents the different components of the processing suite, chained as represented besides.

Snooppix → DataTransfer → Spica → Snooppix

Snooppix

Perl daemon that downloads automatically data via a http or ftp protocol.



Transfer speed between CADC and TERAPIX

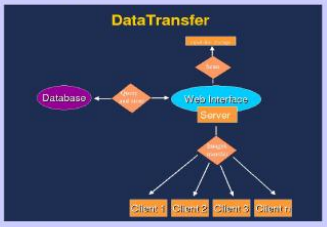
Based on wget (free tool for non-interactive files download from the web), Snooppix can scan a web page and download data sending simultaneous wget on each web page's link.

In order to avoid duplication, Snooppix stores in a DBM file the downloaded filenames. A log file is also available to check download status. Snooppix can be managed from web or perl/TK interface.

Snooppix uses a configuration file to load user preferences. This file describes snooppix's connection parameters about server (web or ftp) and informations about client (directory where are downloaded data, ...).

DataTransfer

Set of perl and php scripts that dispatches data and managing transfers on a cluster.



In order to transfer and sort out quickly images coming from CADC on our cluster, we have developped a software based on php and perl scripts which allows to dispatch data across a cluster.

DataTransfer is managed from a web page. So, you can transfer images by simple clicks. DataTransfer is not only a tool for data transfer, it can also display :

- FITS header
- files distribution across nodes.
- transfer history
- free disk space on each nodes

Spica

Processing tools to produce calibrated data from large astronomical images

To reduce MegaCam data (the world biggest CCD camera : one square degree), we have designed a pipeline software named Spica (Software Pipeline for Images and
Catalogs Analysis). It produces calibrated images (astrometry, photometry and coaddition/resampling, catalog extraction). It can be installed as a standalone application or as a client/server one.
Once the data are stored on the cluster, Spica checks their integrity and loads them in the SQL database (see RDBix). MegaCam images are uncompressed if needed and sorted out by filter and runid.
The quality of each image is controlled by Qualityfits assessment (see companion poster) and this information is used to select which data could be used by Spica.
There are two ways to use it :

Automated web mode

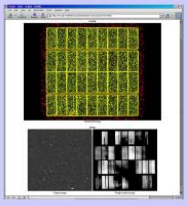
Interactif web mode

On each node, Perl scripts manage input/output from each pipeline application, database storage and send command line scripts to the spica daemon which run processing according to priority rules. Generated files are sent to the output storage disk. Input data are processed with rules written in configuration files.

A web interface allows authorised users to reduce their own data. They can choose :
1) which data to process according to their preferences,
2) which applications (including configurations) to run.
All steps are recorded in the database for further analysis.

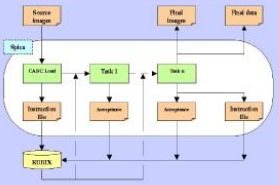
Administration interface written in PHP can check the pipeline status. It can also display spica produced data.

The users have their personal web spaces allowing to create a customised environment. Forms help users to build their setup processes.



RDBix

RDBIX is the memory of the whole pipeline. It is a MySQL relational database designed for the storage of catalogues and metadata produced by the processing software. RDBIX organizes these metadata to maintain history of the images processing.



Each step of the pipeline process communicates with RDBIX to get data about images (coordinates, storage path, ...) and to inform RDBIX, via a piece of software called dbclient, about images modifications. An XML file, called instruction file, serves as a support for this communication with RDBIX. Other data, such as acceptance files (quality data) are also integrated into RDBIX.

RDBIX is based on a completely dynamic design, which allows the database to grow without heavily impacting the performances.
Finally, RDBIX has been designed to run in a distributed environment : each node of the cluster runs a client version of dbClient (and its own pipeline package), while the master node runs a server version of dbClient and hosts the MySQL RDBIX database which is used by every node.

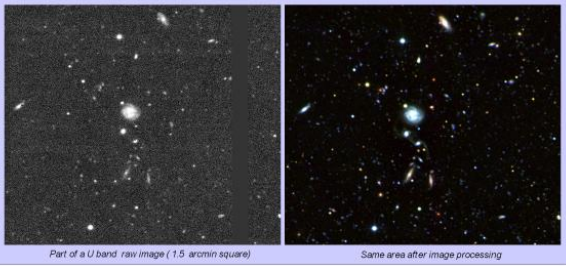
Results

Typical processing of MegaCam images.

On the left is a raw U band 860 seconds exposure. On the right, a stacked U+G+R image composed of 44 Megacam images processed with Swarp software and corresponding respectly to an integration time of 9580 s, 9500 s and 7000 s.

Processing time is around 40 hours on a dual XP1800+ computer (will be down to 30 hours with Opteron processor).

Color image generated by Stiff which is able also to convert image from FITS format to PNG.



Part of a U band raw image (1.5 arcmin square) *Same area after image processing*

FIG. 2.3 – Fonctionnement de TERAPIX

Le pipeline TERAPIX (cf poster 2.3 présenté à une conférence internationale, ADASS IV en 1994) est composé de plusieurs outils détaillés à la section 2.6.2.

2.4 Etat des connaissances sur le sujet au sein de TERAPIX

L'équipe TERAPIX travaille depuis plusieurs années sur le pipeline et le développement a été entièrement assuré par ses membres. Le logiciel créé à l'issu du stage doit s'intégrer directement au sein du pipeline Spica et constituer une des étapes de traitement des images MegaCam. TERAPIX est donc l'entité la plus à même d'encadrer le stage et d'orienter le travail durant 6 mois.

M. Emmanuel BERTIN, mon maître de stage, a développé durant sa thèse SExtractor, un logiciel d'extraction de sources provenant d'images des plaques photographiques des télescopes de Schmidt ou des caméras CCD panoramiques. Le logiciel utilise des réseaux de neurones et est optimisé pour le travail sur des images astronomiques. M. BERTIN a donc déjà une connaissance approfondie de la nature des images, de leurs spécificités vis-à-vis des techniques de traitement du signal ; mais également du fonctionnement des réseaux de neurones et de leurs possibilités. Il a donc toute l'expertise pour me guider durant la période du stage.

2.5 Etat des connaissances du stagiaire sur le sujet

Mes connaissances sur le sujet touchent principalement au traitement de données et aux réseaux de neurones. Concernant l'aspect scientifique et astrophysique du projet, mon savoir se résume à de l'amateurisme.

Mes motivations concernant ce stage sont tout d'abord de découvrir plus précisement l'astrophysique professionnelle qui est un domaine auquel j'attache beaucoup d'intérêt. Ensuite, mettre en pratique les cours dispensés durant la dernière année par M. Yves-Jean Daniel sur les réseaux de neurones et apporter mes compétences de développement à une équipe dont les qualités sont principalement les connaissances astrophysiques.

2.6 Contexte précis de travail

2.6.1 Matériel

Dès mon arrivée au sein de TERAPIX, une machine m'a été fournie avec une totale liberté de configuration et d'utilisation. J'ai conservé le système d'exploitation installé, Mandrake Linux 9.2. Tous les outils nécessaires ont été mis à ma disposition.

Athlon XP 1800+
256Mo de RAM
Disque dur de 90Go
ATI Rage Pro AGP 4x
Carte ethernet 3com 3c905C-TX (fast-etherlink)
Mandrake Linux 9.2

TAB. 2.1 – Configuration du poste de travail

TERAPIX possède un ensemble de machines bien organisé pour optimiser les différents traitements à effectuer, d'abord pour le pipeline, mais aussi pour le maintien d'un site Internet, le travail personnel et le développement, le transfert avec le CADC, etc. La figure 2.4 montre l'organisation globale du parc informatique de TERAPIX.

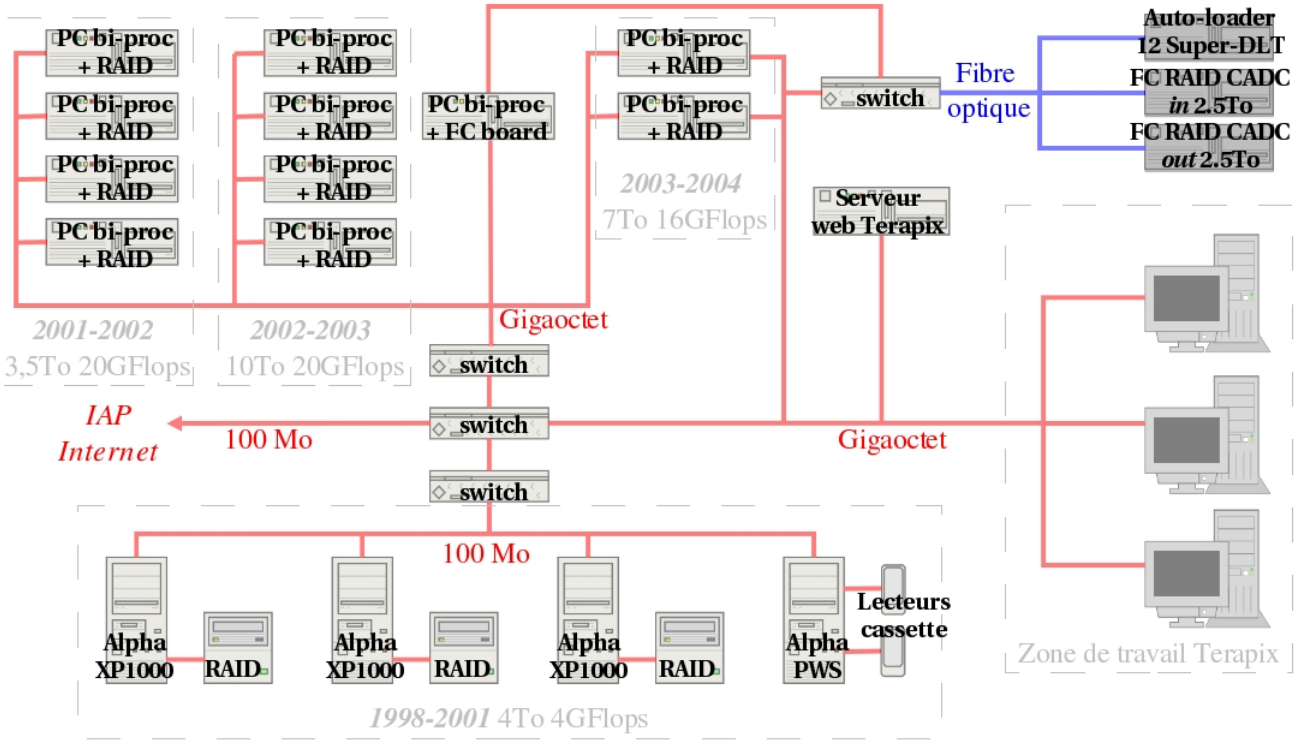


FIG. 2.4 – Le réseau de TERAPIX

En plus de ce parc, Terapix est relié au réseau de l'IAP qui comprend un grand nombre de machines et du matériel divers comme indiqué dans le tableau 2.2, entièrement accessible et utilisable.

Connexion Internet haut-débit
Scanner
Imprimante laser couleur
Imprimante laser noir et blanc
Imprimante laser couleur A0
Photocopieuse

TAB. 2.2 – Matériel divers

2.6.2 Logiciels

Les principaux logiciels mis à ma disposition (en dehors évidemment des logiciels sous license GPL, tels que autotools, emacs, gcc, open office) étaient Spica, Panorapix et DS9. L'interface web de Spica me permettant de trouver sur le cluster des images appropriées à l'exécution de tests. Panorapix et DS9 sont deux outils de visualisation d'images au format particulier FITS, le premier étant développé par Nicolas de Coussemaker [dC99] et Frédéric Magnard spécialement pour TERAPIX.

Toutes les informations sur Spica se trouvent sur le site Internet de Terapix, la documentation de Panorapix sur le site Internet dédié au logiciel (voir 10).

2.6.3 Rapports professionnels

Les différents membres de l'équipe ont été disponibles en permanence pour me donner des explications sur les diverses fonctionnalités du pipeline dans lequel doit s'inscrire le projet.

L'organisation de l'équipe m'a immédiatement donné une certaine autonomie tout en contrôlant régulièrement l'état d'avancement du projet.

Les personnes compétentes étaient disponibles pour répondre à mes questions. Pour les points spécifiques, on peut citer Laurent Domisse pour l'utilisation de Spica, MM. Bertin et Mellier pour l'astrophysique et les mathématiques, M. Magnard pour l'organisation du parc informatique et l'utilisation de Panorapix et enfin M. McCracken pour la rédaction de documents en anglais.

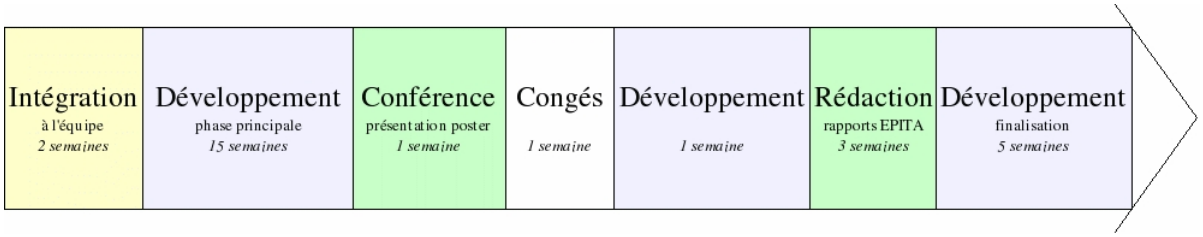
3 Aspects organisationnels

3.1 Découpage du projet

3.1.1 Chronogramme global

Le chronogramme 3.1 donne le découpage global du stage selon les différentes étapes rencontrées.

- La première période d'*intégration* correspond à la découverte de l'équipe, de son fonctionnement ainsi qu'à la prise en main des différents outils nécessaires au développement du projet.
- Une fois l'intégration terminée, le *développement* du projet en lui-même a pu commencer. C'est la période la plus longue et la plus importante puisqu'elle a conditionné le déroulement des étapes suivantes, notamment la préparation pour la conférence.
- En effet, la période *conférence* correspond à la semaine passée en Italie pour présenter le projet à la communauté scientifique. Cette présentation dépendait de l'état d'avancement du projet.
- A la suite de la conférence, une courte période de *développement* a précédé la *rédaction* des documents de suivi du stage, c'est-à-dire, le présent rapport de stage, le résumé en anglais et le compte-rendu XML.
- Puis, jusqu'à la fin du stage, le *développement* s'est poursuivi dans le but d'obtenir le meilleur logiciel possible.



TAB. 3.1 – Chronogramme global

3.1.2 Chronogramme hebdomadaire

Le chronogramme 3.2 présente de façon plus détaillée le déroulement du stage selon les étapes définies dans le chronogramme global de la figure 3.1.

Janvier	
Semaine 1	Prise de contact et prise en main des outils (spica, panorapix, format de fichiers FITS)
Semaine 2	Développement d'applications de test des librairies cfitsio et CCFits
Semaine 3	Début du développement du projet, mise en place d'une arborescence autotools
Semaine 4	Développement de la partie de gestion des fichiers FITS avec cfitsio
Semaine 5	Etude des différentes méthodes de réduction de dimensionnalité.
Février	
Semaine 6	Développement de l'étape de réduction de dimensionnalité par ACP (Analyse en composantes principales). Etude de différentes méthodes d'apprentissage.
Semaine 7	Tests des Support Vector Machines à l'aide du logiciel SVM Light. Choix de la méthode d'apprentissage pour les perceptrons.
Semaine 8	Développement de la partie apprentissage.
Semaine 9	Développement d'un simulateur de défauts et de la partie de création de masques à partir du réseau entrainé.
Mars	
Semaine 10	Développement d'une gestion correcte des erreurs.
Semaine 11	Premiers tests.
Semaine 12	Modification de la méthode envisagée pour s'approcher d'un système de vision hiérarchique.
Semaine 13	Développement d'une structure hiérarchique pour l'apprentissage.
Semaine 14	Tests complémentaires.
Avril	
Semaine 15	Modification de la structure hiérarchique pour limiter la perte d'information entre les niveaux.
Semaine 15	Tests et rectifications de code.
Semaine 16	Réalisation de tests de configuration des paramètres du système
Semaine 17	Poursuite des tests. Création du poster pour la conférence ADA III en conséquence (Cf Annexe A.1).
Semaine 18	Conférence ADA III à Sorrento en Italie
Mai	
Semaine 19	Congés

Semaine 20	Modification de la structure hiérarchique du système de vision (cf. 4.2.5). Départ de l'équipe Terapix pour le meeting CFHTLS à Victoria (Canada). Début de la rédaction du rapport de stage.
Semaine 21	Rédaction du résumé en anglais, du rapport XML et du rapport de stage.
Semaine 22	Rédaction du rapport de stage. Retour de l'équipe Terapix.
Juin	
Semaine 23	Rédaction du rapport de stage.
Semaine 24	
Semaine 25	
Semaine 26	
Semaine 27	
Juillet	
Semaine 28	

TAB. 3.2: Chronogramme hebdomadaire

3.2 Respect des délais et critique

Il est difficile de parler de respect des délais étant donné qu'aucun planning n'avait été défini préalablement. Le seul découpage réel correspond à la conférence ADA III de la semaine 18 car il fallait que le projet soit suffisamment avancé pour pouvoir présenter un poster décent. La date de la conférence et l'inscription du projet à la liste des posters ont été définies durant le mois de février, et constituaient de fait un premier délai à respecter.

Cette étape intermédiaire a été utile car elle m'a incité à développer une version fonctionnelle du projet, même si ce n'était pas la version finale.

Il y a donc eu une période assez intense juste avant la conférence ADA III durant laquelle je devais présenter un poster décrivant le fonctionnement du projet (Cf. Annexe A.1). Le poster devant contenir des images créées par le programme, il a fallu réaliser de nombreux entraînements des réseaux de neurones avec différents paramètres pour obtenir les meilleurs résultats possibles. L'apprentissage demandant un temps non négligeable, il a été nécessaire de passer des heures à exécuter le programme et à attendre les résultats. Finalement, le poster a été imprimé au dernier moment avec des images satisfaisantes. Pour se faire, à certains paramètres, au lieu d'être calculés ou testés, ont été affectés des valeurs approximatives que l'expérience du Dr. Bertin a permis de définir. Suite à la conférence, les estimateurs statistiques définissant les paramètres ont été intégrés dans le code du projet.

Globalement, le découpage du projet n'a pas été pénalisant dans la mesure où le temps imparti au projet a été correctement exploité, la conférence ayant servi de moteur vers la moitié de la période de développement.

3.3 Nature et fréquence des points de contrôle

3.3.1 Tera-meeting

Tous les vendredis après-midi, une réunion de quelques heures permet à chaque membre de l'équipe de faire le point sur l'avancement de son travail. C'est donc l'occasion pour chacun de se tenir informé de l'état global du programme. Ces "Tera-meeting" sont aussi l'occasion de faire des démonstrations des logiciels lorsque cela s'avère utile ou nécessaire. Par conséquent, j'ai profité de ces réunions pour comprendre autant que possible le fonctionnement de l'équipe, découvrir les logiciels utilisés, les projets développés et saisir le programme scientifique mené par l'équipe.

Ces réunions hebdomadaires permettaient de renseigner mon maître de stage ainsi que le PI de Terapix de ma progression dans le développement de l'outil. Cependant, cela ne représentait pas vraiment un point de contrôle.

3.3.2 Séance de réflexion

En effet, d'autre part, le Dr. Bertin et moi-même avions régulièrement des discussions sur l'évolution des résultats et sur la pertinence des procédés choisis. Et lorsque le besoin s'en faisait ressentir, nous organisions une séance de réflexion avec le Dr. Mellier afin de définir un véritable contrôle de l'avancement du projet et de nouvelles directions à prendre le cas échéant. Ce genre de séance de réflexion a eu lieu environ une fois par mois, définissant pratiquement à chaque fois de nouvelles pistes à explorer.

Ces pistes ont parfois été abandonnées par la suite mais avaient dans tous les cas un certain potentiel pédagogique et scientifique. En effet, l'exploration de telles pistes me permettaient de découvrir de nouvelles techniques, éventuellement de les mettre en pratique et de déterminer pourquoi elles convenaient ou non. C'est notamment le cas des recherches menées sur le système de vision.

Dans tous les cas, les séances de réflexion permettaient de relancer le projet et de dynamiser le travail de développement.

3.3.3 Quelques points de contrôle importants

La première séance de réflexion a eu lieu la semaine de mon arrivée, lorsqu'il a fallu m'expliquer en détail les objectifs techniques et scientifiques du projet, ses difficultés et ses contraintes. Cette réunion m'a donné le point de départ pour mes premières recherches.

La deuxième séance de réflexion importante a eu lieu durant la 12ème semaine, après l'obtention de premiers résultats concluants à partir d'un seul réseau de neurones. Dans le but de calquer encore un peu plus la biologie, le projet s'est orienté vers un modèle de vision hiérarchique basé sur plusieurs réseaux de neurones.

3.3.4 Conférence ADA III

La conférence *Astronomical Data Analysis* (ADA) s'est déroulée en Italie, près de Sorrento. Sur les trois jours de la conférence, de nombreux intervenants se sont succédés afin de présenter leurs recherches dans différents domaines : mathématiques, statistiques, data mining, maîtrise d'ouvrage, etc.

Parallèlement à ces présentations orales, une vingtaine de posters étaient affichés dans une salle prévue à cet effet. Notamment celui que j'avais réalisé pour présenter le projet et qui est reproduit en annexe A.1.

3.4 Procédures en urgence

La seule réelle date butoir du projet était la fin de la période du stage. Celle-ci n'étant pas atteinte à la date du rendu du rapport, aucune procédure d'urgence n'a du être prise. A priori, il n'y aura aucunement besoin de ce genre de mesure durant les semaines restantes.

4 Aspects techniques

4.1 Techniques envisagées

4.1.1 Gestion des fichiers FITS

Les données de MegaCam, de par leur grande taille, arrivent dans un format spécifique utilisé par la majorité de la communauté scientifique astrophysique, le format FITS.

```
SIMPLE  =                    T / Standard FITS
BITPIX  =                   16 / Bits per pixel (not applicable)
NAXIS   =                    0 / No image data with primary header
EXTEND  =                    T / File contains extensions
NEXTEND =                   36 / Number of extensions
COMMENT
COMMENT Observation Summary
COMMENT -------------------
CMMTOBS = 'N/A     '
CMMTSEQ = 'LDP7[2/7] - Ref + [-15.0 -30.0]'
OBJECT  = 'W1.+3+2 '
OBSERVER= 'QSO Team'
PI_NAME = 'Wide_Synoptic CFHTLS'
RUNID   = '03BL02  '
FILENAME= '720095p00.def'      / Current filename after processing
PATHNAME= '/data/piko/megacam/03BQ03-N14' / Original directory name at acquisiti
DATE    = '2003-10-01T12:25:02' / UTC Date of file creation
HSTTIME = 'Wed Oct 01 02:25:02 HST 2003' / Local time in Hawaii
IMAGESWV= 'CFHT DetCom v4.36 (Aug 29 2003)' / Image creation software version
COMMENT file 720095o, raster FULL, etype OBJECT, etime 620, filter filter_name
COMMENT  Fully Characterized and Analysed by Elixir @ CFHT
DETECTOR= 'MegaCam '
COMMENT
COMMENT General Information
...
```

FIG. 4.1 – Extrait du premier en-tête d'un fichier MEF

Cette norme définit les images grâce à un en-tête contenant les informations numériques

de l'image (dimension, profondeur, ...) mais aussi les informations scientifiques (instrument, position dans le ciel, temps d'exposition...). Les données suivent immédiatement l'en-tête. Le format FITS permet de créer des fichiers appelés Multi-Extensions Fits (MEF) permettant de regrouper plusieurs images dans un seul fichier. C'est le cas des images CFHTLS qui sont composées de 36 extensions, une par CCD.

Le développement d'applications utilisant des fichiers FITS s'appuie sur deux librairies, l'une en C nommées cfitsio et l'autre en C++ nommée CCFits, surcouche objet de la première. Les informations sur le format et les librairies se trouvent sur le site Internet FITSIO référencé au chapitre 10.

cfitsio propose une gestion complète des fichiers FITS et MEF ainsi qu'une gestion d'erreur à l'aide d'une variable. CCFits propose une surcouche intégrant tous les aspects de modélisation objet du langage C++ ainsi qu'une gestion d'erreur utilisant les exceptions.

4.1.2 Simulateur

Pour pouvoir entraîner le système d'apprentissage supervisé, il est nécessaire d'avoir un jeu de données suffisamment grand pour obtenir une bonne généralisation. Les données utiles dans le cas du projet sont les images MegaCam en entrée du système et les masques correspondant à chaque image comme modèle d'apprentissage.

Si un grand nombre d'images MegaCam sont disponibles, ce n'est pas le cas des masques. En effet, même si certains masques sont créés manuellement, leur format n'est pas adapté aux besoins du logiciel. Tous les défauts sont identifiés de la même façon dans un autre format que le FITS.

Le problème est alors de créer avec facilité et rapidité un jeu d'entraînement conséquent. Deux possibilités sont envisageables, la création manuelle de masques pour obtenir un jeu d'entraînement correspondant exactement aux données réelles ou bien le développement d'un simulateur de défauts créant simultanément des défauts sur une image MegaCam et le masque correspondant au format FITS. Si la deuxième méthode est évidemment plus rapide et moins pénible, elle comporte tout de même l'inconvénient de constituer un simulateur de défauts idéalisés, c'est-à-dire plus ou moins proche des données réelles. D'un autre côté, il est difficile de collecter un échantillon non-simulé de volume suffisant.

La majorité des défauts apparaissant sur les images astronomiques sont provoqué par des phénomèmes de diffraction* sur le télescope. C'est le cas des aigrettes. Certains phénomèmes de réflexion sont eux responsables des halos.

Halos

Les halos sont des disques lumineux provoqués par les objets très brillants. Ce sont des réflexions des étoiles brillantes sur le capteur CCD puis sur les optiques qui se réflechissent à nouveau vers le CCD. Sur l'image 2.1, on distingue également des discontinuités du halos dues aux bras de support du miroir du télescope.

Aigrettes

Les aigrettes de diffractions sont également visibles autour des objets brillants. Elles sont provoqués par les palettes d'araignée tenant le miroir secondaire du réflecteur. L'effet est particulièrement évident pour de longs temps d'exposition. Le nombre d'aigrettes (comme le nombre de discontinuités des halos) dépend du nombre et de la position géométrique des palettes, chacune provoquant deux défauts à 180^o d'écart.

Trainées de satellite

Les trainées de satellite sont de longues traces lumineuses traversant les images généralement de part en part (voir 2.1). Ces trainées sont visibles car les satellites ont le temps de parcourir le champ du téléscope durant les poses longues.

4.1.3 Réduction de dimensionnalité

La réduction de dimensionnalité est une des deux phases importantes du projet. En effet, elle doit permettre de représenter le plus fidèlement possible les données, pour une utilisation efficace durant l'apprentissage, tout en réduisant au maximum la taille nécessaire à la représentation de l'information. Plusieurs possibilités, éventuellement complémentaires ont éte envisagées.

Tout d'abord une simple étape de réduction de l'échelle de l'image en moyennant les valeurs d'un bloc de pixels pour en obtenir qu'un en sortie. En effet, les images d'origine sont largement suréchantillonnées, la perte d'information serait donc minime. Implicitement, cela signifie que ke redimensionnement est fait à une échelle proche de la réponse impusionnelle.

Ensuite, une étape de soustraction du fond de ciel et de réduction de la dynamique de l'image afin d'optimiser le contenu des données en vue du traitement par le réseau de neurones.

Enfin, une projection des images dans une autre base doit permettre de réduire le nombre de paramètres nécessaires pour représenter l'information significative des images. On peut penser à une transformée de Fourier, une Analyse en Composantes Principales (ACP) ou une Analyse en Composantes Indépendantes (ACI). Il est également important de déterminer s'il est préférable d'effectuer la projection dans l'espace des attributs ou dans l'espace des données.

4.1.4 Apprentissage

Pour effectuer la reconnaissance des défauts sur les images, les méthodes statistiques ou mathématiques existantes, bien qu'efficaces dans un certain nombre de cas, n'ont pas été retenues. En effet, le but du projet était de produire un outil générique, capable de traiter de nouveaux types de défauts sans nouvelles étapes de développement.

L'idée était donc d'utiliser un système d'apprentissage qui fournisse un traitement générique. En effet, à partir d'exemples de masques, éventuellement générés par un simulateur, il faut que le système puisse être capable de généraliser l'information. C'est le cas des réseaux de neurones.

Pour effectuer cet apprentissage, deux types de réseaux semblaient être adaptés :

- les perceptrons, pour leur simplicité de mise en œuvre et leur rapidité d'exécution ;
- les machines à support de vecteurs (SVM), pour les très bons résultats obtenus dans les récentes utilisations de ce modèle.

4.1.5 Système de vision humain

Une des modélisations les plus courantes du système de vision humain est le modèle de Thorpe.

Le schéma 4.2 illustre le fonctionnement global du cerveau selon le modèle de Thorpe. Les signaux provenant de l'œil suivent un parcours précis pour parvenir à l'analyse des images.

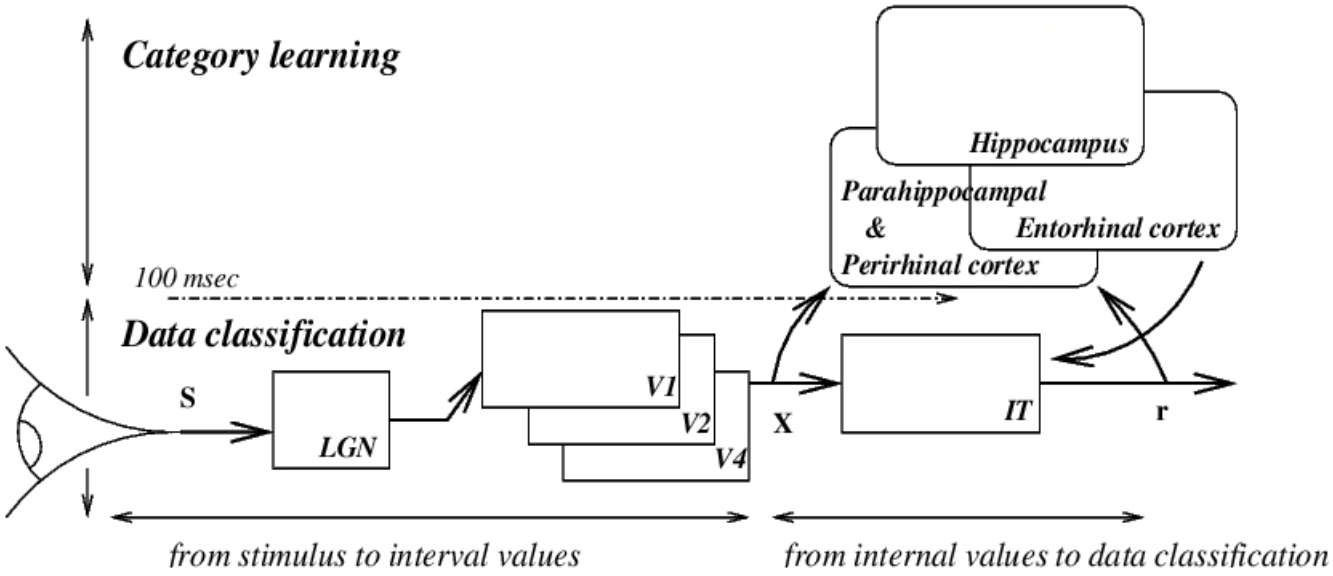


FIG. 4.2 – Vue simplifiée du modèle de Thorpe

Des expériences récentes ont montré que le cerveau humain est capable de discriminer des objets de leur environnement en 100 ms, c'est-à-dire avant de les avoir réellement analysés. Le modèle de Thorpe est compatible avec ces observations. Il semblerait en fait que la classification de données mène à l'analyse visuelle et non le contraire. Ce qui signifie qu'un objet est reconnu avant que sa silhouette ou sa forme ne soit perçue.

Sur le schéma 4.2, on distingue deux parties qui sont la classification des données (*data classification*) et l'apprentissage de catégories (*category learning*). La classification est effectuée rapidement par les aires V1 à V4 du cerveau avant d'être plus précisément analysées par l'hippocampe et d'autres zones du cerveau afin d'obtenir une classification définitive plus détaillée. Des explications plus précises se trouvent dans [VC03].

Le parallélisme qui nous intéresse dans le développement d'un système de vision sur des images astronomiques correspond aux deux parties du schéma de Thorpe. Tout d'abord une classification rapide permettant de discriminer les informations (un seuillage par exemple), puis une analyse des informations ainsi discriminées pour déterminer leur nature. En l'occurrence, pour les images astronomiques, s'il s'agit d'un défaut ou non. Le système de vision devrait donc comprendre deux phases, une étape de classification puis une étape d'analyse de la classification. Plusieurs modèles sont détaillés dans la partie 4.2.5.

4.2 Solution retenue, justification

4.2.1 Gestion des fichiers FITS

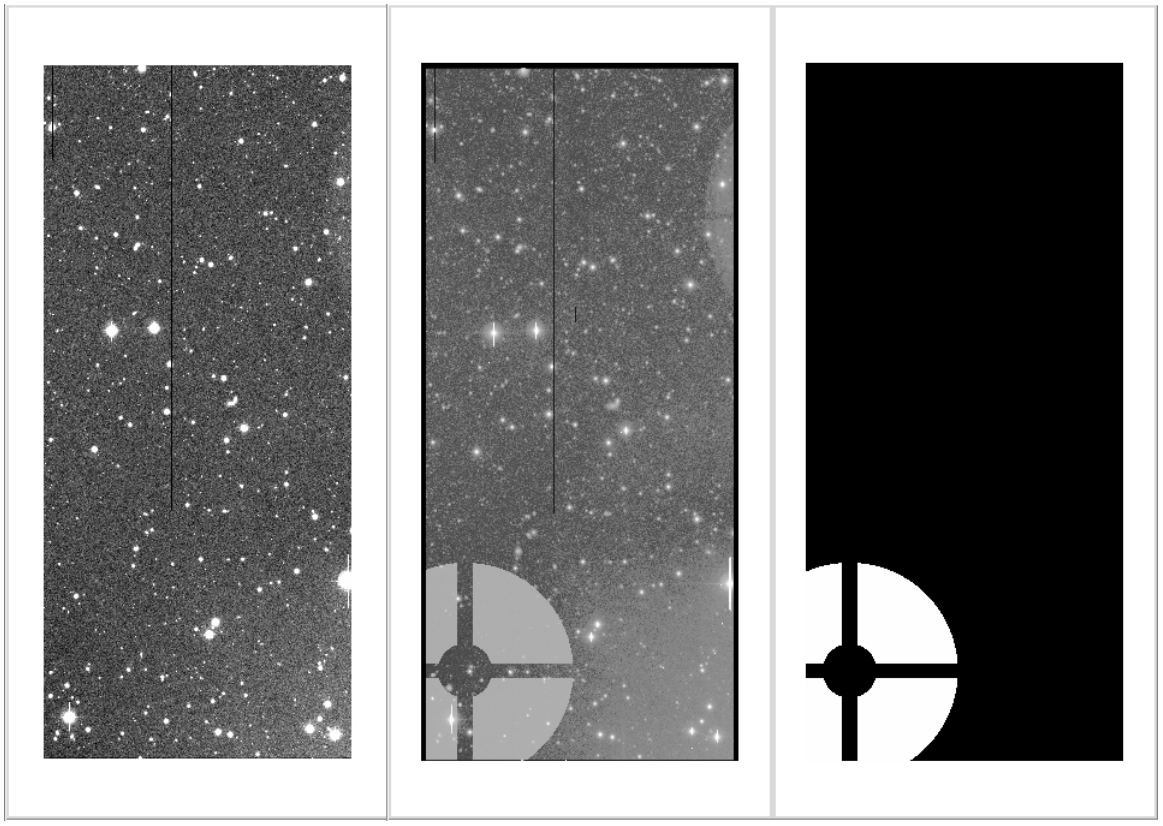
L'utilisation faite des fichiers FITS par le projet est minime, elle concerne principalement la lecture et l'écriture de MEF. Pour cela, la librairie C est amplement suffisante et la surcouche C++ est superflue. Une simple classe a été codée pour répondre aux besoins du projet. La gestion des erreurs s'intègre plus facilement avec la librairie cfitsio qu'avec les exceptions de la librairie CCFits.

4.2.2 Simulateur

Etant donné le cadre du projet, il est apparu indispensable de créer un simulateur de défauts. En effet, l'objectif étant d'automatiser la création des masques, demander une étape de construction manuelle paraît inapproprié.

Le simulateur utilise des fichiers source au format MEF et génère de nouvelles images :
- Un MEF identique au MEF d'origine sur lequel a été ajouté les défauts demandés.
- Un MEF contenant uniquement les défauts et correspondant au masque du premier MEF généré.

Le simulateur fonctionne de façon générique, la seule partie variable étant la routine de création des défauts. Il en existe déjà deux, celle de création de halos et celle de création de trainées de satellites. La figure 4.3 montre les images produites par le simulateur de halos.



a. *b.* *c.*

FIG. 4.3 – a. Image originale ; b. Image simulée avec réduction de dynamique ; c. Masque correspondant à l'image *b*

Un point important à souligner est que le simulateur fonctionne également avec une étape de réduction de dimensionnalité, i.e. un redimensionnement et une réduction de la dynamique de l'image. En effet, pour obtenir des images simulées proches des images réelles qui seront traitées par le système d'apprentissage, les étapes de prétraitement doivent être les mêmes. La section suivante explique les procédures de redimensionnement et de compression de dynamique.

4.2.3 Réduction de dimensionnalité

La solution retenue pour la réduction de dimensionnalité se décompose en trois phases :

1. Redimensionnement ;
2. Réduction de la dynamique de l'image ;
3. Analyse en Composantes Principales (ACP) dans l'espace des données.

L'ACP n'est pas utilisée par le simulateur car elle n'est pas utile dans ce cas. En effet, l'intérêt de cette analyse est de représenter au mieux les images en minimisant l'information pour réduire le nombre d'entrées du système d'apprentissage. Le simulateur traite uniquement les pixels.

Redimensionnement

La première étape est réalisée simplement en calculant un pixel construit avec la valeur moyenne d'un bloc de taille donnée. La figure 4.4 montre le calcul de la valeur d'un pixel r à partir d'un bloc de $N \times N$ pixels p. Ce calcul étant effectué pour tous les blocs de $N \times N$ pixels de l'image d'origine, le résultat est une image de dimension réduite par N sur chaque axe, ce qui correspond à une réduction totale de $N \times N$.

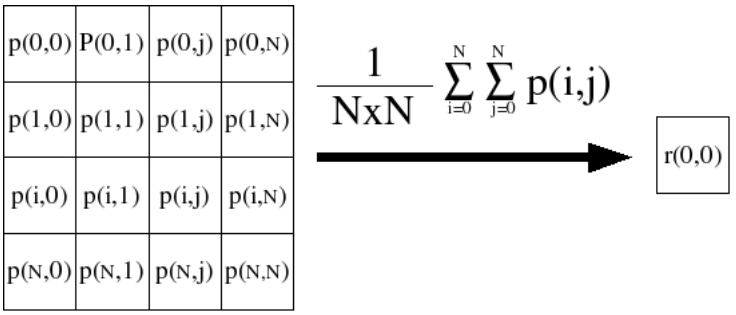


FIG. 4.4 – redimensionnement d'un bloc par N

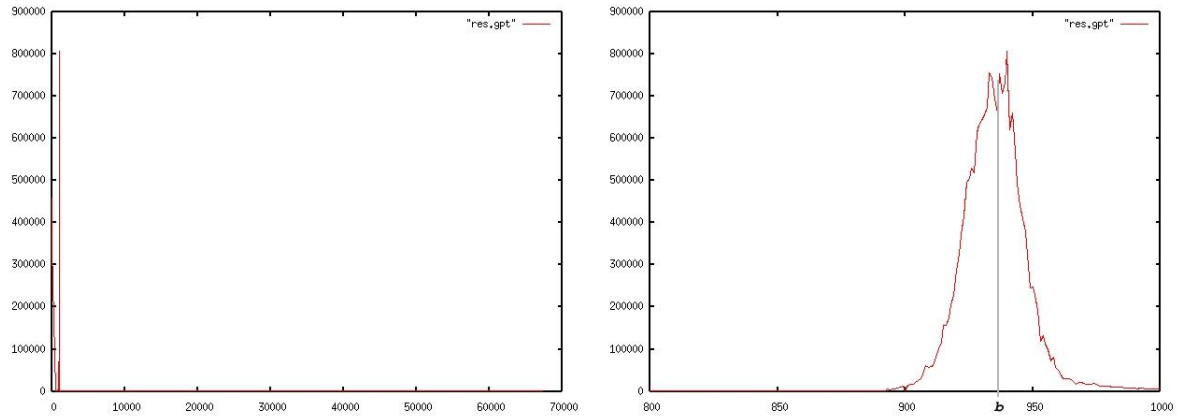
Réduction de la dynamique de l'image

La seconde étape comprend elle-même plusieurs opérations successives :

1. Calcul de la valeur b du fond de ciel de l'image. Comme le montre la figure 4.5, b correspond à la valeur la plus élevée de l'histogramme de l'image. La méthode utilisée est expliquée en annexe B.1.
2. Soustraction de la valeur du fond de ciel à chaque pixel de l'image $\forall x, y(x) = x - b$;
3. Calcul de la déviation standard du bruit de fond σ ;
4. Compression à partir de la fonction de transfert suivante :

$$z(y) = \frac{y}{|y|} \ln\left(1 + \frac{|y|}{\sigma}\right)$$

où $y = y(x)$ est la valeur du pixel de l'image d'origine après soustraction du fond de ciel et σ la déviation standard du bruit de fond.



a. *b.*

FIG. 4.5 – a. Histogramme d'une image de MegaCam. b. Partie de l'histogramme *a.* et valeur de *b.* (Abscisses : intensité du pixel (0 noir ; 65 535 blanc). Ordonnées : nombre de pixels dans l'image)

Evidemment, pour récupérer la dynamique originale de l'image, la fonction suivante est utilisée :

$$y(z) = \sigma \frac{y}{|y|} \exp |y|,$$

$y(z)$ étant la valeur du pixel avec soustraction du fond de ciel.

Analyse en Composantes Principales

L'ACP se déroule en plusieurs phases :

1. Décomposition de l'image en blocs de $M \times M$ pixels ;
2. Transformation de l'image en une matrice telle que chaque ligne corresponde aux $M \times M$ pixels d'un bloc de l'image. La matrice obtenue contient $M \times M$ colonnes et autant de lignes que de blocs de $M \times M$ pixels se trouvant dans l'image, i.e. K. Cette transformation est l'illustre la figure 4.6 ;
3. Calcul des composantes principales en utilisant l'algorithme 4.7. Les vecteurs propres obtenus constituent une base de Karhunen-Loève (KL) comme illustré par la figure 4.8 ;
4. Projection de l'image d'origine sur la base des m premières composantes (Transformée de KL).

Typiquement, $M = 8$ ou 16 et $m = 16$.

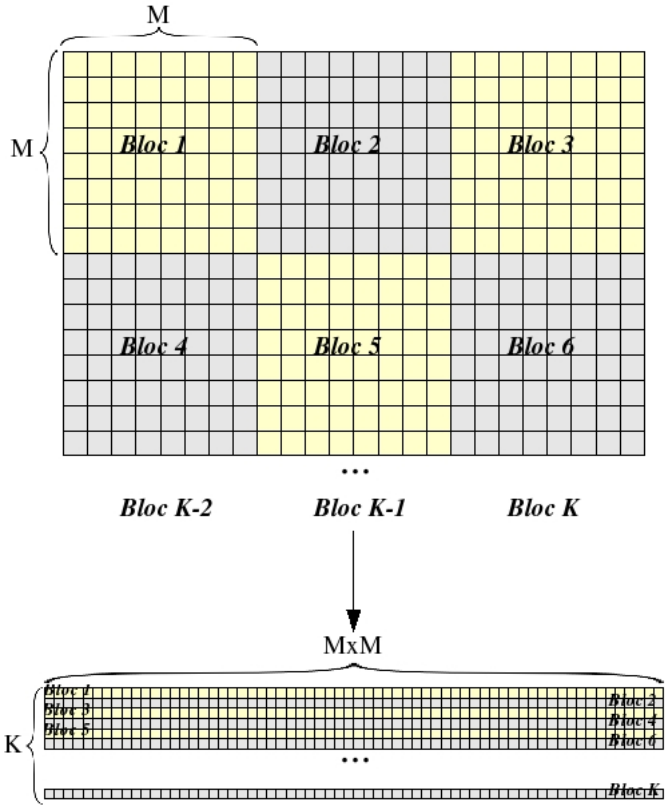


FIG. 4.6 – Transformation de l'image en matrice avec $M = 8$

Le modèle de l'ACP est

$$u = Wx$$

où u est le vecteur projeté de dimension m et x le vecteur original de dimension d.

On peut montrer que les m vecteurs projetés qui maximisent la variance de u, i.e., les axes principaux, sont les vecteurs propres $e_1, \ldots, e_m$ de la matrice de covariance C des données, correspondant aux m valeurs propres non nulles les plus grandes, $\lambda_1, \ldots, \lambda_m$.
La matrice de covariance C est calculée gâce à l'équation :

$$C = \frac{1}{n-1} \sum_{i=1}^{n} (x - \mu)(x - \mu)^T$$

Les valeurs et vecteurs propres sont ensuite obtenus en résolvant le système d'équations :

$$(C - \lambda_i I)e_i = 0, i = 1, \ldots, d$$

Les vecteurs propres sont ensuite classés par valeur propre décroissante et les m premiers sont sélectionnés comme étant les composantes principales. La matrice de projection est alors $W = E^T$, les colonnes de E étant les vecteurs propres.

FIG. 4.7 – l'algorithme ACP

On peut voir sur la figure 4.8 que les 16 premières composantes contiennent une information structurée alors que les dernières composantes sont principalement du bruit. La décomposition sur une base de KL permet donc de diminuer significativement le nombre de composantes nécessaire à la représentation des données de l'image d'entrée. La perte d'information en utilisant uniquement les 16 premières composantes parmi les 64 est minime. Pour preuve, la figure 4.9 montre la reconstruction d'une image obtenue avec 16 composantes sur 64.

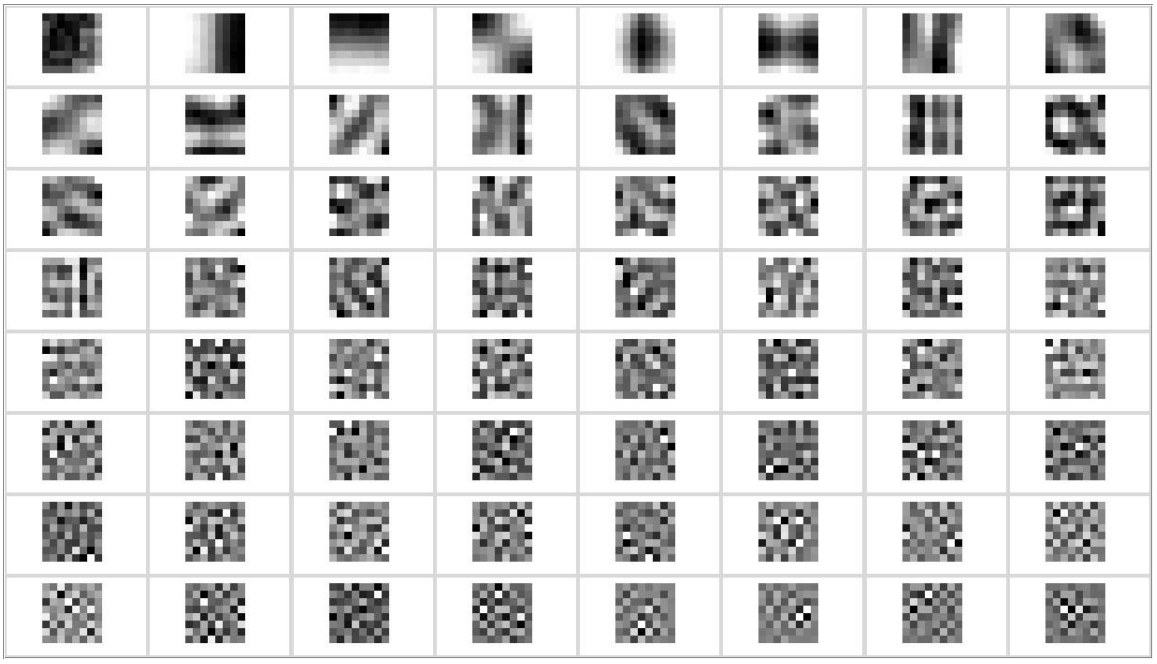
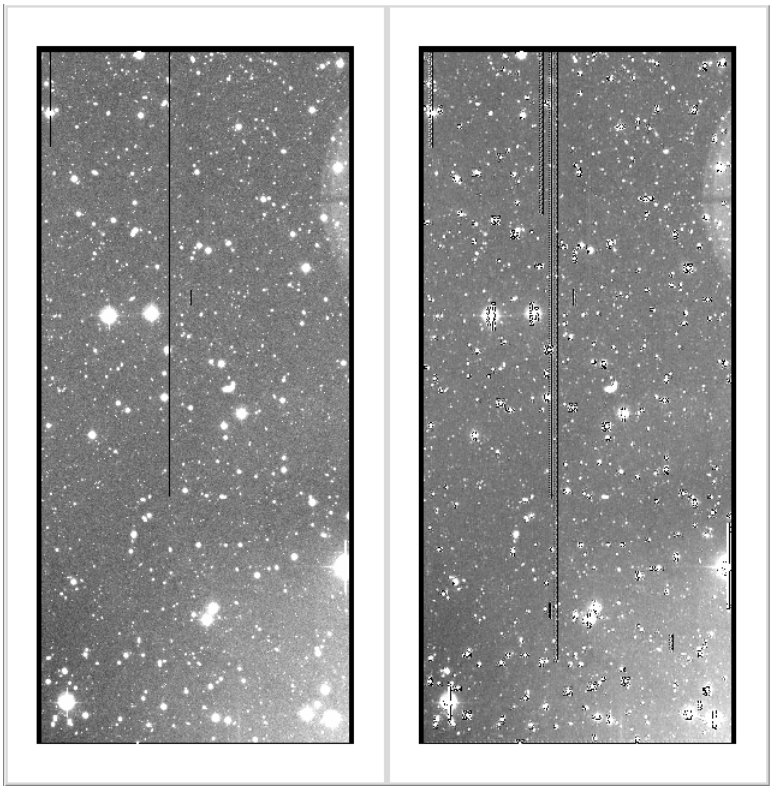


FIG. 4.8 – Résultat de l'ACP pour des blocs de 8x8 pixels



a. *b.*

FIG. 4.9 – a. l'image originale ; b. la même image reconstruite avec 16 composantes principales sur 64.

Les principes de la transformée de KL sont détaillés dans l'annexe B.2 et dans [Ber03].

Sélection des blocs

Une optimisation qui n'est pas codée au moment de l'écriture de ce rapport est la sélection intelligente des blocs à utiliser pour le calcul de la matrice de covariance. En effet, la méthode actuelle permet de créer une base de KL susceptible de représenter le mieux possible la totalité des motifs compris dans les images. Cette méthode permet de créer des échantillons non stationnaires. En sélectionnant principalement les blocs dont le pixel central fait partie d'un défaut, la base de KL obtenue sera optimisée pour la représentation des défauts. Ce qui simplifiera la tâche du système d'apprentissage.

Justification des choix

La première étape de redimensionnement se justifie par la taille des images MegaCam. Cette compression s'avère efficace et rapide. Après ce premier traitement, toute autre opération sur les pixels des images demande moins de temps processeur et moins de mémoire. Typiquement, un redimensionnement avec $N = 4$ permet de diviser par 16 la taille en mémoire et donc le nombre de pixels à traiter.

La perte de définition est également de 16 puisque pour chaque bloc de 16 pixels sur l'image d'origine correspondra une seule valeur sur le masque de sortie. Les masques résultants de l'opération apparaîtront donc comme des mosaiques de carrés de 4x4 pixels. Etant donné le travail demandé, cette première étape avec $N = 4$ était un bon compromis. Cependant, le facteur de compression N peut être plus ou moins grand voire même unitaire, conservant ainsi les dimensions originales. Cette étape reste donc configurable par l'utilisateur en fonction de ses besoins, contraintes et possibilités.

La réduction de la dynamique de l'image est une étape importante avant l'apprentissage. La fonction de transfert choisie a plusieurs avantages :

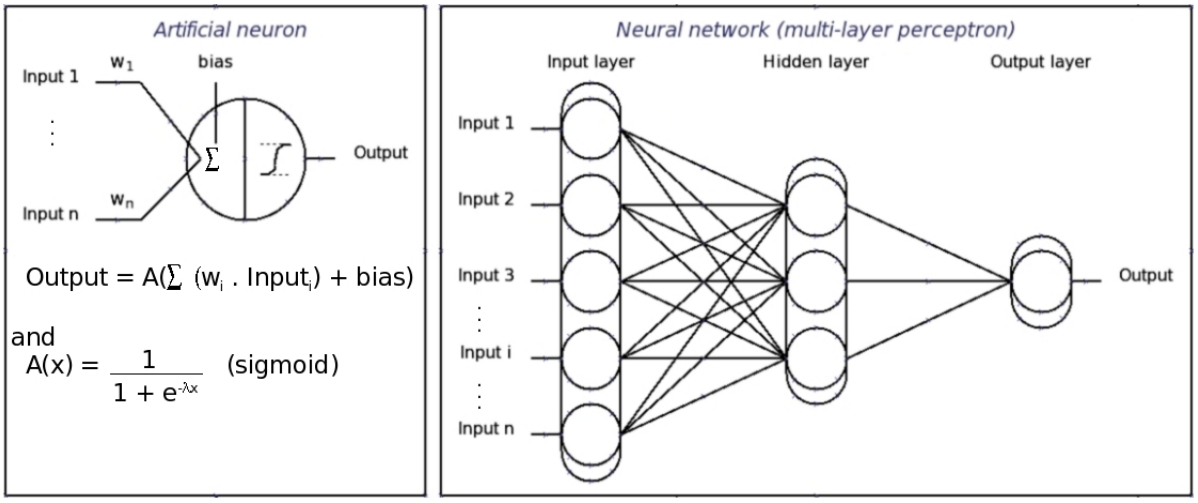
- La dynamique de l'image est réduite à environ 20dB (10x) ;
- Les propriétés du bruit des images résultantes sont indépendantes de la profondeur des images d'origine ;
- La fonction z est quasiment linéaire pour les valeurs de pixels proches de b et asymptotiquement logarithmique pour les valeurs des sources astronomiques et des défauts optiques lumineux. Les résidus de l'ACP pour les objets brillants s'expriment donc en pourcentage du flux, ce qui est la propriété souhaitée.

Enfin, le choix de l'ACP est justifié par la nature des images. En effet, les images contiennent un nombre relativement faible de motifs qui se répètent selon un processus stationnaire. Dans ce cas, l'ACP conduit necessairement à des fonctions de bases proches d'une DCT. Cette base est largement suffisant pour représenter les données avec un nombre de composantes relativement faible. Il semble que calculer une ACP soit superflu et que l'utilisation d'une DCT précalculée serait suffisante. Cependant, la spécificité des fonctions de l'ACP peut êtte améliorée en effectuant une sélection des blocs comme expliqué au paragraphe **??**. C'est dans cette optique que l'ACP devient intéressante et reste une composamte importante de l'étape de compression.

Etant donné le grand nombre de blocs créés à partir d'un MEF, l'ACP est calculée dans l'espace des signaux et non dans l'espace des attributs. En effet, les attributs étant les pixels, la taille de la matrice de covariance dans l'espace des données est de $(M \times M) \times (M \times M)$; alors que les signaux étant les blocs, la taille de la matrice de covariance dans l'espace des attributs est $K \times K$, sachant que de façon générale, $K \gg M \times M$ puisque K est de l'ordre de 10^5 alors que $M = 8$ ou 16. Voir l'annexe B.2 pour plus de détails sur la transformée de Karhunen-Loève et l'Analyse en Composantes Principales.

4.2.4 Apprentissage

Perceptrons Multi-Couches (PMC)



a. *b.*

FIG. 4.10 – a. Neurone artificiel b. Réseau de neurone de type perceptron

L'apprentissage d'un perceptron s'effectue de façon supervisée grâce à un algorithme de rétropropagation. L'algorithme choisi dans le cadre du projet est *rprop*, un des algorithmes les plus rapides actuellement parmi ceux fonctionnant en mode *batch*.

Le calcul d'un vecteur de sortie à partir d'un vecteur d'entrée s'effectue simplement après l'apprentissage.

Plus de détails sur les perceptrons et les algorithmes d'apprentissage sont donnés en annexe B.3.

Données

Les données utilisées par le système d'apprentissage se divisent en deux groupes, celles nécessaires à l'apprentissage et celles à traiter.

Pour l'apprentissage, les données d'entrée sont les images simulées et les données de sortie sont les masques correspondants. Les vecteurs d'entrée du réseau sont constitués de la transformée de KL sur la base des m axes principaux. Ainsi chaque bloc est représenté par m entrées

du perceptron comme representé sur la figure 4.11. Quant à la sortie du réseau, il s'agit d'une valeur unique correspondant au pixel central du bloc sur le masque. La sélection de cette valeur est également illustrée par la figure 4.11.

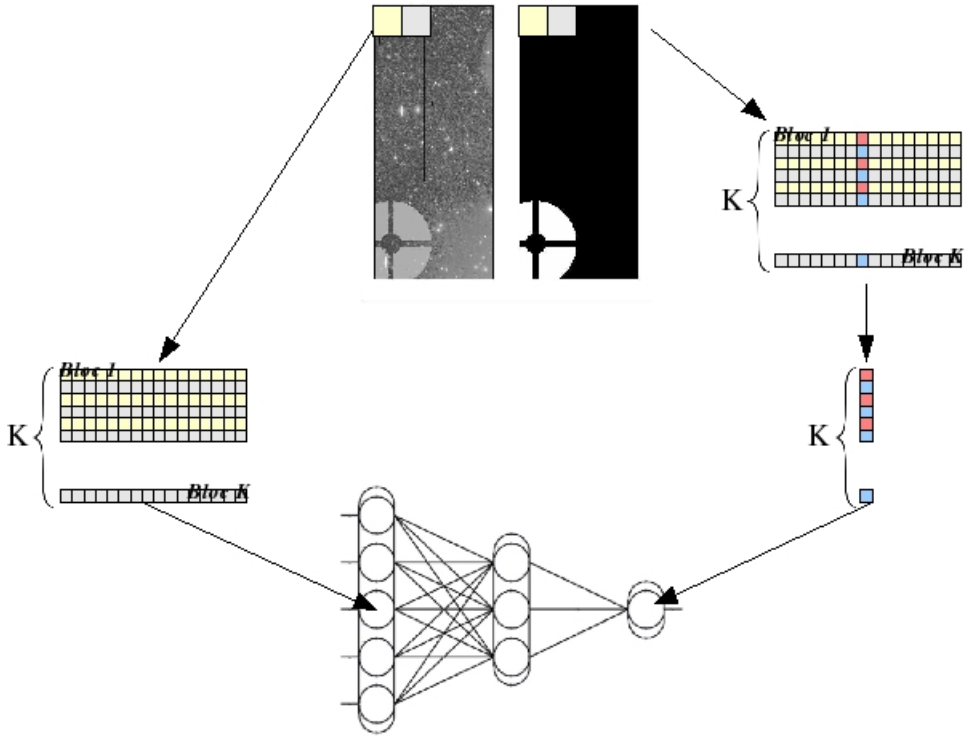


FIG. 4.11 – Données pour l'apprentissage

Pour le calcul, les données d'entrée sont les images de Megacam pour lesquelles on souhaite créer les masques. Les images sont transformées puis représentées sur la base de KL de la même façon que les images d'entrée de l'apprentissage. Le réseau calcule alors la sortie d'un pixel du masque, le pixel situé au centre du carré. Pour obtenir un masque de pixels pour la totalité de l'image, les blocs d'entrées sont obtenus par une fenêtre coulissante.

Au final, le perceptron ainsi conçu agit comme un filtre non linéaire invariant par translation.

Justification des choix

La préférence apportée aux perceptrons est motivée par leur simplicité et de leur facilité d'utilisation. Le modèle d'architecture est simple et les algorithmes de rétropropagation sont bien connus et efficaces. De plus, grâce à l'algorithme *rprop*, l'apprentissage est rapide. Etant donnée la nature de la classification à effectuer, un PMC était suffisant.

4.2.5 Système de vision

Le système de vision a été développé de façon hiérarchique. Le premier modèle donnait des résultats médiocres car la perte d'informations entre les différents niveaux hiérarchiques était

trop importante. Un deuxième modèle donnait des résultats acceptables mais pas satisfaisants car le traitement était redondant et donc plus lent sans que les masques soient significativement meilleurs. Enfin, un dernier modèle plus simple, plus logique et probablement plus efficace que le second a été pensé mais n'est pas encore achevé. Ce système basé sur des redimensionnements successifs devrait être développé durant les trois semaines de stage restantes.

Premier modèle

Le premier modèle était composé de deux couches successives. La première couche correspond au réseau construit à la figure 4.11. La deuxième couche donne une vue locale plus large. En effet, pour chaque bloc étudié, le réseau de la deuxième couche va s'intéresser aux blocs voisins. Pour cela, la valeur centrale des c blocs adjacents du masque calculée par le premier réseau est utilisée comme entrée au second. La figure 4.12 montre la création des vecteurs d'entrée pour les réseaux de chaque couche ainsi que la valeur de sortie calculée par chacun des deux. La sortie est identique pour les deux réseaux puisque le second est censé confirmer ou infirmer la classification du premier. Ce contrôle est effectué par une simple multiplication des deux valeurs obtenues pour un pixel.

Si on fait le parallélisme avec le modèle de Thorpe (cf. 4.2), la première couche est la classification et la deuxième couche l'analyse de la classification.

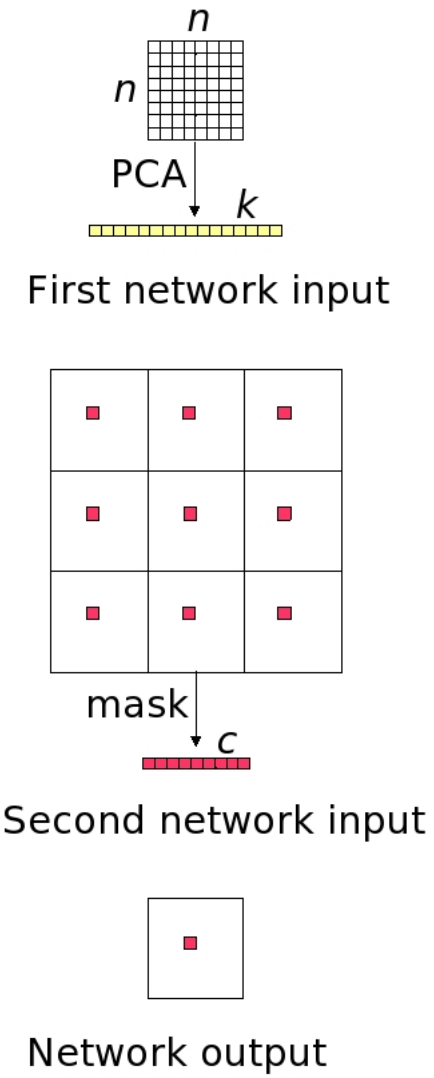


FIG. 4.12 – Vecteurs utilisés par le premier modèle

Ce modèle donnait des résultats médiocres car l'utilisation du pixel central de chaque bloc provoquait des erreurs comme cela est expliqué par la figure 4.13. Pour l'apprentissage des trainées de satellite, on utilise des blocs comme ceux indiqués par le premier schéma. La reproduction de ce schéma déclenche donc le réseau qui reconnaît le défaut. Cependant, dans le cas du second schéma, les pixels centraux de l'image sont tous aussi lumineux qu'une trainée de satellite, le réseau va donc se déclencher et provoque une fausse détection.

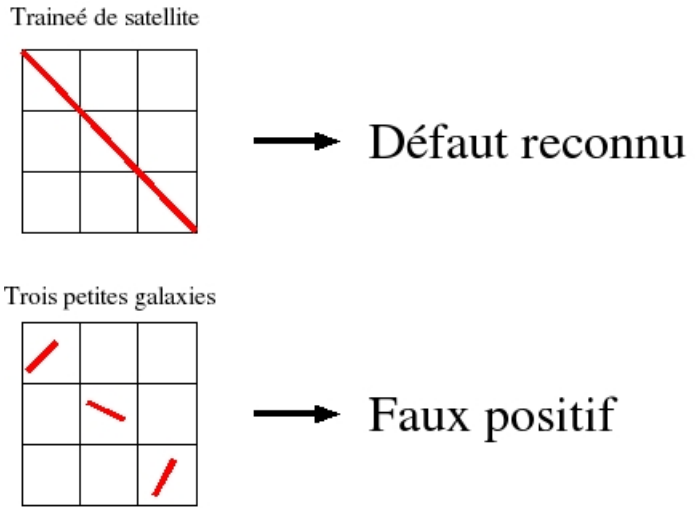


FIG. 4.13 – Exemple de problème obtenu avec le premier modèle

Ce modèle simpliste est donc assez peu efficace car il est peu efficace dans les cas difficiles. Or, le but du modèle est de conforter ou d'infirmer les choix du premier réseau.

Second modèle

Le second modèle est celui qui a été présenté à la conférence ADA III et qui est expliqué sur le poster A.1. La première couche reste la même que pour le premier modèle, la différence est apportée à la conception du second réseau afin de limiter la perte d'information entre les deux niveaux (voir figure 4.13).

Le second modèle tente donc de réduire cette perte en modifiant le vecteur d'entrée du second réseau. Ce vecteur une concaténation entre le vecteur d'entrée de la première couche et le vecteur d'entrée de la seconde couche du premier modèle. En effet, la figure 4.14 montre qu'il se compose :

- des valeurs de la PCA du bloc central ;
- des valeurs des pixels centraux des blocs adjacents.

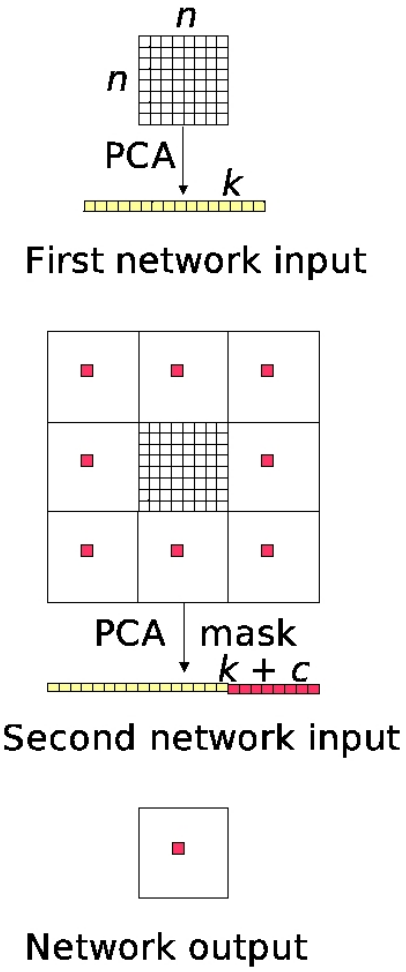


FIG. 4.14 – Vecteurs utilisés par le second modèle

Pour cette architecture aussi, la première couche est la classification et la deuxième couche l'analyse de la classification selon le modèle de Thorpe (cf. 4.2),.

Ainsi, la perte d'information est significativement réduite. Pour reprendre l'exemple 4.13, le fait que les pixels centraux des blocs adjacents soient activés ne constituent qu'une confirmation. Le réseau se déclenchera sur le premier schéma car le bloc central est une diagonale et que les blocs adjacents confirment la direction de cette diagonale. Par contre, sur le deuxième schéma, le bloc central contient une galaxie et ne déclenche donc pas une détection. Les informations fournies par les blocs adjacents permettent de repérer de fausses détections.

Ce second modèle est visiblement plus précis et plus efficace que le premier car la perte d'information est moindre. Ainsi, dans l'exemple de la figure 4.13, la galaxie ne sera pas considérée comme un défaut car elle ne correspond pas à une partie de trainée. Cependant, l'architecture est redondante puisque les composantes principales du bloc central sont utilisées séparément dans chacun des réseaux. Les performances de ce modèle sont donc moyennes.

Troisième modèle

Le troisième et dernier modèle est en cours de développement au moment de la rédaction du rapport. Sa hiérarchie est totalement différente des deux premiers. La modélisation de la hiérarchie est cette fois obtenue par un redimensionnement de l'image. Cette modélisation a plusieurs avantages :

- une seule représentation des données ;
- un seul type de réseau ;
- pas de restructuration des données ;
- un niveau de hiérarchie pouvant facilement être augmenté.

Pour chaque niveau de la hiérarchie la même méthode est appliquée, un réseau de neurones comme celui utilisé sur la première couche du premier modèle (figures 4.11).

Ce modèle est une approche différente par rapport au modèle de Thorpe. La classification se fait d'abord en étudiant l'image le plus globalement possible, puis la classification est analysée plus finement pour établir une nouvelle classification. En ce sens, ce modèle est plus proche de celui de Thorpe. Il présente tout de même une différence qui est de pouvoir contenir plusieurs étapes d'analyse de plus en plus fines, donnant des classifications successives plus précises.

Une autre modification majeure apportée par ce modèle est l'ordre dans lequel les couches sont utilisées. Au lieu de partir du plus précis vers le plus large, ce système fonctionne à l'inverse. L'intérêt de cette inversion est de limiter le nombre de blocs étudiés au fur et à mesure. Si aucun défaut n'est détecté au sein d'un bloc à grande échelle, il est inutile de rechercher plus précisement à l'intérieur de ce bloc. L'avantage est évident pour un CCD défectueux par exemple.

Cette méthode présente l'inconvénient d'imposer le calcul de l'ACP pour chaque niveau de redimensionnement. Cependant, il est fort probable que le gain sur le nombre de blocs à traiter sur chaque niveau compense largement ce surplus de calcul. De plus, l'ACP demande du temps lors de l'entraînement uniquement, pour le calcul des masques, c'est la base de KL précalcuée qui est utilisée.

Ce troisième modèle apparait finalement plus simple mais probablement plus efficace, aussi bien au niveau des manipulations de pixels et des temps de calcul que des performances de classification.

4.2.6 Schéma global

Le schéma 4.15 suivant représente le fonctionnement global du projet, le fonctionnement de chaque module pouvant être modifié séparement.

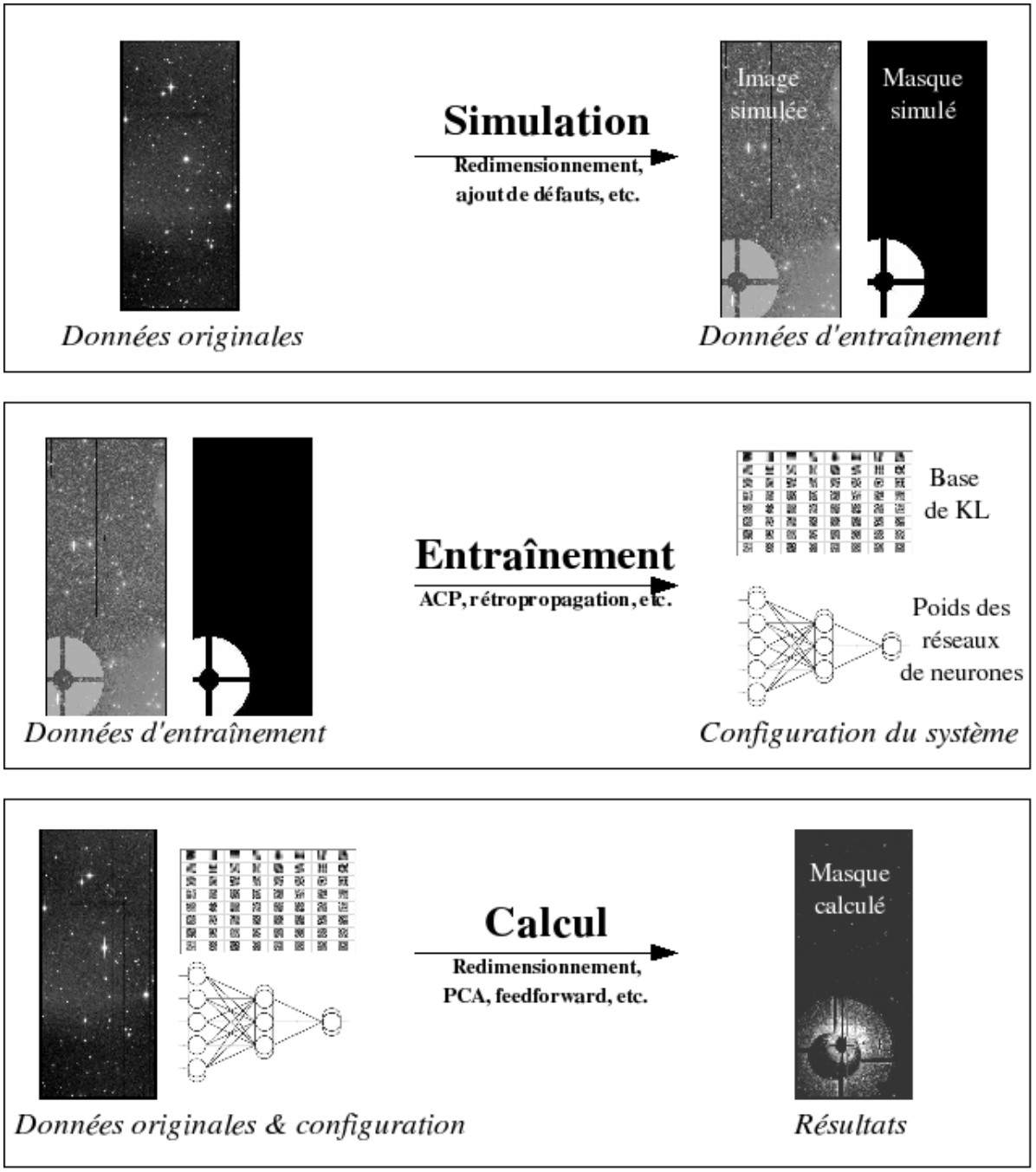


FIG. 4.15 – Fonctionnement global du projet

5 Bilan

5.1 Résultats

Les premiers masques crées par le logiciel sont plutôt satisfaisants. Ci-dessous (figure 5.1) se trouve l'exemple d'un masque généré à partir de données réelles et du deuxième modèle de vision. Les données du test sont les suivantes :

- Entraînement
  - Taille des blocs ($N \times N$) : 8x8 pixels
  - Nombre de blocs (K) : 352 152
  - Nombre de composantes principales utilisées (m) : 16
  - Temps de calcul de la base de KL : environ 1 minute
  - Temps d'apprentissage des réseaux : environ 1 heure
- Calcul
  - Taille des blocs ($N \times N$) : 8x8 pixels
  - Nombre de blocs (K) : 22 068 288
  - Nombre de composantes principales utilisées (m) : 16
  - Temps de calcul de la base de KL : Précalculé
  - Temps de calcul des masques à partir des réseaux : environ 4 minutes

Ces résultats sont données à titre indicatif et ne sont pas représentatifs des performances finales du projet, aussi bien en temps de calcul qu'en qualité de masquage.

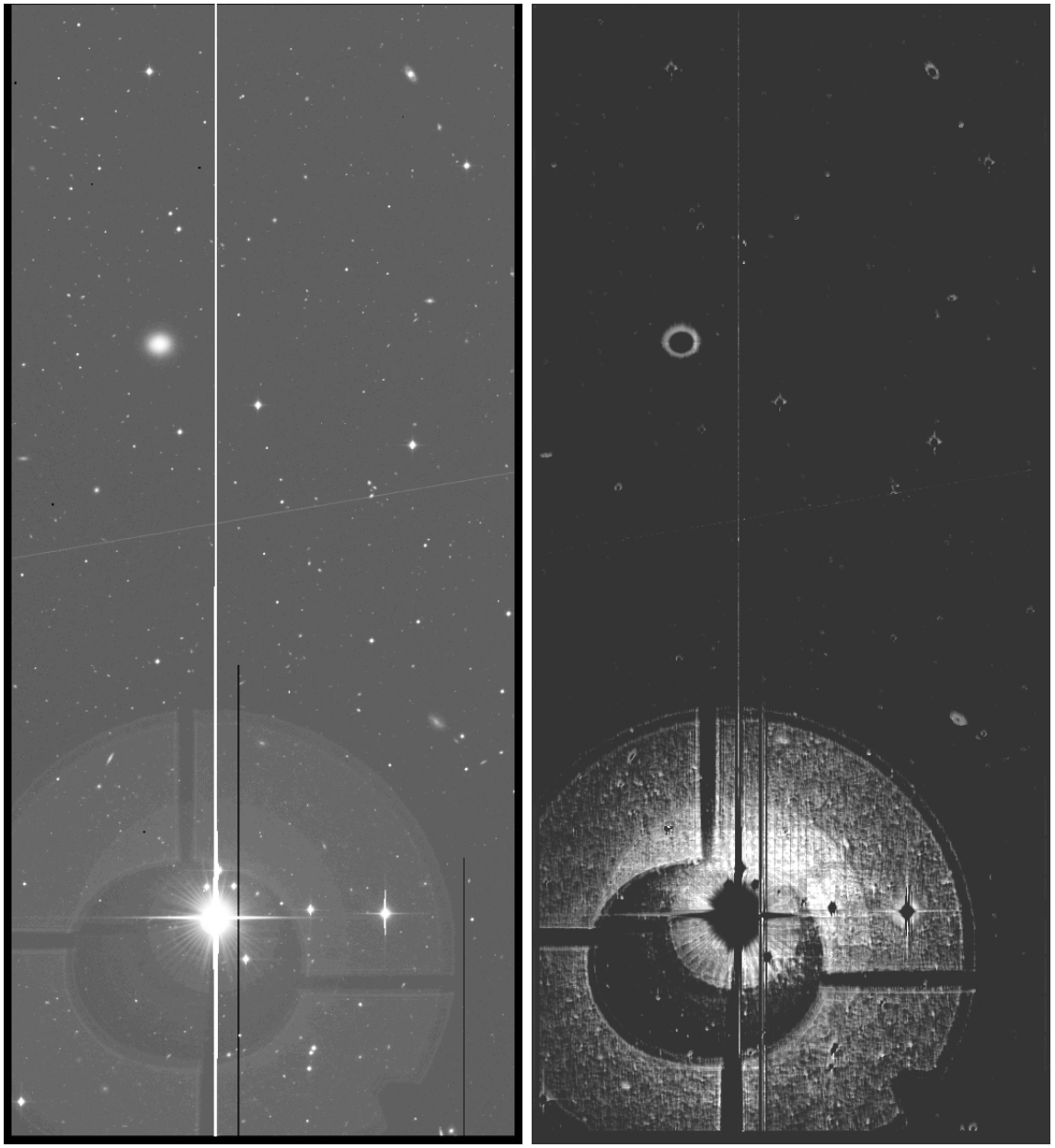


FIG. 5.1 – Exemple d'un masque calculé pour un CCD

Concernant les performances propres du réseau, quelques calculs ont été faits. L'erreur quadratique de la classification (c'est-à-dire le carré de la différence entre la valeur du masque calculé et la valeur du masque attendu) à partir de masques simulés a une valeur moyenne de 0.027. Même si ce n'est qu'un résultat préliminaire, un score si faible est encourageant pour la suite des tests.

5.2 Intérêt du stage pour l'IAP

Le projet a un intérêt évident pour l'IAP puisque le fonctionnement du logiciel permettrait d'une part de gagner un temps considérable pour le traitement des images et d'autre part de permettre aux astronomes de travailler sur des tâches nettement plus intéressantes.

A court terme, le projet devrait s'inscrire dans le pipeline TERAPIX afin d'obtenir des meilleurs résultats pour les mesures astrométriques.

A long terme, le projet pourrait devenir plus générique et permettre de détecter automatiquement des défauts divers sur des images de toute sorte. En théorie, cela doit être possible à partir du moment où le simulateur est correct et que l'apprentissage est suffisamment long. Cependant, une telle évolution pourrait demander d'autres phases de prétraitement différentes ou complémentaires.

5.3 Intérêt personnel

Durant le stage, j'ai découvert le fonctionnement d'un laboratoire de recherche fondamentale alors que mon cursus scolaire correspond plus à un profil d'ingénieur en industrie. Cette découverte est une expérience enrichissante car elle m'a apporté un autre point de vue sur la façon d'organiser son travail personnel mais aussi le travail dans une équipe où les compétences sont variées.

En effet, l'équipe TERAPIX comptant trois astronomes et des ingénieurs dans des domaines liés à l'astrophysique, j'ai appris beaucoup dans ce domaine. A l'inverse, mes connaissances en développement m'ont été utiles pour aider les membres de l'équipe, ce qui m'a donné l'occasion de donner des explications techniques à des personnes dont le niveau et les compétences étaient différents des miens.

Un intérêt majeur du stage a été la conférence en Italie durant laquelle j'ai pu rencontrer des scientifiques du monde entier, réunis pour discuter de problèmes concrets de traitement de données. De plus, la présentation du projet durant la conférence constitue une publication à mon nom.

Pour l'aspect plus technique, j'ai acquis de nouvelles connaissances grâce aux développement de méthodes mathématiques que je n'avais pas encore étudié (comme l'ACP). J'ai également consolidé mon savoir sur les réseaux de neurones en utilisant une nouvelle fois les perceptrons et en découvrant, bien que partiellement, les machines à support de vecteurs (SVM).

5.4 Conclusion et retour d'expérience

De façon générale, le déroulement du stage a été satisfaisant. Les techniques choisies se sont révélées adaptées et les optimisations à venir devraient apporter un surplus d'efficacité. Le planning a été respecté même si le projet peut subir des améliorations.

L'IAP étant un institut de recherches, la plupart des personnes y travaillant viennent de la recherche scientifique, en physique, astronomie, astrophysique, mais très peu de gens ont une réelle formation en informatique et développement. Mon cursus était donc particulièrement adapté à la réalisation du projet puisque j'apportais des compétences complémentaires.

6 Liste des figures

7 Liste des tableaux

8 Glossaire

Astrométrie : Partie de l'astronomie qui détermine la position des astres sur la sphère céleste.

Astrophysique : Branche de l'astronomie dont l'objet est de comprendre la naissance, l'évolution et la mort des astres et des systèmes célestes selon les termes des lois physiques qui les gouvernent.

Capteur CCD : Charge-Coupled Device, ou capteur à transfert de charge, composant basé sur une technologie de composants à semi-conducteurs. Voir http ://telesun.insa-lyon.fr/ telesun/Acquisition/L03/titre.html.

Diffraction : En physique, phénomène ondulatoire dans lequel une onde se disperse en passant au bord d'un solide ou en traversant une fente étroite, au lieu de continuer tout droit. Les ondes diffractées tendent à s'annihiler ou à se renforcer mutuellement.

Dimensionnalité : Propriété spatiale d'avoir des dimensions.

Photométrie : Partie de la physique qui concerne la mesure des intensités lumineuses.

Relevé (de télescope) : programme d'observation du ciel.

9 Bibliographie

[Ber01] Emmanuel Bertin. *EyE User's guide*, 2001.

[Ber03] Emmanuel Bertin. *Traitement du signal, notes de cours.* DESS Outils et Systèmes de l'Astronomie et de l'Espace, Module Analyse Numérique, 2003.

[dC99] Nicolas de Coussemaker. *Panorapix, the TERAPIX image visualization tool.* Internship report, I.U.P.G.M.I., 1999.

[Lé96] Pierre Léna. *Méthodes physiques de l'observation.* CNRS Editions, 1996.

[VC03] Thierry Viéville and Sylvie Crahay. A deterministic biologically plausible classifier. Technical report, INRIA, 2003.

10 Webographie

Le site de l'IAP, http ://www.iap.fr

Le site de TERAPIX, http ://terapix.iap.fr

Le site du logiciel Panorapix, http ://terapix.iap.fr/cplt/oldSite/soft/panorapix/

Le site du logiciel DS9, http ://hea-www.harvard.edu/RD/ds9/

Le site de FITSIO, http ://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html

Annexes

A Travail produit

A.1 Poster réalisé pour la conférence ADA III

TERAPIX

http://terapix.iap.fr IAP – CNRS

Automatic identification of optical defects on astronomical images

A. BAILLARD for the Terapix team (IAP)

Imaging surveys originating from wide-field instruments suffer from a number of optical defects as diffraction spikes, halos, etc. We describe a system based on machine learning which can identify automatically these artifacts in pixel data. The software is still under development and testing. When it is completed, this new Terapix tool will be made available to the community under the GPL license as *Defectix*.

Introduction

Defectix must identify large artifacts directly from pixel data because it must create masks as pixel data as well. The main difficulty lies in the large scale range of defects.

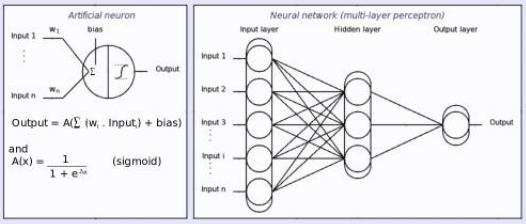
Compared to "real-life" images, lightning and perspective effects are not relevant for the analysis of astronomical data. For this reason machine learning is a simple and efficient approach.

A supervised system based on neural networks should be able to treat arbitrary defects without the need to code any new detection algorithms. Our software deals with artifacts such as halos, diffraction spikes, satellite trails and scattered light.

Once the network has been trained, *Defectix* acts as a non linear translation invariant filter.

Neural networks

Neural networks used in *Defectix* are three layers Multi-Layer Perceptrons (MLP).



They are trained using *rprop* algorithm (M. Riedmiller and H. Braun), a fast batch backpropagation method.

Simulations

Original image

Prereduction

Original images are first rebinned to better match the seeing.
A background model is then substracted.

Defect addition

Defects are added with random positions, intensities and sizes. Corresponding masks are created simultaneously.

Postprocessing

Images are dynamically compressed using the following function :

$$y(x) = \frac{x}{|x|} \ln\left(1 + \frac{|x|}{\sigma}\right)$$

where σ is the standard deviation of the global background.

Image with defects

Corresponding mask

Training

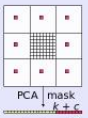
Segmentation

Principal Component Analysis (PCA) is carried out on blocks of $n \times n$ pixels. Only the k first components are used as inputs to the neural networks.
Typically, $n = 8$ and $k = 16$ (among 64).

First network input

Training networks

The first network is trained with :
- input : principal components of each block.
- output : value of the mask for the central pixel of the block.



Second network input

The second network is trained with :
- input : principal components of each block and value of the mask for the central pixel of the c closest blocks (Typically $c = 8$ or $c = 24$).
- output : value of the mask for the central pixel of the central block.

Network output

Mask creation

Sliding window

Images are analysed as blocks as for training. Blocks slide along the image to compute a mask value for each pixel of the original image.

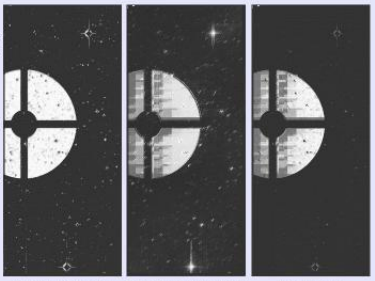
Mask computation

The first network output value (o_1) is considered to be the mask value.

The output from the second network, computed neighbouring blocks, is used to validate the mask values calculated by the first network.

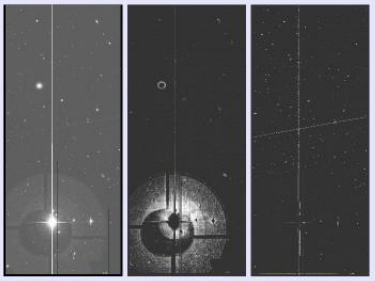
The final mask value is given by:

$$m = o_1 . (o_2)^{1/2}$$



First network result | Second network result | Final result

Real data



Original image | Mask for halos | Mask for trails

Conclusions

Time evaluation

* Training and Computing time do not include the PCA computation time because a precomputed KL base is loaded.
Tests were done on a 64 bit Opteron clocked at 1.8GHz.

	Simulating	Training	Computing	PCA
36 CCD (2112x4644) MEF file Halos	30s	-	7 min 10s	1 min 15s
36 CCD (528x1161) MEF file Halos	-	55 min	6 min 45s	1 min 15s
36 CCD (2112x4644) MEF file Satellite trails	20s	-	7 min 10s	1 min 15s
36 CCD (528x1161) MEF file Satellite Trails	-	5 min	6 min 45s	1 min 15s

Defectix is still being developed and improved.

Todo list

- Optimise data sets to contain mainly blocks with defects ;
- Determine the optimal configuration for the second network ;
- Provide realistic simulations of a wilder range of defects ;
- Optimise code ;

Defectix version 1.0 should be released in July 2004.

B Algorithmes et méthodes

B.1 Calcul de la valeur du fond de ciel

La valeur du fond de ciel est déterminée grâce à un algorithme récursif travaillant sur l'histogramme de l'image. Plusieurs variables sont utilisées :

- H histogramme de l'image
- P nombre de pixels de l'histogramme H de 1 à P
- p taille d'un bloc de pixels de l'histogramme H
- n nombre de blocs de taille p dans l'histogramme de taille P, de 1 à n
- $H(i)$ valeur de l'histogramme pour l'élément i
- $H(k,i)$ valeur de l'histogramme pour l'élément i du bloc k, $H(k,i) = H(k \times p + i)$
- σ_0 deviation standard de la récursion précédente
- σ_k deviation standard du bloc k
- m_k moyenne du bloc k
- c_k valeur centrale du bloc k
- b valeur du fond de ciel recherchée.

Eventuellement, avant d'utiliser l'algorithme ci-dessous, il est possible de n'utiliser qu'une partie de l'histogramme en récupérant aléatoirement une fraction des P pixels.

1. Init σ_0 to 0 and $p < P$
2. Sort H by ascending values
3. Compute

$val \leftarrow 0$
for $k \leftarrow 1$ to n
    if $H(k,p) - H(k,0) > val$
        $val \leftarrow H(k,p) - H(k,0)$
        $pos \leftarrow k$
    ifend
forend
if $\sigma_{pos} - \sigma_0 > 0.2\sigma_0$
    $\sigma_0 \leftarrow \sigma_{pos}$
    $p \leftarrow p/2$
    back to step 3
ifend

if $H(pos, 0) > m_{pos} - 3\sigma_{pos} H(pos, p) < m_{pos} + 3\sigma_{pos}$
$b \leftarrow m_{pos}$
else
$b \leftarrow 2.5c_{pos} - 1.5m_{pos}$
ifend

B.2 ACP, espace des attributs et espace des données

B.3 Perceptrons et rétropropagation

B.3.1 Comparaison d'algorithmes de rétropropagation

mode batch et mode stochastique

Il existe deux types d'algorithme de rétropropagation :
- les algorithmes en mode stochastiques qui ajustent les poids du réseau après chaque nouvel exemple de jeu de données ;
- les algorithmes en mode batch qui ajustent les poids après avoir testé l'ensemble du jeu de données.

Il existe un algorithme standard pour ces deux modes, une simple rétropropagation de gradient. Les algorithmes plus performants sont généralement des optimisations d'un des deux. C'est le cas de *rprop* qui est une amélioration de la rétropropagation en mode batch.

Les trois algorithmes qui vont être comparés sont
- la rétropropagation en mode stochastique ;
- la rétropropagation en mode batch ;
- rprop.

Motif d'apprentissage

Le motif utilisé pour comparer les différents algorithmes est le suivant :

```
...**.....
..****....
.******...
..****....
...**.....
```

Vitesses de convergence

Algorithme	Temps CPU
Backpropagation stochastique	0 :07.08
Backpropagation batch	0 :05.16
rprop	0 :00.29

TAB. B.1 – Vitesses de convergence em temps CPU

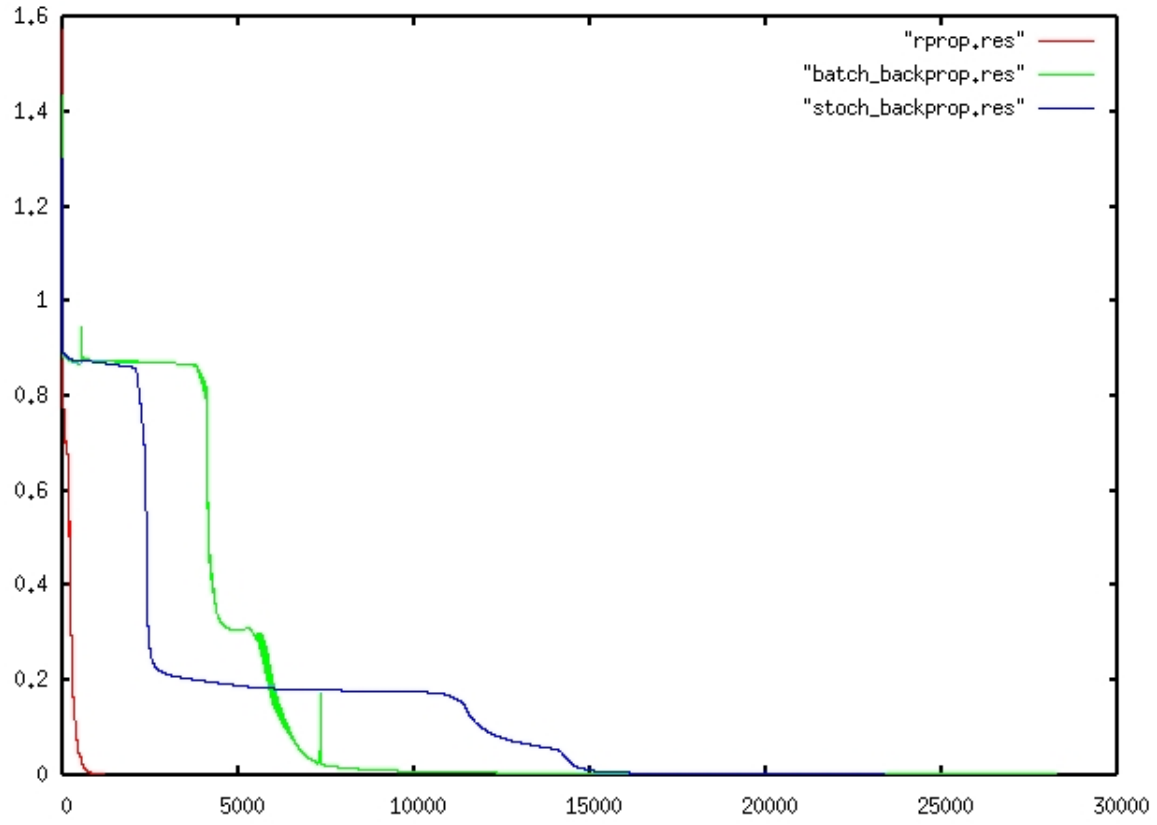


FIG. B.1 – Vitesses de convergence en nombre d'itérations

B.3.2 Algorithmes de rétropropagation de gradient

- n nombre de couches de 0 à $n-1$
- $nb(c)$ nombre de neurones de la couche c
- $N_{c,i}$ neurone i de la couche c
- $w_{c,i,j}$ poids de la connexion entre $N_{c-1,j}$ et $N_{c,i}$
- $A_{c,i}$ valeur d'activation de $N_{c,i}$
- $S_{c,i}$ valeur de sortie de $N_{c,i}$
- d_i valeur de sortie désirée pour $N_{n-1,i}$
- y_i valeur de sortie obtenue pour $N_{n-1,i}$ ($y_i = S_{n-1,i}$)
- x_i valeur d'entrée i
- $\varphi_{c,i}$ fonction d'activation pour $N_{c,i}$
- $T = (x_i, d_i)$ base d'entrainement

rétropropagation en mode stochastique

C'est l'algorithme standard où les poids sont modifiés pour chaque exemple.

1. Init weights
2. Give an example
3. Compute

   for $i \leftarrow 1$ to $nb(n-1)$

$\left(\frac{\delta E}{\delta A}\right)_{n-1,i} \leftarrow 2\varphi'_{n-1,i}(A_{n-1,i})(S_{n-1,i} - D_i)$

forend

for $c \leftarrow n-2$ downto 1

for $i \leftarrow 1$ to $nb(c)$

$\left(\frac{\delta E}{\delta A}\right)_{c,i} \leftarrow 0$

for $k \leftarrow 1$ to $nb(c+1)$

$\left(\frac{\delta E}{\delta A}\right)_{c,i} \leftarrow \left(\frac{\delta E}{\delta A}\right)_{c,i} + w_{c+1,k,i}\left(\frac{\delta E}{\delta A}\right)_{c+1,k}$

forend

forend

forend

$\left(\frac{\delta E}{\delta A}\right)_{c,i} \leftarrow \left(\frac{\delta E}{\delta A}\right)_{c,i} \varphi'_{c,i}(A_{c,i})$

for $c \leftarrow 1$ to $n-1$

for $i \leftarrow 1$ to $nb(c)$

for $j \leftarrow 1$ to $nb(c-1)$

$w_{c,i,j} \leftarrow w_{c,i,j} - \varepsilon\left(\frac{\delta E}{\delta A}\right)_{c,i} S_{c-1,j}$

forend

forend

forend

4. Back to step 2 until training is validated

rétropropagation en mode batch

Cet algorithme est différent de celui en mode stochastique car les poids ne sont modifiés qu'une fois tous les exemples de la base d'entrainement passés dans le réseau.

1. Init weights
2. Init $\Delta w_{c,i,j}$

for $c \leftarrow 1$ to $n-1$

for $i \leftarrow 1$ to $nb(c)$

for $j \leftarrow 1$ to $nb(c-1)$

$\Delta w_{c,i,j} \leftarrow 0$

forend

forend

forend

3. Compute

for x_d in T

for $i \leftarrow 1$ to $nb(n-1)$

$\left(\frac{\delta E}{\delta A}\right)_{n-1,i} \leftarrow 2\varphi'_{n-1,i}(A_{n-1,i})(S_{n-1,i} - D_i)$

forend

for $c \leftarrow n-2$ downto 1

for $i \leftarrow 1$ to $nb(c)$

$\left(\frac{\delta E}{\delta A}\right)_{c,i} \leftarrow 0$

for $k \leftarrow 1$ to $nb(c+1)$

$\left(\frac{\delta E}{\delta A}\right)_{c,i} \leftarrow \left(\frac{\delta E}{\delta A}\right)_{c,i} + w_{c+1,k,i}\left(\frac{\delta E}{\delta A}\right)_{c+1,k}$

forend

forend

forend

$\left(\frac{\delta E}{\delta A}\right)_{c,i} \leftarrow \left(\frac{\delta E}{\delta A}\right)_{c,i} \varphi'_{c,i}(A_{c,i})$

for $c \leftarrow 1$ to $n-1$

for $i \leftarrow 1$ to $nb(c)$

for $j \leftarrow 1$ to $nb(c-1)$

$\Delta w_{c,i,j} \leftarrow \Delta w_{c,i,j} + \left(\frac{\delta E}{\delta A}\right)_{c,i} S_{c-1,j}$

forend

forend

forend

forend

for $c \leftarrow 1$ to $n-1$

for $i \leftarrow 1$ to $nb(c)$

for $j \leftarrow 1$ to $nb(c-1)$

$w_{c,i,j} \leftarrow w_{c,i,j} - \varepsilon \Delta w_{c,i,j}$

forend

forend

forend

4. Back to step 2 until training is validated

Rprop

C'est une méthode batch efficace. Elle utilise le signe de la dérivée partielle pour calculer la mise à jour des poids. La mise à jour est définie par $\Delta_{ij}^{(t)}$:

$$\Delta w_{ij}^{(t)} = -\Delta_{ij}^{(t)}, if \frac{\delta E}{\delta w_{ij}}(t) > 0$$

$$\Delta w_{ij}^{(t)} = +\Delta_{ij}^{(t)}, if \frac{\delta E}{\delta w_{ij}}(t) < 0$$

$$\Delta w_{ij}^{(t)} = 0, else$$

Rprop ne demande pas de réglage particulier des paramètres, même si plusieurs constantes sont utilisées dans l'algorithme. Les meilleurs résultats en moyenne sont obtenus pour :

$\Delta_0 = 0.1$
$\Delta_{max} = 50.0$
$\Delta_{min} = 1e^{-6}$
$\eta^+ = 1.2$
$\eta^- = 0.5$

Δ_{max} et Δ_{min} représentent les bornes de Δ. Δ_0 est la valeur d'initialisation de Δ. η^+ et $\eta-$ sont les facteurs de croissance et de décroissance de Δ.

1. $\forall i, j : \Delta_{ij}(t) = \Delta_0$

   $\forall i, j : \frac{\delta E}{\delta w_{ij}}(t-1) = 0$

2. Compute Gradient $\frac{\delta E}{\delta w}$

   For all weights and biases
   if $(\frac{\delta E}{\delta w_{ij}}(t-1) * \frac{\delta E}{\delta w_{ij}}(t-1) > 0)$ then
   $\Delta_{ij}(t) = \min(\Delta_{ij}(t-1) * \eta^+, \Delta_{max})$
   $\Delta w_{ij}(t) = -sign(\frac{\delta E}{\delta w_{ij}}(t)) * \Delta_{ij}(t)$
   $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
   $\frac{\delta E}{\delta w_{ij}}(t-1) = \frac{\delta E}{\delta w_{ij}}(t)$

   else if $(\frac{\delta E}{\delta w_{ij}}(t-1) * \frac{\delta E}{\delta w_{ij}}(t-1) < 0)$ then
   $\Delta_{ij}(t) = \max(\Delta_{ij}(t-1) * \eta^-, \Delta_{min})$
   $\frac{\delta E}{\delta w_{ij}}(t-1) = 0$

   else if $(\frac{\delta E}{\delta w_{ij}}(t-1) * \frac{\delta E}{\delta w_{ij}}(t-1) = 0)$ then
   $\Delta w_{ij}(t) = -sign(\frac{\delta E}{\delta w_{ij}}(t)) * \Delta_{ij}(t)$
   $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
   $\frac{\delta E}{\delta w_{ij}}(t-1) = \frac{\delta E}{\delta w_{ij}}(t)$

3. Return in step 2 until the training is validated