



De l'exécution distribuée des chaînes de traitements d'images à l'acquisition distribuée de connaissance par les traitements

Flavien Vernier

► To cite this version:

Flavien Vernier. De l'exécution distribuée des chaînes de traitements d'images à l'acquisition distribuée de connaissance par les traitements. Informatique. Université Savoie Mont-Blanc, 2021. tel-03417439

HAL Id: tel-03417439

<https://hal.science/tel-03417439>

Submitted on 5 Nov 2021

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.

HABILITATION A DIRIGER DES RECHERCHES

Spécialité : Sciences et technologie de l'information et de la communication

De l'exécution distribuée des chaînes de traitements
d'images à l'acquisition distribuée de connaissance par les
traitements

Flavien Vernier

Soutenue à Annecy-le-Vieux le 25/6/2021

Jury

Christian Perez	Président	Directeur de Recherche	LIP, INRIA
Lydie du-Bousquet	Rapportrice	Professeur	LIG, UGA
Eddy Caron	Rapporteur	McF HDR	LIP, ENS Lyon
Pierre Sens	Rapporteur	Professeur	LIP6, Université Sorbonne
Sébastien Monnet	Examineur	Professeur	LISTIC, USMB
Emmanuel Trouvé	Examineur	Professeur	LISTIC, USMB

Remerciements

La rédaction de mon HDR fut une transition, un pont entre mes activités sur le traitement d'images et celles sur les objets sages. La réalisation de l'ensemble de ces travaux n'a pu se réaliser sans la collaboration, le soutien ou la simple présence de personnes que je tiens particulièrement à remercier.

Dans un premier lieu je tiens à sincèrement remercier les membres de mon jury, Christian Perez et tout particulièrement mes rapporteurs, Lydie du-Bousquet, Eddy Caron et Pierre Sens, d'avoir accepté la tâche délicate d'examiner, rapporter un travail transdisciplinaire vous menant par certains aspects hors de vos domaines d'expertises.

Je souhaite également remercier tout spécialement Emmanuel Trouvé, Patrick Lambert et Sébastien Monnet. Emmanuel pour m'avoir fait découvrir le traitement de séries d'images et m'avoir embarqué dans le projet EFIDIR qui fut la genèse de mes travaux sur le traitement d'images. Patrick pour avoir cru et soutenu en tant que directeur du LISTIC une idée peut-être saugrenue, les objets sages et au fait que ces derniers pouvaient rêver. Sébastien, pour ton accompagnement tout au long de cette expérience particulière qu'est l'HDR.

Mes travaux sur les objets sages n'auraient jamais pu se réaliser sans le soutien d'Ilham, qui dès les premières réflexions a été présente, pour nos discussions et nos échanges, je te remercie chaleureusement.

Au-delà de mes collègues qui ont eu un rôle important dans mes travaux, le soutien de ma famille fut indispensable à leurs réalisations. Je dédie donc ces travaux à mes enfants, Mandoline, Evan et Maël, mes plus grandes fiertés, et à toi, MPP, pour ta présence au quotidien.

Table des matières

Synthèse	1
1 Curriculum Vitae	2
2 Activités pédagogiques :	3
3 Activités administratives	6
4 Activités de recherche	6
4.1 Traitement distribué de séries d'images	7
4.2 Modélisation et traitement de connaissances distribuées	9
4.3 Projets de recherche	11
4.4 Encadrements	11
4.5 Rayonnement scientifique	13
4.6 Liste des publications	14
 Travaux de recherche	 23
5 Chaînes de traitements de séries d'images	27
5.1 Contexte	28
5.2 Prétraitements	29
5.2.1 Sélection des images	29
5.2.2 Recalage des séries	30
5.3 Mesure de déplacement	32
5.3.1 Mesure par corrélation simple	32
5.3.2 Fermeture temporelle	36
5.3.3 Médian glissant sur de multiple « common/master » séries	38
5.3.4 Mesure en environnement virtuel	41
5.3.5 Rendu métrique	45
5.4 Approche conceptuelle des chaînes de traitements	46
5.5 Conclusion & réflexion	54

6	Architecture de traitement de l'habitude	55
6.1	Contexte	56
6.2	De la sagesse à l'intelligence	57
6.2.1	Acquisition de la sagesse	57
6.2.2	Infrastructure du traitement de la sagesse	61
6.3	Simulation de comportements	62
6.3.1	Timer	62
6.3.2	Action	65
6.3.3	Simulateur	67
6.4	Modélisation de l'habitude et détection de l'inhabituel	68
6.4.1	Modélisation de la mémoire	68
6.4.2	Mesure de la distance à l'habituel	70
6.5	Applications	71
6.5.1	Domotique : synthèse de contrôleurs	71
6.5.2	Domotique : Wise Internet of Things	73
6.6	Perspectives	75

PREMIÈRE PARTIE : SYNTHÈSE

1 Curriculum Vitae

Situation administrative

Nom : Vernier

Prénom : Flavien

Date & lieu de naissance :

Adresse :

Fonction : Maître de Conférences (MCF) – classe normale

Date d'installation :

Laboratoire : LISTIC – Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance – Polytech Annecy-Chambéry – Université Savoie Mont-Blanc

Établissement : Polytech Annecy Chambéry – Université Savoie Mont-Blanc

Parcours professionnel :

- 2006 – aujourd'hui : MCF, Université Savoie Mont-Blanc – Polytech Annecy Chambéry – LISTIC
- 2005 – 2006 : ATER, Université Henri Poincaré Nanci-I – Laboratoire lOrrain de Recherche en Informatique et ses Applications (LORIA)
- 2004 – 2005 : ATER, IUT de Belfort-Montbéliard – Laboratoire d'Informatique de L'université de Franche-Comté (LIFC)
- 2001 – 2004 : Moniteur, IUT de Belfort-Montbéliard – LIFC

Formation :

- 2001 – 2004 : Thèse en Informatique au sein de l'équipe Algorithmique Numérique Distribuée (AND) du laboratoire LIFC de l'Université de Franche-Comté
- 2000 – 2001 : DEA Systèmes Distribués et Réseaux (SDR) à l'Université de Franche-Comté
- 1997 – 2000 : IUP Informatique à l'Université de Franche-Comté

2 Activités pédagogiques :

Mon activité pédagogique a débuté au cours de mon année de DEA, où j'ai eu en charge des interventions en TD et TP, puis s'est poursuivie tout au long de mon activité professionnelle. J'ai ainsi pu dispenser un ensemble de cours diversifiés et côtoyer un public varié et hétérogène. La liste ci-après donne une vue d'ensemble non exhaustive des domaines dans lesquels je suis intervenu en enseignement.

- Algorithmique et programmation fonctionnelle, procédurale et objet
- Algorithmique distribuée
- Base de données relationnelle
- Conception UML
- Programmation de robots de services
- Théorie des graphes & Langages
- Logique & Programmation
- Informatique industrielle
- Initiation à la photogrammétrie
- Systèmes d'exploitation et réseaux
- Qualité, test et maintenance logiciels
- Technologies Web
- ...

Ces cours ont été dispensés à des publics de tout niveau post-Bac. Je suis intervenu dans des filières en licence informatique à l'Université de Franche-Comté lors de mon DEA, en IUT informatique lors de ma thèse et mon ATER au sein de l'IUT de Belfort-Montbéliard, en licence physique/chimie à l'Université Henri Poincaré Nancy lors de ma seconde année d'ATER. Depuis ma prise de poste à l'Université Savoie Mont-Blanc, la diversité du public est restée importante notamment par mon affectation à Polytech où j'interviens dans l'ensemble des filières – Environnement Bâtiment Énergie (EBE), Instrumentation Automatique Informatique (IAI), Informatique Données Usages (IDU), Mécanique et Mécatronique (MM), alternance Mécanique Productique (MP-ITII) – et niveaux de l'école (cycle préparatoire PEIP, cycle ingénieur). Depuis ma prise de poste, je suis également intervenu ou intervient encore au département de l'IUT d'Annecy, à l'Institut d'Administration des Entreprises (IAE) ainsi qu'au département Sciences et Montagne (SceM). Le tableau ci-après donne un aperçu quantitatif de mon enseignement sur les quatre dernières années universitaires, soit une charge annuelle moyenne d'environ 400 h/an effectives et environ 400 h équivalent TD en intégrant stages et décharges.

Matière	Cours (h/an)	TD/TP (h/an)	Public
Algorithmique, programmation fonctionnelle, procédurale et objet	9	20 30	L1 depuis 2015 Ingé-3 EBE, IAI, MM 2006-2016
Algorithmique distribuée	6	21	M2 Info depuis 2015
Base de données relationnelle	6	30	Ingé-3 EBE, IAI, MM depuis 2006
Conception UML		56	Ingé-3 IAI depuis 2014
Programmation de robots de services		16 12 20	Ingé-3 IAI, Ingé-5 ITII depuis 2018 M1 international master MT depuis 2017 Ingé-3, 4, 5 IAI APP RS depuis 2011
Théorie des Graphes & Langages	10 12	10 28	L3 IAE Systèmes d'info. 2008 à 2018 Ingé-3 IDU depuis 2018
Logique & Programmation	6	26	Ingé-3 IDU depuis 2018
Informatique industrielle		28	Ingé-5 alternants ITII depuis 2012
Initiation à la photogrammétrie	4.5	12.5	Ingé-4 IAI 2013-2016
Qualité, test et maintenance logiciels	9	30 12	Ingé-4 IAI 2008-2014 Ingé-4 IAI depuis 2016

Au cours de mes enseignements, principalement ceux dont j'étais responsable ou dont je faisais partie de l'équipe pédagogique qui les a mis en place, j'ai développé différentes approches pédagogiques en fonction du cours, du public visé et mon attrait pour les pédagogies innovantes. Je détaillerai ci-après trois cours que j'ai mis en place avec une réflexion pédagogique adaptée.

L'un des cours que je dispense depuis le plus long temps est celui d'initiation aux bases de données relationnelles. La particularité de ce cours est que je le dispense à l'ensemble des élèves ingénieurs de 3e année, soit environ 220 étudiants faisant partie des 4 spécialités de l'école. Les deux difficultés majeures de cet enseignement sont le nombre d'étudiants et la diversité de leurs origines. Sur une base classique imposée CM, TD et TP, j'ai fait évoluer ce cours grâce aux retours des étudiants avec comme objectif une structuration claire en 3 parties (CM) où chaque partie se conclut par les attendus de connaissance la concernant et est attaché à un TD permettant de mettre en pratique les connaissances vues dans le CM associé. Les 3 parties correspondent à la modélisation conceptuelle, la modélisation logique et l'algèbre relationnelle, ce qui me permet également de proposer une évaluation structurée sur ces trois parties. Les TP, quant à eux, sont une mise en pratique concrète sur des exemples réels et relatifs à leurs futurs métiers, ceci est mis en pratique par la différenciation des TP en fonction des filières. Ce cours est en amélioration continue, la structure précise intégrant clairement les attendus m'a permis de mettre en place, petit à petit, une évaluation où chacune des parties du cours doit être validée pour valider l'enseignement, ce qui m'a également permis d'aboutir à une évaluation par compétences.

L'un des cours les plus récents que j'ai eu à mettre en place est celui de « théorie des graphes » dispensé en licence 3 à l'IAE puis au sein de la nouvelle spécialité IDU de Polytech. Dans sa première version, « Arbres et Graphes », pour les étudiants de l'IAE, non-spécialistes du domaine informatique, mon approche fut de favoriser la mise en pratique. Chaque concept théorique abordé a été immédiatement mis en pratique ; pour ce faire, l'ensemble des CM et TD se sont réalisés en salles machines. Lors de son passage à la spécialité IDU, spécialisé dans le domaine, j'ai reformaté l'ensemble du cours afin qu'il soit au mieux adapté à ce nouveau public. J'ai notamment approfondi les concepts théoriques abordés, mais je l'ai également étendu à la théorie des langages. L'approche initiale de ce cours a été abordée de manière classique CM théorique, TD d'application directe de la théorie, un CM étant toujours associé à un TD et des TP de mise en application sur des problèmes concrets liés au futur métier des ingénieurs IDU. Afin de « casser » cette approche classique, j'ai mis en place en fin de module une séance d'auto-révision où chaque étudiant a été réparti dans des groupes qui ont eu pour but de concevoir une activité de révision sur l'une des parties du cours. Activités libres qui ont été des fiches de révision classique à un jeu de société. Ces activités ont par la suite, été présentées aux autres groupes pour que l'ensemble de la promotion ait des activités de révisions ludiques réalisées par eux même.

Ces deux cours précédemment présentés ont été mis en place par mes soins, mais validés par l'ensemble du groupe pédagogique intervenant dans les modules.

Le dernier enseignement que je détaillerai dans ce dossier est celui basé sur l'apprentissage par projet (APP). Une équipe pédagogique, dont je fais partie, a été mise en place à Polytech sous l'initiative de M. Jean-Jacques Curtelin. Nous avons ainsi, depuis plusieurs années, défini et fait évoluer un apprentissage pluriannuel (années 3 à 5) basé sur un projet. Les étudiants sont répartis selon leur choix motivé dans différentes thématiques : Bâtiment Intelligent (BI), Imagerie pour l'environnement (IE), Gestion des Énergies Renouvelables (GER), Robotique de Service (RS) et Santé (S). Chaque thème ouvert a ainsi un groupe de 5 à 10 étudiants qui travaillent, pendant 4 semestres (S6 à S9) sur un projet. Pour ma part, je suis responsable de la thématique RS, accompagné pour ce thème par 2 collègues. L'organisation semestrielle que j'ai mise en place au sein de la thématique RS est la suivante :

- S6 découverte des robots et leur programmation
- S7 définition des cahiers des charges et de conception
- S8 réalisation
- S9 valorisation

Cette organisation a l'avantage de montrer aux étudiants les différentes phases de la réalisation d'un projet d'ingénierie. Chaque semestre est pourvu de 40 h de projets, 60 h pour le S8, avec un encadrement de plus en plus lâche au cours des semestres afin de les mener vers l'autonomie. Notamment, au S6, ils ont un ensemble de cours imposés pour la manipulation des robots. Ces cours ayant toujours comme objectif de les mener à l'autonomie, ils suivent ainsi toujours la même trame :

1. présentation des consignes de sécurité pour l'humain, mais surtout pour le robot,
2. illustration des modes de programmation des robots par l'exemple,
3. réalisation, par les étudiants, d'un comportement nécessitant l'utilisation de la documentation du robot.

Le dernier point étant, de mon point de vue, le plus important, car il permet aux étudiants de prendre conscience qu'ils sont capables d'aller au-delà de la formation initiale de prise en main des robots, en s'appuyant sur les concepts de base que nous avons abordés et une documentation. L'autonomie ne signifie pas travailler seul, nous sommes toujours présents pour suivre leur travail et les guider, dans ce cadre, nous avons mis en place un système de jetons que les étudiants peuvent utiliser à leur guise pour faire appel à des experts enseignants lorsqu'ils sont bloqués sur un problème où ils n'ont pas les compétences. Les jetons qu'ils affectent aux experts se transforment en heures rémunérées pour ces derniers. Le projet se construit ainsi jusqu'au S9, où je leur demande une valorisation, c'est-à-dire fournir un produit avec une ou plusieurs communications. Ce dernier semestre est le plus ouvert et leur laisse une grande liberté dans leurs réalisations. Au cours des différentes années et promotions, certains ont réalisé des « goodies » associés à leur projet, des affiches publicitaires, des posters explicatifs, des publicités...

Suite à la mise en place de cette approche pédagogique dirigée par le projet, je l'ai décliné en une forme plus concise, en une journée de découverte de la robotique de service. Cette journée est décomposée de la même manière que les APP RS : 2 h de présentation d'un robot (consignes de sécurité et un mode de programmation), 4 h de réalisation d'un mini-projet libre et 2 h d'un mini projet imposé faisant guise d'évaluation. Cette découverte de la robotique de service est dispensée en anglais au sein de différentes filières attachées à Polytech : 5e année de Master international « Advence Mecatronic », 3e année de la filière IAI et 5e année de la filière en alternance ITII.

Outre ces trois cours sur lesquels je me suis attardé, j'interviens activement dans mes enseignements, à l'écoute de mes élèves afin de leur apporter la pédagogie qui leur sera le plus adaptée. Je favorise généralement la pratique, afin d'en extraire les concepts théoriques. J'interviens ainsi dans un certain nombre de projets, le dernier que j'ai mis en place étant dédié à la nouvelle filière IDU3, dans le cadre d'une mise en pratique concrète du cours de « Graphes & Langages » que je dispense.

Mon engagement pédagogique m'amène également à suivre 3 à 5 stages étudiants de 4e et 5e années de la filière IAI chaque année et suivre en permanence 2 étudiants alternants de la filière ITII. Je participe également, chaque année, au recrutement par entretien ou dossier des étudiants PEIP, ingénieurs et alternants. Pour terminer, j'ai été responsable du 8e semestre de la spécialité IAI de 2012 à 2018, je suis responsable de la 4e année de la spécialité IDU depuis 2019 et je suis responsable de la discipline informatique de Polytech (50 modules pour 17 enseignants) depuis 2012.

3 Activités administratives

Responsabilités au sein de Polytech Annecy-Chambéry

- Responsable d'année de la Spécialité IDU (depuis 2019)
- Responsable de semestre de la Spécialité IAI (2012 – 2018)
- Responsable des projets Robotique de Services (depuis 2011)
- Responsable de la discipline informatique et génie informatique (depuis 2012)

Responsabilités au sein du laboratoire LISTIC

- animateur scientifique du groupe de recherche Répresentation, Gestion et traitement des Données pour l'humain (ReGaRD) (depuis 2019)
- animateur scientifique du groupe de recherche Réseau et Systèmes Logiciels Répartis (RSLR) (2015 – 2018)
- Membre du bureau du laboratoire (depuis 2015)
- Membre du conseil de laboratoire (depuis 2011)
- Membre de la commission informatique (depuis 2009)

4 Activités de recherche

Dans cette partie, je présenterai succinctement mes travaux de recherche en rappelant mes origines doctorale et postdoctorale sur lesquelles se base mon activité de recherche et je détaillerai plus les travaux que j'ai effectués depuis ma nomination au LISTIC en 2006. Ces travaux, par souci de cohérence, ne seront pas présentés par ordre chronologique, mais selon un fil conducteur conceptuel.

Mon activité de thèse et postdoctorale s'organisait autour de la thématique des systèmes distribués et de l'algorithmique distribuée. J'ai principalement travaillé sur la problématique de l'équilibrage de charge lors de la résolution asynchrone de systèmes d'Équations Différentielles Partielles (EDP). Ces travaux ont principalement mis en évidence la complémentarité de l'équilibrage de charge et l'asynchronisme dans les systèmes distribués, ces deux approches ayant le même but : accélérer la résolution distribuée et itérative d'un problème modélisé sous forme d'EDP [Bahi et al., 2009]. Ces travaux m'ont amené à étudier différents problèmes algorithmiques distribués comme la détection asynchrone de convergence d'un algorithme asynchrone [Bahi et al., 2005] ou encore l'exclusion mutuelle partielle de données distribuées [Quinson, Vernier, 2009].

Publications Relatives

[Bahi et al., 2005] Bahi J. M., Contassot-Vivier S., Couturier R., **Vernier F.** (2005). A decentralized convergence detection algorithm for asynchronous parallel iterative algorithms. IEEE Transactions on Parallel and Distributed Systems, vol. 16, pp. 4–13.

[Bahi et al., 2009] Bahi J. M., Couturier R., **Vernier F.** (2009). Synchronous load balancing on asynchronous iterative computation. Journal of Algorithms and Computational Technology, vol. 3, No. 1, pp. 135–153.

[Quinson, Vernier, 2009] Quinson M., **Vernier F.** (2009, février). Byte-Range Asynchronous Locking in Distributed Settings. In 17th Euromicro International Conference on Parallel, Distributed and

network-based Processing–PDP 2009. Weimar, Germany.

4.1 Traitement distribué de séries d’images

À mon arrivée au LISTIC, j’ai intégré le groupe de recherche « Réseaux et Systèmes Logiciels Répartis » (RSLR) et débuté des travaux transversaux avec le groupe « Connaissance, Image et Télé-détection » (CIT). Ces travaux se sont déroulés initialement au sein du projet ANR EFIDIR, l’objectif étant de définir et développer un cadre logiciel pour le traitement de séries d’images de manière normalisée, optimisée et distribuée. La problématique venant de la nature intrinsèque des images traitées, le projet EFIDIR se focalisant sur les images satellitaires radar à synthèse d’ouverture (SAR). La particularité de ces images étant leur format – données complexes – et leur taille, de quelques centaines de mégaoctets à plusieurs gigaoctets par image. Ces travaux ont été réalisés au travers d’encadrement de différentes thèses et de projets en collaboration avec d’autres laboratoires présentés ci-après.

La mise en place de chaînes de traitements nécessite un formalisme de description, interprétable, du point de vue informatique, pour son exécution. Mes travaux en collaboration avec Matthieu Volat, ingénieur CNRS puis UGA au laboratoire ISTerre (Grenoble), ont permis de mettre en place ce formalisme de description en nous appuyant sur le langage Python et l’outil « Make » pour l’exécution des chaînes de traitements. Le point fort de notre approche est que l’exécution d’une chaîne sur une machine de calcul multicœur est complètement transparente pour l’utilisateur. Quant à une exécution distribuée, la configuration se limite à fournir la liste des machines utilisables. Dans tous les cas, notre système gère la répartition des tâches à exécuter en tenant compte de leur dépendance et du nombre de nœuds de calcul disponibles. Cet outil, que nous nommons « dispacheur », est disponible au sein de EFIDIR-Tools1 [Volat et al., 2014] et a été déployé dans la chaîne de traitements NSBAS développée au laboratoire ISTerre et intégrée à l’atelier Multi Satellite, multi-Capteurs, pour des données multi-Temporelles (MUSCATE) du CNES dans le cadre de Form@Ter.

Traitement des Séries temporelles d’images optiques pour la mesure de déplacement d’objets géophysiques L’un des premiers outils que j’ai réalisés dans les EFIDIR-Tools a été l’optimisation, la parallélisation et la distribution d’un algorithme de corrélation [Vernier et al., 2011]. L’optimisation a été étudiée via une implémentation s’appuyant sur un jeu de fenêtres glissantes afin d’éliminer tout calcul redondant au cours de l’exécution de l’algorithme. La parallélisation offre, au travers d’une version dédiée, une implémentation multi-threads prenant en charge la gestion des systèmes multicœurs. La distribution quant à elle, a été mise en place via une chaîne de traitements qui découpe les images à corrélérer, génère une tâche par découpe et fusionne les résultats. Cette chaîne a fortement été utilisée au cours des thèses de Renaud Fallourd et Yajing Yan, thèses encadrées par des collègues des laboratoires pluridisciplinaires LISTIC, Télécom ParisTech et ISTerre.

Dans le cadre de la thèse d’Ha Thai Pham, financée par le gouvernement vietnamien et le LISTIC, nous nous sommes intéressés à la mise en place et au déploiement de chaînes de traitements pour la mesure de déplacement d’objets géophysiques tels que les glaciers, par photogrammétrie terrestre. Nous nous sommes focalisés sur la chaîne dans sa globalité [Benoit et al., 2015]. C’est-à-dire la définition des prétraitements indispensables, la mesure de déplacement et les post-traitements nécessaires afin de rendre les résultats le plus exploitables possible par un utilisateur non spécialiste du traitement numérique.

Dans la continuité des travaux d’Ha Thai Pham, nous avons poursuivi le déploiement de chaînes de traitements pour la mesure de déplacement d’objets géophysiques, au travers de la thèse d’Hela

Hadhri financée sur le projet ANR Phoenix, en nous focalisant sur l’automatisation de différentes phases du traitement. L’un des points majeurs de ces travaux de thèse concerne le calcul des champs de déplacements de l’objet observé. Dans ses travaux, Hela a mis en place une approche redondante d’un point de vue spatio-temporel afin de réduire les erreurs de calcul [Hadhri et al., 2019]. Nous avons également, au travers des travaux de stage de Master de Guilhem Marsy, automatisé la sélection des images initiales. En effet, une des particularités des images acquises automatiquement en milieu naturel est qu’elles peuvent être perturbées par différents phénomènes naturels (pluie, brouillard, éclat de lumière...), ce qui les rend inutilisables.

Les chaînes de traitements de séries d’images pour la mesure de déplacement d’objets géophysiques devenant fonctionnelles, en collaboration avec l’entreprise Tenevia et le laboratoire EDYTEM, nous avons entamé la thèse CIFRE de Guilhem Marsy en vue de l’opérationnalisation et la mise en place de dispositifs matériels et de traitements pour la surveillance des risques et aléas. Les travaux de Guilhem ont porté sur les préétudes d’installation des systèmes d’observation, afin de prédire les capacités d’observation et de mesure du système à installer, sur l’ensemble de la chaîne monoscopique, mais également sur la partie stéréoscopique des traitements et le rendu des mouvements 3D de l’objet observé [Marsy et al., 2018].

L’ensemble de mes travaux sur les chaînes de traitements ont naturellement été publiés dans différentes revues et conférences applicatives dont une sélection est donnée ci-après, mais a également donné lieu à des ateliers pratiques dans deux écoles de printemps et ont été une base de travail lors des thèses de Fanny Ponton et Hoang Viet Tuan Nguyen, thèses encadrées par des collègues des laboratoires LISTIC, ISTerre, EDYTEM, GIPSA et LIRIS.

Approche virtuelle pour la mesure de déplacement d’objets géophysiques Avec Haixing He, en collaboration avec le laboratoire EDYTEM et dans le cadre d’une allocation de thèse de l’établissement (USMB), nous avons abordé la problématique de la mesure de déplacement d’objets géophysiques d’une manière originale. Ayant constaté les problèmes liés aux installations photogrammétriques in situ et en nous appuyant sur l’explosion des systèmes d’observation par drones, nous avons abordé la problématique par une approche multivue. L’originalité de notre approche tient au fait que la mesure ne se fasse plus sur les données « brutes » (les images), mais dans un environnement virtuel : la reconstruction 3D obtenue à partir des acquisitions multivues. À partir de modèles 3D multitudes, nous avons le contrôle total, sans phénomène extérieur, afin d’effectuer les mesures de déplacement 3D des objets observés [He et al., 2015].

Prise de contrôle de programmes Outre mes travaux sur les chaînes de traitements de séries temporelles et afin de mettre en place ces dernières, j’ai étudié et développé un système permettant de prendre le contrôle automatique du lancement d’un exécutable. L’idée de base repose sur le fait que l’exécution d’un programme est conditionnée par ses paramètres. Il est donc nécessaire de les connaître. Si un programme tiers veut contrôler le lancement d’un programme, par exemple au sein d’une chaîne de traitements, il doit connaître ses paramètres de manière statique, c’est-à-dire fournis par le développeur, ce qui est peu satisfaisant, ou de manière dynamique en interrogeant le programme à exécuter. Pour ce faire, le programme interrogé doit se connaître lui-même afin de fournir ses paramètres. Ainsi j’ai défini et implémenté une bibliothèque de gestion de paramètre qui permet à un développeur de définir de manière simple – déclarative – les paramètres de ses programmes. L’intérêt est que, par l’utilisation de la bibliothèque, le programme peut être interrogé et fournir l’ensemble de ses paramètres métiers (définis par le développeur), l’information fournie est, pour l’heure, standardisée dans un format XML et

intègre le nommage, le typage, le caractère optionnel et la documentation associée à chaque paramètre. Grâce à ces informations, la bibliothèque génère également de manière automatique la documentation associée au programme. Ces travaux sont intégralement intégrés aux EFIDIR Tools et à leurs chaînes de traitements et permettent une sécurité de typage et d’obligation des paramètres et une interaction avec l’utilisateur en cas de défaillance de l’un d’eux. Une des perspectives de ces travaux est, en particulier, l’analyse des exécutions en vue d’une prédiction des temps d’exécution en fonction des paramètres fournis, ceci grâce à un apprentissage basé sur les exécutions précédemment réalisées.

Publications Relatives

[Benoit et al., 2015] Benoit L., Dehecq A., Pham H.-T., **Vernier F.**, Trouvé E., Moreau L. et al. (2015). Multi-method monitoring of Glacier d’Argentière dynamics. *Annals of Glaciology*, vol. 56, No. 70, pp. 118–128.

[Hadhri et al., 2019] Hadhri H., **Vernier F.**, Atto A., Trouvé E. (2019, avril). Time-lapse optical flow regularization for geophysical complex phenomena monitoring. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 135–156.

[He et al., 2015] He H., **Vernier F.**, Villemin T., Ployon E., Deline P., Morra Di Cella U. (2015, juillet). 3D displacement retrieval on glacial areas by airborne photogrammetry. In *8th International Workshop on the Analysis of Multitemporal Remote Sensing Images*, vol. Analysis of Multitemporal Remote Sensing Images, pp. 1–4. Annecy, France.

[Marsy et al. 2018] Marsy G., Bodin X., Malet E., Castaings W., **Vernier F.**, Trouvé E. (2018, juin). Monitoring rock glaciers by optical stereoscopic time-lapse device. *5th European Conference on Permafrost*. Bourget du Lac, France.

[Vernier et al., 2011] **Vernier F.**, Fallourd, R., Friedt, J.M. et al. (2011, novembre). Fast correlation technique for glacier flow monitoring by digital camera and space-borne SAR images. *EURASIP Journal on Image and Video Processing*.

[Volat et al., 2014] Volat M., **Vernier F.**, Doin M.-P., Lasserre C., Trouvé E., Pathier E. (2014, novembre). Improving the execution of workflows for SAR image analysis. In *Conference on Big Data from Space*. Frascati, Italy.

4.2 Modélisation et traitement de connaissances distribuées

Le contrôle des programmes via leurs paramètres est à la base des travaux que j’ai initiés sur la modélisation et le traitement de connaissances distribuées. S’il est possible de contrôler un programme via ces paramètres, il en va de même pour un objet dans un paradigme orienté objet via le contrôle de ses attributs. Ainsi, il est possible d’acquérir de la connaissance sur l’utilisation de chaque objet qui constitue un programme.

Les objets Sages À partir de cette réflexion, j’ai lancé en 2014, en collaboration avec Ilham Alloui, enseignante/chercheuse au LISTIC, une nouvelle activité de recherche au sein du LISTIC. La problématique de base étant la modélisation, la conception et l’implémentation d’objets logiciels capable d’acquérir leur propre connaissance sur eux même ; par analogie humaine, nous nommons ces objets des objets sages (WO pour Wise Object) [Alloui et al., 2015]. Cette activité s’intègre pleinement dans les axes principaux de recherche du laboratoire, puisqu’elle allie un caractère distribué de la connaissance où chaque objet a sa propre connaissance, une problématique de modélisation de la connaissance et

pour finir un problème de fusion afin que le système acquière sa propre connaissance sur lui même en fusionnant celles des objets qui la constituent.

Dans le cadre de ses travaux, nous avons développé un « framework » objet fondé sur le concept d'Objet Sage (Wise Object Framework : WOF), implanté en Java. WOF permet la création d'objets sages [Alloui, Vernier, 2017], avec une forte capacité d'introspection et d'analyse comportementale. Il permet la création de systèmes sages, c'est-à-dire des systèmes composés d'objets sages. Dans le domaine de l'Internet des objets « Internet of Things (IoT) » [Alloui et al., 2018], WOF nous permet d'associer à travers une interface logicielle dédiée, des avatars logiciels (WO) aux objets physiques et/ou logiciels. Ces avatars apprennent de manière autonome, grâce aux mécanismes d'introspection et de monitoring fournis par WOF, sur le comportement des objets IoT et sur l'usage qu'on en fait. Ces travaux se sont, jusqu'à présent, appuyés sur des stages de Master qui ont, pour certain, donné lieu à publications. Début octobre 2020, la thèse d'Abdelhafid Dahhani a débuté et porte plus particulièrement sur la modélisation de la connaissance – la sagesse de WO – et sa fusion vers de plus hauts niveaux sémantiques.

Vers une domotique adaptative Bien qu'antérieurs à mes travaux sur les objets sages, les travaux que nous avons menés avec Fabien Sartor dans le cadre de sa thèse APS avec la société Overkiz (thèse type CIFRE) sont pleinement liés aux objets sages et à leur application à la domotique. Au cours de ces travaux, nous nous sommes intéressés à la scénarisation des comportements domotiques d'une maison. En nous basant sur une représentation par graphes d'états des différents éléments domotiques de la maison, nous avons étudié la composition des graphes des différents éléments interagissant afin de proposer des systèmes composés de plus hauts niveaux et avons ainsi proposé une nouvelle approche originale de la synthèse de contrôleurs avec un point de vue génie logiciel en nous appuyant sur une connaissance de base des éléments du système : les graphes États/Transitions qui seront fournis par le développeur logiciel [Moreaux et al., 2012]. Les objets sages apportent à ces travaux une garantie de cohérence entre le code exécuté et le graphe de comportement – graphe d'états – d'un élément puisque l'objet extrait automatiquement ce graphe du code et n'est plus fourni par le développeur. Cependant, les WO perdent tout l'aspect sémantique que peut apporter un graphe fourni par un humain.

Publications Relatives

[Alloui et al., 2018] Alloui I., Benoit E., Perrin S., **Vernier F.** (2018, juillet). WIoT : Interconnection between Wise Objects and IoT. In ICISOFT 2018, the 13th International Conference on Software Technologies. Porto, Portugal.

[Alloui et al., 2015] Alloui I., Esale D., **Vernier F.** (2015, juillet). Wise Objects for Calm Technology. In 10th International Conference on Software Engineering and Applications (ICSOFT-EA 2015), pp. 468–471. Colmar, France, SciTePress 2015.

[Alloui, Vernier 2017] Alloui I., **Vernier F.** (2017, juillet). A Wise Object Framework for Distributed Intelligent Adaptive Systems. In ICISOFT 2017, the 12th International Conference on Software Technologies. Madrid, Spain.

[Alloui, Vernier, 2018] Alloui I., **Vernier F.** (2018, juin). WOF : Towards Behavior Analysis and Representation of Emotions in Adaptive Systems. Communications in Computer and Information Science, vol. 868, pp. 244–267.

[Moreaux et al. 2012] Moreaux P., Sartor F., **Vernier F.** (2012, février). An Effective Approach for Home Services Management. In Parallel, Distributed and Network-Based Processing (PDP), pp. 47–51. Garching, Germany, IEEE.

4.3 Projets de recherche

- AAP USMB 2021 : « in NetwORk MACHine Learning (NORMAL) » (budget 14 000 €) – contributeur
- ANR IA AIACOPAHV : « Application de l'Intelligence Artificielle à la COMpréhension et la Prédiction de l'Activité Humaine et Volcanique » (2020-2024) (budget USMB 324 000 €) – contributeur
- AAP USMB 2020 : « DeEp DIstribué Connectés Aide mainTiEn Domicile (DEDICATED) » (budget 18 000 €) – contributeur
- ANR VIP Mont-Blanc : « Vitesses des Processus contrôlant les évolutions morphologiques et environnementales du massif du Mont-Blanc » (2014-2019) (budget LISTIC 16 000 €) – responsable scientifique LISTIC
- ANR Phoenix : « Parsimony, Huge Observations of Earth Non-stationarities from Images Time Series » (2015-2019) (budget LISTIC 280 000 €) – contributeur
- AAP USMB 2019 : Instrumentation Répartie Connectée (IRC) (budget 6 500 €) – porteur
- AAP USMB 2018 : COMunauté sage d'objets connectés pour la mesure d'hAbitude de l'usager (COMDA) (budget 6 300 €) – contributeur
- AAP USMB 2017 : Modélisation de la connaissance pour les objets sages (Mc-WO) (budget 7 000 €) – porteur
- CNRS Défis Mastodons : « Tera_SAR : Massification du calcul des déformations de surface par imagerie radar multitemporelle, multispectrale et haute résolution » (2013-2014-2015) (budget LISTIC 15 000 €) – responsable scientifique LISTIC
- ANR EFIDIR : « Extraction et Fusion d'Informations pour la mesure de Déplacements par Imagerie Radar » (2008-2012) (budget LISTIC 165 000 €) – responsable développements informatiques

4.4 Encadrements

Masters

- 6 étudiants sur le thème des Objets Sages « Wise Object (WO) »
 - Réalisation d'un « framework » pour l'implémentation de WO, David Esale Yoka (2015) [Alloui et al., 2015]
 - Réalisation d'un simulateur de comportement des WO, Tiphaine Bulou (2016) [Alloui, Vernier, 2017]
 - Analyse des log d'un WO en vue d'en extraire une habitude comportementale, Abderrahmann Boudrai Uahid (2016), Ilyass Zaki (2016), Koffi-Martial Konan (2017) [Alloui, Vernier, 2018]
 - « Monitoring » d'un système de WO, Crina Cioara (2018)
- 13 étudiants sur projets R&D :
 - Reconception d'un robot Peeke I sur une base Raspberry Pi et Arduino, Pierre Le Borgne & Enzo Bouilly (2016), Ludovic Chapelet & Arnaud Mignotte (2017)
 - Conception d'un « framework » pour la fusion floue d'informations, Nicolas Java & Damien Morel (2018)
 - Développement d'un système d'acquisition d'image autonome et connecté, Nawar Remmo, Fabien Lalande, Erwan Prospert (2019-2020)
 - Conception d'un outil de gestion des enseignants, Robert Charavit (2019-2020)

- Re-factoring d'un outil de gestion des enseignants, Maxence Beaurez, Théophane Boué, Quentin Potet (2020-2021)
- 4 étudiants sur l'implémentation de chaînes de traitements de séries d'images SAR et optique au sein des EFIDIR-Tools
 - Géolocalisation de photographies numériques et intégration dans les systèmes d'informations géographiques, Haixing He (2012)
 - Traitements de séries temporelles de photos pour quantifier les champs de déplacements de glaciers rocheux, Guilhem Marsy (2016)
 - Développement d'un système d'acquisition d'image autonome et connecté, Nawar Remmo, Hajar Bentahir (2019)

Thèses

- Abdelhafid Dahhani, « Représentation et gestion de la connaissance pour les systèmes logiciels intelligents », coencadrée à 40 % avec Ilham Alloui et Sébastien Monnet, thèse en cours depuis 2020
- Étienne Mauffret, « Gestion de données dans les systèmes de stockage à larges échelles : tolérance aux fautes, placement et cohérence modulaire », coencadrée à 40 % avec Sébastien Monnet, thèse en cours depuis 2017
- Guilhem Marsy, « Apport de l'imagerie optique <Time-Lapse> stéréoscopique pour la quantification à haute résolution spatio-temporelle (4D) des dynamiques de versants de montagne », coencadrée à 40 % avec Emmanuel Trouvé et Xavier Bodin, thèse soutenue le 22-09-2020
- Hela Hadhri, « Analyse de séries temporelles d'images stéréoscopiques : application à l'observation des milieux naturels. », coencadrée à 40 % avec Abdourrahmane Atto et Emmanuel Trouvé, thèse soutenue le 25-06-2020
- Haixing He, « Application de la photogrammétrie par multicorrélation d'images pour le suivi d'évolutions morphologiques rapides en milieux rocheux ou englacés. », coencadrée à 80 % avec Thierry Villemin, thèse soutenue le 17-01-2017
- Ha Thai Pham, « Analyse de <Time-Lapse> optiques stéréo et d'images radar satellitaires: application à la mesure du déplacement de glaciers. », coencadrée à 60 % avec Emmanuel Trouvé, thèse soutenue le 24-02-2015
- Fabien Sartor, « Modélisation de l'interopérabilité d'objets communicants et de leur coopération: application à la domotique. », coencadrée à 60 % avec Patrice Moreau, thèse soutenue le 05-07-2012

Ingénieurs d'étude & recherche

- Co-encadrement des ingénieurs du projet ANR EFIDIR
 - Sophie Martz (2010) : conception et implémentation des outils de base des EFIDIR-Tools
 - Felicity Lodge (2011) : conception et implémentation des outils de base des EFIDIR-Tools
- Co-encadrement d'un ingénieur sur les projets CNRS MASTODONS
 - Matthieu Volat (2013-2017) : conception, implémentation et déploiement des chaînes de traitements des EFIDIR-Tools

4.5 Rayonnement scientifique

Jury de thèse

- Dr Lama Sleem « Design and implementation of lightweight and secure cryptographic algorithms for embedded devices », thèse soutenue le 17 janvier 2020 (examinateur).
- Dr Soumeiya-Leila Hernane « Modèles et algorithmes de partage de données cohérentes pour le calcul parallèle distribué à haut débit. », thèse soutenue le 20 juin 2013 (examinateur).
- Dr Renaud Fallourd « Suivi des glaciers alpins par combinaison d'informations hétérogènes : images SAR Haute Résolution et mesures terrain. », thèse soutenue le 4 avril 2012 (examinateur).
- Dr Abderrahmane Sider, « Contributions to load balancing : a most to least loaded strategy and influence of data transfers on convergence time. », thèse soutenue en Algérie le 1er septembre 2011 (rapporteur).

Animations/Formation

- Session chair IGARSS 2020 « IEEE International Geoscience and Remote Sensing Symposium », Virtual Symposium, September 26 – October 2 2020
- Session chair ICSoft 2018 « Software Engineering and Systems Development », Porto (Portugal), 26 au 28 juillet 2018
- École Thématique « Image mining and information extraction from multi-source data : theory and application in geosciences », Barcelonnette (France), 26 au 30 août 2013
- Séminaire/atelier « Photogrammétrie terrestre & dynamique de surface en montagne », Barcelonnette (France), 15 et 16 octobre 2013
- EFIDIR Spring School, « Practical Work – EFIDIR-Tools : Phase Filtering and Phase Unwrapping using 2D local frequency », les Houches school of physics, Chamonix (France), 1er au 6 mai 2011
- Session chair PDP 2009 « 17th Euromicro International Conference on Parallel, Distributed and Network-Based Processing », Weimar (Germany), 18-20 February 2009

Comité de programmes

- Depuis 2019 : The Sixth International Conference on Advances and Trends in Software Engineering (SOFTENG)
- Depuis 2019 : International Conference on High Performance Computing & Simulation (HPCS)
- Depuis 2019 : International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOM)
- Depuis 2018 : Conférence Française de Photogrammétrie et de Télédétection (CFPT), conférence annuelle de la Société Française de Photogrammétrie et de Télédétection (SFPT)
- Depuis 2013 : journées de jeunes chercheurs en vision artificielle ORASIS de l'Association Française pour la Reconnaissance et l'Interprétation des Formes (AFRIF)

Comité d'organisation

- IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (IEEE CIMSA), Annecy (France), 26 au 28 juin 2017
- International Workshop on the Analysis of Multitemporal Remote Sensing Images (Multi-temp), Annecy (France), 22 au 24 juillet 2015
- Séminaire/atelier « Photogrammétrie terrestre & dynamique de surface en montagne », Barcelonnette (France), 15 et 16 octobre 2013
- EFIDIR Spring School, les Houches school of physics, Chamonix (France), 1er au 6 mai 2011

Relecteur pour revues

- ISPRS Journal of Photogrammetry and Remote Sensing
- IEEE Transactions on Parallel and Distributed Systems (TPDS)
- Journal of Computational Science (JOCS), Elsevier
- EURASIP Journal on Image and Video Processing, Springer

4.6 Liste des publications

Synthèse des publications :

Type	ACL	ACLN	OS	ACTI	ACTN	COM
Nombre	11	2	3	24	4	2

ACL : Article en revue internationale avec comité de lecture

ACLN : Article en revue nationale avec comité de lecture

OS : Ouvrage scientifique

ACTI : Communication avec actes dans un congrès international

ACTN : Communication avec actes dans un congrès national

COM : Communication orale sans actes

Article de journal international

2019

[ACL 1] Ilham Alloui, Eric Benoit, Stephane Perrin, **Flavien Vernier**. Wise Objects for IoT (WIoT) : Software Framework and Experimentation. Communications in Computer and Information Science, Springer Verlag, 2019, Software Technologies 13th International Joint Conference, ICSOFT 2018, Revised Selected Papers, pp.349-371. (hal-02274782)

(Computer Science : ranking 19967, H-index 40, Impact-Factor 0,49)

[ACL 2] Héla Hadhri, **Flavien Vernier**, Abdourrahmane Atto, Emmanuel Trouvé. Time-lapse optical flow regularization for geophysical complex phenomena monitoring. ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier, 2019, 150, pp.135-156. (hal-02057126)

(Computer Science : ranking 665, H-Index 110, Impact-Factor 8,19)

2018

[ACL 3] Ilham Alloui, **Flavien Vernier**. WOF : Towards Behavior Analysis and Representation of Emotions in Adaptive Systems. Communications in Computer and Information Science, Springer Verlag, 2018, Software Technologies 12th International Joint Conference, ICSOFT 2017, Revised Selected Papers, 868, pp.244-267. (10.1007/978-3-319-93641-3_12). (hal-01871009)
(*Computer Science : ranking 19967, H-index 40, Impact-Factor 0,49*)

2016

[ACL 4] Yajing Yan, Amaury Dehecq, Emmanuel Trouvé, Gilles Mauris, Noel Gourmelen, **Flavien Vernier**. Fusion of Remotely Sensed Displacement Measurements : Current status and challenges. IEEE geoscience and remote sensing magazine, IEEE, 2016, 4 (1), pp.6-25. (10.1109/MGRS.2016.2516278). (hal-01303960)
(*Geoscience & Computer Science : ranking 841, H-index 19, Impact-Factor 7,16*)

2015

[ACL 5] Lionel Benoit, Amaury Dehecq, Ha-Thai Pham, **Flavien Vernier**, Emmanuel Trouvé, et al. Multi-method monitoring of Glacier d'Argentiére dynamics. Annals of Glaciology, International Glaciological Society, 2015, 56 (70), pp.118-128. (10.3189/2015AoG70A985). (hal-01154174)
(*Geoscience : ranking 2730, H-index 72, Impact-Factor 3,06*)

2014

[ACL 6] Fanny Ponton, Emmanuel Trouvé, Michel Gay, Andrea Walpersdorf, Renaud Fallourd, Jean-Marie Nicolas, **Flavien Vernier**, et al. Observation of the Argentiére Glacier Flow Variability from 2009 to 2011 by TerraSAR-X and GPS Displacement Measurements. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, IEEE, 2014, pp.3274-3284. (10.1109/JS-TARS.2014.2349004). (hal-01118440)
(*Geoscience & Computer Science : ranking 2224, H-index 64, Impact-Factor 3,77*)

2011

[ACL 7] **Flavien Vernier**, Renaud Fallourd, Jean-Michel Friedt, Yajing Yan, Emmanuel Trouvé, et al. Fast correlation technique for glacier flow monitoring by digital camera and space-borne SAR images. EURASIP Journal on Image and Video Processing, Springer, 2011, 11, Open access. (10.1186/1687-5281-2011-11). (hal-00648027)
(*Computer Science : ranking 12658, H-index 33, Impact-Factor 1,93*)

[ACL 8] Renaud Fallourd, Olivier Harant, Emmanuel Trouvé, Jean-Marie Nicolas, Michel Gay, Andrea Walpersdorf, Jean-Louis Mugnier, Jonathan Serafini, Diana Rosu, Lionel Bombrun, Gabriel Vasile, Nathalie Cotte, **Flavien Vernier**, et al. Monitoring Temperate Glacier Displacement by Multi-Temporal TerraSAR-X Images and Continuous GPS Measurements. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, IEEE, 2011, 4 (2), pp.372-386. (10.1109/JS-TARS.2010.2096200). (halsde-00554873)
(*Geoscience & Computer Science : ranking 2224, H-index 64, Impact-Factor 3,77*)

2009

[ACL 9] Jacques M. Bahi, Raphaël Couturier, **Flavien Vernier**. Synchronous load balancing on asynchronous iterative computation. Journal of Algorithms and Computational Technology, 2009, 3 (1), pp.135-153. (hal-00620883)
(*Computer Science : ranking 20424, H-index 9, Impact-Factor 0,64*)

2005

[ACL 10] Jacques M. Bahi, Raphaël Couturier, **Flavien Vernier**. Synchronous Distributed Load Balancing on Dynamic Networks. Journal of Parallel and Distributed Computing, Elsevier, 2005, 65 (11), pp.1397-1405. (hal-00620884)

(Computer Science : ranking 11014, H-index 73, Impact-Factor 2,25)

[ACL 11] Jacques M. Bahi, Sylvain Contassot-Vivier, Raphaël Couturier, **Flavien Vernier**. A decentralized convergence detection algorithm for asynchronous parallel iterative algorithms. IEEE Transactions on Parallel and Distributed Systems, Institute of Electrical and Electronics Engineers, 2005, 16, pp.4-13. (hal-00096362)

(Computer Science : ranking 5262, H-index 120, Impact-Factor 4,64)

Article de journal francophone

2018

[ACLN 1] Guilhem Marsy, **Flavien Vernier**, Xavier Bodin, William Castaings, Emmanuel Trouvé. Détection automatique de zones en mouvement dans des séries d'images non recalées : application à la surveillance des mouvements gravitaires. Revue Française de Photogrammétrie et de Télédétection, Société Française de Photogrammétrie et de Télédétection, 2018, pp. 25-31. (hal-01887600)

(Computer Science : ranking 24214, H-index 7, Impact-Factor 0,05)

2011

[ACLN 2] Emmanuel Trouvé, Jean-Marie Nicolas, Laurent Ferro-Famil, Michel Gay, Virginie Pinel, Marie-Pierre Doin, Nicolas Méger, Cécile Lasserre, Gilles Mauris, **Flavien Vernier**, et al. EFIDIR : extraction et fusion d'informations pour la mesure de déplacements par imagerie radar. Traitement du Signal, Lavoisier, 2011, 28 (3-4), pp.375-416. (10.3166/ts.28.375-416). (hal-00646032)

(Computer Science : ranking 28903, H-index 3, Impact-Factor 0,42)

Chapitre de livre

2017

[OS 1] Xavier Bodin, Emmanuel Malet, **Flavien Vernier**, Guilhem Marsy, Héra Hadhri, et al. L'apport d'appareils photo reflex autonomes pour le suivi quasi-continu de la dynamique des glaciers rocheux. Exemple dans les Alpes et dans les Andes. Emmanuel Malet ; Laurent Astrade. Monitoring en milieu naturel : retours d'expériences en terrains difficiles., Collection EDYTEM, pp. 47-54, 2017, 978-2-918435-11-2. (hal-02071517)

2014

[OS 2] Yajing Yan, V. Pinel, **Flavien Vernier**, Emmanuel Trouvé. Mesures de déplacement. Florence Tupin, Jean-Marie Nicolas, Jordi Inglada Eds. Imagerie de télédétection, Hermes Science - Lavoisier, pp. 269-301, 2014, Traité IC2, série Signal et image. (hal-00968258)

[OS 3] Yajing Yan, V. Pinel, **Flavien Vernier**, Emmanuel Trouvé. Displacement measurements. Florence Tupin, Jordi Inglada, Jean-Marie Nicolas Eds. Remote Sensing Imagery, ISTE - John Wiley & Sons, pp. 251-282, 2014, Digital and image processing series, 978-1-84821-508-5. (hal-00968264)

Acte de conférence internationale avec comité de lecture

2019

[ACTI 1] Etienne Mauffret, **Flavien Vernier**, Sébastien Monnet. CANDoR : Consistency Aware Dynamic data Replication. IEEE International Symposium on Network Computing and Applications (NCA 2019), Sep 2019, Cambridge, United States. (hal-02274165)
(*Computer Science : acceptance rate 27,6%, ranking NA (2016 : 19247), H-index NA (2016 : 6), Impact-Factor NA (2016 : 1,09)*)

2018

[ACTI 2] Guilhem Marsy, Xavier Bodin, Emmanuel Malet, William Castaings, **Flavien Vernier**, et al. Monitoring rock glacier by optical stereoscopic time-lapse device. Edited by Philip Deline, Xavier Bodin and Ludovic Ravelin; Co-editors : Chloé Barboux; Reynald Delaloye; Christophe Lambiel; Florence Magnin; Marco Marcer; Paolo Pogliotti; Philippe Schoeneich. 5 th European Conference On Permafrost, Jun 2018, Chamonix, France. Edytem, 2018, 5th European Conference On Permafrost – Book of Abstracts. (hal-01887538)

(*Geoscience : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA*)

[ACTI 3] Ilham Alloui, Eric Benoit, Stéphane Perrin, **Flavien Vernier**. WIoT : Interconnection between Wise Objects and IoT. ICSOFT 2018, the 13th International Conference on Software Technologies, Jul 2018, Porto, Portugal. (hal-01801009)

(*Computer Science : acceptance rate 30-40%, ranking NA (2016 : 22712), H-index NA (2016 : 3) , Impact-Factor NA (2016 : 0,45)*)

2017

[ACTI 4] Ilham Alloui, **Flavien Vernier**. A Wise Object Framework for Distributed Intelligent Adaptive Systems. ICSOFT 2017, the 12th International Conference on Software Technologies, Jul 2017, Madrid, Spain. (hal-01617993)

(*Computer Science : acceptance rate 30-40%, ranking NA (2016 : 22712), H-index NA (2016 : 3) , Impact-Factor NA (2016 : 0,45)*)

[ACTI 5] Héli Hadhri, **Flavien Vernier**, Abdourrahmane Atto, Emmanuel Trouvé. Inverse Formulation of Temporal Closure and Proposed Solutions for Offset Tracking of Natural Scenes. MultiTemp 2017, Jun 2017, Bruges, Belgium. (hal-01618013)

(*Geoscience & Computer Science : acceptance rate NA, ranking NA (2015 : 17856), H-index NA (2015 : 3), Impact-Factor NA*)

2016

[ACTI 6] Yajing Yan, Amaury Dehecq, Emmanuel Trouvé, Gilles Mauris, Noel Gourmelen, **Flavien Vernier**. An overview to remotely sensed displacement measurements fusion : Current status and challenges. IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2016), Jul 2016, Beijing, China. (hal-01332555)

(*Geoscience & Computer Science : acceptance rate 42,1%, ranking 14743, H-index 58, Impact-Factor 0,51*)

[ACTI 7] Youen Pericault, Catherine Pothier, Nicolas Méger, Emmanuel Trouvé, **Flavien Vernier**, et al. Grouped frequent sequential patterns derived from terrestrial image time series to monitor landslide behaviour – Application to the dynamics of the Sanières/Roche Plombée rockslide. Geophysical Research Abstracts - EGU General Assembly 2016, Apr 2016, Vienna, Austria. pp.1-1. (hal-01306556)

(Geoscience : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

[ACTI 8] Haixing He, **Flavien Vernier**, Estelle Ployon, Thierry Villemin. 3D displacement retrieval on a scaled model of mountain slope by virtual multi-view photogrammetry. 2nd Virtual Geoscience Conference, 2016, Bergen, Norway. (hal-01880618)

(Geoscience : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

2015

[ACTI 9] Youen Pericault, Catherine Pothier, Nicolas Meger, Christophe Rigotti, **Flavien Vernier**, et al. A swap randomization approach for mining motion field time series over the Argentiere glacier. 2015 8th International Workshop on the Analysis of Multitemporal Remote Sensing Images (Multi-Temp), Jul 2015, Annecy, France. pp.1-4, (10.1109/Multi-Temp.2015.7245757). (hal-01237206)

(Geoscience & Computer Science : acceptance rate NA, ranking 17856, H-index 3, Impact-Factor NA)

[ACTI 10] Haixing He, **Flavien Vernier**, Thierry Villemin, Estelle Ployon, Philip Deline, et al. 3D displacement retrieval on glacial areas by airborne photogrammetry. 8th International Workshop on the Analysis of Multitemporal Remote Sensing Images, Jul 2015, Annecy, France. pp.1 - 4, (10.1109/Multi-Temp.2015.7245756). (hal-01229308)

(Geoscience & Computer Science : acceptance rate NA, ranking 17856, H-index 3, Impact-Factor NA)

[ACTI 11] Ilham Alloui, David Esale, **Flavien Vernier**. Wise Objects for Calm Technology. 10th International Conference on Software Engineering and Applications (ICSOFT-EA 2015), Jul 2015, Colmar, France. pp.468-471, (10.5220/0005560104680471). (hal-01226219)

(Computer Science : acceptance rate 30-40%, ranking 27187, H-index 4 , Impact-Factor NA)

2014

[ACTI 12] Matthieu Volat, **Flavien Vernier**, Marie-Pierre Doin, Cécile Lasserre, Emmanuel Trouvé, et al. Improving the execution of workflows for SAR image analysis. Conference on Big Data from Space, Nov 2014, Frascati, Italy. (hal-01487781)

(Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

2013

[ACTI 13] Yajing Yan, Laurent Ferro-Famil, Michel Gay, Renaud Fallourd, Emmanuel Trouvé, **Flavien Vernier**. Attempt of alpine glacier flow modeling based on correlation measurements of high resolution SAR images. IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2013), Jul 2013, Melbourne, Australia. pp.240-243, (10.1109/IGARSS.2013.6721136). (hal-00953770)

(Geoscience & Computer Science : acceptance rate NA, ranking 14743, H-index 58, Impact-Factor 0,51)

2012

[ACTI 14] **Flavien Vernier**, Renaud Fallourd, Jean-Michel Friedt, Yajing Yan, Emmanuel Trouvé, et al. Glacier flow monitoring by digital camera and space-borne SAR images. Image Processing Theory, Tools and Applications (IPTA 2012), Oct 2012, Turkey. pp.25 - 30, (10.1109/IPTA.2012.6469541). (hal-00936106)

(Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

[ACTI 15] Jean-Marie Nicolas, Emmanuel Trouvé, Renaud Fallourd, **Flavien Vernier**, Florence Tupin, et al. A first comparison of Cosmo-Skymed and TerraSAR-X data over Chamonix Mont-Blanc test-site. IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2012), Jul 2012, Munich, Germany. pp.4141. (hal-00797185)

(Geoscience & Computer Science : acceptance rate 30,5%, ranking 14743, H-index 58, Impact-Factor 0,51)

[ACTI 16] Patrice Moreaux, Fabien Sartor, **Flavien Vernier**. An Effective Approach for Home Services Management. Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euro-micro International Conference on, Feb 2012, Garching, Germany. pp.47-51, (10.1109/PDP.2012.45). (hal-00994913)

(Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

2011

[ACTI 17] **Flavien Vernier**, Renaud Fallourd, Yajing Yan, Diana Rosu, Emmanuel Trouvé, et al. Combining Space-Borne SAR Data and Digital Camera Images to Monitor Glacier Flow by Remote and Proximal Sensing. IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2011), Jul 2011, Vancouver, Canada. (hal-00621275)

(Geoscience & Computer Science : acceptance rate 67,3%, ranking 14743, H-index 58, Impact-Factor 0,51)

2010

[ACTI 18] Renaud Fallourd, **Flavien Vernier**, Jean-Michel Friedt, G. Martin, Emmanuel Trouvé, et al. Monitoring temperate glacier with high resolution automated digital cameras - Application to the Argentière glacier. Photogrammetric Computer Vision and Image Analysis (PCV 2010), ISPRS Commission III Symposium, Sep 2010, Paris, France. (halsde-00554849)

(Computer Science : acceptance rate 65%, ranking NA, H-index NA, Impact-Factor NA)

[ACTI 19] Renaud Fallourd, **Flavien Vernier**, Yajing Yan, Jean-Marie Nicolas, Andrea Walpersdorf, et al. Alpine glacier 3D displacement derived from ascending and descending TerraSAR-X images on Mont-Blanc test site. 8th European Conference on Synthetic Aperture Radar (EUSAR 2010), Jun 2010, Aachen, Germany. pp.n.c. (halsde-00554823)

(Signal Processing & Computer Science : acceptance rate NA, ranking 13143, H-index 17, Impact-Factor 0,71)

2009

[ACTI 20] Martin Quinson, **Flavien Vernier**. Byte-Range Asynchronous Locking in Distributed Settings. 17th Euromicro International Conference on Parallel, Distributed and network-based Processing - PDP 2009, Feb 2009, Weimar, Germany. (inria-00338189)

(Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

2007

[ACTI 21] Jacques M. Bahi, Raphaël Couturier, **Flavien Vernier**. Synchronous Distributed Load Balancing on Totally Dynamic Networks. IEEE. 21th International Parallel and Distributed Processing Symposium (IPDPS 2007), Proceedings, 26-30 March 2007, Long Beach, California, USA, Mar 2007, Long Beach, California, USA, United States. IEEE, pp.1-8, 2007. (hal-00547875)

(Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

2006

[ACTI 22] Emmanuel Jeannot, **Flavien Vernier**. A Practical Approach of Diffusion Load Balancing Algorithms. Euro-Par 2006, Aug 2006, Dresden, Germany. (inria-00097245)

(Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA)

2003

[ACTI 23] Jacques M. Bahi, Raphaël Couturier, **Flavien Vernier**. Accelerated Diffusion Algorithms on General Dynamic Networks. Parallel Processing and Applied Mathematics (PPAM'2003), Sep 2003, Czesochowa, Poland. pp.77–82. (hal-00621267)

(*Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA*)

[ACTI 24] Jacques M. Bahi, Raphaël Couturier, **Flavien Vernier**. Broken Edges and Dimension Exchange. Parallel, Distributed and Network based Processing (PDP), Feb 2003, Genova, Italy. pp.140–145. (hal-00621270)

(*Computer Science : acceptance rate NA, ranking NA, H-index NA, Impact-Factor NA*)

Acte de conférence francophone avec comité de lecture

2018

[ACTN 1] Yajing Yan, Emmanuel Trouvé, Matthias Jauvin, Rémi Prebet, Alexandre Hippert-Ferrer, Nicolas Meger, Abdourrahmane Atto, **Flavien Vernier**, et al. Utilisation des séries temporelles d'images Sentinel-1 pour l'observation des glaciers des Alpes. PNTS 2018, Mar 2018, Grenoble, France. (hal-01741918)

[ACTN 2] Guilhem Marsy, **Flavien Vernier**, Xavier Bodin, William Castaings, Emmanuel Trouvé. Détection automatique de zones en mouvement dans des séries d'images non recalées : application à la surveillance des mouvements gravitaires. Conférence française de photogrammétrie et télédétection, 2018, Marne la Vallée, France. (hal-01887597)

2017

[ACTN 3] Héra Hadhri, **Flavien Vernier**, Abdourrahmane Atto, Emmanuel Trouvé. Traitement automatique de "time lapse" : application à la surveillance de glaciers alpins. ORASIS 2017, GREYC, Jun 2017, Colleville-sur-Mer, France. (hal-01866629)

2014

[ACTN 4] Ha-Thai Pham, Haixing He, **Flavien Vernier**, Emmanuel Trouvé, Lionel Benoit, et al. Analyse de "Time-Lapse" stéréo pour la mesure de déformation 3D, application au suivi du glacier d'Argentière. Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014, Jun 2014, France. (hal-00988890)

Communications

2006

[COM 1] Emmanuel Jeannot, **Flavien Vernier**. A Practical Approach of Diffusion Load Balancing Algorithms. [Research Report] RR-5875, INRIA. 2006. (inria-00071394)

2002

[COM 2] Jacques M. Bahi, Sylvain Contassot-Vivier, Raphaël Couturier, **Flavien Vernier**. Asynchronisme et équilibrage de charge dans la grille de calcul. In École d'hiver GRID 2002. 2002, pp.365–373. (hal-00621271)

DEUXIÈME PARTIE : TRAVAUX DE RECHERCHE

sommaire

5	Chaînes de traitements de séries d'images	27
5.1	Contexte	28
5.2	Prétraitements	29
5.3	Mesure de déplacement	32
5.4	Approche conceptuelle des chaînes de traitements	46
5.5	Conclusion & réflexion	54
6	Architecture de traitement de l'habitude	55
6.1	Contexte	56
6.2	De la sagesse à l'intelligence	57
6.3	Simulation de comportements	62
6.4	Modélisation de l'habitude et détection de l'inhabituel	68
6.5	Applications	71
6.6	Perspectives	75

Introduction

Dans cette partie, je vais tenter de vous mener dans mon univers de recherche. Ou comment, à partir de travaux sur les chaînes de traitements d'images, je me suis orienté vers l'acquisition de connaissances au sein même des logiciels. Comment ai-je introduit le concept d'objets sages et de sagesse d'un logiciel – la connaissance qu'un logiciel peut acquérir sur lui-même – comme brique de base à son intelligence.

Le point de départ des travaux présentés ci-après s'attache aux traitements de séries temporelles d'images pour le suivi d'objets géophysiques et plus particulièrement les glaciers alpins. À l'origine, la question était de savoir s'il était possible de mesurer le déplacement d'un objet géophysique en utilisant des appareils photo classiques installés aux environs de l'objet en question, ces derniers fournissant les séries temporelles d'images. Les problématiques découlant des séries temporelles d'images proviennent notamment de la masse de données en raison du grand nombre d'images, de une tous les 6 jours en imagerie SAR satellitaire, à plusieurs par jours en imagerie optique, et ceci pouvant s'étendre sur plusieurs années. Mais la masse de données provient également de la taille des images, de plusieurs 10^{ne} de Mo pour chaque image optique terrestre, à plusieurs Go pour chaque image SAR satellitaire. Le traitement opérationnel de ces séries par des chaînes de traitements pose des problèmes d'optimisation, de répartition des traitements et des données, et de connaissances sur la chaîne elle-même. Ce dernier point est celui qui a été investigué dans mes travaux, notamment l'acquisition de connaissances sur les chaînes de traitements en particulier et sur le logiciel de manière plus générale. Ceci a nécessité, dans un premier temps, la mise en place de chaînes de traitements opérationnelles, ce qui est une problématique de recherche à part entière pour le traitement de séries temporelles d'images acquises en milieu naturel.

Au cours de ces travaux et l'encadrement de 7 thèses sur le sujet, nous avons investigué différentes étapes de ces chaînes de traitements. Nous nous sommes intéressés à la problématique du recalage d'images, problème induit naturellement par le système d'acquisition qui se dilate avec les effets thermiques en milieu naturel. Cette dilatation induit d'infimes mouvements du système qui sont perceptibles dans les images acquises. Nous nous sommes également intéressés à la sélection des images, outre l'effet thermique, les conditions climatiques peuvent perturber l'acquisition des images par des effets météorologiques (pluie, nuage, neige...) et rendre ces dernières inutilisables. L'objectif opérationnel étant le suivi des déplacements des objets observés, nous avons proposé plusieurs approches originales permettant de calculer, à partir des séries, le déplacement métrique 3D des objets. Le point commun et original de nos approches est d'utiliser au maximum la redondance d'information afin de minimiser les erreurs de calcul. Ceci implique l'explosion de la quantité de calcul et une problématique de traitement. Sur ce point, nous avons proposé une solution opérationnelle permettant, via une description du flux de traitements de la chaîne, de la distribuer de manière transparente sur une architecture multicœur et quasi transparente sur un cluster de calcul.

D'autres questionnements se sont ouverts sur l'exécution de ces chaînes de traitements. Notamment, comment acquérir de la connaissance sur leur fonctionnement afin de détecter, par exemple, des dysfonctionnements? Cette question, d'un point de vue logiciel, est plus large que simplement l'acquisition de connaissances sur des chaînes de traitements, ces dernières étant ni plus ni moins qu'une application distribuée, ou distribuable pour être précis, et par conséquent un cas particulier de logiciel. Afin d'étendre et généraliser la question précédente, je me suis intéressé à la problématique de comment, de manière générique, acquérir de la connaissance par apprentissage automatique non-supervisé (IA) sur une application elle-même, ceci sans en perturber son utilisation ou son développement. D'un

point de vue conceptuel, ce questionnement pose la problématique de la séparation des métiers et du découplage des tâches, un développeur métier n'est pas un développeur IA et vice versa. Le concept d'objets sages (WO) que j'ai introduit, tente de répondre à cette problématique en intégrant une couche de sagesse – la connaissance propre de l'objet sur lui-même – entre les couches métier et IA. Ce concept a ouvert un certain nombre de problématiques dérivées – modélisation conceptuelle des WO, modélisation et agrégation de leur connaissance, testabilité des WO... – dont certaines sont abordées dans ce mémoire et d'autres sont plus en perspectives et ouvertures.

La présentation de mon activité de recherche s'organise de la manière suivante. Le chapitre 5 présente les travaux sur les chaînes de traitements d'images, travaux réalisés au cours des thèses de Ha Thai Pham, Haixing He, Hella Hadhri et Guilhem Marsy. Pour cette présentation, nous nous focalisons uniquement sur les apports majeurs de ces travaux et prenons un point de vue plus numérique qu'applicatif. Le chapitre 6 s'attarde plus sur les travaux réalisés sur l'acquisition de la sagesse, travaux qui s'appuient sur ceux de la thèse d'Étienne Mauffret, notamment pour la modélisation de la mémoire et se poursuivront dans la thèse de Abdelhafid Dahhani et dont l'une des applications est celle de Fabien Sartor soutenue en 2012.

5 Chaînes de traitements de séries d'images

5.1	Contexte	28
5.2	Prétraitements	29
5.2.1	Sélection des images	29
5.2.2	Recalage des séries	30
5.3	Mesure de déplacement	32
5.3.1	Mesure par corrélation simple	32
5.3.2	Fermeture temporelle	36
5.3.3	Médian glissant sur de multiple « common/master » séries	38
5.3.4	Mesure en environnement virtuel	41
5.3.5	Rendu métrique	45
5.4	Approche conceptuelle des chaînes de traitements	46
5.5	Conclusion & réflexion	54

5.1 Contexte

L'acquisition de séries temporelles d'images stéréoscopiques en milieu naturel illustrée par la figure 1, présente un ensemble de problématiques qui vont de l'instrumentation [Poncet, 2016a, Poncet, 2016b] à l'analyse géophysique du phénomène observé [Bodin et al., 2018] en passant par le traitement des données [Eltner et al., 2016, Scaioni et al., 2018], traitements faisant part entière de la photogrammétrie [Hartley and Zisserman, 2003]. Le système d'acquisition présenté en figure 1 fournit, suivant une périodicité définie, une série d'images monoscopique ou stéréoscopique selon qu'un ou deux appareils photo sont installés, cette série est fréquemment nommée « time-lapse ». Le traitement de ces séries nécessite des chaînes opérationnelles dédiées aux caractéristiques de ces dernières et de leur mode d'acquisition. En effet, les conditions extérieures - climatiques - ont un impact à la fois sur le système d'acquisition et sur les acquisitions elles-mêmes. Notamment des effets thermiques sur le dispositif induisent de légers mouvements de ce dernier qui apparaissent comme des déplacements sur les acquisitions [Gance et al., 2014, Neyer, 2016, Roncella et al., 2014, Schwalbe and Maas, 2017]. De plus, certaines conditions climatiques rendent les images inutilisables.

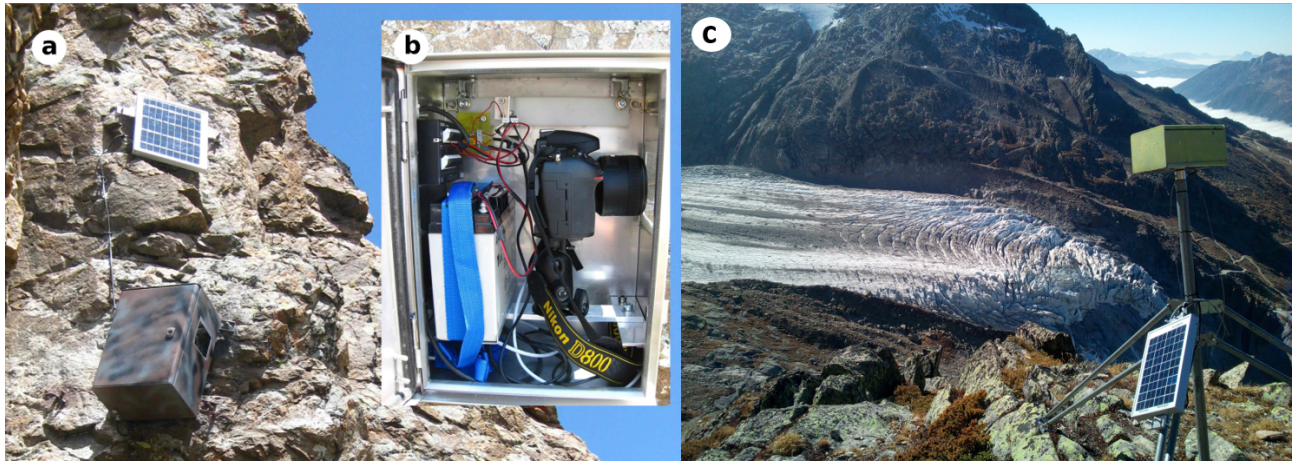


FIGURE 1 – Système d'acquisition photogrammétrique autonome in-situ. a – Installation en falaise sur le site du Laurichard. b – Vue interne du dispositif. c – Installation sur trépied sur le site d'Argentière.

Des prétraitements sont donc nécessaires afin : (i) de sélectionner, dans les séries d'images brutes, les images utilisables ; (ii) corriger les mouvements induits par les mouvements du système d'acquisition. Suite à ceci, les traitements principaux, le calcul de déplacements d'objets géophysiques dans notre cas, peut être opéré. Cette approche s'applique aussi bien au suivi des déplacements de surface de glaciers blancs [Maas et al., 2013, Lenzano et al., 2014, Benoit et al., 2015, Schwalbe and Maas, 2017], de glaciers rocheux [Neyer, 2016, Kenner et al., 2018] ou encore de glissements de terrain [Gance et al., 2014, Liu and Huang, 2016, Gabrieli et al., 2016, Desrues et al., 2019], et suit toujours la chaîne de traitements macroscopique présentée sur la figure 2.

Les sections 5.2 et 5.3 présentent les approches originales que nous avons proposées lors de différentes thèses, respectivement pour la résolution des problèmes de prétraitements et ceux de mesures de déplacements. La masse de traitement du fait de la quantité de données et la complexité numérique des traitements a donné lieu, pour chacune des thèses, à la réalisation de chaînes de traitements. Dans le cadre de travaux qui me sont propres, l'étude de la description de ces dernières et leur déploiement sur des machines ou des clusters de calcul sont présentés en section 5.4. La section 5.5 conclut ces

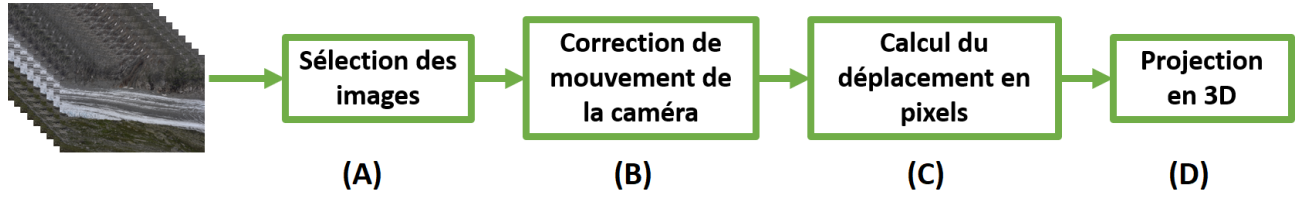


FIGURE 2 – Chaîne de traitements macroscopique pour le traitement de séries temporelles d'images acquises en milieu naturel.

travaux et introduit une réflexion sur ces chaînes de traitements et l'acquisition de connaissance sur leur exécution et plus particulièrement sur la problématique d'acquisition de la connaissance et sur l'utilisation de ces dernières. Les chaînes de traitements étant un logiciel particulier, cette réflexion s'ouvre donc également sur les logiciels de manière générale.

5.2 Prétraitements

5.2.1 Sélection des images

L'une des problématiques avec l'acquisition d'images en milieu naturel est la sélection des images valides pour le traitement. Les perturbations (nuage, pluie, neige...) peuvent rendre les images inutilisables. Une phase de sélection est donc indispensable. Cette dernière peut être manuelle, dans le cas de petite série d'images [Neyer, 2016] ou automatique ce qui nécessite la mise en place de critères appropriés.

Une première approche a été étudiée dans le cadre de la thèse d'Ha Thai Pham [Pham, 2015]. Sur les N images d'une série, prenons K zones que nous nommerons imagerie, sur des zones fixes autour de l'objet géophysique observé. Ces imagerie ont le rôle initial de point de recalage, elles servent à calculer le décalage entre chaque acquisition et définir les points de référence pour le recalage global de la série d'images, recalage que nous verrons par la suite. Les décalages entre chaque date d'une imagerie k sont calculés par une méthode de corrélation. Les valeurs de déplacement sont regroupées dans des matrices D_x^k et D_y^k correspondant respectivement aux décalages en colonnes et en lignes :

$$D_x^k = \begin{pmatrix} d_x^k(1,1) & \dots & d_x^k(1,N) \\ \vdots & \ddots & \vdots \\ d_x^k(N,1) & \dots & d_x^k(N,N) \end{pmatrix}, \quad D_y^k = \begin{pmatrix} d_y^k(1,1) & \dots & d_y^k(1,N) \\ \vdots & \ddots & \vdots \\ d_y^k(N,1) & \dots & d_y^k(N,N) \end{pmatrix}. \quad (1)$$

Les valeurs $d_x^k(n, m)$ et $d_y^k(n, m)$ représentent, pour l'imagerie k , les décalages respectifs en colonne et en ligne de l'image m par rapport à l'image n . Les matrices D_x^k et D_y^k étant théoriquement antisymétriques, une estimation des erreurs pour chaque imagerie k dans chaque image n est donnée par :

$$Err_n^k = \sum_m (d_x^k(n, m) + d_x^k(m, n))^2 + \sum_m (d_y^k(n, m) + d_y^k(m, n))^2.$$

Ces valeurs permettent d'écarter certaines imagerie sur certaines images, voire des images entières qui ne seraient pas utilisables en raison notamment de conditions météorologiques. Une erreur de recalage

pour une image n est alors donnée par :

$$Err_n = \sum_k Err_n^k.$$

La mise à l'écart d'images ou d'images entières nécessite naturellement la détermination d'un seuil d'erreurs défini dans notre cas de manière empirique.

Dans le cadre de la thèse d'Héla Hadhri [Hadhri, 2020], nous avons investigué une autre approche inspirée des travaux [Chu et al., 2017], basée sur la définition de descripteurs permettant l'évaluation et la classification de la visibilité du glacier. Pour ce faire, nous proposons deux descripteurs très simples :

- un descripteur τ_G sensible aux textures du glacier,
- un descripteur τ_S quantifiant les pixels blancs.

Ces indices sont ensuite utilisés conjointement afin d'effectuer la classification binaire : « glacier visible/glacier non visible ». τ_G est construit à partir d'un filtre de Gabor [Jain et al., 1997] comme étant la somme de l'énergie spatiale obtenue :

$$\tau_G = \sum_{xy} (I(x, y) \otimes G_{S_\Delta}).$$

Il nous donne ainsi un niveau de visibilité de la structure apparente sur l'image. Dans notre cas, le masque est choisi pour être en cohérence avec la structure du glacier. Certaines textures typiques des glaciers – crevasse, ligne de force – possèdent une périodicité grossièrement stationnaire qui peut rester marquée dans toutes conditions climatiques, même avec un couvert neigeux important, s'il est attaché aux crevasses. Par exemple, un masque dédié au glacier d'Argentière – suivant ses lignes de forces et crevasses – a été choisi. Il est ainsi discriminant pour quantifier la visibilité du glacier. Le second indice de visibilité τ_S est construit afin de discriminer les images où les crevasses sont visibles, mais où le glacier est enfui sous une épaisse couche de neige. En effet, celles-ci introduisent un biais important dans les mesures dû à l'absence de texture. La méthode la plus classique et la moins coûteuse reste la quantification des pixels de couleur claire sur une zone de la scène qui n'est recouverte que lors d'épisodes neigeux importants. Un cadre de délimitation (bounding box) de taille 10x10 pixels s'est empiriquement montré suffisant afin de détecter ces images et les écarter des futurs traitements.

D'autres approches comme la quantité de gradients ou celle de points d'intérêts (SIFT, SURF, ORB...) ont également été étudiées afin de discriminer les images utilisables de celles perturbées par un effet météorologique. L'ensemble de ses méthodes sont efficaces, mais nécessitent toutes, pour l'heure, la définition d'un seuil déterminé de manière empirique.

5.2.2 Recalage des séries

Outre la sélection des images, la seconde problématique du traitement d'images acquises en milieu naturel est le recalage de ces dernières. Même acquises depuis un support fixe, encré sur une zone immobile, les changements atmosphériques – température, hygrométrie, ensoleillement... – ont un effet sur le système d'acquisition, notamment des effets de dilatation/contraction thermiques qui génèrent des mouvements au sein des images acquises. Une phase de recalage est donc indispensable.

En nous basant sur une approche « common/master », comme proposée dans la thèse d'Ha Thai Pham [Pham et al., 2014] – c'est-à-dire recalcr l'ensemble des images sur une seule image de référence – il convient dans un premier temps de définir l'image de référence. En nous appuyant sur la définition des matrices de distances 1, pour chaque image k , les distances entre les couples d'images (n, m)

sont alors calculées :

$$d^k(n, m) = \sqrt{d_x^k(n, m)^2 + d_y^k(n, m)^2}.$$

On peut ainsi chercher l'image « médiane » qui minimise la somme des distances aux autres images :

$$d_{min}^k = \min_n \left(\sum_m d^k(n, m) \right), \quad (2)$$

$$n_{ref}^k = \arg \min_n \left(\sum_m d^k(n, m) \right),$$

et par conséquent l'erreur de recalage. Les distances $d^k(n, m)$ dépendant de l'imagette k , l'image de référence recherchée n'est donc pas forcément la même selon l'imagette utilisée. Le choix va donc se porter sur l'image n_{ref} qui apparaît le plus fréquemment comme image de référence :

$$n_{ref} = \text{mode}_k \left(n_{ref}^k \right),$$

où mode est la valeur modale du vecteur constitué des n_{ref}^k , c'est-à-dire l'indice de l'image qui apparaît le plus souvent comme référence. En cas d'égalité, s'il y a plusieurs images de référence possibles, les valeurs données par l'équation 2 seront discriminantes. Ces résultats donnent également pour chaque image, les points d'intérêts relatifs à l'image de référence n_{ref} . Soit (x^k, y^k) le point au centre de l'imagette k . Les positions des points d'intérêts sur l'image m sont alors données par la matrice P^m :

$$P^m = \begin{pmatrix} x^1 + d_x^1(n_{ref}, m), & y^1 + d_y^1(n_{ref}, m) \\ \vdots & \vdots \\ x^k + d_x^k(n_{ref}, m), & y^k + d_y^k(n_{ref}, m) \\ \vdots & \vdots \\ x^K + d_x^K(n_{ref}, m), & y^K + d_y^K(n_{ref}, m) \end{pmatrix}$$

Ces points d'intérêts sont ensuite utilisés pour la détermination d'une transformation de recalage.

Cette approche, reposant sur le recalage par rapport à une image de référence unique, pose problème lors de traitements de longues séries plurisaisonnières. En effet, avec le temps la surface observée change et il n'est plus possible de retrouver les zones similaires avec une portée temporelle trop grande. La figure 5, sur laquelle nous reviendrons, illustre ce phénomène. Ainsi dans le cadre des travaux de thèse d'Hela Hadhri, une approche s'appuyant sur plusieurs images de référence au fil du temps de la série a été mise en place. Nous avons ainsi opté pour la démarche suivante :

1. le time-lapse est découpé par saison,
2. l'image ayant le plus fort indice de visibilité du glacier τ_G est choisie comme image de référence pour recaler la 1^{re} saison,
3. la dernière image de la saison recalée est ajoutée à la saison suivante et définie comme image de référence.

Concernant les méthodes de recalage que nous avons déployé, elles sont classiques et vont de la simple estimation d'une transformation de type rotation/translation dans le cas de séries d'images de faible résolution (3 700 x 2 500 pixel) à la détermination d'homographies – transformation à 8 degrés de liberté incluant rotation, translation, zoom et projection – de manière itérative afin de minimiser l'erreur de recalage en zone immobile pour les séries d'images acquises en très haute résolution (7 300 x 4 900 pixel) ; la résolution en pixel étant naturellement donnée avec une focale équivalente.

5.3 Mesure de déplacement

La mesure de déplacement est au cœur de ces traitements. De manière générale le problème consiste à déterminer, entre 2 images, le maximum de similarité $D_{k,l}(\hat{p}, \hat{q})$:

$$D_{k,l}(\hat{p}, \hat{q}) = \max_{p,q} (D_{k,l}(p, q)). \quad (3)$$

où (k, l) est un pixel dans une image et $(k + p, l + q)$ un pixel dans l'autre image. Les approches par corrélation simple [Benoit et al., 2015, Gabrieli et al., 2016] ou flot optique [Lucas and Kanade, 1981, Weinzaepfel et al., 2013], avec une stratégie « Common/Master » ou « LeapFrog » donnent des résultats cohérents, et ont été utilisées dans la cadre de la thèse de Ha Thai Pham [Pham, 2015]. Cependant, elles montrent leurs limites pour des portées temporelles faibles où l'objet se déplace relativement peu. En ordre de grandeur en milieu naturel, lorsque le déplacement est proche du 10^e de pixel, il se noie dans le bruit de mesure. Bruit notamment dû aux erreurs de recalage et d'appariement entre les images. Afin de limiter l'effet de ces erreurs, nous avons abordé deux approches différentes dans le cadre des thèses d'Héla Hadhri et Guilhem Marsy. Ces deux approches se basent sur le principe de la redondance de mesures temporelles ou le fait que tout déplacement $d_{i,j}$ est la somme des déplacements $d_{i,k_1} + d_{k_1,k_2} + \dots + d_{k_n,j}$.

5.3.1 Mesure par corrélation simple

La première méthode que nous avons mis en œuvre lors de la thèse de Ha Thai Pham [Pham, 2015] utilise comme fonction de similarité, celle de corrélation normalisée. Le fonctionnement de cette algorithme est illustré par la figure 3. L'une des images est définie comme image maître I et la seconde comme image esclave I' . Pour un pixel (k, l) donné, une fenêtre maître M est définie autour de ce dernier ainsi qu'une fenêtre de recherche plus grande S . Pour tout déplacement possible (p, q) de la fenêtre maître, dans la fenêtre de recherche, la fonction de corrélation croisée normalisée (NCC) donnée par l'équation 4 est calculée.

$$D_{k,l}(p, q) = \frac{\sum_{i,j} P_{k,l}(p, q, i, j)}{N_{k,l}(p, q)}, \quad (4)$$

où

$$P_{k,l}(p, q, i, j) = M(i, j) \cdot S(i - p, j - q), \quad (5)$$

et

$$N_{k,l}(p, q) = \sqrt{\sum_{i,j} M(i, j)^2 \cdot \sum_{i,j} S(i - p, j - q)^2}, \quad (6)$$

les indices (i, j) balayant la fenêtre maître affectés du décalage (p, q) sur l'image esclave. Au final, la valeur maximale des $D_{k,l}(p, q)$ donne le maximum de similarité entre un pixel $M(k, l)$ dans l'image maître, représenté par une fenêtre englobante – la fenêtre maître – et un pixel $S(k + p, l + q)$ dans l'image esclave, représenté également par la fenêtre maître. Ainsi, le déplacement de (k, l) entre M et S est donné par (p, q) .

Dans une implémentation naïve le calcul de NCC est très chronophage. Pour une taille de fenêtre maître $M_c \times M_r$, l'équation 6 requière $2(M_c \times M_r) + 2$ opérations en double précision, l'équation 5 une opération et les combinant pour obtenir une valeur de similitude, équation 4, $3(M_c \times M_r) + 3$. Afin de calculer toutes ces valeurs pour toutes les positions possibles dans une fenêtre de recherche de taille $S_c \times S_r$, nous obtenons $((S_c - M_c) \times (S_r - M_r)) \times (3(M_c \times M_r) + 3)$ opérations; ceci étant à répéter

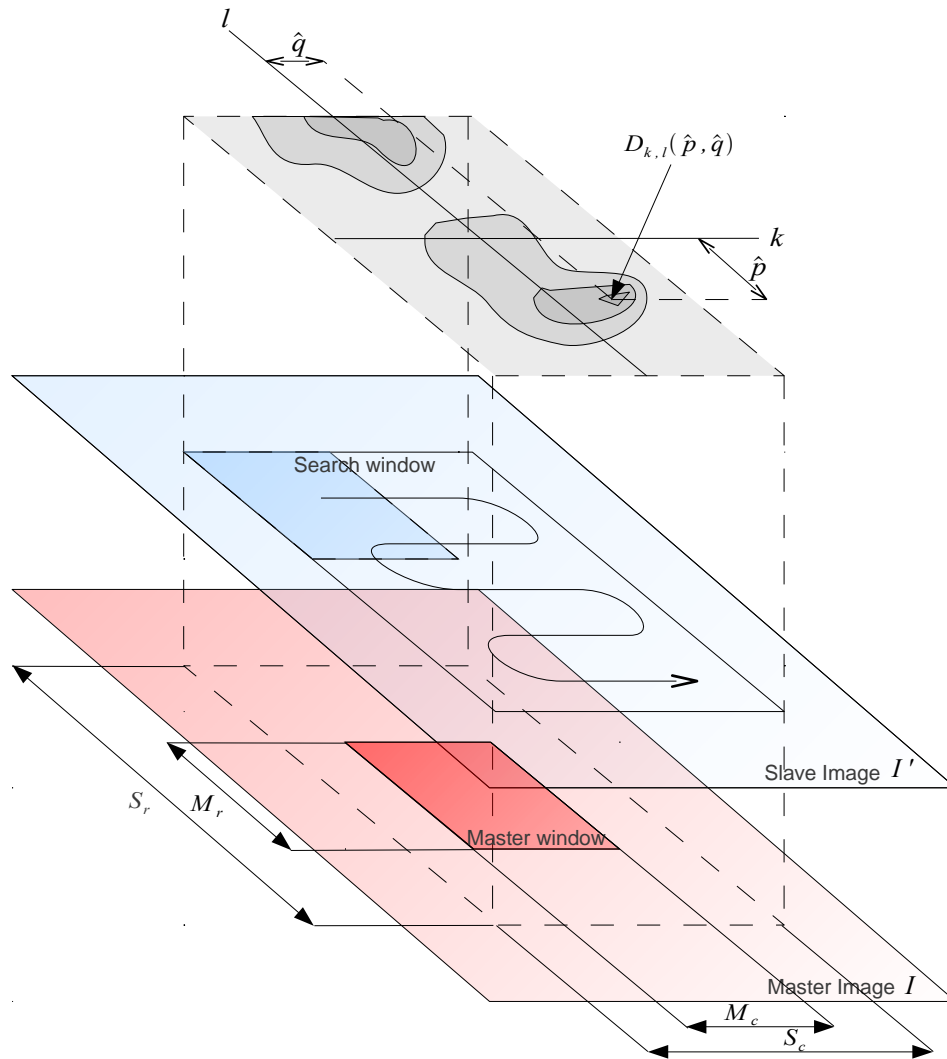


FIGURE 3 – Illustration schématique du fonctionnement de l'algorithme de corrélation.

pour tout pixel de l'image. Pour avoir une idée du coût de ce traitement, selon les données disponibles, leur type (image optique ou SAR), l'ampleur du déplacement recherché, les temps de calcul varient de l'ordre du jour à celui du mois.

Afin de réduire ce temps, j'ai entrepris l'étude de l'optimisation, la parallélisation et la distribution de ce calcul. Ce travail s'est effectué en 3 étapes d'optimisations successives, chaque version, naïve comprise, a permis de valider la suivante. Ci-après est présenté le dernier niveau d'optimisation. L'optimisation consiste en la réécriture du problème afin de mettre en évidence les dépendances et indépendances de calcul. Notamment, l'équation 4 peut se réécrire sous la forme :

$$D_{k,l}(p, q) = \frac{U_{k,l}(p, q)}{\sqrt{V_{k,l} \times W_{k,l}(p, q)}},$$

où U , V et W sont respectivement calculés de manière indépendante telle que :

$$\begin{aligned} U_{k,l}(p, q) &= \sum_{i,j} S(i, j) \cdot R(i - p, j - q), \\ V_{k,l} &= \sum_{i,j} S(i, j)^2, \\ W_{k,l}(p, q) &= \sum_{i,j} R(i - p, j - q)^2. \end{aligned}$$

Dans l'ensemble du processus, chaque calcul décalé d'un pixel est fortement dépendant du calcul d'un pixel adjacent. Ainsi, selon le pixel précédent sur la même colonne, $U_{k,l}(p, q)$ peut s'exprimer de la manière suivante :

$$\begin{aligned} U_{k,l}(p, q) &= U_{k,l-1}(p, q) \\ &\quad - \sum_i S(i, j_0 - 1) \cdot R(i - p, j_0 - 1 - q) \\ &\quad + \sum_i S(i, j_n) \cdot R(i - p, j_n - q), \end{aligned} \quad (7)$$

si $l \neq l_0$ ou selon le pixel précédent sur la même ligne si $k \neq k_0$:

$$\begin{aligned} U_{k,l}(p, q) &= U_{k-1,l}(p, q) \\ &\quad - \sum_j S(i_0 - 1, j) \cdot R(i_0 - 1 - p, j - q) \\ &\quad + \sum_j S(i_n, j) \cdot R(i_n - p, j - q). \end{aligned} \quad (8)$$

De la même manière, $V_{k,l}$ et $W_{k,l}(p, q)$ sont donnés par :

$$\begin{aligned} V_{k,l} &= V_{k,l-1} \\ &\quad - \sum_i S(i, j_0 - 1)^2 \\ &\quad + \sum_i S(i, j_n)^2, \end{aligned} \quad (9)$$

$$\begin{aligned} W_{k,l}(p, q) &= W_{k,l-1}(p, q) \\ &\quad - \sum_i R(i - p, j_0 - 1 - q)^2 \\ &\quad + \sum_i R(i - p, j_n - q)^2, \end{aligned} \quad (10)$$

si $l \neq l_0$, ou par :

$$\begin{aligned} V_{k,l} &= V_{k-1,l} \\ &\quad - \sum_j S(i_0 - 1, j)^2 \\ &\quad + \sum_j S(i_n, j)^2, \end{aligned} \quad (11)$$

$$\begin{aligned} W_{k,l}(p, q) &= W_{k-1,l}(p, q) \\ &\quad - \sum_j R(i_0 - 1 - p, j - q)^2 \\ &\quad + \sum_j R(i_n - p, j - q)^2, \end{aligned} \quad (12)$$

si $k \neq k_0$. Ainsi, pour tout pixel différent de (k_0, l_0) et en supposant des fenêtres carrées ($M_c = M_r$ et $S_c = S_r$), $U_{k,l}(p, q)$ requière $4 \times M_c + 2$ opérations, $V_{k,l}$ $3 \times M_c + 2$ opérations et $W_{k,l}(p, q)$ également

$3 \times M_c + 2$ opérations. On notera que les valeurs élevées au carré ne le sont faites qu'une fois et mémorisées pour leurs utilisations futures. Le calcul de $D_{k,l}(p, q)$ ne prend plus que $10 \times M_c$ en ordre de grandeur contre $3 \times M_c^2$ pour la version naïve.

Cette représentation optimisée du problème s'implémente naturellement sous la forme de matrices glissantes, une par valeur (U , V , W , S^2 et R^2), soit 5 matrices. De manière schématique, une matrice glissante fonctionne de la manière suivante. Pour une matrice 3×3 , elle est initialisée sous forme d'un vecteur de dimension 3×3 plus le nombre de décalages prévus.

11	12	13	21	22	23	31	32	33	-	-	-
----	----	----	----	----	----	----	----	----	---	---	---

Puis le vecteur est mappé sur sa forme matricielle afin d'être utilisé :

11	12	13
21	22	23
31	32	33

Le décalage consiste en un remappage des débuts de lignes décalés de 1 cellule :

11	12	13	21
	22	23	31
	32	33	-

Il convient pour finir le décalage, de recalculer uniquement la dernière colonne :

11	12	13	14
	22	23	24
	32	33	34

D'un point de vue implémentation, les décalages ne se font que le long des lignes, et le long de la première colonne. Cette dernière fonctionnant via une copie mémoire afin de redémarrer une ligne en fonction de la première colonne de la ligne précédente. Les problèmes de traitements d'images sont par nature facilement distribuables. Une découpe de l'image par blocs de lignes et le traitement de ces blocs sur des machines distinctes s'effectue assez facilement en prenant naturellement en compte le recouvrement des traitements et la fusion des résultats. Une chaîne de traitements selon ces trois étapes – découpe, traitement et fusion – a été développée. Une approche parallèle – multi-thread – compatible avec celle distribuée a également été implémentée, cependant elle n'apporte pas plus de gain que celle distribuée considérant chaque coeur d'une machine comme une machine distincte.

Rappelons que le problème ne se limite pas à 2 images, mais à une série de N images où N peut varier de quelques images à plusieurs centaines. Ce qui explique d'autant plus les temps de traitements qui, in fine, sera une contrainte pour N . Pour effectuer ce traitement sur la série, il est possible de prendre, par exemple, toujours la première image de la série comme image maître, nous obtenons ainsi une approche « Common/Master », et tous les déplacements seront relatifs à l'image maître. Ou alors, l'image maître définie par NCC peut être glissante et l'image esclave décalée de n images, nous serons ainsi dans une approche « LeapFrog ». Ces 2 approches sont connues pour être sensibles au bruit. De plus, l'approche « Common/Master » a une limite temporelle, pour un trop grand décalage entre l'image maître et l'image esclave, la mesure de déplacement n'est plus cohérente. Et concernant l'approche « LeapFrog », elle est connue pour cumuler les erreurs. Nous avons donc poursuivi ces travaux en cherchant à pallier ces problèmes.

5.3.2 Fermeture temporelle

La première approche utilisant la redondance de mesure, basée sur la méthode SBAS [Berardino et al., 2002], consiste à minimiser les erreurs de fermeture temporelle. Pour ce faire, les approches « Common/Master » ou « LeapFrog » sont envisageables [Hadhri et al., 2019]. Avec $\mathbf{F}_{i,j}$ le déplacement calculé entre les instants i et j et ϵ l'erreur de fermeture temporelle.

◦ les contraintes de type « Common/Master » sont :

$$\begin{aligned} \forall a \in \{1..n\}, \forall i \in \{a..n\}, \forall \Delta t \in \{1..n-i\} : \\ \mathbf{F}_{a,i+\Delta t} = \mathbf{F}_{a,i} + \mathbf{F}_{i,i+\Delta t} + \epsilon_{\text{CM}}, \end{aligned} \quad (13)$$

le déplacement entre une image maître a et toute autre image $i + \Delta t$ est la somme du déplacement entre a et i , et du déplacement entre i et $i + \Delta t$,

◦ les contraintes de type « LeapFrog » sont :

$$\begin{aligned} \forall a \in \{1..n\}, \forall \Delta t \in \{1..n-a\} \text{ et } \forall ds \in \{1..a-1\} : \\ \mathbf{F}_{a,a+\Delta t} = \sum_{i=a}^{a+\Delta t-1} \mathbf{F}_{i,i+1} + \epsilon_{\text{LF}}, \\ \mathbf{F}_{a,a-\Delta t} + \sum_{i=a-1}^1 \mathbf{F}_{i,i+1} + \epsilon_{\text{LF}}, \end{aligned} \quad (14)$$

le déplacement entre a et $a + \Delta t$ est la somme des déplacements successifs entre ces 2 dates. Ces contraintes sont particulièrement intéressantes, car elles sont modélisables via des applications linéaires. En effet, considérant une série de déplacements de N dates, elles peuvent être formulées sous la forme matricielle suivante :

$$\begin{pmatrix} F_{0,1} \\ F_{0,2} \\ \vdots \\ F_{0,N} \\ F_{1,0} \\ F_{1,2} \\ \vdots \\ F_{N,N-2} \\ F_{N,N-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 1 \\ -1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 & -1 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} F_{0,1} \\ F_{0,2} \\ F_{0,3} \\ \vdots \\ F_{0,N-2} \\ F_{0,N-1} \\ F_{0,N} \end{pmatrix} \quad (15)$$

pour « Common/Master », et

$$\begin{pmatrix} F_{0,1} \\ F_{0,2} \\ \vdots \\ F_{0,N} \\ F_{1,0} \\ F_{1,2} \\ \vdots \\ F_{N,N-2} \\ F_{N,N-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 1 & 1 & \dots & 1 & 1 \\ -1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & -1 & -1 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & -1 \end{pmatrix} \begin{pmatrix} F_{0,1} \\ F_{1,2} \\ F_{2,3} \\ \vdots \\ F_{N-2,N-1} \\ F_{N-1,N} \end{pmatrix}, \quad (16)$$

pour « LeapFrog ». D'une manière générale, en l'absence de données manquantes, les matrices des équations 15 et 16 obtenues seront surdéterminées, car le nombre de contraintes dépasse la dimension. Dans ce contexte, il est d'usage de minimiser la norme $L = 2$ de l'erreur d'ajustement au modèle via la pseudo-inverse analytique ou encore une décomposition en valeurs singulières de la matrice ternaire.

Soit un time-lapse formé de $N = 19$ images sur lequel nous voulons générer des séries temporelles de suivi constituées de 18 estimations de vitesses (modèle « LeapFrog ») ou de positions relatives (modèle « Common/Master ») en tous pixels. Avec cette série nous obtiendrons donc 19×18 (342) mesures redondantes afin d'estimer les vitesses ou positions. Le graphe de la figure 4 illustre l'ensemble de ses mesures qui serviront à minimiser les erreurs d'estimation.

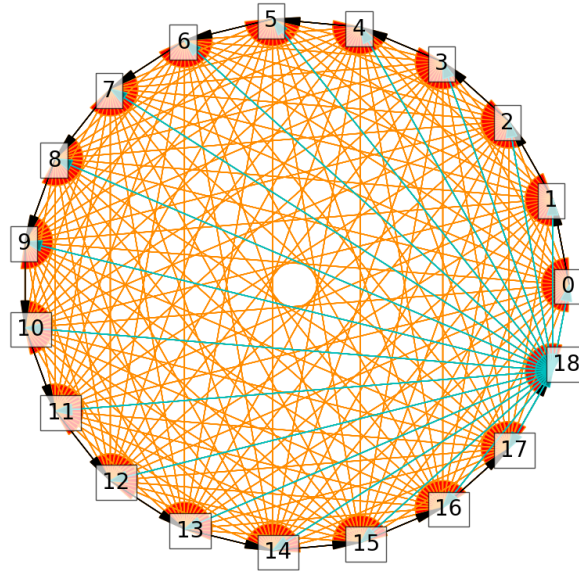


FIGURE 4 – Graphe d'indices pour un time-lapse à 19 dates. 342 flèches orange (et vertes) à double sens désignent les mesures, 18 flèches noires désignent les estimations de vitesse recherchées et 18 flèches cyan, celles des positions recherchées.

La prise en compte d'une portée temporelle infinie n'est pas réaliste. Au-delà d'un certain seuil, dépendant de l'objet observé, il n'est plus possible de détecter de points homologues sur l'objet. Ceci est notamment dû au changement de surface de l'objet au cours de son déplacement : rotation de l'objet, fonte de glace dans le cas de glaciers... La figure 5 présente l'indice de corrélation en fonction de la portée temporelle dans le cas d'observation acquise sur le glacier d'Argentière. Chaque courbe représente, pour une image, l'indice de corrélation en fonction de toutes les autres images. En abscisse 1 se superpose toutes valeurs de corrélation entre les images i et $i + 1$, en 2 celles entre i et $i + 2$, la courbe verte en donne les moyennes. Cette figure met ainsi en évidence l'effet de décorrélation temporelle. Dans le cas particulier de ce time-lapse, nous avons limité de manière empirique, la portée temporelle à plus ou moins 5 jours. Les équations 15 et 16 sont naturellement mises à jour en conséquence – en limitant les $F_{i,j}$ à $|i - j| \leq 5$ –, elles se retrouvent plus éparées qu'avec une portée infinie, mais restent surcontraintes.

Cette portée temporelle est également contrainte par les suites d'observations manquantes. En effet, certaines conditions climatiques – brume, neige, pluie... – empêchent l'acquisition d'images uti-

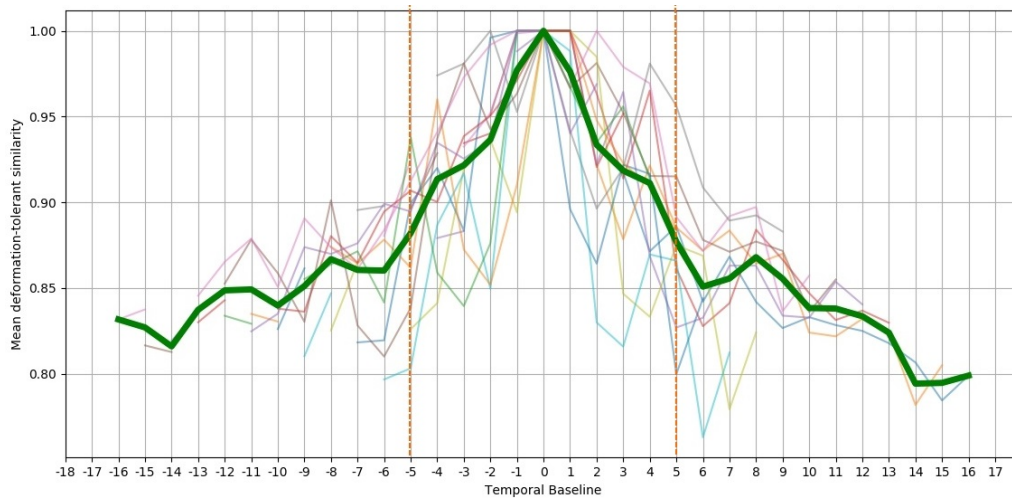


FIGURE 5 – Mise en évidence de la décorrélation des zones fixes des images avec la portée temporelle en jours (temporal baseline). L'indice de corrélation normalisée en ordonnée provient d'une mise en correspondance avec DeepMatching. En vert : moyenne sur toutes les portées temporelles avec 1 image du glacier d'Argentière par jour, sur 17 jours.

lisables. Ces images auront été écartées lors des prétraitements de sélection d'images. Si la portée temporelle choisie est inférieure à la plus grande suite sans donnée d'acquisition, le système est traité comme 2 time-lapses indépendants, puisqu'il ne peut pas les relier entre eux. Avec une portée temporelle assez grande, le système est en capacité de reconstruire les données manquantes. Cependant, ces reconstructions n'ont pas de sens physique si le système n'est pas surcontraint par un modèle de comportement de l'objet observé.

Outre le fait qu'une portée temporelle limite l'injection d'erreur dans le système et garantisse la continuité de mesure en cas de données manquantes, elle limite également la quantité de traitements à effectuer. En reprenant l'exemple pour un time-lapse de 19 images, en considérant une portée temporelle de plus ou moins 5 images, la quantité de mesure passe de 342 mesures à calculer à 190. Sachant que ces mesures sont toujours la tâche la plus chronophage des traitements, la détermination de la portée temporelle est cruciale afin d'optimiser la redondance pour minimiser l'erreur et minimiser le temps de calcul.

5.3.3 Médian glissant sur de multiple « common/master » séries

Nous avons proposé, dans le cadre de la thèse de Guilhem Marsy [Marsy, 2020] une autre approche pour l'utilisation de la redondance temporelle. L'approche se divise en 2 étapes. La première est le calcul des déplacements sur de petites portées temporelles : concrètement, sur des laps de temps où les changements de surface ne sont pas trop grands et permettent un calcul fiable des déplacements [Marsy et al., 2018a] [Marsy et al., 2018b] [Marsy et al., 2020b]. La seconde étape traite du calcul des déplacements à plus grandes portées temporelles, lorsque les déplacements ne peuvent plus être directement calculés entre deux images à cause de changements de surface trop importants [Marsy et al., 2020a] [Marsy et al., 2020c]. Nous avons proposé la méthode MMCMS pour résoudre la première étape et S-MMCMS pour la seconde.

Faible portée temporelle Considérons tous les déplacements relatifs entre toutes les images de la série d'images. La portée temporelle étant considérée assez courte pour que toute information de déplacement soit pertinente. Nous définissons une série temporelle de déplacements relatifs (RDTS pour Relative Displacement Time Series) comme étant les déplacements entre toutes les dates $j \in [0 : N[$ et une date de référence $i \in [0 : N[$ (approche « common/master »). Notons $\vec{d}_{i,j}$ le déplacement entre les dates i et j . Ainsi, RDTS_i représente le RDTS relatif à l'image i , et est défini de la manière suivante :

$$\text{RDTS}_i = \left\{ \vec{d}_{i,0}, \dots, \vec{d}_{i,j}, \dots, \vec{d}_{i,N-1} \right\}$$

Nous pouvons remarquer que, sans bruit ni erreur, la différence entre 2 RDTS relatifs aux dates i et j est donné par :

$$\text{RDTS}_j - \text{RDTS}_i = \vec{d}_{j,k} - \vec{d}_{i,k} = \vec{d}_{i,j} \quad \forall k \in [0 : N[$$

Tenant compte de cette remarque et du fait que le système ne soit pas sans bruit ni erreur, nous pouvons estimer le décalage $\vec{d}_{i,j}$ entre 2 RDTS par la moyenne entre les déplacements en chaque date :

$$\vec{d}_{i,j} = \frac{\sum_{k \in N} \vec{d}_{k,j} - \vec{d}_{k,i}}{N}. \quad (17)$$

Connaissant le décalage \vec{d} entre chaque paire de séries, il est possible d'en choisir une de référence i , recalculer toutes les j autres sur celle-ci via les \vec{d} et de décaler l'ensemble des résultats afin que la 1^{re} date soit la date de référence comme illustrée par la figure 6. Nous obtenons ainsi D_{r_i} tel que :

$$D_{r_i} = \begin{Bmatrix} \vec{d}_{r_0}^1 & \vec{d}_{r_1}^1 & \dots & \vec{d}_{r_N}^1 \\ \vec{d}_{r_0}^2 & \vec{d}_{r_1}^2 & \dots & \vec{d}_{r_N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \vec{d}_{r_0}^N & \vec{d}_{r_1}^N & \dots & \vec{d}_{r_N}^N \end{Bmatrix},$$

où

$$\vec{d}_{r_j}^k = \vec{d}_{r_j}^{0,k} = \vec{d}_{j,k} - \vec{d}_{j,0} - \vec{\alpha}_{i,j}.$$

Quant à la détermination de la référence i , nous choisissons le D_{r_i} qui minimise l'erreur :

$$\text{error} = \frac{\sum_{i \in N} \sum_{j \in N} \|\vec{d}_{ref}^i - \vec{d}_{r_j}^i\|}{N^2}. \quad (18)$$

Par conséquent, tous les D_{r_i} sont calculés afin de déterminer :

$$D_{ref} = \begin{Bmatrix} \vec{d}_{r_0}^1 & \vec{d}_{r_1}^1 & \dots & \vec{d}_{r_N}^1 \\ \vec{d}_{r_0}^2 & \vec{d}_{r_1}^2 & \dots & \vec{d}_{r_N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \vec{d}_{r_0}^N & \vec{d}_{r_1}^N & \dots & \vec{d}_{r_N}^N \end{Bmatrix},$$

À partir du moment où la série de référence est sélectionnée, un RDTS robuste peut être calculé. Pour ce faire, la position médiane et l'écart absolu médian (MAD) sont calculés pour chaque date. À

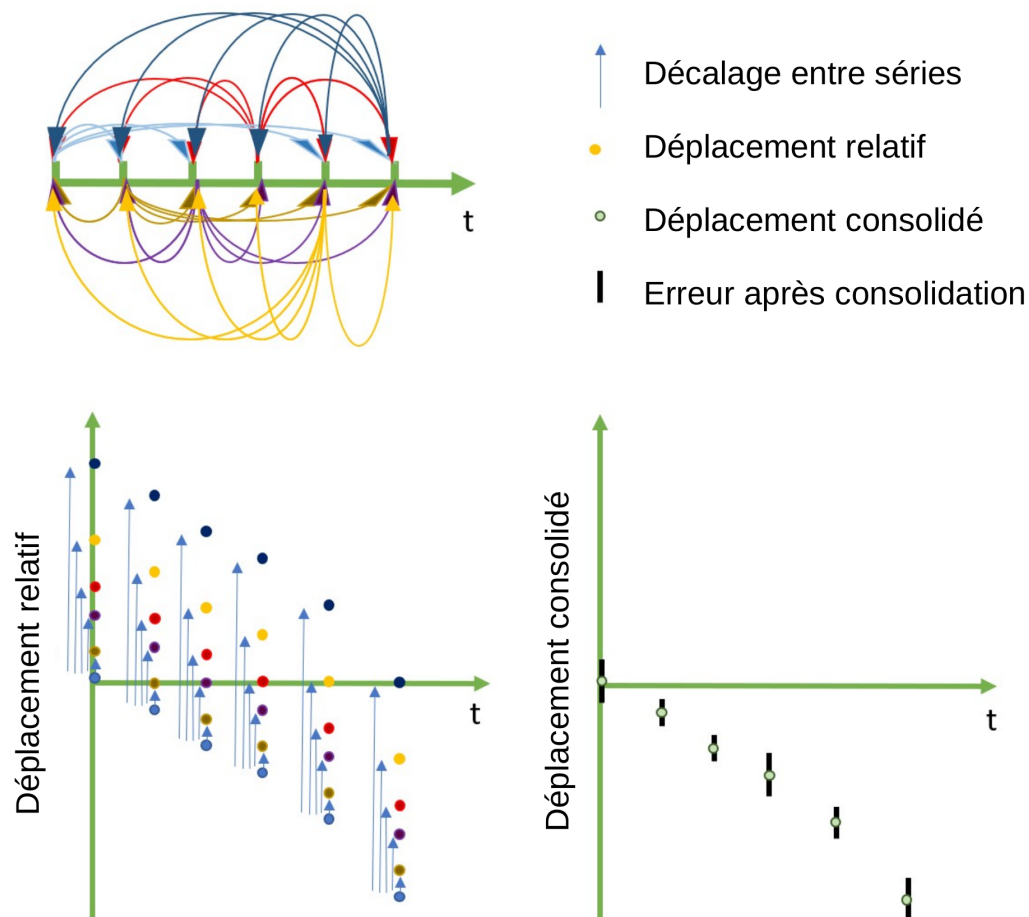


FIGURE 6 – Principe de consolidation des séries temporelles de déplacements. En haut gauche est illustré les N séries « common/master » (une couleur pour chaque série i) possibles avec une série de 6 images ($N = 6$). En bas gauche nous retrouvons toutes les séries de déplacements relatifs ($RDTS_i$). En bas droite nous avons la série temporelle de déplacements consolidés en fonction de $RDTS_{ref}$, et décalé pour être relatif à la date initiale.

partir de ces calculs, les valeurs aberrantes sont retirées des observations initiales et le RDTS robuste est recalculé. Les valeurs médianes sont, in fine, utilisées comme mesure robuste de déplacement. La figure 7 et l'algorithme 1 illustrent ce processus. Que ce soit pour la figure ou l'algorithme, le résultat n'est pas recalé sur la première date afin d'être utilisé de manière glissante pour les grandes portées temporelles.

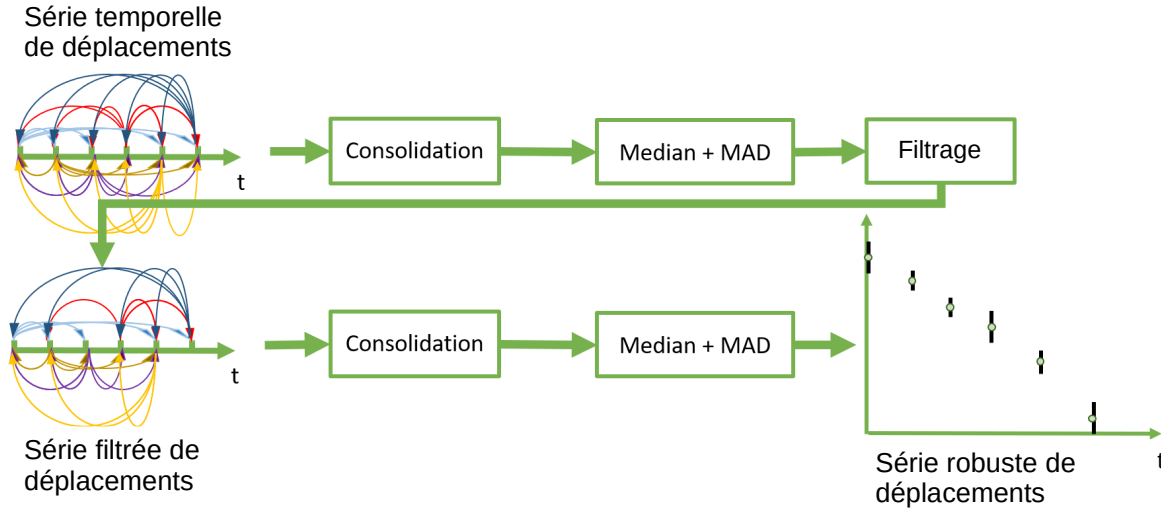


FIGURE 7 – Chaîne de traitements complète pour la construction d'un RDTS robuste : double consolidation avec filtrage intermédiaire.

Grande portée temporelle Lorsque la portée temporelle est trop grande, il n'est pas possible de calculer le déplacement entre 2 images. Soit Δt_{max} l'intervalle de temps maximal entre deux images tel que le déplacement entre ces 2 images puisse être calculé. La série initiale de tous les déplacements $DATA$ est donc traitée par blocs de sous-séries $sDATA_k$ telles que :

$$\vec{d}^{i,j} \in sDATA_k \quad \forall i, j \in [k - \Delta t_{max} : k + \Delta t_{max}], \vec{d}^{i,j} \in DATA$$

La sous-série qui minimise l'erreur donnée par l'équation 18 est sélectionnée comme sous-série de référence et initialise le RDTS final. Par la suite, et de manière itérative, chaque sous-RDTS calculé à partir de chaque sous-série est fusionné au RDTS final selon l'algorithme 2. L'ensemble des sous RDTS pouvant donner plusieurs valeurs à un offset près pour une même date, la valeur médiane corrigée de l'offset est conservée. Au final, nous obtenons un unique RDTS pour l'ensemble des mesures de déplacement.

5.3.4 Mesure en environnement virtuel

Les approches de mesures de déplacement que nous avons abordées dans les sections précédentes ont toutes un point commun. Partant d'une installation stéréoscopique installée de manière permanente sur le terrain, les images sont traitées de manière monoscopique avec ou sans redondance, puis l'apport de la stéréoscopie 3D est utilisée en post traitement afin de remonter à une mesure métrique. Les

Data : $DATA$: array of $[d^{(i,j)}] \forall i, j \in [0 : N[$
Result : A RDTS : array of $[d^{(0,i)}] \forall i \in [0 : N[$;
Function $\text{buildCM}(DATA : \text{array of } [d^{(i,j)}] \forall i, j \in [0 : N[) :$
 foreach $d_{i-j} \in DATA$ **do**
 | $CM[i][j] \leftarrow d_{i-j}$
 end
 return CM ;
End Function;
 $\text{min}_{score} \leftarrow -1$;
 $\text{min}_{idx} \leftarrow -1$;
 $CM = \text{buildCM}(DATA)$;
for $i \leftarrow 0$ **to** $N - 1$ **do**
 | $error \leftarrow 0$;
 for $j \leftarrow 0$ **to** $N - 1$ **do**
 | $error \leftarrow error + \text{calcError}(CM[i], CM[j])$ (see Equation 18);
 end
 if $\text{min}_{score} = -1$ **or** $error < \text{min}_{score}$ **then**
 | $\text{min}_{score} \leftarrow error$;
 | $\text{min}_{idx} \leftarrow i$;
 end
end
for $i \leftarrow 0$ **to** $N - 1$ **do**
 | $o \leftarrow \text{offset}(CM[\text{min}_{idx}], CM[i])$ (see Equation 17);
 for $j \leftarrow 0$ **to** $N - 1$ **do**
 | $RDTSDATA[j].\text{add}(CM[i][j] - o)$
 end
end
for $i \leftarrow 0$ **to** $N - 1$ **do**
 | $RDTS[i] \leftarrow \text{median}(RDTSDATA[i])$;
end
for $i \leftarrow 0$ **to** $N - 1$ **do**
 | $RDTS[i] \leftarrow RDTS[i] - RDTS[0]$;
end

Algorithme 1 : MMCMS algorithm.

Data : array of $[\vec{d}^{(i,j)}] \forall i, j \in [0 : N[$
Result : A RDTs : array of $[\vec{d}^{(0,i)}] \forall i \in [0 : N[$;
Function getSubSeries(array DATA of $[\vec{d}^{(i,j)}] \forall i, j \in [0 : N[, date, \Delta t_{max}] :$
 foreach $\vec{d}^{(i,j)} \in DATA$ **do**
 if $date - \Delta t_{max} < i < date + \Delta t_{max}$ and $date - \Delta t_{max} < j < date + \Delta t_{max}$ **then**
 $sDATA.add(\vec{d}^{(i,j)})$;
 end
 end
 return $sDATA$;
End Function ;
for $i \leftarrow 0$ **to** $N - 1$ **do**
 $sRDTs[i] \leftarrow MMCMS(getSubSeries(i, \Delta t_{max}))$;
 $errors[i] \leftarrow sRDTs[i].error$;
end
 $min_{idx} \leftarrow \text{indexOfMin}(errors)$;
 $RDTs \leftarrow sRDTs[min_{idx}]$;
 $j \leftarrow \max(0, min_{idx} - \Delta t_{max})$;
for $k \leftarrow 0$ **to** $\text{len}(sRDTs[min_{idx}]) - 1$ **do**
 $RDTs_{DATA}[j].add(sRDTs[min_{idx}][k])$;
 $RDTs_{Errors}[j].add(errors[min_{idx}])$;
 $j \leftarrow j + 1$
end
for $i \leftarrow min_{idx} + 1$ **to** $N - 1$ **do**
 $j \leftarrow \max(0, i - \Delta t_{max})$;
 $o \leftarrow \text{offset}(RDTs, sRDTs[i])$ (see Equation 17);
 for $k \leftarrow 0$ **to** $\text{len}(sRDTs) - 1$ **do**
 $RDTs_{DATA}[j].add(sRDTs[k] - o)$;
 $RDTs_{Errors}[j].add(errors[i])$;
 $RDTs[j] \leftarrow \text{median}(RDTs_{DATA}[j])$;
 $j \leftarrow j + 1$
 end
end
for $i \leftarrow min_{idx} - 1$ **to** 0 **do**
 $j \leftarrow \max(0, i - \Delta t_{max})$;
 $o \leftarrow \text{offset}(RDTs, sRDTs[i])$ (see Equation 17);
 for $k \leftarrow 0$ **to** $\text{len}(sRDTs) - 1$ **do**
 $RDTs_{DATA}[j].add(sRDTs[k] - o)$;
 $RDTs_{Errors}[j].add(errors[i])$;
 $RDTs[j] \leftarrow \text{median}(RDTs_{DATA}[j])$;
 $j \leftarrow j + 1$
 end
end

Algorithmme 2 : Version glissante de l'algorithme MMCMS.

travaux originaux de la thèse d’Haixing He [He, 2017] prennent le problème dans le sens inverse. Des acquisitions multivues sont mises en place pour réaliser des modèles numériques 3D métriques multitemporels. Celles-ci peuvent être aéroportées (avion, drone, dirigeable...) ou terrestres avec des séries de prise de vue en se déplaçant autour de l’objet observé. Ces modèles 3D de la scène observée seront l’environnement de mesure de déplacement [He et al., 2016, He et al., 2015]. L’avantage visé par cette approche est de ne travailler que dans un environnement 3D virtuel afin de maîtriser l’ensemble des paramètres. Il est ainsi possible, dans cet environnement, de positionner de manière exacte les appareils photo, de définir sans erreur leurs paramètres intrinsèques et de ne pas souffrir des effets en environnement réel : dilatation thermique du matériel, distorsion optique...

La première étape consiste à définir les caractéristiques (dimension et position) de la scène. Pour ce faire, nous définissons une sphère englobante de celle-ci, elle permettra de définir les positions et paramètres des appareils photo. Vient en suite la définition de l’appareil photo virtuel. Notons que, étant dans un environnement virtuel, nous ne sommes pas limités aux appareils de type sténopé (projection conique), nous pouvons définir un appareil à projection orthogonale. Choix que nous avons fait afin de faciliter les écritures et calculs, en effet, la projection orthogonale ne nécessite pas de définition de longueur focale et ne nécessite pas l’utilisation de coordonnées homogènes. Une fois l’appareil photo défini, il ne reste plus qu’à définir ses positions sur la sphère englobante en fonction d’un écartement n’excédant pas 20° entre 2 appareils côte à côte, ordre de grandeur de la limite communément admise en photogrammétrie. Le système d’acquisition virtuel étant ainsi défini, il est

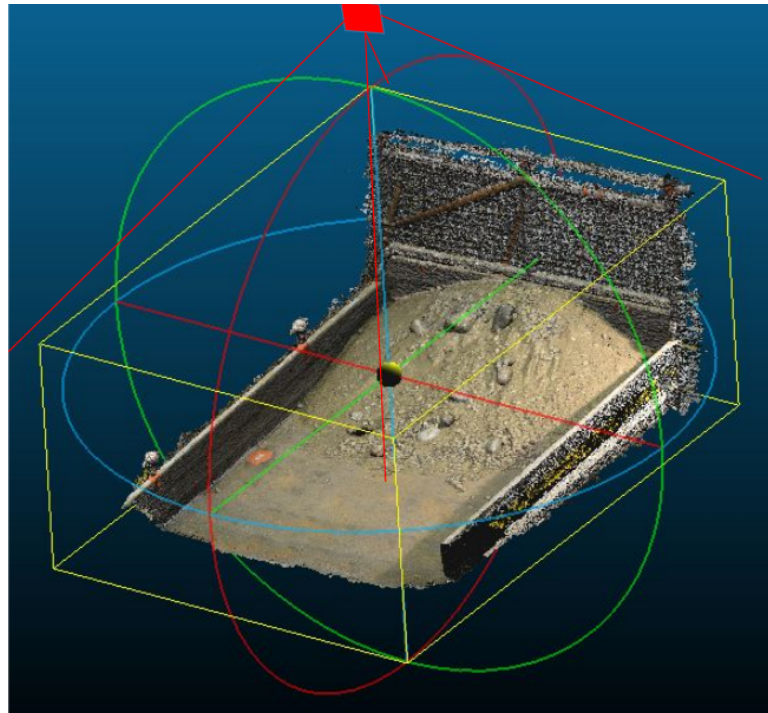


FIGURE 8 – Modèle 3D d’acquisition virtuel reconstruit à partir d’une acquisition multivue. Pavé jaune : boîte englobante. Cercle rouge, vert et bleu : sphère englobante. Rectangle rouge : position initiale d’acquisition.

utilisé sur chaque reconstruction 3D au cours du temps et fournit, pour chaque position d’appareil, une série nomoscopique virtuelle qui est traitée en utilisant les approches précédentes. Dans le cadre de la

thèse d'Haixing, nous nous sommes limités à la corrélation classique afin de déterminer les changements dans les séries monoscopiques.

Du point de vue numérique, étant dans un environnement 3D virtuel, nous nous sommes appuyés sur l'approche « LookAt », standard dans le domaine 3D numérique. Ainsi, les paramètres extrinsèques des appareils photo sont définis par :

- C le vecteur position du centre de la caméra dans le repère virtuel,
- p le vecteur position dans le repère virtuel du point que la caméra vise,
- Up le vecteur direction, dans le repère virtuel, du haut de la caméra.

Ceci définit la matrice extrinsèque (3x3) M^i de chaque appareil i . Matrice qui est, à une discrétisation près, la matrice de projection (3x3) P^i . Posons P_j^i la j ème ligne de P^i . L'acquisition d'une photo virtuelle pour un appareil i s'écrit sous la forme matricielle suivante :

$$P_{2D}^i = \begin{bmatrix} P_0^i \\ P_1^i \end{bmatrix} P_{3D},$$

où P_{2D}^i sont les points 2D (la photo) et P_{3D} les points 3D de la scène. Notons que la 3e ligne de la matrice de projection n'est pas utilisée du fait de la projection orthogonale.

L'avantage de cette approche est une totale maîtrise des paramètres dans l'environnement virtuel. Cependant, les problématiques apparaissant dans les autres approches ne sont que déplacées. Si le problème de recalage de série d'images n'est plus à gérer dans cette approche, les séries de reconstructions 3D, quant à elles, doivent être initialement recalées sur des zones fixes. Il n'en reste pas moins que l'approche est fonctionnelle et permet d'observer les changements d'une scène en milieu naturel.

5.3.5 Rendu métrique

Dans le cadre des chaînes de traitements de séries d'images pour l'observation d'objets géophysiques, les post-traitements consistent fréquemment à retrouver une mesure 3D métrique à partir des mesures pixeliques effectuées depuis les séries d'images. Des résultats typiques sont illustrés en figure 9. Il existe 2 approches classiques que nous avons utilisées dans nos travaux.

La première, qui est la plus simple et la plus fréquemment utilisée, consiste à projeter les vecteurs de déplacement pixelique sur un modèle 3D. Dans cette approche, l'apport de l'installation stéréoscopique est de fournir un modèle 3D métrique de référence, sur lequel est reprojeté les vecteurs de déplacement en chaque pixel de l'image. La reprojection du vecteur pixelique peut s'effectuer de différentes manières. Connaissant le pixel de départ et celui d'arrivée (au 10e de pixel près), un lancer de rayon peut être utilisé afin de déterminer les points 3D de départ et d'arrivée. La reconstruction ayant été réalisée à partir des images traitées, la correspondance 2D/3D est connue, il est donc possible de retrouver les points 3D de départ et d'arrivée. Dans tous les cas, les points 3D d'arrivée seront interpolés, il est quasi impossible de tomber exactement sur un point existant. De plus, le modèle 3D étant relativement moins dense que les images d'origines, certains points seront à reconstruire.

La seconde approche se base sur l'inversion de modèle. Celle-ci a notamment été utilisée dans l'environnement virtuel, l'ortho projection limitant le nombre de paramètres. Un déplacement dans une image étant la projection du déplacement 3D dans cette image, le système peut s'écrire de la manière

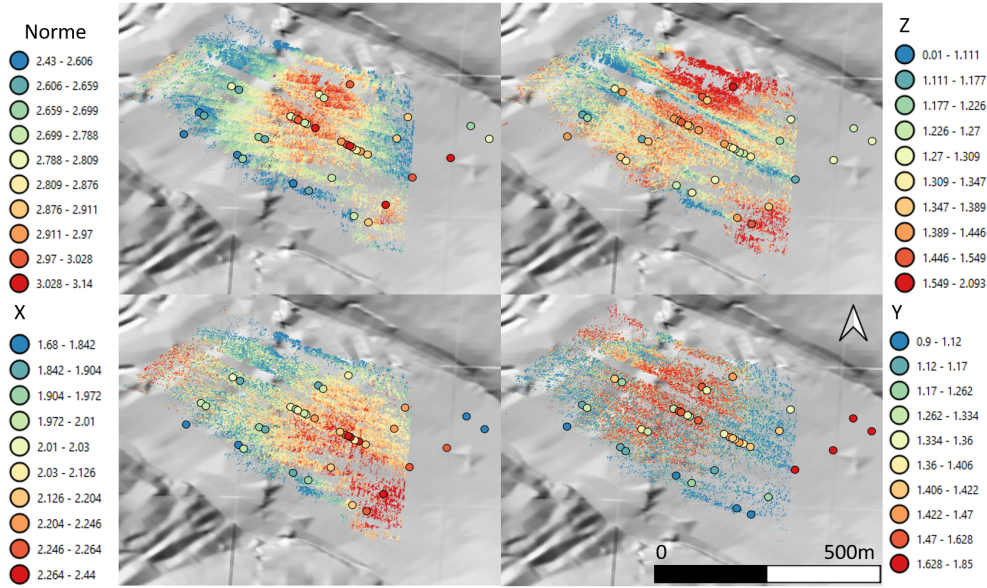


FIGURE 9 – Rendu 3D d'un déplacement métrique calculé par stéréo photogrammétrie. Chaque disque étant la vérité terrain mesurée par GPS. Nous noterons l'erreur plus importante en Y qui correspond majoritairement à l'axe de visée du système d'acquisition.

suivante :

$$\begin{pmatrix} d_x^0 \\ d_y^0 \\ \vdots \\ d_x^N \\ d_y^N \end{pmatrix} = \begin{bmatrix} P_0^0 \\ P_1^0 \\ \vdots \\ P_0^N \\ P_1^N \end{bmatrix} \begin{pmatrix} d_X \\ d_Y \\ d_Z \end{pmatrix},$$

où les d_x^i, d_y^i sont respectivement les déplacements 2D en x et y calculés sur l'image i . Par moindres carrés ou SVD, connaissant les d_x^i, d_y^i et les matrices de projection, nous retrouvons le déplacement 3D d_X, d_Y, d_Z . Comme nous maîtrisons les acquisitions dans l'environnement virtuel, il est facile de surcontraindre le système.

5.4 Approche conceptuelle des chaînes de traitements

L'ensemble des travaux présentés précédemment ont donné lieu à la réalisation de chaînes de traitements illustrées par les figures 10, 11, 12 et 13. Elles correspondent respectivement aux travaux réalisés dans les thèses de Ha Thai Pham [Pham, 2015], Héli Hadrhi [Hadrhi, 2020], Guilhem Marsy [Marsy, 2020] et Haixing He [He, 2017]. Chacune d'elles est implémentée soit de manière parallèle, pour machines multicœurs, soit de manière distribuée pour des clusters de calcul ou machines multicœurs. Nous retrouvons dans ces chaînes des traitements similaires telles que le recalage de série d'images, explicitement représenté dans les figures 10 et 12b, et implicite dans la figure 11 où les séries initiales sont considérées recalées ; ou la reprojection des champs de vecteurs déplacements 2D sur un modèle 3D afin d'obtenir des déplacements 3D dans les figures 11 et 12c ; notons également que la fin de la chaîne de Haixing, à partir des orthoimages, figure 13 est la chaîne de Ha Thai après

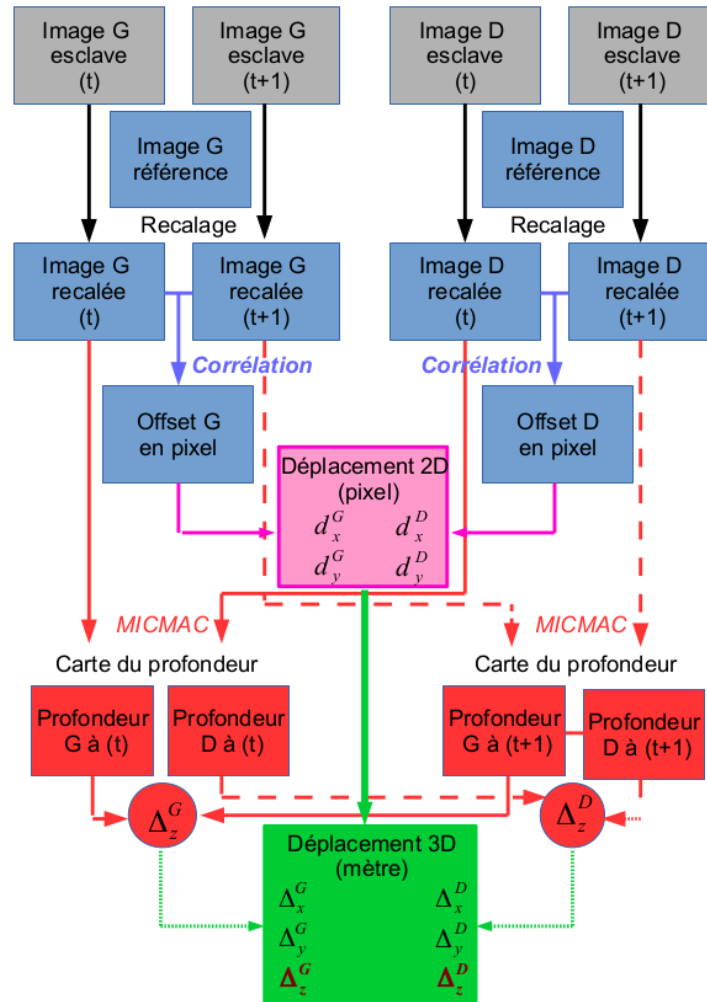


FIGURE 10 – Chaîne de traitements effectuant le calcul de déplacement 3D à partir de séries d'images stéréoscopiques, chaîne réalisée dans le cadre de la thèse d'Ha Thai Pham.

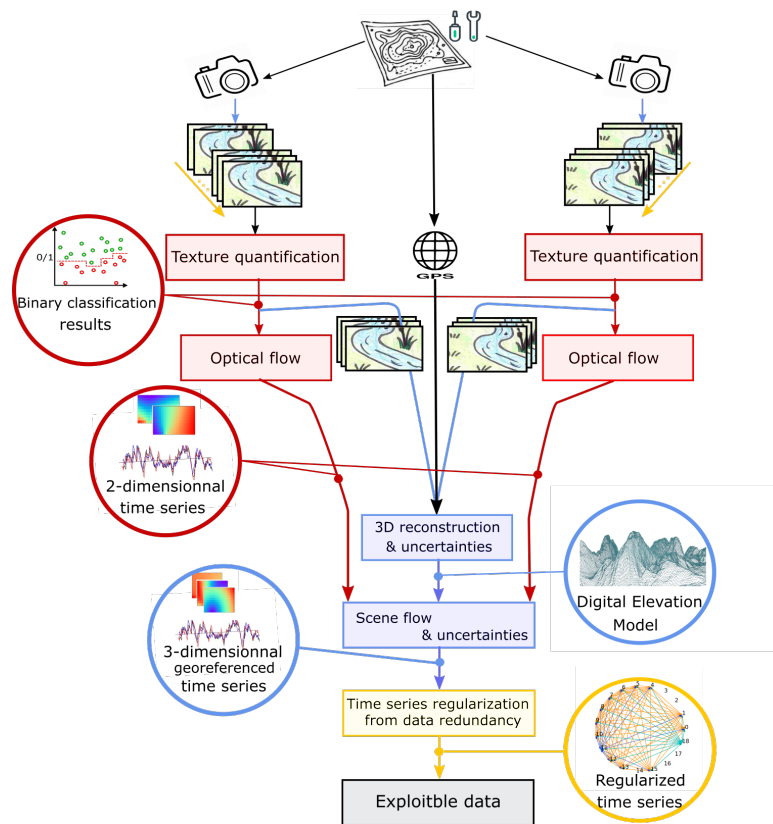


FIGURE 11 – Chaîne de traitements effectuant le calcul de déplacement 3D à partir de séries d’images stéréoscopiques et minimisant l’erreur de mesure par redondance de calcul, fermeture temporelle et inversion de modèle ; chaîne réalisée dans le cadre de la thèse d’Héla Hadhri.

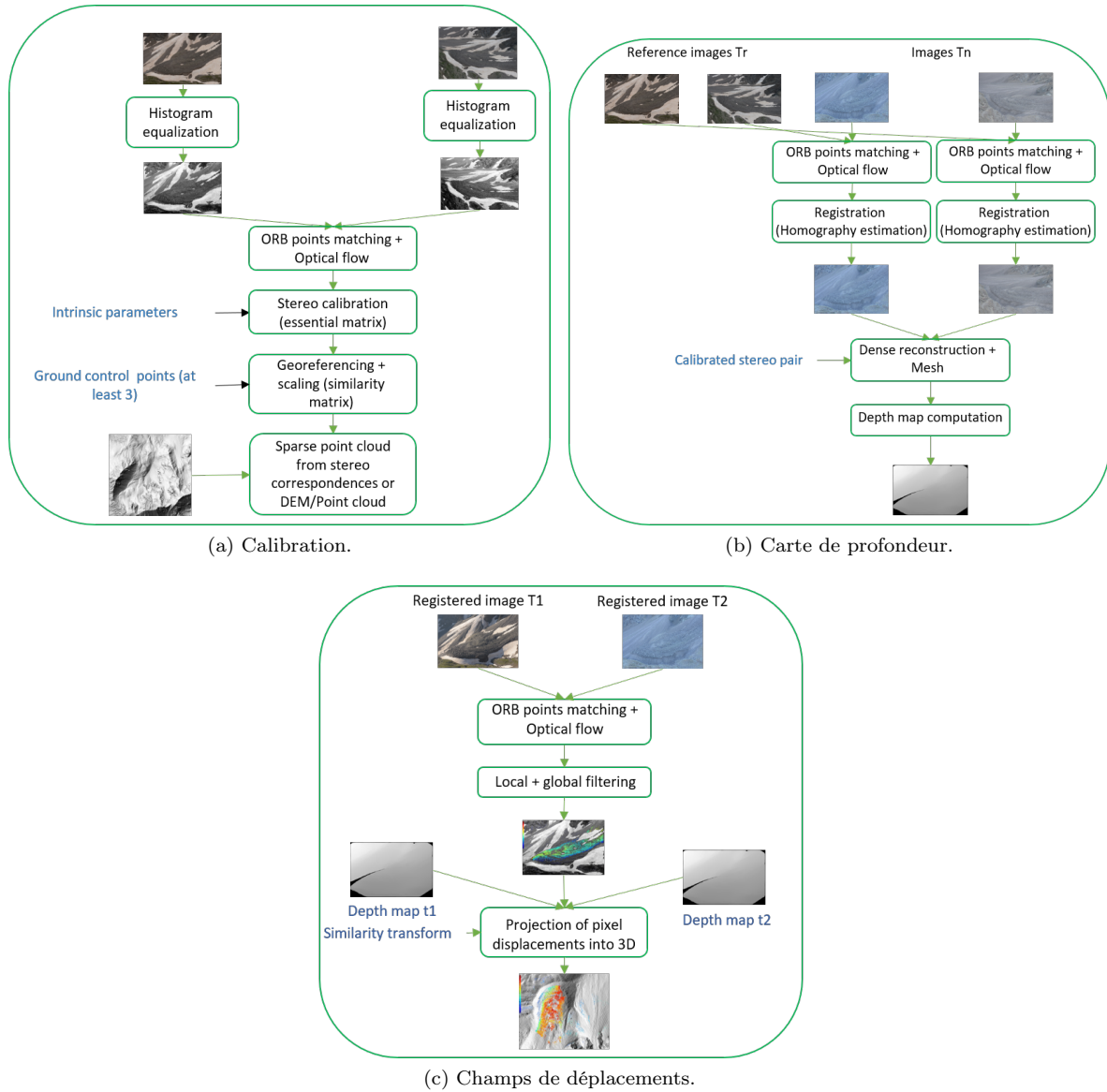


FIGURE 12 – Chaîne de traitements effectuant le calcul de déplacement 3D à partir de séries d'images stéréoscopiques et minimisant l'erreur de mesure par redondance de calcul et fusion de résultats par médians glissants, chaîne réalisée dans le cadre de la thèse de Guilhem Marsy.

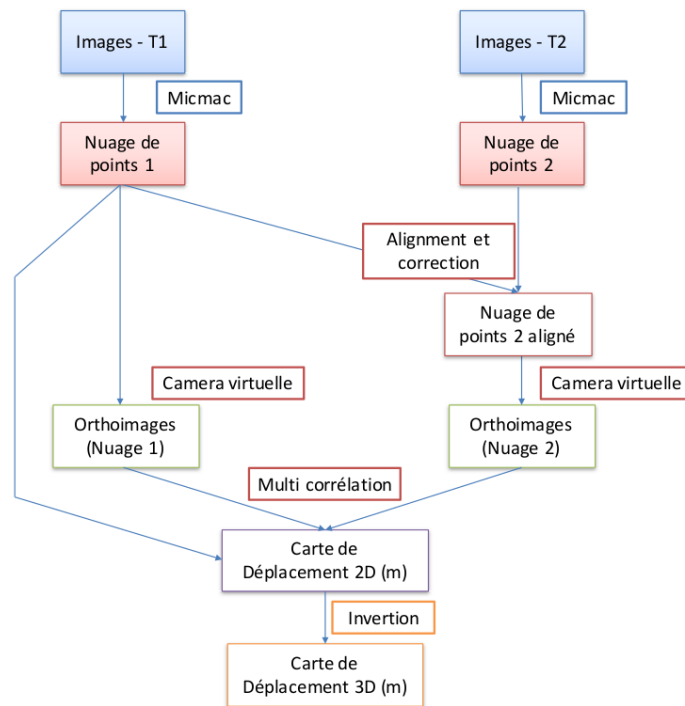


FIGURE 13 – Chaîne de traitements effectuant le calcul de déplacement 3D dans un environnement virtuel reconstruit à partir d’acquisitions multivues, chaîne réalisée dans le cadre de la thèse Haixing He.

recallage figure 10. Chaque chaîne apporte également ses traitements originaux qui lui sont propres. La chaîne figure 10 est la base minimale pour obtenir un champ de déplacement 3D, Héla, figure 11, explore une méthode de flow optique en remplacement de la corrélation figure 10 et introduit surtout dans notre problématique les techniques de fermeture temporelle pour réduire l'erreur et régulariser le résultat ; Guilhem, figure 12c, introduit une approche originale de filtrage utilisant la redondance de mesures pour obtenir les champs de déplacement 2D. Quant à Haixing, figure 13, l'originalité vient de l'approche inverse en commençant par la reconstruction 3D afin de travailler dans un environnement virtuel 3D pour tous les traitements.

De manière générale, chacune de ces chaînes est un DAG de flot de traitements ayant la particularité (i) de traiter des séries d'images où chaque image varie entre la dizaine de Mo pour les images optiques terrestres et plusieurs Go pour les images SAR satellitaires, (ii) de produire une quantité de résultats intermédiaires nettement supérieure au flux d'entrée, (iii) de nécessiter des temps minimums de traitements de l'ordre du jour et pouvant facilement atteindre l'ordre de la semaine. Ceci lève les problématiques suivantes : comment spécifier ces chaînes, comment les exécuter de manière efficace sur des architectures multicœurs et de type cluster de calcul, et ceci de manière la plus intuitive et la moins intrusive pour l'utilisateur final. Au niveau exécution, comment faire pour que l'orchestrateur gère les architectures multicœurs, CPU/GPU et de type cluster de calcul de manière la plus efficace ? Comment analyser les exécutions afin d'en étudier les briques chronophages pour les refactoriser, les optimiser ou les distribuer, et les retester dans des conditions similaires ?

Dans un premier temps, nous nous sommes intéressés à la spécification et l'exécution des chaînes en tenant particulièrement compte des utilisateurs directement concernés au travers de différents projets. Tout d'abord, en raison de la taille et la quantité de données, le transfert de données entre les différentes étapes s'effectue au travers de fichiers. Ceci permet également de conserver des résultats intermédiaires de grande taille afin de ne pas les recalculer lors de besoins futurs. Pour les chaînes, nous avons donc opté pour une description de dépendance de tâche se basant sur les fichiers d'entrées/sorties de celle-ci. L'avantage de ce choix est qu'il existe un certain nombre d'outils de traitement de chaîne de traitements scientifique, notamment l'outil « GNU make », qui répondent à cette problématique ainsi que celle de la gestion des architectures multicœurs. Le choix de « GNU make », comme gestionnaire de chaîne de traitements et non l'utilisation d'un système dédié comme Anduril [Cervera et al., 2019], Apache AirFlow ou autres c'est naturellement justifié pour les raisons suivantes : (i) « GNU make » est généralement installé sur les stations de travail utilisées, (ii) il ne nécessite pas l'installation d'un système tiers complet qui ne soit pas déjà installé en standard sur les stations et (iii) dans le contexte des EFIDIR Tools où ces chaînes sont développées, les « autotools », dont « GNU make », sont déjà intégrés comme système de compilation. Cependant, « GNU make » ne gère pas l'exécution sur cluster (multi machines) et son utilisation – la description d'une chaîne de traitements selon son formalisme – est très rébarbative et nécessite une expertise sur son utilisation.

Afin de pallier le problème de convivialité de description, nous avons proposé un formalisme Python encapsulant celui de description d'un « Makefile ». Celui-ci est défini selon le squelette donné dans le listing 1. Chaque tâche (nœud du DAG, opérateur...) est définie selon cette structure, ce qui donne, in fine, une liste de « targets » dépendantes selon leurs fichiers d'entrée/sortie. L'exécution ne se fait que sur demande suivant le code illustré dans le listing 2. La chaîne de traitements est tout d'abord créée, puis exécutée si besoin. L'ensemble de ce système est implémenté dans la suite d'outils logiciels EFIDIR-Tools puis utilisé comme base de refactoring de la chaîne NSBAS [Volat et al., 2014].

Quant au problème d'exécution sur cluster de calcul, il est double. Premièrement, l'exécution distribuée requiert un système de communication et de gestion de processus adapté, le choix c'est

```

1  # Define F00 job
2
3  # list of output files
4  outputs = [ ... ]
5
6  # list of input files
7  inputs = [ ... ]
8
9  # list of commands for F00 job
10 commands = [
11     "foo"
12     + " " + inputs[0]
13     + " " + inputs[1]]
14
15 # create target for F00 job and add it into the list of targets
16 targets.append(target_create(commands, inputs, outputs))

```

Listing 1 – Description d’une tâche (étape) d’une chaîne de traitements.

```

1 chain = workflow_create("foo.mk", targets)
2 workflowrun(chain)

```

Listing 2 – Création et exécution d’une chaîne de traitements.

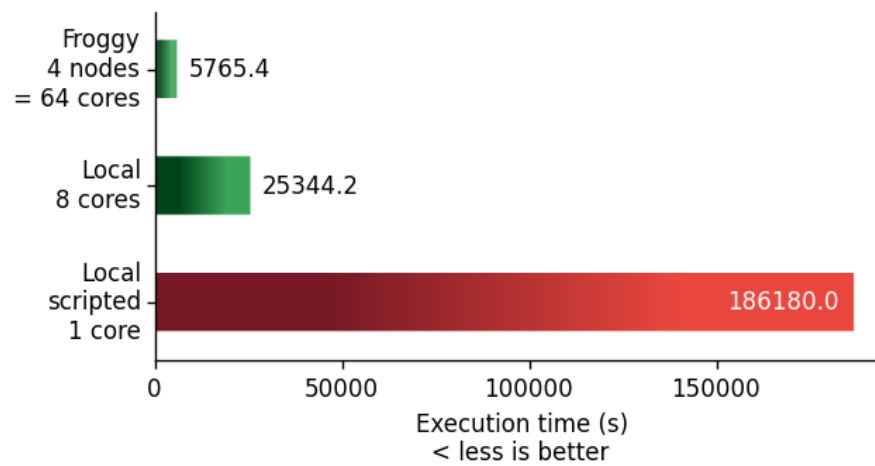
naturellement porté sur MPI, disponible de manière standard sur les infrastructures visées. Deuxièmement, « GNU make » génère un contexte d’exécution au sein de son environnement, ce contexte lui permet de gérer son flot, mais également ses propres exécutions récursives « make dans make », ce qui correspond dans notre cas à ce qu’une chaîne de traitements soit une sous-chaîne d’une chaîne plus large. Il apparaît que le problème est de plus grande ampleur, car :

- quand « GNU make » demande l’exécution sur une machine tierce (cluster), le contexte est perdu sur cette machine,
- quand « GNU make » demande l’exécution d’une sous-chaîne, il exécute un script Python dans notre implémentation – la sous-chaîne – qui lui-même lance l’exécution d’un nouveau « GNU make » et le contexte initial est perdu, que ce soit sur cluster ou machine multicœurs.

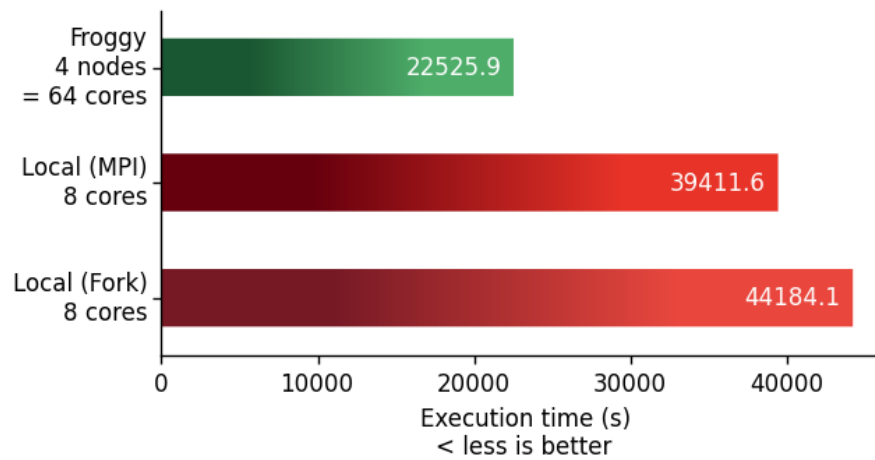
De ces constatations, l’encapsulation de « GNU make » que nous proposons :

- capture le contexte d’exécution du « GNU make » initial afin de pouvoir le propager selon les besoins, ceci pour chaque chaîne et sous-chaîne par nature,
- intercepte, dans le cas d’exécution sur cluster, l’exécution de sous-chaînes afin de les réaffecter au dispatcher de tâche (le « GNU make » initial), ceci afin qu’une sous-chaîne ne monopolise pas une machine en attendant que ses tâches soient terminées.

Ce formalisme, associé à l’intergiciel implémenté au sein des EFIDIR-Tools, permet de prendre pleinement la main sur l’exécution des chaînes de traitements, et ainsi monitorer leurs exécutions. De surcroît, grâce à l’appui sur l’outil « GNU make », il permet l’interruption et la reprise de l’exécution de la chaîne, la reprise sur erreur et le traitement parcellaire en cas de modification de paramètres pour une réexécution. Les figures 14a et 14b illustrent les gains apportés pour une chaîne de mesure de déplacement EFIDIR et la chaîne de mesure d’interférogrammes NSBAS. De par la nature des problèmes et leur implémentation, le gain est quasi linéaire pour la chaîne EFIDIR, alors qu’il est nettement moins important pour NSBAS. Ceci s’explique simplement par le fait qu’EFIDIR ait été conçu pour être distribué, alors que NSBAS était préexistant à ces travaux et n’a pas été conçu dans ce sens.



(a) Chaîne EFIDIR de mesure de déplacement par corrélation.



(b) Chaîne NSBAS de calcul d'interférogrammes.

FIGURE 14 – Benchmark des outils de chaînage EFIDIR au sein d'une chaîne EFIDIR et de la chaîne NSBAS.

5.5 Conclusion & réflexion

L'ensemble de ces chaînes de traitements a été appliqué sur différents sites naturels : glacier d'Argentière (France), glacier du Laurichard (France), Glacier du Miage (Italie)... Les résultats et leurs analyses peuvent être retrouvés dans les mémoires de thèse et articles [Benoit et al., 2015, He et al., 2015, Hadhri et al., 2019, Marsy et al., 2020a, Marsy et al., 2020b], ils ne seront pas détaillés ici puisque le sujet étudié est plus celui du traitement que celui de l'objet observé.

L'une des problématiques majeures du traitement d'images dans le cas de séries temporelles est le temps de traitement qui est de l'ordre du jour, de la semaine, voir du mois pour traiter en pleine résolution l'ensemble des séries. Les approches « deep learning » ont permis, pour certaines tâches, de grandement réduire leur temps de traitement, parfois au détriment de la qualité des résultats, qui n'est pas toujours un objectif, par exemple dans le cas de la reconstruction 3D stéréo ou monoscopique temps réel où le visuel prime plus que la précision. Dans le contexte de la photogrammétrie, contexte dans lequel se placent les travaux que je viens de présenter, la qualité et la précision de la mesure ainsi que la quantification de l'erreur de mesure sont des points primordiaux. Le problème du temps de calcul reste donc un enjeu majeur. Cette problématique est diverse, elle intègre des axes de recherche tels que l'optimisation numérique, la parallélisation/distribution de traitements, la gestion de grandes masses de données... Dans les travaux présentés dans cette partie, outre l'objectif opérationnel qu'est la quantification du mouvement de l'objet observé, l'optimisation des tâches de corrélation, la mise en place du formalisme des chaînes de traitements et leur exécution distribuée visent à réduire les temps de traitement et rendre accessible, voir transparent, l'accès aux architectures multicœurs et de type cluster de calcul.

Pour ma part, je m'intéresse à l'acquisition de connaissances sur l'exécution de ces chaînes. Cette connaissance peut être utile par exemple pour la prédiction des temps de traitement, l'objectif opérationnel étant d'alerter le système et l'utilisateur en cas de temps de traitement inhabituel. Des approches de monitoring d'exécution classique ou d'injection de sondes dans les programmes sont tout à fait envisageables. Cependant, ces approches sont intrusives au niveau du code métier et ne sont pas génériques, chaque programme doit être instrumenté et chaque chaîne monitorée de manière explicite. L'objectif que je vise se veut générique et le plus transparent possible. En s'appuyant sur des approches d'ingénierie logicielle, est-il possible de rendre les chaînes de traitements en particulier, et les logiciels en général, capables de se connaître eux-mêmes, intelligents et interactifs avec l'utilisateur. La section suivante présente cette nouvelle problématique dérivant de la conception des chaînes de traitements. Elle présente les questions et verrous aussi bien du point de vue conceptuel que technologique, ainsi que les travaux déjà réalisés.

6 Architecture de traitement de l'habitude

6.1	Contexte	56
6.2	De la sagesse à l'intelligence	57
6.2.1	Acquisition de la sagesse	57
6.2.2	Infrastructure du traitement de la sagesse	61
6.3	Simulation de comportements	62
6.3.1	Timer	62
6.3.2	Action	65
6.3.3	Simulateur	67
6.4	Modélisation de l'habitude et détection de l'inhabituel	68
6.4.1	Modélisation de la mémoire	68
6.4.2	Mesure de la distance à l'habituel	70
6.5	Applications	71
6.5.1	Domotique : synthèse de contrôleurs	71
6.5.2	Domotique : Wise Internet of Things	73
6.6	Perspectives	75

6.1 Contexte

Le terme intelligent, qui suggère principalement l'aptitude à relier des éléments de connaissance qui sans elle resteraient scindés, est de plus en plus utilisé dans les systèmes logiciels : les « Smart » systèmes. Ceux-ci doivent naturellement se baser sur un savoir – un ensemble de connaissances – afin d'en inférer de nouvelles, les lier entre elles. Intégré à une chaîne de traitements d'images ou de données de manière plus générale, la capacité d'acquérir de la connaissance sur son utilisation et sur son comportement afin de les prédire, lui fournirait un savoir de base indispensable à la rendre intelligente.

Cette acquisition du savoir et l'analyse de celui-ci ne sont pas des problématiques nouvelles, nous les retrouvons dans les analyses de « log » [Fu et al., 2009, Reidemeister et al., 2011, Reidemeister et al., 2009, Russo et al., 2015], la détection d'intrusions [Kumar and Hanumanthappa, 2013, Yusof et al., 2008], les systèmes « intelligents » de manière générale : les systèmes à agents, les systèmes auto-adaptatifs [Kennedy, 2010, Oreizy et al., 1999, Cheng et al., 2009, GEORGE et al., 2003]... Cependant, ces systèmes sont dédiés à un contexte d'application, l'environnement dans lequel ils évolueront, ce qui les rend difficilement génériques du point de vue applicatif, même si leurs concepts structurels le sont. Dans notre contexte de chaînes de traitement, ceci peut poser problème du fait que chaque brique, chaque étape est indépendante et leurs contextes – leurs entrées/sorties – peuvent être très différents (un fichier image, son ou autre, un paramètre numérique, alphabétique, un accès distant...). Et il n'est pas concevable d'étudier, pour chaque brique, chaque chaîne, l'acquisition et l'analyse de sa connaissance. De ce constat, la question qui se pose est :

« Existe-t-il un contexte assez générique du point de vue conceptuel et applicatif où la connaissance, son acquisition et son traitement au sein d'un système informatique puissent être indépendants de ce système ? »

Le paradigme objet et les différents langages qui l'implémentent permettent de répondre à cette problématique. Voyons l'exécution de notre brique de traitement comme un objet – une instance de classe – par composition, notre chaîne de traitement est un objet. Les entrées/sorties des briques ou chaînes sont des attributs de nos objets et, par conséquent des objets. Pour finir, les opérations effectuées sont des méthodes. Ce niveau d'abstraction, qui nous fait perdre la sémantique des données et des traitements effectués nous apporte une vision plus générique du système. En effet, tout élément serait vu comme objet, attribut ou méthode et l'acquisition de la connaissance s'effectuerait sur ces concepts. Un autre avantage qu'apporte les langages-objets est une forte composante d'introspection, ils permettent ainsi à un objet de s'observer lui-même et ainsi acquérir de la connaissance sur lui.

Les objets sages, WO pour « Wise Object » ont été pensés dans ce sens, ils ont été introduits par nos soins pour la première fois dans [Alloui et al., 2015]. Un WO est donc un objet au sens logiciel du terme, qui a la capacité d'acquérir de la connaissance sur lui même, par lui-même : sa sagesse. Cette acquisition de la sagesse se veut sans aucun a priori et de manière totalement non supervisée.

Les exemples pris dans ce chapitre s'appuient sur l'IoT et plus particulièrement la domotique. L'approche n'en reste pas moins générique et peut s'appliquer à tout autre domaine, les traitements domotiques étant nettement moins chronophages que le traitement de séries temporelles d'images. Ce chapitre présente dans une première section, une approche conceptuelle des WO et leur « framework » associé, le WOF. Une seconde section introduit le simulateur à événements discrets permettant la reproductibilité des comportements lors des expérimentations et une approche afin de quantifier l'habitude et détecter les comportements inhabituels. La section 6.5 illustre ces travaux dans le contexte de l'IoT. Ce chapitre se conclut par un ensemble de perspectives qu'ouvrent ces recherches.

6.2 De la sagesse à l'intelligence

6.2.1 Acquisition de la sagesse

La sagesse – la connaissance sur soi – est de deux natures : quelles sont les capacités métiers du WO en question et quelle est l'utilisation qui en est faite. L'objet doit donc être capable d'extraire cette information par lui-même, en s'appuyant sur des mécanismes d'introspection pour le premier et d'analyse de « log » pour le second. Sans oublier qu'il a une tâche métier à réaliser. Nous avons donc défini deux états globaux pour les WO. Le premier est un état d'éveil où un WO fournit les services pour lesquels il a été programmé, cet état est naturellement déclenché lorsque l'objet est sollicité par un tiers. Le second est l'état de rêve, état dans lequel passe le WO dès qu'il n'est plus sollicité, dans cet état il pourra apprendre sur lui-même et son utilisation. Ceci s'apparente à du vol de cycle CPU, au détail près que les WO sont « égocentrique » et ne se soucient pas, pour l'instant, de savoir si d'autres ont besoin du CPU. Le diagramme états/transitions figure 15 illustre ce comportement. De

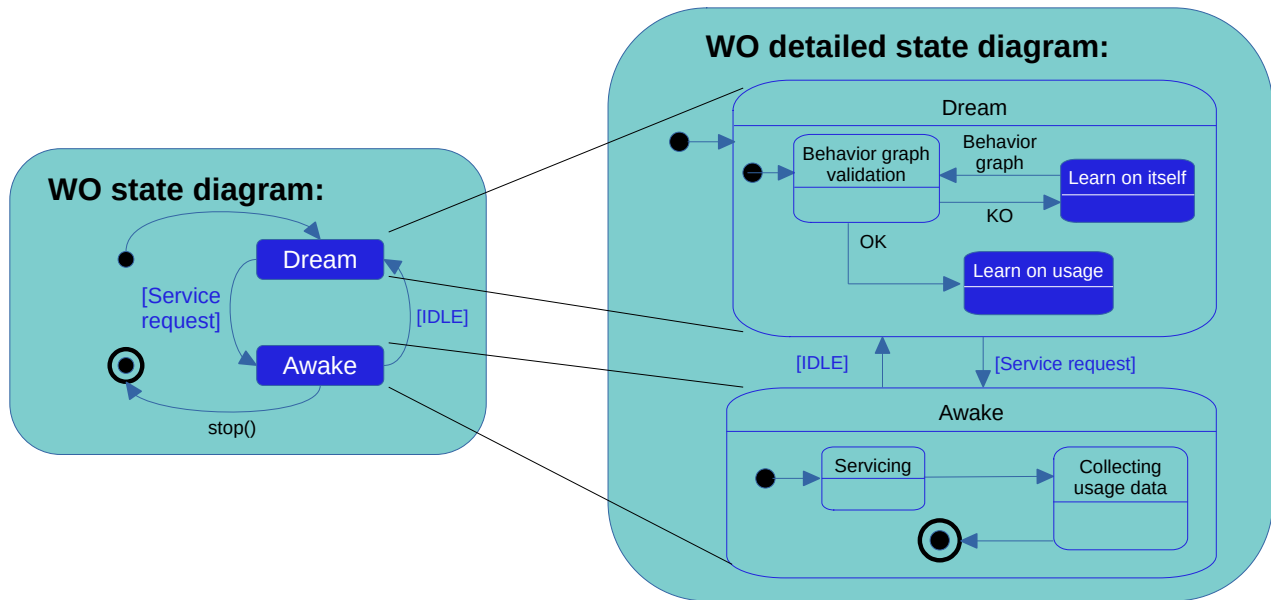


FIGURE 15 – Diagramme états/transitions UML modélisant le comportement d'un WO [Alloui and Vernier, 2017].

manière plus détaillée, pendant son état éveillé, le WO rend le service pour lequel il a été sollicité – exécute la méthode invoquée par un tiers – et « loggue » cette exécution. En phase de rêve il a deux tâches à exécuter. La première est son introspection afin d'apprendre ses capacités métiers, cette phase est détaillée dans le paragraphe suivant. La seconde est l'analyse de ses « log » afin d'apprendre ses habitudes d'utilisation et détecter par exemple des comportements inhabituels. Cette dernière tâche est déléguée à un analyseur de comportement afin d'étudier différentes approches pour modéliser l'habituel et l'inhabituel. Concernant ces analyseurs, la section 6.4.2 détaille notre approche.

L'acquisition de la connaissance sur ses capacités métiers peut être abordée de différentes manières, via de l'analyse de code ou via une modélisation fournie par le développeur, comme nous l'avons effectuée lors de la thèse de Fabien Sartor [Moreaux et al., 2012], [Sartor, 2012]. L'approche analyse de code présente le défaut d'être complexe et gourmande en ressources notamment au travers d'un ana-

lyseur spécifique de « reverse engineering » tel que Moose [Anquetil et al., 2020], quant à l'information fournie par le développeur, elle présente toujours le risque d'être erronée ou non mise à jour suite à des évolution ou « refactoring » du code. Nous avons donc choisi, pour les WO, d'étudier une nouvelle approche, le WO construit son graphe états/transitions métier à la volée. Cette problématique d'inférence de comportement a déjà été étudiée via des approches boîte noire [Groz et al., 2018], ou boîte blanche en reverse engineering [Ziadi et al., 2011]. En ce qui concerne les WO, nous nous plaçons dans une approche boîte grise, nous avons accès à la totalité du code, mais nous ne nous appuyons que sur les attributs et les appels de méthodes. L'originalité tient principalement au fait que l'entité, le WO, infère lui-même son comportement afin d'utiliser cette nouvelle connaissance. Il peut ainsi chercher à construire un graphe complet, comme c'est le cas à l'heure actuelle, ou construire un graphe partiel corrélé à son utilisation, ceci pourra être utile à l'avenir pour pallier en partie le problème d'extensibilité lié à l'utilisation de ce type de graphe pour un système. Le principe de base pour construire ce graphe est, pendant sa phase de rêve, le WO invoque ses propres méthodes métiers et observe ses changements d'état, un état étant défini par un ensemble de valeurs de ses attributs. Ceci induit que, en plus des capacités d'introspection, le WO doit avoir par nature des capacités de réflexivité. Cette approche par invocation automatique est une problématique connue dans le domaine de la génération automatique de test, notamment dans le choix des valeurs des paramètres des méthodes [Oriat, 2005]. De ce fait, pour les WO, nous nous limitons, dans un premier temps, à des méthodes non paramétrées, cependant, des approches basées sur la mémorisation des paramètres déjà utilisés par le passé en mode éveillé et avec variations de valeur ou d'état seront envisagées pour prendre en compte les méthodes paramétrées. Par chaque invocation qu'il effectue, un WO est donc capable de reconstruire son graphe états/transitions métier, notons que si une méthode invoquée en phase d'éveil l'amène dans un état qu'il ne connaît pas encore, le graphe est naturellement mis à jour. Les conséquences sont que, dans sa phase de rêve, le WO change d'état, un système de sauvegarde et restauration d'état est donc en place au sein des WO pour que leur état en sortie de rêve soit identique à celui en entrée en phase de rêve. La figure 16 donne la représentation UML d'un WO. Un WO s'appuie sur une interface de graphe qui lui permet de stocker sa connaissance – son graphe états/transitions – qu'il construit par réflexivité et introspection ; ce graphe est surchargé des différentes dates de transitions acquises à partir des « log » d'utilisation, afin d'inférer ses habitudes d'utilisation. Dans un second temps, nous remarquons qu'un WO est abstrait. Ainsi, la classe WO, au sein du WOF, apporte la capacité à un objet tiers d'apprendre sur lui-même, pour ce faire il doit étendre la classe WO. Les méthodes « `methodInvoke()` » et « `methodInvoked()` » sont les 2 méthodes qui décorent celles de l'objet tiers pour collecter les « log » d'utilisation.

De manière plus globale, au sein du système, les WO interagissent entre eux et l'ensemble du système. Ce qui pose problème lorsqu'un WO est en phase de rêve, en effet, dans cette phase, le WO peut invoquer par lui-même ses propres méthodes (celles de sa classe fille), cependant, dans cette phase, il ne doit pas avoir d'effet sur le système. Cette contrainte est la base de conception du WOF et de l'interconnexion des WO. Pour pallier cette problématique – l'action d'un WO ne soit effective sur le système que quand il est éveillé et qu'elle ne le soit pas quand il rêve – le système implémenté par le WOF s'articule autour d'un bus de communication. De ce fait, l'interaction entre WO s'effectue au travers d'un système à événements asynchrones, ainsi quand un WO est éveillé, il écoute le bus et publie ses actions sur le bus et quand il rêve, il ne fait que l'écouter afin de réagir à des sollicitations, mais ses actions ne sont pas publiées sur le bus et n'engendrent pas de réaction dans le système. Ce choix implique que le système d'action/réaction entre WO soit géré de manière externe au WO. Un manager a donc été mis en place afin de faire l'appairage entre les WO.

L'exemple de diagramme de séquence UML présenté en figure 17 illustre l'interconnexion entre

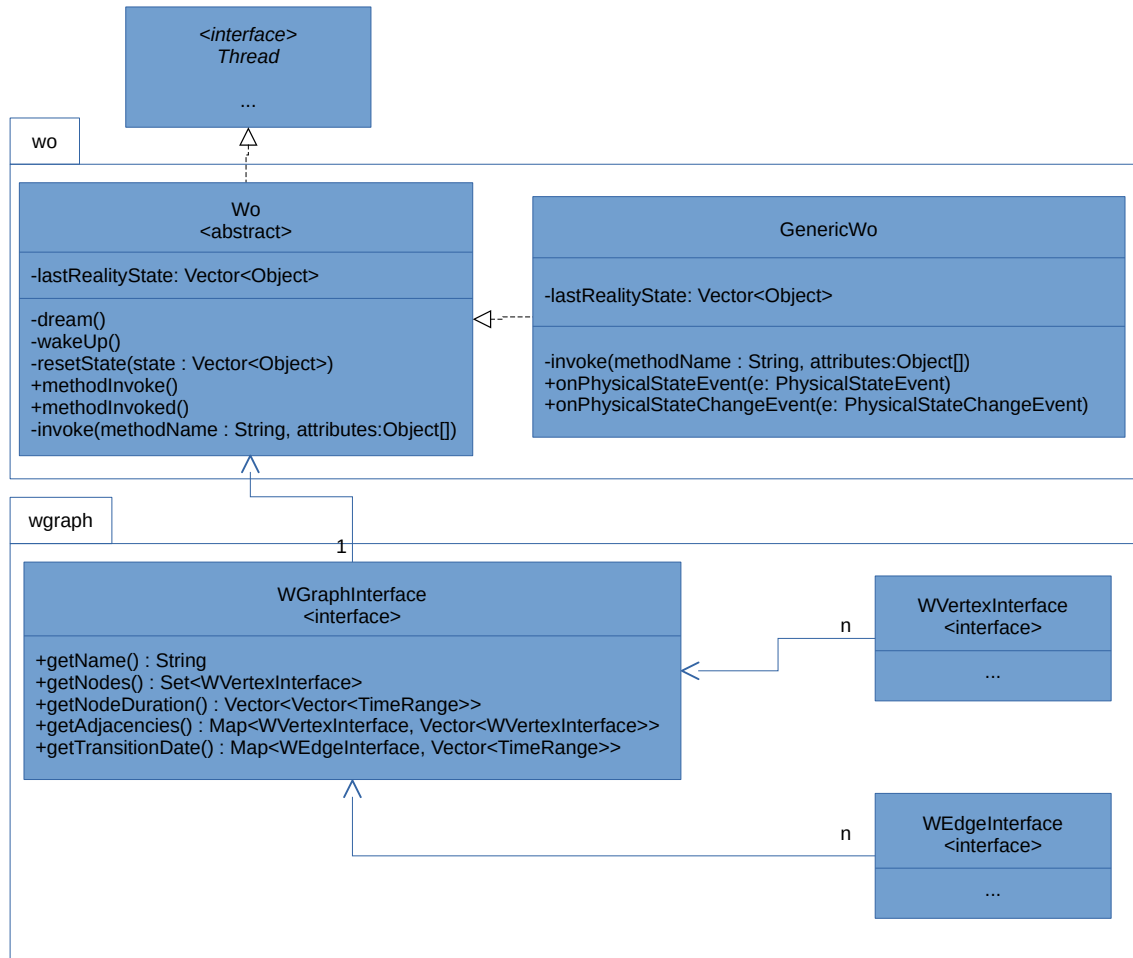


FIGURE 16 – Diagramme de classe UML d'un WO [Alloui et al., 2018].

deux objets physiques, un interrupteur et un volet, que nous appelons « :Acteur ». Quelques libertés de notations ont été prises afin de mettre en évidence certains points importants, notamment l'interaction entre un objet et sa partie sage et le fait que le système puisse agir sur ses acteurs. Ainsi, nous avons

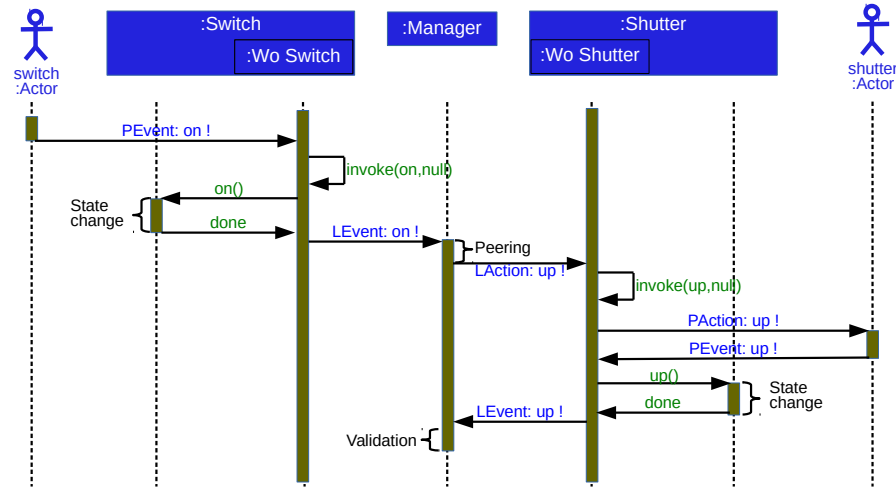


FIGURE 17 – Diagramme de séquence UML illustrant l'interconnexion entre deux objets physiques [Al-loui et al., 2018].

fait le choix de noter au sein du même objet « :Switch » la représentation de l'objet de classe Switch et « :WoSwitch » la représentation de la partie WO – partie héritée – de l'objet de classe Switch. Nous avons également opté pour les notations suivantes :

- « PEvent : foo ! » : est un événement provenant du système physique (externe à l'application) indiquant que l'action foo a été réalisée,
- « LEvent : foo ! » : est un événement provenant du système logique (interne à l'application) indiquant que l'action foo a été réalisée,
- « PAction : foo ! » : est un ordre envoyé au système physique (externe à l'application) demandant la réalisation de l'action foo,
- « LAction : foo ! » : est un ordre envoyé au système logique (interne à l'application) demandant la réalisation de l'action foo
- « foo() » : est l'invocation de la méthode foo.

Nous remarquerons, comme indiqué précédemment, que toutes les interactions entre WO sont événementielles et passent par un manager qui a en charge de vérifier la bonne réaction des WO.

Notons que de manière générale l'acteur initial – l'interrupteur – et final – le volet – peuvent tout à fait être virtuels, c'est-à-dire être une partie de l'application ou un autre programme. Pour aller plus loin, l'un, ou les deux acteurs peuvent ne pas exister ou agir directement sur le code métier de l'interrupteur ou du volet. Ces variantes et leurs combinaisons sont toutes prises en compte dans le « framework » afin de ne pas limiter par l'utilisation du « framework » les approches conceptuelles classiques.

6.2.2 Infrastructure du traitement de la sagesse

L'une des problématiques que nous souhaitons étudier est la représentation des habitudes comportementales et la détection de l'inhabituel. Il existe un grand nombre d'approches permettant la modélisation du comportement, celles basées sur les logiques, d'autres sur les probabilités, possibilités, ou encore celles basées sur les réseaux de neurones. Afin de ne pas nous limiter dans une seule approche, le WOF fournit un système de « pluggin » permettant la connexion d'un ou plusieurs analyseurs de « log » réalisant la détection de l'inhabituel avec l'une ou toute autre approche suscitée ou non. Hormis la flexibilité, ce concept permet la comparaison de plusieurs analyseurs pour une même exécution.

D'un point de vue fonctionnel, chaque WO lance ses analyseurs lorsqu'il passe en phase de rêve et les coupe lorsqu'il est sollicité pour rendre un service à un tiers. Lors de son exécution, l'analyseur a accès à tous les « log » de son WO au travers du graphe états/transitions du WO qui stocke les dates d'occurrences de chaque événement. Afin de mettre en place ce système, le mécanisme de « pluggin » a été réalisé comme illustré par le diagramme UML figure 18 . Notons tout d'abord que la classe WO

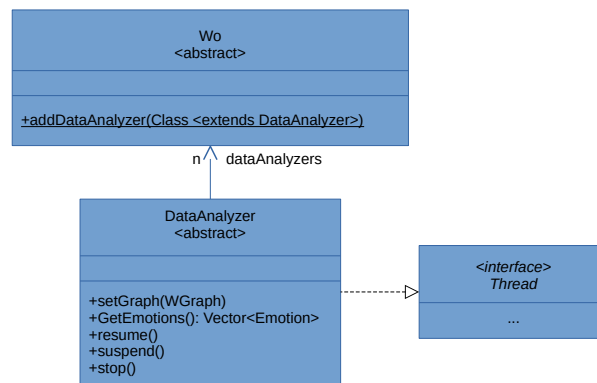


FIGURE 18 – Diagramme UML du système de « pluggin » des analyseurs de comportement d'un WO.

permet de manière statique, par une méthode de classe, d'enregistrer des classes d'analyseurs de « log ». Ainsi chaque WO du système possède les mêmes classes d'analyseurs. Ce qui est plus commode pour une première approche, mais qui pourrait, par la suite, être différencié afin de prendre en compte la puissance de calcul disponible pour chaque WO, notamment dans une approche distribuée du WOF. Bien qu'un WO possède une liste d'analyseurs, la méthode permettant de les ajouter ne prend pas en paramètre un objet de classe « DataAnalyseur », mais bien une classe héritant de « DataAnalyseur ». Ceci vient du fait que l'instanciation d'un WO s'effectue au niveau de la couche métier de l'application alors que l'analyse est au niveau système. La partie système du WO – sa couche sage – doit donc instancier elle-même ses analyseurs afin que le développeur métier n'ait pas cette tâche à gérer, tâche en l'occurrence propre au système et non au métier. Pour finir, un objet de classe « DataAnalyseur » est une instance active – un thread – qui doit pouvoir être suspendu et réactivé, il pourra également être interrogé pour fournir sa vision de l'état du système au travers d'une représentation que nous nommons « sensation ».

Ces travaux ont débuté en 2017, et sont toujours en cours, notamment pour la conception de nouveaux analyseurs, mais également pour une définition plus formelle de la notion d'« sensation » – un « feed-back » sur l'état du WO – et la mise en place de « réactions ». Ces sensations devant permettre au WO concerné de faire remonter à son système un comportement inhabituel, voir déclencher une

réaction afin de revenir dans une situation habituelle ou la plus appropriée au contexte.

6.3 Simulation de comportements

L'acquisition de données réelles, concernant l'utilisation d'un système, n'est pas toujours aisée et nécessite un temps d'utilisation du système long afin d'avoir un ensemble de données représentatif de la manière dont il est utilisé. De plus, la reproductibilité de ce comportement, dans un cadre d'utilisation réelle, n'est pas stable, les utilisateurs auront toujours quelques variations comportementales qui peuvent perturber l'étude et l'analyse de ces comportements. Enfin, la capacité de rejouer un comportement déjà vécu par le WO lui-même lui permettra de se focaliser sur les comportements habituels et leurs variantes et non plus de chercher à avoir un graphe états/transitions complet.

Afin de pallier ce problème, nous avons modélisé et implémenté un simulateur, il permet de simuler des comportements ou d'en rejouer des réels enregistrés. Le choix de ne pas nous appuyer sur un Simulateur à Événement Discret (SED) [Zeigler et al., 2000] existant est que ce dernier doit pouvoir s'intégrer au cœur des WO dans leurs phases de rêve et qu'à notre connaissance il n'en existait pas répondant à notre problématique de rejouer une exécution code Java. Ce simulateur, que nous qualifions de « Simulateur à Invocation Discrète (SID) » se base sur les concepts des SED). La différence notoire entre ces deux systèmes est qu'un SED génère des événements selon des lois de probabilité, ces derniers sont stockés dans des files d'attente puis traités par différents serveurs ; dans le cas de SID, des méthodes d'objets du système sont directement invoquées selon des lois de probabilité. Notons cependant, que la nuance est conceptuelle, l'un pouvant implémenter l'autre et réciproquement. Avec ce simulateur, nous pouvons avoir une vision globale de l'environnement structurel des WO, la figure 19 illustre ce dernier. La conception du SID se décompose en trois niveaux présentés ci-après : la gestion du temps, la gestion des actions et pour finir le simulateur à proprement parler.

6.3.1 Timer

La gestion du temps, appelée « Timer », permet la génération des laps de temps à attendre avant une invocation. Cette conception est générique et se base sur un patron composite, ce qui permet d'enrichir les méthodes de génération de laps de temps. La figure 20 illustre cette conception. Un « Timer » est une interface qui implémente un « Iterator ». Ainsi le « Timer » fournit, au travers de la méthode « next() », le prochain laps de temps à attendre en fonction de sa définition. Plusieurs implémentations de « Timer » ont déjà été réalisées.

- « Random » est un générateur de temps qui se base sur des lois de probabilité : Poisson, Gaussienne, Exponentielle ou uniforme. D'autres pouvant être aisément ajoutées.
- « Periodic » est un générateur de temps qui se base sur une période fixe, avec la possibilité d'un démarrage différé défini par l'attribut « at ».
- « OneShot » est un générateur de temps qui ne donne qu'une seule valeur, puis s'arrête.
- « Date » fournit des laps de temps définis à partir d'une liste de date, il permet notamment de rejouer l'enregistrement d'un comportement réel pour lequel les dates des actions ont été enregistrées.

Notre modèle présente également trois compositions possibles de « Timer ». Un « Bounded » limite la portée d'un « Timer » entre une date de début s et une date de fin e . Pour ce faire, un « Bounded » t_b encapsule un « Timer » existant t_w . Le premier temps Δt_b^0 fournit par t_b est la somme des premiers temps fournis par t_w telle que Δt_b^0 soit la plus petite valeur supérieure à s et inférieure à

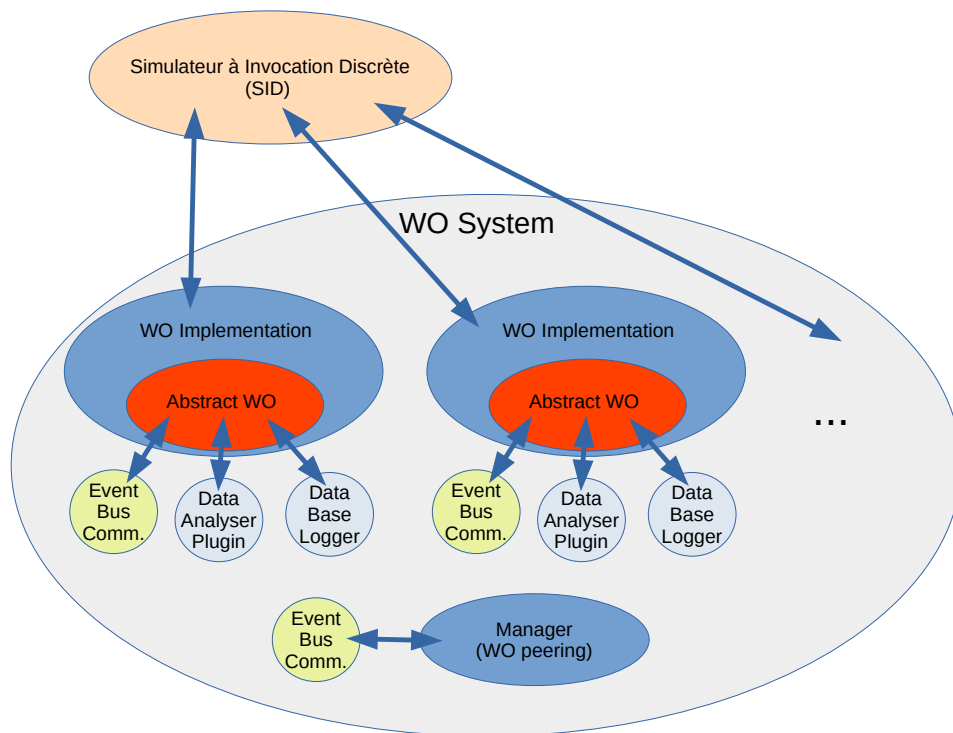


FIGURE 19 – Ensemble des outils permettant le déploiement et le test d'un système sage.

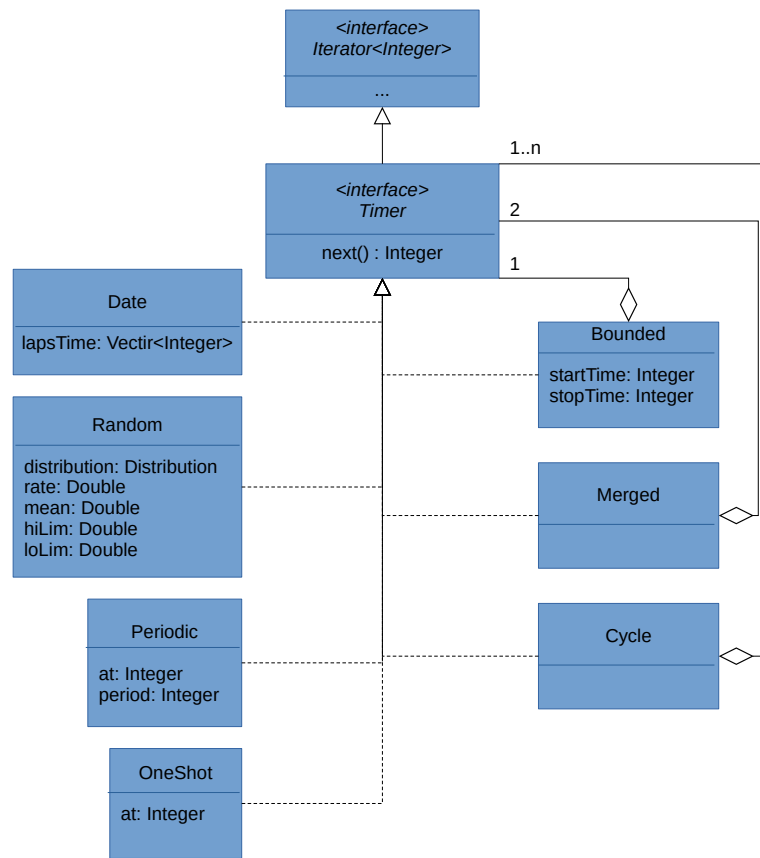


FIGURE 20 – Diagramme UML modélisant la gestion du temps.

e. Il est défini par :

$$\Delta t_b^0 = \sum_{i=0}^{n_{min}} \Delta t_w^i,$$

où

$$n_{min} = \operatorname{argmin}_n \left(\sum_{i=0}^n \Delta t_w^i \mid s \leq \sum_{i=0}^n \Delta t_w^i \leq e \right).$$

Le dernier temps Δt_b^l fourni par t_b est la somme des premiers temps fournis par t_w telle que Δt_b^l soit la plus grande valeur supérieure à s et inférieure à e . Il est défini par :

$$\Delta t_b^l = \sum_{i=0}^{n_{max}} \Delta t_w^i,$$

où

$$n_{max} = \operatorname{argmax}_n \left(\sum_{i=0}^n \Delta t_w^i \mid s \leq \sum_{i=0}^n \Delta t_w^i \leq e \right).$$

Un « Merged » t_m est défini à partir de deux « Timer » existants, t_1 et t_2 . Il fournit à chaque étape la somme des temps fournis par t_1 et t_2 pour l'étape correspondante :

$$\Delta t_m^i = \Delta t_1^i + \Delta t_2^i.$$

Pour finir, un « Cycle » est défini par une liste de « Timer » existants et fournit le temps de chaque « Timer » les uns après les autres et recommence au début de la liste après avoir fourni le temps du dernier « Timer » de la liste :

$$\Delta t_s^i = \Delta t_{i \bmod n}^k$$

où « mod » est l'opérateur de modulo, n le nombre de « Timer » de la liste et k est donné par :

$$k = \left\lfloor \frac{i}{n} \right\rfloor.$$

En d'autres termes, les n premiers temps Δt_s^i sont les premiers temps Δt_i^0 des n « Timer » de la liste, les n Δt_s^i suivants sont les seconds temps Δt_j^1 des « Timer » de la liste, etc.

Les « Timer » composés permettent de modéliser un certain nombre de comportements humains en intégrant de la variabilité. Un « Merged » peut combiner un « Timer » périodique avec un autre aléatoire afin de représenter un comportement humain périodique tout en intégrant de légères variations temporelles, par exemple une action effectuée tous les jours à la même heure à quelques minutes près.

6.3.2 Action

En nous appuyant sur le modèle précédent qui permet la gestion du temps, nous pouvons modéliser les actions à réaliser en fonction du temps. Comme pour les « Timer », nous nous basons sur un patron composite pour modéliser les actions, voir figure 21. L'action de base, appelée « DiscretAction », implémente l'interface « DiscretActionInterface » qui définit les caractéristiques communes des actions. C'est à dire la possibilité d'accéder à l'objet sur lequel est effectuée l'action, la méthode à appliquer à l'objet et le laps de temps d'attente avant d'appliquer la méthode. La méthode « next() » quant à elle permet d'obtenir l'action. Dans le cas le plus simple, seule l'information de « Timer » est mise à jour

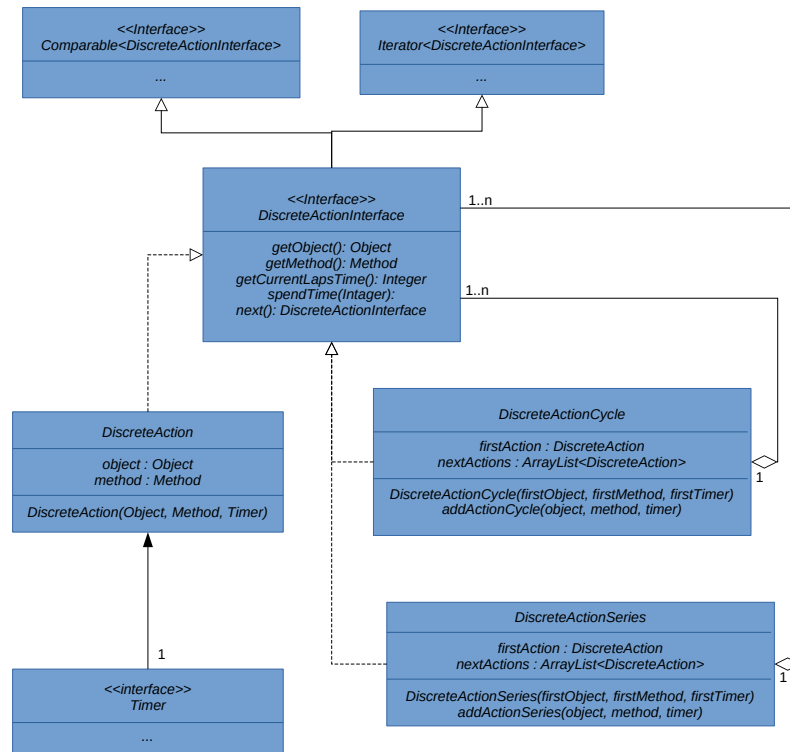


FIGURE 21 – Diagramme UML modélisant la gestion des actions.

à chaque itération, l'objet et la méthode ne changent pas au cours du temps, dans les cas composés, l'objet et la méthode peuvent changer à chaque itération.

Nous remarquerons que dans notre patron, il n'existe qu'une seule action élémentaire – « DiscretAction » – qui est définie à partir d'un seul objet et une seule méthode. Les deux compositions présentées dans la figure 21 permettent de créer des dépendances entre actions à réaliser. « DiscretActionCycle » cycle sur une liste d'actions. Elle réalise les actions les unes après les autres, puis recommence à la première lorsqu'elle arrive à la fin de la liste. Un exemple simple de cycle d'actions est par exemple une action d'allumage suivi d'une action d'extinction, elles se suivent toujours l'une après l'autre. « DiscretActionSeries » exécute la première action jusqu'à ce que son « Timer » se termine, c'est-à-dire qu'il ne donne plus de valeur, avant de passer à la seconde action de la série.

Notons que les actions, selon la figure 21, doivent être comparables. Elles le sont par rapport à leur temps d'attente avant exécution. Ceci est indispensable au simulateur pour qu'il puisse les ordonner et les exécuter dans le bon ordre.

6.3.3 Simulateur

Le dernier niveau est la simulation à proprement parler. Le rôle du simulateur est donc de gérer une liste d'actions à réaliser. La figure 22 présente notre conception. Le simulateur est une entité active

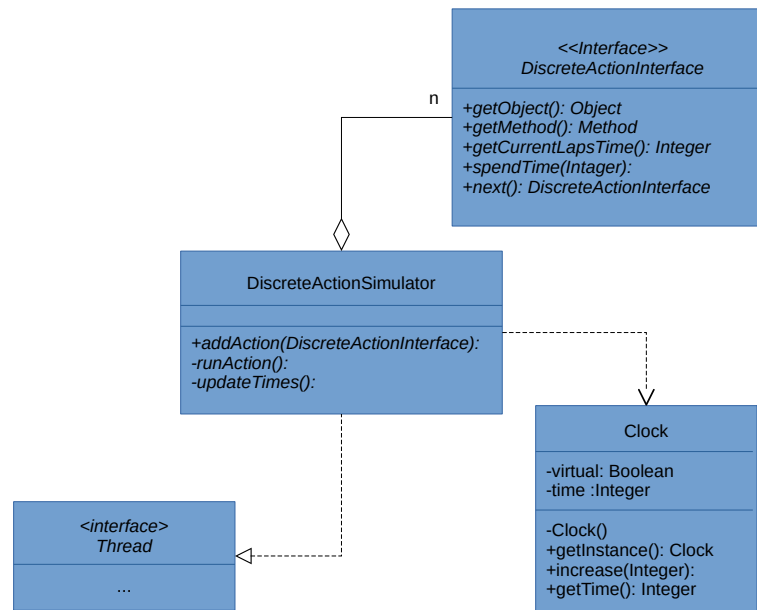


FIGURE 22 – Diagramme UML du simulateur.

– un thread – qui gère à la fois une liste ordonnée d'action à réaliser et une horloge afin de simuler un temps virtuel. Continuellement, le simulateur effectue les opérations suivantes :

- extraire la première action de la liste (celle qui a le plus petit temps d'attente),
- incrémenter l'horloge avec le temps d'attente Δ_t de l'action,

- décrémenter le temps d'attente de toutes les autres actions de la liste du temps Δ_t ,
- exécuter l'action,
- réinjecter l'action dans la liste après sa mise à jour.

Notons que, dans cette version, nous ne prenons pas en compte le temps d'exécution de la méthode au sein de l'horloge, nous considérons qu'il est négligeable. Ceci est un point d'amélioration futur de notre simulateur.

Pour terminer cette présentation du simulateur, comme nous pouvons le remarquer au travers des différents diagrammes, ce dernier n'est pas dédié aux WO et peut être utilisé avec n'importe quel système implémenté en Java. Outre l'utilisation que nous en faisons, l'injection d'un comportement humain afin de stimuler notre système et analyser ses réactions, le simulateur est parfaitement adapté au test logiciel. En effet, il permet de rejouer des scénarios comportementaux réels et tester un fonctionnement spécifié en phase de développement ou garantir la non-régression d'un système en phase d'évolution. Outre l'aspect simulation, l'objectif de ce SED est d'être intégré au WO afin de rejouer, pendant la phase de rêve, des séquences d'utilisations logguées et permettre au WO d'apprendre sur des variantes dans ces séquences enregistrées.

6.4 Modélisation de l'habitude et détection de l'inhabituel

La simulation nous permet d'exécuter et reproduire des comportements d'utilisation du système. Cependant, le système lui-même ne connaît pas a priori ces comportements, il doit les apprendre. Pour ce faire, il est dans l'obligation de mémoriser l'ensemble de ses événements et les analyser afin de modéliser les habitudes des utilisateurs.

6.4.1 Modélisation de la mémoire

Déterminer ce qui est habituel afin de détecter l'inhabituel nécessite une analyse du passé et par conséquent une connaissance de ce passé : la mémoire. Dans le cadre de la thèse d'Étienne Mauffret, nous nous sommes intéressés à ce problème et présentons la description générique suivante.

Soit une mémoire M et sa fonction d'effacement f_M (fonction d'oubli). Un événement e est défini par une action effectuée a , sa source s et l'instant t auquel il est survenu. Son âge Δ_t (laps de temps depuis la survenue de l'événement) étant défini par $\Delta_t = \text{now} - t$, où now est l'instant courant. La fonction $f_M : e \times t \rightarrow [0, 1]$ quant à elle, définit le poids des événements passés, par conséquent, le type de mémoire utilisé.

Nous présentons ainsi trois types de mémoires classiques que nous définissons par :

- \mathcal{M}_{CM} : Une mémoire intemporelle qui utilise tous les événements passés pour ses estimations.
- \mathcal{M}_{PM} : Une mémoire récente qui n'a accès qu'aux derniers événements survenus.
- \mathcal{M}_{FM} : Une mémoire évanescente qui applique un poids aux événements passés [Volterra, 1959, Wiener, 1958, Boyd and Chua, 1985].

Mémoire intemporelle Afin d'être capable de manipuler tous les événements de son historique lors de ses estimations, \mathcal{M}_{CM} utilise la fonction d'effacement constante $f_{\mathcal{M}_{CM}}(e, t) = 1$. La fonction étant constante, un événement ayant eu lieu au début de l'exécution aura le même poids qu'un événement récent. Ainsi, cette gestion de la mémoire permet à \mathcal{M}_{CM} d'être robuste face aux irrégularités temporaires : les événements exceptionnels. En effet, si les sources sont suffisamment actives en temps normal, ces événements occasionnels seront négligeables lors de l'utilisation de \mathcal{M}_{CM} .

Cette robustesse rend cependant \mathcal{M}_{CM} peu performante pour la gestion de changements d'habitudes. Si la fréquence ou la suite d'événements change dans le temps, les premières occurrences de la suite d'événements ayant autant de poids que les récentes, la prise en compte de changements est minimisée par le poids du passé. L'utilisation d'une mémoire intemporelle apparaît alors comme adaptée aux systèmes stables, dont les habitudes ne changent pas.

Mémoire récente Le principe de la mémoire récente consiste à complètement oublier les événements les plus anciens. Pour ce faire, 2 approches sont possibles, soit l'utilisation d'une fenêtre spatiale qui ne conserve que les N derniers événements, soit une fenêtre temporelle qui ne conserve que les événements dont l'âge Δ_t est inférieur à un seuil τ . Le poids de e à l'instant $t - f_{\mathcal{M}_{PM}}$ – peut alors être défini par :

$$f_{\mathcal{M}_{PM}}(e, t) = \begin{cases} 1, & \text{si } \Delta_t \leq \tau \\ 0 & \text{si } \Delta_t > \tau. \end{cases}$$

Un événement e tel que $\Delta_t > \tau$ n'est plus pris en compte, et supprimé de \mathcal{M}_{PM} . L'utilisation de \mathcal{M}_{PM} permet donc d'oublier un passé trop ancien et ainsi de s'adapter aux changements d'habitudes qui peuvent apparaître.

La définition de τ ou N est critique dans la définition de \mathcal{M}_{PM} . Avec des valeurs trop petites, il serait impossible de détecter des habitudes, avec des valeurs trop grandes nous nous rapprochons d'une mémoire intemporelle et des problèmes qu'elle engendre. La mémoire récente est donc adaptée aux systèmes dont le comportement change.

Mémoire évanescente Un compromis entre la mémoire intemporelle et la mémoire récente est la mémoire évanescente. Le principe de ce type de mémoire est d'« oublier » progressivement les événements en réduisant leur importance avec le temps. Il existe une infinité de fonctions d'effacements, à partir du moment où une fonction respecte les critères suivants [Boyd and Chua, 1985] :

- être définies par morceaux sur l'ensemble du temps de l'exécution ;
- être décroissantes ;
- ne pas être constantes sur leur ensemble ;
- être définies sur $]0, 1]$,

c'est une fonction d'effacement valide. Ces conditions permettent à $f_{\mathcal{M}_{FM}}$ de fournir une valeur pour tous les événements passés tout en assurant que ceux-ci seront moins impactant avec le temps sans être totalement oubliés. Il est possible d'utiliser une fonction continue ou définie par morceaux. Par exemple, la fonction linéaire $f_{\mathcal{M}_{FM}}(e, t) = 1 - \frac{\Delta_t}{T}$ qui donne un poids inversement proportionnel à l'âge, ou celle par morceau $f_{\mathcal{M}_{FM}}(e, t) = \frac{1}{2} \lfloor \frac{t}{\tau} \rfloor$, après chaque période τ , l'impact des événements est divisé par deux.

Avec la mémoire évanescente, la définition de $f_{\mathcal{M}}$ prend tout son sens et permet d'adapter la mémoire à l'utilisation qui en est faite. Toute la complexité de ce type de mémoire tient donc dans la définition de $f_{\mathcal{M}}$ à la fois en temps de traitement et en qualité de l'information conservée.

Enfin, il est possible de considérer d'autres types de mémoire plus complexe, comme une mémoire évanescente récente ou la mémoire à recouvrement. Cette dernière ressemble aux mémoires évanescentes, mais leur fonction d'effacement peut être ponctuellement croissante. Elles sont adaptées à la détection d'habitudes récurrentes : lorsqu'un nouvel événement ou une séquence ressemble suffisamment à un ancien, alors $f_{\mathcal{S}_{FM}}(t)$ s'adapte pour anticiper une habitude connue.

6.4.2 Mesure de la distance à l'habituel

L'une des contraintes de la réalisation d'un analyseur de comportement avec le WOF est que l'analyseur doit avoir une réactivité quasi temps réel. Nous avons donc, pour le premier analyseur, décidé de l'implémenter en nous basant sur le principe statistique de stationnarité, et plus particulièrement sur la définition de la stationnarité faible : « weak-sense stationarity » (WSS). L'idée étant qu'un comportement stationnaire est habituel et vice versa [Alloui and Vernier, 2017, Alloui and Vernier, 2018]. Dans le domaine de la détection d'anomalie dans les séries temporelles, cela s'apparente à une méthode mu-sigma [Mason et al., 2019] que nous avons adaptée à notre problématique. L'avantage de cette approche, d'un point de vue calculatoire, est qu'elle se calcule facilement en flux de données, et, par conséquent, répond bien à la problématique de temps réel. Une comparaison de différentes méthodes de détection d'anomalies, qui pourraient être utilisées comme modèles d'habitudes sont présentées dans [Mason et al., 2019], cette comparaison met notamment en évidence, l'avantage de mu-sigma pour le temps réel.

Les comportements que nous étudions étant impactés par l'humain, ils ne peuvent jamais être parfaitement stationnaires, nous avons donc adapté WSS. Soit $x(i)$ un processus aléatoire, ce processus est un WSS processus si et seulement si :

$$\begin{aligned} E[x(i)] &= \mu \quad \forall i, \\ Var[x(i)] &= \sigma^2 \neq \infty \quad \forall i, \\ Cov[x(i), x(i-k)] &= f(k) = \rho_k \quad \forall i \forall k, \end{aligned}$$

une moyenne constante, une variance constante et différente de l'infini et une covariance constante pour un k donné.

Comme les habitudes peuvent changer au cours du temps, et que nous souhaitons le prendre en compte, la stationnarité est calculée sur une fenêtre glissante – mémoire récente – de taille w :

$$\begin{aligned} E[x(i)] &= \mu(t) \quad \forall i \in [t-w, t], \\ Var[x(i)] &= \sigma^2(t) \neq \infty \quad \forall i \in [t-w, t], \\ Cov[x(i), x(i-k)] &= f(k, t) = \rho_k(t) \forall i \in [t-w, t] \forall k, \end{aligned}$$

où la série temporelle X^t est définie par les temps inter-événement $x(i) = \Delta_i - \Delta_{i-1}$ tels que i soit compris entre $t-w$ et t , pour la suite d'événements $[e_{\tau}^{t-w} \dots e_{\tau}^i \dots e_{\tau}^t]$, comme présenté dans la section sur les mémoires.

Comme précisé précédemment, le comportement étudié ne peut pas être parfaitement stationnaire. Afin de prendre en compte ce critère, nous avons défini que le système est considéré comme stationnaire si et seulement si, lorsqu'une nouvelle occurrence arrive à $t+1$, la nouvelle moyenne, variance et covariance prennent des valeurs connues dans la fenêtre mémoire :

$$\begin{aligned} \min([\mu(t-w), \mu(t)]) &\leq \mu(t+1) \leq \max([\mu(t-w), \mu(t)]), \\ \min([\sigma^2(t-w), \sigma^2(t)]) &\leq \sigma^2(t+1) \leq \max([\sigma^2(t-w), \sigma^2(t)]), \\ \min([\rho_k(t-w), \rho_k(t)]) &\leq \rho_k(t+1) \leq \max([\rho_k(t-w), \rho_k(t)]). \end{aligned}$$

En d'autres termes, si la moyenne, la variance et la covariance sont dans l'intervalle de valeurs précédemment connues dans la fenêtre de mémoire, le nouvel événement survenu à l'instant $t+1$ est considéré comme habituel, sinon il est inhabituel.

En nous appuyant sur cette définition, nous avons défini la notion de sensation comme la distance $d(x(t+1))$ entre les caractéristiques de stationnarité – moyenne, variance et covariance – normalisées

centrées en mémoire et celles du nouvel événement à l'instant $t + 1$ normalisé centré par les données en mémoire. Ainsi $d(x(t + 1))$ est défini par :

$$d(x(t + 1)) = \begin{cases} d(E[X^{t+1}]), \\ d(Var[X^{t+1}]), \\ d(Cov_k[X^{t+1}, X^{t+1}]) \quad \forall k \in [t - w, t], \end{cases}$$

où

$$d(E[X^{t+1}]) = \frac{\mu(t + 1) - \overline{\mu(i)}}{(\max(\mu(i)) - \min(\mu(i)))/2},$$

$$d(Var[X^{t+1}]) = \frac{\sigma^2(t + 1) - \overline{\sigma^2(i)}}{(\max(\sigma^2(i)) - \min(\sigma^2(i)))/2},$$

$$d(Cov_k[X^{t+1}, X^{t+1}]) = \frac{\rho_k(t + 1) - \overline{\rho_k(i)}}{(\max(\rho_k(i)) - \min(\rho_k(i)))/2},$$

avec $i \in [t - w, t]$, et $\overline{\mu(i)}$, $\overline{\sigma^2(i)}$ et $\overline{\rho_k(i)}$ respectivement les moyennes des moyennes, variances et covariances sur l'intervalle $[t - w, t]$.

Ainsi, quand un nouvel événement arrive, $d(x(t + 1))$ est recalculé. Si l'ensemble des valeurs de $d(x(t + 1))$ c'est à dire $d(E[X^{t+1}])$, $d(Var[X^{t+1}])$ et les $d(Cov[X^{t+1}, X^{t+1}])$, sont comprises entre -1 et 1, le comportement est habituel, sinon la valeur de $d(x(t + 1))$ supérieur à 1 ou inférieur à -1 donne une « force » de l'inhabituel détecté, par rapport à la connaissance, c'est-à-dire la mémoire.

Ces travaux sont préliminaires, mais donnent des résultats encourageants présentés dans la section suivante. La thèse d'Abdelhafid Dahhani, qui débute, traite de ces problématiques, notamment sur la partie conceptuelle des analyseurs et en particulier sur leur « feed-back » – la notion de sensation – et la fusion de ces dernières de manière hiérarchique dans le système.

6.5 Applications

6.5.1 Domotique : synthèse de contrôleurs

Bien qu'antérieurs à ces travaux, ceux réalisés dans le cadre de la thèse de Fabien Sartor [Sartor, 2012] sont une application directe des objets sages. Lors de ses travaux, Fabien Sartor a proposé un modèle de représentation du comportement des systèmes domotiques basé sur les IOSTS, ces derniers étant des automates (graphes) états/transitions étendus. Un IOSTS est un tuple $\langle D, \Theta, L, l_0, \Sigma, T \rangle$ tel que :

- $D = V \cup P$ est un ensemble fini de données, composé d'un ensemble fini V de variables et d'un ensemble fini P de paramètres. Soit un élément $d \in D$, $\text{Dom}(d)$ détermine le domaine de valeur de d . Si $e \in \text{Dom}(d)$, alors e est une valeur compatible avec d .
- Θ est l'ensemble des conditions initiales, c'est un prédicat sur V .
- L est un ensemble fini non vide de localités et $l_0 \in L$ est la localité initiale. Une localité est un ensemble d'états $l \subseteq \text{Dom}(V)$.
- Σ est l'alphabet, un ensemble fini et non vide d'actions. Il est composé de l'union disjointe de l'ensemble $\Sigma^?$ d'actions d'entrée, de l'ensemble $\Sigma^!$ d'actions de sortie et de l'ensemble Σ^τ d'actions internes. Pour chaque action $a \in \Sigma$, sa signature $\text{sig}(a) = \langle p_1, \dots, p_k \rangle \in P^k (k \in \mathbb{N})$ est un tuple de paramètres. La signature des actions internes est toujours un tuple vide.

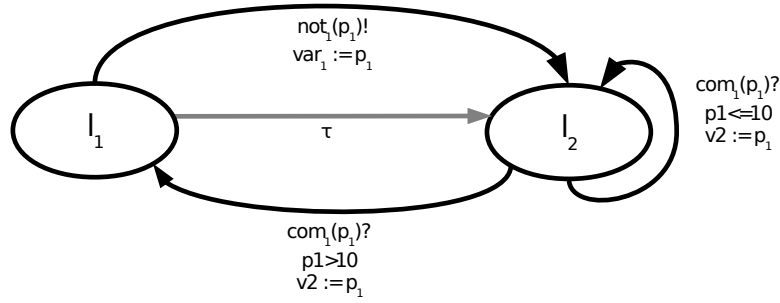


FIGURE 23 – Représentation graphique d'un IOSTS.

- T est un ensemble de transitions fini, tel que chaque transition est un tuple $t = \langle l^o, a, G, A, l^d \rangle$ défini par :
 - une localité $l^o \in L$, appelée l'origine de la transition,
 - une action $a \in \Sigma$, appelée l'action de la transition,
 - un prédicat G sur $V \cup sig(a)$, appelé la garde,
 - une fonction d'affectation $A : Dom(V) \mapsto Dom(V)$, qui est un ensemble d'expression de la forme $(x := A^x)_{x \in V}$ tel que, pour tout $x \in V$, $A^x \in V \cup sig(a)$,
 - une localité $l^d \in L$, appelée la destination de la transition.

La figure 23 en est une illustration graphique fictive. Elle modélise un système infini : il existe une infinité d'états représentés par les variables var_1 et var_2 puisque leur domaine est l'ensemble des réels, $type(var_{\{1,2\}}) = \mathbb{R}$. Par rapport à cette approche, Fabien introduit la différenciation entre les états (localités) stables \underline{L} qui correspondent aux états réels du dispositif et les états (localités) virtuels qui correspondent aux états où le système attend une réponse d'un dispositif. L'état réel du dispositif physique n'étant pas connu du système. Fabien introduit ainsi le concept d'automate de comportement. Un automate de comportement \mathcal{S}_{AC} est un tuple $\langle D, \Theta, L_{AC}, L^0, \Sigma, T \rangle$ tel que :

- $L_{AC} = \underline{L}_{AC} \cup \overline{L}_{AC}^v$ est un ensemble fini non vide de localités et L^0 est l'ensemble des localités initiales $\underline{L}_{AC}^0 \subseteq \underline{L}_{AC}$ ($L_{AC}^0 = \underline{L}_{AC}$ dans la plupart des cas).
- $T = T^! \cup T^? \cup T^\tau$ est un ensemble de transitions. Chaque transition de $T^\#$ est un tuple $\langle l^o, a, G, A, l^d \rangle$ tel que $a \in \Sigma^\#$ avec $\# \in \{?, !, \tau\}$.

De plus, $\forall \langle l, a, G, A, l' \rangle$ tel que $a \in T^\tau, G = \top, A = \emptyset, l \in \overline{L}_{AC}^v, l' \in \underline{L}_{AC}$.

- \underline{L}_{AC} est l'ensemble non vide de localités stables tel que :

$$l \in \underline{L}_{AC} \iff \forall \langle l', a, G, A, l \rangle \in T, a \in \Sigma^! \cup \Sigma^\tau$$

- \overline{L}_{AC}^v est l'ensemble des localités virtuelles tel que :

$$l^v \in \overline{L}_{AC}^v \iff \begin{cases} \exists ! \langle l^o, a, G, A, l^v \rangle \in T, \text{ avec } a \in \Sigma^? \\ \exists ! \langle l^v, a, \top, \emptyset, l^o \rangle \in T, \text{ avec } a \in \Sigma^\tau \\ \exists ! \langle l^v, a, G', A', l^d \rangle \in T, \text{ avec } a \in \Sigma^! \end{cases}$$

La figure 24 représente un automate de comportement d'un volet physique, du point de vue du contrôleur (le système). Les événements « com(x)? » et « not(x)! » étant respectivement des commandes et notifications paramétrées, quant à τ , il représente un événement interne, la gestion d'un événement inattendu. Les états « Si » sont les états virtuels et les autres, « Fermé » et « Ouvert », sont les états stables. Ces automates de comportement sont utilisés afin de réaliser, de manière originale, de la synthèse de contrôleur, le contrôleur étant le manager dans le contexte de WO.

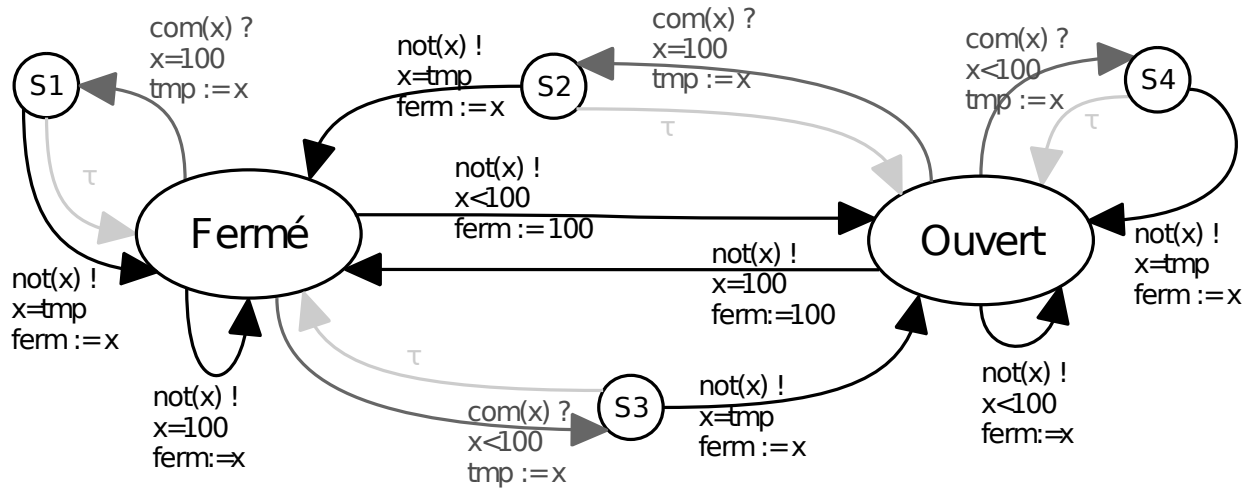


FIGURE 24 – Représentation graphique détaillée d'un automate de comportement d'un volet roulant.

En comparant les automates de comportement proposés dans la thèse de Fabien et les diagrammes états/transitions des WO, tous deux illustrés par les figures 24 et 25, nous remarquons que les automates de comportement ont une capacité descriptive plus fine et plus condensée à la fois. Ils permettent l'utilisation d'événements paramétrés, l'utilisation de valeurs réelles, modélisent l'inconnu, apportent un plus haut niveau sémantique et condensent les états. Cependant, cette approche oblige le développeur à fournir l'automate, avec tous les risques qu'il encourt : erreur de description, évolution du code désynchronisée de l'automate... Les WO apportent à ces travaux la capacité de fournir, pour l'heure, une base structurée sûre pour les IOSTS, le graphe états/transitions, puisqu'il est construit naturellement par le WO, sans action externe. Cependant, il ne porte pas de sémantique sur ses états, comme le montre la figure 25, et présente un niveau de granularité très fine, contrairement à un IOSTS équivalent proposé en figure 24 qui est nettement plus compressé et apporte un plus au niveau de sémantique. Le lien entre ces 2 représentations du comportement est donc une perspective de recherche, notamment en ce qui concerne leur alignement et le fait que plusieurs IOSTS sont possibles selon les points de vue, la figure 24 considère par exemple, d'un point de vue sémantique, le volet « partiellement ouvert » et non « partiellement fermé », puisque s'il n'est pas fermé à 100%, il est dans l'état « Ouvert ».

6.5.2 Domotique : Wise Internet of Things

Dans le cadre des WO, différentes expérimentations ont été développées lors de ces travaux de recherche. Afin d'illustrer les rendus d'un système sage, je présente ci-après deux des expérimentations réalisées.

La première expérimentation est basée sur la simulation d'une salle de classe. Elle prend en compte l'éclairage, la lumière et les volets roulants, chacun de ces éléments étant un composant IoT monitorable et contrôlable. La simulation est définie de la manière suivante, chaque jour :

- le chauffage se met automatiquement en route à 6h00 et se coupe à 18h00 ;
- à 8h00 (+5 min), le volet roulant est ouvert par l'enseignant ou les étudiants qui débutent le premier cours. Onze heures plus tard, vers 19h00 (+30 min), le gardien passe dans la salle et ferme le volet ;

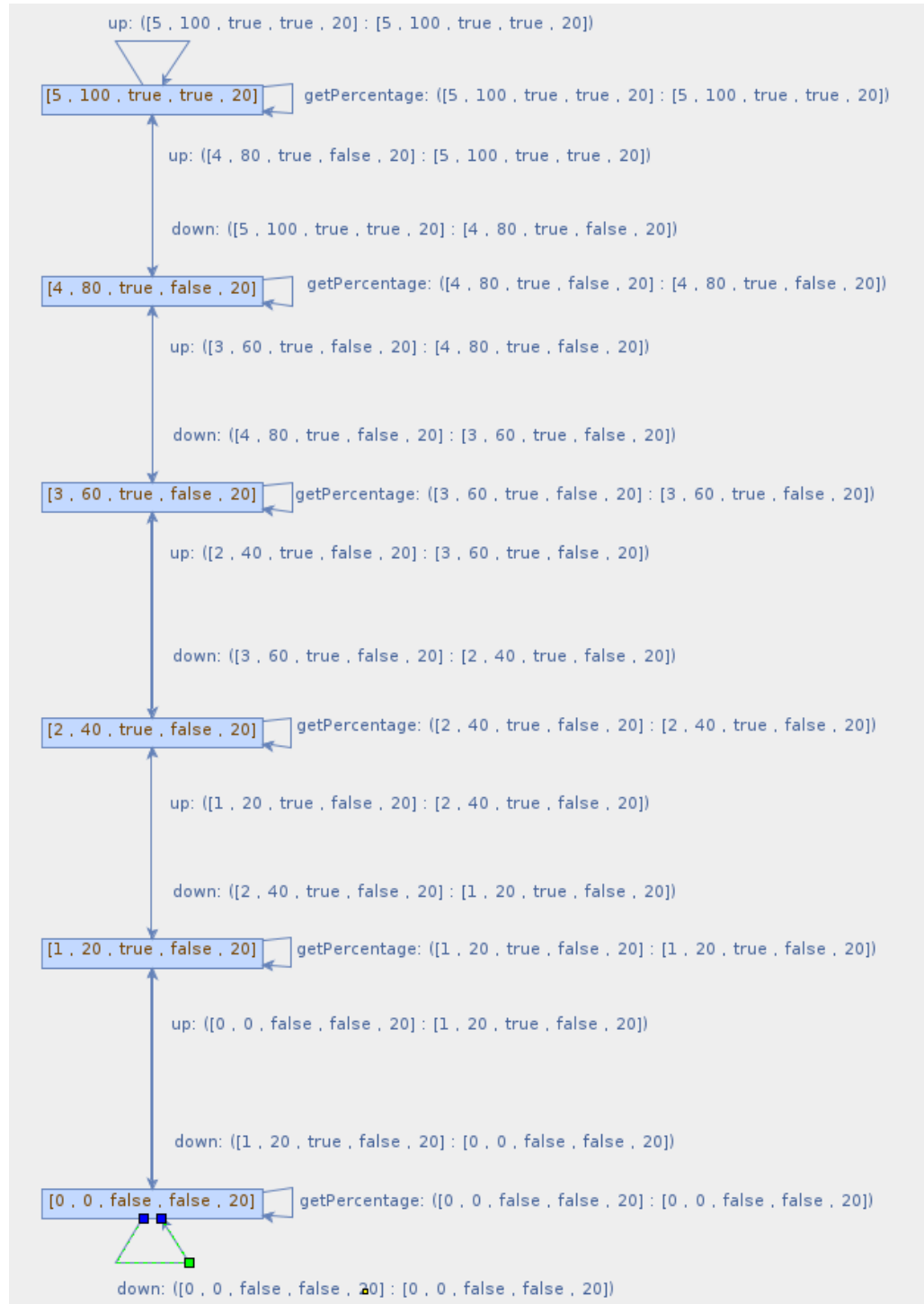


FIGURE 25 – Représentation graphique d'un diagramme états/transitions autoconstruit par le WO d'un volet roulant.

- la lumière de la salle est allumée à 8h00 (+5 min), au démarrage du premier cours et est éteinte à la fin de ce cours à 9h00 (+5 min).

Nous noterons que cette simulation est simplifiée, elle a pour but d'illustrer le fonctionnement du système et non de coller parfaitement à la réalité, l'autre expérimentation présentée a été faite dans ce sens.

Dans cette présentation, nous nous attacherons uniquement aux événements : délais d'interinstant d'allumage, d'extinction, d'ouverture ou fermeture. Sachant que nous pourrions également observer les temps de transitions et de passage dans les différents états. La figure 26 décrit un comportement habituel du système sans perturbation inhabituelle, en présentant uniquement les indicateurs de moyenne (bleu) et d'écart-type (rouge) précédemment présentés. Les valeurs d'habitudes étant normalisées centrées, tout comportement habituel est compris entre -1 et 1, lignes vertes, et tout comportement inhabituel est en dehors de cet intervalle. Les résultats sont obtenus avec une fenêtre mémoire w de 100 événements, ce qui correspond à 3 mois pour notre simulation.

Concernant le chauffage, complètement automatisé, son comportement est parfaitement stationnaire et n'a d'intérêt que pour référence. Pour chacune des courbes intégrant un comportement humain dans la figure 26, nous observons d'abord une phase d'apprentissage initial – sans aucune connaissance a priori – plutôt rapide, où tout nouvel événement est inhabituel. Par la suite, le comportement est rapidement considéré comme habituel à quelques comportements aléatoires près.

Afin de mettre en évidence l'apparition d'un comportement inhabituel, nous reproduisons l'expérimentation précédente en intégrant une coupure de courant à 8h30 après 150 jours d'exécution (21 mai). La figure 27 présente ces résultats. On observe clairement l'impact de la coupure sur l'éclairage et le chauffage, actif lors de la coupure. L'effet mémoire est également mis en évidence, dans un premier temps lors de la survenue de l'événement inhabituel et dans un second temps lors de son oubli. Ce phénomène est classique et bien connu et peut être pallié en utilisant une mémoire évanescence.

Une seconde réalisation plus réaliste a consisté à simuler un capteur de présence sage d'un amphithéâtre de l'IUT d'Annecy. Le comportement du capteur, au travers du simulateur, a été alimenté par des données réelles de présence de l'amphithéâtre en 2017. Ce dernier est appairé à l'éclairage de l'amphithéâtre afin d'en obtenir une gestion eco-responsable. La figure 28 illustre les mesures de l'habitude d'allumage (figure 28a) et d'extinction (figure 28b) de l'éclairage en utilisant fenêtre mémoire de 15 jours. Cette expérimentation met en évidence la phase d'apprentissage de 30 jours, soit 15 jours pour remplir la mémoire de données brutes plus 15 jours pour remplir les mémoires de moyenne, variance... Elle met également en évidence les comportements inhabituels ainsi que les changements d'habitudes. Selon son habitude, les WE apparaissent comme un événement inhabituel de faible intensité alors que les vacances sont des événements inhabituels de forte intensité. En fin d'année, lorsque l'amphithéâtre est moins utilisé, les WE et les vacances entrent dans un comportement habituel.

Ces expérimentations valident le fonctionnement primaire des objets sages et du WOF, en allant du simulateur au système de « plugin » d'analyse du comportement. Je parle ici de comportement primaire, car elle met également un certain nombre de lacunes : effet mémoire, multianalyses pour un même élément... Ces problèmes sont des axes de recherche pour de futurs travaux.

6.6 Perspectives

L'approche sage pour les systèmes logiciels – apprentissage sur leur propre fonctionnement par eux même – est une approche originale et innovante. Elle ouvre un grand nombre de perspectives scientifiques dans les domaines du monitoring, de la synthèse de contrôleurs, du test logiciel... Le

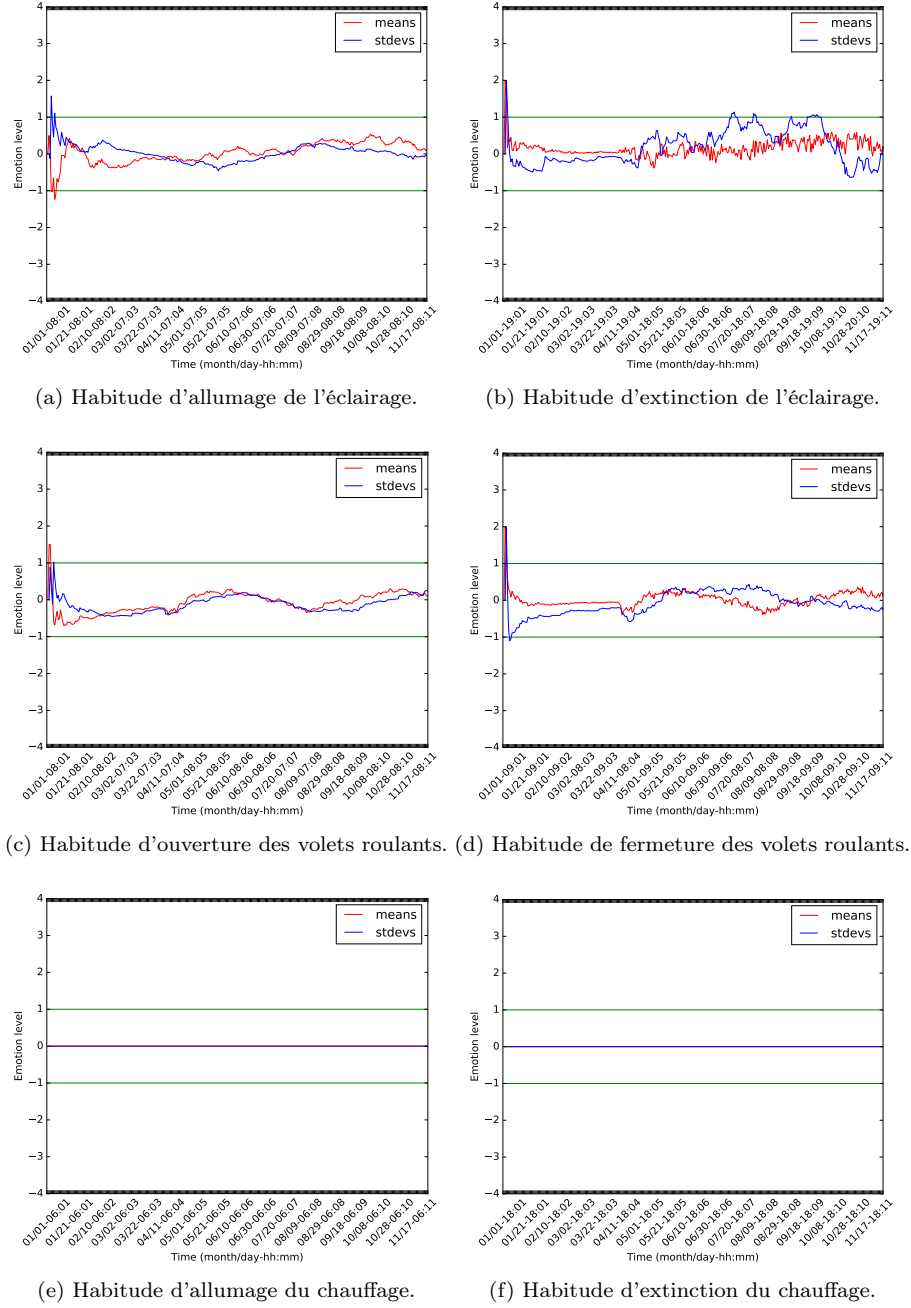


FIGURE 26 – Représentation de comportements habituels pour l'éclairage, les volets et le chauffage d'une pièce [Alloui et al., 2019].

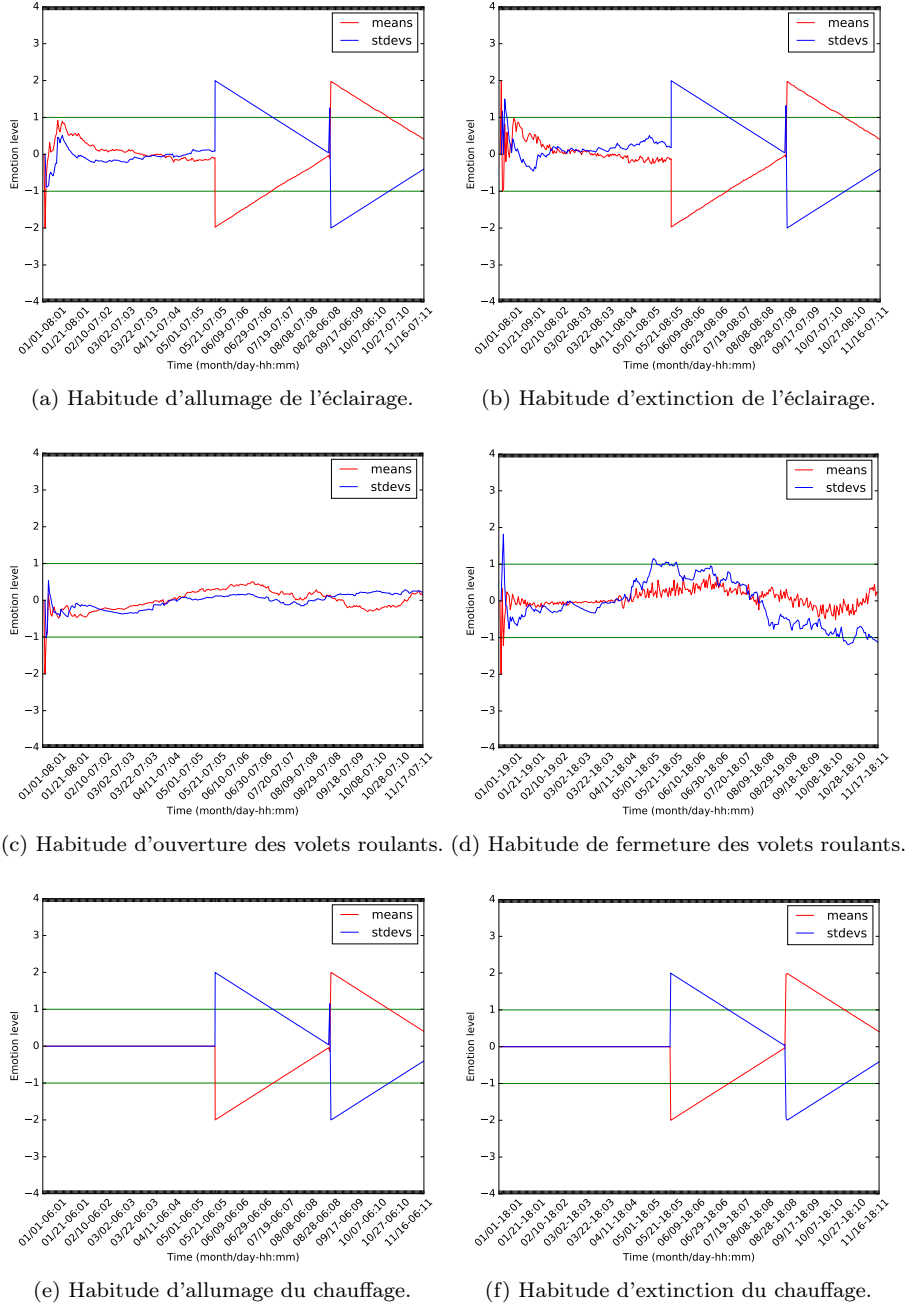
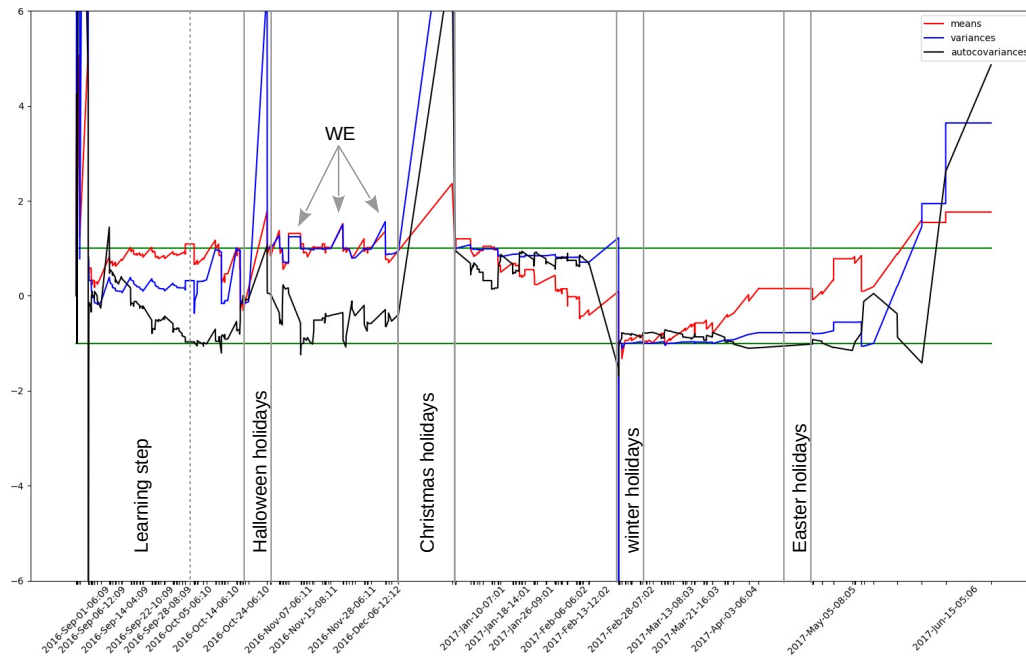
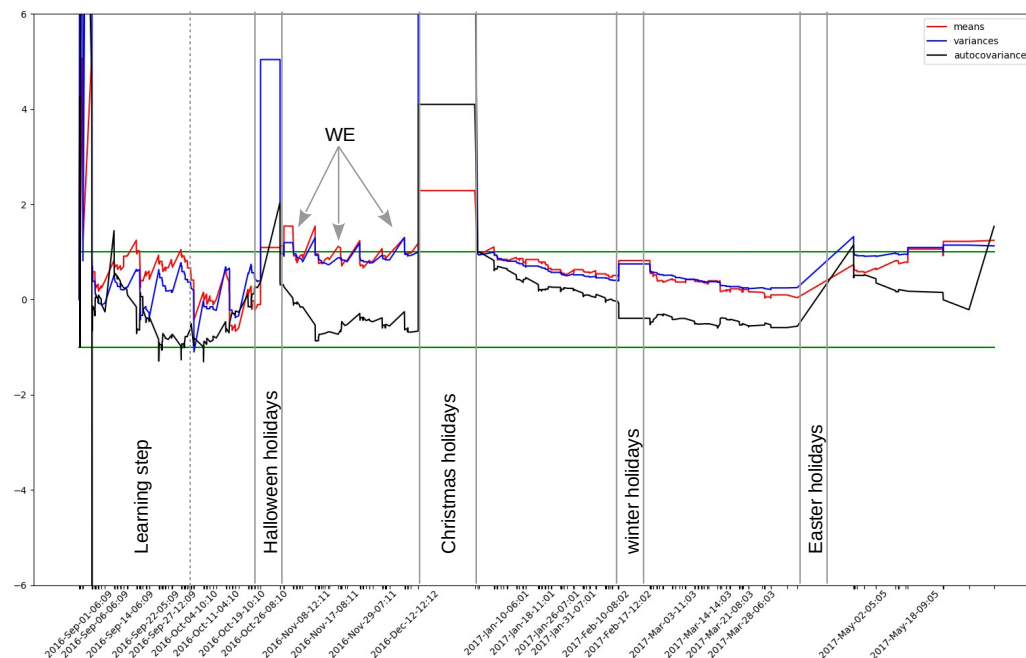


FIGURE 27 – Représentation de comportements habituels et détection d'événements inhabituels pour l'éclairage, les volets et le chauffage d'une pièce [Alloui et al., 2019].



(a) Habitude d'allumage.



(b) Habitude d'extinction.

FIGURE 28 – Représentation d'un comportement habituel et détection d'événements inhabituels d'un éclairage contrôlé par un capteur de présence sage [Alloui et al., 2019].

monitoring non supervisé étant la base des systèmes sages, leurs applications directes afin de suivre le comportement d'un système sont naturelles, la détection de comportements inhabituels étant l'un des objectifs du monitoring. La synthèse de contrôleurs consiste en la création d'un contrôleur à partir de règles de comportement ; les managers au sein du WOF remplissent ce rôle. Du fait de la forte capacité d'introspection et de réflexivité des systèmes sages, ils ont la capacité de s'exécuter eux-mêmes, et, dans le cadre du test logiciel, et plus particulièrement celui associé à l'IA, ils apportent une base pour analyser les habitudes comportementales de l'utilisateur face aux réactions des IA connectées au système. Au travers du simulateur, qui se base également sur les capacités d'introspection et de réflexivité, il est également possible de reproduire un comportement qui peut être une base de test pour garantir le bon comportement du système. Les WO apportent une base de connaissance complémentaire à celle qu'acquière classiquement un agent d'un Système Multi-Agents SMA qui apprend plus naturellement sur son environnement que sur lui-même, ces approches sont donc complémentaires et il est tout à fait pertinent de construire un SMA en s'appuyant sur le WOF. Ces perspectives sont une liste non exhaustive des ouvertures qu'offrent les approches sages.

Les perspectives que je vise plus particulièrement concernant les WO s'orientent plus vers la conception logicielle de systèmes distribués sages et leur autoadaptation. La conception de managers sages et l'association, composition ou agrégation, au sens UML, d'objets sages sont des points clés à étudier. Ceci nécessite des travaux d'approfondissement sur la modélisation de l'habitude et la détection de l'inhabituel, notamment en abordant des modélisations de type chaîne de Markov cachée ou non et réseaux de neurones. Concernant l'autoadaptation, l'approche initiale visera à faire réagir le système en cas de détection d'inhabitude afin de minimiser la distance entre son comportement courant et son comportement habituel.

La modélisation de l'habitude que nous avons envisagée est non sémantique, purement numérique. Ce qui est original dans les domaines d'applications visés, mais se justifie du fait que l'information est, pour l'instant, au plus bas dans l'architecture du système. Il faudra envisager au sein du système, dans les couches plus hautes, d'intégrer un système sémantique en gardant la même approche de généricité et d'évolutivité. Une première étape sera de lier ces travaux avec ceux réalisés dans le cadre de la thèse de Fabien Sartor où nous avons abordé le comportement à partir d'un plus haut niveau conceptuel, avec une connaissance a priori sur les graphes de comportement des éléments.

Cette approche conceptuelle des WO, intégrant l'observation indirecte du comportement humain sur le logiciel, dans le logiciel, apporte un indicateur dynamique et en temps réel sur ce dernier. Cet indicateur brut présente déjà un intérêt dans le suivi de cycle de vie du logiciel, notamment pour suivre les évolutions comportementales de l'utilisateur dans le temps et face aux évolutions du logiciel. L'apport d'un plus haut niveau sémantique permettra d'intégrer encore plus l'utilisateur au cycle de vie du logiciel en raison d'une interaction plus naturelle avec lui.

Ces travaux s'intègrent en partie dans la thèse d'Abdelhafid Dahhani qui débute, et dans le nouveau groupe de travail GL-IA du GDR GPL auquel nous participons.

Conclusion & Perspectives

Des chaînes de traitements synchrones de séries d'images, à la conception de chaînes de traitements événementiels de séries d'événements discrets, mes travaux de recherche m'ont mené à étudier les problématiques allant du traitement de série d'images acquises en milieu naturel à l'acquisition de connaissance sur les systèmes logiciels. Ces travaux pluridisciplinaires – géosciences, photogrammétrie, systèmes distribués, génie logiciel – ont été réalisés en collaboration avec différents laboratoires – ISTERre, EDYTEM, GIPSA, LIRIS – sur des thématiques variées. Ils m'ont permis de faire émerger une nouvelle thématique de recherche originale s'appuyant sur le concept « d'Objets Sages » (WO) ; concept original que j'ai introduit.

En ce qui concerne la conception de chaînes de traitements d'images, les travaux ont abouti à la proposition de plusieurs chaînes opérationnelles traitant la problématique générale sous différents points de vue. Au travers de l'encadrement de plusieurs thèses, les problématiques abordées vont de celles du nettoyage préalable des séries aux traitements à proprement parler. L'objectif opérationnel étant d'obtenir des champs de déplacements 3D de l'objet observé. Au travers de ces travaux, principalement les thèses de Ha Thai Pham, Haixing He et Hela Hadhri, nous avons proposé différentes approches originales pour le recalage de séries d'images, la sélection automatique des images valides – qui ne sont pas perturbées par des phénomènes climatiques – mais également pour la mesure de déplacement 3D métrique des scènes observées. L'ensemble de ces travaux ont une approche commune qui est l'utilisation de la redondance de mesure afin de minimiser l'erreur de mesure finale. Et nous avons, à la fois, abordé les problématiques algorithmiques et opérationnelles, notamment dans le cadre de la thèse CIFRE de Guilhem Marsy où, outre les apports algorithmiques originaux proposés, nous avons validé l'intérêt applicatif pour la société TENEVIA, partenaire de la thèse.

Lors de la mise en place de ces chaînes de traitements, et en particulier la mise en place de la redondance des traitements afin de minimiser les erreurs, une problématique récurrente apparaît. Celle de la masse de données à traiter et surtout celle du temps de traitement ; les algorithmes utilisés et proposés étant par nature chronophages. De cette problématique, nous proposons une solution opérationnelle simple se basant sur une description, en Python, du flux de traitements, cette description étant traduite par la suite en une description pour l'outil « GNU make » ; rappelons que ce dernier est, à la base, un ordonnanceur. Cette première proposition a un aspect très pragmatique et permet un déploiement simple et efficace afin d'utiliser de manière optimale des machines multicœurs ou des clusters de calcul pour exécuter les chaînes.

Néanmoins, la problématique du temps de traitement reste une problématique majeure de ces chaînes. Rappelons que nous parlons de temps qui vont, au minimum, du jour de traitement à plusieurs semaines. Les questions sous-jacentes s'attachent à savoir quelles parties de la chaîne sont les plus chronophages et seraient à optimiser, paralléliser ou distribuer ; quels sont les résultats intermédiaires qu'il est préférable de conserver pour éviter de les recalculer ; est-ce que l'exécution se passe normalement ? etc. Tout un ensemble de questionnement en rapport avec la connaissance sur la chaîne elle-même ; connaissance généralement acquise par expérience. Ces questionnements m'ont mené à aborder le problème d'un point de vue théorique, dans une approche plus générique – en ne me focalisant pas uniquement sur les chaînes de traitements – et d'un plus au niveau conceptuel :

« Comment une chaîne de traitements, ou un logiciel pour être générique, peut-il acquérir de l'information sur lui-même afin de fournir des informations pertinentes sur son comportement, et ceci sans que cela n'impacte, ou à minima, les développeurs et les utilisateurs ? »

Cette question est l'une des bases de la conception des WO : comment faire pour qu'un logiciel se connaisse lui-même, qu'il soit sage ? Il découle également de cette question une approche conceptuelle de décorrélation entre le développement métier d'une application et sa partie intelligente. Le concept de WO, introduit par mon activité de recherche, tente d'apporter une solution à cette problématique. Le framework WOF développé au sein de cette activité répond déjà partiellement à la problématique. Il permet, grâce à une approche introspective, réflexive et peu intrusive, d'acquérir une connaissance non sémantique sur les capacités du logiciel et de ses objets constitutifs ainsi que sur la manière habituelle dont il est utilisé. Afin de mettre en évidence son fonctionnement, différentes expériences ont été réalisées, elles se basent sur un simulateur à invocations discrètes développé pour ces travaux. Un simulateur à invocations discrètes étant une variante de simulateur à événements discrets qui invoque les méthodes d'une application afin de simuler le comportement d'un utilisateur. Les résultats obtenus sont très encourageants, car ils mettent en évidence la capacité du WOF à apprendre les habitudes et détecter l'inhabituel.

Les résultats préliminaires, tout comme les limites actuelles d'un système sage, mettent en avant les lacunes de notre approche et par conséquent mettent en évidence les perspectives scientifiques. Pour l'heure, notre approche se base sur un bus de communication afin de faire communiquer les WO, ce qui apporte des avantages non négligeables, cependant une approche plus classique par composition de WO n'est pas encore possible, la question de la composition de la sagesse restant encore en suspens. La sagesse acquise par le WO est au niveau événementiel, son agrégation au sein de l'objet dans un premier temps, puis du système de WO distribués est une problématique ouverte. En l'état actuel, les WO sont contraints de travailler avec des méthodes non paramétrées en ce qui concerne les classes héritant de WO, ceci est fortement limitant et nécessite une étude sur la prise en compte des méthodes paramétrées dans l'acquisition de la sagesse. L'une des originalités des WO est de travailler à un niveau non sémantique, numérique, l'objectif étant de fournir une information pertinente à l'utilisateur, donc sémantique, le lien entre l'information numérique et sémantique est une perspective. Ces problématiques particulières seront étudiées lors de la thèse d'Abdelhafid Dahhani qui débute. Quant au lien avec les chaînes de traitements, qui sont l'une des origines des WO, les travaux sont encore en cours, notamment au travers du projet NORMAL débuté en 2021. Ce projet vise à étudier les systèmes de flux de traitements distribués intelligents, et en ce qui me concerne leur capacité d'auto-apprentissage en vue de l'optimisation de leur distribution.

Les problématiques citées précédemment ne sont qu'une infime partie des ouvertures introduites par le concept de WO. Au travers de la thèse de Fabien Sartor, nous montrons qu'ils peuvent présenter un intérêt pour une approche originale de la synthèse de contrôleur, où le système contrôlé est capable de fournir une information plus riche sur son comportement et ses capacités au contrôleur. L'implémentation du concept nécessite des langages de haut niveau ayant une forte capacité introspective et réflexive, le choix de Java reste une solution efficace, cependant certaines limites sont imposées par le langage lui-même. Le fait qu'un WO soit capable de détecter des comportements inhabituels, ouvre des portes sur la capacité de tester l'adéquation entre les attentes d'un utilisateur et l'adaptabilité du système. Les WO sont par nature distribués, les envisager comme avatars de données mobiles est donc tout à fait pertinent, ils seraient ainsi une source de connaissance distribuée pour le système afin d'optimiser le placement des données de manière réactive ou proactive. Je pourrais continuer l'énumération, au final, les WO apportent une connaissance brute, au plus bas niveau de la conception du logiciel et les applications possibles sont celles que nous sommes en capacité d'imaginer.

Glossaire

- DAG** Graphe Orienté Acyclique (Directed Acyclic Graph). 51
- GL-IA** Groupe de recherche sur le Genie Logiciel et l'IA au sein du GDR GPL. 79
- GPS** Système de Positionnement Global. 46
- IA** Intelligence Artificielle). 25, 26, 79
- IOSTS** Systèmes de transitions symboliques à entrée sortie (Input Output Symbolic Transition System). 71, 73
- IoT** Internet des Objets (Internet of Things). 56, 73
- MAD** Écart absolu médian (Median Absolute Deviation). 39
- MMCMS** Médian sur de multiple séries Common/Master (Median of Multiple Common Master Series). 38, 42, 43
- offset** décalage spatial ou temporel. 41
- ORB** Détecteur de points d'intérêt locaux rapide et robuste (ORiented FAST, rotated BRIEF). 30
- RDTS** Série de déplacements relatifs (Relative Displacement Time Series). 39, 41–43
- S-MMCMS** Version glissante (Sliding-MMCMS) de MMCMS. 38
- SAR** Radar à synthèse d'ouverture. 25, 34, 51
- SED** Simulateur à Événement Discret. 62
- SID** Simulateur à Invocation Discrète. 62
- SIFT** Transformation de caractéristiques visuelles invariante à l'échelle (Scale-Invariant Feature Transform). 30
- SMA** Système Multi-Agents). 79
- SURF** Caractéristiques robustes accélérées (Speeded Up Robust Features). 30
- série** voir série temporelle. 28, 36, 70
- série temporelle** Ensemble de données ordonnées par le temps, également nommé série. 25, 28
- time-lapse** série temporelle d'images acquise par un appareil photo. 28, 31, 37, 38
- UML** Langage de Modélisation Unifié (Unified Modeling Language). 57–61, 64, 66, 67, 79
- WO** Objet Sage (Wise Object). 26, 56–62, 68, 72, 73, 79, 81, 82
- WOF** Cadre logiciel des Objets Sages (Wise Object Framework). 56, 58, 61, 70, 75, 79, 82
- WSS** Principe de stationnarité statistique faible (weak-sense stationarity). 70

Références

- [Alloui et al., 2018] Alloui, I., Benoit, E., Perrin, S., and Vernier, F. (2018). Wiot : Interconnection between wise objects and iot. In ICSOFT 2018, the 13th International Conference on Software Technologies, Porto, Portugal.
- [Alloui et al., 2019] Alloui, I., Benoit, E., Perrin, S., and Vernier, F. (2019). Wise objects for iot (wiot) : Software framework and experimentation. In van Sinderen, M. and Maciaszek, L. A., editors, Software Technologies, pages 349–371, Cham. Springer International Publishing.
- [Alloui et al., 2015] Alloui, I., Esale, D., and Vernier, F. (2015). Wise objects for calm technology. In 10th International Conference on Software Engineering and Applications (ICSOFT-EA 2015), ICSOFT-EA 2015, pages 468–471, Colmar, France. SciTePress 2015.
- [Alloui and Vernier, 2017] Alloui, I. and Vernier, F. (2017). A wise object framework for distributed intelligent adaptive systems. In ICSOFT 2017, the 12th International Conference on Software Technologies, Madrid, Spain.
- [Alloui and Vernier, 2018] Alloui, I. and Vernier, F. (2018). Wof : Towards behavior analysis and representation of emotions in adaptive systems. Communications in Computer and Information Science, 868 :244–267.
- [Anquetil et al., 2020] Anquetil, N., Etien, A., Houekpetodji, M. H., Verhaeghe, B., Ducasse, S., Toullec, C., Djareddir, F., Sudich, J., and Derras, M. (2020). Modular moose : A new generation software reverse engineering environment. In International Conference on Software Reuse, Tunis, Tunisia.
- [Benoit et al., 2015] Benoit, L., Dehecq, A., Pham, H.-T., Vernier, F., Trouvé, E., Moreau, L., Martin, O., Thom, C., Pierrot-Deseilligny, M., and Briole, P. (2015). Multi-method monitoring of glacier d’argentiére dynamics. Annals of Glaciology, 56(70) :118–128.
- [Berardino et al., 2002] Berardino, P., Fornaro, G., Lanari, R., and Sansosti, E. (2002). A new algorithm for surface deformation monitoring based on small baseline differential sar interferograms. IEEE Transactions on Geoscience and Remote Sensing, 40(11) :2375–2383.
- [Bodin et al., 2018] Bodin, X., Thibert, E., Sanchez, O., Rabatel, A., and Jaillet, S. (2018). Multi-annual kinematics of an active rock glacier quantified from very high-resolution dems : An application-case in the french alps. Remote Sensing, 10(4) :547.
- [Boyd and Chua, 1985] Boyd, S. and Chua, L. (1985). Fading memory and the problem of approximating nonlinear operators with volterra series. IEEE Transactions on Circuits and Systems, 32(11) :1150–1161.
- [Cervera et al., 2019] Cervera, A., Rantanen, V., Ovaska, K., Laakso, Marko and Nuñez-Fontarnau, J., Alkodsí, A., Casado, J., Faccioto, C., Häkkinen, A., Louhimo, R., Karinen, S., Zhang, K., Lavikka, K., Ljly, L., Pal Singh, M., and Hautaniemi, S. (2019). Anduril 2 : upgraded large-scale data integration framework. Bioinformatics, 35(19) :3815–3817.
- [Cheng et al., 2009] Cheng, B. H. C., de Lemos, R., Giese, H., Inverardi, P., Magee, J., Andersson, J., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Di Marzo Serugendo, G., Dustdar, S., Finkelstein, A., Gacek, C., Geihs, K., Grassi, V., Karsai, G., Kienle, H. M., Kramer, J., Litoiu, M., Malek, S., Mirandola, R., Müller, H. A., Park, S., Shaw, M., Tichy, M., Tivoli, M., Weyns, D., and Whittle, J. (2009). Software Engineering for Self-Adaptive Systems : A Research Roadmap, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Chu et al., 2017] Chu, W.-T., Zheng, X.-Y., and Ding, D.-S. (2017). Camera as weather sensor : Estimating weather information from single images. Journal of Visual Communication and Image Representation, 46 :233 – 249.

- [Desrues et al., 2019] Desrues, M., Malet, J.-P., Brenguier, O., Point, J., Stumpf, A., and Lorier, L. (2019). Tsm—tracing surface motion : A generic toolbox for analyzing ground-based image time series of slope deformation. *Remote Sensing*, 11(19) :2189.
- [Eltner et al., 2016] Eltner, A., Kaiser, A., Castillo, C., Rock, G., Neugirg, F., and Abellán, A. (2016). Image-based surface reconstruction in geomorphometry-merits, limits and developments. *Earth Surface Dynamics*, 4(2).
- [Fu et al., 2009] Fu, Q., Lou, J.-G., Wang, Y., and Li, J. (2009). Execution anomaly detection in distributed systems through unstructured log analysis. In *International conference on Data Mining (full paper)*. IEEE.
- [Gabrieli et al., 2016] Gabrieli, F., Corain, L., and Vettore, L. (2016). A low-cost landslide displacement activity assessment from time-lapse photogrammetry and rainfall data : Application to the tessina landslide site. *Geomorphology*, 269 :56–74.
- [Gance et al., 2014] Gance, J., Malet, J.-P., Dewez, T., and Travelletti, J. (2014). Target detection and tracking of moving objects for characterizing landslide displacements from time-lapse terrestrial optical images. *Engineering geology*, 172 :26–40.
- [GEORGE et al., 2003] GEORGE, J.-P., GLEIZES, M.-P., and GLIZE, P. (2003). Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie amas. *Revue d’intelligence artificielle*.
- [Groz et al., 2018] Groz, R., Simao, A., Bremond, N., and Oriat, C. (2018). Revisiting ai and testing methods to infer fsm models of black-box systems. In *2018 IEEE/ACM 13th International Workshop on Automation of Software Test (AST)*, pages 16–19.
- [Hadhri, 2020] Hadhri, H. (2020). *Traitement automatique de time-lapses stéréoscopiques de milieux naturels*. Theses, Université Savoie Mont Blanc.
- [Hadhri et al., 2019] Hadhri, H., Vernier, F., Atto, A., and Trouvé, E. (2019). Time-lapse optical flow regularization for geophysical complex phenomena monitoring. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150 :135–156.
- [Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.
- [He, 2017] He, H. (2017). *Application of the photogrammetry by multi-correlation of images for fast morphological evolution survey in rocky or glacial natural areas*. Theses, Université Grenoble Alpes.
- [He et al., 2016] He, H., Vernier, F., Ployon, E., and Villemin, T. (2016). 3d displacement retrieval on a scaled model of mountain slope by virtual multi-view photogrammetry. In *2nd Virtual Geoscience Conference*, Bergen, Norway.
- [He et al., 2015] He, H., Vernier, F., Villemin, T., Ployon, E., Deline, P., and Morra Di Cella, U. (2015). 3d displacement retrieval on glacial areas by airborne photogrammetry. In *8th International Workshop on the Analysis of Multitemporal Remote Sensing Images*, volume *Analysis of Multitemporal Remote Sensing Images*, pages 1 – 4, Annecy, France. IEEE.
- [Jain et al., 1997] Jain, A. K., Ratha, N. K., and Lakshmanan, S. (1997). Object detection using gabor filters. *Pattern Recognition*, 30(2) :295 – 309.
- [Kennedy, 2010] Kennedy, C. M. (2010). Decentralised metacognition in context-aware autonomic systems : Some key challenges. In *Metacognition for Robust Social Systems, Papers from the 2010 AAAI Workshop, Atlanta, Georgia, USA, July 11, 2010*, volume *WS-10-04 of AAAI Workshops*. AAAI.
- [Kenner et al., 2018] Kenner, R., Phillips, M., Limpach, P., Beutel, J., and Hiller, M. (2018). Monitoring mass movements using georeferenced time-lapse photography : Ritigraben rock glacier, western swiss alps. *Cold Regions Science and Technology*, 145 :127–134.

- [Kumar and Hanumanthappa, 2013] Kumar, M. and Hanumanthappa, M. (2013). Scalable intrusion detection systems log analysis using cloud computing infrastructure. In 2013 IEEE International Conference on Computational Intelligence and Computing Research, pages 1–4.
- [Lenzano et al., 2014] Lenzano, M., Lannutti, E., Toth, C., Lenzano, L., and Lovecchio, A. (2014). Assessment of ice-dam collapse by time-lapse photos at the perito moreno glacier, argentina. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences.
- [Liu and Huang, 2016] Liu, W.-C. and Huang, W.-C. (2016). Close range digital photogrammetry applied to topography and landslide measurements. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 41.
- [Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Maas et al., 2013] Maas, H.-G., Casassa, G., Schneider, D., Schwalbe, E., and Wendt, A. (2013). Photogrammetric techniques for the determination of spatio-temporal velocity fields at glacier san rafael, chile. Photogrammetric Engineering & Remote Sensing, 79(3) :299–306.
- [Marsy, 2020] Marsy, G. (2020). Apport de l'imagerie optique «time-lapse» stéréoscopique pour la quantification à haute résolution spatio-temporelle (4D) des dynamiques de versants en montagne. Theses, Université Savoie Mont Blanc.
- [Marsy et al., 2018a] Marsy, G., Vernier, F., Bodin, X., Castaings, W., and Trouvé, E. (2018a). Détection automatique de zones en mouvement dans des séries d'images non recalées : application à la surveillance des mouvements gravitaires. In Conférence française de photogrammétrie et télédétection, Marne la Vallée, France.
- [Marsy et al., 2018b] Marsy, G., Vernier, F., Bodin, X., Castaings, W., and Trouvé, E. (2018b). Détection automatique de zones en mouvement dans des séries d'images non recalées : application à la surveillance des mouvements gravitaires. Revue Française de Photogrammétrie et de Télédétection, 217-218 :pp. 25–31.
- [Marsy et al., 2020a] Marsy, G., Vernier, F., Bodin, X., Castaings, W., and Trouvé, E. (2020a). Temporal consolidation strategy for ground based image displacement time series. In IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium.
- [Marsy et al., 2020b] Marsy, G., Vernier, F., Bodin, X., Cusicanqui, D., Castaings, W., and Trouvé, E. (2020b). Monitoring mountain cryosphere dynamics by time lapse stereophotogrammetry. In ISPRS Journal of Photogrammetry and Remote Sensing. Elsevier.
- [Marsy et al., 2020c] Marsy, G., Vernier, F., Bodin, X., Trouvé, E., Castaings, W., Walpersdorf, A., Malet, E., and Girard, B. (2020c). Temporal consolidation strategy for ground based imagedisplacement time series. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.
- [Mason et al., 2019] Mason, A., Zhao, Y., He, H., Gompelman, R., and Mandava, S. (2019). Online anomaly detection of time series at scale. In 2019 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA), pages 1–8.
- [Moreaux et al., 2012] Moreaux, P., Sartor, F., and Vernier, F. (2012). An effective approach for home services management. In Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on, pages 47–51, Garching, Germany. IEEE.
- [Neyer, 2016] Neyer, F. M. (2016). Monitoring Rock Glaciers by Combining Photogrammetric and GNSS-Based Methods. PhD thesis, ETH Zurich.

- [Oreizy et al., 1999] Oreizy, P., Gorlick, M., Taylor, R., Heimhigner, D., Johnson, G. F., Medvidovic, N., Quilici, A., Rosenblum, D. S., and Wolf, A. (1999). An architecture-based approach to self-adaptive software. *IEEE Intell. Syst.*, 14 :54–62.
- [Oriat, 2005] Oriat, C. (2005). Jartege : A tool for random generation of unit tests for java classes. In Reussner, R., Mayer, J., Stafford, J. A., Overhage, S., Becker, S., and Schroeder, P. J., editors, *Quality of Software Architectures and Software Quality*, pages 242–256, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Pham, 2015] Pham, H. T. (2015). *Analysis of optical stereo Time Lapse and radar satellite images : application to the measurement of glacier displacement*. Theses, Université Grenoble Alpes.
- [Pham et al., 2014] Pham, H.-T., He, H., Vernier, F., Trouvé, E., Benoit, L., Moreau, L., and Girard, B. (2014). Analyse de "time-lapse" stéréo pour la mesure de déformation 3d, application au suivi du glacier d'argentières. In *Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014*, France.
- [Poncet, 2016a] Poncet, M. (2016a). Observation des glissements de terrain nouvelles technologies d'instrumentation à distance. Technical report, Cerema.
- [Poncet, 2016b] Poncet, M. (2016b). Observation des glissements de terrain nouvelles technologies d'instrumentation in situ. Technical report, Cerema.
- [Reidemeister et al., 2011] Reidemeister, T., Miao Jiang, and Ward, P. A. S. (2011). Mining unstructured log files for recurrent fault diagnosis. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 377–384.
- [Reidemeister et al., 2009] Reidemeister, T., Munawar, M. A., Jiang, M., and Ward, P. A. S. (2009). Diagnosis of recurrent faults using log files. In *CASCON*.
- [Roncella et al., 2014] Roncella, R., Forlani, G., Fornari, M., and Diotri, F. (2014). Landslide monitoring by fixed-base terrestrial stereo-photogrammetry. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(5) :297.
- [Russo et al., 2015] Russo, B., Succi, G., and Pedrycz, W. (2015). Mining system logs to learn error predictors : a case study of a telemetry system. *Empir. Softw. Eng.*, 20(4) :879–927.
- [Sartor, 2012] Sartor, F. (2012). *Modeling of interoperability of communicating object their cooperation : implementation to home automation*. Theses, Université de Grenoble.
- [Scaioni et al., 2018] Scaioni, M., Crippa, J., Corti, M., Barazzetti, L., Fugazza, D., Azzoni, R., Cernuschi, M., and Diolaiuti, G. (2018). Technical aspects related to the application of sfm photogrammetry in high mountain. In *2018 ISPRS TC II Mid-term Symposium "Towards Photogrammetry 2020"*, volume 42-2, pages 1029–1036. International Society for Photogrammetry and Remote Sensing.
- [Schwalbe and Maas, 2017] Schwalbe, E. and Maas, H.-G. (2017). The determination of high-resolution spatio-temporal glacier motion fields from time-lapse sequences. *Earth Surface Dynamics*, 5(4) :861.
- [Volat et al., 2014] Volat, M., Vernier, F., Doin, M.-P., Lasserre, C., Trouvé, E., and Pathier, E. (2014). Improving the execution of workflows for sar image analysis. In *Conference on Big Data from Space, Conference on Big Data from Space*, Frascati, Italy.
- [Volterra, 1959] Volterra, V. (1959). *Theory of functionals and of integral and integro-differential equations*. Dover, New York, NY.
- [Weinzaepfel et al., 2013] Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). DeepFlow : Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia.
- [Wiener, 1958] Wiener, N. (1958). *Nonlinear Problems in Random Theory*. MIT Press Cambridge.

-
- [Yusof et al., 2008] Yusof, R., Selamat, S. R., and Sahib, S. (2008). Intrusion alert correlation technique analysis for heterogeneous log. IJCSNS International Journal of Computer Science and Network Security, 8(9).
- [Zeigler et al., 2000] Zeigler, B., Prähofer, H., and Kim, T. G. (2000). Theory of Modeling and Simulation : Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press.
- [Ziadi et al., 2011] Ziadi, T., d. Silva, M. A. A., Hillah, L. M., and Ziane, M. (2011). A fully dynamic approach to the reverse engineering of uml sequence diagrams. In 2011 16th IEEE International Conference on Engineering of Complex Computer Systems, pages 107–116.

Résumé

Après une formation universitaire hésitante entre les sciences physiques et l'informatique, entre l'industrie et la recherche, j'ai embrassé une carrière dans l'enseignement supérieur et la recherche en informatique. Au travers d'un équilibre difficilement maintenu entre l'enseignement et la recherche, j'enseigne dans un large domaine de l'informatique allant de l'informatique industrielle au génie logiciel, en passant par les systèmes distribués et l'informatique théorique. Du côté recherche, mes travaux ont débuté dans le domaine du HPC lors de ma thèse, puis ont évolué à mon arrivée au LISTIC vers le traitement de séries temporelles d'images. Ces derniers, en m'interrogeant sur une approche conceptuelle de la réalisation de chaînes de traitements, m'ont permis de faire émerger une nouvelle thématique de recherche : les objets sages. Ces objets, au sens informatique du terme, ont la capacité originale d'apprendre sur eux-mêmes, par eux-mêmes.

Ce mémoire d'HDR présente mon environnement d'enseignement, synthétise mes travaux de recherche passés et ouvre vers ceux à venir.

Mots clés : Objet Sage, Design, Gestion de la Connaissance, Apprentissage Automatique

Abstract

After a master degrees in computer science, I hesitated between industry and academic research, at the end of the day, higher education and research in computer science have welcomed me. Maintaining a balance between the teaching activities and the research activities, I teach in a wide spectrum of computer science disciplines : industrial computing, software engineering, distributed systems and theoretical computer science. On the research side, I defend a PhD thesis in the High Performance Computing field, then my research field evolves towards image time series processing when I joined LISTIC laboratory. The latter, by searching for a conceptual approach to the implementation of processing chains, allowed me to bring out a new research theme : wise objects. These objects, in the computer sense of the term, have the original capacity to learn about themselves, by themselves.

This HDR thesis presents my teaching activities, synthesizes my past research works and opens perspectives to my future research lines.

Keywords : Wise Object, Conception, Knowledge Management, Machine Learning