Contributions à l'apprentissage de représentations pour l'indexation basée sur le contenu visuel

David Picard

Université // Paris Seine

HABILITATION À DIRIGER DES RECHERCHES

présentée en section 61, génie informatique, automatique et traitement du signal

Contributions à l'apprentissage de représentations pour l'indexation basée sur le contenu visuel

par

David Picard

ETIS UMR 8051, Université Paris Seine, Université de Cergy-Pontoise, ENSEA, CNRS
6 avenue du Ponceau, 95014 Cergy-Pontoise, France

Soutenue le 30 novembre 2017 devant le jury composé de :

STÉPHANE CANU, Professeur des Universités, LITIS EA 4108 - INSA de Rouen.	Rapporteur
FRÉDÉRIC PRECIOSO, Professeur des Universités, I3S - UMR CNRS 7271 - Université de Nice Sophia Antipolis.	Rapporteur
STÉPHANE MARCHAND-MAILLET, Professeur, VIPER, Université de Genève.	Rapporteur
NICOLE VINCENT, Professeur des Universités, LIPADE, Université Paris Descartes.	Examinatrice
FLORENCE D'ALCHÉ-BUC, Professeur, LTCI, Télécom Paristech.	Examinatrice
FLORENT PERRONNIN, Deputy Lab Manager, Naver Labs Europe.	Examinateur
DAN VODISLAV, Professeur des Universités, ETIS, Université de Cergy-Pontoise.	Garant

Université // Paris Seine

HTTP ://WWW.U-CERGY.FR

First printing, Oct 2017

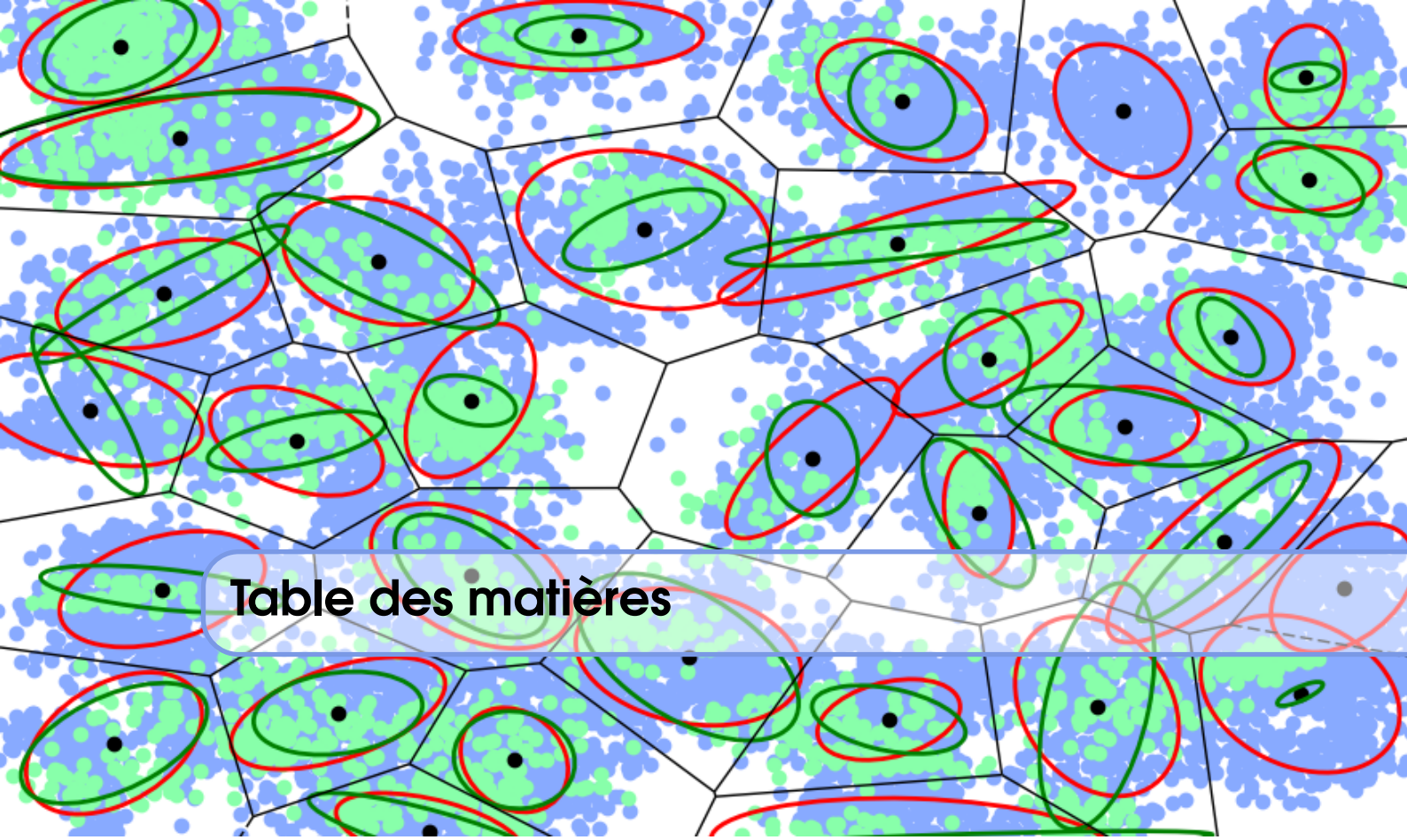


Table des matières

1 Curriculum vitae 5

2 Résumé des activités de recherche et d'enseignement 9

2.1 Activités d'enseignement 9

2.1.1 Résumé des activités d'enseignement 9

2.1.2 Responsabilités liées à l'enseignement 11

2.2 Activités de recherche 11

2.2.1 Résumé des activités de recherche 12

2.2.2 Projets et collaborations 12

2.2.3 Organisation 12

2.2.4 Responsabilités scientifiques 13

2.2.5 Encadrement 14

2.3 Bibliographie personnelle 15

Articles 15

Conférences 16

3 Détail des travaux de recherche 19

3.1 Apprentissage de représentations pour l'analyse d'images 19

3.1.1 Approximation tensorielle de noyaux d'appariement 19

3.1.2 Information spatiale 22

3.1.3 Compression des représentations 24

3.2 Apprentissage de représentations pour l'analyse de vidéos 26

3.2.1 Modélisation des descripteurs bas niveau 27

3.2.2 Information structurée 30

3.3 Apprentissage distribué de représentations visuelles 33

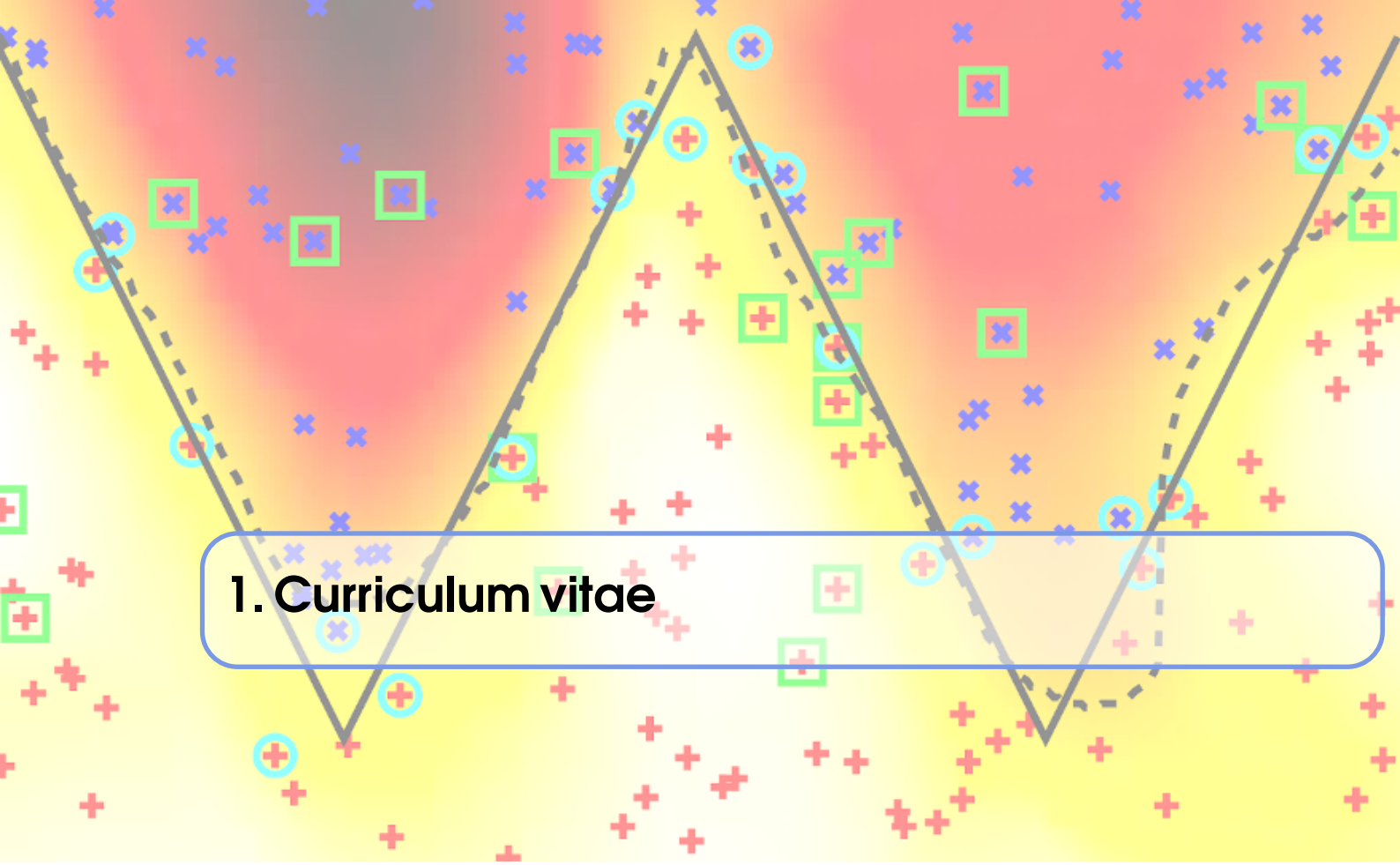
3.3.1 Apprentissage statistique distribué à l'aide de protocoles gossip 33

3.3.2 Apprentissage distribué de réseaux de neurones profonds 40
3.4 Applications aux collections culturelles et patrimoniales 40
3.4.1 Indexation d'images 41
3.4.2 Reconnaissances de papier photographique 42
3.4.3 Reconnaissance de filigranes 44

4 Projet de recherche 47
4.1 Noyaux de matching approximés pour l'apprentissage profond de représentations 47
4.1.1 Fonction de matching 47
4.1.2 Objectif 1 : Fonctions de matching dans les réseaux convolutifs 48
4.1.3 Objectif 2 : Apprentissage structuré d'information spatiale dans les fonctions de matching 48
4.2 Apprentissage de représentation avec fortes contraintes de ressources 49
4.2.1 Objectif 1 : Réduction de la taille complexité des modèles 49
4.2.2 Objectif 2 : Modèles embarqués 49
4.3 Apprentissage profond distribué 50
4.3.1 Objectif 1 : Entraînement par distribution de modèles 50

5 Sélection d'articles 51

Bibliography 105



1. Curriculum vitae

David Picard

ETIS - ENSEA
6 avenue du Ponceau
95014 Cergy-Pontoise Cedex

Tel : +33 6 79 64 11 96
Email : picard@ensea.fr
Webpage : http://perso-etis.ensea.fr/~picard

Données personnelles

— Naissance : 1er Octobre 1982 à Leipzig (Allemagne)
— Nationalité : Française
— Langues : Anglais, Allemand, Français

Thèmes de recherche

— Reconnaissance et indexation de contenus visuels
— Traitement d'images et de vidéos, vision par ordinateur
— Apprentissage statistique, optimisation distribuée

Expérience

— Depuis Sep. 2017 : Délégation CNRS au LIP6, Université Pierre et Marie Curie, Paris, France.
— Fév. 2015 - Oct. 2017 : **Directeur du Département d'Informatique et des Techniques Numériques** de l'ENSEA, Cergy, France.
— Depuis Sep. 2010 : **Maître de conférences** à l'ENSEA (École Nationale Supérieure de L'Électronique et de ses Applications), Cergy, France.
— Jan. 2009 - Sep. 2010 : **Post-doctorant** au LIP6, Université Pierre et Marie Curie, Paris, France.

Formation

— 2008, **Ph.D.**, Université de Cergy-Pontoise (France) : *"Recherche d'images sur un réseau à l'aide d'un système multi-agents"* (*"Distributed Image Retrieval using Multi-Agent Systems"*), Directeurs : Arnaud Revel and Matthieu Cord.
— 2005, **Diplôme d'ingénieur en Électronique**, option Informatique et Systèmes, ENSEA (France).
— 2005, **DEA Traitement des Images et du Signal**, Université de Cergy-Pontoise (France).

Séjours invités

— **Content Based Indexing at Amsterdam Conservation Center** : Collaboration avec le Reijksmuseum (R. G. Erdmann), Amsterdam, 1 mois en 2015.
— **DAAD** : **Learning low level visual descriptors for image and video categorization**, collaboration à la TU Darmstadt (V. Willert), 2 mois en 2014.

Primes

— **PEDR** : Prime d'encadrement doctoral et de recherche, 2015-2019.

Responsabilités

Responsabilités externes

— Assesseur au comité scientifique du NICAS 2015, 2017 (Netherlands Institute for Conservation, Art and Science, dépendant de la NWO, Pays-Bas)

Responsabilités locales

— Membre du conseil scientifique de l'ENSEA (Oct. 2011 - Dec. 2015).
— Membre du comité technique de l'ENSEA (Oct. 2011 - Dec. 2014).
— Responsable NTIC dans la pédagogie de l'ENSEA (Sept. 2010, Oct. 2017).

Organisation

— Éditeur associé pour *"Journal of Electronic Imaging special issue on Image Processing for Cultural Heritage"*, Jan/Feb 2017.
— Co-organisateur de la session spéciale *"Image processing for cultural heritage"* à IPTA 2015, Nov. 2015, Orléans, France.
— Organisateur de la session spéciale *"Traitement du signal et des images pour l'art et le patrimoine"* à Gretsi 2015, Sept. 2015, Lyon, France.
— Co-organisateur du workshop GeoDiff à VISAPP 2013, Feb. 2013, Barcelona, Spain.
— Organisateur de la session spéciale *"Machine Learning and Multimedia"* à ESANN 2013, Apr. 2013, Brugge, Belgium.

Comité de programme

— Comité technique de programme de CVPR 2015, 2016, 2017, 2018.
— Comité technique de ESANN 2014, 2015, 2016, 2017, 2018, Brugge, Belgium.
— Comité technique de 3DOR 2014, 2015.

Activité de relecture

— Relecteur pour les journaux IEEE Trans. on Pattern Analysis and Machine Intelligence, IEEE Trans. on Multimedia, IEEE Trans. on Robotics, IEEE Signal Processing Letters,

Journal of Machine Learning Research, Neurocomputing, Computer Vision and Image Understanding, Machine Vision and Applications, Neural Processing Letters, Multimedia Tools and Applications.
— Relecteur pour les conférences CVPR (depuis 2015), ICCV (depuis 2015), ESANN (depuis 2013).

Encadrement

Ph.D.

— Marie-Morgane Paumard, "Apprentissage de ré-assemblage numérique de fragments 3D", co-encadrée avec Hedi Tabie (MCF ENSEA), Vivien Barrière (MCF UCP, Archéologue) et Dan Vodislav (PU UCP), démarrée en oct. 2017.
— Pierre Jacob, "Automatic Labeling for Image Collections Exploration", co-encadré avec A. Histace (PU ENSEA) et E. Klein (Chercheur, Pôle Judiciaire de la Gendarmerie Nationale), démarrée en mars 2017.
— Diogo Luvizon, "Activity recognition and classification from 3D videos", co-encadré avec H. Tabia (MCF ENSEA), démarrée en oct. 2015.
— Jérôme Fellus, "Algorithmes décentraliséset asynchrone pour l'apprentissage statistique large échelle et application à l'indexation multimédia", co-encadré avec P.-H. Gosselin (PU ENSEA), 2012-2017.
— Romain Negrel, "Représentations optimales pour la recherche d'images dans des bases patrimoniales", co-encadré avec P.-H. Gosselin (PU ENSEA), 2011-2014.

Post-doc

— Yi Ren, "Automatic labeling of cultural heritage images", 2015.
— Olivier Kihl, "Low-level Visual Descriptors for Video Categorization", 2012-2014.

Logiciels développés

— JKernelMachines : Java Library for easy research in Kernel Machines. `http://mloss.org/software/view/409/`, plus de 9000 téléchargements.
— VLAT : C/C++ library to compute efficient tensor based image features. `http://www.vlat.fr`
— STA : Matlab library for spatial tensor aggregation. `https://github.com/davidpicard/sta`

Bibliométrie

— 10 revues internationales avec comité de relecture (PR, CVIU, Neurocomputing, JMLR, PRL, ...)
— 1 revue nationale avec comité de relecture (TS)
— 32 conférences internationales avec comité de relecture (CVPR, ICIP, ICPR, ESANN, ...)
— 2 conférences nationales avec comité de relecture (Gretsi, Orasis)

2. Résumé des activités de recherche et d'enseignement

Dans ce chapitre, nous résumons nos activités d'enseignement et de recherche sur la période commençant à notre recrutement en qualité de maître de conférences à l'ENSEA.

2.1 Activités d'enseignement

Les enseignements à l'ENSEA sont répartis entre 5 départements : DITN (Département d'informatique et des techniques numériques), DEP (département d'électronique et de physique), DA (département d'automatique), DST (département de signal et télécommunications) et DSH (département de sciences-humaines). Nos activités d'enseignement se sont principalement déroulées dans le cadre du DITN à travers des cours/TD/TP d'informatique et de génie informatique, ainsi que dans le cadre du DST pour des cours/TD/TP de traitement des images et d'apprentissage statistique. Le master M2 I&ISC (informatique et ingénierie des systèmes complexes) de l'université de Cergy-Pontoise a aussi contribué à une partie de nos activités.

2.1.1 Résumé des activités d'enseignement

Une année type en terme d'activités d'enseignement est présentée dans le tableau 2.1 et montre une répartition par département et par type. Il est a noter que les cours en troisième année se déroulent par section de maximum 24 étudiants, ce qui permet une pédagogie dans laquelle les cours et les travaux dirigés peuvent être mélangés.

Pour les cours dont nous avons eu la responsabilité, nous nous sommes efforcés de mettre à jour autant le contenu du cours que la forme pédagogique avec il a été enseigné. En particulier, nous détaillons les cours suivants :

- **Java** : Conscient que 6 heures d'amphithéâtre et 12 de travaux pratiques sont extrêmement peu pour acquérir des compétences en programmation objet, nous avons pris le parti de numériser en grande parti ce cours. Nous avons donc enregistré la partie cours proprement sous la forme d'une douzaine de vidéos courtes portant chacune sur un concept de programmation orientée objet particulier (notion de classe, héritage, polymorphisme, etc) et associées à une série de questions de cours auto-générées, le tout disponible sur la plate-forme moodle de l'établissement. Avant chaque séance d'amphithéâtre, les élèves-ingénieurs sont supposés

	Cours	TD	TP
DITN			
Java (2A)	6	-	12
C++ (2A)	8	-	4
Systèmes d'exploitation (3A)	14	4	24
Programmation parallèle (3A)	4	2	24
DST			
Théorie des jeux et optimisation convexe (2A)	4	-	4
Analyse de contenus multimédia (3A)	2	-	
DA			
Intelligence artificielle pour la commande (3A)	4	4	8
Projets (hors départements)			
Projet d'initiation à l'électronique (1A)	-	-	20
Projets de fin d'études (3A)	5		
Master I&ISC (M2)			
Traitement des images	2		
Indexation multimédia	8		

TABLE 2.1: Répartition des cours lors de l'année 2016-2017. 2A désigne les cours de deuxième année (eq. M1) et 3A désigne les cours de troisième année (eq. M2). Les cours en gras sont ceux pour lesquels la responsabilité est assumée.

avoir vu les vidéos correspondantes et les heures de cours servent de *"séance retour"* durant laquelle nous pouvons donner des précisions sur des éléments non compris ou bien corriger les questions trop complexes. Les TP suivent la même progression que les séances d'amphithéâtre et les vidéos.

- **Intelligence artificielle pour la commande** : Ce cours s'adresse à une section d'électroniciens spécialisés en automatique et a pour but de faire une introduction aux méthodes d'apprentissage statistique pour la commande des systèmes. Le cours présente rapidement le principe de minimisation du risque empirique puis détaille quelques algorithmes basés sur des réseaux de neurones et comment l'apprentissage peut se substituer à des problèmes de modélisation ou des problèmes inverses très complexes. Les séances de TD sur machine consistent à utiliser un réseau de neurones pour faire de l'identification de processus puis de la commande optimale. Les séances de TP forment une très rapide introduction à l'apprentissage par renforcement à travers la commande d'un pendule inversé par Q-learning (DQN).
- **Indexation multimédia** : Ce cours de M2 est très orienté recherche et vise à offrir un panorama des méthodes d'apprentissage de représentations pour l'indexation par le contenu visuel. Nous l'avons découpé en trois parties, à savoir l'apprentissage de représentations pour les images (*Bag of Words*, *Fisher Vectors*, *etc*), les méthodes d'indexation (*Inverted Files*, *LSH*, *Product Quantization*, *etc*) et les méthodes de classification (SVM, MKL, Boosting, Decision Trees, *etc*). Le cours s'appuie sur la présentation d'algorithmes d'apprentissage statistique classiques et d'articles de la littérature de vision par ordinateur très récents (principalement du deep learning ces dernières années).

Le total des heures de cours enseignées par année se trouve dans le tableau 2.2. Il faut noter deux points pour expliquer le nombre important d'heures complémentaires effectuées : Premièrement, l'ENSEA est un petit établissement ($\sim$800 étudiants) en sous-effectif ce qui implique le recours à des intervenants extérieurs faisant souvent faux bond au dernier moment. D'autre part, un grande

partie de ces heures sont liées à diverses responsabilités administratives ou pédagogiques qui sont comptabilisées sous forme de missions d'enseignement (environ 80h eq. TD en 2016).

Année	h eq. TD
2010-2011	235
2011-2012	287
2012-2013	373
2013-2014	340
2014-2015	286
2015-2016	320
2016-2017	267

TABLE 2.2: Nombre d'heures équivalent TD enseignées par année.

2.1.2 Responsabilités liées à l'enseignement

- Moodle : Dès notre arrivée à l'ENSEA, nous avons été amené à prendre la responsabilité des activités de Nouvelles Technologies de l'Information et de la Communication (NTIC) dans l'établissement. Ceci consiste principalement à administrer et animer la plateforme Moodle de l'école qui est aujourd'hui la seule source numérique de contenus pédagogiques de l'établissement. Nous avons participé à un effort de numérisation de la pédagogie de l'établissement à travers des formations à l'utilisation de la plateforme (*i.e.*, comment créer et gérer du contenu sur moodle), ainsi que par la création de cours servant d'exemples (comme Java) sur le changement de pédagogie possible avec ces outils. D'autre part, nous avons fait en sorte que la plateforme Moodle soit utilisable pour avoir une interaction avec les étudiants, que ce soit pour la remise de devoirs ou bien pour le choix des cours électifs (pour lesquels nous avons créé les programmes nécessaires).
- Responsabilité de département : En février 2015, nous avons été élu responsable du département d'informatique et des techniques numériques. Outre les aspects administratifs de la gestion d'un département (budget, personnel), nous avons animé la refonte des programmes de tronc commun de l'établissement (2 premières années, soit 2 fois 250 étudiants environ). Les changements structurels sur le déroulement des études étant importants (plus de place aux stages, choix de majeures/mineures thématiques en deuxième année), nous avons proposé une nouvelle maquette pédagogique pour les enseignements d'informatique et d'électronique numérique issue des réunions de département que nous avons organisées. En particulier, cette réforme introduit des TD sur machine qui n'existaient pas avant et qui nous semblent essentiels à l'acquisition de compétences dans le numérique. Nous avons d'autre part mis en place la création de nouveaux cours dans le cadre de cette réforme, notamment une majeure sur les Systèmes et Réseaux qui permet d'aborder plus sereinement les spécialités de troisième année concernées, ainsi qu'une option Intelligence Artificielle et Big Data qui devrait permettre de préparer les élèves-ingénieurs intéressés à suivre le master I&ISC de l'université.
- Commission de recrutement des PRAG affectés dans le supérieur : Nous avons participé à ces commissions locales de recrutement pour des postes affectés à l'ENSEA en 2015, 2016 et 2017.

2.2 Activités de recherche

Les activités décrites ici couvrent la période depuis le recrutement à l'ENSEA (septembre 2010) et ont été effectuées au sein du laboratoire ETIS (UMR 8051).

2.2.1 Résumé des activités de recherche

Le bilan de publication à l'été 2017 est décrit dans le tableau 2.3. Il contient les publications effectuées en thèse et en post-doctorat, mais celle-ci ont un impact négligeable (2 journaux et 5 conférences internationales) sur la production totale. La grande majorité de cette production est liée à l'encadrement doctoral ou post-doctoral, le reste étant lié à des collaborations et dans un partie infime des cas à un travail strictement personnel.

Article de revues internationales	10
Article de revues francophones	1
Articles de conférences internationales	34
Articles de conférences nationales	2
Communications sans actes	4

TABLE 2.3: Bilan de production scientifique.

Ces travaux de recherches peuvent être classés en quatre catégories qui forment les sections du chapitre 3, et qui sont :

- Représentations pour les images : Les travaux de cette catégorie visent à concevoir des méthodes d'apprentissage de représentations qui permettent de comparer des images. Les travaux sont principalement basés sur la linéarisation de noyaux d'appariement ce qui permet d'utiliser toutes les méthodes à noyaux par la suite.
- Représentations pour les vidéos : Les travaux de cette catégorie sont très similaires à la précédente si ce n'est qu'ils se concentrent sur l'apprentissage de représentations afin de réaliser des tâches de reconnaissance d'action ou d'activités dans des vidéos.
- Apprentissage décentralisé : Cette catégorie concerne l'adaptation d'algorithmes d'apprentissage à un contexte décentralisé dans lequel chaque machine participante possède une partie du jeu de données et aucun orchestrateur n'existe. Nous nous sommes basés sur les protocoles Gossip dit *sum-weights* qui permettent d'effectuer des calculs de moyenne pondérée de manière décentralisée et avons adapté de nombreux algorithmes d'apprentissage statistique de manière à s'exprimer sous une forme exploitable par ces protocoles.
- Applications aux collections culturelles et patrimoniales : Cette catégorie concerne les applications des travaux précédemment cités à des problématiques rencontrées dans le cadre de partenariats avec des institution culturelles et patrimoniales, principalement pour l'annotation, la reconnaissance ou l'indexation de collections d'images.

2.2.2 Projets et collaborations

Les projets auxquels nous avons pris part sont listés dans le tableau 2.4. La très grande majorité de ces projets (en montant) a servi à financer des contrats doctoraux ou post-doctoraux, ainsi que des séjours invités à l'étranger (Darmstadt 2014, Amsterdam 2015).

La liste des collaborations passées ou encore actives est présentée par thématique dans le tableau 2.5. Uniquement les collaborations ayant donné lieu à une publication, un projet, l'organisation d'une manifestation ou la participation à un challenge sont listées.

2.2.3 Organisation

Nous avons également participé à l'organisation des points listés dans le tableau 2.6. En particulier, l'organisation de l'édition spécial de Journal of Electronical Imaging sur le patrimoine a été un véritable succès puisque 30 articles ont été publiés dans cette édition et même quelques articles surnuméraires de bonne qualité ont été publiés dans les numéros standards suivants.

Nom	Année	Financement	Montant
GeoDiff	2011-2012	PEPS CNRS	15k
Représentations pour la recherche d'images	2011-2014	Thèse Patrima	105k
Culture 3D Cloud	2012-2015	PIA	155k
Terrarush	2013-2015	PIA	90k
Learning low-level descriptors	2014	DAAD (Allemagne)	3k
Qwant	2014	Qwant (contrat recherche privé)	10k
CBI at Amsterdam Conservation Center	2015	BQR ENSEA	3k
Fast learning of Multiple Kernel Machines	2015	BQR ENSEA	1k
ASAP	2015	Patrima	60k
Activity recognition from 3D videos	2015-2018	Thèse CNPQ (Brésil)	105k
ALICE	2017-2020	Thèse i-Site Paris Seine	105k
Archepuz'3D	2017-2020	Thèse Patrima	105k
Total			757k

TABLE 2.4: Liste des projets effectués, les projet en gras sont ceux dont la responsabilité a été assumée.

Représentations d'images	
UPMC - LIP6	Matthieu Cord
CNAM Paris	Nicolas Thome
LIG CNRS UMR 5217	George Quénot
Apprentissage statistique	
Technische Universität Darmstadt	Dr.-Ing. Volker Willert, Akad. Oberrat
LITIS, Université de Rouen	Alain Rakotomamonjy
Collections patrimoniales	
Rijksmuseum et University of Amsterdam	Robert G. Erdmann
Cornell University	Prof. C. Richard Johnson
Western Washington University	Andy Klein
PapierStruktur	Georg Dietz
ENS Lyon	Patrice Abry
Polytech'Orléans	Aladine Chetouani

TABLE 2.5: Liste des collaborations par thématique.

2013	ESANN special session on ML for Multimedia Retrieval	Co-organisateur et Chairman
2013	Workshop Geodiff	chairman
2015	Gretsi session spéciale sur le traitement du signal pour l'art	Co-organisateur et chairman
2015	IPTA special session on image processing for art	Co-organisateur
2016	Journée GDR ISIS traitement d'images pour le patrimoine	Co-organisateur et chairman
2016	JEI special issue on image processing for cultural heritage	associate editor

TABLE 2.6: Liste des participations à l'organisation de conférences ou de numéros spéciaux.

2.2.4 Responsabilités scientifiques

Nous avons participé aux comités de sélection pour le recrutement de maîtres de conférences de la liste suivante :

— École Nationale Supérieure de L'Électronique et de ses Applications, 2012 (local)
— École Nationale Supérieure de L'Électronique et de ses Applications, 2013 (local)

— Université de Rouen, 2013
— Université de Nice, 2015
— Université Pierre et Marie Curie, 2015
— Université de Cergy-Pontoise , 2016
— École Nationale Supérieure de L'Électronique et de ses Applications, 2017 (local)
— École Nationale Supérieure de L'Électronique et de ses Applications, 2017 (local)

Dautre part, nous avons participé aux jurys de thèse en tant qu'examinateur précisés dans la liste suivante :

— Hicham Randrianarivo, CNAM de Paris, 2016
— Ngoc Bich Dao, Université de La Rochelle, 2017

2.2.5 Encadrement

Le bilan de l'encadrement doctoral depuis le recrutement est de 225% dont 100% parmi les thèses soutenues. La liste des thèses encadrées est la suivante :

Doctorant :	Romain Negrel
Date :	2011-2014 (Soutenance le 3 décembre 2014)
Sujet :	Représentations optimales pour la recherche d'images dans des bases patrimoniales
Encadrement :	**David Picard (50%)**, Philippe-Henri Gosselin (50%)
Jury :	Florent Perronnin, Frédéric Jurie, George Quénot, Bernard Merialdo
Situation actuelle :	Maître de conférences ESIEE

Doctorant :	Jérôme Fellus
Date :	2012-2017 (Soutenance le 3 octobre 2017)
Sujet :	Algorithmes décentraliséset asynchrone pour l'apprentissage statistique large échelle et application à l'indexation multimédia
Encadrement :	**David Picard (50%)**, Philippe-Henri Gosselin (50%)
Jury :	Francis Bach, Alain Rakotomamonjy, Nicolas Thome, Frédéric Precioso, Elsa Dupraz
Situation actuelle :	Ingénieur UCP

Doctorant :	Diogo Luvizon
Date :	2015-2018
Sujet :	Activity recognition and classification from 3D videos
Encadrement :	**David Picard (45%)**, Hedi Tabia (45%), Dan Vodislav (10%)

Doctorant :	Pierre Jacob
Date :	2017-2020
Sujet :	Automatic Labelling for Image Collections Exploration
Encadrement :	**David Picard (33%)**, Édouard Klein (33%), Aymeric Histace (33%)

Doctorant :	Marie Morganne Paumard
Date :	2017-2020
Sujet :	Apprentissage de ré-assemblage numérique de fragments 3D
Encadrement :	**David Picard (40%)**, Hedi Tabia (40%), Vivien Barrière (10%), Dan Vodislav (10%)

D'autre part, en plus de ces thèses, nous avons encadré un stage de M2 (Romain Carrara, 2016) à 100% ainsi que des projets de recherche des étudiants de M2 (d'une durée équivalente de 150h sur leur programme d'enseignement) à raison de deux en moyenne par an. Certains de ces projets ont d'ailleurs donné lieu à des publications.

Enfin, nous avons aussi participé à l'élaboration et la réalisation des projets de recherche de deux post-doctorants lors de leur séjour au laboratoire, à savoir Olivier Kihl (2012-2014) et Yi Ren (2015).

2.3 Bibliographie personnelle

Articles

[1] Diogo Carbonera LUVIZON, Hedi TABIA et David PICARD. "Learning features combination for human action recognition from skeleton sequences". In : *Pattern Recognition Letters* (2017).

[2] Olivier KIHL, David PICARD et Philippe-Henri GOSSELIN. "Local polynomial space–time descriptors for action classification". In : *Machine Vision and Applications* 27.3 (2016), pages 351–361.

[3] Jerome FELLUS, David PICARD et Philippe-Henri GOSSELIN. "Asynchronous gossip principal components analysis". In : *Neurocomputing* 169 (2015), pages 262–271.

[4] Jérôme FELLUS, David PICARD et Philippe-Henri GOSSELIN. "Indexation multimédia par dictionnaires visuels en environnement décentralisé. Une approche par protocoles Gossip." In : *Traitement du Signal* 32.1 (2015), pages 39–64.

[5] Olivier KIHL, David PICARD et Philippe-Henri GOSSELIN. "A unified framework for local visual descriptors evaluation". In : *Pattern Recognition* 48.4 (2015), pages 1174–1184.

[6] David PICARD, Philippe-Henri GOSSELIN et Marie-Claude GASPARD. "Challenges in Content-Based Image Indexing of Cultural Heritage Collections". In : *Signal Processing Magazine, IEEE* 32.4 (2015), pages 95–102.

[7] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. "Web scale image retrieval using compact tensor aggregation of visual descriptors". In : *IEEE Multimedia* 20.3 (2013), pages 24–33.

[8] David PICARD et Philippe-Henri GOSSELIN. "Efficient image signatures and similarities using tensor products of local descriptors". In : *Computer Vision and Image Understanding* 117.6 (2013), pages 680–687.

[9] David PICARD, Nicolas THOME et Matthieu CORD. "JKernelMachines : A simple framework for Kernel Machines". In : *Journal of Machine Learning Research* 14.May (2013), pages 1417–1421.

[10] David PICARD, Arnaud REVEL et Matthieu CORD. "An application of swarm intelligence to distributed image retrieval". In : *Information Sciences* 192.June 2012 (2012), pages 71–81.

[11] David PICARD, Matthieu CORD et Arnaud REVEL. "Image retrieval over networks : Active learning using ant algorithm". In : *IEEE Transactions on Multimedia* 10.7 (2008), pages 1356–1365.

Conférences

[1] Patrice ABRY et al. "Wove paper analysis through texture similarities". In : *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE. 2016, pages 144–148.

[2] David PICARD. "Preserving local spatial information in image similarity using tensor aggregation of local features". In : *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE. 2016, pages 201–205.

[3] David PICARD, Thomas HENN et Georg DIETZ. "Non-negative dictionary learning for paper watermark similarity". In : *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE. 2016, pages 130–133.

[4] Nicolas CAZIN, Aymeric HISTACE, David PICARD et Benoıt GAUDOU. "On The Joint Modeling of The Behavior of Social Insects and Their Interaction With Environment by Taking Into Account Physical Phenomena Like Anisotropic Diffusion". In : *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, Cham. 2015, pages 151–164.

[5] Jerome FELLUS, David PICARD et Philippe-Henri GOSSELIN. "Asynchronous decentralized convex optimization through short-term gradient averaging". In : *Proceedings*. Presses universitaires de Louvain. 2015, page 255.

[6] Hervé LE BORGNE et al. "IRIM at TRECVid 2015 : semantic indexing". In : *Proceedings of TRECVID*. 2015.

[7] David PICARD et Philippe-Henri GOSSELIN. "Évaluation de descripteurs visuels pour l'annotation automatique d'images patrimoniales". In : *Gretsi 2015*. 2015.

[8] Nicolas BALLAS et al. "Irim at TRECVID 2014 : Semantic indexing and instance search". In : *Proceedings of TRECVID*. 2014.

[9] Thibaut DURAND, David PICARD, Nicolas THOME et Matthieu CORD. "Semantic pooling for image categorization using multiple kernel learning". In : *ICIP*. 2014.

[10] Thibaut DURAND, Nicolas THOME, Matthieu CORD et David PICARD. "Incremental learning of latent structural svm for weakly supervised image classification". In : *ICIP*. 2014.

[11] Jerome FELLUS, David PICARD et Philippe-Henri GOSSELIN. "Dimensionality reduction in decentralized networks by Gossip aggregation of principal components analyzers". In : *ESANN 2014*. 2014, pages 171–176.

[12] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. "Dimensionality reduction of visual features using sparse projectors for content-based image retrieval". In : *IEEE Int. Conf. on Image Processing (ICIP)*. 2014, pages 2192–2196.

[13] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. "Efficient Metric Learning Based Dimension Reduction Using Sparse Projectors For Image Near Duplicate Retrieval". In : *ICPR*. 2014.

[14] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. "Evaluation of second-order visual features for land-use classification". In : *Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on*. IEEE. 2014, pages 1–5.

[15] David PICARD et Inbar FIJALKOW. "Second order model deviations of local Gabor features for texture classification". In : *Signals, Systems and Computers, 2014 48th Asilomar Conference on*. IEEE. 2014, pages 917–920.

[16] David PICARD, Ngoc-Son VU et Inbar FIJALKOW. “Photographic paper texture classification using model deviation of local visual descriptors”. In : *IEEE Int. Conf. on Image Processing*. 2014, pages 5701–5705.

[17] Hedi TABIA, Hamid LAGA, David PICARD et Philippe-Henri GOSSELIN. “Covariance descriptors for 3D shape matching and retrieval”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pages 4185–4192.

[18] Mehdi BADR et al. “Multi-criteria search algorithm : an efficient approximate k-nn algorithm for image retrieval”. In : *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE. 2013, pages 2901–2905.

[19] Jerome FELLUS, David PICARD et Philippe-Henri GOSSELIN. “Decentralized K-means using randomized Gossip protocols for clustering large datasets”. In : *IEEE 13th International Conference on Data Mining Workshops*. IEEE. 2013, pages 599–606.

[20] Jérôme FELLUS, David PICARD et Philippe-Henri GOSSELIN. “Calcul décentralisé de dictionnaires visuels pour l’indexation multimédia dans les bases de données réparties sur les réseaux”. In : *ORASIS : Orasis, Congrès des jeunes chercheurs en vision par ordinateur*. 2013.

[21] Philippe-Henri GOSSELIN et David PICARD. “Machine Learning and Content-Based Multimedia Retrieval”. In : *ESANN 2013*. 2013, pages 251–260.

[22] Olivier KIHL, David PICARD et Philippe Henri GOSSELIN. “Local polynomial space-time descriptors for actions classification”. In : *Proceedings of the 13. IAPR International Conference on Machine Vision Applications, MVA 2013, Kyoto, Japan, May 20-23, 2013*. IAPR. 2013, pages 327–330.

[23] Olivier KIHL, David PICARD et Philippe-Henri GOSSELIN. “A unified formalism for video descriptors”. In : *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE. 2013, pages 2416–2419.

[24] David PICARD, Aymeric HISTACE et Marie-Charlotte DESSEROIT. “Joint MAS-PDE Modeling of Forest Pest Insect Dynamics : Analysis of the Bark Beetle’s Behavior”. In : *VISIGRAPP (Workshop GEODIFF)*. 2013, pages 29–38.

[25] Hedi TABIA, David PICARD, Hamid LAGA et Philippe-Henri GOSSELIN. “3D shape similarity using vectors of locally aggregated tensors”. In : *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE. 2013, pages 2694–2698.

[26] Hedi TABIA, David PICARD, Hamid LAGA et Philippe-Henri GOSSELIN. “Compact vectors of locally aggregated tensors for 3d shape retrieval”. In : *Eurographics workshop on 3D object retrieval*. 2013.

[27] Hedi TABIA, David PICARD, Hamid LAGA et Philippe-Henri GOSSELIN. “Fast Approximation of Distance Between Elastic Curves using Kernels”. In : *British Machine Vision Conference*. BMVA Press. 2013.

[28] Corina IOVAN, David PICARD, Nicolas THOME et Matthieu CORD. “Classification of urban scenes from geo-referenced images in urban street-view context”. In : *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*. Tome 2. IEEE. 2012, pages 339–344.

[29] Romain NEGREL, David PICARD et P GOSSELIN. “Compact Tensor Based Image Representation for Similarity Search”. In : *International Conference on Image Processing*. 2012.

[30] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. "Using Spatial Pyramids with Compacted VLAT for Image Categorization". In : *ICPR*. Tome 610. 2012.

[31] David PICARD, Nicolas THOME, Matthieu CORD et Alain RAKOTOMAMONJY. "Learning geometric combinations of Gaussian kernels with alternating Quasi-Newton algorithm". In : *ESANN 2012*. 2012, pages 79–84.

[32] David PICARD et Philippe-Henri GOSSELIN. "Improving Image Similarity With Vectors of Locally Aggregated Tensors". In : *Image Processing (ICIP), 2011 18th IEEE International Conference on*. 2011, pages–669.

[33] David PICARD, Nicolas THOME et Matthieu CORD. "An efficient system for combining complementary kernels in complex visual categorization tasks". In : *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE. 2010, pages 3877–3880.

[34] David PICARD, Matthieu CORD et Eduardo VALLE. "Study of SIFT Descriptors for Image Matching based Localization in Urban Street View Context". In : *CMRT09 - CityModels, Roads and Traffic*. GITC. 2009, pages 193–198.

[35] Eduardo VALLE, David PICARD et Matthieu CORD. "Geometric consistency checking for local-descriptor based document retrieval". In : *Proceedings of the 9th ACM symposium on Document engineering*. ACM. 2009, pages 135–138.

[36] David PICARD, Arnaud REVEL et Matthieu CORD. "Image retrieval over networks : Ant algorithm for long term active learning". In : *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*. IEEE. 2008, pages 439–445.

[37] David PICARD, Arnaud REVEL et Matthieu CORD. "Long term learning for image retrieval over networks". In : *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE. 2008, pages 929–932.

[38] David PICARD, Matthieu CORD et Arnaud REVEL. "Cbir in distributed databases using a multi-agent system". In : *Image Processing, 2006 IEEE International Conference on*. IEEE. 2006, pages 3205–3208.

[39] David PICARD, Arnaud REVEL et Matthieu CORD. "Performances of mobile-agents for interactive image retrieval". In : *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society. 2006, pages 581–586.

[40] Arnaud REVEL, David PICARD et Matthieu CORD. "Ant-like mobile agents for Content-Based Image Retrieval in distributed databases". In : 2005, page 29.

3. Détail des travaux de recherche

3.1 Apprentissage de représentations pour l'analyse d'images

Dans cette section, nous détaillons nos travaux dans le domaine de l'apprentissage de représentations pour l'analyse d'images. Nous nous sommes attaqués aux problèmes d'annotation automatique et de recherche par similarité dans des collections d'images. Nous avons porté une attention particulière aux méthodes qui passent à l'échelle, c'est à dire aux méthodes qui permettent de traiter des grands volumes de données.

Nous organisons la présentation de nos travaux en 3 parties complémentaires. La première partie concerne l'apprentissage de représentations d'images très discriminantes et se base sur la linéarisation des fonctions d'appariement de descripteurs locaux. La seconde partie explore l'incorporation de l'information spatiale dans les représentations d'images et peut être vue comme une extension de la première visant à préserver la disposition des motifs pertinents. La troisième partie concerne la compression de représentations. En effet, les travaux que nous avons menés donnent lieu à des représentations de très grande dimension qu'il est nécessaire de rendre plus compactes pour permettre leur exploitation dans de très grands volumes d'images (plusieurs millions d'images).

Ces travaux ont fait l'objet des encadrements en thèse de Romain Negrel et du démarrage de la thèse de Pierre Jacob, ainsi que du stage de M2 de Romain Carrara. L'article NEGREL, PICARD et P.-H. GOSSELIN 2013 est disponible en annexe.

3.1.1 Approximation tensorielle de noyaux d'appariement

Un premier axe de nos travaux de recherche sur l'apprentissage de représentations pour les images s'est concentré sur l'approximation d'appariements de descripteurs locaux par plongement dans un espace hilbertien bien choisi.

Nous nous sommes intéressés à la recherche de quasi-copies, c'est à dire le cas où l'image requête et l'image cible sont des clichés très similaires. Il peut s'agir par exemple du même objet pris sous des angles de vue différents, ou bien à des instants différents (jour et nuit, par exemple). Dans ce cas de figure, les méthodes classiques de recherche d'images par similarité qui offrent les meilleures performances sont souvent basées sur la recherche d'appariements de points d'intérêts

équipés de caractéristiques discriminantes. Encore aujourd'hui, et malgré les récents progrès du deep learning, ces méthodes reste état de l'art sur des jeux de données comme Holydays décrit dans Hervé JÉGOU, DOUZE et Cordelia SCHMID 2008.

Par exemple, la populaire méthode SIFT de LOWE 2004 extrait un grand nombre de régions d'intérêt (détecteur en différence de gaussiennes) et associe à chacune un vecteur de description contenant des histogrammes locaux d'orientation de gradients (information de géométrie des contours). Afin de calculer la similarité entre une image requête I_q et une image cible I_t, on parcourt chaque descripteur de la requête, et on détermine le plus proche voisin parmi les descripteurs de la cible en terme de distance euclidienne dans l'espace de description. On considère qu'il y a appariement (match en anglais) si cette distance au plus proche voisin respecte un certain critère (un seuil, ou un ratio par rapport à la distance au second plus proche voisin). On considère qu'une image cible est d'autant plus similaire à la requête qu'il y a d'appariements.

Plusieurs problèmes se posent avec ce type de méthodes. D'abord, elle sont très coûteuses. En effet, si nous extrayons 1k descripteurs par image (cas standard des SIFT), le calcul de la similarité fait intervenir 1M de distances dans l'espace des descripteurs pour chaque paire d'images que nous voulons comparer. Dans le cas où une extraction dense des descripteurs est effectuée (cas des réseaux de convolution), il n'est pas rare d'obtenir 10^5 descripteurs par image et donc plus de 10^{10} calculs de distances pour chaque paire d'images. D'autre part, cette similarité est asymétrique, c'est à dire que le nombre d'appariements entre I_q et I_t n'est pas le même qu'entre I_t et I_q. Elle ne respecte pas non plus l'inégalité triangulaire, ce qui donne lieu à des contradictions quand on utilise cette mesure pour agréger les résultats de différentes requêtes. Ces deux propriétés rendent l'exploitation de telle mesure impossible dans de nombreux algorithmes d'apprentissage statistique qui nécessitent d'avoir une similarité ayant des propriétés de produit scalaire.

Pour ces raisons, nous nous sommes intéressés à concevoir une mesure de similarité qui soit semi-définie positive (c'est à dire correspondant à un produit scalaire dans un espace induit par une transformation possiblement non-linéaire), qui soit rapide à l'évaluation (c'est à dire en mettant de côté tout pré-traitement hors-ligne) et qui soit aussi proche que possible de l'appariement de points d'intérêt pour bénéficier de leur puissance de discrimination.

Nous sommes partis d'une relaxation des méthodes d'appariement utilisant les noyaux sur sacs décrits dans SHAWE-TAYLOR et CRISTIANINI 2004. Une manière de compter le nombre d'appariements entre deux ensembles de descripteurs est de se doter d'une mesure de similarité entre descripteurs que l'on note $k(\cdot,\cdot)$ et de faire la sommes des similarités entre toutes les paires de descripteurs des deux images :

$$K(I_q, I_t) = \sum_{x_r \in I_q, x_s \in I_t} k(x_r, x_s) \tag{3.1}$$

Le choix de la fonction de noyau mineur k est crucial, car c'est elle qui permet au comptage d'être représentatif du nombre d'appariements. À titre d'exemple, en supposant que nous avons une fonction de quantification h qui à chaque descripteur associe un symbole issue d'un alphabet fini de telle sorte que des descripteurs proches dans l'espace de description soient associés à des symboles identiques, le noyau de quantification suivant permet d'apparier des descripteurs s'ils tombent dans la même zone de la partition :

$$k_q(x_r, x_s) = \delta(h(x_r), h(x_s)), \quad \delta(i,j) = 1 \text{ si } i = j, 0 \text{ sinon} \tag{3.2}$$

La quantification est obtenue par apprentissage non-supervisé en effectuant un partitionnement de l'espace des descripteurs, par exemple à l'aide de k-means. Le noyau somme doté de ce noyau mineur est alors équivalent à la méthode *Bag of Words* de SIVIC et A. ZISSERMAN 2003.

L'intérêt d'écrire l'appariement avec le noyau somme réside dans les cas où le noyau mineur peut être linéarisé. Dans le cas du noyau de quantification, il suffit de construire la transformation

H qui associe à un descripteur donné un vecteur binaire de la taille de l'alphabet de quantification, composé entièrement de zéros sauf pour la composante correspondante au symbole associé au descripteur. Le noyau mineur est alors simplement le produit scalaire entre les transformées des descripteurs :

$$k_q(x_r, x_s) = \langle H(x_r), H(x_s) \rangle \tag{3.3}$$

De fait, le noyau somme peut alors être simplifié en utilisant la linéarité du produit scalaire :

$$K(I_q, I_t) = \left\langle \sum_{x_r \in I_q} H(x_r), \sum_{x_s \in I_t} H(x_s) \right\rangle \tag{3.4}$$

Le traitement $\sum_{x_r \in I} H(x_r)$ peut alors être fait hors ligne, et la comparaison entre deux images revient à faire le produit scalaire entre deux vecteurs.

Plutôt que de considérer uniquement le noyau de quantification, nous nous sommes intéressés au noyau gaussien qui à l'avantage d'être régulier et donc de ne pas présenter d'effets de bord dûs à une quantification. Nous avons considéré le développement de Taylor de l'exponentielle :

$$k_e(x_r, x_s) = e^{-\gamma \|x_r - x_s\|^2} = e^{-\gamma \|x_r\|^2} e^{-\gamma \|x_s\|^2} e^{2\gamma \langle x_r, x_s \rangle} \tag{3.5}$$

$$= e^{-\gamma \|x_r\|^2} e^{-\gamma \|x_s\|^2} \sum_p^{\infty} \alpha_p \langle x_r, x_s \rangle^p \tag{3.6}$$

$$= e^{-\gamma \|x_r\|^2} e^{-\gamma \|x_s\|^2} \sum_p^{\infty} \alpha_p \langle x_r^{\otimes p}, x_s^{\otimes p} \rangle \tag{3.7}$$

où $x^{\otimes p}$ désigne le produit tensoriel d'ordre p de x avec lui-même et est obtenu par définition même du produit tensoriel. Si on considère des descripteurs normés, alors le noyau gaussien est directement proportionnel aux produits scalaires des tenseurs d'ordre p des descripteurs. En pratique, on s'arrête à un ordre p relativement petit (souvent $= 2$), d'une part car l'erreur par rapport à la gaussienne est suffisamment petite et d'autre part car cela limite la taille des tenseurs à stocker. Ainsi, le noyau somme peut être échangé avec le produit scalaire et donner la similarité image suivante :

$$K(I_1, I_2) = \left\langle \left[\sum_{x_r \in I_1} vec(x_r^{\otimes p}) \right]_p, \left[\sum_{x_r \in I_2} vec(x_s^{\otimes p}) \right]_p \right\rangle \tag{3.8}$$

avec $[\cdot]$ la concaténation et $vec()$ le déroulement du tenseur obtenu en vecteur de dimension équivalente.

Pour finir, la méthode que nous avons proposée (VLAT, pour *Vector of Locally Aggregated Tensors*) combine le noyau de quantification avec le noyau gaussien et utilise une fois de plus la définition du produit tensoriel pour linéariser le résultat :

$$K_{q \times e}(I_1, I_2) = \sum_{x_r \in I_1, x_s \in I_2} k_q(x_r x_s) k_e(x_r, x_s) \tag{3.9}$$

$$\approx \sum_{x_r \in I_1, x_s \in I_2} \langle H(x_r), H(x_s) \rangle \left\langle \left[vec(x_r^{\otimes p}) \right]_p, \left[vec(x_s^{\otimes p}) \right]_p \right\rangle \tag{3.10}$$

$$= \left\langle \sum_{x_r \in I_1} H(x_r) \otimes \left[vec(x_r^{\otimes p}) \right]_p, \sum_{x_s \in I_2} H(x_s) \otimes \left[vec(x_s^{\otimes p}) \right]_p \right\rangle \tag{3.11}$$

Où la somme peut être effectuée en pré-traitement et la comparaison d'images est un simple produit scalaire sur les vecteurs obtenus. Il s'agit alors d'un noyau doté d'une fonction d'injection explicite.

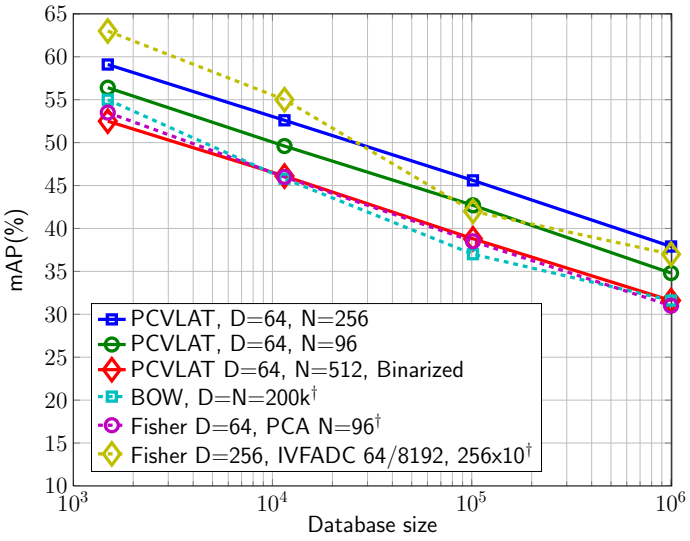


FIGURE 3.1: Évolution du mAp en fonction du nombre de distracteurs ajoutés dans la base Holidays.

Nous avons remarqué que les représentations obtenues sont plus discriminantes en introduisant différents centrages, ce qui s'explique par un gain de dynamique pour le produit scalaire. En nous limitant à l'ordre 2, nous avons alors la représentation VLAT suivante :

$$VLAT(I) = \sum_{x \in I} H(x) \otimes vec((x - h(x))^{\otimes 2} - \sigma_{h(x)}) \tag{3.12}$$

Où $\sigma_{h(x)}$ correspond à la matrice de covariance de la zone de la partition associée à x.

Dans PICARD et P.-H. GOSSELIN 2013 ; NEGREL, PICARD et P. GOSSELIN 2012 ; PICARD et P.-H. GOSSELIN 2011, nous avons montré que cette représentation pouvait obtenir des résultats du niveau de l'état de l'art tant en recherche de quasi-copies par similarité (jeu de données Holidays), qu'en annotation automatique d'images (jeu de données PASCAL VOC 2007). La figure 3.1 montre la précision moyenne (mAp) sur le jeu de données Holidays en fonction de la taille de la collection considérée et pour de petits dictionnaires (64 symboles). Pour faire des tests de passage à l'échelle, nous ajoutons des images de bruit au jeu de données original. Les résultats obtenus sont très compétitifs avec les méthodes de l'état de l'art au moment de ces publications, notamment les Fisher Vectors de F. PERRONNIN et al. 2010.

Dans le cas de l'annotation automatique, la représentation obtenue est alors simplement utilisée en entrée d'un classifieur (par exemple un SVM), et nous avons observé que cette représentation est suffisamment discriminante pour obtenir de bonnes performances avec un classifieur linéaire. L'essentiel de ces travaux ont été effectués lors de l'encadrement de la thèse de Romain Negrel.

3.1.2 Information spatiale

Nous nous sommes ensuite intéressés à l'intégration de l'information spatiale dans les représentations issues de l'approximation des noyaux d'appariement. En effet, lors que nous sommons les transformées (produit tensoriel, vectorisation, concaténation, etc) des différents descripteurs, la localisation de ceux-ci dans l'image est perdue. Or, dans les méthodes d'appariement de descripteurs, un post-traitement est souvent effectué afin de garantir l'intégrité géométrique des appariements trouvés. En effet, si on considère que les objets présents dans l'image sont indéformables (voire même raisonnablement déformables), alors si deux descripteurs sont proches dans l'image requête, leurs appariements dans l'image cible devraient être proches.

Une méthode classique de l'état de l'art pour prendre en compte la disposition des descripteurs dans l'image consiste à découper celle-ci en pyramide de grilles spatiales décrite dans LAZEBNIK,

Cordelia SCHMID et PONCE 2006. Une représentation est calculée dans chaque cellule de la grille de chaque échelle et celles-ci sont concaténées. Ainsi, les descripteurs vont d'autant plus contribuer à la similarité entre deux images qu'ils sont localisés au même endroit dans l'image. Cependant, cette méthode n'est valide que pour des dispositions de scène (ce pourquoi elle fut proposée) et en aucun cas pour des objets qui peuvent être à n'importe quel endroit dans l'image.

Plutôt que de nous intéresser à la disposition globale de l'image, nous nous somme intéressés à la disposition locale des descripteurs autour d'un point d'intérêt. Nous avons repris l'idée que si deux descripteurs sont proches dans l'image requête, alors leurs appariements sont proches dans l'image cible, indépendamment de leur position absolue. Pour rendre compte de la notion de voisinage, nous définissons un support spatial $\Omega(x)$ qui correspond à un ensemble de descripteurs voisins de x, par exemple situés dans un rayon de n pixels autour de x. Nous avons alors proposé le noyaux mineur suivant qui lui-même se base sur un noyau mineur générique k :

$$k_{STA}(x_r,x_s) = k(x_r,x_s) \sum_{x_u\in\Omega(x_r),x_v\in\Omega(x_s)} k(x_u,x_v) \tag{3.13}$$

Ainsi, la similarité entre x_s et x_r est pondérée par un mini noyau d'appariement entre les voisinages de x_r et x_s. Cette similarité est donc d'autant plus élevée que les voisinages respectifs de x_r et x_s peuvent être appariés.

Comme noyau mineur, nous avons considéré le produit entre le noyau de quantification et le produit scalaire, ce qui donne le noyau somme suivant :

$$K_{STA}(I_1,I_2) = \sum_{x_r\in I_1,x_s\in I_2} k_q(x_r,x_s)\langle x_r,x_s\rangle \sum_{x_u\in\Omega(x_r),x_v\in\Omega(x_s)} k_q(x_u,x_v)\langle x_u,x_v\rangle \tag{3.14}$$

$$= \left\langle \sum_{x_r\in I_1}\sum_{x_u\in\Omega(x_r)} H(x_r)\otimes H(x_u)\otimes x_r\otimes x_u, \sum_{x_s\in I_2}\sum_{x_v\in\Omega(x_s)} H(x_s)\otimes H(x_v)\otimes x_s\otimes x_v \right\rangle \tag{3.15}$$

De même que pour les VLAT, la somme peut être calculée en pré-traitement hors ligne ce qui donne une représentation que nous avons appelée STA pour *Spatial Tensor Aggregation*. Nous avons montré dans PICARD 2016 que cette représentation pouvait donner des résultats très compétitifs par rapport à l'état de l'art, tout en étant simple d'implémentation. En particulier, nous avons pu utiliser avec succès aussi bien des descripteurs locaux de type SIFT ou équivalent que des sorties de réseaux de neurones convolutionnels.

Enfin, nous nous sommes intéressés à simplifier le choix du support spatial Ω en tentant de l'apprendre sur les données. En effet, plutôt que fixer un voisinage arbitrairement, nous avons considéré le cas où celui-ci peut être appris statistiquement sur les données. Nous avons proposé d'introduire une pondération dans le support :

$$k_{STA_\alpha}(x_r,x_s) = k(x_r,x_s) \sum_{x_u\in\Omega(x_r),x_v\in\Omega(x_s)} \alpha(u-r)\alpha(v-s)k(x_u,x_v) \tag{3.16}$$

Dans lequel $\alpha(u-r)$ désigne le poids associé à la position correspondante au décalage entre x_r et x_u.

Ceci donne lieu à une représentation d'image dont la définition de voisinage est apprise :

$$STA_\alpha(I) = \sum_{x_r\in I}\sum_{x_u\in\Omega(x_r)} \alpha(u-r)H(x_r)\otimes H(x_u)\otimes x_r\otimes x_u \tag{3.17}$$

Par contre, il est trop complexe d'apprendre les α de manière non-supervisée. Dans le cas de la classification d'images, nous pouvons simplement alterner l'optimisation du classifieur avec celle de α, dans le même style que l'optimisation de SVM structurés. Dans le cas de la recherche par

similarité, nous optimisons les α avec une fonction de coût par triplet qui cherche à maximiser la similarité entre images correspondantes tout en minimisant la similarité entre images non-correspondantes.

Ces travaux très récent sur l'apprentissage du support spatial on été fait dans le cadre de l'encadrement en master de Romain Carrara (stage), et Nicolas Benoît (projet) et du démarrage de la thèse de Pierre Jacob et sont en cours de publication.

3.1.3 Compression des représentations

Enfin, nous nous sommes intéressés au passage à l'échelle des méthodes précédemment proposées. En particulier, si la linéarisation des noyaux d'appariement permet d'utiliser un simple produit scalaire comme mesure de similarité, cela n'est possible qu'à un coût de stockage très élevé. Le produit tensoriel faisant croître exponentiellement la dimension, les représentations obtenues sont souvent plongées dans des espaces de plusieurs centaines de milliers, voire millions, de dimensions. Ces très grandes dimensions impliquent que le stockage des représentations de toute une collection ne tienne plus en mémoire centrale mais soit fait sur disque, ce qui donne des performances catastrophiques lors d'une requête. Pour remédier à cela, nous avons proposé de réduire fortement la taille des représentations.

Dans un premier temps, nous avons proposé une méthode qui repose sur deux étapes de réduction de dimension. Une pré-projection réduit la dimension des descripteurs avant la montée en tenseurs, tandis-qu'une post-projection réduit la dimension des représentations obtenues après la montée en tenseur. Pour la pré-projection, il s'agit de conserver le maximum d'information par rapport aux descripteurs initiaux, ce qui correspond à une analyse en composante principale. Afin de minimiser la perte d'information, nous avons proposé d'effectuer cette projection pour chaque partie de la quantification de l'espace des descripteurs, ce que nous avons appelé *cluster-wise PCA*. Ceci donne alors la signature VLAT suivante :

$$VLAT_{wpca}(I) = \sum_{x \in I} H(x) \otimes vec((U_{h(x)}^{\top}(x - h(x)))^{\otimes 2} - L_{h(x)}) \tag{3.18}$$

Avec U_h, L_h les vecteurs/valeurs propres de la matrice de covariance de la partition associée à x.

Pour la post-projection, nous avons considéré une projection linéaire de telle sorte que l'erreur moyenne entre le produit scalaire calculé dans l'espace de départ et celui calculé dans l'espace projeté soit minimale. On peut remarquer que cela revient à faire une approximation de rang faible de la matrice de Gram, ou de manière équivalente à résoudre un problème de kernel PCA avec un noyau linéaire, c'est à dire une analyse en composante principale sans centrage des données. En notant X la matrice des représentations VLAT en vecteurs colonnes pour une collection d'images, la projection est alors :

$$P_t = X U_t L_t^{-\frac{1}{2}} \tag{3.19}$$

Avec $X^{\top}X = ULU^{\top}$ et U_t les t vecteurs propres associés au t valeurs propres L_t de plus grande magnitude. Les projetés sont simplement les $y = P_t^{\top}X$.

L'avantage majeur de cette post-projection est qu'elle permet de choisir une taille cible raisonnable (par exemple 1024 dimensions), et nous avons montré qu'elle obtenait d'excellents résultats en recherche par similarité à des points de fonctionnement de très petite taille dans NEGREL, PICARD et P.-H. GOSSELIN 2013. Cependant, la matrice de projection P est extrêmement volumineuse, en particulier dans les cas ou on souhaite conserver plusieurs milliers de dimensions en sortie, ce qui mène à des modèles pouvant faire plusieurs dizaines de Go. Ceci est particulièrement dommageable dès lors que le modèle doit être déployé (par exemple transmis à travers un réseau).

Pour s'abstraire du coût exorbitant de la projection linéaire tant en stockage qu'en calcul, nous avons proposé des matrices de projection creuses. Pour cela, nous avons considéré le problème

d'optimisation suivant :

$$\min_P \mathbb{E}\left[\|X_i^\top X_j - X_i^\top PP^\top X_j\|^2\right] \text{ tel que } \|P\|_0 = m \tag{3.20}$$

Avec X_i, X_j des représentations VLAT, et m le nombre de composantes non nulles dans la matrice de projection P. Bien sûr, un tel problème de minimisation sous contrainte de norme ℓ_0 est particulièrement complexe à résoudre de manière exacte. Nous avons proposé deux méthodes de résolution approchée, l'une non-supervisée et l'autre supervisée.

La méthode non-supervisée propose de découper la projection creuse en deux sous-parties composées d'une réduction de dimension creuse P_c peu coûteuse mais qui introduit une erreur par rapport à la projection pleine P_t et d'une projection pleine correctrice R dans l'espace réduit :

$$P = RP_c \text{ tel que } \|P_c\|_0 = m \tag{3.21}$$

Chacune de ces matrices est associée à un problème propre beaucoup plus simple à résoudre :

$$\min_{P_c} \mathbb{E}\left[\|P_t - P_c\|^2\right] \text{ tel que } \|P_c\|_0 = m \tag{3.22}$$

$$\min_R \mathbb{E}\left[\|X_i^\top P_t P_t^\top X_j - X_i^\top P_c R R^\top P_c^\top X_j\|^2\right] \tag{3.23}$$

La projection creuse P_c la plus semblable à la projection pleine initiale est obtenue trivialement par seuillage des valeurs de plus petite magnitude. Pour la correction R, nous avons proposé une formule analytique :

$$R = VD^{\frac{1}{2}} \tag{3.24}$$

Avec V, D les vecteurs/valeurs propres de :

$$W = Y^+ X^\top PP^\top XY^{+\top} \tag{3.25}$$

et $Y^+ = (YY^\top)^{-1}Y$ la pseudo inverse des projetés pleins $Y = P_t^\top X$. Nous avons montré qu'avec cette méthode, il était possible de réduire le coût de stockage et de projection par un facteur 1000 tout en ayant une perte de qualité des résultats négligeable par rapport à un projection pleine. La figure 3.2 montre l'évolution du mAp en fixant la dimension de sortie à 256 et en fonction du taux de parcimonie des projecteurs (comptabilisé en nombre d'opérations nécessaires à la projection). On peut remarquer que de nombreux points de fonctionnement se situent au dessus de 80% de mAp, ce qui est un très bon score pour des méthodes basées sur des descripteurs locaux de type SIFT. Ceci est d'autant plus vrai qu'on ne conserve que 256 dimensions pour la représentation finale.

La méthode supervisée consiste à apprendre directement une projection creuse avec une fonction objective de triplet. Pour cela, nous divisons pour chaque requête q l'ensemble des images disponibles en deux sous-ensemble $\mathscr{P}_q$ pour les images similaires à la requête et $\mathscr{N}_q$ pour celles qui lui sont dissimilaires. Nous voulons alors minimiser la fonction objective suivante correspondant aux erreurs de tri entre images positives et négatives :

$$\min_P \sum_q \sum_{p\in\mathscr{P}_q} \sum_{n\in\mathscr{N}_q} \left[1 + X_n^\top PP^\top X_q - X_p^\top PP^\top X_q\right]_+ \text{ tel que } \|P\|_0 = m \tag{3.26}$$

Avec $[x]_+ = \max(0, x)$. Le problème d'une telle fonction objective est qu'elle introduit un nombre cubique de terme, ce qui devient rapidement impossible à résoudre correctement, même en échantillonnant les triplets.

Pour résoudre cette difficulté, nous avons remarqué que tous les triplets n'étaient pas nécessaires à la résolution du problème. En effet, si l'élément le plus similaire à la requête parmi les négatifs

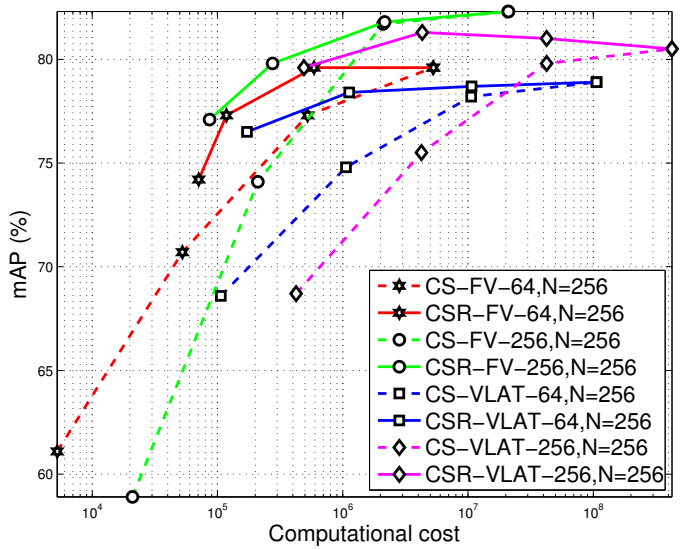


FIGURE 3.2: Évolution du mAp sur Holidays en fonction de la complexité des projecteurs induite par le taux de parcimonie choisi. La courbe en pointillés correspond à la projection creuse sans correction.

possède un score de similarité plus faible que tous les positifs, alors le tri est parfait. Nous avons donc proposé de se concentrer sur cet élément que nous appelons *pivot*. Le problème devient alors linéaire en $\mathscr{P}_q$ puisqu'il ne s'agit plus que de faire passer tous les positifs devant le pivot négatif. Ceci est fait en utilisant une descente de gradient stochastique dans laquelle nous échantillonnons des exemples positifs violant la contrainte du pivot pour un certain nombre de requêtes à chaque itération. Périodiquement, le pivot est recalculé et la contrainte de norme ℓ_0 est respectée par projection (seuillage).

Nous montrons en figure 3.3 les évolutions sur l'ensemble d'entraînement de la fonction objectif et du mAp pour différents degrés de parcimonie des projecteurs sur la base Holidays. Ces résultats nous montrent d'une part que la fonction objectif proposée permet bien de maximiser le mAp et d'autre part que la méthode de gradient projeté permet d'optimiser convenablement cette fonction. Comme le montrent les courbes, plus la contrainte de parcimonie est forte, plus il est difficile de trouver des projecteurs qui permettent un tri parfait des exemples d'apprentissage. D'autre part, le fait que le problème soit non convexe semble ne pas avoir de gros impact sur la qualité de la solution trouvée par descente de gradient projeté.

Tous ces travaux sur la compression de représentation ont été effectués lors de l'encadrement de la seconde partie de la thèse de Romain Negrel et ont été publiés dans NEGREL, PICARD et P.-H. GOSSELIN 2014a ; NEGREL, PICARD et P.-H. GOSSELIN 2014b.

3.2 Apprentissage de représentations pour l'analyse de vidéos

Dans cette section, nous détaillons nos travaux sur l'apprentissage de représentations pour l'indexation et l'interprétation de contenus vidéo. Comme pour les images, nous nous sommes intéressés aux problèmes d'annotation automatique, en particulier à la reconnaissance d'actions ou d'activités dans les vidéos.

La présentation de ces travaux s'articule en deux grands axes qui attaquent le problème par des angles différents, mais complémentaires. Dans un premier temps, nous nous sommes attachés à déterminer le type d'information de bas niveau qui semblait le plus performant pour réaliser l'interprétation de contenus vidéo. Dans un second temps, nous nous sommes intéressés

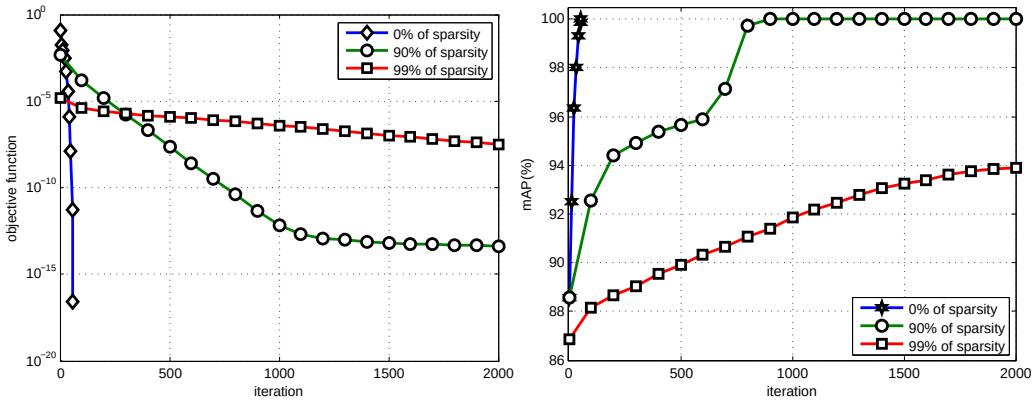


FIGURE 3.3: Évolution de la valeur de la fonction objectif (gauche) et du mAp en fonction du nombre d'itérations pour différent taux de parcimonie.

à l'utilisation d'information structurée pour la reconnaissance d'activité, notamment en étendant les modèles d'agrégation développés pour les images à une information possédant à la fois une structure spatiale et temporelle.

3.2.1 Modélisation des descripteurs bas niveau

Dans un premier temps, nous nous sommes intéressés à comprendre l'impact de différentes informations sur les performances de reconnaissance dans les vidéos. En effet, il existe en vidéo une multitude de descripteurs bien plus grande que pour les images. Par exemple, on peut prendre en compte l'apparence uniquement ce qui revient à considérer une vidéo comme un ensemble non ordonné d'images. On peut évidemment lui ajouter l'information temporelle, auquel cas les variations d'apparence dans le temps peuvent apporter une information supplémentaire très pertinente.

La conception de descripteurs bas-niveaux pour la vidéo résulte du choix d'un ensemble d'opérations et de traitements particuliers parmi l'ensemble des outils disponibles en traitement du signal et en apprentissage statistique. Afin de clarifier l'importance de chacun de ces choix, nous avons proposé dans KIHL, PICARD et P.-H. GOSSELIN 2015 (disponible en annexe) un cadre d'étude formel qui permet de représenter la grande majorité des descripteurs bas-niveau utilisés en analyse de vidéo. Ce cadre d'étude décompose la conception de descripteurs bas-niveau en trois étapes :

1. Primitive : Choix du type d'information locale considérée
2. Coding : Choix d'un encodage pour cette information
3. Aggregation : Choix d'une manière de résumer les codes locaux afin de produire un descripteur de région

Nous détaillons ci-après chacune de ces étapes.

Primitive

Au niveau de la primitive, il s'agit d'extraire un type d'information bas-niveau spécifique avec pour objectif de détecter certaines propriétés locales du signal (image ou vidéo). Généralement, cette étape est réalisée par un filtrage en haute fréquence et le choix de l'information pertinente à conserver introduit une perte par rapport au signal de départ, perte que l'on estime être négligeable, voire bénéfique.

Les descripteurs locaux populaires utilisent comme primitive l'information de gradient (SIFT ou HOG dans Navneet DALAL et Bill TRIGGS 2005 ; LOWE 2004), des bancs de filtres comme les ondelettes de Haar (SURF dans BAY et al. 2008), le flot optique (HOF dans N. DALAL, B.

TRIGGS et C. SCHMID 2006) ou le gradient du flot optique (MBH dans N. DALAL, B. TRIGGS et C. SCHMID 2006).

Coding

L'étape d'encodage consiste à plonger de manière non-linéaire les primitives dans un espace de plus grande dimension. L'objectif est d'améliorer la représentation en groupant ensemble les primitives similaires et en orthogonalisant les primitives dissimilaires.

Dans la littérature des descripteurs locaux, l'encodage le plus courant consiste à quantifier les orientations du champ de vecteur de la primitive (généralement 8 orientations), comme c'est le cas dans SIFT, HOG, HOF et MBH. Dans SURF, il s'agit d'un codage en valeur absolue qui introduit de la redondance :

$$\begin{aligned}
\mathscr{A}(\mathbf{x},0) &= G_x(\mathbf{x}) \\
\mathscr{A}(\mathbf{x},1) &= G_y(\mathbf{x}) \\
\mathscr{A}(\mathbf{x},2) &= |G_x(\mathbf{x})| \\
\mathscr{A}(\mathbf{x},3) &= |G_y(\mathbf{x})|
\end{aligned}$$

Enfin, un codage par rectification est proposé dans EFROS et al. 2003 :

$$\mathscr{R}(\mathbf{x},0) = \begin{cases} \mathscr{U}(\mathbf{x}) & \text{if } \mathscr{U}(\mathbf{x}) > 0 \\ 0 & \text{else} \end{cases}$$

$$\mathscr{R}(\mathbf{x},1) = \begin{cases} |\mathscr{U}(\mathbf{x})| & \text{if } \mathscr{U}(\mathbf{x}) < 0 \\ 0 & \text{else} \end{cases}$$

$$\mathscr{R}(\mathbf{x},2) = \begin{cases} \mathscr{V}(\mathbf{x}) & \text{if } \mathscr{V}(\mathbf{x}) > 0 \\ 0 & \text{else} \end{cases}$$

$$\mathscr{R}(\mathbf{x},3) = \begin{cases} |\mathscr{V}(\mathbf{x})| & \text{if } \mathscr{V}(\mathbf{x}) < 0 \\ 0 & \text{else} \end{cases}$$

Ces encodages on pour effet de multiplier par 8 ou 4 la dimension de la primitive d'entrée. Nous montrons certaines images de primitives encodées en figure 3.4.

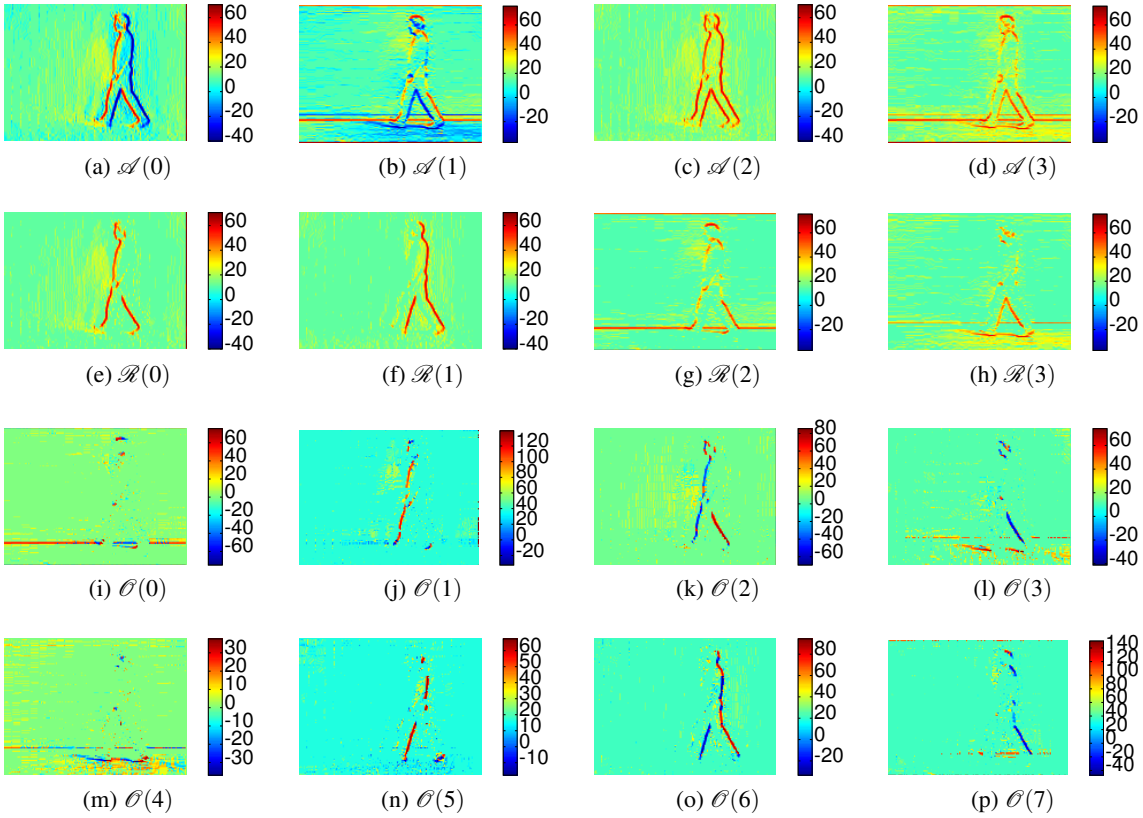
Aggregation

Enfin, l'agrégation modélise la répartition spatial des primitives encodées dans la région d'intérêt considérée. L'objectif est d'ajouter de la robustesse aux descripteurs en autorisant des appariements inexacts entre des régions déformées d'images ou de vidéos. L'agrégation la plus courante consiste à calculer la moyenne de chaque composante des codes dans les cellules d'une grille carrée/cubique (typiquement 4×4 cellules de 10 pixels), ce qui est le cas de HOG, HOF et leurs variantes.

Nous avons proposé un nouveau mode d'agrégation basé sur des fonctions oscillantes (polynômes dans KIHL, PICARD et P.-H. GOSSELIN 2016 et sinusoïdes dans KIHL, PICARD et P.-H. GOSSELIN 2015), permettant d'améliorer les descripteurs de l'état de l'art par rapport à l'agrégation en grille.

Le choix d'une primitive, d'un codage et d'une agrégation donne ainsi un type de descripteur local aux propriétés quasiment uniques. Notons d'ailleurs qu'une couche d'un réseau de neurones convolutifs peut s'interpréter comme un enchaînement Primitive/Coding/Aggregation dans lequel la primitive est le banc de filtres de convolution, le codage est souvent une simple rectification (ReLU) et l'agrégation est généralement un max-pooling dans des cellules 2×2.

Pour choisir les descripteurs pertinents parmi la myriade que notre cadre permet d'explorer, nous avons proposé d'utiliser des machines à noyaux multiples. Chaque descripteur local permet de construire une signature de la vidéo (par exemple à l'aide de Fisher Vector, ou des VLAT décrits



(a) $\mathscr{A}(0)$ (b) $\mathscr{A}(1)$ (c) $\mathscr{A}(2)$ (d) $\mathscr{A}(3)$

(e) $\mathscr{R}(0)$ (f) $\mathscr{R}(1)$ (g) $\mathscr{R}(2)$ (h) $\mathscr{R}(3)$

(i) $\mathscr{O}(0)$ (j) $\mathscr{O}(1)$ (k) $\mathscr{O}(2)$ (l) $\mathscr{O}(3)$

(m) $\mathscr{O}(4)$ (n) $\mathscr{O}(5)$ (o) $\mathscr{O}(6)$ (p) $\mathscr{O}(7)$

FIGURE 3.4: Exemples de coding ; première ligne : Absolute coding du gradient ($\mathscr{A}(0)$, $\mathscr{A}(1)$, $\mathscr{A}(2)$, $\mathscr{A}(3)$; seconde ligne : Rectified coding du gradient ($\mathscr{R}(0)$, $\mathscr{R}(1)$, $\mathscr{R}(2)$, $\mathscr{R}(3)$) ; troisième et quatrième ligne : Orientation coding du gradient ($\mathscr{O}(0)$, $\mathscr{O}(1)$, $\mathscr{O}(2)$, $\mathscr{O}(3)$, $\mathscr{O}(4)$, $\mathscr{O}(5)$, $\mathscr{O}(6)$, $\mathscr{O}(7)$)

dans la section 3.1). Une telle signature f munie du produit scalaire permet de définir un noyau explicite $k_f(x_1,x_2)$ entre 2 vidéos x_1 et x_2. Le problème d'apprentissage lié à la classification des vidéos est alors un problème classique de *Multiple Kernel Learning* (MKL) où on apprend un noyau combiné $\sum_f \beta_f k_f(\cdot,\cdot)$ avec une contrainte de parcimonie $\|\beta\|_1 = 1$ sur les poids de la combinaison :

$$\begin{aligned}
&\min_{\beta}\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\sum_f \alpha_i\alpha_j y_i y_j \beta_f k_f(x_i,x_j) \\
&\quad \text{s.t. } \forall f, \beta_f \geq 0 \\
&\qquad \sum_f \beta_f = 1 \\
&\qquad \forall i, 0 \leq \alpha_i \leq C
\end{aligned}$$

Ce problème est résolu à l'aide de la bibliothèque de machines à noyaux *JKernelMachines* que nous avons développée dans PICARD, THOME et CORD 2013 à l'occasion de nos travaux de recherche sur les machines à noyaux.

Nous présentons en table 3.1 les résultats obtenus par les meilleures combinaisons de descripteurs locaux explorées par notre approche sur le jeu de données Hollywood2. Ces résultats sont du niveau de l'état de l'art au moment de leur publication, ce qui montre l'importance de bien choisir les descripteurs sur lesquels s'appuient les méthodes d'agrégation. Ces résultats sont d'autant plus

Method	ND	NL	Results
Gilbert GILBERT, ILLINGWORTH et BOWDEN 2011	3	X	50.9%
Ullah ULLAH, PARIZI et LAPTEV 2010 HOG+HOF	2	X	51.8%
Wang H. WANG et al. 2011 traj	1	X	47.7%
Wang H. WANG et al. 2011 HOG	1	X	41.5%
Wang H. WANG et al. 2011 HOF	1	X	50.8%
Wang H. WANG et al. 2011 MBH	1	X	54.2%
Wang H. WANG et al. 2011 all	4	X	**58**.3%
baseline HOG (2,3)	1		51,4%
baseline HOF (2,3)	1		49,5%
baseline MBH (2,3)	1		53,4%
A = G + ori + Cell (2,4)	1		**52**,**0**%
B = M + rect + Poly (2,3)	1		**54**,**5**%
C = GM + rect + Poly (2,3)	1		**57**,**0**%
baseline	3		57,2%
A+B+C	3		**60**.3%

TABLE 3.1: mAp sur le jeu Hollywood2 ; ND : nombre de descripteurs ; NL : classifieurs non-linéaires.

intéressant que nous avons utilisé des classifieurs linéaires : les noyaux de la combinaison MKL étant explicites, nous pouvons exprimer le classifieur final comme la concaténation pondérée des hyperplans correspondant à chaque descripteur.

3.2.2 Information structurée

Dans un second temps, nous nous sommes intéressés lors du début de thèse de Diogo Luvizon à la reconnaissance d'actions dans des vidéos en se basant sur de l'information structurée. En particulier, nous nous sommes intéressés à prédire l'action effectuée par un sujet à partir de la série temporelle des coordonnées de son squelette 3D. Ce squelette, composé d'un graphe de parties humaines (tête, épaules, coudes, mains, etc), peut être obtenu directement avec les kits de développement de certaines caméras comme la Kinect.

La chaîne de traitement que nous avons proposée (*cf* figure 3.5) repose sur une amélioration des méthodes d'agrégation de descripteurs locaux en image. En effet, une vidéo est composée d'une séquence temporelle de positions 3D pour chaque partie du squelette. Nous découpons ces positions à l'aide d'une fenêtre glissante temporelle ce qui permet d'associer un sac de trajectoires locales à une vidéo. Ces trajectoires sont alors analogues aux descripteurs locaux en reconnaissance d'images, que nous avons proposé d'agréger avec un modèle de *Bag Of Words* avancé (VLAD de H. JÉGOU et al. 2010).

Le problème des méthodes de *Bag Of Words* est leur sensibilité au dictionnaire utilisé pour l'encodage qui peut avoir une grande influence sur les performances finales. Pour réduire cet effet, nous avons proposé l'utilisation de plusieurs dictionnaires calculés indépendamment les uns des autres et donnant lieux à des représentations différentes.

Pour combiner ces différentes représentations, nous avons alors proposé une approche basée sur l'apprentissage de métrique (analogue à LMNN de WEINBERGER et SAUL 2009) qui nous permet d'entraîner de manière supervisée une succession de projections linéaires $\mathbf{L}$ opérant sur la concaténations des représentations issues des différents dictionnaires. L'idée est de trouver une projection telle que les données projetées d'une même classe soient proches (au sens de la métrique apprise $D_{\mathbf{L}}(x_i,x_j) = \|\mathbf{L}(x_i - x_j)\|$), alors que les données projetées de classes différentes soient

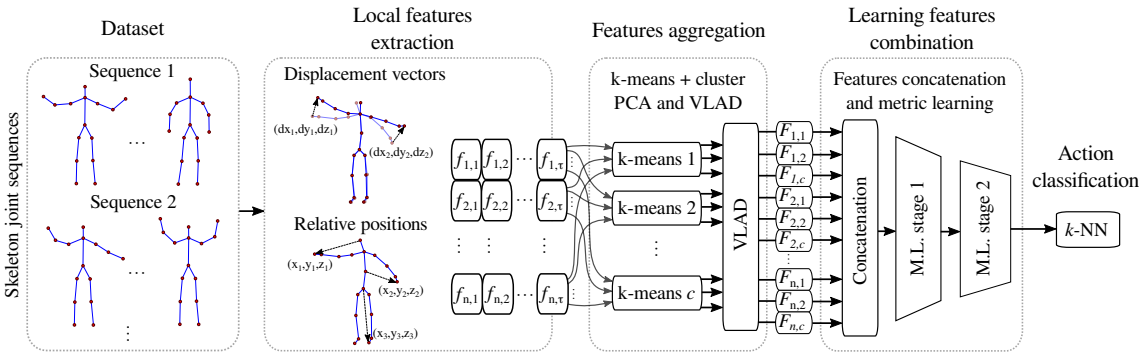


FIGURE 3.5: Chaîne de classification d'action dans des vidéos basée sur le squelette 3D.

éloignées. Ceci se traduit par deux forces opposées que nous nommons *push* et *pull* :

$$\mathscr{E}_{pull}(\mathbf{L}) = \sum_{j \to i} D_{\mathbf{L}}(x_i, x_j)$$
$$\mathscr{E}_{push}(\mathbf{L}) = \sum_{i,j \to i} \sum_{l \not\to i} \max(0, \xi + D_{\mathbf{L}}(x_i, x_j) - D_{\mathbf{L}}(x_i, x_l))$$

Avec la notation $j \to i$ indiquant que l'exemple j appartient à la même classe que i et $l \not\to i$ indiquant que l'exemple l n'appartient pas à la même classe que i.

Le problème d'optimisation que nous voulons résoudre est consiste à minimiser ces deux forces d'attraction/répulsion augmentées d'un terme de régularisation pour éviter d'apprendre des projections trop complexes :

$$\min_{\mathbf{L}} \mathscr{E}(\mathbf{L}) = (1-\mu)\mathscr{E}_{pull}(\mathbf{L}) + \mu \mathscr{E}_{push}(\mathbf{L}) + \gamma \|\mathbf{L}^\top \mathbf{L} - \mathbf{I}\|^2$$

En pratique, la résolution de ce problème d'optimisation est obtenue par descente de gradient stochastique lors de laquelle le gradient est estimé sur un sous-ensemble d'échantillons. Nous entraînons deux projections, la première servant à réduire fortement la dimension des représentations, alors que la seconde sert à faire les regroupements par classe. Cette séparation en plusieurs projections permet de réduire d'autant plus les effets de sur-apprentissage tout en n'ajoutant aucun coût en évaluation grâce à la combinaison $\mathbf{L} = \mathbf{L}_2\mathbf{L}_1$.

Ces projections ont trois objectifs, d'abord réduire la dimension des représentations (devenue importante à cause de la multiplicité des dictionnaires), ensuite il s'agit de supprimer la redondance introduite par l'utilisation de plusieurs dictionnaires potentiellement corrélés, enfin il s'agit de ne garder que les composantes des représentations qui sont utiles à l'objectif de classification.

L'utilisation de plusieurs dictionnaires lors de l'encodage couplé à l'apprentissage de métrique pour la combinaison nous a permis d'augmenter la qualité des résultats de classification tout en réduisant la variance sur les performances du système. Ces travaux ont été publiés dans LUVIZON, TABIA et PICARD 2017.

Nous présentons dans la table 3.2 les résultats obtenus par notre méthode sur trois jeux de données de la littérature, à savoir MSR-Action3D, UTKinect-Action3D et Florence 3D Actions. Comme nous pouvons le remarquer, notre méthode surpasse toutes celles trouvées dans la littérature au moment de la soumission de l'article. Ceci est dû à la combinaison de plusieurs facteurs comme la réduction de la variance par l'utilisation de plusieurs dictionnaires visuels et la sélection d'attributs pertinents par apprentissage de métrique. Nous avons montré dans l'article que la suppression de la moindre étape dans la chaîne de traitement proposée menait à une réduction drastique des performances.

Dataset / Method	MSR-Action3D protocol of J. WANG et al. 2012	MSR-Action3D protocol of OREIFEJ et Z. LIU 2013	UTKinect-Action3D	Florence 3D Actions
J. WANG et al. 2012	88.2%	—	—	—
XIA, CHEN et AGGARWAL 2012	—	—	90.92% ± 1.74%	—
LUO, W. WANG et H. QI 2013	96.7%	—	—	—
OREIFEJ et Z. LIU 2013	88.36%	82.15% ± 4.18%	—	—
SEIDENARI et al. 2013	—	—	—	82.0%
TRAN et LY 2013	88.89%	—	—	—
DEVANNE et al. 2014	92.1%	87.28% ± 2.41%	91.5%	87.04%
LU, JIA et TANG 2014	95.62%	—	—	—
VEMULAPALLI, ARRATE et CHELLAPPA 2014	89.48%	—	97.08%	90.88%
YANG et TIAN 2014	93.09%	—	—	—
VEERIAH, ZHUANG et G.-J. QI 2015	92.03%	—	—	—
LI et LEUNG 2016	92.2%	—	—	—
RAHMANI et al. 2016	—	86.5%	—	—
Our method	**97.1%**	**90.36% ± 2.45%**	**98.00% ± 3.49%**	**94.39%**

TABLE 3.2: Précision de notre méthode sur 3 jeux de données. Les colonnes 2 et 3 représentent les résultats sur MSR-Action3D en utilisant les protocoles de J. WANG et al. 2012 et OREIFEJ et Z. LIU 2013, respectivement. Les colonnes 4 et 5 montrent les résultats sur UTKinect-Action3D et Florence 3D Actions.

3.3 Apprentissage distribué de représentations visuelles

Dans cette section, nous détaillons nos travaux sur l'apprentissage de représentations dans un contexte distribué. Plus que le simple gain de temps obtenu par la parallélisation des calculs, nous nous sommes intéressés aux cas où les données sont naturellement distribuées sur un réseau de machines pour lequel il n'existe aucune autorité centrale.

Puisque les représentations visuelles qui nous intéressent s'appuient sur des méthodes d'apprentissage statistique, nous présentons dans un premier temps les méthodologies que nous avons développées pour étendre des algorithmes d'apprentissage classiques à un contexte décentralisé. Puis, dans une seconde partie, nous détaillons nos travaux sur l'apprentissage de réseaux de neurones profonds à l'aide de ces mêmes outils, puisque ceux-ci représentent une part non négligeable des méthodes d'apprentissage de représentations modernes.

Ces travaux ont été majoritairement réalisés dans le cadre de la thèse de Jérôme Fellus, et dans une moindre mesure lors d'une collaboration au LIP6 dans le cadre de la thèse de Michael Blot pour la partie concernant l'apprentissage de réseaux de convolution dans un contexte décentralisé.

3.3.1 Apprentissage statistique distribué à l'aide de protocoles gossip

Nous avons considéré le cas où l'on dispose d'un ensemble de données X réparti sur N machines (ou agents) reliées en réseau donnant ainsi naissance à un ensemble de données locales X_i en chaque machine i du réseau. Le problème d'apprentissage (classification, régression, estimation de densité, *etc*) auquel on s'intéresse consiste alors à trouver les paramètres θ optimaux d'une fonction objectif dépendant des données présentes sur tout le réseau. On suppose que cette fonction peut s'écrire comme la somme de fonctions objectif locales à chaque machine :

$$\min_{\theta} J(X,\theta) = \sum_{i=1}^{N} J_i(X_i,\theta)$$

On se propose alors de résoudre ce problème par une approche par consensus dans laquelle chaque agent résout son problème local sous la contrainte que les solutions locales soient égales :

$$\begin{aligned} &\min_{\theta_1,\ldots,\theta_N} \sum_{i=1}^{N} J_i(X_i,\theta_i) \\ &\quad \text{s.t. } \theta_1 = \ldots = \theta_N \end{aligned}$$

Cette approche par consensus permet de faire apparaître des variables locales qui sont régies par deux forces contradictoire, à savoir l'optimisation du problème local et la contrainte d'égalité entre agents, le compromis entre les deux permettant d'atteindre l'optimum du problème global. Pour résoudre le problème global, nous proposons alors d'alterner entre des mises à jour locales optimisant le problème local et des échanges permettant la résolution du problème de consensus. Dans la plupart des cas, cette stratégie nous permet de garantir l'équivalence avec un algorithme d'optimisation sur la fonction objectif centralisée.

Notons que le consensus peut s'exprimer aussi sous la forme suivante :

$$\forall i, \theta_i = \frac{1}{N} \sum_{m=1}^{N} \theta_m$$

C'est à dire que chaque variable locale doit être égale à la moyenne des variables de tous les agents. Dans ce cas, résoudre le problème de consensus revient à résoudre un problème de moyenne distribuée.

Pour résoudre le problème de moyenne distribuée, nous nous sommes intéressés aux protocoles dit *Gossip* qui permettent une résolution de manière décentralisée à l'aide de communication

pair-à-pair. On représente alors les variables à moyenner dans un vecteur θ et les communications permettant de faire le moyennage peuvent s'écrire sous la forme d'une matrice de communication K telle que :

$$\theta(t+1)^\top = \theta(t)^\top \mathbf{K}$$

Une entrée (i,j) de $\mathbf{K}$ contient alors la proportion de la variable stockée chez l'agent i qui est transférée à l'agent j. Pour résoudre le problème de moyennage, $\mathbf{K}$ doit combiner deux propriétés essentielles que sont la conservation de masse et la stabilité de la moyenne. La conservation de masse impose que la somme des valeurs des agents doit rester constante par communication, ce qui est équivalent à $\mathbf{K1} = \mathbf{1}$ ($\mathbf{1}$ est un vecteur propre droit de $\mathbf{K}$). La stabilité impose que la moyenne soit un point fixe du processus de communication, et donc que $\mathbf{1}^\top \mathbf{K} = \mathbf{1}^\top$ ($\mathbf{1}$ est un vecteur propre gauche de $\mathbf{K}$). Une matrice de communication ayant ces deux propriétés est dite doublement stochastique. Cette double stochasticité implique malheureusement des cycles dans le processus de communication, c'est à dire des nœuds qui sont à la fois émetteurs et récepteurs. Or, le fait d'avoir des nœuds à la fois émetteurs et récepteurs suppose des communications synchrones, c'est à dire des agents qui doivent attendre d'avoir reçu un message avant de poursuivre leurs opérations.

Pour ne pas subir la contrainte des communications synchrones, nous avons utilisé des protocoles dits *sum-weight* décrits dans BÉNÉZIT et al. 2010, pour lesquels une des contraintes de stochasticité est relâchée et remplacée par un poids w_i associé à chaque valeur θ_i. En faisant évoluer les θ_i et les w_i avec les mêmes matrices de communication, on peut montrer que le ratio $\frac{\theta_i}{w_i}$ tend vers la moyenne $\frac{1}{N}\sum_m \frac{\theta_m}{w_m}$ pour tout agent i.

De fait, l'utilisation de matrices simplement stochastiques avec un jeu de poids associé à chaque variable nous permet de proposer des versions décentralisées de plusieurs algorithmes d'apprentissage statistique utilisés couramment dans la littérature de l'apprentissage de représentations. En particulier, nous détaillons par la suite nos propositions pour l'estimation de densité et le clustering, la réduction de dimension, et l'optimisation convexe pour la classification linéaire.

Clustering

Une première contribution a été de proposer une version décentralisée de l'algorithme k-means qui reste une référence pour la construction de dictionnaires visuels. À partir d'un ensemble d'observations $X = \{x_i\}$, l'objectif est de trouver une partition de $X = \cup_k C_k$ en parties C_k associées à des centroïdes μ_k de manière à minimiser l'erreur de quantification des x_i sur les μ_k de la partie à laquelle ils correspondent :

$$\min_{C_k,\mu_k} \sum_k \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

Cet objectif est minimisé à l'aide d'un algorithme de type Excpectation-Maximization (EM) qui correspond dans ce cas particulier à une descente de gradient alternée à pas optimal :

$$\text{E } (\mu_k \text{ fixés, mise-à-jour de } C_k) : \forall k, C_k = \{x_i | \mu_k = \operatorname*{argmin}_c \|x_i - \mu_c\|^2\}$$

$$\text{M } (C_k \text{ fixées, mise-à-jour de } \mu_k) : \forall k, \mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

Dans le cas distribué, les données sont réparties sur plusieurs nœuds $X = \cup_n X^n$. On peut alors définir des partitions locales C_k^n associées à des centroïdes locaux μ_k^n et définir le problème distribué équivalent incluant une contrainte de consensus :

$$\min_{C_k^n,\mu_k^n} \sum_n \sum_k \sum_{x_i^n \in C_k^n} \|x_i^n - \mu_k^n\|^2$$
$$\text{s.t.} \forall (m,n,k), \mu_k^m = \mu_k^n$$

La contrainte de consensus rend le problème très facile à résoudre en utilisant la même stratégie EM :

$$\text{E} : \forall (n,k), C_k^n = \{x_i^n | \mu_k^n = \underset{c}{\text{argmin}} \, \|x_i^n - \mu_c^n\|^2\}$$

$$\text{M} : \forall (n,k), \mu_k^n = \frac{1}{\sum_m |C_k^m|} \sum_m \sum_{x_i^m \in C_k^m} x_i^m$$

Comme on peut le voir, l'étape E peut se faire localement en chaque nœud sans faire intervenir de communication, et l'étape M est un calcul de moyenne pondérée distribuée pour lequel les protocoles gossip sum-weight sont bien adaptés. Nous avons donc proposé l'algorithme AGKM (Asynchronous Gossip K-Means) décrit dans l'algorithme 1.

Algorithm 1 AGKM : Asynchronous Gossip K-means (en chaque noeud n)

Entrées : Jeu d'échantillons $\mathbf{X} \in \mathbb{R}^{N_n \times D}$
Paramètres : K : nombre de cellules désirées
M : nombre de messages envoyés à chaque itération locale

∘ **Procédure émission**

1: $C \leftarrow$ partition aléatoire de $\mathbf{X}$ en K cellules
2: $\forall k \in \{1,\ldots,K\}, \quad w_k \leftarrow w_k^{\text{old}} \leftarrow |C_k|; \quad \mu_k \leftarrow \mu_k^{\text{old}} \leftarrow \sum\{x \in C_k\}$
3: **Boucle**
4: **Pour** m **de** 1 **à** M **faire**
5: $\forall k, \; w_k \leftarrow w_k/2; \quad \mu_k \leftarrow \mu_k/2$
6: $j \leftarrow$ nœud voisin aléatoire
7: Envoyer $(\mu_k, w_k)_{k=1}^K$ à j
8: **Fin Pour**
9: $\forall k, \; C_k \leftarrow \{x \in \mathbf{X}^{(i)} : k = \text{argmin}_l \, \|x - s_l/w_l\|_2^2\}$
10: $w_k \leftarrow w_k - w_k^{\text{old}} + |C_k|; \quad \mu_k \leftarrow \mu_k - \mu_k^{\text{old}} + \sum\{x \in C_k\}$
11: $w_k^{\text{old}} \leftarrow |C_k|; \quad \mu_k^{\text{old}} \leftarrow \sum\{x \in C_k\}$
12: **Fin Boucle**

∘ **Procédure réception**

1: **Boucle**
2: Attendre réception d'un message $(\mu'_k, w'_k)_{k=1}^K$
3: $\forall k, \; w_k = w_k + w'_k; \quad \mu_k = \mu_k + \mu'_k$
4: **Fin Boucle**

Cet algorithme comporte deux procédures concurrentes exclusives s'exécutant en chaque nœud du réseau. La procédure d'émission alterne entre une étape E de calcul des partitions locales (ligne 9) et une partie de l'étape M de mise à jour des centroïdes (lignes 4 à 8 et 10 à 11). Cette étape M consiste à propager l'information locale aux autres nœuds en envoyant des moyennes partielles issues des partitions locales. La procédure de réception consiste à agréger l'information reçue des autres nœuds afin d'améliorer la précision sur le consensus. Notons enfin qu'AGKM fait intervenir un mécanisme d'auto-correction (lignes 10-11, $\mu_k \leftarrow \mu_k - \mu_k^{\text{old}} + \sum\{x \in C_k\}$) avec lequel les centroïdes et leur population sont mis-à-jour de la variation par rapport à l'étape de partitionnement précédente. Ce mécanisme permet de conserver l'information obtenue par réception des messages survenus entre deux étapes de partitionnement.

Nous avons montré l'existence d'une borne sur le nombre de messages M que chaque nœud doit envoyer, au delà duquel le consensus est atteint avec une précision suffisante pour que l'étape E soit identique en chaque nœud. Le comportement d'AGKM est alors complètement identique à celui

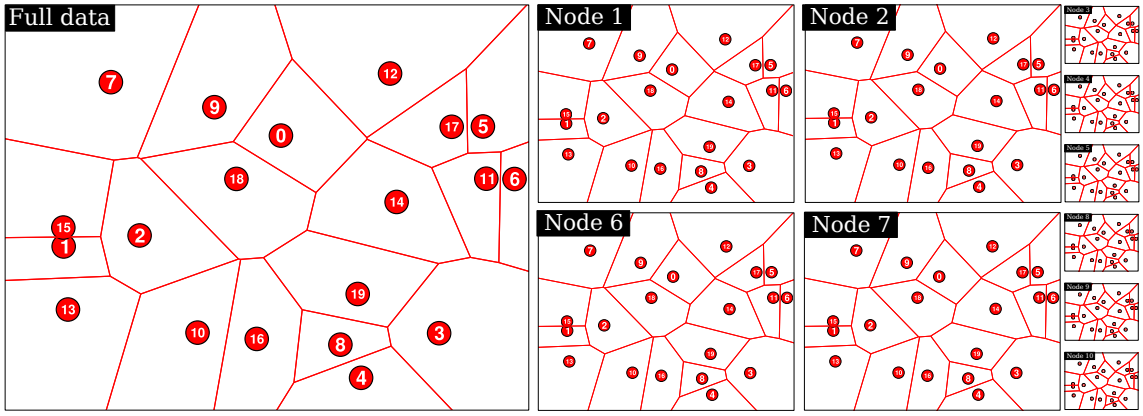


FIGURE 3.6: Résultat produit par AGKM sur un jeu de données synthétiques. Chaque couleur représente une catégorie obtenue (ici $K = 20$). Tous les noeuds, bien qu'ils n'hébergent chacun qu'un petit sous-ensemble d'entraînement, converge vers un partitionnement identique (frontières en rouge). Ce partitionnement modélise correctement les catégories présentes dans les données du réseau (en haut à gauche).

d'un k-means classique centralisé, avec le bénéfice de la répartition des calculs. La valeur théorique de cette borne est difficile a obtenir, et nos efforts en ce sens ont donné des valeurs au niveau du broadcast (c'est à dire que chaque nœud doit communiquer avec tous les autres). Cependant, en pratique on observe qu'un nombre de messages par machine logarithmique en nombre de voisins est largement suffisant pour obtenir une convergence correcte.

Expérimentalement, nous montrons sur la figure 3.6 un jeu de données synthétiques pour lequel nous arrivons à obtenir des partitions correctes et identiques en chaque nœud même dans le cas où les données ne sont pas uniformément distribuées sur les nœuds.

Enfin, nous montrons en figure 3.7 les évolutions correspondantes de l'erreur de quantifaction (J1) et de l'erreur de consensus (J2). L'erreur de quantification décroît de manière rapide en comparaison du k-means centralisé du fait du bénéfice de la parallélisation des calculs. On remarque également que l'erreur de consensus décroît exponentiellement avec le nombre de message échangés. Celle-ci peut remonter de temps à autres (en raison des étapes de partitionnement locales), mais finit par être annulée. Ces travaux ont été publiés dans Jérôme FELLUS, PICARD et P.-H. GOSSELIN 2015 ; Jerome FELLUS, PICARD et P.-H. GOSSELIN 2013.

Réduction de dimension

Nous nous sommes ensuite intéressés à la réduction de dimension par projection linéaire et en particulier les méthodes équivalentes à l'analyse en composantes principales. Ces méthodes sont très populaires en apprentissage de représentations visuelles, que ce soit en pré-traitement pour conditionner des descripteurs visuels à un processus d'agrégation, ou bien en post-traitement afin de compresser des représentations visuelles trop volumineuses.

De manière analogue à la PCA, nous nous sommes intéressés aux méthodes d'approximation de rang faible de la matrice de Gram d'un ensemble de vecteurs. En effet, les données que nous voulons traiter sont souvent évaluées par le produit scalaire. Dès lors, la conservation au mieux du produit scalaire devient une propriété importante de toute méthode de réduction de dimension. Cela revient alors à considérer le problème de valeurs propres suivant :

$$\max_{U} \|U^\top X X^\top U\|_F^2$$
$$\text{s.t. } U^\top U = I$$

La solution de ce problème est obtenue pour les vecteurs propres de la matrice de covariance $XX^\top$

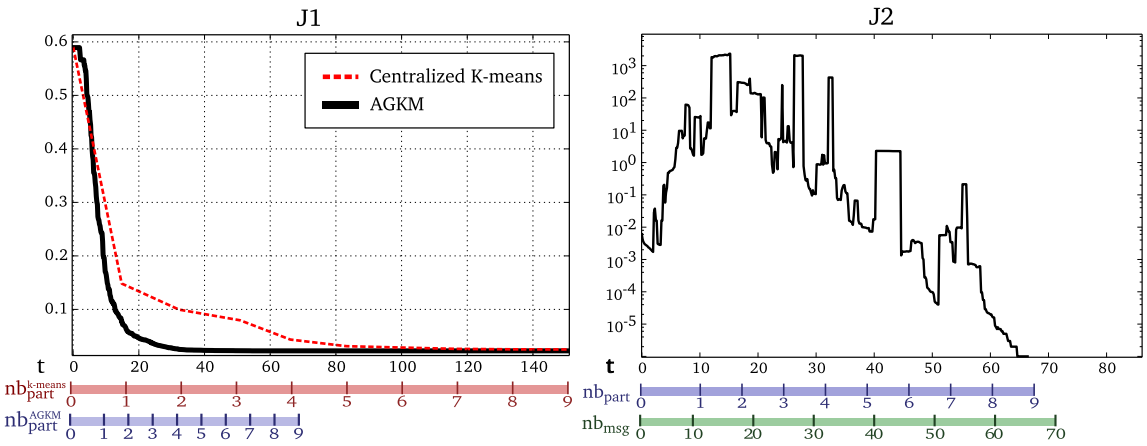


FIGURE 3.7: J_1 converge vers le même minimum local de la MSE qu'un K-means centralisé, mais plus rapidement. J_2 tend vers zéro exponentiellement. Ici, nb_{part}^{AGKM},$nb_{part}^{\text{K-means}}$ désignent respectivement le nombre moyen de partitions calculées en chaque noeud et le nombre d'itérations de K-means. nb_{msg} désigne le nombre moyen de messages émis par chaque nœud.

Dans un contexte décentralisé, nous ne connaissons pas la totalité des échantillons X mais uniquement les sous-ensembles X_i stockés en chaque nœud. Une manière naïve de résoudre le problème de valeurs propres consiste alors à faire l'estimation distribuée de la matrice de covariance de X, ce qui correspond à un problème de moyenne pondérée distribuée facilement résolu par protocole gossip sum-weight. Le problème est qu'il faut transférer sur le réseau des matrices de covariances partielles $X_i X_i^\top$ qui peuvent être extrêmement volumineuses.

Nous avons alors proposé une méthode pour résoudre le problème de valeurs propres dans le domaine compressé, c'est à dire en ne transférant que des matrices de taille $d \times q$ ou d est la dimension d'entrée, et q le nombre de dimensions à conserver. Cet algorithme basé sur un protocole sum-weight est présenté en algorithme 2.

À la différence des autres protocoles sum-weight que nous avons proposés, la réception d'un message nécessite la combinaison de 2 ensembles de vecteurs orthonormaux afin de produire un unique ensemble de vecteurs orthonormaux engendrant l'espace propre de la solution. Cette combinaison est réalisée par la décomposition de la reconstruction de la matrice de covariance obtenue par fusion des deux ensembles considérés (lignes 4-8). Les parenthèses sont mises afin de signaler l'ordre des produits à effectuer pour bénéficier d'une faible complexité de calcul.

Nous avons montré que sous certaines conditions sur les données et la dimension cible, à savoir $rang(X) \leq q$, cette méthode converge vers les mêmes projecteurs qu'une PCA centralisée. Nous montrons en figure 3.8 un exemple de convergence dans lequel la dimension cible est exactement égale au rang de la matrice de covariance tout en étant bien plus faible que la dimension des données. Nous comparons la version naïve (Late-PCA) et la version dans le domaine compressé à la solution centralisée. L'algorithme converge d'autant plus rapidement que les données sont réparties sur un faible nombre de nœuds, ce qui est attendu puisqu'il y a d'autant moins de combinaisons de bases à effectuer. Cependant, il faut noter que la distribution sur un grand nombre de machines entraîne des calculs bien moins complexes. En effet, l'initialisation en chaque nœud nécessite la factorisation de la matrice de Gram locale qui est d'autant plus petite qu'il y a un faible nombre de données locales.

Nous avons étendu cette proposition au cas où ce sont les composantes qui sont réparties. Ces travaux ont été publiés dans Jerome FELLUS, PICARD et P.-H. GOSSELIN 2015 ; Jerome FELLUS, PICARD et P.-H. GOSSELIN 2014. L'article Jerome FELLUS, PICARD et P.-H. GOSSELIN 2015 est disponible en annexe.

Algorithm 2 AGPCA-DS : Asynchronous Gossip PCA (en chaque nœud n)

Entrées : Jeu d'échantillons $\mathbf{X} \in \mathbb{R}^{N_n \times D}$
Paramètres : q : nombre de composantes principales désirées

○ **Procédure émission**

$\mathbf{a} \leftarrow \mathbf{X1}$; $\mathbf{G} \leftarrow \mathbf{X}^\top\mathbf{X}$; $w \leftarrow N_n$
$(\mathbf{V},\mathbf{L}) \leftarrow \text{eig}_q(\mathbf{G})$
$\mathbf{U} \leftarrow \mathbf{XVL}^{-\frac{1}{2}}$
Boucle
  $\mathbf{a} \leftarrow \mathbf{a}/2$; $\mathbf{L} \leftarrow \mathbf{L}/2$; $w \leftarrow w/2$
  $j \leftarrow$ nœud voisin aléatoire
  Envoyer $(\mathbf{a},\mathbf{U},\mathbf{L},w)$ à j
Fin Boucle

○ **Procédure réception**

Boucle
  Attendre réception d'un message $(\mathbf{a}',\mathbf{U}',\mathbf{L}',w')$
  $\mathbf{a} \leftarrow \mathbf{a}+\mathbf{a}'$; $w \leftarrow w+w'$
  $\mathbf{Q}_0 \leftarrow \mathbf{U}$
  Pour t **de** 1 **à** *max_t* **faire**
    $(\mathbf{Q}_{t+1},\mathbf{R}_{t+1}) \leftarrow \text{QR}(\mathbf{UL}(\mathbf{U}^\top\mathbf{Q}_t)+\mathbf{U}'\mathbf{L}'(\mathbf{U}'^\top\mathbf{Q}_t))$
  Fin Pour
  $\mathbf{U} \leftarrow \mathbf{Q}_{max\_t}$; $\mathbf{L} \leftarrow \text{diag}(\mathbf{R}_{max\_t})$
Fin Boucle

Classification linéaire

Nous nous sommes enfin intéressés à distribuer l'étape de classification des images en apprenant de manière décentralisée un SVM linéaire. En partant de signature type BOW avancé (VLAD, VLAT, Fisher Vectors, *etc*), un classifieur linéaire est normalement largement suffisant pour obtenir de très bons résultats en classification.

Pour cette étape de classification distribuée, nous considérons que des signatures X^n sont réparties sur un réseau de machines et que nous essayons d'entraîner un hyperplan minimisant un risque empirique global mesurant l'erreur de prédiction par rapport aux labels y_i associés à chaque signature x_i, augmenté d'une régularisation convexe :

$$\min_w \frac{1}{2}\|w\|^2 + \frac{1}{\lambda \sum_n |X^n|} \sum_n \sum_{x_i in X^n} max(0, 1 - y_i w^\top x_i)$$

Pour distribuer l'optimisation de ce problème, nous avons proposé une stratégie inspirée de SAG introduit dans SCHMIDT, LE ROUX et BACH 2017 et qui utilise un gradient moyenné sur un ensemble d'exemples contenus dans une file de type FIFO. Un buffer suit l'évolution du gradient moyenné. À chaque itération, un nouvel exemple est tiré au hasard, ajouté à la file et le buffer de gradient est incrémenté du gradient correspondant. Si la file a dépassé une taille critique, le premier élément en est retiré et le buffer de gradient est décrémenté en conséquence. Nous avons appelé cette stratégie Short Term Averaged Gradient (STAG), ce qui correspond a la mise-à-jour suivante :

$$w_{t+1} = (1 - \eta\lambda)\, w_t - \frac{\eta}{L} g(t), \qquad g(t) = \sum_{l=0}^{L} \nabla h_i(w_t), \qquad h_i(w_t) = \max(0, 1 - y_i w_t^\top x_i)$$

Les (x_i, y_i) étant tirés de manière uniforme sans remise parmi les exemples d'apprentissage.

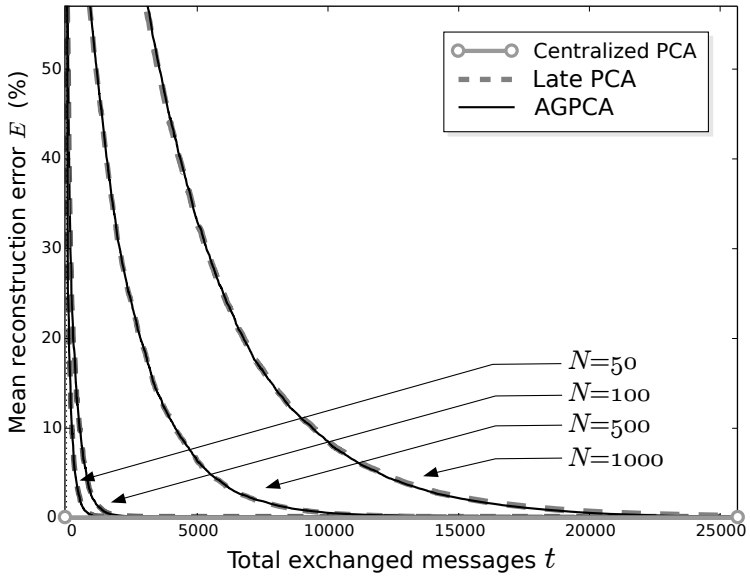


FIGURE 3.8: Convergence d'AGPCA comparée à une stratégie Late PCA et à la solution analytique de la PCA sur un jeu de données synthétiques généré suivant un mélange de lois normales ($D = 200$, $p = 30$, $N = 10000$), réparties sur $N = 100$ nœuds, avec $q = 30 = p \ll D$.

L'intérêt de STAG est qu'il permet d'être facilement étendu au cas distribué en imposant une contrainte de consensus sur les buffers de gradient. Cet algorithme que nous avons appelé AGSTAG exécute de manière concurrente une boucle d'émission et une boucle de réception comme toutes nos autres propositions. La boucle d'émission met à jour le buffer de gradient local et l'envoie à M voisins. La boucle de réception accumule les messages reçus dans le buffer de gradient local.

Nous avons testé cet algorithme pour faire de la classification d'images sur la base PASCAL VOC 2007. Les résultats présentés en figure 3.9 montrent que l'on peut obtenir les mêmes performances qu'une classification centralisée tout en bénéficiant de l'accélération liée à la distribution des calculs. La figure 3.10 montre que ce résultat reste valable même avec peu d'échanges entre les nœuds.

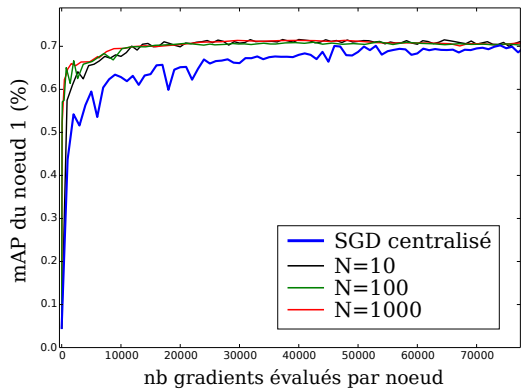


FIGURE 3.9: Performances du système en classification pour différentes tailles de réseau N (ici, $L = 1$ et $M = 10$).

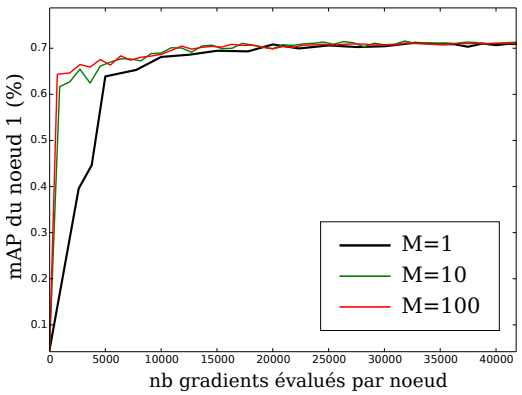


FIGURE 3.10: Influence du paramètre M sur la convergence du système pour N=10.

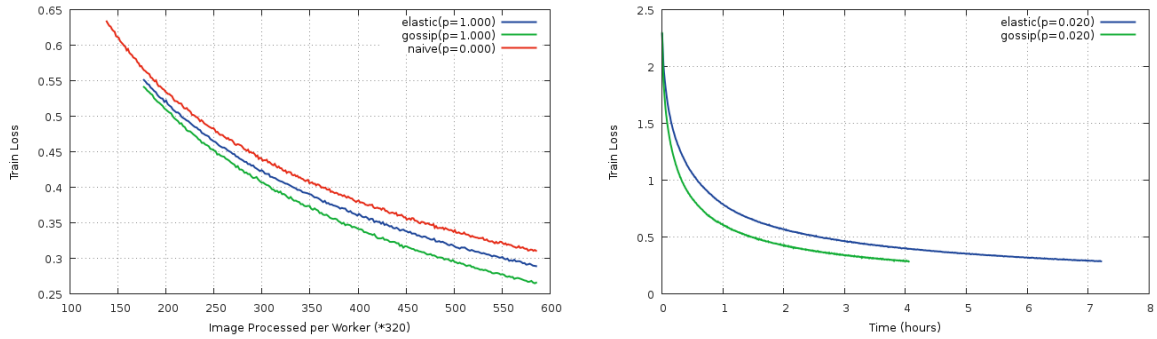


FIGURE 3.11: Évolution de la fonction objectif en fonction du temps pour différente stratégie d'optimisation distribuée et avec différente fréquence de communication.

3.3.2 Apprentissage distribué de réseaux de neurones profonds

Nous avons enfin voulu étendre le principe d'apprentissage statistique distribué au réseaux de neurones profonds et en particulier aux réseaux de convolution (CNN). Le principal challenge de l'extension des méthodes gossip aux réseaux de convolution est le nombre de paramètres que ceux-ci comportent. En effet, les CNN dont les performances sont au niveau de l'état de l'art comportent généralement plusieurs dizaines voire centaines de millions de paramètres. Dès lors, imposer une contrainte de consensus entre des modèles qui seraient répartis nécessite de faire transiter entre ces machines des centaines de Mo à chaque échange de messages.

Nous avons proposé un algorithme d'entraînement dans lequel les nœuds possèdent chacun leur version du modèle à entraîner et l'optimisent sur des batches d'exemples tirés localement. Nous avons ajouté une contrainte de consensus sur les modèles en faisant envoyer le modèle local sur le réseau par chaque nœud à l'aide d'un protocole sum-weight. Nos contributions sur cet aspect sont doubles. D'abord, nous avons montré qu'imposer un consensus sur les modèles à l'aide d'un protocole sum-weight était équivalent à effectuer une descente de gradient stochastique sur des batches de plus grande taille. Sachant qu'une taille de batch minimale est souvent nécessaire pour s'assurer de la convergence du CNN vers une solution satisfaisante, ceci permet d'entraîner de plus gros CNN exploitant plus de ressources en local. Enfin, nous avons montré expérimentalement que même avec des taux de communication très faibles (1 message toute les 100 mises-à-jour du modèle), les modèles convergent vers un modèle consensus très performant.

Ces résultats sont illustrés sur la figure 3.11 qui montre l'évolution de la fonction objective de notre approche gossip pour un CNN entraîné sur la base CIFAR10, comparé à une approche naïve (centralisée) et à l'approche elastic-net décrite dans ZHANG, CHOROMANSKA et LECUN 2015 pour différentes fréquences de communication.

3.4 Applications aux collections culturelles et patrimoniales

Dans cette section, nous présentons les applications des travaux de recherche présentés précédemment aux collections culturelles et patrimoniales. La création du Labex Patrima sur le site de Cergy à en effet fait émerger une forte quantité de problématiques liées aux acteurs du patrimoine culturel (Bibliothèque nationale de France, Louvre, etc) pour lesquelles l'apprentissage de représentations visuelles à permis d'apporter des solutions. Les collaborations avec les acteurs du patrimoine culturel créées grâce à ce labex ont permis de faire reconnaître une certaine expertise dans le domaine de la vision pour le patrimoine sur le site de Cergy qui nous a amené à développer des collaborations à l'international sur des sujets connexes. Nous présentons dans cette partie les travaux issus de collaborations ayant donné lieu à des publications scientifiques.

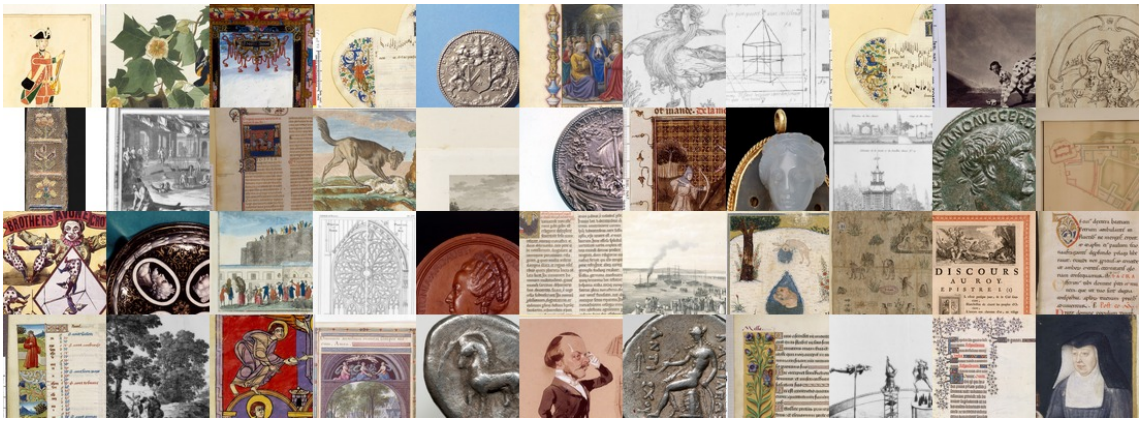


FIGURE 3.12: Montage d'extraits des dimages de notre jeu de données issu de la BnF.

3.4.1 Indexation d'images

L'indexation des images numérisées est un des problèmes importants de la Bibliothèque nationale de France. En effet, la bibliothèque possède une immense collection d'images (plus de 275k images) issues de campagnes de numérisation continues. Ces images sont accessibles en ligne (`http://images.bnf.fr`) à l'aide d'une interface de recherche par mots-clefs. Le problème de cette méthode d'indexation est que chaque image a dû être examinée par un documentaliste afin de produire l'ensemble de mots-clefs qui lui sont associés. Deux problèmes principaux se posent : d'abord bien sûr le temps humain nécessaire à l'exécution de cette tâche colossale, ensuite la cohérence des annotations produites. Même s'il existe un thésaurus commun à tous les documentalistes, celui-ci peut varier dans le temps (certains mots-clefs peuvent se séparer ou se regrouper au fil du temps) et son interprétation est de toute façon subjective.

Nous avons donc proposé d'étudier la faisabilité d'un système d'annotation semi-automatique dans lequel le documentaliste se verrait proposer une série de mots-clefs parmi lesquels choisir. Pour se faire, nous avons étudié différentes méthodes d'annotation automatique dans le cadre du projet ASAP (Annotation Semi-Automatique des images Patrimoniales, financement Patrima) en partenariat avec la BnF.

Dans se projet, nous avons construit un jeu de données composé d'environ 3000 images pour lesquelles nous avons une série d'annotations constituées de plusieurs mots-clefs pertinents par image parmi 569 classes que nous avons séparées en 5 catégories (Visuelle, Semantique, Historique, Geographique et Physique). Un montage d'extraits d'images de ce jeu de données est présenté en figure 3.12.

Nous avons mesuré deux types de tâches d'annotation afin de comparer les performances de différentes représentations visuelles couramment utilisées dans la littérature. Nous avons séparé les représentations visuelles en 3 sous-groupes : les représentations issues de statistiques globales (histogramme de couleur, GIST), les méthodes à sac de mots (Fisher Vectors, VLAT) et les réseaux de neurones convolutionnels. Tout le nécessaire pour reproduire ces évaluations est disponible en ligne (`http://perso-etis.ensea.fr/~picard/bnf_bench/`).

La première tâche est une tâche de classification durant laquelle un classifieur de type SVM est entraîné pour chacune des classes sur un sous-ensemble du jeu de données. Les performances sont évaluées avec la précision moyenne (mean average precision - mAp) par validation croisée à 5 plis (5-fold cross-validation). Les résultats de cette tâche sont détaillés en tableau 3.3. Comme on peut le voir, ce sont les représentations basées sur les CNN qui offrent les meilleures performances. Cependant, ces résultats ne sont pas très éloignés des méthodes de BoW avancées comme les Fisher Vectors, et il y a fort à parier que des méthodes hybrides de type NetVLAD détaillée

Des.	Phy.	Vis.	Sém.	His.	Géo.	Moy.
lab	4.8%	4.0%	2.1%	3.5%	8.5%	4.6%
gist	19.3%	21.8%	9.0%	10.9%	16.6%	15.5%
llc	8.3%	19.0%	5.2%	3.0%	13.7%	9.8%
fv256	25.5%	32.0%	13.5%	12.8%	27.5%	22.3%
dl8	34.6%	31.1%	24.7%	13.1%	30.1%	24.7%
dl19	30.9%	29.1%	15.5%	17.2%	31.0%	24.8%

TABLE 3.3: Précision moyenne pour chaque groupe de catégories en fonction des descripteurs visuels.

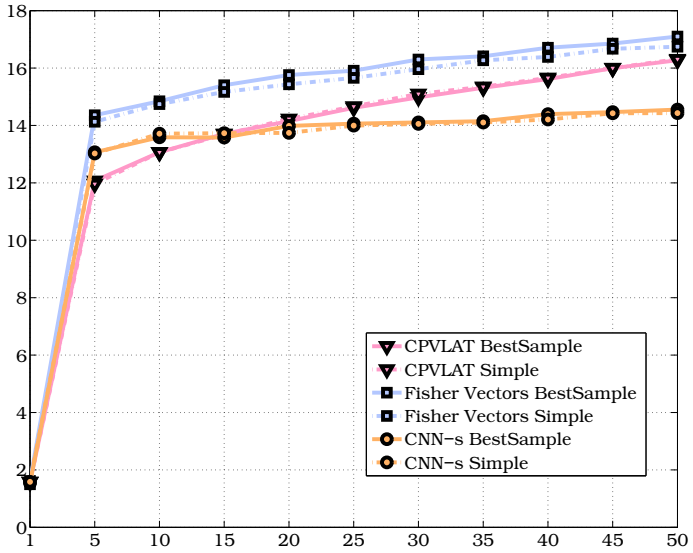


FIGURE 3.13: Évolution du mAp en fonction du nombre d'images annotées pour différentes combinaisons de représentations visuelles et de stratégies actives.

dans ARANDJELOVIC et al. 2016 amélioreraient encore les résultats.

La seconde tâche est une tâche d'annotation interactive pour laquelle nous entraînons un classifieur à l'aide d'une stratégie d'apprentissage actif. À chaque itération, une image est sélectionnée dans le jeu de données et ajoutée à l'ensemble d'entraînement. Deux stratégies classiques ont été étudiées : la sélection de l'image ayant le meilleur score de classification (BestSample) et la sélection de l'image la plus incertaine (Simple, tirée de TONG et KOLLER 2001). Il s'agit alors de comparer les performances des différents descripteurs visuels à classifieur et méthode active égaux. Nous montrons l'évolution du mAp en fonction du nombre d'images annotées par la stratégie active sur la figure 3.13. Il faut noter que dans cette tâche, les méthodes de BoW avancées font mieux que les CNN, ce qui est peut être lié à leurs meilleures capacités de précision là où les CNN excellent plutôt en généralisation. Il faut noter que puisque les images annotées sont comptabilisées dans l'évaluation du mAp et sachant que peu de catégories possèdent plus de 50 images, un score de 17% est très faible et peut être fortement amélioré. Ces travaux ont été publiés dans PICARD, P.-H. GOSSELIN et GASPARD 2015.

3.4.2 Reconnaissances de papier photographique

Une seconde application de nos travaux aux collections patrimoniales a été la participation à une collaboration internationale sur la reconnaissance de papier photographique organisée par Paul Messier (Yale University) et C. Richard Johnson (Cornell University). La problématique

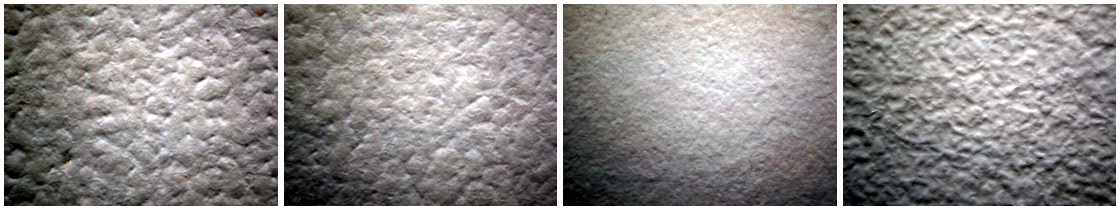


FIGURE 3.14: Exemples d'images de papier photographique tirées du jeu de données Black&White.

	HOG	LBP	POEM	VLAT+HOG	VLAT+LBP	VLAT+POEM
inkjet	63.8	45.7	66.3	69.7	67.8	**73.6**
bw	26.9	24.1	26.3	31.0	26.4	**31.3**

TABLE 3.4: mAp pour différente combinaisons de représentations visuelles. la première ligne correspond au jeu de données *inkjet* tandis-que la seconde correspond au jeu *bw*.

est de reconnaître sur quel type de papier des photographies anciennes ont été imprimées, ceci à des fins de conservation. En effet, dans le cas de photographies historiques, c'est souvent le photographe lui-même qui procédait à l'impression (tirage) et cette étape est rarement documentée. Or, la composition du papier a une influence énorme sur les conditions de stockage ou d'exposition des photographies afin de les préserver des dégradations du temps.

Pour réaliser l'identification du papier utilisé, une méthode non destructive a bien sûr été mise en œuvre. Le dispositif d'acquisition est constitué d'un appareil photo fixé par rapport à la zone de papier numérisée de manière à avoir la même résolution (pixel par mm) pour toutes les images. Le papier est éclairé en lumière rasante de manière à avoir les reliefs de la texture de papier apparaissant sous forme d'ombres. Quelques images de papiers photos sont présentées en figure 3.14.

Nous avons proposé l'utilisation de plusieurs méthodes de sac de mots avancées pour réaliser l'identification du type de papier. Dans PICARD, VU et FIJALKOW 2014, nous proposons d'utiliser les VLAT présentés en section 3.1 combinés à des descripteurs locaux POEM de VU et CAPLIER 2012 qui sont particulièrement efficaces pour la reconnaissance de textures. Nous avons obtenu les résultats présentés dans le tableau 3.4 qui montrent que la combinaison de descripteurs locaux adaptés à la reconnaissance de texture avec des méthodes d'agrégation basées sur les noyaux d'appariement permet d'obtenir de meilleures performances que l'utilisation de ces descripteurs seuls ou bien l'agrégation de descripteurs non-adaptés.

Pour continuer dans cette direction, nous avons proposé dans PICARD et FIJALKOW 2014 de combiner des méthodes d'agrégation de second ordre de type VLAT avec un banc de filtres de Gabor. Nous avons décidé de remplacer le dictionnaire visuel par un mélange de gaussiennes et de procéder à une affectation souple pour chaque gaussienne c, ce qui se rapproche la méthode d'agrégation des Fisher Vectors :

$$\Gamma_{I,c} = \frac{\sum_{p \in I} h_c(d(p))(d(p) - \mu_{I,c})(d(p) - \mu_{I,c})^\top}{\sum_{p \in I} h_c(d(p))}$$

avec

$$\mu_{I,c} = \frac{\sum_{p \in I} h_c(d(p)) d(p)}{\sum_x h_c(d(p))}$$

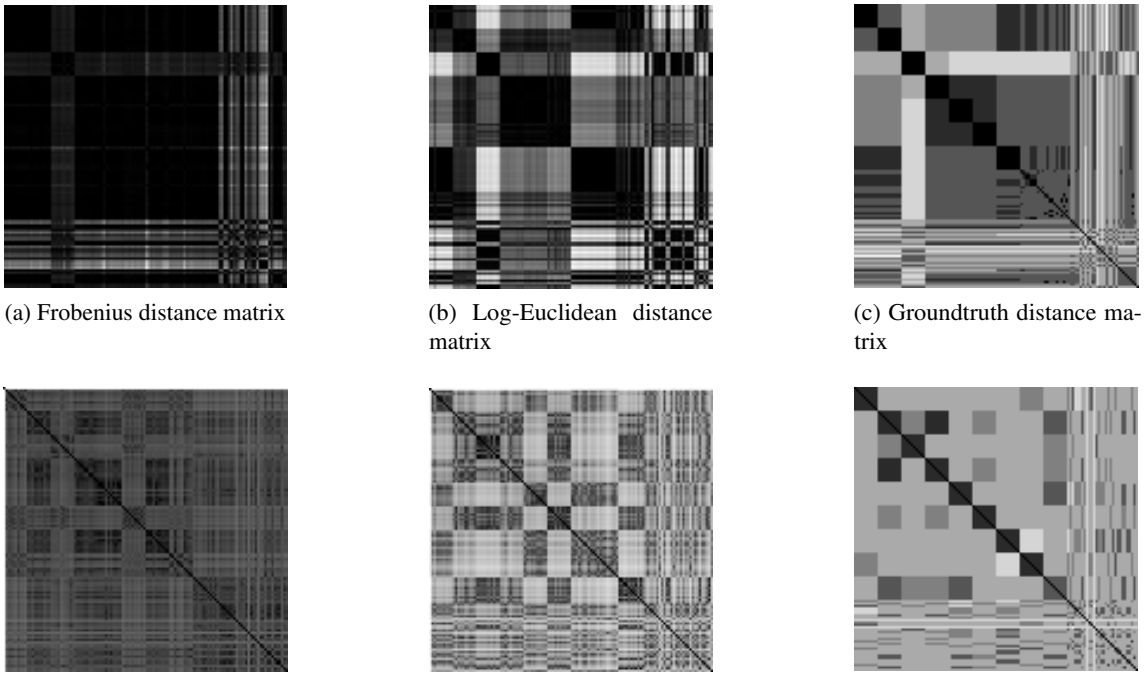
$d(p)$ étant le vecteur des sorties du banc de filtres de Gabor au pixel p.

Les $\Gamma_{I,c}$ étant des matrices symétriques définies positives ($\in \mathbb{S}_{++}$), nous avons proposé d'utiliser des distances adaptées à cette espace pour comparer les représentations visuelles obtenues, et en

particulier la distance log-euclidienne :

$$d(I_1, I_2)^2 = \sum_c \| \log(\Gamma_{I_1,c}) - \log(\Gamma_{I_2,c}) \|_F^2$$

Des résultats qualitatifs sont présentés en figure 3.15 sous la forme de matrices de distance sur chacun des jeux de données. Les images sont rangées par catégorie et on s'attend à trouver une structure bloc diagonale comme le montrent les matrices de vérité-terrain. Comme on peut le voir, l'utilisation d'une distance adaptée aux représentations apprises permet d'améliorer qualitativement les résultats.



(a) Frobenius distance matrix

(b) Log-Euclidean distance matrix

(c) Groundtruth distance matrix

(d) Frobenius distance matrix

(e) Log-Euclidean distance matrix

(f) Groundtruth distance matrix

FIGURE 3.15: Matrice de distance en fonction de la distance utilisée pour les jeux de données inkjet (première ligne) et bw (seconde ligne).

3.4.3 Reconnaissance de filigranes

Une troisième application de nos travaux aux collections patrimoniales concerne la reconnaissance de filigranes de papiers anciens. Il s'agit d'une collaboration initiée avec Georg Dietz (PapierStruktur, Dresden) lors du projet de master de Thomas Henn et dont les résultats ont été publiés dans PICARD, HENN et DIETZ 2016. Les filigranes de papiers anciens sont une source d'information précieuse pour les historiens de l'art, puisqu'ils permettent de dater et localiser les œuvres utilisant ce type de support. La création du filigrane est lié à la technique de moulage du papier utilisée du XVième siècle au XIXième siècle. Le moule est composé de filaments de métal en tamis, chaque fabriquant créant un motif avec ces filaments qui se retrouve en relief sur les feuilles de papier.

Nous nous sommes intéressés à la reconnaissance de ce qu'on appelle les *tracings* de filigranes qui sont une numérisation de l'image du filigrane par un opérateur à l'aide d'un feutre numérique. On obtient ainsi une image binaire comme présentée en figure 3.16.

FIGURE 3.16: Exemples de tracings de filigranes de papiers anciens montrant la diversité des marques de différents fabriquants.

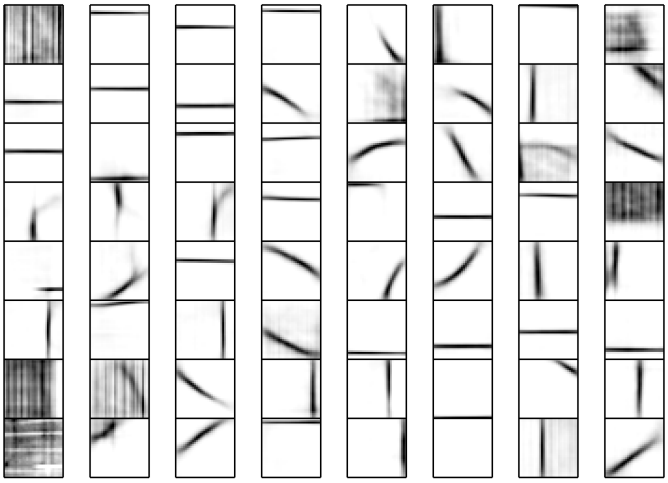


FIGURE 3.17: Exemple de citionnaire avec 64 atomes de taille 32×32 pixels.

Les représentations visuelles que nous avons proposées reposent sur l'apprentissage de dictionnaire et en particulier la factorisation en matrices non-négatives. Nous avons découpé l'image en vignettes que nous reconstruisons à l'aide d'un dictionnaire de motifs élémentaires appris sur l'ensemble des vignettes. À la fois les coefficients des atomes du dictionnaire et les coefficients de la combinaison obéissent à des contraintes de non-négativité, ce qui donne le problème d'optimisation suivant :

$$\begin{aligned} \min_{\mathbf{D},\mathbf{x_r}} \quad & \frac{1}{2} \sum_{\mathbf{r}\in\{\mathbf{r}\}} \|\mathbf{r} - \mathbf{D}\mathbf{x_r}\|^2 \\ \text{s.t.} \quad & \forall i, \mathbf{D}_i \geq 0, \|\mathbf{D}_i\|_2^2 = 1 \\ & \forall \mathbf{r}, \mathbf{x_r} \geq 0, \|\mathbf{x_r}\|_1 \leq k \end{aligned}$$

Avec $\mathbf{D}$ le dictionnaire, $\mathbf{r}$ la vignette à reconstruire et $\mathbf{x_r}$ le vecteur de combinaison des atomes du dictionnaire correspondant. La résolution de ce problème est effectuée à l'aide de la méthode en ligne proposée dans MAIRAL et al. 2010. Pour l'obtention des vecteurs de reconstruction $\mathbf{x_r}$, nous utilisons l'algorithme de Frank-Wolfe décrit dans FRANK et WOLFE 1956 qui se prête très bien aux contraintes de non-négativité et de norme ℓ_1.

Nous obtenons des dictionnaires similaires à celui présenté en figure 3.17. Comme on peut le voir, les atomes obtenus correspondent majoritairement à des morceaux de contours ainsi qu'à des peignes de raies de certaines fréquences qui correspondent à certaines marques que l'on trouve sur les tracings permettant d'identifier l'espacement des filaments composant le moule.

dictionary	support	step	clusters	order	average 1st
64	32	8	64	1	39.8
64	32	8	64	2	**30.6**

TABLE 3.5: Rang moyen de la première image pertinente en fonction des paramètres de NMF et d'agrégation.

Pour effectuer la reconnaissance, nous agrégeons les descripteurs locaux $\mathbf{x_r}$ obtenus dans une représentation de type VLAT qui est analogue à l'appariement de ces descripteurs. Nous évaluons la performance des descripteurs obtenus sur un tout petit jeu de données pour lequel nous avons des correspondances entre images. Pour chaque image requête, nous trions les images restantes par ordre décroissant de similarité et nous mesurons le rang moyen de la première image pertinente. Nous obtenons des rangs présentés dans le tableau 3.5. Comme nous pouvons le voir, l'utilisation d'une agrégation d'ordre 2 (VLAT) permet d'améliorer significativement les résultats par rapport à l'ordre 1 (VLAD). Malgré tout, les résultats sont assez décevant, car un rang moyen de 30 est très éloigné des scores obtenus pour les images naturelles. Ceci montre la difficulté à obtenir de bons descripteurs sur cette tâche.

4. Projet de recherche

Dans ce chapitre, nous détaillons notre projet de recherche qui est réparti en trois axes principaux en relation les uns avec les autres. Le premier axe concerne l'extension de nos travaux sur les méthodes d'agrégation issues des fonctions d'appariement aux réseaux de neurones profonds. Le second concerne l'apprentissage de modèles économes et fait écho à une thématique démarrée avec la thèse de Pierre Jacob en collaboration avec l'équipe ASTRE du laboratoire ETIS. Le troisième axe concerne l'extension de nos premiers travaux sur l'apprentissage distribués de CNN à n'importe quel modèle de deep learning en explorant les méthodes de distributions issue des protocoles Gossip.

4.1 Noyaux de matching approximés pour l'apprentissage profond de représentations

Dans ce projet, nous proposons d'allier le meilleur des deux mondes en définissant de nouvelles couches dans les réseaux de neurones profonds qui permettent d'extraire une information plus précise que la convolution. En particulier, l'objectif est de pouvoir définir des architectures profondes génériques qui peuvent être entraînées afin de répondre à des problèmes de reconnaissance très précis tels que la classification à grain très fin. nous décrivons dans cette section les outils utilisés ainsi que les deux objectifs scientifiques du projet.

4.1.1 Fonction de matching

Une des méthodes les plus populaires pour faire de la recherche par similarité consiste à apparier des caractéristiques locales (ou descripteurs locaux). L'appariement consiste pour chaque caractéristique de l'image requête à déterminer si l'image cible possède une caractéristique similaire. La similarité consiste simplement à compter le nombre d'appariements, ce qu'on appelle des *"fonctions de matching"*. Le problème de ces méthodes est double : d'une côté il est difficile de déterminer les caractéristiques locales à utiliser, et d'un autre côté, le comptage des appariements est extrêmement coûteux car il croît de manière quadratique avec le nombre de caractéristiques locales.

Inspiré de Hervé JÉGOU, Florent PERRONNIN et al. 2012, nous avons proposé dans NEGREL, PICARD et P.-H. GOSSELIN 2013 ; PICARD et P.-H. GOSSELIN 2013 ; PICARD et P.-H. GOSSELIN 2011 des méthodes de linéarisation de ces fonctions de matching qui consistent à injecter l'ensemble des caractéristiques locales d'une image dans un espace de représentation, et donc à résumer une ensemble de caractéristiques par une unique représentation, de telle sorte que le produit scalaire dans cet espace soit une approximation de la fonction de matching. Ces travaux ont fait l'objet d'un encadrement de thèse (Romain Negrel, thèse soutenue en 2014) et de plusieurs publications.

Plus récemment, nous avons proposé une extension dans PICARD 2016 qui permet de linéariser des co-occurrences d'appariements entre des descripteurs spatialement couplés. Cette extension est basée sur le même formalisme de linéarisation de fonctions de matching que nous proposons d'utiliser.

4.1.2 Objectif 1 : Fonctions de matching dans les réseaux convolutifs

Le premier objectif consiste à répondre à la question du choix des caractéristiques locales. Comme expliqué précédemment, les méthodes de deep-learning permettent d'apprendre des jeux de caractéristiques à partir des données. Dans le cas des réseaux convolutifs, il s'agit d'apprendre une succession de bancs de filtres. À la différence du filtrage traditionnellement utilisé en traitement d'images, le signal d'entrée est ici multidimensionnel en chaque position, et la convolution est alors définie de la manière suivante :

$$(f * h)(x) = \int \langle f(x-t), h(t) \rangle dt,$$

f étant l'image et h le filtre optimisé par apprentissage statistique.

On peut interpréter cette convolution comme une fonction de matching dans laquelle le produit scalaire correspond à l'appariement entre deux caractéristiques locales et l'intégrale correspond au comptage des appariements.

nous proposons donc de remplacer le produit scalaire par des fonctions d'appariement non-linéaire plus discriminantes telles que celles utilisées dans les travaux de recherche par similarité. Puis, nous proposons d'utiliser le formalisme de linéarisation que nous avons développé ces dernières années afin de pouvoir apprendre les filtres. Nous proposons également d'explorer les extensions récentes au co-occurences d'appariements dans ce contexte-ci. Enfin, pour éviter le problème de sur-apprentissage qui peut apparaître avec ces fonctions non-linéaires, nous proposons d'étudier des concepts de régularisation basés sur des factorisations tensorielles qui se prêtent particulièrement bien au formalisme que nous avons développé.

4.1.3 Objectif 2 : Apprentissage structuré d'information spatiale dans les fonctions de matching

Le second objectif concerne l'extension du premier objectif à l'apprentissage structuré d'information spatiale. En particulier, les travaux menés dans PICARD 2016 montrent que l'on peut linéariser des fonctions de matching qui conservent de l'information spatiale. Dans ces fonctions, certaines invariances spatiales (translation, échange, rotation, *etc*) peuvent être définies par construction et sont conservées par la méthode de linéarisation.

Nous proposons de découvrir ces invariances par apprentissage statistique sur un jeu de données. Pour cela, nous proposons d'implanter ces fonctions de matching linéarisées dans un réseaux convolutif profond afin de faire apparaître les invariances spatiales sous forme de paramètres du réseau qui sont donc optimisés au même titre que les filtres.

Nous proposons une extension non triviale qui consiste à modéliser l'invariance non plus au niveau de la collection d'images, mais des images elles-mêmes. Pour cela, les paramètres d'invariance sont considérés comme des variables latentes qu'il faut inférer pour chaque image.

Cette proposition s'inspire de la reconnaissance d'objet faiblement supervisée dans laquelle la position des objets est modélisée par des variables latentes comme dans DURAND, THOME et CORD 2015, et la transcrit dans le cadre des fonctions de matching spatialement couplées que nous comptons introduire dans les réseaux convolutifs.

4.2 Apprentissage de représentation avec fortes contraintes de ressources

Les modèles qui donnent aujourd'hui les meilleures représentations sont particulièrement gourmands en ressources. On parle de centaines de millions de paramètres (160M pour VGG19 de SIMONYAN et Andrew ZISSERMAN 2014) et de dizaines de GFLOP (milliards d'opérations en virgule flottante). C'est particulièrement vrai pour les CNN, mais aussi pour les architectures hybrides type NetVLAD de ARANDJELOVIC et al. 2016 ou même des agrégations VLAT ou Fisher Vectors qui seraient basées sur des sorties de CNN. De fait, ces modèles nécessitent des clusters de GPU très performants (on parle de cartes GPU réalisant des TFLOPS et ayant plusieurs dizaines de giga-octets de mémoire chacune, consommant plusieurs centaines de watt) pour l'entraînement, mais aussi pour l'inférence. L'ambition de ce projet est de pouvoir réduire ces coûts, ou au moins de trouver une compromis acceptable entre consommation de mémoire du modèle, nombre d'opérations de l'inférence et précision des résultats.

Cette thématique nous semble importante pour une raison d'applicabilité des méthodes de reconnaissance visuelle. En effet, les méthodes d'apprentissage de représentations visuelles sont bien plus performantes quand elle sont orientées vers une application particulière et entraînées sur les données correspondantes. Dès lors, cela veut dire que chaque domaine d'activité qui peut être amélioré par l'utilisation de ces méthodes doit se doter du matériel adéquat. L'utilisation de méthodes performantes ne pourra se faire que si leur coût mérite l'investissement, c'est à dire si les modèles produits sont suffisamment économes. C'est d'autant plus vrai quand on parle d'embarqué, domaine dans lequel il est impensable d'avoir plusieurs cartes consommant des centaines de watt dans le dispositif final.

4.2.1 Objectif 1 : Réduction de la taille complexité des modèles

Notre premier objectif ce situe sur le plan de la conception de modèles de représentations visuelles, desquels nous voulons réduire l'empreinte mémoire et la complexité calculatoire. Pour se faire, nous proposons d'étudier les techniques récemment proposées, notamment la quantification des poids comme dans CARVALHO et al. 2016, l'élagage des réseaux comme dans SUZUKI, HORIBA et SUGIE 2001, l'approximation des filtres convolutifs par des objets de complexité moindre comme dans IOANNOU et al. 2015, ou encore l'introduction de parcimonie comme dans X. LIU et al. 2017.

Nous proposons aussi d'évaluer leur influence sur des modèles hybrides (CNN+VLAT par exemple) et sur des modèles de réseaux profonds utilisant des fonctions de matching comme présenté précédemment. Nous pensons que ces modèles ayant de meilleures capacités de représentation que la convolution peuvent être moins demandeurs de précision numérique ou peuvent simplement se contenter de moins de ressources.

4.2.2 Objectif 2 : Modèles embarqués

Un second objectif déjà entamé lors du démarrage de la thèse de Pierre Jacob (Co-encadré avec Aymeric Histace - PU ETIS équipe ASTRE et Édouard Klein - Chercheur PJGN) en mars 2017 consiste à embarqué ces modèles de représentations visuelles sur des cartes à faible consommation énergétique et dont les ressources (mémoire, calcul) sont très limitées.

Cet objectif est beaucoup plus pratique puisqu'il s'agit de fixer un cadre d'exécution concret et de mesurer comment les différentes méthodes se situent sur le compromis entre consommation de ressources et la précision des résultats obtenus. Parmi le matériel envisagé, on se concentrera

sur trois types essentiellement : les petits ordinateurs embarqués (type Raspberry Pi), les cartes disposant d'un accélérateur type GPU (NVidia Tegra par exemple) et les cartes intégrant un SOC FPGA (Zynq chez Xilinx et DE-SoC chez Altera). Les deux premières catégories de matériel sont assez simples d'utilisation et ne diffèrent pas vraiment de l'utilisation d'ordinateurs classiques, si ce n'est la quantité de mémoire limitée. Pour les cartes embarquant un FPGA il s'agit de poursuivre la collaboration avec les électroniciens de l'équipe ASTRE et d'identifier les opérations les plus consommatrices afin de créer les accélérateurs matériels correspondants, ou bien d'adapter les architectures afin d'exploiter au mieux les accélérateurs disponibles.

4.3 Apprentissage profond distribué

Le troisième axe de notre projet de recherche concerne l'apprentissage profond distribué et est la continuité des travaux présentés en 3.3.2. Il s'agit aussi d'avoir une interaction cohérente avec le projet sur les modèles embarqués détaillé dans la section précédente. En effet, si nous nous orientons vers des modèles qui doivent s'exécuter sur du matériel ayant de faibles ressources et en considérant l'évolution de l'interconnexion de ce type de cartes (notamment l'expansion de l'internet des objets), il nous semble indispensable d'explorer l'entraînement de modèles sur une flotte de matériels inter-connectés.

4.3.1 Objectif 1 : Entraînement par distribution de modèles

Nous envisageons un objectif qui est la continuité de nos travaux sur l'entraînement distribué de modèles de représentations (CNN, VLAT, hybrides) à l'aide de protocoles sum-weight gossip. Dans les travaux que nous avons exposés, il s'agissait de distribution de données : chaque nœud possède une copie du modèle et traite des exemples différents. La communication sert à assurer le consensus entre les différents modèles des différents nœuds. Cette solution à l'avantage d'être simple à mettre en œuvre, mais elle nécessite de pouvoir faire tenir un modèle entier sur chaque nœud.

Nous proposons alors d'explorer la deuxième grande stratégie en calcul réparti : la distribution de modèles. Dans ce cas, chaque nœud possède une partie du modèle et les communications servent à faire transiter les données (initiales, ou sorties intermédiaires de parties de modèle). Nous voyons deux intérêts à ce type de distribution. D'une part, cela permet d'optimiser des modèles beaucoup plus gros que ce permettent les ressources disponibles sur chaque carte car le découpage du modèle peut être dimensionné justement par rapport à ces ressources. D'autre part, la duplication de certaines parties du modèle sur plusieurs nœuds peut être vue comme similaire aux méthodes d'exploration stochastique de modèle telles que le dropout de SRIVASTAVA et al. 2014, les réseaux à profondeur stochastique de HUANG et al. 2016, ou le maxout de GOODFELLOW et al. 2013. Le consensus est alors à imposer entre parties équivalentes du modèle de manière à s'assurer que les inférences obtenues en sortie de modèle soient identiques peu importe le chemin pris dans le graphe du modèle distribué.

5. Sélection d'articles

Web scale image retrieval using compact tensor aggregation of visual descriptors

Romain Negrel, David Picard, Philippe-Henri Gosselin

Web scale image retrieval using compact tensor aggregation of visual descriptors

Romain Negrel, David Picard and Philippe-Henri Gosselin
ETIS/ENSEA - University of Cergy-Pontoise - CNRS, UMR 8051
6, avenue du Ponceau, BP44, F95014 Cergy-Pontoise, France
{*romain.negrel,picard,gosselin*}*@ensea.fr*

***Abstract*—The main issues of web scale image retrieval are to achieve a good accuracy while retaining low computational time and memory footprint. In this paper, we propose a compact image signature by aggregating tensors of visual descriptors. Efficient aggregation is achieved by preprocessing the descriptors. Compactness is achieved by projection and quantization of the signatures. We compare our method to other efficient signatures on a 1 million images dataset, and show the soundness of the approach.**

***Index Terms*—Image/video retrieval, Image Processing and Computer Vision.**

I. Introduction

With the globalization of internet, collections with tremendous amounts of images are available. For instance, more than 6 billions images were hosted on Flickr[1] in 2011. Images similarity search in these web scale databases is thus becoming a hot topic in the multimedia indexing community. Given a query image, image similarity search is to find similar images in a huge collection of images. Similar images are defined as the images with similar visual content (same object, same action, same scene, ...), without any meta data such as textual tags, time or location.

The two main problems of this task are the search time and the storage size of indexes. To index an image, common systems use a set of local visual descriptors extracted from images called "bag of descriptors". The main problem of bags of descriptors is their prohibitive storage cost. Many methods consist in computing a lightweight signature using the bag of descriptors.

In this paper, we propose a very compact signature which gives good performance in similarity search with a linear metric. Our signature is based on compressed aggregation of tensor products of local descriptors. In the first step, we perform a preprocessing on the descriptors. Then, we aggregate tensors of preprocessed descriptors. Finally, we compress the signature by projection in a well chosen subspace. Extra compression is achieved by binary quantization of the projected signatures.

The paper is organized as follows: First, we give an overview of the state-of-the-art to compute similarity between images. Then we detail our propositions in the third section. In section IV, we present results for similarity search tasks on well known web scale datasets and compare with recent methods. In the last section, we conclude and discuss the possible improvements of our method, as well as the ongoing challenges in web scale image indexing.

[1]As stated on Flickr's blog on August 4th 2011.
http://blog.flickr.net/en/2011/08/04/6000000000/

II. State-of-the-art

Most similarity search methods use a two steps scheme. In the first step, a set of local visual descriptors is extracted from the images. Regions of interest in the image can be selected by automatic point of interest detection, or by uniform sampling. The most commonly used visual descriptors are highly discriminant local descriptors [1] (HOG, SIFT, SURF, ...). The set of descriptors extracted from an image is called a *bag*. We denote by $\mathbf{B}_i = \{\mathbf{b}_{ri}\}_r$ the set of descriptors $\mathbf{b}_{ri} \in \mathbb{R}^D$ in image i. $\mathcal{B}$ is the union of $\mathbf{B}_i$ for all image i in the dataset.

In the second step, a similarity between two bags of descriptors is defined. There are two main approaches to compute such similarities. The first approach performs a straight matching between descriptors in bags, for instance using a voting approach. The second approach is to compute a signature (generally a single vector) from the bag of descriptors, and then to use similarity measures between vectors.

In both cases, the similarity measure is used to sort all images of the database according to a query image. To work with web-scale image databases, it is essential to have extremely fast similarity computation.

A. Voting based approaches

In the approaches based on voting, the descriptors of the query image are matched to the descriptors of the dataset $\mathcal{B}$. Each descriptor of the query votes for its k-Nearest Neighbors (k-NN) in $\mathcal{B}$. Then each image counts the number of votes obtained by its descriptors. The image with the most votes is the most similar image. The similarity score of bags $\mathbf{B}_j$ relative to a query $\mathbf{B}_i$ is thus obtained with the following equation:

$$\mathrm{k}(\mathbf{B}_i, \mathbf{B}_j) = \sum_{\mathbf{b}_{ri} \in \mathbf{B}_i} \mathrm{card}\left(\underset{\mathcal{B} \setminus \mathbf{B}_i}{k\text{-NN}}(\mathbf{b}_{ri}) \cap \mathbf{B}_j \right). \quad (1)$$

Naive k-NN search has a complexity linear with the number of descriptors in $\mathcal{B}$, which is prohibitive at web scale. Computation time can be saved using approximated k-NN search,

where a subset $\mathcal{B}'(\mathbf{b})$ of candidate is selected thanks to a sub-linear algorithm. A subset $\mathcal{B}'(\mathbf{b})$ is defined for each query descriptor $\mathbf{b}$ as:

$$\mathcal{B}'(\mathbf{b}) = \{\mathbf{b}_i \in \mathcal{B} | \mathbb{P}\left(\mathrm{d}(\mathbf{b}_i, \mathbf{b}) < R\right) > P\} \quad (2)$$

with R the distance threshold, P a probability of being similar and d a distance function.

Locality Sensitive Hashing (LSH) [2] uses hash functions to produce the descriptor subset. The hash function h is defined such that:

- if $\mathrm{d}(\mathbf{b}_i, \mathbf{b}) \leq R_1$ then $\mathbb{P}(\mathrm{h}(\mathbf{b}_i) = \mathrm{h}(\mathbf{b})) \geq P_1$,
- if $\mathrm{d}(\mathbf{b}_i, \mathbf{b}) \geq R_2$ then $\mathbb{P}(\mathrm{h}(\mathbf{b}_i) = \mathrm{h}(\mathbf{b})) \leq P_2$,
- $R_2 > R_1$,
- $P_1 > P_2$.

By properly choosing the (R_1, R_2, P_1, P_2) parameters, it is guaranteed that the descriptors that are colliding (same hash) have a high probability of being similar.

Another approach is to split the descriptor space with a hierarchical tree structure such that all elements of a leaf are very similar. Lejsek et al. [3] propose a method called Nearest Vector Tree (NV-Tree). In this method each node of the tree contains a subset of the descriptors, and each child node a splitting of this subset. The nearest neighbor candidates of query descriptor are all elements of the leaf to which it belongs.

Voting based approaches give good results in similarity search with very short response time. However, these approaches require the storage of all descriptors in $\mathcal{B}$ and also the index structure for approximate nearest neighbor search. In [4], the authors estimate around 100-500 bytes per descriptor for the LSH indexing. For web scale databases with more than 1 billion descriptors (around 1 million images), the storage cost of these approaches is prohibitive and not tractable.

B. Kernels on Bags approaches

Kernels on Bags approaches are an extension of kernel functions commonly used in machine learning. These approaches are similar to voting based approaches as they estimate the number of similar descriptors. Unlike voting based approaches, they use similarity functions to weight the vote. The similarity function between two descriptors is called minor kernel and is defined as:

$$\mathrm{k} : (\mathbb{R}^D, \mathbb{R}^D) \rightarrow [0, 1]. \quad (3)$$

The minor kernel is chosen such that, for similar descriptors $\mathrm{k}(\cdot, \cdot) \approx 1$ and for dissimilar descriptors $\mathrm{k}(\cdot, \cdot) \approx 0$.

In [5], the authors proposed to compute the sum of similarity of all possible pairings between elements of $\mathbf{B}_i$ and $\mathbf{B}_j$:

$$\mathrm{K}(\mathbf{B}_i, \mathbf{B}_j) = \sum_r \sum_s \mathrm{k}(\mathbf{b}_{ri}, \mathbf{b}_{sj}). \quad (4)$$

Thus the higher the number of similar descriptors the more the two bags are similar.

However, such kernel on bags produces a similarity of low variance. To overcome this problem, Lyu proposed in [6] to raise the minor kernel to power p. Therefore only highly similar descriptors are considered.

Kernels on bags approaches have good results in the fields of image retrieval and classification, but are rarely used in web scaled problems [7]. Indeed the computational cost of these approaches is prohibitive when the size of the bags becomes too large, especially with dense sampling extraction strategies. To compute the similarity between two bags of 10,000 descriptors, 100 million evaluations of the minor kernel have to be performed.

To address these computational problems, only the most similars descriptors of the bags can be considered, like in voting based approaches. In [8] the problem is seen as the following kernel on bag:

$$\mathrm{K}_{fast}(\mathbf{B}_i, \mathbf{B}_j) = \sum_r \sum_s \mathrm{k}(\mathbf{b}_{ri}, \mathbf{b}_{sj}) f(\mathbf{b}_{ri}, \mathbf{b}_{sj}), \quad (5)$$

with $f(\cdot, \cdot)$ a indicator function based on k-NN:

$$f(\mathbf{b}_r, \mathbf{b}_s) = \begin{cases} 1 & \text{if } d(\mathbf{b}_r, \mathbf{b}_s) < R, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

$f(\mathbf{b}_r, \mathbf{b}_s)$ is obtained by previously described methods such as LSH. These methods result in fast kernels on bags, but they have same problem of storage cost as voting based approaches.

C. Statistical approaches

Statistical approaches have been inspired by text retrieval methods. In these approaches, we assume a visual codebook composed of descriptor prototypes (called visual words) can be computed. A bag can then be described by a statistical analysis of occurrences of visual words. The visual codebook is generally computed by a clustering algorithm (*e.g.*, k-means) on a large set of descriptors. We denote by C the number of visual words in the codebook.

The first method of this kind, named Bag of Words (BoW)[9] counts the number of descriptors belonging to each cluster. The size of the signature is C.

Avila et al. [10] suggested an extension of BoW called Bag Of Statistical Sampling Analysis (BOSSA). This method aims to keep more information on the distribution of descriptors in the clusters. In this method, histograms of distances from centers of clusters are computed. The signature size is $C \times H$ with H the number of bins in distance histograms.

However, BoW approach is subject to codeword ambiguity. This problem arises when a descriptor lies at the boundary between two clusters or away from all the cluster centers. To solve this problem Gemert et al. [11] proposed a robust alternative to histograms using kernel density estimation (typically Gaussian functions) to smooth the local neighborhood of descriptors. This method allows for a soft assignment of a descriptor to several codewords. The size of the signature is C. These approaches obtain better results than BoW approaches.

D. Coding approaches

The coding approaches are borrowed from the telecommunications and signal processing communities. The main idea of these approaches is to use coding methods based on reconstruction problems [12] (notably used in data compression). In

most cases the encoding methods minimize a reconstruction error.

The signature is obtained with a two-step scheme. The first step consists in encoding each descriptor of the bag (coding step). The second step consists in aggregating all codes in a single vector (pooling step). Many coding functions have been proposed with different structural constraints on the code.

A sparsity regularization term is usually added in order to have good compression and aggregation properties on the code. Wang et al. [13] proposed a coding constraint such that similar descriptors are always coded with the same visual words by adding a Locality-constrained term:

$$q_{llc}(\mathbf{B}_i) = \arg\min_{C_i} \sum_r \|\mathbf{b}_{ri} - \mathbf{D}\mathbf{c}_{ri}\|^2 + \lambda \|\mathbf{d}_{ri} \odot \mathbf{c}_{ri}\|^2 \quad (7)$$

with $\mathbf{d}_{ri}$ a locality constraint and $\odot$ the Hadamard product.

The most common polling methods are:

- sum pooling : $\mathbf{c}_i = \sum_r \mathbf{c}_{ri}$
- max pooling : $\mathbf{c}_i = \max_r(\mathbf{c}_{ri})$

where "max" functions in a row-wise manner, returning a vector of size C.

E. Model Deviation approaches

Model Deviation approaches are based on a model of the descriptors space. The signature of a bag of descriptors is the deviation between the descriptors of the bag and the model.

Recently, Perronnin et al. [14] proposed a successful method called Fisher Vectors. The authors proposed to model the descriptors space by a probability density function denoted by u_λ of parameters λ. To describe the image, they compute the derivative of the log-likelihood of image descriptors to the model:

$$\mathcal{G}_\lambda^{\mathbf{B}_i} = \frac{1}{T} \nabla_\lambda \log u_\lambda(\mathbf{B}_i). \quad (8)$$

The model used is a Gaussian Mixture Model (GMM) of parameters $\boldsymbol{\mu}_c$ and σ_c. Elements of the Fisher Vector for each Gaussian c can be written as:

$$\mathcal{G}_{\boldsymbol{\mu},c}^{\mathbf{B}_i} = \frac{1}{T\sqrt{\omega_c}} \sum_r \gamma_c(\mathbf{b}_{ri}) \left(\frac{\mathbf{b}_{ri} - \boldsymbol{\mu}_c}{\sigma_c} \right), \quad (9)$$

$$\mathcal{G}_{\boldsymbol{\sigma},c}^{\mathbf{B}_i} = \frac{1}{T\sqrt{\omega_c}} \sum_r \gamma_c(\mathbf{b}_{ri}) \left[\frac{(\mathbf{b}_{ri} - \boldsymbol{\mu}_c)^2}{\sigma_c^2} - 1 \right]. \quad (10)$$

Where $\mathbf{b}_{ri}$ are the descriptors of image i, $(\omega_c, \boldsymbol{\mu}_c, \sigma_c)$ are the weight, mean and standard deviation of Gaussian c, and $\gamma_c(\mathbf{b}_{ri})$ the normalized likelihood of $\mathbf{b}_{ri}$ to Gaussian c. The final descriptor is obtained by concatenation of $\mathcal{G}_{\boldsymbol{\mu},c}^{\mathbf{B}_i}$ and $\mathcal{G}_{\boldsymbol{\sigma},c}^{\mathbf{B}_i}$ for all Gaussians. Fisher Vectors achieve very good results [14]. However, Fisher Vectors are limited to the simple model of mixtures of Gaussians with diagonal covariance matrices. Moreover, the GMM algorithm is computationally very intensive.

Jegou et al. [15] proposed a simplified version of Fisher Vector by aggregating local descriptors, called Vectors of Locally Aggregated Descriptors (VLAD). They proposed to model the descriptors space by a small codebook obtained by clustering a large set of descriptors. The model is simply the sum of all centered descriptors $\mathbf{B}_{ci} = \{\mathbf{b}_{rci}\}_r \subseteq \mathbf{B}_i$ from image i and cluster c:

$$\boldsymbol{\nu}_{ci} = \sum_r \mathbf{b}_{rci} - \boldsymbol{\mu}_c \quad (11)$$

with $\boldsymbol{\mu}_c$ the center of cluster c. The final signature is obtained by a concatenation of $\boldsymbol{\nu}_c$ for all c. The signature size is $D \times C$.

Picard et al. [16] proposed an extension of VLAD by aggregating tensor products of local descriptors, called Vector of Locally Aggregated Tensors (VLAT). They proposed to use the covariance matrix of the descriptors of each cluster. Let us denote by "$\boldsymbol{\mu}_c$" the mean of cluster c and "$\mathcal{T}_c$" the covariance matrix of cluster c with $\mathbf{b}_{rci}$ descriptors belonging to cluster c:

$$\boldsymbol{\mu}_c = \frac{1}{|c|} \sum_i \sum_r \mathbf{b}_{rci} \quad (12)$$

$$\mathcal{T}_c = \frac{1}{|c|} \sum_i \sum_r (\mathbf{b}_{rci} - \boldsymbol{\mu}_c)(\mathbf{b}_{rci} - \boldsymbol{\mu}_c)^\top, \quad (13)$$

with $|c|$ being the total number of descriptors in cluster c.

For each cluster c, the signature of image i is the sum of centered tensors of centered descriptors belonging to cluster c:

$$\mathcal{T}_{ic} = \sum_r (\mathbf{b}_{rci} - \boldsymbol{\mu}_c)(\mathbf{b}_{rci} - \boldsymbol{\mu}_c)^\top - \mathcal{T}_c. \quad (14)$$

Each $\mathcal{T}_{ic}$ is flattened into a vector $\mathbf{v}_{ic}$. The VLAT signature $\mathbf{v}_i$ for image i consists of the concatenation of $\mathbf{v}_{ic}$ for all clusters:

$$\mathbf{v}_i = (\mathbf{v}_{i1} \dots \mathbf{v}_{iC}). \quad (15)$$

For better results, normalization steps are added:

$$\mathbf{x}_i = \frac{\mathbf{v}'_i}{\|\mathbf{v}'_i\|}, \quad \forall j, \mathbf{v}'_i[j] = \text{sign}\left(\mathbf{v}_i[j]\right) |\mathbf{v}_i[j]|^\alpha, \quad (16)$$

with α typically set to 0,5. $\mathbf{x}_i$ is the normalized VLAT signature.

As the $\mathcal{T}_{ic}$ matrices are symmetric, only the diagonal and the upper part are kept while flattening $\mathcal{T}_{ic}$ into a vector $\mathbf{v}_{ic}$. The size of the signature is then $C \times \frac{D \times (D+1)}{2}$.

III. Compact VLAT

In this paper, we propose to improve VLAT by increasing their discriminative power while reducing their size. The first improvement consists in preprocessing the descriptors to optimize the model $(\boldsymbol{\mu}_c, \mathcal{T}_c)_c$. Then we present a method to reduce the size of the VLAT signatures while preserving the dot product. Our dimensionality reduction is based on linear projections that have been made more efficient thanks to the model optimization.

A. PCA cluster-wise of VLAT

The signature is composed of deviations between covariance matrices of the clusters and covariance matrices of the image descriptors. To optimize this deviation, we propose to perform a Principal Component Analysis (PCA) within each cluster.

First, we compute the Takagi decomposition of the covariance matrix of each cluster c:

$$\mathcal{T}_c = \mathbf{V}_c \mathbf{D}_c \mathbf{V}_c^\top, \quad (17)$$

where $\mathbf{D}_c$ is a real non-negative diagonal matrix (eigenvalues), and $\mathbf{V}_c$ is unitary (eigenvectors). Then we project the centered descriptors belonging to c on the eigenvectors:

$$\mathbf{b}'_{rci} = \mathbf{V}_c^\top (\mathbf{b}_{rci} - \boldsymbol{\mu}_c). \tag{18}$$

Combining eq.(18) and eq.(14), we get:

$$\begin{aligned}\mathcal{T}_{ic} &= \mathbf{V}_c^\top \left(\sum_r (\mathbf{b}_{rci} - \boldsymbol{\mu}_c)(\mathbf{b}_{rci} - \boldsymbol{\mu}_c)^\top - \mathcal{T}_c \right) \mathbf{V}_c \\ &= \sum_r \mathbf{V}_c^\top \left((\mathbf{b}_{rci} - \boldsymbol{\mu}_c)(\mathbf{b}_{rci} - \boldsymbol{\mu}_c)^\top \right) \mathbf{V}_c - \mathbf{D}_c \\ &= \sum_r \left(\mathbf{V}_c^\top (\mathbf{b}_{rci} - \boldsymbol{\mu}_c) \right) \left(\mathbf{V}_c^\top (\mathbf{b}_{rci} - \boldsymbol{\mu}_c) \right)^\top - \mathbf{D}_c.\end{aligned}$$

The new VLAT signature of image i in cluster c is the sum of tensors of projected descriptors $\mathbf{b}'_{rci}$ belonging to cluster c, centered by $\mathbf{D}_c$:

$$\mathcal{T}_{ic} = \sum_r \mathbf{b}'_{rci} \mathbf{b}'^\top_{rci} - \mathbf{D}_c. \tag{19}$$

The optimized VLAT signature is obtained by the same steps of flattening, concatenation and normalization as the standard signature. This optimization has the very interesting property that most of the variance is concentrated among the first dimensions of each cluster.

B. Compact VLAT

We propose to reduce drastically the size of the VLAT signature while retaining its discriminative power. We seek a linear projection into a subspace in which the original similarity between two signatures is retained. Hence, we want to solve the following problem:

$$\begin{gathered}\mathbf{P}_N = \arg\min_{\mathbf{A}} \sum_{\mathbf{x}_i \in \mathcal{S}} \sum_{\mathbf{x}_j \in \mathcal{S}} (\langle \mathbf{x}_i | \mathbf{x}_j \rangle - \langle \mathbf{A}^\top \mathbf{x}_i | \mathbf{A}^\top \mathbf{x}_j \rangle)^2 \\ \text{s.t. } \mathbf{A} \in \mathcal{M}_{S,N} \text{ with } N < L \ll W\end{gathered}$$

with $\mathcal{S}$ a training set of L images, N the size of subspace and W the size of VLAT signature. We solve this problem by performing a low rank approximation of the Gram matrix and computing the linear projectors of the associated subspace.

We compute the Gram matrix of a training set $\mathcal{S}$ ($L \times L$):

$$\mathbf{G}_{ij} = (\mathbf{x}_j^\top \mathbf{x}_i)_{ij} \tag{20}$$

Then, we perform the Takagi factorization of $\mathbf{G}$:

$$\mathbf{G} = \mathbf{U}\mathbf{L}\mathbf{U}^\top \tag{21}$$

$$\mathbf{L} = \operatorname{diag}(\lambda_1, \ldots, \lambda_L) \tag{22}$$

$$\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_L) \tag{23}$$

with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$ and $\mathbf{u}_i$ the eigenvector associated with the eigenvalue λ_i . We denote by $\mathbf{L}_N$ the matrix with the N largest eigenvalues of $\mathbf{L}$ on the diagonal:

$$\mathbf{L}_N = \operatorname{diag}(\lambda_1, \ldots, \lambda_N) \tag{24}$$

and we denote by $\mathbf{U}_N$ the matrix of the N first eigenvectors of $\mathbf{U}$:

$$\mathbf{U}_N = (\mathbf{u}_1, \ldots, \mathbf{u}_N). \tag{25}$$

The approximated Gram matrix is then:

$$\mathbf{G}_N = \mathbf{U}_N \mathbf{L}_N \mathbf{U}_N^\top \tag{26}$$

We compute the projection matrix signatures in the subspace:

$$\mathbf{P}_N = \mathbf{X}\mathbf{U}_N \mathbf{L}_N^{-1/2}. \tag{27}$$

For each image, we compute the projection of VLAT in the sub-space as:

$$\mathbf{y}_i = \mathbf{P}_N^\top \mathbf{x}_i. \tag{28}$$

$\mathbf{y}_i$ contains an approximate and compressed version of $\mathbf{x}_i$. The subspace defined by the projectors preserves most of the similarity even for very a small dimension and for small training sets because the optimization of section III-A concentrated the information in a small number of dimensions. One can note this procedure is analog to that of a kernel PCA with a linear kernel.

For a more robust similarity, we use the dot product associated with Mahalanobis distance:

$$\mathrm{k}(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}_j^\top \mathbf{L}_N^{-1} \mathbf{y}_i. \tag{29}$$

This normalization can be integrated in our projection step:

$$\mathrm{k}(\mathbf{y}_i, \mathbf{y}_j) = (\mathbf{L}_N^{-\frac{1}{2}} \mathbf{y}_j)^\top (\mathbf{L}_N^{-\frac{1}{2}} \mathbf{y}_i), \tag{30}$$

$$\mathbf{y}'_i = \mathbf{L}_N^{-\frac{1}{2}} \mathbf{P}_N^\top \mathbf{x}_i. \tag{31}$$

The compact signature has a size N, therefore $4 \times N$ bytes of storage space (in single precision) are used.

C. Binarized Compact VLAT

The storage size of signatures is a key point in the field of web scale similarity search. To produce ultra compact signatures, we propose to perform a binary quantization of compact VLAT signatures. We assume that signatures are sampled from a normal distribution which is consistent with the projections used in eq. (31). To maximize the retained information, we propose to set the threshold such that each class contains 50% of density. The binarized compact signature is then computed as:

$$\hat{\mathbf{y}}_i[j] = \begin{cases} 1 & \text{if } \mathbf{y}_i[j] \geq 0, \\ -1 & \text{otherwise.} \end{cases} \tag{32}$$

This quantization reduces the signatures size to $N/8$ bytes of storage space.

To sum up, our signature is computed in three steps: First we perform an optimization of the model with a PCA for each cluster of codebook. Secondly, we compute the VLAT signatures with preprocessed descriptors. Then, we compress the signatures by projection onto a subspace with a low rank approximation of some training Gram matrix. Finally, we reduce the storage size with a binary quantization of Compact VLAT signatures.

Fig. 1. Images from Holidays dataset.

IV. Experiments

In this section, we evaluate and compare our Compact VLAT signatures and our Binarized Compact VLAT signatures with the state-of-the-art. We use two evaluation datasets (INRIA Holidays and Oxford datasets) and three additional independent datasets to evaluate the performance of all methods:

INRIA Holidays dataset (Fig. 1) is a set of images drawn from personal holidays photos, created to test similarity search methods. It contains 1,491 images gathered in 500 subgroups, each of them being a distinct scene or object.
Oxford dataset is a set of images collected from Flickr by searching for particular Oxford landmarks. It contains 5,062 images gathered in 11 different landmarks, each represented by 5 possible queries.
Holidays Flickr1M dataset is a set of high quality pictures from Flickr. It contains 1 million images, commonly used as distractors for testing the Holidays dataset in large scale context.
Oxford Flickr100k dataset is a set of high quality pictures from Flickr. It contains 100,000 images, commonly used as distractors for testing the Oxford dataset in large scale context.
Holidays Flickr60K dataset is a set of high quality pictures from Flickr. It contains 60,000 images, commonly used as training set.

The three Holidays datasets are completely independent and include SIFT descriptors [17]. For the two Oxford datasets, we use a dense extraction of HOG descriptors.

For the INRIA Holidays dataset, we use the same evaluation setup as Jegou et al. [15] and for the Oxford dataset, we use the same evaluation setup as Philbin et al. [18].

For both, the accuracy of search is measured by the mean Average Precision (mAP).

To evaluate our methods at web scale, we merge a large images set (distactors set) with the standard evaluation dataset. For the INRIA Holidays dataset, we use the Flickr1M dataset as distactors set and for the Oxford datasets, we use the Flickr100k dataset as distactors set.

To study the influence of the parameters of our method, we use the INRIA Holidays dataset. For all experiments on INRIA Holidays, we compute a set of codebooks (32, 64, 128 visual words) with SIFT descriptors from the Flickr60K dataset. For each cluster of each codebook, we compute their mean and covariance matrix $(\boldsymbol{\mu}_c, \mathcal{T}_c)_c$ with SIFT descriptors of the Flickr60K dataset. We use these covariance matrices to compute the cluster-wise PCA. To compute the projectors of Compact VLAT signatures, we use a sample of 10k images extracted from Flickr60K dataset.

	32	64	128	256	9000	FULL
VLAT	-	-	-	-	-	64.0
PVLAT	-	-	-	-	-	**66.4**
CVLAT	46.3	49.6	53.3	54.9	**58.2**	-
CPVLAT	**47.1**	**51.9**	**53.9**	**55.6**	55.0	-
CVLAT-M	47.1	50.0	55.1	57.5	**70.0**	-
CPVLAT-M	**48.5**	**54.3**	**57.3**	**60.6**	**70.0**	-

TABLE I
Parameters study on Holidays dataset with $D = 64$ (mAP).

In this section we denote by D the number of clusters in the codebooks and by N the size of the signatures. We denote by "CVLAT" the Compact VLAT signatures, "CPVLAT" for the Compact VLAT signatures with Cluster-wise PCA and "-M" suffix denotes the use of the dot product associated with Mahalanobis distance.

A. Parameters study

In this section, we study the behavior of Compact VLAT signatures according to their parameters. All experiments are done with Holidays dataset, unless another setup is specified.

Table I shows the influence of the different stages of our method on the mAP. Rows are the differents configuration of our methods stages and columns represent the size N of the signature ("FULL" means uncompressed signature). We observe a gain of 2.4% between VLAT and PVLAT which highlights the improvements brought by the model optimization. Rows 3 and 4 show that the model optimization allows to retain more information at higher compression ratio (typically $N \leq 256$). We can see that using of the dot product associated with Mahalanobis distance greatly increases the performance with compressed signature. For $D = 64$ and $N = 256$, we divided by 2,000 the signature size for a loss of only 3.4% of mAP.

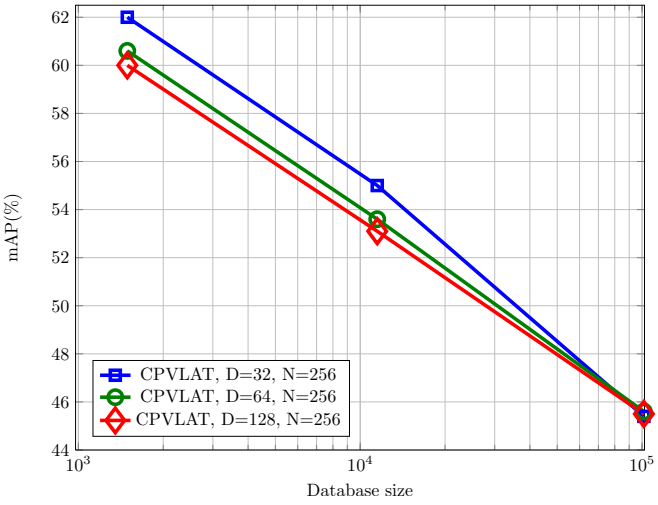


Fig. 2. Comparison of Compact VLAT signatures with cluster-wise PCA as a function of the database size and the D numbers of clusters in codebook.

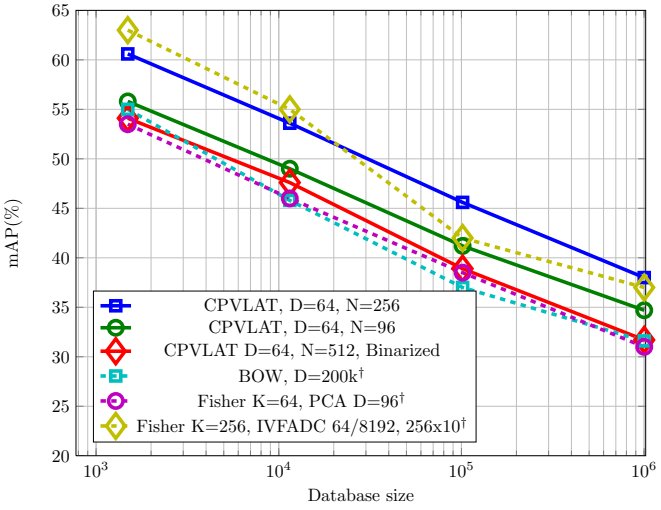


Fig. 3. Comparison of state-of-the-art signatures as a function of the database size († extract from [15]).

	CPVLAT, D=64					
N	16	32	64	128	256	512
Standard	22.1	33.5	38.9	42.7	45.6	47.6
Binarized	2.1	9.6	18.3	28.3	34.7	38.9

TABLE II
COMPARISON OF BINARIZED AND STANDARD COMPACT VLAT SIGNATURES WITH CLUSTER-WISE PCA ON 100K EXTENDED HOLIDAYS DATASET (MAP).

To study the influence of the number D of clusters on CPVLAT signature, we fixed the size to $N = 256$. Figure 2 shows the variation of the mAP according to the size of the database on Extended Holidays dataset. We show that for databases with fewer images, a small codebook gives better results. However, the results become similar when numbers of images in the database increases. This shows that a medium codebook ($D = 64$) leading to less computational time of projection gives sufficiently good results at larger scale.

To study the influence of binarization, we consider CPVLAT signature, and a codebook of 64 visual words. Table II shows the mAP (%) with the columns representing the size N of the signatures. We show that binarization reduces drastically the accuracy. However, since it leads to a strong compression of the storage size, a larger number of projectors can then be retained. Furthermore, we note that the loss of accuracy is lower for larger projections.

B. Comparison with the state-of-the-art

In this section, we compare our signatures with the results of [15] on the Extended Holidays dataset and with the results of [18] on the Oxford dataset.

For the Holidays dataset, we compute the CPVLAT signatures with a codebook of 64 visual words. CPVLAT signatures are computed with a subspace projection of size $N = 96$ and $N = 256$. We compute the Binarized CPVLAT signatures with a subspace projection of size $N = 512$. Results are shown in Figure 3.

Compare to BoW computed with a codebook of 200k visual words, CPVLAT signatures computed with a smaller codebook give better results. With CPVLAT signatures of size $N = 96$, we have a gain of $\sim 5\%$ of mAP, while our signatures are about 2,000 times smaller.

Compared to the Fisher signature computed with a codebook of size 64 (different from our codebook) and keeping the first 96 dimensions with PCA, we obtain better results with same size of codebook and signature. We also have similar results with Binarized CPVLAT signatures. However, our storage size is much smaller with 64 bytes compared to 384 bytes for the Fisher signature with PCA.

Compared to the Fisher Vectors signature indexed by IVFADC with a codebook of size 256, we obtain lower results on small size databases. However, this signature is more sensitive to the increased number of images. For more than 10k images, we have better results with a smaller codebook.

To test the universality of our method, we use default parameters on Oxford datasets. We use Oxford images as training set for all parameters. We compute the VLAD, VLAT, and CPVLAT signature with a dense extraction of HOG descriptors. We use the same codebook of 64 visual words for all signatures. For compressed VLAD signatures, we use the same protocol as in [18]. Results are shown in Table III.

We can see that using dense extraction of HOG descriptors increases the performance of VLAD signature of 6%. The compression of VLAD@HOG signature has about the same loss that the compression of VLAD in [18]. We observe that the VLAT signature has much better performance than the VLAD signature. With this setup, we observe that our method has much better performances at large scale for the same size (around 20% mAP improvement).

C. Scalability

In this section, we study the influence of the storage size of our signatures. We compute the CPVLAT signatures with

	Oxford	Oxford + 100k
Fisher [18]	31.7	-
VLAD [18]	30.4	-
Fisher-PCA (N=128) [18]	24.3	-
VLAD-PCA (N=128) [18]	25.7	-
VLAD@HOG	36.6	-
VLAT@HOG	50.7	-
PVLAT@HOG	54.2	-
VLAD@HOG-PCA (N=128)	32.7	25.6
CPVLAT@HOG (N=128)	**54.3**	**46.6**

TABLE III
PERFORMENCE OF THE ROW DESCRIPTORS AS WELL AS DESCRIPTORS COMPRESSED ON OXFORD DATASET AND OXFORD DATASET + 100K DISTRACTORS (MAP).

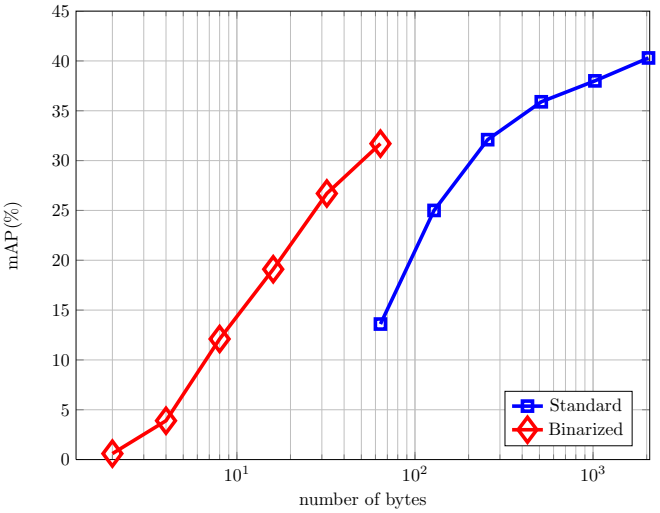


Fig. 4. Comparison of Binarized and Standard Compact VLAT signatures with cluster-wise PCA on 1M Extended Holidays dataset ($D = 64$).

varying number of selected projectors and the same binarized versions. We set the size D of the codebook to 64. In Figure 4, we plot the standard CPVLAT signature and Binarized CPVLAT signature against the storage size for 1 million images. We observe that binarized version of the signature leads to much better results at similar storage size. For a storage size of 64 bytes, we obtain a mAP of 14.6% with a standard CPVLAT signature and a mAP of 31.6% with a Binarized CPVLAT signature (gain of 17% of mAP). With the Binarized CPVLAT signature of dimension $N = 512$, all signatures of the Extended Holidays dataset are stored in only 61 MB of memory.

V. CONCLUSION

In this paper, we proposed a new compact signature for similarity search in web scale databases called Compact VLAT. Our method belongs to the model deviation approaches. First, we preprocess descriptors using PCA in each cluster to ensure good properties for the compression step. We use an aggregation of tensors of preprocessed descriptors. Then we compress the signatures using projections onto a subspace analog to kernel-PCA. We carried out similarity search experiments on the Extended INRIA Holidays dataset (1M images) and Oxford dataset (100k images). We presented the impact of the signatures size on its performance. We compared our results with popular methods, and showed the competitiveness of our approach for large scale datasets.

Future works include the following issues: First, combining VLAT and VLAD signatures before performing the projection step; Secondly, using a soft assignment of descriptors inspired by coding techniques; Third, using a non-binary quantization for the extra compression step. Finally, we want to stress that the next challenge to be addressed in web scale image retrieval will be the loss of performances occurring when the number of distractors increases.

REFERENCES

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, 2004.
[2] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, ser. VLDB '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 518–529.
[3] H. Lejsek, B. T. Jónsson, and L. Amsaleg, "Nv-tree: nearest neighbors at the billion scale," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ser. ICMR '11. New York, NY, USA: ACM, 2011, pp. 54:1–54:8.
[4] H. Jegou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316–336, 2010.
[5] J. Shawe-Taylor and N. Cristianini, *Kernel methods for Pattern Analysis*. Cambridge University Press, ISBN 0-521-81397-2, 2004.
[6] S. Lyu, "Mercer kernels for object recognition with local features," in *IEEE International Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 223–229.
[7] P.-H. Gosselin, M. Cord, and S. Philipp-Foliguet, "Kernel on bags for multi-object database retrieval," in *ACM International Conference on Image and Video Retrieval*, Amsterdam, The Netherlands, July 2007, pp. 226–231.
[8] F. Precioso, M. Cord, D. Gorisse, and N. Thome, "Efficient bag-of-feature kernel representation for image similarity search," in *International Conference on Image Processing*, 2011, pp. 109–112.
[9] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1470–1477.
[10] S. Avila, N. Thome, M. Cord, E. Valle, and A. De A. Araújo, "BOSSA: extended BoW formalism for image classification," in *International Conference on Image Processing*, Brussels, Belgique, Sep. 2011, cAPES, CNPq, FAPESP (Brazil), COFECUB (France).
[11] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *ECCV 2008, PART III. LNCS*. Springer, 2008, pp. 696–709.
[12] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *in IEEE Conference on Computer Vision and Pattern Recognition(CVPR*, 2009.
[13] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.
[14] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision*, 2010, pp. 143–156.
[15] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, qUAERO.
[16] D. Picard and P.-H. Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *International Conference on Image Processing*, Brussels, Belgique, September 2011.

[17] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*. Springer, 2008, pp. 304–317.
[18] H. Jégou, F. Perronnin, M. Douze, C. Schmid *et al.*, "Aggregating local image descriptors into compact codes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1704–1716, 2012.

VI. Biography

Romain Negrel is currently a second year PhD student in multimedia retrieval at the ETIS Lab at the University of Cergy-Pontoise (France), working under the joint supervision of Philippe Henri Gosselin and David Picard. His thesis title is "Représentations Optimales pour la Recherche dans les Bases d'Images Patrimoniales."

David Picard received the M.Sc. in Electrical Engineering in 2005 and the Ph.D. in image and signal processing in 2008. He joined the ETIS laboratory at the ENSEA (France) in 2010 as an associate professor within the MIDI team. His research interests include computer vision and machine learning for visual information retrieval, with focus on kernel methods for multimedia indexing.

Philippe-Henri Gosselin received the PhD degree in image and signal processing in 2005, and joined the MIDI Team in the ETIS Lab as an assistant professor in 2007. His research focuses on machine learning for online multimedia retrieval. This includes studies on kernel functions on histograms, bags and graphs of features, but also weakly supervised semantic learning methods.

A Unified framework for local visual descriptors evaluation

Olivier Kihl, David Picard, Philippe-Henri Gosselin

A Unified framework for local visual descriptors

Olivier Kihl[a,*], David Picard[a], Philippe-Henri Gosselin[a,b]

[a] *ETIS/ ENSEA - Université Cergy-Pontoise, CNRS, UMR 8051*
6 avenue du Ponceau, CS 20707 CERGY, F 95014 Cergy-Pontoise Cedex France
Telephone:+33 1 30 73 66 10 and Fax:+33 1 30 73 66 27
[b] *INRIA Rennes Bretagne Atlantique*
Campus de Beaulieu, 35042 Rennes Cedex France

Abstract

Local descriptors are the ground layer of recognition feature based systems for still images and video. We propose a new framework to explain local descriptors. This framework is based on the descriptors decomposition in three levels: primitive extraction, primitive coding and code aggregation. With this framework, we are able to explain most of the popular descriptors in the literature such as HOG, HOF, SURF. We propose two new projection methods based on approximation with oscillating functions basis (sinus and Legendre polynomials). Using our framework, we are able to extend usual descriptors by changing the code aggregation or adding new primitive coding method. The experiments are carried out on images (VOC 2007) and videos datasets (KTH, Hollywood2 and UCF11), and achieve equal or better performances than the literature.

Keywords: Image Processing and Computer Vision, Vision and Scene Understanding, Video analysis, Image/video retrieval retrieval, Object recognition, Feature representation

1. Introduction

Most multimedia retrieval systems compare multimedia documents (image or video) thanks to three main stages: extract a set of local visual descriptors from the multimedia document; learn a mapping of the set of descriptors into a single vector to obtain a signature; compute the similarity between signatures. In this paper, we focus on the computation of visual descriptors. The main goal of local visual descriptors is to extract local properties of the signal. These properties are chosen to represent discriminative characteristic atoms of images or videos. Since local descriptors are the ground layer of recognition systems, efficient descriptors are necessary to achieve good accuracies. Such descriptors have become essential tools in still image classification [1, 2] and video action classification [3, 4, 5].

The main contribution of this paper is a unified framework for visual descriptors that includes all the usual descriptors from the literature such as SIFT (Scale-invariant feature transform) [6], SURF (Speeded Up Robust Features) [7], HOG (Histogram of Oriented Gradient) [8], HOF (Histogram of Oriented Flow) and MBH (Motion Boundary Histogram) [9]. This framework is based on the decomposition of the descriptor in three levels: primitive extraction, primitive coding and code aggregation. Each popular descriptor is composed by a given primitive, a given coding and a given aggregation. Moreover, our framework allows to extend every descriptor by changing one or more of the three levels, e.g. changing the primitive level of HOG (gradient) by the motion primitive produces the HOF descriptor. The second contribution of this paper is the proposal of new coding and aggregation steps, the later being based on oscillating functions (Sinus and Polynomials), leading to new descriptors. Finally, we propose an exploration of the possible combinations of these primitives, coding and aggregation methods provided by the framework, that allows us to design more efficient and complementary descriptors.

The paper is organized as follows. In section 2, we present the most popular descriptors in the literature, for still images and for human action videos. We also present the most common approaches to compute the signature of a multimedia document from a set of descriptors. Then, in section 3, we present our framework, explain the most popular descriptors, and extend them by modifying some of these three steps. Finally, in section 4, we carry out experiments on one still image classification dataset and three action classification datasets for several descriptors and combinations of them.

2. Related work

In this section, we present the most popular descriptors in the literature, first for still image and then for human action video. We also present the most common approaches to compute the signature of a multimedia document from a set of descriptors.

*Corresponding author
Email addresses: olivier.kihl@ensea.fr (Olivier Kihl), picard@ensea.fr (David Picard), philippe-henri.gosselin@ensea.fr (Philippe-Henri Gosselin)

2.1. Still image descriptors

In the past ten years, several descriptors have been proposed for key-points matching and successfully used for still image classification. The most commonly used are SIFT [6], SURF [7] and Histogram of oriented gradient (HOG) [9]. SIFT and SURF are both interest points detector and local image descriptor. In this paper, we only consider the descriptors. SIFT and HOG descriptors rely on a histogram of orientation of gradient. Locally, the orientation of the gradient is quantized in o orientations (typically 8). For a given spatial window, a HOG (or a SIFT) descriptor is computed by decomposing the window with a grid of $N \times N$ cells. Each cell contains the histogram of orientations of the gradient. The descriptor is obtained by the concatenation of the $N \times N$ histograms.

The SURF [7] descriptor has been developed as a faster alternative to SIFT. In the case of SURF, descriptors of each cells are computed using weighted sum of responses to 2D Haar-wavelets along horizontal axis (dx) and vertical axis (dy), the absolute value of dx and the absolute value of dy.

More recently, new descriptors have been proposed with the aim to decrease the computation time without loss of performance. The GLOH [10] is an extension of the SIFT, in which the rectangular grid is replaced by a polar grid. The authors propose to use 3 bins in radial direction and 8 in angular direction. The gradient orientation is quantized in 16 bins inside each cell. To reduce the dimension of the descriptor, a principle components analysis (PCA), computed on several GLOH, is applied. Similarly, Daisy [11] is a SIFT like descriptor designed to be faster to compute in the case of dense matching extraction. The sum in histogram cells are replaced by computing the convolution of the orientation maps with Gaussian kernels. Moreover, the sampling positions of the descriptor are not aligned with a rectangular grid, but in concentric circles at several distances to the descriptor center. For a given distance, the sample points are associated to a particular Gaussian kernel size, increasing with the distance to the center.

2.2. Action descriptors

In the early work on action recognition, silhouette based descriptors, also called motion appearance models were used. These descriptors are computed from the evolution of a silhouette obtained by background subtraction methods or by taking the difference of frames (DOF). From a sequence of binary images, Bobick and Davis [12] propose descriptors called "Motion Energy Image" (MEI) representative of the energy of movement and "Motion History Image" (MHI) providing information about the chronology of motion. These two descriptors are modeled by seven Hu moments. In [13] Kellokumpu et al. use histograms of "Local Binary Patterns" (LBP) [14] to model the MHI and MEI images. In [15], they propose an extension of the LBP directly applied on the image pixels with successful results. Wang and Suter [16] use two other descriptors, namely the "Average Motion Energy" (AME) and the "Mean Motion Shape" (MMS). The AME is a descriptor close to the MHI representing the average image of silhouettes. The MMS is defined from boundary points of the silhouette in complex coordinates with the origin placed at the centroid of the 2D shape. As time is an important information in video, Gorelick et al. [17, 18] study the silhouettes as space-time volumes. Space-time volumes are modeled with Poisson equations. From these, they extract seven spatio-temporal characteristic components.

The main drawback of all these methods is the computation of silhouettes. Indeed, this computation is not very robust, making these methods only relevant in controlled environments such as the Weizmann dataset [17] and the KTH dataset [5]. However, they tend to fail on more realistic data-sets such as UCF11 [19] or Hollywood2 [4] datasets.

Assuming that action recognition is closely linked to the notion of movement, many authors have proposed descriptors based on the modeling of optical flow. The optical flow represents the displacement of pixels from two consecutive frames. The result can be represented by vector field with two components. Here, $\mathcal{U}$ denotes the horizontal component of motion and $\mathcal{V}$ the vertical component. Early works with respect to this approach were proposed by Polana and Nelson [20]. The vector field is first decomposed according to a spatial grid. Then, in each cell of the grid, the magnitude of motion is accumulated. This method can only process periodic actions such as running or walking.

Efros et al. [21] propose a descriptor computed on a figure-centric spatio-temporal volume for each person in a video. The vector field representing the motion between two consecutive frames of the volume is computed with the Lucas and Kanade optical flow algorithm [22]. The two components $\mathcal{U}$ and $\mathcal{V}$ of the vector field are decomposed with a half-wave rectification technique. The resulting four components are blurred using a Gaussian filter and normalized. They are directly used as a descriptor. The obtained descriptors are compared using the normalized correlation measure. This descriptor is used and/or extended by several authors in [23, 24].

Tran et al. propose the motion context descriptor in [25]. It is also a figure-centric descriptor based on the silhouette extraction. They use the vector field and the binary silhouette as three components. The components of the field are blurred with a median filter. Then, the three components are subdivided with a grid of 2×2 cells. Each cell is decomposed in 18 radial bins, each covering 20 degrees. Inside the radial bins, the sum of each component is computed. This provides, for each component, 4 histograms composed with 18 bins. The concatenation of these histograms provides a 216-dimensional vector which is the movement pattern of a given field. From this pattern, the "Motion Context" is created. It is composed of the 216-dimensional vector of the current frame plus the first 10 vectors of the PCA models of the 5 previous frames, the first 50 vectors of the PCA models of 5 current frames and finally the first 10 vectors PCA models of 5 next frames.

Ali and Shah [26] begin by computing many kinematic features on the field and then compute kinematic modes with a spatio-temporal principal component analysis.

Finally, the most successful descriptors developed in recent years are extensions to video of still image descriptors. The most commonly used are the Histogram of Oriented Flow (HOF) [9] and the Motion Boundary Histogram (MBH) [9].

HOF is the same as HOG but is applied to optical flow instead of gradient. The MBH models the spatial derivatives of each component of the optical flow vector field with a HOG.

In this context, several extension of still image descriptors have been proposed. The cuboid [27] is a space-time descriptor, represented by a space-time volume. For a given volume, the gradient is computed on the three directions and the descriptor are the flattening of the gradient in a vector. Consistent with this, Klaser et al. [28] extend HOG to 3DHOG. A 3-dimensional extention of SIFT is proposed in [29]. ESURF [30] is an extention of SURF with 3D Haar-wavelets.

Descriptors based on a polynomial approach for modeling global optical flow are proposed in [31] and [32]. From this preliminary works, a local descriptor for motion named Series of Polynomial approximation of Flow (SoPAF) is proposed in [33]. The descriptor is based on two local modeling, a spatial model and a temporal model. The spatial model is computed by the projection of the optical flow onto bivariate orthogonal polynomials. Then, the time evolution of spatial coefficients is modelled with a one dimension polynomial basis.

Recently, Wang et al. [3] propose to model these usual descriptors along dense trajectories. The time evolution of trajectories, HOG, HOF and MBH is modelled using a space time grid following pixels trajectories. The use of dense trajectories for descriptor extraction tends to increase the performances of popular descriptors (HOG, HOF and MBH).

2.3. Signatures

Once a set of descriptors is obtained from the video, a popular way of comparing images (or videos) is to map the set of descriptors into a single vector and then to measure the similarity between the obtained vectors (for example in [34], [35] and [3]). The most common method for such embeddings is inspired by the text retrieval community and is called the "Bag of Words" (BoW) approach [36]. It consists in computing a dictionary of descriptor prototypes (usually by clustering a large number of descriptors) and then computing the histogram of occurrences of these prototypes (called "Visual Words") within the set.

In still images classification, these approaches have been formalized in [37] by a decomposition of the mapping into two steps. The first step, namely the "coding step", consists in mapping each descriptor into a codeword using the aforementioned dictionary. The second step is to aggregate the codewords into a single vector and is called the "pooling step". Structural constraints such as sparsity [38] or locality [37] can be added to the coding process to ensure most of the information is retained during the pooling step. Common pooling processes include averaging the codewords or retaining the entry-wise maximum among the codewords (max pooling). Extensions of the BoW model have been recently proposed to include more precise statistical information. In [39], the authors propose to model the distribution of distances of descriptors to the clusters centers. In the "coding/pooling" framework, each descriptor is coded by 1 in the bin corresponding to its distance to the cluster's center to which it belongs, and 0 otherwise. The polling is simply the averaging over all codewords.

In [40], the authors proposed a coding process where the deviation between the mean of the descriptors of the set and the center of the cluster to which they belong to is computed. The whole mapping process can be seen as the deviation between a universal model i.e. the dictionary) and a local realization (i.e. the set of descriptors). Using this model deviation approach, higher order statistics have been proposed, like "super-vectors" in [41], "Fisher Vectors" in [42] or "VLAT" in [43, 44]. Fisher Vectors are known to achieve state of the art performances in image classification challenges [2].

To compare the performances of descriptors, in this paper, we consider a compressed version of VLAT which is known to achieve near state of the art performances in still images classification with very large sets of descriptors [45]. In our case, the dense sampling both in spatial and temporal directions leads to highly populated sets, which is consistent with the statistics computed in VLAT signatures. Given a clustering of the descriptors space with C clusters computed on some training set, the first and second order moments μ_c and τ_c are computed for each cluster c:

$$\mu_c = \frac{1}{|c|}\sum_i\sum_r \nu_{rci} \tag{1}$$

$$\tau_c = \frac{1}{|c|}\sum_i\sum_r (\nu_{rci}-\mu_c)(\nu_{rci}-\mu_c)^T \tag{2}$$

with ν_{rci} the descriptor r of the video i which is in cluster c, and $|c|$ being the number of descriptors ν_{rci} of video i in cluster c, for all videos in the training set. To allow a dimension reduction of the signature, the eigen decomposition of the covariance matrix τ_c for each cluster c is then performed:

$$\tau_c = \mathbf{V}_c\mathbf{D}_c\mathbf{V}_c^\top \tag{3}$$

Using this decomposition, descriptors are projected on the subspace generated by the eigenvectors V_c. The compressed VLAT signature $\tau_{i,c}$ of video i is computed for each cluster c with the following equation:

$$\tau_{i,c} = \sum_r (\mathbf{V}_c(\nu_{rci}-\mu_c))(\mathbf{V}_c(\nu_{rci}-\mu_c))^\top - \mathbf{D}_c \tag{4}$$

$\tau_{i,c}$ are then flattened into vectors $\mathbf{v}_{i,c}$. The complete VLAT signature $\mathbf{x}_i$ of video i is obtained by concatenation of $\mathbf{v}_{i,c}$ for all clusters c:

$$\mathbf{v}_i = (v_{i,1}\dots v_{i,C}) \tag{5}$$

It is advisable to perform a normalization step for best performance.

$$\forall j,\quad \mathbf{v}'_i[j] = sign(\mathbf{v}_i[j])|\mathbf{v}_i[j]|^\alpha, \tag{6}$$

$$\mathbf{x}_i = \frac{\mathbf{v}'_i}{\|\mathbf{v}'_i\|} \tag{7}$$

With $\alpha = 0.5$ typically. The size of the compacted VLAT signature depends on the number d_c of eigenvectors retained in each cluster, and is equal to $\sum_c \frac{d_c(d_c+1)}{2}$ (thanks to the matrices $\tau_{i,c}$ being symmetric, only half of the coefficients are kept).

3. Primitive/Coding/Aggregation framework

In this section, we present the main contribution of this paper. We propose a framework providing a formal description of the steps needed to design local visual descriptors. Our framework splits descriptors extraction in three levels: primitive extraction, primitive coding and code aggregation.

3.1. Primitive extraction

At the primitive level, we extract a specific type of low-level information from an image or a video. Such primitives include the gradient (HOG), the responses to 2D Haar-wavelets (SURF), the motion flow (HOF), or the gradient of motion flow (MBH). In Fig. 1, we show three examples of primitive used in literature, the gradient, the motion flow and the gradient of the motion flow. The objective is to extract local properties of the signal. Generally, it relies on a high frequency filtering, linear for gradient or non-linear in the case of motion (optical flow), filters banks such as Haar (SURF), easy extension of popular filters [46], or non-linear operators. The primitive extraction induces a choice in relevant information and introduces data loss.

3.2. Primitive coding

The primitive coding corresponds to a non-linear mapping of the primitive to a higher dimensional space. The objective is to improve the representation by grouping together the primitive properties that are similar.

In the literature, the most popular primitive coding is the quantization of local vector field orientations [6, 8]. The quantization is usually performed on 8 bins. Let $G_x(\mathbf{x})$,$G_y(\mathbf{x})$ be the horizontal and vertical derivative of an image at position $\mathbf{x}$, the principal orientation bin is computed by:

$$o(\mathbf{x}) = \lfloor \frac{(atan2(G_y(\mathbf{x}), G_x(\mathbf{x})) \mod 2\pi) \times 4}{\pi} \rfloor \quad (8)$$

In order to limit the effect of floor on coding, the distance to the next orientation bin is computed by

$$r(\mathbf{x}) = o(\mathbf{x}) - \left\lfloor \frac{(atan2(G_y(\mathbf{x}), G_x(\mathbf{x})) \bmod 2\pi) \times 4}{\pi} \right\rfloor \quad (9)$$

The value associated to the bin $o(\mathbf{x})$ and $o(\mathbf{x}) + 1$ are

$$O(\mathbf{x}, o(\mathbf{x})) = \rho(\mathbf{x}) \times (1 - r(\mathbf{x})) \quad (10)$$

$$O(\mathbf{x}, (o(\mathbf{x}) + 1) \bmod 8) = \rho(\mathbf{x}) \times r(\mathbf{x}) \quad (11)$$

with $\rho(\mathbf{x})$ the magnitude of horizontal and vertical derivative ($\rho(\mathbf{x}) = \sqrt{G_x(\mathbf{x})^2 + G_y(\mathbf{x})^2}$). This primitive coding do not introduce any loss of information or redundancy.

Another primitive coding is proposed in SURF [7]. Here, we call it "absolute coding". In the SURF descriptor, it is applied to the gradient primitive. This is a four dimension code defined as:

$$\mathcal{A}(\mathbf{x}, 0) = G_x(\mathbf{x}) \quad (12)$$

$$\mathcal{A}(\mathbf{x}, 1) = G_y(\mathbf{x}) \quad (13)$$

$$\mathcal{A}(\mathbf{x}, 2) = |G_x(\mathbf{x})| \quad (14)$$

$$\mathcal{A}(\mathbf{x}, 3) = |G_y(\mathbf{x})| \quad (15)$$

This primitive coding introduces redundancy. However, it produces lower dimensions code than orientation coding.

In the context of action recognition and classification, the rectified coding proposed by Efros et al. . [21] has been used by several authors. They decompose the horizontal ($\mathcal{U}$) and vertical ($\mathcal{V}$) components of a vector field (usually obtained by optical flow approaches) with a technique of half-wave rectification:

$$\mathcal{R}(\mathbf{x}, 0) = \begin{cases} \mathcal{U}(\mathbf{x}) & \text{if } \mathcal{U}(\mathbf{x}) > 0 \\ 0 & \text{else} \end{cases} \quad (16)$$

$$\mathcal{R}(\mathbf{x}, 1) = \begin{cases} |\mathcal{U}(\mathbf{x})| & \text{if } \mathcal{U}(\mathbf{x}) < 0 \\ 0 & \text{else} \end{cases} \quad (17)$$

$$\mathcal{R}(\mathbf{x}, 2) = \begin{cases} \mathcal{V}(\mathbf{x}) & \text{if } \mathcal{V}(\mathbf{x}) > 0 \\ 0 & \text{else} \end{cases} \quad (18)$$

$$\mathcal{R}(\mathbf{x}, 3) = \begin{cases} |\mathcal{V}(\mathbf{x})| & \text{if } \mathcal{V}(\mathbf{x}) < 0 \\ 0 & \text{else} \end{cases} \quad (19)$$

Orientation coding, absolute coding and rectified coding are the most used in literature.

We also propose a new primitive coding called double rectified coding. This coding corresponds to the 4 components of the rectified coding and the 4 components of the absolute coding. Examples of these primitive coding are shown in Fig. 2.

3.3. Code Aggregation

Finally, the code aggregation is used to model the encoded primitives. The objective of aggregation is to improve the robustness to deformation by allowing inexact matching between deformed image or video patches. Most descriptors from the literature (HOG, HOF, MBH, SURF) use accumulation of each primitive coding (typically with a simple sum). In order to improve robustness, the accumulation is done on the cell of a grid of $N \times N$ cells. In the case of video, the grid could be extended in $N \times N \times T$ cells with T the number of cell bins in time direction. The spatial window could be pondered by a Gaussian to give more importance to the cells which are close to the center, like in SIFT. We show a 4×4 cell aggregation in Fig. 3a.

The regular grid can replaced with concentric circles arranged in a polar manner, as it is proposed in DAISY [11]. The final pattern resembles a flower, and is shown in Fig. 3b. In the following, we name this code aggregation "Flower". The flower aggregation is defined by three parameters R, Q and T. The radius R defines the distance from the center pixel to the outer most grid point. The quantization of the radius Q defines the number of convolved primitives layer associated to different size of Gaussian ($Q = 3$ in Fig. 3b). The parameter T defines the angular quantization of the pattern at each layer ($T = 8$ in Fig. 3b).

The aggregation proposed in [31] is based on the projection of primitive on a two dimensional orthogonal polynomial basis. The family of polynomial functions with two real variables is defined as follows:

$$P_{K,L}(x_1, x_2) = \sum_{k=0}^{K} \sum_{l=0}^{L} a_{k,l}\, x_1^k\, x_2^l \quad (20)$$

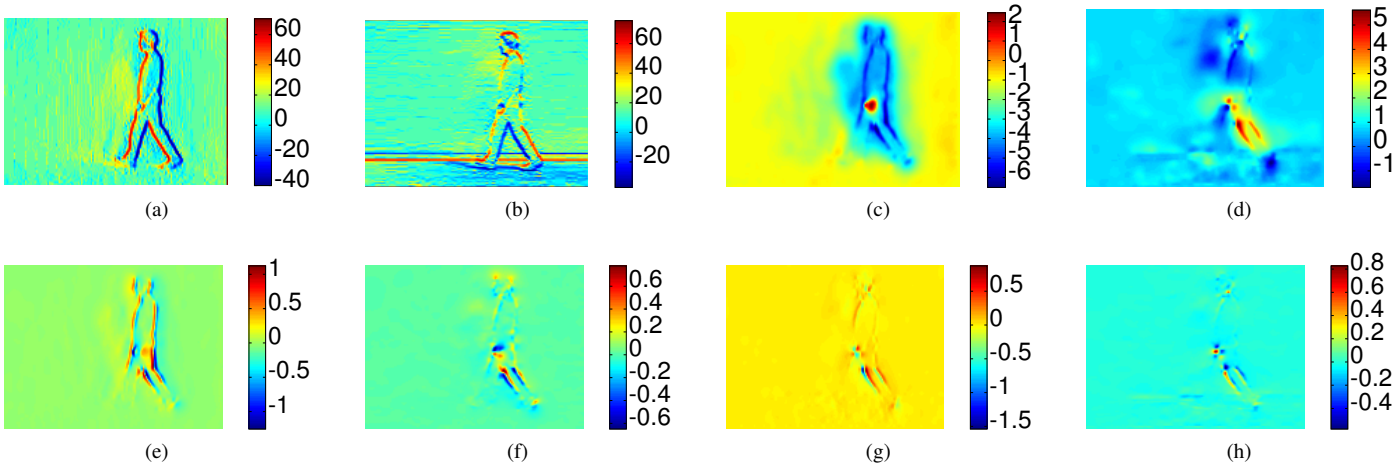


Figure 1: Example of primitive ; (a) Horizontal gradient ; (b) Vertical gradient ; (c) horizontal motion flow ; (d) Vertical motion flow ; (e) Horizontal gradient of horizontal motion flow ; (f) Vertical gradient of horizontal motion flow ; (g) Horizontal gradient of vertical motion flow ; (h) Vertical gradient of vertical motion flow

where $K \in \mathbb{N}^+$ and $L \in \mathbb{N}^+$ are respectively the maximum degree of the variables (x_1, x_2) and $\{a_{k,l}\}_{k\in\{0..K\},l\in\{0..L\}} \in \mathbb{R}^{(K+1)\times(L+1)}$ are the polynomial coefficients. The global degree of the polynomial is $D = K + L$.

Let $\mathcal{B} = \{P_{k,l}\}_{k\in\{0..K\},l\in\{0..L\}}$ be an orthogonal basis of polynomials. A basis of degree D is composed by n polynomials with $n = (D+1)(D+2)/2$ as follows:

$$\begin{aligned}\mathbb{B} = \quad & \{B_{0,0}, B_{0,1}, \cdots, B_{0,L}, B_{1,0}, \cdots \\ & \cdots, B_{1,L-1}, \cdots, B_{K-1,0}, B_{K-1,1}, B_{K,0}\}\end{aligned} \tag{21}$$

An orthogonal basis can be created using the following three terms recurrence:

$$\begin{cases} B_{-1,l}(\mathbf{x})=0 \\ B_{k,-1}(\mathbf{x})=0 \\ B_{0,0}(\mathbf{x})=1 \\ B_{k+1,l}(\mathbf{x})=(x_1-\lambda_{k+1,l})B_{k,l}(\mathbf{x})-\mu_{k+1,1}B_{k-1,l}(\mathbf{x}) \\ B_{k,l+1}(\mathbf{x})=(x_2-\lambda_{k,l+1})B_{k,l}(\mathbf{x})-\mu_{k,l+1}B_{k,l-1}(\mathbf{x}) \end{cases} \tag{22}$$

where $\mathbf{x} = (x_1, x_2)$ and the coefficients $\lambda_{k,l}$ and $\mu_{k,l}$ are given by

$$\begin{aligned} \lambda_{k+1,l} &= \frac{\langle x_1 B_{k,l}(\mathbf{x}) | B_{k,l}(\mathbf{x})\rangle}{\|B_{k,l}(\mathbf{x})\|^2} & \lambda_{k,l+1} &= \frac{\langle x_2 B_{k,l}(\mathbf{x}) | B_{k,l}(\mathbf{x})\rangle}{\|B_{k,l}(\mathbf{x})\|^2} \\ \mu_{k+1,l} &= \frac{\langle B_{k,l}(\mathbf{x}) | B_{k,l}(\mathbf{x})\rangle}{\|B_{k-1,l}(\mathbf{x})\|^2} & \mu_{k,l+1} &= \frac{\langle B_{k,l}(\mathbf{x}) | B_{k,l}(\mathbf{x})\rangle}{\|B_{k,l-1}(\mathbf{x})\|^2} \end{aligned} \tag{23}$$

and $\langle \cdot \mid \cdot \rangle$ is the usual inner product for polynomial functions:

$$\langle B_1 \mid B_2 \rangle = \iint_\Omega B_1(\mathbf{x}) B_2(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} \tag{24}$$

with w the weighting function that determines the polynomial family and Ω the spatial domain covered by the window $W(i, j, t)$. Legendre polynomials ($w(\mathbf{x}) = 1, \forall \mathbf{x}$) are usually used.

Using this basis, the approximation of a decomposed primitive component $\mathcal{P}$ is:

$$\tilde{\mathcal{P}} = \sum_{k=0}^{D} \sum_{l=0}^{D-k} \tilde{u}_{k,l} \frac{B_{k,l}(\mathbf{x})}{\|B_{k,l}(\mathbf{x})\|} \tag{25}$$

The polynomial coefficients $\tilde{u}_{k,l}$ are given by the projection of component $\mathcal{U}$ onto normalized $\mathcal{B}$ elements:

$$\tilde{p}_{k,l} = \frac{\langle \mathcal{P} \mid B_{k,l}(\mathbf{x})\rangle}{\|B_{k,l}(\mathbf{x})\|} \tag{26}$$

We show the polynomials associated to a 4 degree basis in Fig. 3c. The polynomials are defined in a spatial domain of 32×32 pixels. In the case of video classification, space-time aggregation is considered. Kihl et al. . propose [31] to model spatial polynomial coefficients with a d degree temporal basis of Legendre polynomial defined by

$$\begin{cases} B_{-1}(t) = 0 \\ B_0(t) = 1 \\ T_n(t) = (t - \langle tB_{n-1}(t) | B_{n-1}(t)\rangle)B_{n-1}(t) - B_{n-2}(t) \\ B_n(t) = \dfrac{T_n(t)}{|T_n|} \end{cases} \tag{27}$$

Using this basis of degree d, the approximation of $\mathbf{P}_{k,l}(i, j, t)$ is:

$$\tilde{\mathbf{p}}_{k,l}(i, j, t) = \sum_{n=0}^{d} \tilde{p}_{k,l,n}(i, j, t) \frac{B_n(t)}{\|B_n(t)\|} \tag{28}$$

The model has $d + 1$ coefficients $\tilde{\mathbf{p}}_{k,l}(i, j, t)$ given by

$$\tilde{p}_{k,l,n}(i, j, t) = \frac{\langle \mathbf{p}_{k,l}(i, j, t) \mid B_n(t)\rangle}{\|B_n(t)\|} \tag{29}$$

The time evolution of a given coefficient $\tilde{p}_{k,l}(i, j)$ is given by the vector $\mathbf{m}_{l,k}(i, j, t_0)$ as defined in equation (30)

$$\begin{aligned}\mathbf{m}_{l,k}(i, j, t_0) = [&\tilde{p}_{k,l,0}(i, j, t_0), \tilde{p}_{k,l,1}(i, j, t_0), \\ &\ldots, \tilde{p}_{k,l,d}(i, j, t_0)]\end{aligned} \tag{30}$$

Finally, the descriptor is the concatenation of all the $\mathbf{m}_{l,k}(i, j, t_0)$ vectors for each coded primitive. In this paper, we also propose an easy extension of this aggregation using a Sine basis, in place of the Legendre polynomials.

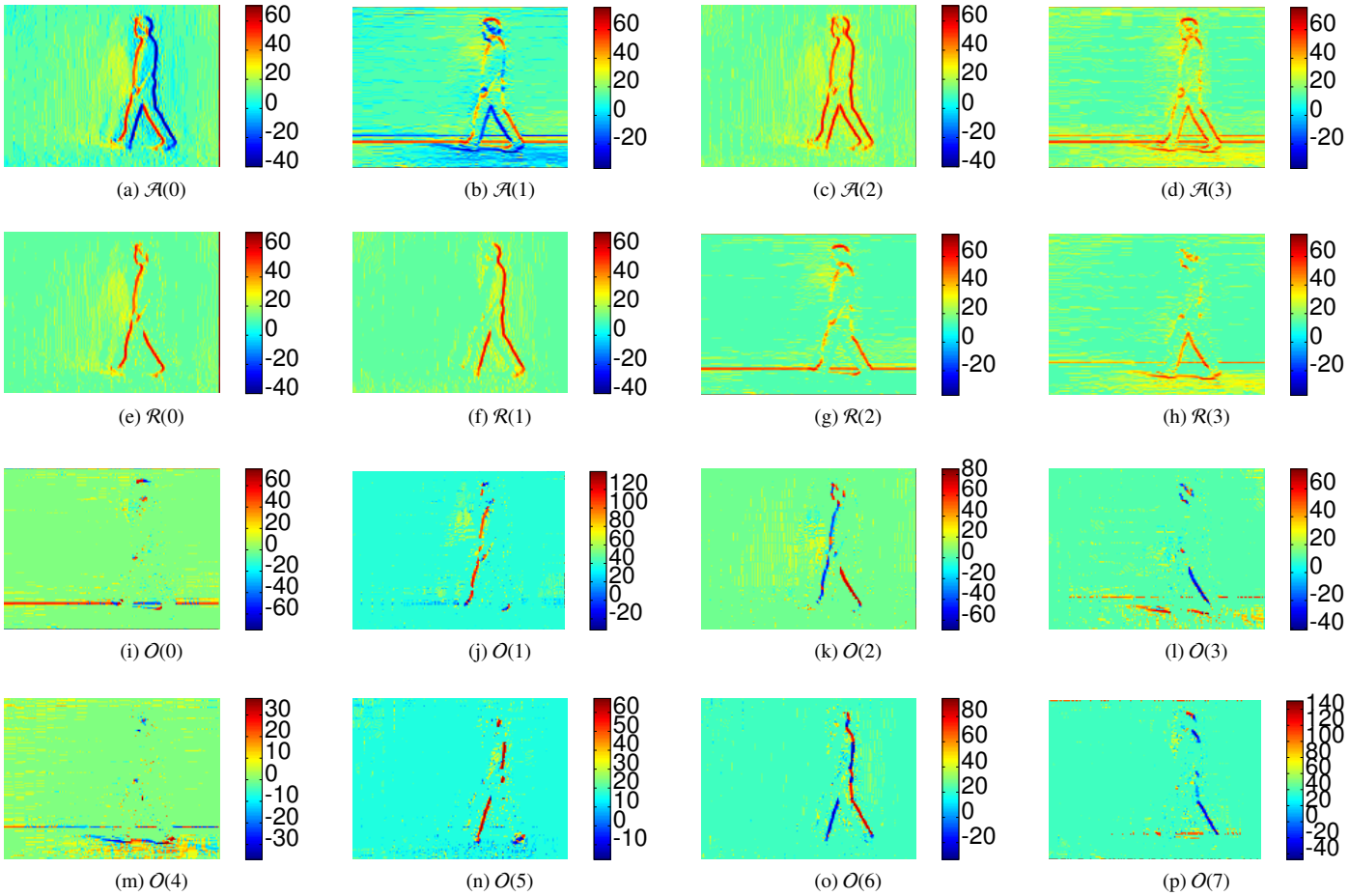


Figure 2: Example of coding ; on the first line: Absolute coding of the gradient primitive ($\mathcal{A}(0)$, $\mathcal{A}(1)$, $\mathcal{A}(2)$, $\mathcal{A}(3)$; on the second line: Rectified coding of the gradient primitive ($\mathcal{R}(0)$, $\mathcal{R}(1)$, $\mathcal{R}(2)$, $\mathcal{R}(3)$) ; on the third and fourth lines: Orientation coding of the gradient primitive ($\mathcal{O}(0)$, $\mathcal{O}(1)$, $\mathcal{O}(2)$, $\mathcal{O}(3)$, $\mathcal{O}(4)$, $\mathcal{O}(5)$, $\mathcal{O}(6)$, $\mathcal{O}(7)$) ;

Primitive	Coding	Aggregation
gradient	raw	Regular cells
motion	rectified	Flower
Haar	absolute	polynomial basis
motion gradient	orientation	sine basis
⋮	⋮	⋮

Table 1: A new framework for local descriptors

Name	Primitive	Coding	Aggregation
HOG	gradient	orientations	Regular cells
Daisy	gradient	orientations	Flower
HOF	motion	orientations	Regular cells
MBH	motion gradient	orientations	Regular cells
SURF	Haar	abs	cells
Efros	motion	rectified	Regular cells
SoPAF	motion	raw	polynomial basis

Table 2: Rewriting of the usual descriptors ; raw means the vector field is represented by the horizontal and vertical components

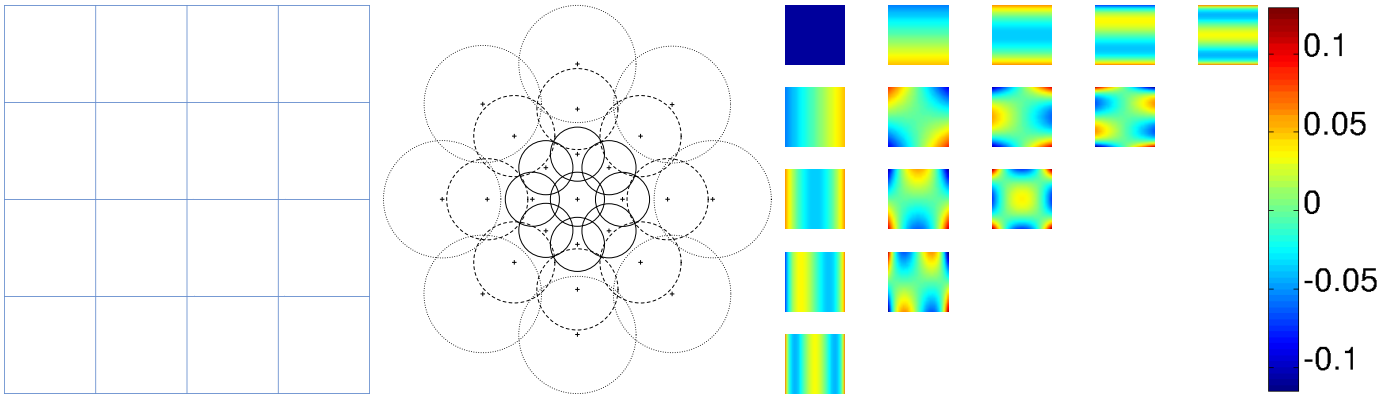
3.4. A unified framework for descriptors

In Table 1, we summarize the different primitives, coding and aggregations currently used for classification. According to specific combinations of primitive, coding and aggregation, we can explain most of the usual descriptors. In Table 2, we explain the usual descriptors of the literature with our framework.

Each new Primitive, Coding or Aggregation defines a new family of descriptors and each new combination of Primitive-Coding-Aggregation defines a new descriptor. Since different primitives correspond to different properties of the signal, we argue that adapted coding and aggregation schemes have to be used to produce efficient descriptors. Indeed, our framework allows to explore and evaluate the possible combinations so as to find the best descriptors.

4. Experiments

In the experiments, we compare several combination of primitives, coding and aggregations provided by our framework in order to evaluate still image descriptors and action descriptors.



(a) (b) (c)

Figure 3: Examples of aggregation ; (a) 4×4 cells aggregation ; (b) Flower aggregation with $Q = 3$ and $T = 8$; (c) Representaion of 4 degree basis spatial polynomials aggregation

As dense sampling outperforms key-point extraction [1, 3] for categories recognition, we use dense sampling in all our experiments. We carry out experiments on an image dataset (VOC2007) and three well known human action recognition datasets (KTH dataset [5], Hollywood2 Human Actions dataset [4] and UCF11 [19]).

For the experiments, we obtain signatures from our descriptors by using the VLAT indexing method [47] as explained in section 2.3.

4.1. Still image classification

We first present results on still image categorization. The gradient is the only primitive considered. The gradient is extracted with the simple one order approximation difference method, at a single resolution.

Pascal VOC 2007 dataset

The PASCAL-VOC 2007 dataset [1] consists in about 10,000 images and 20 categories, and is divided into 3 parts: "train", "val" and "test". We use linear SVM classifier trained on "train" + "val" sets and tested on the "test" set.

We use four primitive coding: absolute, rectified, double rectified and orientation. These coding are combinatorially associated to the following three code aggregations: regular cells, flower and polynomials basis. For the regular grid aggregation, we use 4×4 cells. The cells are evaluated at four scales: 4×4 pixels, 6×6 pixels, 8×8 pixels and 10×10 pixels. For the flower aggregation, the parameter Q is set to 3 and the parameter T is set to 8. We consider the flower aggregation at four scales by setting radius R at 9 pixels, 12 pixels, 15 pixels and 18 pixels. For the polynomial aggregation, we set the basis degree to 4. The polynomial spatial domain is considered at four scales: 16×16 pixels, 24×24 pixels, 32×32 pixels and 40×40 pixels.

For the VLAT signature, the number of projectors in equation (3) is set to 70. We use a dictionary of 256 visual words.

Figure 4: Images from PASCAL Visual Object Classes Challenge 2007

4.1.1. Experimental results on still images

Results for each descriptor are shown in table 3. We remark orientation coding clearly outperforms the other primitive coding for all the code aggregations experimented on this dataset. The GoF, GoC and GoP (c.f. Table 3) provide the best results. We remark that the three features with highest mean average precision are GoC, GoF and GoP for each category of the VOC2007 dataset. Using a simple concatenation of the signatures, we obtain 64.2% of mean average precision.

This result is reported in table 4 and compared with the results from [34]. Note that our approach provides a global image signature which does not include any kind of spatial information like Spatial Pyramidal Matching (SPM) [48] or object detectors [49]. We compare our results to those of Sanchez et al. [34] which gives results without spatial information. In [34]

	mAP	aeroplane	bicycle	bird	boat	bottle	bus
Our method	**64.2**	**83.3**	**73.0**	**59.9**	**73.5**	33.2	**71.2**
SIFT + FV [34]	62.7	80.2	69.1	52.8	72.9	**37.6**	69.5
	car	cat	chair	cow	table	dog	horse
Our method	**84.2**	**65.7**	53.3	**49.5**	58.8	**52.3**	**83.0**
SIFT + FV [34]	81.8	61.8	**54.9**	47.2	**61.5**	50.5	79.1
	bike	person	plant	sheep	sofa	train	tv
Our method	**72.0**	**87.5**	37.2	**47.4**	55.4	**85.5**	58.0
SIFT + FV [34]	67.1	85.8	**37.6**	46.6	**57.0**	82.3	**59.0**

Table 4: Image classification results on Pascal VOC 2007 dataset

name	**Coding**	**Aggregation**	**mAP**	**usual name**
GaC	absolute	regular cells	58,2	SURF
GaF	absolute	flower	56,6	x
GaP	absolute	polynomial basis	57.6	x
GrC	rectified	regular cells	58.1	x
GrF	rectified	flower	57.2	x
GrP	rectified	polynomial basis	57.4	x
GoC	orientation	regular cells	63.2	HOG
GoF	orientation	flower	63.7	DAISY
GoP	orientation	polynomial basis	63.2	x
GdC	double	regular cells	58,2	x
GdF	double	flower	56,9	x
GdP	double	polynomial basis	57,8	x

Table 3: Classification results exprimed by mean average precision for combination of primitives, coding and aggregations on VOC2007 dataset

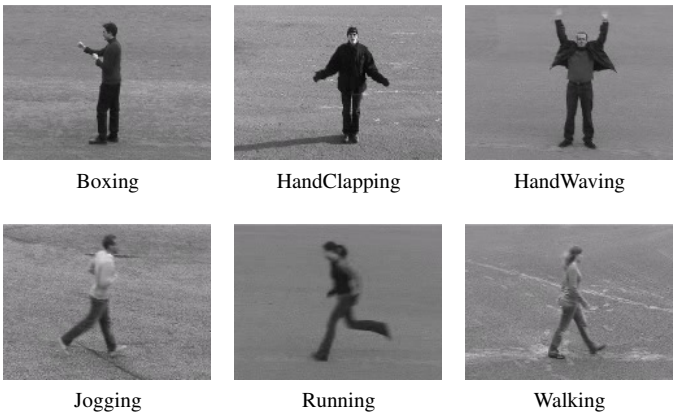


Figure 5: Example of videos from KTH

the SIFT descriptors are highly dense extracted at 7 resolutions and then aggregated with the Fisher Vector signature approach.

We show that our framework allows easy extension of HOG (GoC), for example by changing the codes aggregation from cell to polynomial. According to this new descriptor, we improve the categorization results obtained with only HOG descriptors. Moreover, our framework is compatible with adding spatial information like in [48], which should further improve the results.

4.2. Video actions recognition

In this section, we present results on video actions recognition. First, we evaluate our framework on the KTH [5] dataset. Then, we evaluate our method on two more challenging datasets of the literature, Hollywood2 [4]) and UCF11 [19] and compare our results to that of the literature. We compare three primitive extractions (gradient, motion and gradient of motion), three primitive coding (raw, rectified and orientations) and three code aggregations (cells, polynomials and sinus). We extract the gradient with the simple one order approximation difference method. For motion estimation, we use a Horn and Schunk optical flow algorithm [50] with 25 iteration and the regularization λ parameter is set to 0.1. We extract the primitives at 1 resolution for KTH, 7 resolutions for Hollywood2 and 5 resolutions for UCF11. The resolution factor is set to 0.8 . The resolutions are obtained by down sampling images, we do not use any up sampling in this work. We aggregate the extracted descriptors with the compressed VLAT signature approach as defined in the section 2.3.

4.2.1. Evaluation of our framework on KTH dataset

The KTH dataset [5] contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping (Fig. 5). These actions are performed by 25 different subjects in four scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, inside. For all experiments, we use the same experimental setup as in [5, 3], where the videos are divided into a training set (8 persons), a validation set (8 persons) and a test set (9 persons). The best hyperparameters are selected through cross-validation using the training and validation sets. The classification accuracy results are obtained on the test set.

We experiment several descriptors according to our framework for several spatial and time modeling. We present on Tables 5, 6 and 7 the main results obtained for each primitive extraction on KTH dataset. We present in Table 5 the results associated with the gradient primitive. As for still image experiments, the orientation coding clearly outperforms the other primitive coding for the three code aggregations. When the orientation coding is associated with the cell aggregation, it produces the best results for the gradient primitive extraction. The best results are obtained for code aggregations with the lower level of modeling along time axis for all code aggregations.

dim	coding	Gradient			SP	TP	Usual name
		Cell	Poly	Sinus			
32	raw	80.4			4	1	x
36	raw		81.0		2	2	x
40	raw	82.8			2	5	x
40	raw			86.8	1	4	x
64	raw		84.5		2	4	x
80	raw		83.1		3	3	x
48	rect			88.5	1	2	x
48	rect	84.8			2	3	x
60	rect		83.2		4	0	x
64	rect	86.5			4	1	x
64	rect			83.3	3	0	x
72	rect		84.5		2	2	x
80	rect	87.2			2	5	x
80	rect			88.5	1	4	x
128	rect		88.0		2	4	x
144	rect	88.5			3	4	x
96	ori	92.4			2	3	HOG
96	ori			91.4	1	2	x
120	ori		**92.6**		4	0	x
128	ori	**93.4**			4	1	HOG
128	ori			**93.3**	3	0	x

Table 5: Results for combination of gradient primitives, coding and aggregation on the KTH dataset ; dim means the dimension of the descriptor ; coding represent the code primitives (raw, rectified or orientation) ; SP means the number of spatial cells or the degree D of spatial polynomials or the spatial degree of the sinus basis ; TP means the number of temporal cells, or the degree d of temporal polynomials or the degree of sinus basis

dim	coding	Flow			SP	TP	Usual name
		Cell	Poly	Sinus			
32	raw	87.0			4	1	x
32	raw			85.1	3	0	x
36	raw		89.8		2	2	SoPAF
40	raw	89.6			2	5	x
40	raw			88.0	1	4	x
64	raw		90.4		2	4	SoPAF
80	raw		91.1		3	3	SoPAF
48	rect	90.7			2	3	x
48	rect			**91.3**	1	2	x
60	rect		90.7		4	0	x
64	rect	90.4			4	1	x
64	rect			87.7	3	0	x
72	rect		90.5		2	2	x
80	rect	91.4			2	5	x
80	rect			91.0	1	4	x
128	rect		**91.7**		2	4	x
144	rect	**92.0**			3	4	x
96	ori	89.2			2	3	HOF
96	ori			90.0	1	2	x
120	ori		90.6		4	0	x
128	ori	91.8			4	1	HOF
128	ori			87.8	3	0	x

Table 6: Results for combination of motion primitive, coding and aggregations on the KTH dataset ; The legend is the same as Table 5

dim	coding	Gradient of Motion			SP	TP	Usual name
		Cell	Poly	Sinus			
48	raw	90.0			2	3	x
48	raw			90.0	1	2	x
60	raw		90.3		4	0	x
64	raw	90.0			4	1	x
72	raw		90.6		2	2	x
80	raw	89.9			2	5	x
80	raw			89.4	1	4	x
128	raw		91.0		2	4	x
32	rect	92.2			2	1	x
32	rect			91.5	1	0	x
48	rect		93.1		2	0	x
96	rect	**94.2**			2	3	x
96	rect			**93.4**	1	2	x
120	rect		**93.7**		4	0	x
64	ori	92.5			2	1	MBH
64	ori			91.5	1	0	x
96	ori		93.6		2	0	x

Table 7: Results for combination of gradient of motion primitive, coding and aggregations on the KTH dataset ; The legend is the same as Table 5

We present in Table 6 the results associated with the motion primitive. In the case of motion primitive, the rectified coding allows to obtain good results for polynomial aggregation and sine aggregation. For the motion primitive, higher time modeling improves the results for a given spatial modeling. For instance, for the rectified coding and the polynomial aggregation with a spatial polynomial basis of degree 2, if the time polynomial basis is of degree 2 the classification accuracy is 90.5% and if the time polynomial basis is of degree 4, the accuracy is 91.7.

We present in Table 7 the results associated with the gradient of motion primitive. The best results, for each code aggregation, are obtained with rectified coding. It is interesting to note that we have only generated descriptors whose size does not exceed 144 dimensions. Note the motion of gradient primitive provides 4 components and the orientation coding decomposes each components in 8 orientation maps. So, the size of descriptors associating Motion of gradient primitive and orientation coding are easily of high dimension.

We present in Table 8 the classification accuracy results of several combinations of descriptors on KTH. We show the best descriptor results of our study for each primitives and codes aggregation, and compare them to recent results from the literature. Every single descriptor presented in Table 8 are comparable to those proposed by Wang et al. in [3]. Moreover, simple concatenation of all our signature (9) outperform the classification accuracy of Wang [3] and Gilbert [51]. Let us note that our approach uses linear classifiers, and thus leads to better efficiency both for training classifiers and classifying video shots, as opposed to methods of [3] and [51]. Moreover, we do not used dense trajectory to follow descriptors along time axis as in [3].

Method	ND	NL	Results
Wang (HOG+traj) [3]	1	X	86.5%
Wang (HOF+traj) [3]	1	X	93.2%
Wang (MBH+traj) [3]	1	X	**95.0**%
Wang (All) [3]	4	X	94.2%
Gilbert [51]	3*	X	94.5%
A = Gradient + ori + Cell (4,1)	1		93.4%
B = Gradient + ori + Poly (4,0)	1		92.6%
C = Gradient + ori + Sine (3,0)	1		93.3%
D = Motion + ori + Cell (4,1)	1		91.8%
E = Motion + rect + Poly (2,4)	1		91.7%
F = Motion + rect + Sine (1,2)	1		91.3%
G = Grad of Motion + rect + Cell (2,3)	1		94.2%
H = Grad of Motion + rect + Poly (4,0)	1		93.7%
I = Grad of Motion + rect + Sine (1,2)	1		93.4%
A + **D** + **G**	3		94.2%
B + **E** + **H**	3		94.4%
C + **F** + **I**	3		93.5%
A+...+I	9		**94.7**%

Table 8: Classification accuracy on the KTH dataset ; ND means the number of descriptors used ; NL stands for non-linear classifiers ; * In [51], the same feature is iteratively combined with itself 3 times

4.2.2. Comparison to State of the art

For further experiments and comparisons to literature on Hollywood2 and UCF11 dataset, we use the nine best descriptors identified thanks to our experiments on KTH dataset.

Hollywood dataset

The Hollywood2 [4] dataset consists of a collection of video clips and extracts from 69 films in 12 classes of human actions (Fig. 6). It accounts for approximately 20 hours of video and contains about 150 video samples per actions. It contains a variety of spatial scales, zoom camera, deleted scenes and compression artifact which allows a more realistic assessment of human actions classification methods. We use the official train and test splits for the evaluation.

UCF11 dataset

The UCF11 [19] dataset is an action recognition data set with 11 action categories, consisting of realistic videos taken from youtube (Fig. 7). The data set is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background and illumination conditions. The videos are grouped into 25 groups, where each group consists of more than 4 action clips. The video clips in the same group may share some common features, such as the same person, similar background or similar viewpoint. The experimental setup is a leave one group out cross validation.

Results

We select the best setup according to gradient primitive associated with cells and polynomials projections (c.f. Table 5), the best setup according to Motion primitive associated with cells and polynomials projections (c.f. Table 6) and the best setup according to Gradient of motion primitive associated with cells and polynomials projections (c.f. Table 7). These setups are evaluated on the Hollywood2 dataset and results are reported in Table 9. The results presented here improve the state of the art for single descriptor setups when comparing to HOG (gradient primitive), to HOF (motion primitive) and to MBH (gradient of motion primitive). Note that, opposed to [3], we do not use the dense trajectories to obtain these results. Our framework allows to increase the number of descriptor for a fixed number of primitives. Finally, by combining nine primitives, we obtain a mean average precision of 60.2%.

We evaluate the same descriptors on the UCF11 dataset and we report our results in Table 10. On UCF11 dataset, for all the primitives extraction, the cell aggregation and polynomial aggregation improve the results of Wang et al. [3] for single descriptor corresponding to that primitive. However, the Sine aggregation produces lower results in the case of Gradient primitive and Gradient of Motion primitive. When combining descriptors, we improve the results of Wang et al. [3] without using dense trajectories. The results obtained on the challenging UCF11 and Hollywood2 datasets with the combination of several descriptors highlight the soundness of our framework.

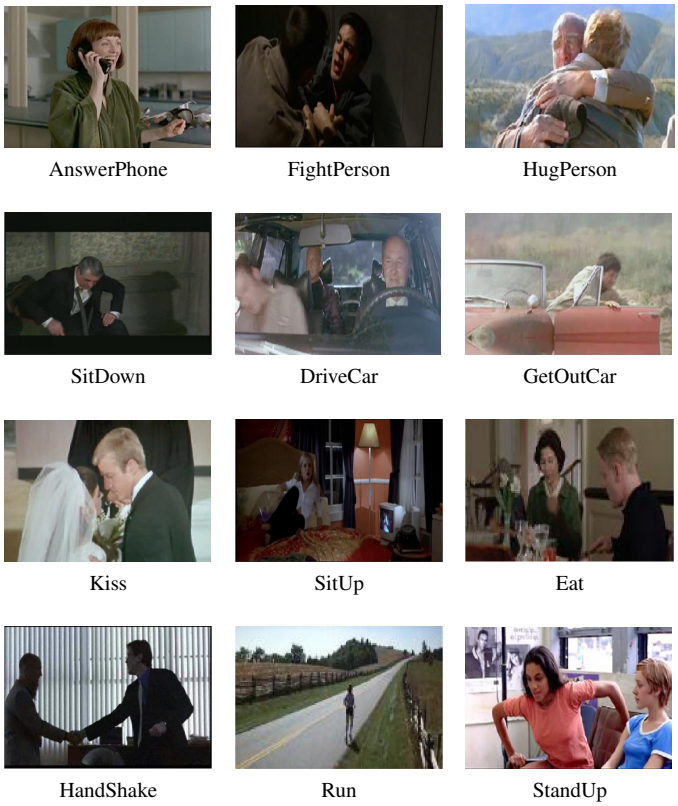


Figure 6: Example of videos from Hollywood2 dataset

Figure 7: Example of videos from UCF11

5. Conclusion

In this paper, we introduced a new framework to describe local visual descriptors. This framework consists in the decomposition of descriptors in three levels: primitive extraction, primitive coding and code aggregation. Our framework allows us to easily explain popular descriptors of the literature. Moreover, our framework allows us to propose extensions of popular descriptors, for instance by introducing a function based aggregation.

Using our framework, we experimented several combination of primitives extraction, primitive coding and code aggregation, some of them being drawn from the most popular descriptors. We obtain better or equivalent results for than the usual descriptors of literature on a popular still image categorization dataset and on three well known videos recognition datasets. This confirms the validity and relevance of our framework to create new descriptors. We are confident our framework can be used to implement descriptors families not covered in this paper (for example dense trajectories).

Furthermore, it is interesting to compare our framework to the coding/pooling approaches [37] used to compute signatures. Indeed, the two last steps of our framework (primitive coding and code aggregation) are close in their objective to the coding step and the pooling step in signature computing methods. In this case, the primitive extraction can be replaced by the extraction of a set of local descriptors. Future work involves adapting recent signature computation methods to the descriptors using our framework. For example, dictionary based approaches [36, 37] and model deviation approaches [40, 42, 44] can be used for the coding and aggregation steps. Future work also involves the optimization of the primitive step by using machine learning algorithms. For example, the primitive can be an adapted filter bank trained on some training set, in a similar way of the deep learning approaches [53] or the infinite kernel learning approaches [54].

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.
[2] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: BMVC, Vol. 76, 2011, pp. 1–12.
[3] H. Wang, A. Klaser, C. Schmid, C. Liu, Action recognition by dense trajectories, in: Conference on CVPR, IEEE, 2011, pp. 3169–3176.
[4] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Conference on CVPR, IEEE, 2008, pp. 1–8.
[5] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: A local svm approach, in: ICPR, Vol. 3, IEEE, 2004, pp. 32–36.
[6] D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.
[7] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, ECCV (2006) 404–417.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Conference on CVPR, IEEE, 2005, pp. 886–893.
[9] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, ECCV (2006) 428–441.
[10] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630.
[11] E. Tola, V. Lepetit, P. Fua, Daisy: An efficient dense descriptor applied to wide-baseline stereo, Transactions on Pattern Analysis and Machine Intelligence 32 (5) (2010) 815–830.
[12] J. Davis, A. Bobick, The representation and recognition of action using temporal templates, in: Conference on CVPR, IEEE, 1997, pp. 928–934.

Method	ND	NL	Results
Gilbert [51]	3	X	50.9%
Ullah [52] HOG+HOF	2	X	51.8%
Ullah [52]	2⋆	X	55.3%
Wang [3] traj	1	X	47.7%
Wang [3] HOG	1	X	41.5%
Wang [3] HOF	1	X	50.8%
Wang [3] MBH	1	X	54.2%
Wang [3] all	4	X	**58.3**%
A = Gradient + ori + Cell (4,1)	1		44.4%
B = Gradient + ori + Poly (4,0)	1		48.4%
C = Gradient + ori + Sine (3,0)	1		45.0%
D = Motion + rect + Cell (3,4)	1		53.3%
E = Motion + rect + Poly (2,4)	1		52.7%
F = Motion + rect + Sine (1,2)	1		49.5%
G = Grad of Motion + rect + Cell (2,3)	1		56.2%
H = Grad of Motion + rect + Poly (4,0)	1		54.9%
I = Grad of Motion + rect + Sine (1,2)	1		52.0%
A + **D** + **G**	3		59.1%
B + **E** + **H**	3		58.8%
C + **F** + **I**	3		56.4%
A+...+I	9		**60.2**%

Table 9: Mean Average Precision on the Hollywood2 dataset ; ND: number of descriptors ; NL: non-linear classifiers ; ⋆ In [52] HOG/HOF descriptors are accumulated on over 100 spatio-temporal regions each one leading to a different BoW signature

Method	ND	NL	Results
Wang [3] traj	1	X	67.2%
Wang [3] HOG	1	X	74.5%
Wang [3] HOF	1	X	72.8%
Wang [3] MBH	1	X	83.9%
Wang [3] all	4	X	**84.2**%
A = Gradient + ori + Cell (4,1)	1		80.1%
B = Gradient + ori + Poly (4,0)	1		81.8%
C = Gradient + ori + Sine (3,0)	1		73.0%
D = Motion + rect + Cell (3,4)	1		81.0%
E = Motion + rect + Poly (2,4)	1		82.6%
F = Motion + rect + Sine (1,2)	1		75.9%
G = Grad of Motion + rect + Cell (2,3)	1		84.2%
H = Grad of Motion + rect + Poly (4,0)	1		86.0%
I = Grad of Motion + rect + Sine (1,2)	1		79.2%
A + **D** + **G**	3		86.1%
B + **E** + **H**	3		**87.6**%
C + **F** + **I**	3		82.8%
A+...+I	9		86.5%

Table 10: Mean Average Precision on the UCF11 dataset ; ND: number of descriptors ; NL: non-linear classifiers ; ⋆ In [52] HOG/HOF descriptors are accumulated on over 100 spatio-temporal regions each one leading to a different BoW signature

[13] V. Kellokumpu, G. Zhao, M. Pietikäinen, Texture Based Description of Movements for Activity Analysis, in: VISAPP, Vol. 1, 2008, pp. 206–213.

[14] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 971–987.

[15] V. Kellokumpu, G. Zhao, M. Pietikäinen, Human activity recognition using a dynamic texture based method, in: BMVC, 2008, pp. 885–894.

[16] L. Wang, D. Suter, Learning and matching of dynamic shape manifolds for human action recognition, IEEE Transactions on Image Processing 16 (6) (2007) 1646.

[17] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: ICCV, Vol. 2, IEEE, 2005, pp. 1395–1402.

[18] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2247–2253.

[19] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: Conference on CVPR, IEEE, 2009, pp. 1996–2003.

[20] R. Polana, R. Nelson, Low level recognition of human motion, in: Proc. IEEE Workshop on Nonrigid and Articulate Motion, 1994, pp. 77–82.

[21] A. Efros, A. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: ICCV, Vol. 2, IEEE, 2003, pp. 726–733.

[22] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proceedings of the 7th international joint conference on Artificial intelligence, Vol. 2, 1981, pp. 674–679.

[23] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: Conference on CVPR, IEEE, 2008, pp. 1–8.

[24] S. Danafar, N. Gheissari, Action recognition for surveillance applications using optic flow and svm, in: ACCV, Vol. 4844, 2007, pp. 457–466.

[25] D. Tran, A. Sorokin, Human activity recognition with metric learning, ECCV (2008) 548–561.

[26] S. Ali, M. Shah, Human action recognition in videos using kinematic features and multiple instance learning, Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 288–303.

[27] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance,, IEEE, 2005, pp. 65–72.

[28] A. Klaser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: BMVC, 2008.

[29] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th international conference on Multimedia, ACM, 2007, pp. 357–360.

[30] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, ECCV (2008) 650–663.

[31] O. Kihl, B. Tremblais, B. Augereau, M. Khoudeir, Human activities discrimination with motion approximation in polynomial bases, in: ICIP, IEEE, 2010, pp. 2469–2472.

[32] V. F. Mota, E. Perez, M. B. Vieira, L. Maciel, F. Precioso, P.-H. Gosselin, A tensor based on optical flow for global description of motion in videos, in: 25th SIBGRAPI Conference on Graphics, Patterns and Images, IEEE, 2012, pp. 298–301.

[33] O. Kihl, D. Picard, P.-H. Gosselin, Local polynomial space-time descriptors for actions classification, in: IAPR MVA, Kyoto, Japon, 2013.

[34] J. Sánchez, F. Perronnin, T. d. Campos, Modeling the spatial layout of images beyond spatial pyramids, Pattern Recognition Letters.

[35] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: BMVC, 2009.

[36] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: ICCV, Vol. 2, IEEE, 2003, pp. 1470–1477.

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Conference on CVPR, IEEE, 2010, pp. 3360–3367.

[38] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Conference on CVPR, IEEE, 2009, pp. 1794–1801.

[39] S. Avila, N. Thome, M. Cord, E. Valle, A. de A Araujo, Bossa: Extended bow formalism for image classification, in: ICIP, IEEE, 2011, pp. 2909–2912.

[40] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Conference on CVPR, IEEE, 2010, pp. 3304–3311.

[41] X. Zhou, K. Yu, T. Zhang, T. Huang, Image classification using super-vector coding of local image descriptors, Computer Vision–ECCV 2010

(2010) 141–154.
[42] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 1704–1716.
[43] D. Picard, P.-H. Gosselin, Efficient image signatures and similarities using tensor products of local descriptors, CVIU 117 (6) (2013) 680–687.
[44] D. Picard, P.-H. Gosselin, Improving image similarity with vectors of locally aggregated tensors, IEEE, 2011, pp. 669–672.
[45] R. Negrel, D. Picard, P. Gosselin, Using spatial pyramids with compacted vlat for image categorization, in: ICPR, 2012, pp. 2460–2463.
[46] M. Varma, A. Zisserman, A statistical approach to material classification using image patch exemplars, Transactions on Pattern Analysis and Machine Intelligence 31 (11) (2009) 2032–2047.
[47] R. Negrel, D. Picard, P. Gosselin, Compact tensor based image representation for similarity search, in: ICIP, IEEE, 2012, pp. 2425–2428.
[48] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Conference on CVPR, Vol. 2, IEEE, 2006, pp. 2169–2178.
[49] L.-J. Li, H. Su, E. P. Xing, L. Fei-Fei, Object bank: A high-level image representation for scene classification and semantic feature sparsification, Advances in Neural Information Processing Systems 24.
[50] B. Horn, B. Schunck, Determining optical flow, Artificial intelligence 17 (1) (1981) 185–203.
[51] A. Gilbert, J. Illingworth, R. Bowden, Action recognition using mined hierarchical compound features, Transactions on Pattern Analysis and Machine Intelligence (99) (2011) 883–897.
[52] M. Ullah, S. Parizi, I. Laptev, Improving bag-of-features action recognition with non-local cues, in: BMVC, 2010.
[53] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1915–1929.
[54] A. Rakotomamonjy, R. Flamary, F. Yger, Learning with infinitely many features, Machine Learning 91 (1) (2013) 43–66. doi:10.1007/s10994-012-5324-5.

Asynchronous gossip principal components analysis

Jerome Fellus, David Picard, Philippe-Henri Gosselin

Asynchronous Gossip Principal Components Analysis

Jerome FELLUS[a,*], David PICARD[a], Philippe-Henri GOSSELIN[a],

[a]*ETIS - UMR CNRS 8051 - ENSEA - Universite de Cergy-Pontoise*
6 Avenue du Ponceau 95014 Cergy, France

Abstract

This paper deals with Principal Components Analysis (PCA) of data spread over a network where central coordination and synchronous communication between networking nodes are forbidden. We propose an asynchronous and decentralized PCA algorithm dedicated to large scale problems, where "large" simultaneously applies to dimensionality, number of observations and network size. It is based on the integration of a dimension reduction step into a Gossip consensus protocol. Unlike other approaches, a straightforward dual formulation makes it suitable when observed dimensions are distributed. We theoretically show its equivalence with a centralized PCA under a low-rank assumption on training data. An experimental analysis reveals that it achieves a good accuracy with a reasonable communication cost even when the low-rank assumption is relaxed.

Keywords: Distributed Machine Learning, Dimensionality reduction, Gossip protocols

1. Introduction

Dimensionality reduction plays an important role in solving large scale machine learning problems where input data usually consists of a huge number of observations in a high-dimensional space. Classification, regression, or similarity ranking of such raw data often raise computation and storage issues. In practice, the intrinsic dimensionality of observed phenomena is much lower than the extrinsic dimension of the input space. Dimensionality reduction then aims at pro-

*Corresponding author
Email addresses: jerome.fellus@ensea.fr (Jerome FELLUS), picard@ensea.fr (David PICARD), gosselin@ensea.fr (Philippe-Henri GOSSELIN)

jecting input data into a lower-dimensional space such that subsequent learning stages keep a maximal accuracy.

Principal Components Analysis (PCA) is a linear approach to dimensionality reduction [1, 2]. Given a sample matrix $\mathbf{X} \in \mathbb{R}^{D \times n}$ made of n observations in $\mathbb{R}^D$, PCA finds an orthonormal basis $\mathbf{U}^\star = [\mathbf{u}_1 \dots \mathbf{u}_q]$, $\mathbf{u}_k \in \mathbb{R}^D$ that projects the input sample $\mathbf{X}$ into the q-dimensional subspace, $q < D$, that retains the maximal variance in $\mathbf{X}$. Equivalently, the PCA solution is the linear projection that best conserves the Gram matrix (*i.e.*, the matrix of pairwise inner-products):

$$\mathbf{U}^\star = \underset{\mathbf{U} \in \mathbb{R}^{D \times q}}{\arg\min} \left\| \mathbf{X}^\top \mathbf{X} - \mathbf{X}^\top \mathbf{U} \mathbf{U}^\top \mathbf{X} \right\|_F^2 \quad \text{s.t.} \quad \mathbf{U}^{\star\top} = \mathbf{U}^{\star -1} \tag{1}$$

The optimal conservation of the inner product makes PCA particularly suited to feed algorithms that solely rely on the inner product instead of the input data [3] (*e.g.*, Support Vector Machines). PCA enjoys a closed-form solution, as $\mathbf{U}^\star$ is made of the q leading eigenvectors of the sample covariance matrix [2]:

$$\mathbf{C} = \frac{1}{n} \mathbf{X} \mathbf{X}^\top - \mu \mu^\top \quad \text{where} \quad \mu = \frac{1}{n} \mathbf{X} \mathbf{1} \quad \text{is the sample mean} \tag{2}$$

Like most statistical learning tools, PCA was formulated for centralized setups where all data are available at a single location. This assumes that the solution can be computed by a single machine and that all intermediary results fit in the main memory. However, this assumption is unrealistic in most applicative fields that deal with very large sample matrices. For instance, in biomedical, multimedia, or remote sensing applications, D and n often grows up to millions. The sample and covariance matrices then scale in Terabytes. Moreover, the $O(D^3)$ complexity of covariance eigendecomposition translates into an exa-flop computation cost. Besides, along with the democratization of connected devices, data tends to originate from an increasing number of distributed sources with reasonable computing capacity, immersed in large unreliable networks without any central coordination. This has led to a number of so-called Distributed PCA algorithms, designed to deal with the spread of input data over multiple networking nodes.

Because computing μ and $\mathbf{C}$ would involve the full data $\mathbf{X}$ which is unknown to individual nodes, distributed PCA requires dedicated algorithms combining node-local optimization and inter-node communications. Distributed PCA encompasses two main scenarios, depending of the way the entries of $\mathbf{X}$ are spread over the networking nodes. Consider a network made of N (strongly-connected) nodes. In a Distributed Samples (DS) scenario, each node i holds a distinct sample $\mathbf{X}_i \in \mathbb{R}^{D \times n_i}$ of n_i observations (*i.e.*, the *columns* of $\mathbf{X}$ are distributed). Con-

versely, in a Distributed Coordinates (DC) scenario, each node holds all n observations, but only gets a subset $\mathbf{X}_i \in \mathbb{R}^{D_i \times n}$ of their components (*i.e.*, the *rows* of $\mathbf{X}$ are distributed). On average, each $\mathbf{X}_i$ is then N times smaller than $\mathbf{X}$, thus $n_i \ll D$ in DS case while $D_i \ll n$ in DC case. No assumption is made on their exact sizes, as they may be very different at each node. In both DS and DC scenarios, a typical objective is to provide all nodes with compressed representations of their locally-hosted observations that account for the contribution of all components. Nodes then have to find a consensus basis $\mathbf{U}^\star$ that minimizes the PCA objective defined in Equation (1) over the complete data. Usually, one also wants an operator that allows projection of future observations into the same output space, but not all approaches are able to provide such operator at a low cost. In spite of similar goals, the DS and DC scenarios have been tackled separately with very different approaches in the distributed PCA litterature [4].

In this work, we consider the asynchronous decentralized PCA problem, which specializes distributed PCA by adding the following constraints:

(C1) No sample exchange - Samples cannot be exchanged between nodes, for size, privacy or property reasons.

(C2) Asynchrony - Nodes must never wait for each other.

(C3) Decentralization - All nodes and links must play the same role. Formally, nodes and links must be selected for communication with the same probability and all nodes must run the same procedures.

(C4) Consensus - All nodes must obtain the same orthogonal basis. Node-local solutions must allow projection of future observations.

Satisfying these four constraints, a distributed PCA algorithm gains applicability to a wider range of networking situations such as sensor networks, Internet-enabled multimedia devices, cloud computing systems, etc, where central coordination or synchronous functioning can be inapplicable.

In this paper, we propose a decentralized and asynchronous algorithm called Asynchronous Gossip Principal Component Analysis (AGPCA) that satisfy all the above constraints. Our algorithm is a revision and extension of previous work presented in [5]. The original contributions of this paper are the following:

- We give a formal and in-depth description of AGPCA in the DS case, as well as the intuitions leading to the algorithm.

- We propose an extension to the DC case through a dual transcription.
- Provided a low rank property is met on the data, we give a theoretical guarantee that AGPCA yields the exact solution of the centralized PCA.
- We present experiments for both DS and DC scenarios, as well as results for various network topologies.

The remaining of this paper is organized as follows: In the next Section, we detail related works on distributed PCA algorithms. Then, we present our AGPCA algorithm for the DS case in Section 3. We present the extension to the DC case in Section 4. In section 5, we theoretically show that the output of AGPCA is identical to a centralized PCA under a low-rank assumption on the data. The last section gathers experimental results both in DS and DC case, before we conclude.

2. Related Work

In this section, we present existing algorithms for distributed PCA. We first present methods that integrate prior information on the input data. Then, we compare existing algorithms in terms of decentralization and asynchrony. Finally, we discuss the benefits of approaches based on model aggregation over those based on iterative optimization passes over the data.

2.1. Prior knowledge about input data

Existing distributed PCA approaches can be first distinguished by their level of prior knowledge about the input data matrix $\mathbf{X}$. Indeed, $\mathbf{X}$ can either carry node-local observations independently of their network relationships or integrate properties of the network graph itself. In the latter case, a typical object of interest is the adjacency matrix of the weighted network graph, whose entries correspond to some scalar relationship *between* nodes. For instance, when these entries represent estimated geographic distances between neighbouring sensors, their absolute geographic position can be recovered by computing the three principal components through distributed PCA [6].

Other methods have considered the case where the data distribution inherits specific characteristics from the network structure, such as statistical dependencies. This happens when, *e.g.*, data is generated by flowing through directed paths along the network structure, thus making data at downstream nodes dependent of data at upstream nodes, but independent from each other. Properly modeling these statistical dependencies through Graphical Models, either undirected (*e.g.*,

Decomposable Gaussian Graphical Models [7]) or directed (*e.g.*, Directed Gaussian Acyclic Graphical Models [8]), one can benefit from the natural sparsity of the concentration matrix (*i.e.*, the inverse covariance) or its Cholesky factor to estimate the principal subspace with reduced communication costs.

On the contrary, in this work the network topology has no relevance in the desired result, as information is solely carried by the nodes and not by the links. Still, link-related data can be seen as observations relative to one or both of their ends, making methods aimed at node-related data suitable for link-related data.

2.2. Decentralization and asynchrony

Another classification criterion separates decentralized approaches from those which assign node-specific roles and asynchronous approaches from those based on synchronous communications.

In [9], a parallel PCA algorithm is proposed. Sufficient statistics $\mathbf{X}_i\mathbf{1}$ and $\mathbf{X}_i\mathbf{X}_i^\top$ are locally computed at all nodes and transmitted to a master node that performs a global Singular Value Decomposition (SVD) to obtain the PCA result. This approach is only suitable when the master node can handle the $O(D^3)$ complexity of the SVD and assumes that $D \times D$ covariance matrices can be exchanged on the network, which is unrealistic in many large scale contexts.

In [4], a distributed PCA algorithm for the DC scenario is proposed. Exchanging only $q \times q$ matrices, nodes iteratively maximize the variance retained by the reduced basis. Even though the process is decentralized, nodes have to update their estimates synchronously before performing any further computation, thus violating (**C2**). The whole system performance is then limited by the slowest networking node. Moreover, synchronous updating is hard to sustain in large networks and can result in overwhelming idle phases even when nodes have identical computing resources.

A fully asynchronous and decentralized Power Iteration method is proposed in [10] using random matrix sparsifications and a nested Sum-Weight Gossip averaging protocol to reduce communication costs. However, its original formulation only provides the first principal component. Synchronous passes would be required to sequentially obtain the next ones.

2.3. Multiple passes over the data vs. model aggregation

An important feature of some methods is to require only one pass over node-local data. This is the case of [9, 11], contrarily to [6, 10] which require multiple access to the $\mathbf{X}_i$ because of their iterative optimization scheme. Temporary access

to data is frequently considered in online approaches like [4] to handle stream data under short-time ergodicity and stationarity assumptions.

In [11], the authors propose an operator to aggregate Mixtures of Probabilistic PCA models (MPPCA, [12]) in a maximum-likelihood sense, without resorting to the original data used to train the models. Multiple models can thus be trained in a first phase, and aggregated in a second phase. Used in conjunction with the Gossip optimization framework proposed in [13], one can easily obtain a decentralized and asynchronous PCA algorithm. However, as developped in [14], model aggregation give best results when models are further selected through neighbors cross-validation, which seems non-trivial to achieve in an asynchronous fashion.

In [15], the authors first compute sufficient local statistics and aggregate them by means of a distributed consensus algorithm before locally computing PCA on the aggregated statistics. Aggregation of the local statistics requires exchanging $D \times D$ matrix estimates, which can be prohibitive for high-dimensional input spaces. Our approach is akin to [15], but solves this problem by computing PCA *before* the distributed aggregation.

3. Asynchronous Gossip PCA for Distributed Samples scenario

In this section, we introduce our proposed AGPCA algorithm, which is suitable for both DS and DC scenarios. This section details AGPCA for the DS scenario. We present the extension of AGPCA to the DC scenario in the next section.

In a DS scenario, each node hosts a local sample $\mathbf{X}_i \in \mathbb{R}^{D \times n_i}$ made of n_i observations $[\mathbf{x}_1, \ldots, \mathbf{x}_{n_i}]$ in $\mathbb{R}^D$. In such scenario, AGPCA provides all nodes with the same orthogonal basis $\mathbf{U}^\star \in \mathbb{R}^{D \times q}$ that spans the q-principal subspace of the full covariance matrix $\mathbf{C}$ as defined by Equation (2). Each node i can then project its local data $\mathbf{X}_i$ to obtain $\mathbf{Y}_i = \mathbf{U}^{\star\top}\mathbf{X}_i$, where $\mathbf{Y}_i \in \mathbb{R}^{q \times n_i}$ is the compressed representation of $\mathbf{X}_i$. Importantly, any future observation $\mathbf{x}_{new}$ can be compressed in the same way: $\mathbf{y}_{new} = \mathbf{U}^{\star\top}\mathbf{x}_{new}$, even if $\mathbf{X}_i$ is deleted or was only accessible during training (*e.g.*, streaming data case).

The intuition behind AGPCA is built upon the following 4 main principles:

1. Distributed PCA can be formulated as a distributed averaging of covariance matrices followed by local eigendecompositions.
2. Asynchronous and decentralized averaging of covariance matrices is possible through Sum-Weight Gossip consensus protocols [16].
3. By properly defining uniform scaling and sum operators for orthogonal bases, we can extend the Sum-Weight Gossip protocol to exchange only the

Algorithm 1 AGPCA-DS Emission procedure

$\mathbf{a}_i \leftarrow \mathbf{X}_i\mathbf{1}$; $\mathbf{G}_i \leftarrow \mathbf{X}_i^\top\mathbf{X}_i$; $w_i \leftarrow n_i$
$(\mathbf{V}_i, \mathbf{L}_i) \leftarrow$ *eigendecompose*$(\mathbf{G}_i)$
$\mathbf{U}_i \leftarrow \mathbf{X}_i\mathbf{V}_i\mathbf{\Lambda}_i^{-\frac{1}{2}}$
loop
  $j \leftarrow$ *randomNeighbor*(i)
  $(\mathbf{a}_i, \mathbf{L}_i, w_i) \leftarrow \frac{1}{2}(\mathbf{a}_i, \mathbf{L}_i, w_i)$
  Send $(\mathbf{a}_i, \mathbf{U}_i, \mathbf{L}_i, w_i)$ to j
end loop

Algorithm 2 AGPCA-DS Reception procedure

loop
  Upon receipt of $(\mathbf{a}_j, \mathbf{U}_j, \mathbf{L}_j, w_j)$
  $\mathbf{a}_i \leftarrow \mathbf{a}_i + \mathbf{a}_j$; $w_i \leftarrow w_i + w_j$
  $\mathbf{Q}_0 \leftarrow \mathbf{U}_i$
  for $t \in [0, max\_t[$ **do**
    $(\mathbf{Q}_{t+1}, \mathbf{R}_{t+1}) \leftarrow$ *QR*$(\mathbf{U}_i\mathbf{L}_i\mathbf{U}_i^\top\mathbf{Q}_t + \mathbf{U}_j\mathbf{L}_j\mathbf{U}_j^\top\mathbf{Q}_t)$
  end for
  $\mathbf{U}_i \leftarrow \mathbf{Q}_{max\_t}$; $\mathbf{L}_i \leftarrow$ *diag*$(\mathbf{R}_{max\_t})$
end loop

q leading eigenvectors of the local covariance matrices, and consequently achieve a better network efficiency.

4. Using the duality between the eigendecomposition of the covariance and Gram matrices through Singular Value Decomposition (SVD), we can compute the local Gram matrices ($n_i \times n_i$) instead of the local covariance matrices ($D \times D$) and consequently reduce the memory usage at each node.

In the remaining of this section, we detail each of these points leading to AGPCA procedures for the DS scenario to be concurrently run at each node as presented in Algorithm 1 and 2[1].

[1]When implementing AGPCA, care should be taken that iterations of Algorithm 1 and Algorithm 2 be mutually exclusive, to ensure integrity of the concurrently updated variables.

3.1. Distributed PCA as a distributed averaging problem

From Equation (1), the sample correlation matrix $\mathbf{X}\mathbf{X}^\top$ and the sample sum $\mathbf{X}\mathbf{1}$ are sufficient statistics to compute the PCA solution. As observed in [15, 9], these statistics can be obtained by computing partial statistics $\mathbf{X}_i\mathbf{1}$ and $\mathbf{X}_i\mathbf{X}_i^\top$ at each node i and summing over i. As a result, the full sample mean and covariance matrix can be computed as a distributed average of locally-computed means $\mu_i = \frac{1}{n_i}\mathbf{X}_i\mathbf{1}$ and correlation matrices $\mathbf{B}_i = \mathbf{X}_i\mathbf{X}_i^\top$:

$$\mu = \frac{1}{n}\mathbf{X}\mathbf{1} = \frac{1}{n}\sum_i^N \mathbf{X}_i\mathbf{1} = \frac{1}{\sum_i n_i}\sum_i^N n_i\mu_i \tag{3}$$

$$\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top - \mu\mu^\top = \frac{1}{n}\sum_i^N \mathbf{X}_i\mathbf{X}_i^\top - \mu\mu^\top = \frac{1}{\sum_i n_i}\sum_i^N \mathbf{B}_i - \mu\mu^\top \tag{4}$$

Provided these weighted averages are computed in a decentralized and asynchronous fashion, all nodes get the full covariance matrix $\mathbf{C}$. Locally eigendecomposing $\mathbf{C}$ at every nodes finally gives the global PCA solution $\mathbf{U}^\star$.

Remark that $\mathbf{X}_i\mathbf{1}$ and $\mathbf{X}_i\mathbf{X}_i^\top$ are easily computed together in a single pass over $\mathbf{X}_i$. Also notice that once local sufficient statistics are computed, input samples can be dropped without concern. Therefore, such a scheme is suitable when streaming data.

3.2. Gossip protocols for asynchronous decentralized averaging

To compute Equations (3-4) in a decentralized and asynchronous fashion, AG-PCA uses a Sum-Weight Gossip protocol [16, 17].

Sum-Weight protocols are an asynchronous subclass of Gossip consensus algorithms for distributed averaging. Gossip consensus algorithms proceed by iterative linear combinations of node estimates through randomized communications.

An example of Gossip averaging protocol is Newscast [18]. In Newscast, random node pairs regularly average their estimates until reaching consensus. Assuming each node i holds a local vector estimate $\mathbf{v}_i$ in a vector space $\mathcal{V}$, at any random time t two random nodes s and r awake, exchange their estimates and update as follows:

$$\mathbf{v}_s(t+1) = \frac{1}{2}(\mathbf{v}_s(t) + \mathbf{v}_r(t)) \qquad \mathbf{v}_r(t+1) = \frac{1}{2}(\mathbf{v}_s(t) + \mathbf{v}_r(t)) \tag{5}$$

This update rule entails two properties:

- **Mass conservation**. The sum of the estimates over the network is conserved:

$$\sum_i \mathbf{v}_i(t) = \sum_i \mathbf{v}_i(0)$$

- **Convergence to consensus**. The variance of the estimates tends to zero:

$$\lim_{t\to\infty} \sum_i \left| \mathbf{v}_i(t) - \sum_j \mathbf{v}_j(t) \right|^2 = 0$$

A trivial consequence of these properties is $\forall i, \lim_{t\to\infty} \mathbf{v}_i(t) = \frac{1}{N}\sum_j \mathbf{v}_j(0)$. Unfortunately, Newscast requires synchronous updating of random node pairs, thus violating constraint **(C2)**.

Sum-Weight protocols aims at removing pairwise synchronization by adding a new estimate $w_i \in \mathbb{R}$ called weight, with initial value $w_i(0) = 1$. In contrast to Newscast, all nodes send their current sum and weight to randomly selected neighbors following independent Poisson emission clocks and select targets independently at random *without* waiting for their answer. The update rules of the sender node s and receiver node r are modified as follows:

$$\mathbf{v}_s(t+1) = \frac{1}{2}\mathbf{v}_s(t) \qquad w_s(t+1) = \frac{1}{2}w_s(t) \tag{6}$$

$$\mathbf{v}_r(t+1) = \mathbf{v}_r(t) + \frac{1}{2}\mathbf{v}_s(t) \qquad w_r(t+1) = w_r(t) + \frac{1}{2}w_s(t) \tag{7}$$

As first shown in [16] under synchronous assumptions, and then in [17] for the general case, the quotient of the two estimates converges to the desired average.

$$\forall i, \quad \lim_{t\to\infty} \frac{1}{w_i(t)}\mathbf{v}_i(t) = \frac{1}{\sum_j w_j(0)} \sum_j \mathbf{v}_j(0) = \frac{1}{N}\sum_j \mathbf{v}_j(0) \tag{8}$$

Moreover, convergence to the consensus is exponential provided that the network has a sufficient conductance [19]. In this case, the number of message exchanges required to achieve a given estimation error ε scales logarithmically with the number of nodes and ε.

As weighted averages, μ and $\mathbf{C}$ can be estimated using the above-defined Sum-Weight protocol, by defining node-local estimates $\mathbf{a}_i(t), \mathbf{B}_i(t)$ and weights $w_i(t)$ such that:

$$\mathbf{a}_i(0) = \mathbf{X}_i\mathbf{1} \qquad \mathbf{B}_i(0) = \mathbf{X}_i\mathbf{X}_i^\top \qquad w_i(0) = n_i \tag{9}$$

By applying the Gossip protocol defined by Equations (6-7) to $\mathbf{a}_i(t)$,$\mathbf{B}_i(t)$ and $w_i(t)$, we get a covariance estimate $\mathbf{C}_i(t)$:

$$\mathbf{C}_i(t) = \frac{\mathbf{B}_i(t)}{w_i(t)} - \frac{\mathbf{a}_i(t)\mathbf{a}_i(t)^\top}{w_i(t)^2} \tag{10}$$

Note that initial estimates $\mathbf{C}_i(0)$ are the covariance matrices of their corresponding $\mathbf{X}_i$. The limit in (8) shows that each $\frac{\mathbf{a}_i(t)}{w_i(t)}$ tends to the global mean μ and each $\mathbf{C}_i$ tends to the global covariance matrix $\mathbf{C}$:

$$\forall i, \begin{cases} \lim\limits_{t\to\infty} \dfrac{\mathbf{a}_i(t)}{w_i(t)} = \dfrac{\sum_i \mathbf{X}_i^\top \mathbf{1}}{\sum_i n_i} = \dfrac{\sum_i n_i \mu_i}{\sum_i n_i} = \mu \\ \lim\limits_{t\to\infty} \mathbf{C}_i(t) = \dfrac{\sum_i \mathbf{B}_i(0)}{\sum_i w_i(0)} - \mu\mu^\top = \dfrac{1}{n}\sum\limits_i \mathbf{X}_i\mathbf{X}_i^\top - \mu\mu^\top = \mathbf{C} \end{cases} \tag{11}$$

Once each node gets a sufficiently accurate estimate for $\mathbf{C}$, the final PCA result can be locally computed at any node i by eigendecomposition of $\mathbf{C}_i$.

Remark this strategy, which will be referred to as *Late-PCA* in the rest of the paper, is used in [15] in a synchronous consensus framework. Yet it suffers one major drawback: updating estimates $\mathbf{B}_i$ using Equations (6-7) requires transmission of $D \times D$ matrices, which can be too large to exchange on the network. In [5], we thus proposed to reduce matrices $\mathbf{B}_i$ by means of local PCA *before* their transmission, resulting in what we called an *Early-PCA* scheme.

3.3. Gossiping in the compressed domain: Early PCA

There are two main reasons that make the Late PCA approach unwanted, both based on the fact that such a scheme does not benefit from the rank-deficiency of the local covariance matrices implied by $n_i \ll D$. Firstly, the size of exchanged information is homogeneous to the input statistics ($D \times D$), not to the output result ($D \times q$). Therefore, we vainly exchange information that will be canceled in the end, because dimensionality reduction happens *after* the distributed averaging phase. Secondly, the distributed nature of the process does not allow any computational advantage for high dimension, since all nodes have to perform the same $\mathcal{O}(D^3)$ eigendecomposition operation. Consequently, if the eigendecomposition of $\mathbf{C}$ is the computational bottleneck due to large values of D, the distributed scheme is unlikely to bring any speed-up gain.

The Early-PCA approach aims at moving the dimension reduction step *before* the distributed averaging step, in order to take advantage of the rank deficiency

of $\mathbf{C}_i$. To this purpose, we reformulate the updates rules in Equations (6-7) so as to handle eigenpairs instead of the full matrices $\mathbf{B}_i$. This allows us to implicitly compute the average of very large matrices without ever resorting to their explicit form but rather using their factorized expression.

Observe that the senders update rule (6) is a simple uniform scaling. Assuming s holds the decomposed form $\mathbf{U}_s\mathbf{L}_s\mathbf{U}_s^\top$ of its estimate $\mathbf{B}_s$, the update is simply:

$$\mathbf{U}_s(t+1) = \mathbf{U}_s(t) \quad \mathbf{L}_s(t+1) = \frac{1}{2}\mathbf{L}_s(t) \tag{12}$$

This update leads to the emission procedure of AGPCA to be concurently run at every node of the network, as presented in Algorithm 1.

Adapting the receivers update rule (7) is slightly more involving since we need to compute the eigendecomposition of a sum of two matrices given their own eigendecompositions. We propose to rely on the well-known Orthogonal Iterations technique (as in [6]) to fit an orthonormal basis to the principal subspace of an input matrix by iterating QR decompositions. Assume $\mathbf{B}_s(t)$ and $\mathbf{B}_r(t)$ are respectively factorized as $\mathbf{U}_s(t)\mathbf{L}_s(t)\mathbf{U}_s(t)^\top$ and $\mathbf{U}_r(t)\mathbf{L}_r(t)\mathbf{U}_r(t)^\top$. Starting from a random $D \times q$ basis $\mathbf{Q}_0$, and denoting by $QR(\cdot)$ the economy QR decomposition, we iteratively compute

$$\begin{aligned}\mathbf{Q}_{\tau+1}\mathbf{R}_{\tau+1} &= QR((\mathbf{B}_s + \mathbf{B}_r)\mathbf{Q}_\tau) \\ &= QR(\mathbf{U}_s\mathbf{L}_s(\mathbf{U}_s^\top\mathbf{Q}_\tau) + \mathbf{U}_r\mathbf{L}_r(\mathbf{U}_r^\top\mathbf{Q}_\tau))\end{aligned} \tag{13}$$

$\mathbf{Q}_\tau$ tends to become an orthonormal basis for the q-principal subspace of $\mathbf{B}_s(t) + \mathbf{B}_r(t)$, with corresponding eigenvalues on the diagonal of $\mathbf{R}_\tau$. Thus, $\mathbf{Q}_\infty$ gives $\mathbf{U}_r(t{+}1)$ and the diagonal entries of $\mathbf{R}_\infty$, denoted by $diag(\mathbf{R}_\infty)$, give $\mathbf{L}_r(t{+}1)$.

The parentheses in Equation (13) are of great importance. Indeed, observe that instead of expanding $\mathbf{U}_s\mathbf{L}_s\mathbf{U}_s^\top$ and $\mathbf{U}_r\mathbf{L}_r\mathbf{U}_r^\top$, we first multiply $\mathbf{U}_s^\top$ and $\mathbf{U}_r^\top$ by $\mathbf{Q}_\tau$, resulting in a $q \times q$ matrix. By doing so, we never store any $D \times D$ matrix, and the QR decomposition is rather performed on a $D \times q$ matrix.

Finally, the dominant eigenvectors $\mathbf{U}^\star$ of $\mathbf{C}_i$ are obtained by locally computing its eigendecomposition using Orthogonal Iterations (parentheses are also added to highlight the computational gain):

$$\mathbf{C}_i = \frac{1}{w_i(t)}(\mathbf{U}_i\mathbf{L}_i)(\mathbf{U}_i^\top) - \frac{1}{w_i(t)^2}(\mathbf{a}_i(t))(\mathbf{a}_i(t)^\top) \tag{14}$$

This update leads to the reception procedure concurrently run at every node and presented in Algorithm 2.

Since typically $q \ll D$, combining Algorithms 1 and 2 allows to define an averaging protocol that exchanges $D \times q$ messages while staying equivalent to the original Sum-Weight protocol defined by Equations (6-7), provided that the rank of $\mathbf{X}\mathbf{X}^\top$ is lower than q. This is formally supported in the theoretical analysis presented in Section 5.

3.4. Dual relation between covariance and Gramian eigendecompositions

While Early-PCA gets rid of $D \times D$ entities in the networking step, we can further avoid $D \times D$ matrices in the entire algorithm. This is particularly useful when covariance matrices do not fit in node memory. To this end, we must avoid explicit computation and storage of the initial $\mathbf{B}_i(0)$ locally computed in Equation (9), reminding that we are only interested in their q principal subspace. Thankfully, the eigendecompositions $\mathbf{B}_i = \mathbf{U}_i\mathbf{L}_i\mathbf{U}_i^\top$ can be obtained from those of the $n_i \times n_i$ local Gram matrices $\mathbf{X}_i^\top\mathbf{X}_i = \mathbf{V}_i\mathbf{\Lambda}_i\mathbf{V}_i^\top$, through their well-known relation with the Singular Value Decomposition (SVD) of $\mathbf{X}$. Indeed,

$$\mathbf{B}_i^2 = \mathbf{X}_i\mathbf{X}_i^\top\mathbf{X}_i\mathbf{X}_i^\top = \mathbf{X}_i\mathbf{V}_i\mathbf{\Lambda}_i\mathbf{V}_i^\top\mathbf{X}_i^\top = (\mathbf{X}_i\mathbf{V}_i\mathbf{\Lambda}_i^{-\frac{1}{2}})\mathbf{\Lambda}_i^2(\mathbf{X}_i\mathbf{V}_i\mathbf{\Lambda}_i^{-\frac{1}{2}})^\top$$
$$\text{and} \qquad (\mathbf{X}_i\mathbf{V}_i\mathbf{\Lambda}_i^{-\frac{1}{2}})^\top(\mathbf{X}_i\mathbf{V}_i\mathbf{\Lambda}_i^{-\frac{1}{2}}) = \mathbf{I}$$

Then,

$$\mathbf{U}_i = \mathbf{X}_i\mathbf{V}_i\mathbf{\Lambda}_i^{-\frac{1}{2}} \qquad \text{and} \qquad \mathbf{L}_i = \mathbf{\Lambda}_i \tag{15}$$

We thus compute, store and factorize $\mathbf{X}_i^\top\mathbf{X}_i$ (which is $n_i \times n_i$) instead of $\mathbf{X}_i\mathbf{X}_i^\top$ (which is $D \times D$) and obtain $\mathbf{U}_i$ and $\mathbf{L}_i$ with no additional storage cost.

4. Asynchronous Gossip PCA for Distributed Coordinates scenario

In this section, we present the DC scenario and show that AGPCA can solve it as well. Contrarily to the DS scenario, each node hosts a local sample $\mathbf{Z}_i \in \mathbb{R}^{D_i \times n}$ made of $D_i \ll D$ dimensions of the same set of n observations. The full data $\mathbf{Z}$ corresponds to stacking the $\mathbf{Z}_i$ in rows:

$$\mathbf{Z}^\top = [\mathbf{Z}_1^\top, \ldots, \mathbf{Z}_N^\top] \tag{16}$$

Observations thus lie in the direct sum of the locally observed subspaces $\bigoplus_i \mathbb{R}^{D_i}$. Let $\mu = \mathbf{Z}\mathbf{1}/n$ denote the data mean and $\tilde{\mathbf{Z}} = \mathbf{Z} - \mu\mathbf{1}^\top$ the centered data.

In DC scenarios, it is useless to provide nodes with a $D \times q$ orthogonal basis since nodes only hold part of the input vectors and are thus unable to perform

Algorithm 3 AGPCA-DC Emission Procedure

$\mathbf{C}_i \leftarrow \mathbf{Z}_i\mathbf{Z}_i^\top$; $w_i \leftarrow D_i$
$(\mathbf{U}_i, \mathbf{L}_i) \leftarrow$ *eigendecompose*$(\mathbf{C}_i)$
$\mathbf{V}_i \leftarrow \mathbf{Z}_i^\top \mathbf{U}_i \mathbf{L}_i^{-\frac{1}{2}}$
loop
  $j \leftarrow$ *randomNeighbor*(i)
  $(\mathbf{V}_i, \mathbf{L}_i, w_i) \leftarrow \frac{1}{2}(\mathbf{V}_i, \mathbf{L}_i, w_i)$
  Send $(\mathbf{V}_i, \mathbf{L}_i, w_i)$ to j
end loop

Algorithm 4 AGPCA-DC Reception Procedure

loop
  Upon receipt of $(\mathbf{V}_j, \mathbf{L}_j, w_j)$
  $w_i \leftarrow w_i + w_j$
  $\mathbf{Q}_0 \leftarrow \mathbf{V}_i$
  for $t \in [0, max\_t[$ **do**
    $(\mathbf{Q}_{t+1}, \mathbf{R}_{t+1}) \leftarrow$ *QR*$(\mathbf{V}_i\mathbf{L}_i\mathbf{V}_i^\top\mathbf{Q}_t + \mathbf{V}_j\mathbf{L}_j\mathbf{V}_j^\top\mathbf{Q}_t)$
  end for
  $\mathbf{V}_i \leftarrow \mathbf{Q}_{max\_t}$; $\mathbf{L}_i \leftarrow$ *diag*$(\mathbf{R}_{max\_t})$
end loop

the projection. Instead, AGPCA directly provides a compressed representation $\mathbf{Y} \equiv \mathbf{U}^{\star\top}\tilde{\mathbf{Z}}$ that account for all dimensions of all observations.

Using the same ideas that allowed us to improve the efficiency in the DS case, we rely on the duality between the covariance and the Gram eigendecompositions. Recall that

$$\mathbf{C} = \frac{1}{n}\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^\top = \mathbf{U}^\star\mathbf{L}^\star\mathbf{U}^{\star\top} \tag{17}$$

Introducing the SVD of the centered data $\tilde{\mathbf{Z}} = \mathbf{U}\mathbf{L}^{\frac{1}{2}}\mathbf{V}^\top$, where $\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^\top = \mathbf{U}\mathbf{L}\mathbf{U}^\top$ and $\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}} = \mathbf{V}\mathbf{L}\mathbf{V}^\top$, we have $\mathbf{U}^\star = \mathbf{U}$ and $\mathbf{L}^\star = \mathbf{L}/n$. Consequently,

$$\mathbf{Y} = \mathbf{U}^{\star\top}\tilde{\mathbf{Z}} = n\mathbf{L}^{\star\frac{1}{2}}\mathbf{V}^\top \tag{18}$$

Define $\mathbf{D} = \tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}}/D$.

Clearly its eigendecomposition is $\mathbf{V}(\mathbf{L}/D)\mathbf{V}^\top = \mathbf{V}(n\mathbf{L}^\star/D)\mathbf{V}^\top$, and

$$\begin{aligned}\mathbf{D} &= \frac{1}{D}(\mathbf{Z} - \frac{1}{n}\mathbf{Z}\mathbf{1}\mathbf{1}^\top)^\top(\mathbf{Z} - \frac{1}{n}\mathbf{Z}\mathbf{1}\mathbf{1}^\top) \\ &= \left(I - \frac{\mathbf{1}\mathbf{1}^\top}{n}\right)\frac{\mathbf{Z}^\top\mathbf{Z}}{D}\left(I - \frac{\mathbf{1}\mathbf{1}^\top}{n}\right) \end{aligned} \tag{19}$$

Observing that Equation 19 only involves $\mathbf{Z}^\top\mathbf{Z}/D$, we remark that $\mathbf{Z}^\top\mathbf{Z}/D = (\sum_i \mathbf{Z}_i^\top\mathbf{Z}_i)/(\sum_i D_i)$ is a weighted average of the locally-computable uncentered Gram matrices $\mathbf{Z}_i^\top\mathbf{Z}_i$. This average can be computed using the same protocol as for the DS case, defined by Equations (12-13). Using the factors $\mathbf{V}$ and $\mathbf{L}/D$ of $\mathbf{D}$ in Equation (18), we can obtain the compressed representations scaled by $\sqrt{D}$. For most applications, this scaling factor has no impact. In applications for which the output basis needs to be orthonormal (and not only orthogonal), the scale can be recovered by gossiping D_i with an initial weight on 1, provided the size of the network is known at each node.

4.1. Early-PCA aggregation in DC scenarios

Clearly, the Early-PCA aggregation scheme described in Section 3 also applies: We only need to compute and exchange the eigendecomposed forms $\mathbf{Z}_i^\top\mathbf{Z}_i = \mathbf{V}_i\mathbf{L}_i\mathbf{V}_i^\top$. Since local Gram matrices are $n \times n$ and are much larger than the local correlation matrices (of size $D_i \times D_i \ll n \times n$), we use the same dual property as in Section 3 to obtain the Gram eigendecomposition from the local correlation matrices $\mathbf{Z}_i\mathbf{Z}_i^\top = \mathbf{U}_i\mathbf{L}_i\mathbf{U}_i^\top$:

$$\mathbf{Z}_i = \mathbf{U}_i\mathbf{L}_i^{\frac{1}{2}}\mathbf{V}_i^\top \quad \Rightarrow \quad \mathbf{V}_i = \mathbf{Z}_i^\top\mathbf{U}_i\mathbf{L}_i^{-\frac{1}{2}}$$

The initial $\mathbf{V}_i\mathbf{L}_i\mathbf{V}_i^\top$ can then be obtained at a low storage and computation cost. The corresponding emission and reception procedures are shown in Algorithms 3 and 4.

Once the averaging protocol has provided all nodes with the eigendecomposition of $\mathbf{Z}^\top\mathbf{Z}/D$, we can locally apply Orthogonal Iterations to Equation (19) to obtain the eigendecomposition $\mathbf{V}(\mathbf{L}^\star/D)\mathbf{V}^\top$ of $\mathbf{D}$ at all nodes. Each node finally computes the full compressed data using $\mathbf{Y} = n(\mathbf{L}^\star/D)^{\frac{1}{2}}\mathbf{V}^\top$.

This shows that using our AGPCA strategy, we can efficiently solve distributed PCA for both DS and DC scenarios by just swapping the roles of covariance and Gram matrices. A nice consequence is that theoretical results in one case hold in the other one.

4.2. Projection of subsequent observations

In DC scenarios, projecting future observations is not as straightforward as in the DS case, because new observations generate new entries in *all* nodes. Thus, all nodes have to participate in the projection of new observations. Remark the compressed representation $\mathbf{y}$ of a new observation $\mathbf{z}$ is computed by:

$$\mathbf{y} = n(\mathbf{L}^\star/D)^{-\frac{1}{2}}\mathbf{V}^\top \frac{\mathbf{Z}^\top \mathbf{z}}{D} \tag{20}$$

Observing that $\mathbf{Z}^\top \mathbf{z}/D = \sum_i \mathbf{Z}_i^\top \mathbf{z}_i/D$ is also a weighted average, $\mathbf{y}$ can easily be recovered by simply Gossiping $\mathbf{Z}_i^\top \mathbf{z}_i$ with initial weights D_i and combining the result with the previously obtained factors $\mathbf{V}$ and $\mathbf{L}/D$.

5. Theoretical analysis

In this section, we theoretically prove that AGPCA yields the same solution as a centralized PCA, provided assumptions are made on the rank of input data. We limit our analysis to the DS case, as the same arguments can easily be translated to the DC case by substituting the covariance matrices by the Gram matrices. Noting $\mathbf{X} \in \mathbb{R}^D$ the input data, this result is expressed in the following theorem:

Theorem 1. *If $rank(\mathbf{X}) \leq q$, then AGPCA yields the exact solution of a PCA run over $\mathbf{X}$:*

$$\forall i, \mathbf{U}_i^\top \mathbf{X}_i = \mathbf{U}^{\star\top} \mathbf{X}_i \tag{21}$$

PROOF. The centralized PCA uses the following eigendecomposition:

$$\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top - \mu\mu^\top = \mathbf{U}^\star \mathbf{L}^\star \mathbf{U}^{\star\top} \tag{22}$$

The proof sketch is as follows: We first show that the local decompositions $\mathbf{X}_i\mathbf{X}_i^\top = \mathbf{U}_i\mathbf{L}_i\mathbf{U}_i^\top$ perfectly reconstruct the local data $\mathbf{X}_i$. Then, we show our Gossip protocol aggregating the $\mathbf{U}_i$ allows for a perfect reconstruction of $\sum_i \mathbf{X}_i\mathbf{X}_i^\top$ at every node. The proof is completed by remarking AGPCA provides the eigendecomposition of the covariance matrices obtained from the reconstruction of $\sum_i \mathbf{X}_i\mathbf{X}_i^\top$.

To prove local factorizations induce no loss of information, we use the assumption that $rank(X) \leq q$. We thus have:

$$rank(\mathbf{X}) \leq q \Rightarrow rank(\mathbf{X}\mathbf{X}^T) \leq q \tag{23}$$

Let us denote $\mathcal{H}_q = span(\mathbf{X}) \subset \mathbb{R}^q$. Since at any node i, $\mathbf{X}_i$ is a subset of $\mathbf{X}$, we have $span(\mathbf{X}_i) \subseteq \mathcal{H}_q$, and consequently $rank(\mathbf{X}_i\mathbf{X}_i^\top) \leq q$. It follows that the initial local decompositions keeping only the q leading eigenvectors (the others being associated with null eigenvalues) perfectly reconstruct the local correlation matrices:

$$\forall i, \mathbf{U}_i\mathbf{L}_i\mathbf{U}_i^\top = \mathbf{X}_i\mathbf{X}_i^\top = \mathbf{B}_i \tag{24}$$

Considering the Gossip averaging, we first prove our update rules in Equations (12-13) are equivalent to the Gossip updates in Equations (6-7). Remark that $\mathbf{a}_s$ and w_s are explicitly updated using Equation (6-7). Checking validity of the sender update of $\mathbf{B}_s$ (line 6 in Algorithm 1) is trivial:

$$\mathbf{B}_s(t+1) = \mathbf{U}_s(t)\frac{1}{2}\mathbf{L}_s(t)\mathbf{U}_s(t)^T = \frac{1}{2}\mathbf{B}_s(t).$$

On the receiver side, provided that $span(\mathbf{U}_s(t)) \subseteq \mathcal{H}_q$ and $span(\mathbf{U}_r(t)) \subseteq \mathcal{H}_q$, using Othogonal Iterations to obtain the q leading eigenvectors of the weighted sum has two interesting properties: The reconstruction is perfect, that is

$$\mathbf{U}_r(t{+}1)\mathbf{L}_r(t{+}1)\mathbf{U}_r(t{+}1) = \mathbf{U}_r(t)\mathbf{L}_r(t)\mathbf{U}_r(t)^\top + \frac{1}{2}\mathbf{U}_s(t)\mathbf{L}_s(t)\mathbf{U}_s(t)^\top \tag{25}$$

and the obtained basis is in the same subspace, *i.e.*, $\mathbf{U}_r(t{+}1) \subseteq \mathcal{H}_q$. These properties come from the fact that $span(\mathbf{U}_r \cup \mathbf{U}_s) \subseteq \mathcal{H}_q$, and thus $rank(\mathbf{U}_r(t)\mathbf{L}_r(t)\mathbf{U}_r(t)^\top + \mathbf{U}_s(t)\mathbf{L}_s(t)\mathbf{U}_s(t)^\top/2) \leq q$ Remarking that initially $\forall i, span(\mathbf{U}_i(0)) \subset \mathcal{H}_q$, it follows by induction that at any time t:

$$\begin{aligned}
\mathbf{B}_r(t+1) &= \mathbf{U}_r(t{+}1)\mathbf{L}_r(t{+}1)\mathbf{U}_r(t{+}1) \\
&= \mathbf{U}_r(t)\mathbf{L}_r(t)\mathbf{U}_r(t)^\top + \frac{1}{2}\mathbf{U}_s(t)\mathbf{L}_s(t)\mathbf{U}_s(t)^\top \\
&= \mathbf{B}_r(t) + \frac{1}{2}\mathbf{B}_s(t)
\end{aligned}$$

That is, Equation (13) is equivalent to Equation (7).

Convergence of Equations (6-7) to the network average at all nodes has already been proved [16, 17]. For completeness though, we show that Equations (6-7) drive all $\mathbf{C}_i$ to $\mathbf{C}$. Let us introduce the Euclidean errors E_i between covariances

locally reconstructed by our protocol and the exact covariance matrix:

$$\begin{aligned}\forall i, \quad E_i(t) &= \|\mathbf{C}_i(t) - \mathbf{C}\|_F^2 \\ &= \left\| \frac{1}{w_i(t)}\mathbf{B}_i(t) - \frac{1}{w_i(t)^2}\mathbf{a}_i(t)\mathbf{a}_i(t)^\top - \frac{1}{n}\mathbf{X}\mathbf{X}^\top + \mu\mu^\top \right\|_F^2 \\ &\leq \left\| \frac{1}{w_i(t)}\mathbf{B}_i(t) - \frac{1}{n}\mathbf{X}\mathbf{X}^\top \right\|_F^2 + \left\| \frac{1}{w_i(t)^2}\mathbf{a}_i(t)\mathbf{a}_i(t)^\top - \mu\mu^\top \right\|_F^2\end{aligned}$$

From Equations (9), and (3-4), we have $n = \sum_i w_i(0)$, $\mathbf{X}\mathbf{X}^\top = \sum_i \mathbf{B}_i(0)$ and $\mu = (\sum_i \mathbf{a}_i(0))/(\sum_i w_i(0))$. Then, we obtain

$$E_i(t) \leq \left\| \frac{\mathbf{B}_i(t)}{w_i(t)} - \frac{\sum_j \mathbf{B}_j(0)}{\sum_j w_j(0)} \right\|_F^2 + \left\| \frac{\mathbf{a}_i(t)\mathbf{a}_i(t)^\top}{w_i(t)^2} - \frac{(\sum_j \mathbf{a}_j(0))(\sum_j \mathbf{a}_j(0))^\top}{(\sum_j w_j(0))^2} \right\|_F^2 \tag{26}$$

According to Equations (6-7), each single entry of $(\mathbf{B}_i, \mathbf{a}_i, w_i)$ is identically updated. The following Lemma then shows that every quotient of $\mathbf{B}_i$ or $\mathbf{a}_i$ and w_i tends to its network average (the proof is deferred to Appendix A):

Lemma 2. *Under update rules in Equations (6-7):*

$$\forall i, \quad \lim_{t\to\infty} \frac{\mathbf{B}_i(t)}{w_i(t)} = \frac{\sum_j \mathbf{B}_j(0)}{\sum_j w_j(0)} \quad \textit{and} \quad \lim_{t\to\infty} \frac{\mathbf{a}_i(t)}{w_i(t)} = \frac{\sum_j \mathbf{a}_j(0)}{\sum_j w_j(0)} \tag{27}$$

In turn, it implies that both terms in Equation (26) tend to zero, and therefore entails the convergence of local estimates to the covariance matrix:

$$\lim_{t\to\infty} \mathbf{C}_i(t) = \mathbf{C} \tag{28}$$

□

6. Experiments

In this section, we experimentally evaluate AGPCA, both on synthetic and natural data. Experiments for Distributed Samples and Distributed Coordinates scenarios are reported separately. Synthetic data were obtained by sampling n observations from a D-dimensional Gaussian distribution, which covariance matrix

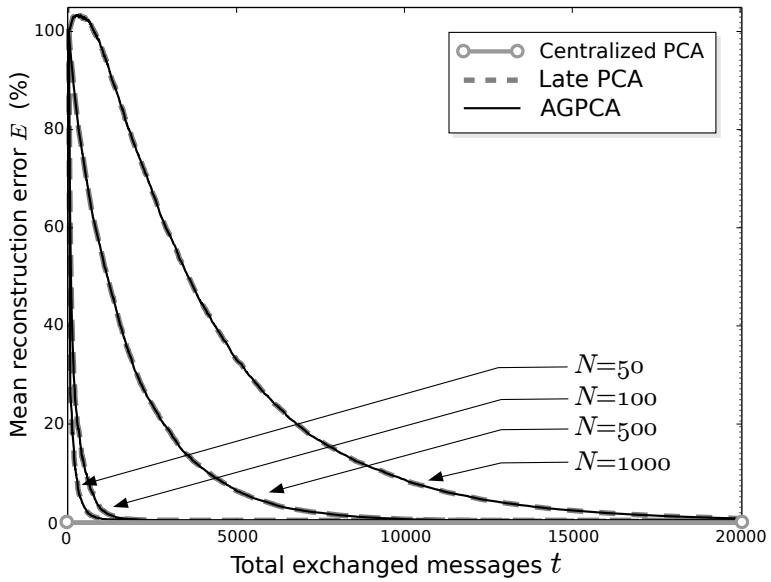


Figure 1: Convergence of AGPCA compared to a Late PCA strategy and the centralized PCA solution on a synthetic dataset drawn from a Gaussian distribution ($D = 200, p = 30, n = 10000$), spread on $N = 100$ nodes. Here q is set to D.

was arbitrarily generated with rank $p \ll D$ and such that all dimensions are correlated in $\mathbb{R}^D$. For natural data, we used the MNIST handwritten digits dataset, which contains 60000 grayscale images of 28×28 pixels, that is, $D = 784$ [20]. Unless specified, the network is assumed fully connected, sender-receiver pairs being uniformly selected at random.

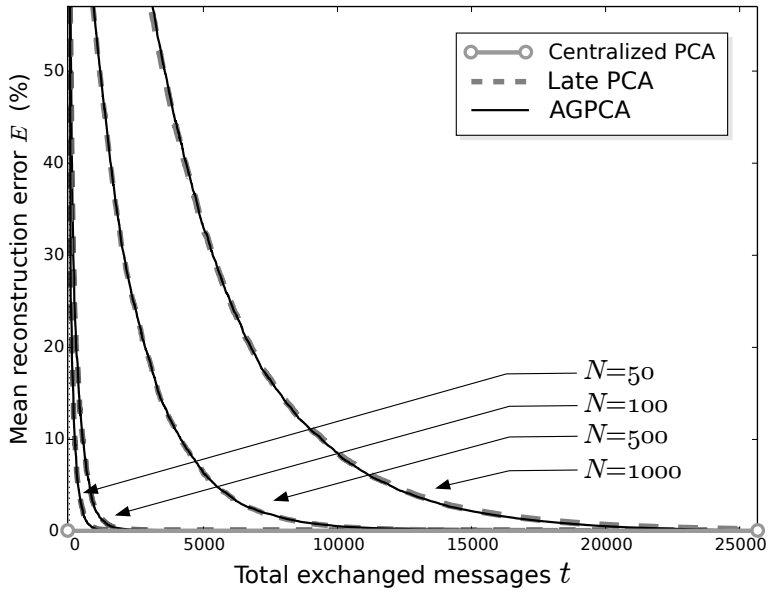


Figure 2: Convergence of AGPCA on the same data and network as Figure 1, but setting $q = 30 = p \ll D$.

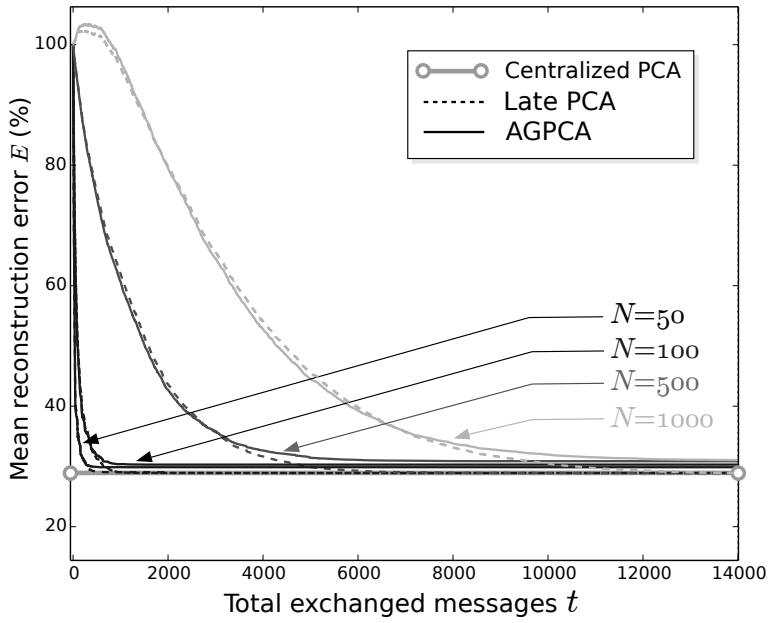


Figure 3: Convergence of AGPCA on the same data and network as Figure 1, but setting $q = 10 < p$.

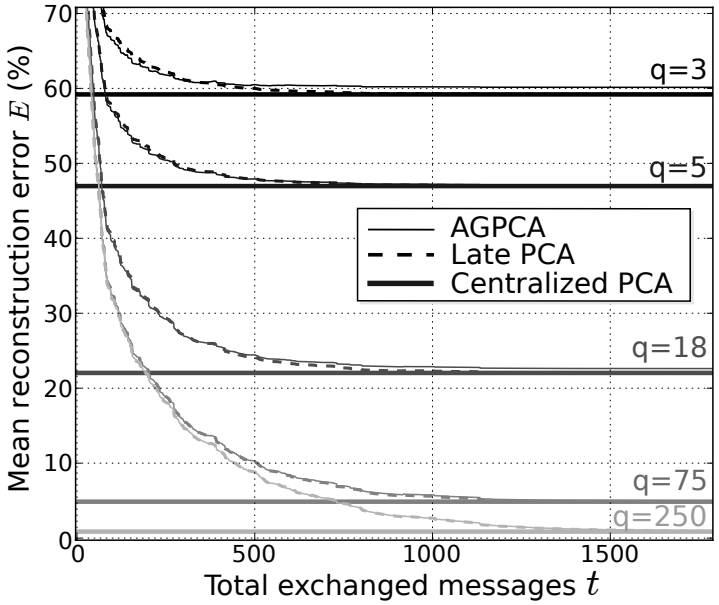


Figure 4: Convergence of AGPCA on MNIST spread over $N = 100$ nodes, for various q.

6.1. Distributed Samples scenario

In DS scenario, the objective of AGPCA is two-fold : *(i)* providing all nodes with the same projection matrix $\mathbf{U}$, *(ii)* making $\mathbf{U}$ as close as possible to the solution $\mathbf{U}^\star$ of a centralized PCA solution computed over the aggregated network data $\mathbf{X}$. We thus measure the average Euclidean reconstruction error between the locally reconstructed covariances $\mathbf{C}_i$ and the exact $\mathbf{C}$, defined by $E = \|\mathbf{C}_i(\infty) - \mathbf{C}\|_F^2 / \|\mathbf{C}\|_F^2$. We considered three cases: $q = D$, $q = p$, and $q < p$:

- When $q = D$, the centralized PCA solution $\mathbf{U}^\star$ reconstructs the covariance with zero error. In this case, experiments on synthetic data show that AGPCA asymptotically provides all nodes with the exact $\mathbf{U}^\star$. Figure 1 illustrates the convergence to this optimum versus time and show that both Early and Late PCA strategies allow a perfect reconstruction.

- When $q = p \ll D$, Figure 2 confirms the theoretical result of Theorem 1, as convergence to the optimum still holds if the intrinsic dimension p of $\mathbf{X}$ is lower than q.

- By contrast, when $q < p$, the ideal projection itself accounts for less than 100% of the variance in the data. In such case, AGPCA still guarantees

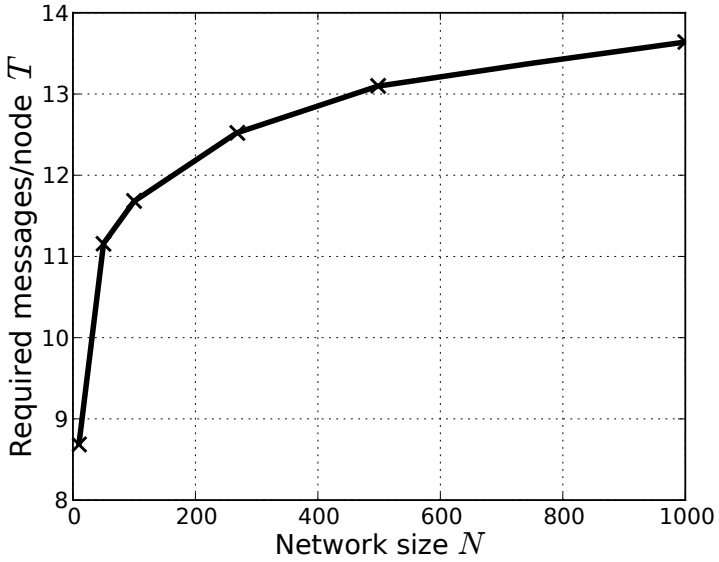


Figure 5: Average number of message emissions per node required to converge against the number of nodes in the network, evaluated on MNIST with $q = 50$.

that all nodes converge to the same projection matrix $\mathbf{U}$. This ensures that any observation is projected into the same output space whichever the node we consider. However, the error induced by the Orthogonal Iterations steps leads to a slight deviation from the optimal global PCA solution $\mathbf{U}^\star$. Experiments reported in Figure 3 for various q reveal that even for high compression rates, the deviation from the optimal basis is still very low.

Similar experiments were conducted on the MNIST dataset, whose intrinsic dimensionality was shown in previous works to be much lower than $D = 784$. Figure 4 highlights that AGPCA achieves the same reconstruction error as a centralized PCA up to the numerical precision for $q \geq 75$. For lower values, it is slightly suboptimal, but the deviation is kept under 1% until $q = 3$, and always under 2%. Figure 4 also shows that the convergence rate of the aggregation protocol is not impacted by switching from a Late PCA to an Early PCA scheme. Interestingly, lower values for q bring faster convergence.

Concerning communication costs, the efficiency of AGPCA is illustrated in Figure 5. This figure displays the number of messages each node has to emit in order to reach convergence (convergence is assumed when E improves by less than 0.01% between two message events). This number of messages per node appears to logarithmically scale with the number of nodes in the network. This

ensures easy scaling to large networks.

6.2. Distributed Coordinates scenario

In a DC scenario, our accuracy criterion is still the Euclidean reconstruction error, but it is now computed as $E = \|\mathbf{Y}_i^T\mathbf{Y}_i - \mathbf{X}^T\mathbf{X}\|$. To evaluate AGPCA in such scenarios, we spread a synthetic dataset similar to Figure 1 by assigning a uniformly drawn number of dimensions D_i to each node i, such that node 1 gets the D_1 first dimensions, node 2 gets the next D_2 and so forth, and such that $\sum_i D_i = D = 10000$. In this scenario, Figure 6 shows that AGPCA behaves identically to the DS case, which is a logical consequence of using the same strategy.

6.3. Influence of network connectivity

Network topology usually has a great impact on speed, accuracy or even applicability of distributed learning algorithms. Gossip protocols have long been acclaimed for their fine behavior in a wide range of networking situations. It is worth highlighting that the Sum-Weight protocol used in AGPCA can be used to build asynchronous generalizations of broadcast, spanning tree, workers-master and scale-free connectivity:

- In an asynchronous broadcast scheme, senders still waken independently, but the same message is sent to all their neighbors. To obtain a broadcast protocol, we just need to replace the $1/2$ coefficient in (6-7) by $1/N$ to respect mass conservation.

- In an asynchronous spanning tree communication scheme, the network has only $N-1$ links, which is the minimal number of links such that the network stays connected. In our experiments, we used the worst conditioned non-degenerate tree topology where nodes have at most 3 neighbors.

- In a workers-master scheme, $N-1$ nodes called workers hold local datasets and can only send messages to a single master node. This master is allowed to send messages to all workers, and may hold a local sample or not (if not, its estimates are initialized to 0). In our experiments, we considered two settings. In the first one, the workers and the master are selected for emission with the same probability, *i.e.*, the master works at the same frequency as the workers. In the second one, the master is selected N times more often than workers, *e.g.*, assuming it is a high-throughput server.

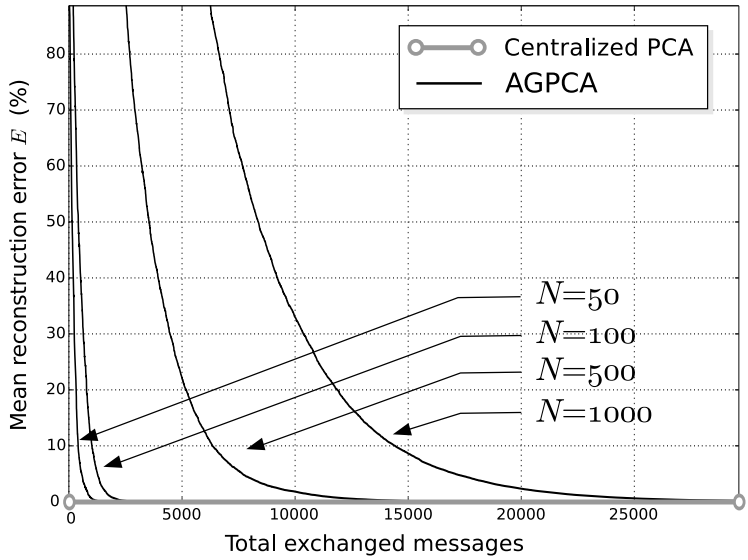


Figure 6: Convergence of AGPCA in a Distributed Coordinates scenario on synthetic data similar to Figure 1 ($D = 10000$, $n = 200, p = 30$). Each node holds a number of dimensions of the complete data drawn uniformly in $\{1, \ldots, \frac{2D}{N} - 1\}$. Here, $q = p = 30$.

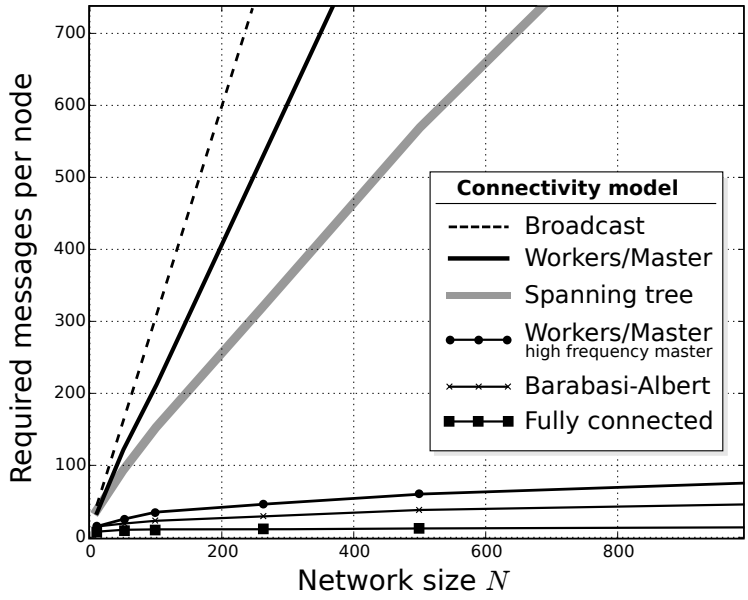


Figure 7: Number of messages per node to reach convergence for various connectivity and networking models. Here, $q = p = 30$ (same data as Figure 6).

- An example of a scale-free network is the Barabási-Albert (BA) random graph model, which has a power-law distribution of node degrees. Nodes are connected to random neighbors such that the number of nodes having k neighbors scales in k^{-3}.

We run AGPCA for each of these connectivity models, varying the number of involved nodes (synthetic data similar to Figure 1). Figure 7 displays the number of messages per node to reach convergence (in the same sense as Figure 5) and compares it to the fully-connected setup we considered so far. Broadcast, workers-master and spanning tree schemes exhibit a poor scaling which appears linear with the number of nodes. By contrast, full and BA connectivity show a logarithmic dependence of the communication costs on the network size. In the workers-master case, we can recover a logarithmic scaling by allowing the master to communicate N-times faster than the workers (in practice, such performances at the master can be unrealistic).

The poor behavior of the broadcast scheme can be explained by remarking that senders always emit $N - 1$ messages containing *identical* estimates. In the point-to-point case each message integrates the contribution of *all* preceding updates, thus making estimates mixing much faster. Concerning the spanning tree topology, its bad mixing properties have long been studied in the Gossip protocols and rumor spreading litterature. Due to its high average path length, information must flow, on average, through a much larger number of intermediary nodes to transit between any two peers.

To summarize, AGPCA, like asynchronous Gossip protocols in general, are best-suited for networks with maximally randomized communications and lowest average path lengths.

7. Conclusion

We presented an asynchronous and decentralized algorithm to solve PCA when data is spread over a network. Based on the integration of a dimensionality reduction operator into a Sum-Weight gossip averaging protocol, it is best suited for large setups with high extrinsic dimension, massive samples and large networks. Unlike other algorithms, it is applicable both in Distributed Samples and Distributed Coordinates scenarios, thanks to the duality between covariance and Gram matrices decompositions. Our theoretical and experimental studies show that it is formally equivalent to running a traditional PCA when the complete data has an intrinsic dimension lower than the output dimension, otherwise providing a low-error approximation of the optimum.

Perspectives include application to large-scale dimension reduction problems, where traditional methods are unapplicable, such as signatures compression in web-scale multimedia retrieval. Besides, while we considered a static scenario where data is provided all at once, Gossip protocols also enjoy nice dynamics when data and/or connectivity are time-evolving. Our approach could then be extended to deal with time-related phenomena, such as concept drift and dynamic networks.

Acknowledgements

This work is funded by the Culture 3D Cloud project as part of the french Funds for Digital Societies.

References

[1] K. Pearson, Liii. on lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2 (11) (1901) 559–572.

[2] H. Hotelling, Analysis of a complex of statistical variables into principal components., Journal of educational psychology 24 (6) (1933) 417.

[3] C. M. Bishop, et al., Pattern recognition and machine learning, Vol. 1, springer New York, 2006.

[4] A. Bertrand, M. Moonen, Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed pca, Signal Processing 104 (2014) 120–135.

[5] J. Fellus, D. Picard, P.-H. Gosselin, et al., Dimensionality reduction in decentralized networks by gossip aggregation of principal components analyzers, in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2013, pp. 171–176.

[6] D. Kempe, F. McSherry, A decentralized algorithm for spectral analysis, Journal of Computer and System Sciences 74 (1) (2008) 70–83.

[7] A. Wiesel, A. O. Hero, Decomposable principal component analysis, Signal Processing, IEEE Transactions on 57 (11) (2009) 4369–4377.

[8] Z. Meng, A. Wiesel, A. O. Hero, Distributed principal component analysis on networks via directed graphical models, in: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012, pp. 2877–2880.

[9] C. Ordonez, N. Mohanam, C. Garcia-Alvarado, Pca for large data sets with parallel data summarization, Distributed and Parallel Databases 32 (3) (2014) 377–403.

[10] S. B. Korada, A. Montanari, S. Oh, Gossip PCA, in: Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, ACM, 2011, pp. 209–220.

[11] P. Bruneau, M. Gelgon, F. Picarougne, Aggregation of probabilistic PCA mixtures with a variational-bayes technique over parameters, in: Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE, 2010, pp. 702–705.

[12] M. E. Tipping, C. M. Bishop, Mixtures of probabilistic principal component analyzers, Neural computation 11 (2) (1999) 443–482.

[13] A. Nikseresht, M. Gelgon, Gossip-based computation of a gaussian mixture model for distributed multimedia indexing, Multimedia, IEEE Transactions on 10 (3) (2008) 385–392.

[14] A. Nikseresht, Estimation de modèles de mélange probabilistes: une proposition pour un fonctionnement réparti et décentralisé, Ph.D. thesis, Université de Nantes (2008).

[15] S. V. Macua, P. Belanovic, S. Zazo, Consensus-based distributed principal component analysis in wireless sensor networks, in: Signal Processing Advances in Wireless Communications (SPAWC), 2010 IEEE Eleventh International Workshop on, IEEE, 2010, pp. 1–5.

[16] D. Kempe, A. Dobra, J. Gehrke, Gossip-based computation of aggregate information, in: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS '03, IEEE Computer Society, Washington, DC, USA, 2003, pp. 482–.

[17] F. Iutzeler, P. Ciblat, W. Hachem, Analysis of sum-weight-like algorithms for averaging in wireless sensor networks, CoRR abs/1209.5912.

[18] M. Jelasity, W. Kowalczyk, M. Van Steen, Newscast computing, Tech. rep., Technical Report IR-CS-006, Vrije Universiteit Amsterdam, Department of Computer Science, Amsterdam, The Netherlands (2003).

[19] D. Shah, Gossip algorithms, Now Publishers Inc, 2009.

[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[21] A. Rhodius, On the maximum of ergodicity coefficients, the dobrushin ergodicity coefficient, and products of stochastic matrices, Linear algebra and its applications 253 (1) (1997) 141–154.

[22] E. Seneta, Non-negative matrices and Markov chains, Springer, 2006.

Appendix A. Proof of Lemma 2

PROOF. Let's independently consider any entry of $\mathbf{B}_i$ or $\mathbf{a}_i$, denoted using the free variable x_i. Let $\mathbf{x} = (x_i)_i$ gather all the corresponding entries from each node i. Using the definition of $\mathbf{x}$, we can rewrite Equations (6-7) into a matrix form:

$$\mathbf{x}(t+1)^\top = \mathbf{x}(t)^\top \mathbf{K}(t) \quad \text{where} \quad \mathbf{K}(t) \equiv \mathbf{I} + \frac{1}{2}\mathbf{e}_s(\mathbf{e}_r - \mathbf{e}_s)^\top \tag{A.1}$$

$\mathbf{K}(t)$ is a random $N \times N$ matrix where (s, r) is the selected sender-receiver pair at time t. We can then write $\mathbf{x}(t)^\top = \mathbf{x}(0)^\top \mathbf{P}(t)$, where $\mathbf{P}(t) \equiv \prod_{u \leq t} \mathbf{K}(u)$. Observe that $\forall t, \mathbf{K}(t)^\top \mathbf{1} = \mathbf{1}$ and $\mathbf{P}(t)^\top \mathbf{1} = \mathbf{1}$, meaning that $\mathbf{K}(t)$ and $\mathbf{P}(t)$ are row-stochastic. A row-stochastic matrix that is irreducible is said to be scrambling (see [21]). Establishing irreducibility of $\mathbf{K}$ amounts to check if the network graph is strongly connected, *i.e.*, for any two nodes (i, j) there exists a route from i to j through available connections. Assuming bidirectional communication is allowed between any pair of connected nodes, the only constraint to ensure $\mathbf{K}$ is irreducible is that the network graph is connected, which is a logical assumption in the distributed computing setup we consider. Consequently, $\mathbf{K}$ is a scrambling matrix. Scrambling matrices are shown to be weakly ergodic [22], that is, the

product of t realizations tends to have identical rows as t grows. This can be proved by introducing the coefficient of ergodicity $\tau(\cdot)$ defined by

$$\forall \mathbf{A} \in \mathbb{R}^{N\times N}, \quad \tau(\mathbf{A}) = \frac{1}{2}\max_{i,j}\|\mathbf{A}^T(\mathbf{e}_i - \mathbf{e}_j)\|_1$$

In words, $\tau(\mathbf{A})$ corresponds to the maximum $\mathcal{L}_1$ distance between any two rows of $\mathbf{A}$. $\mathbf{A}$ has equal rows if and only if $\tau(\mathbf{A}) = 0$. When $\mathbf{P}(t)$ is the forward product of t realizations of a scrambling random matrix $\mathbf{K}$, [22] states

$$\tau(\mathbf{P}(t+1)) = \tau(\mathbf{P}(t)\mathbf{K}(t+1)) \leq \tau(\mathbf{P}(t))\lambda_2(t),$$

where $\lambda_2(t)$ is the second largest eigenvalue of $\mathbf{K}(t+1)$. Then, $\tau(\mathbf{P}(t)) \leq \lambda_2^t$, where λ_2 is the maximal second largest eigenvalue over all realizations of $\mathbf{K}$. $\mathbf{K}(t)$ being row-stochastic, we have $0 < \lambda_2 < 1$. Hence,

$$\lim_{t\to\infty} \tau(\mathbf{P}(t)) = 0$$

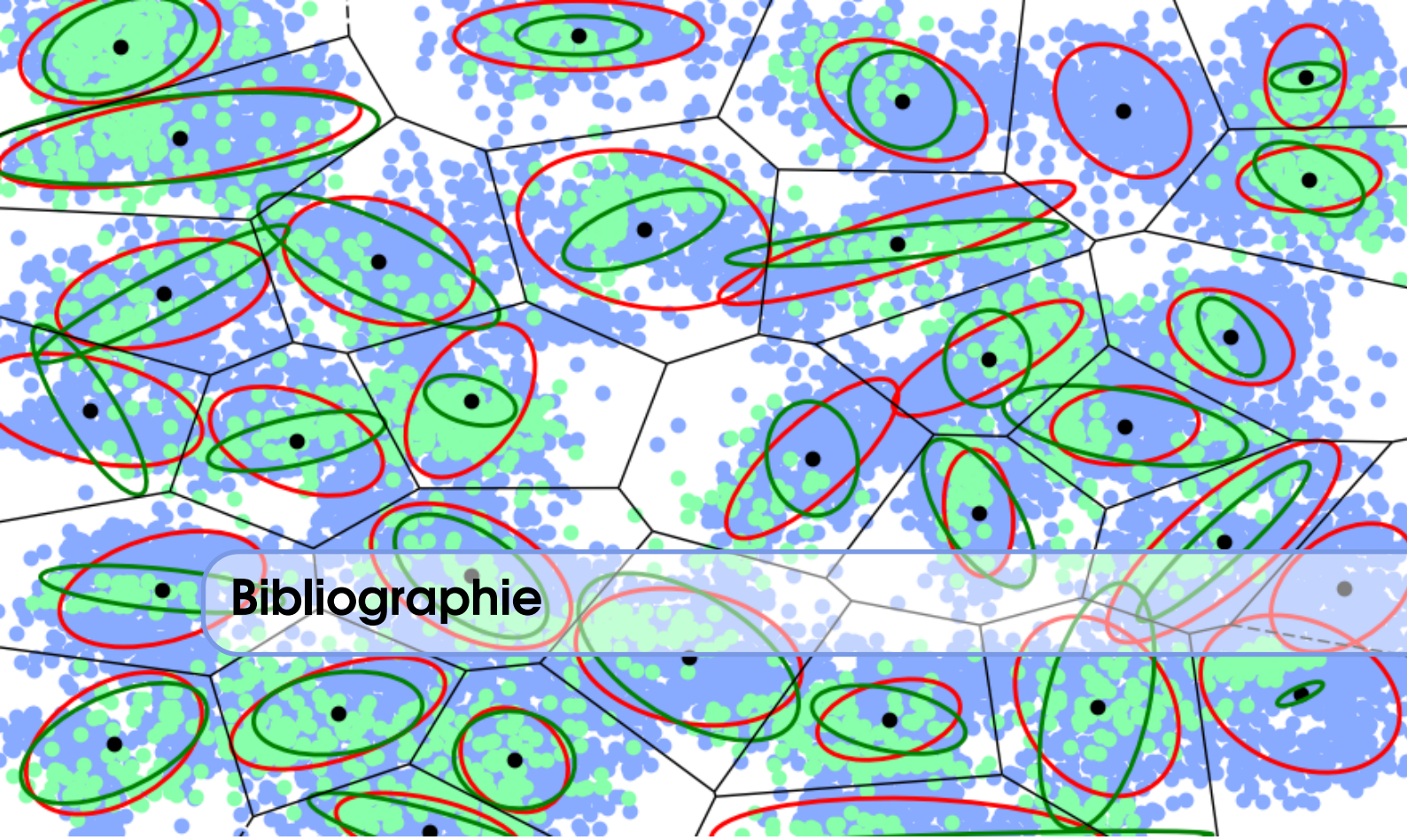
Recalling that $\mathbf{x}^\top(t) = \mathbf{x}^\top(0)\mathbf{P}(t)$, if $\mathbf{P}(t)$ has equal rows we get

$$\forall i, \quad \frac{x_i(t)}{w_i(t)} = \frac{\sum_j x_j(0)\mathbf{P}_{ji}(t)}{\sum_j w_j(0)\mathbf{P}_{ji}(t)} = \frac{\mathbf{P}_{1,i}(t)\sum_j x_j(0)}{\mathbf{P}_{1,i}(t)\sum_j w_j(0)}$$

Simplifying $\mathbf{P}_{1i}(t)$ yields our final result:

$$\forall i, \quad \lim_{t\to\infty} \frac{x_i(t)}{w_i(t)} = \frac{\sum_j x_j(0)}{\sum_j w_j(0)}$$

□



Bibliographie

Bibliographie générale

[1] Xingyu LIU, Song HAN, Huizi MAO et William J DALLY. “Efficient Sparse-Winograd Convolutional Neural Networks”. In : (2017) (cf. page 49).

[2] Diogo Carbonera LUVIZON, Hedi TABIA et David PICARD. “Learning features combination for human action recognition from skeleton sequences”. In : *Pattern Recognition Letters* (2017) (cf. page 31).

[3] Mark SCHMIDT, Nicolas LE ROUX et Francis BACH. “Minimizing finite sums with the stochastic average gradient”. In : *Mathematical Programming* 162.1-2 (2017), pages 83–112 (cf. page 38).

[4] Relja ARANDJELOVIC et al. “NetVLAD : CNN architecture for weakly supervised place recognition”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pages 5297–5307 (cf. pages 42, 49).

[5] Micael CARVALHO et al. “Deep neural networks under stress”. In : *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE. 2016, pages 4443–4447 (cf. page 49).

[6] Gao HUANG et al. “Deep networks with stochastic depth”. In : *European Conference on Computer Vision*. Springer. 2016, pages 646–661 (cf. page 50).

[7] Olivier KIHL, David PICARD et Philippe-Henri GOSSELIN. “Local polynomial space–time descriptors for action classification”. In : *Machine Vision and Applications* 27.3 (2016), pages 351–361 (cf. page 28).

[8] Meng LI et Howard LEUNG. “Graph-based approach for 3D human skeletal action recognition”. In : *Pattern Recognition Letters* (2016), pages -. ISSN : 0167-8655. DOI : `http://dx.doi.org/10.1016/j.patrec.2016.07.021` (cf. page 32).

[9] David PICARD. “Preserving local spatial information in image similarity using tensor aggregation of local features”. In : *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE. 2016, pages 201–205 (cf. pages 23, 48).

[10] David PICARD, Thomas HENN et Georg DIETZ. "Non-negative dictionary learning for paper watermark similarity". In : *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE. 2016, pages 130–133 (cf. page 44).

[11] H. RAHMANI, A. MAHMOOD, D. HUYNH et A. MIAN. "Histogram of Oriented Principal Components for Cross-View Action Recognition". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* PP.99 (2016), pages 1–1. ISSN : 0162-8828. DOI : 10.1109/TPAMI.2016.2533389 (cf. page 32).

[12] Thibaut DURAND, Nicolas THOME et Matthieu CORD. "MANTRA : Minimum Maximum Latent Structural SVM for Image Classification and Ranking". In : *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pages 2713–2721 (cf. page 49).

[13] Jerome FELLUS, David PICARD et Philippe-Henri GOSSELIN. "Asynchronous gossip principal components analysis". In : *Neurocomputing* 169 (2015), pages 262–271 (cf. page 37).

[14] Jérôme FELLUS, David PICARD et Philippe-Henri GOSSELIN. "Indexation multimédia par dictionnaires visuels en environnement décentralisé. Une approche par protocoles Gossip." In : *Traitement du Signal* 32.1 (2015), pages 39–64 (cf. page 36).

[15] Yani IOANNOU et al. "Training cnns with low-rank filters for efficient image classification". In : *arXiv preprint arXiv :1511.06744* (2015) (cf. page 49).

[16] Olivier KIHL, David PICARD et Philippe-Henri GOSSELIN. "A unified framework for local visual descriptors evaluation". In : *Pattern Recognition* 48.4 (2015), pages 1174–1184 (cf. pages 27, 28).

[17] David PICARD, Philippe-Henri GOSSELIN et Marie-Claude GASPARD. "Challenges in Content-Based Image Indexing of Cultural Heritage Collections". In : *Signal Processing Magazine, IEEE* 32.4 (2015), pages 95–102 (cf. page 42).

[18] Vivek VEERIAH, Naifan ZHUANG et Guo-Jun QI. "Differential Recurrent Neural Networks for Action Recognition". In : *IEEE International Conference on Computer Vision (ICCV)*. Déc. 2015 (cf. page 32).

[19] Sixin ZHANG, Anna E CHOROMANSKA et Yann LECUN. "Deep learning with elastic averaging SGD". In : *Advances in Neural Information Processing Systems*. 2015, pages 685–693 (cf. page 40).

[20] Maxime DEVANNE et al. "3D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold". In : *IEEE Transactions on Cybernetics* (août 2014) (cf. page 32).

[21] Jerome FELLUS, David PICARD et Philippe-Henri GOSSELIN. "Dimensionality reduction in decentralized networks by Gossip aggregation of principal components analyzers". In : *ESANN 2014*. 2014, pages 171–176 (cf. page 37).

[22] C. LU, J. JIA et C. K. TANG. "Range-Sample Depth Feature for Action Recognition". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pages 772–779. DOI : 10.1109/CVPR.2014.104 (cf. page 32).

[23] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. "Dimensionality reduction of visual features using sparse projectors for content-based image retrieval". In : *IEEE Int. Conf. on Image Processing (ICIP)*. 2014, pages 2192–2196 (cf. page 26).

[24] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. "Efficient Metric Learning Based Dimension Reduction Using Sparse Projectors For Image Near Duplicate Retrieval". In : *ICPR*. 2014 (cf. page 26).

[25] David PICARD et Inbar FIJALKOW. “Second order model deviations of local Gabor features for texture classification”. In : *Signals, Systems and Computers, 2014 48th Asilomar Conference on*. IEEE. 2014, pages 917–920 (cf. page 43).

[26] David PICARD, Ngoc-Son VU et Inbar FIJALKOW. “Photographic paper texture classification using model deviation of local visual descriptors”. In : *IEEE Int. Conf. on Image Processing*. 2014, pages 5701–5705 (cf. page 43).

[27] Karen SIMONYAN et Andrew ZISSERMAN. “Very deep convolutional networks for large-scale image recognition”. In : *arXiv preprint arXiv :1409.1556* (2014) (cf. page 49).

[28] Nitish SRIVASTAVA et al. “Dropout : A Simple Way to Prevent Neural Networks from Overfitting”. In : *Journal of Machine Learning Research* 15 (2014), pages 1929–1958. URL : http://jmlr.org/papers/v15/srivastava14a.html (cf. page 50).

[29] R. VEMULAPALLI, F. ARRATE et R. CHELLAPPA. “Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group”. In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pages 588–595. DOI : 10.1109/CVPR.2014.82 (cf. page 32).

[30] X. YANG et Y. TIAN. “Super Normal Vector for Activity Recognition Using Depth Sequences”. In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pages 804–811. DOI : 10.1109/CVPR.2014.108 (cf. page 32).

[31] Jerome FELLUS, David PICARD et Philippe-Henri GOSSELIN. “Decentralized K-means using randomized Gossip protocols for clustering large datasets”. In : *IEEE 13th International Conference on Data Mining Workshops*. IEEE. 2013, pages 599–606 (cf. page 36).

[32] Ian GOODFELLOW et al. “Maxout Networks”. In : *International Conference on Machine Learning*. 2013, pages 1319–1327 (cf. page 50).

[33] Jiajia LUO, Wei WANG et Hairong QI. “Group Sparsity and Geometry Constrained Dictionary Learning for Action Recognition from Depth Maps”. In : *IEEE International Conference on Computer Vision (ICCV)*. 2013, pages 1809–1816. DOI : 10.1109/ICCV.2013.227 (cf. page 32).

[34] Romain NEGREL, David PICARD et Philippe-Henri GOSSELIN. “Web scale image retrieval using compact tensor aggregation of visual descriptors”. In : *IEEE Multimedia* 20.3 (2013), pages 24–33 (cf. pages 19, 24, 48).

[35] O. OREIFEJ et Zicheng LIU. “HON4D : Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences”. In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2013, pages 716–723. DOI : 10.1109/CVPR.2013.98 (cf. page 32).

[36] David PICARD et Philippe-Henri GOSSELIN. “Efficient image signatures and similarities using tensor products of local descriptors”. In : *Computer Vision and Image Understanding* 117.6 (2013), pages 680–687 (cf. pages 22, 48).

[37] David PICARD, Nicolas THOME et Matthieu CORD. “JKernelMachines : A simple framework for Kernel Machines”. In : *Journal of Machine Learning Research* 14.May (2013), pages 1417–1421 (cf. page 29).

[38] L. SEIDENARI et al. “Recognizing Actions from Depth Cameras as Weakly Aligned Multi-part Bag-of-Poses”. In : *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2013, pages 479–485. DOI : 10.1109/CVPRW.2013.77 (cf. page 32).

[39] Q. D. TRAN et N. Q. LY. "An effective fusion scheme of spatio-temporal features for human action recognition in RGB-D video". In : *International Conference on Control, Automation and Information Sciences (ICCAIS)*. 2013, pages 246–251. DOI : 10.1109/ICCAIS.2013.6720562 (cf. page 32).

[40] Hervé JÉGOU, Florent PERRONNIN et al. "Aggregating local image descriptors into compact codes". English. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 1 (2012). QUAERO, pages 3304–3311. URL : http://hal.inria.fr/inria-00633013/en/ (cf. page 48).

[41] Romain NEGREL, David PICARD et P GOSSELIN. "Compact Tensor Based Image Representation for Similarity Search". In : *International Conference on Image Processing*. 2012 (cf. page 22).

[42] Ngoc-Son VU et Alice CAPLIER. "Enhanced patterns of oriented edge magnitudes for face recognition and image matching". In : *Image Processing, IEEE Transactions on* 21.3 (2012), pages 1352–1365 (cf. page 43).

[43] Jiang WANG, Zicheng LIU, Ying WU et Junsong YUAN. "Mining Actionlet Ensemble for Action Recognition with Depth Cameras". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2012, pages 1290–1297. DOI : 10.1109/CVPR.2012.6247813 (cf. page 32).

[44] Lu XIA, Chia-Chih CHEN et J. K. AGGARWAL. "View Invariant Human Action Recognition Using Histograms of 3D Joints." In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pages 20–27. ISBN : 978-1-4673-1611-8 (cf. page 32).

[45] A. GILBERT, J. ILLINGWORTH et R. BOWDEN. "Action recognition using mined hierarchical compound features". In : *Transactions on Pattern Analysis and Machine Intelligence* 99 (2011), pages 883–897 (cf. page 30).

[46] David PICARD et Philippe-Henri GOSSELIN. "Improving Image Similarity With Vectors of Locally Aggregated Tensors". In : *Image Processing (ICIP), 2011 18th IEEE International Conference on*. 2011, pages–669 (cf. pages 22, 48).

[47] Heng WANG, Alexander KLÄSER, Cordelia SCHMID et Liu CHENG-LIN. "Action Recognition by Dense Trajectories". In : *IEEE International Conference on Computer Vision and Pattern Recognition*. 2011 (cf. page 30).

[48] Florence BÉNÉZIT et al. "Weighted gossip : Distributed averaging using non-doubly stochastic matrices". In : *Information theory proceedings (isit), 2010 ieee international symposium on*. IEEE. 2010, pages 1753–1757 (cf. page 34).

[49] H. JÉGOU, M. DOUZE, C. SCHMID et P. PÉREZ. "Aggregating local descriptors into a compact image representation". In : *IEEE International Conference on Computer Vision and Pattern Recognition*. Juin 2010, pages 3304–3311 (cf. page 30).

[50] J. MAIRAL, F. BACH, J. PONCE et G. SAPIRO. "Online Learning for Matrix Factorization and Sparse Coding". In : *International Journal on Machine Learning Research* 11 (2010), pages 19–60 (cf. page 45).

[51] F. PERRONNIN, Y. LIU, J. SANCHEZ et H. POIRIER. "Large-scale image retrieval with compressed Fisher vectors". In : *IEEE International Conference on Computer Vision and Pattern Recognition*. Juin 2010 (cf. page 22).

[52] M.M. ULLAH, S. PARIZI et I. LAPTEV. "Improving bag-of-features action recognition with non-local cues". In : *BMVC*. 2010 (cf. page 30).

[53] K.Q. WEINBERGER et L.K. SAUL. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. In : *The Journal of Machine Learning Research (JMLR)* 10 (2009), pages 207–244 (cf. page 30).

[54] H. BAY, A. ESS, T. TUYTELAARS et L. Van GOOL. “SURF : Speeded Up Robust Features”. In : *Computer Vision and Image Understanding* 110.3 (2008), pages 346–359 (cf. page 27).

[55] Hervé JÉGOU, Matthijs DOUZE et Cordelia SCHMID. “Hamming embedding and weak geometric consistency for large scale image search”. In : *European Conference on Computer Vision*. Sous la direction d’Andrew Zisserman DAVID FORSYTH Philip Torr. Tome I. LNCS. Springer, oct. 2008, pages 304–317. URL : http://lear.inrialpes.fr/pubs/2008/JDS08 (cf. page 20).

[56] N. DALAL, B. TRIGGS et C. SCHMID. “Human detection using oriented histograms of flow and appearance”. In : *ECCV* (2006), pages 428–441 (cf. pages 27, 28).

[57] Svetlana LAZEBNIK, Cordelia SCHMID et Jean PONCE. “Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories”. In : *IEEE International Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA : IEEE Computer Society, 2006, pages 2169–2178. ISBN : 0-7695-2597-0. DOI : http://dx.doi.org/10.1109/CVPR.2006.68 (cf. page 22).

[58] Navneet DALAL et Bill TRIGGS. “Histograms of Oriented Gradients for Human Detection”. In : *IEEE International Conference on Computer Vision and Pattern Recognition*. Sous la direction de Cordelia SCHMID, Stefano SOATTO et Carlo TOMASI. Tome 2. INRIA Rhône-Alpes, ZIRST-655, av. de l’Europe, Montbonnot-38334, juin 2005, pages 886–893. URL : http://lear.inrialpes.fr/pubs/2005/DT05 (cf. page 27).

[59] D. LOWE. “Distinctive image features from scale-invariant keypoints”. In : *International Journal of Computer Vision* 2.60 (2004), pages 91–110 (cf. pages 20, 27).

[60] J. SHAWE-TAYLOR et N. CRISTIANINI. *Kernel methods for Pattern Analysis*. Cambridge University Press, ISBN 0-521-81397-2, 2004 (cf. page 20).

[61] A.A. EFROS, A.C. BERG, G. MORI et J. MALIK. “Recognizing action at a distance”. In : *ICCV*. Tome 2. IEEE, 2003, pages 726–733 (cf. page 28).

[62] J. SIVIC et A. ZISSERMAN. “Video Google : A text retrieval approach to object matching in videos”. In : *IEEE International Conference on Computer Vision*. Tome 2. 2003, pages 1470–1477 (cf. page 20).

[63] Kenji SUZUKI, Isao HORIBA et Noboru SUGIE. “A simple neural network pruning algorithm with application to filter synthesis”. In : *Neural Processing Letters* 13.1 (2001), pages 43–53 (cf. page 49).

[64] S. TONG et D. KOLLER. “Support vector machine active learning with application to text classification”. In : *International Journal on Machine Learning Research* 2 (nov. 2001), pages 45–66 (cf. page 42).

[65] Marguerite FRANK et Philip WOLFE. “An algorithm for quadratic programming”. In : *Naval Research Logistics Quarterly* 3.1-2 (1956), pages 95–110. ISSN : 1931-9193. DOI : 10.1002/nav.3800030109. URL : http://dx.doi.org/10.1002/nav.3800030109 (cf. page 45).