Super-resolution of multiscale 3D images from materials science

Dang Phuong Lan Nguyen

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE: MATHÉMATIQUES ET INFORMATIQUE

SPÉCIALITÉ: MATHÉMATIQUES APPLIQUÉES ET CALCUL SCIENTIFIQUE

Par **Dang-Phuong-Lan NGUYEN**

Super-résolution multi-échelle d'images 3D en sciences des matériaux

Sous la direction de : **Jean-Francois AUJOL**
Co-directeur : **Yannick BERTHOUMIEU**
Encadrant: **Dominique BERNARD**

Soutenue le 17 Janvier 2023

Membres du jury :

Agnes DESOLNEUX	Directrice de Recherche	CNRS, ENS Paris Saclay	Rapporteure
Renaud PETERI	Maître de conférences - HDR	MIA, La Rochelle Université	Rapporteur
Michel BERTHIER	Professeur des universités	MIA, La Rochelle Université	Examinateur
Jean-François AUJOL	Professeur des universités	IMB, Université de Bordeaux	Directeur
Yannick BERTHOUMIEU	Professeur des universités	IMS, Bordeaux INP	Co-directeur
Dominique BERNARD	Directeur de Recherche	ICMCB, Université de Bordeaux	Invité

Acknowledgments

This thesis could not have been completed without the support and help of my advisors, co-authors, doctoral committee, colleagues, family, and friends, whom I would like to thank warmly.

First of all, I would like to express my deepest gratitude to my supervisors Professor Jean-Francois AUJOL, Professor Yannick BERTHOUMIEU, and Senior Researcher (Directeur de Recherche) Dominique BERNARD. I am very grateful to them for their support and very nice advice about my scientific study. They also helped me when I was disoriented and corrected a lot of mistakes when I wrote this thesis. I appreciate your friendship, patience, and for having guided me during these three unforgettable years. Especially during the most difficult period in my second year, Jean-Francois and Yannick were very understanding and patiently waiting for me to overcome my mental crisis. So I would like to thank them for their sharing and encouragement.

Next, I deeply appreciate Professor Agnes DESOLNEUX and Professor Renaud PETERI for kindly accepting the role of reviewers. I wish also to express my gratitude to Professor Michel BERTHIER, who accepted to be part of this jury as examiner. To all of them, thanks for taking the time to read this manuscript and for all the evaluation and the constructive comments.

I wish also to express my thanks to the members of my follow-up committee ("comite de suivi individuel"), Professor Jean-Pierre DA COSTA and Professor Yves COUDIERE, for several helpful comments concerning on the progression of my Ph.D. and the working environment during the preparation of this thesis.

In particular, I would like to give my special regards to University of Bordeaux and Agence Nationale de la Recherche (ANR) for providing financial support for my thesis to carry out the SUPREMATIM project. I was very fortunate to be involved in this project during my PhD. In this project, I am very happy to cooperate with Professor Gabriele STEIDL and doctor Johannes HERTRICH from the Technical University of Berlin (TU Berlin) and Professor Claudia REDENBACH from the Technical University of Kaiserslautern-Landau (RPTU) Germany. I am grateful to them all for their comments and contributions to my publications and especially for giving me the chance to collaborate on their research. In addition, I would also like to thank Dr. Abdellatif SAADALDIN for helping me discover and understand the material data of our project at the beginning of my thesis.

Furthermore, I would like to thank all the members of the Institut de Mathématiques de Bordeaux (IMB) in particular the Image Optimization and Probabilities team (IOP) and the Laboratoire de l'Intégration du Matériau au Système (IMS) in particular the Signal and Image Group (GSI), for a warm welcome and providing the facilitatory research environment during these three years of my PhD. Then I would also like to express my gratitude to the secretaries for their kind support and to the IT team of the IMB, in particular Mr. Laurent FACQ, for his kind support in learning how to use the curta computing cluster (MCIA) and the PlaFRIM computing cluster. I want to express my acknowledgment to my

colleagues in the IMB and IMS laboratory for their help and support. I want to thank to Dr. Yiye Jiang, PhD student Paul Freulon, Dr. Nicoletta Prencipe, Dr. Anaïs Gastineau, Dr. Sebastián Tapia García in IMB. Especially Paul and Yiye who have accompanied and shared with me a lot during these 3 years.

Moreover, I wish to express my special thanks to Professor Simon MASNOU for providing me the valuable master internship at Institute Camille Jordan, University Lyon 1. Thanks to him, I have learned about the background knowledge of image processing to prepare for my PhD thesis. Secondly, I would like to thank Professor Clément MARTEAU who was very supportive of my studies and helped me get used to the new learning environment when I first arrived at the University of Lyon for my Master's program. Here, I would also like to thank my friends at the University of Lyon, especially doctoral student Jui-Ting Lu. She did not hesitate to help me discover our university and the city of Lyon with me and to help me integrate with my classmates.

Next, I wish to express my gratitude to the FRENCH-VIETNAM Master 2 in applied Mathematics program (PUF) of University of Science, Vietnam National University Ho Chi Minh City and the French Universities. Thanks to this program, I have access to the research career as well as the opportunity to find master and PhD scholarships in France. Moreover, I wish to express my special thanks to Professor Minh-Duc DUONG, Professor François James, Professor Mounir Haddou, Dr. Thanh-Hai ONG, Associate Professor Thanh-Nhan NGUYEN, Dr. Tri-Dung NGUYEN for their important academic lectures during my undergraduate and graduate programs. I also would like to thank to my Vietnamese friends Dr. Tan-Binh PHAN, Dr. Van-Thanh NGUYEN, Dr. Duc-Thach-Son VU, Mr. Viet-Anh NGUYEN, Mrs. Minh-Thi NGUYEN, Miss Huyen-Mai PHAM, Dr. Thuy-Ha PHAM, Mrs. Nhat-Vy NGO, Miss Thanh-Xuan NGUYEN, Mr. Huu-Nhut LE, Dr. Phuong-Yen TRAN, Mrs Simone Dang Van and her husband.

Finally, I wish to express my gratitude to my family, my boyfriend and his family, who always love, care, and encourage me. I would especially like to thank my parents Mr. Dang-Hau NGUYEN and Mrs. Thi-Hoa VO, my elder sisters Dang-Buu Tram NGUYEN, Dang-Huyen-Tran NGUYEN , my brothers-in-laws Tuan-Khoa NGUYEN and Duc-Hieu NGUYEN and last but not least my niece Diep-Thao-Nguyen NGUYEN, my nephew Duc-Huy NGUYEN. They are my biggest motivation to see this thesis through to completion. I want to thank my parents for always creating the best possible conditions for my high school studies and for supporting my passion for math by allowing me to continue studying it at the university level. I believe they're also pleased with my achievements since completing my thesis. As a belated but heartfelt gesture, I would like to dedicate this thesis as a final gift to my mother. I miss you and I love you.

Bordeaux, January 17, 2023

Dang-Phuong-Lan NGUYEN

Super-résolution multi-échelle d'images 3D en sciences des matériaux

Résumé : Les développements récents des techniques d'imagerie et de l'analyse computationnelle modifient profondément la manière dont les sciences des matériaux sont abordées. L'image des matériaux est passée d'une résolution à l'échelle microscopique à une véritable échelle nanométrique pour analyser les défauts et les détails aux interfaces des matériaux. Ainsi, cette thèse traite du problème de la super-résolution (SR) afin de reconstruire les images de matériaux en haute résolution, par exemple à l'échelle nanométrique. Pour atteindre ce but, cette thèse explore plusieurs approches de la SR pour les images de matériaux.

Les algorithmes de super-résolution d'image unique (SISR) basés sur les patchs ont été remarqués et largement utilisés au cours de la dernière décennie. Récemment, les modèles de mélange gaussien généralisé (GGMM) se sont révélés être un outil approprié pour de nombreux problèmes de traitement d'images en raison de la flexibilité de leur paramètre de forme. Dans un premier temps, nous proposons d'utiliser un GGMM conjoint appris à partir de vecteurs concaténés de patchs d'entraînement à haute et basse résolution pour réaliser une image en super-résolution basée sur la technique de minimum mean square error (MMSE).

Malheureusement, la dimension des échantillons concaténés est très élevée ce qui entraîne une forte complexité de calcul des méthodes d'apprentissage des modèles de mélange tels que le modèle de mélange gaussien et le GGMM. Afin de limiter cette compléxité, nous proposons une extension de ces deux modèles intégrant une réduction de la dimensionnalité des données dans chaque composante du modèle par analyse en composantes principales. Ces modèles sont appelés respectivement PCA-GMM et PCA-GGMM. Pour apprendre les paramètres (de faible dimension) du modèle de mélange, nous proposons deux algorithmes EM différents dont l'étape M nécessite la résolution de problèmes d'optimisation sous contrainte. Ensuite, nous appliquons nos PCA-GMM et PCA-GGMM pour la super-résolution d'images de matériaux 2D et 3D en nous basant sur la méthode MMSE pour le modèle GGMM.

Sur le plan applicatif, l'étude des données de matériaux devient difficile car les images HR et LR possèdent des niveaux de contraste différents. Dans notre dernière contribution, nous étudions une approche d'apprentissage profond considérant le problème du changement de contraste dans les images de matériaux. En effet, nous proposons un réseaux antagonistes génératif (GAN) composé de deux générateurs, chacun répondant à une tâche différente. Le premier générateur traite le problème du changement de contraste, et le second se concentre sur la reconstruction des hautes fréquences du SR.

Toutes les méthodes proposées conduisent à des résultats convaincants, tant quantitatifs que visuels. En particulier, les résultats numériques des méthodes de réduction de la dimensionnalité confirment une influence modérée de la réduction de la taille sur les résultats globaux de la SR.

Mots-clés : Super-résolution, science des matériaux, les modèles de mélange Gaussien, les modèles de mélange Gaussien généralisé, réduction de la dimensionnalité, apprentissage profond, réseaux antagonistes génératif (GAN).

Super-resolution of multiscale 3D images from materials science

Abstract: Recent developments in imaging techniques and computational analysis deeply modify the way materials sciences. The materials image has moved from microscale resolution to true nanoscale to analysis the defects and details at the interfaces of the materials. Thus, this thesis deals with the super-resolution (SR) problem in order to reconstruct the materials images in the high-resolution for instance the nanoscale resolution. To reach this goal, this thesis explores several SR approaches for materials images.

Single Image Super-Resolution (SISR) algorithms based on patch-based have been noticed and widely used over the past decade. Recently, Generalized Gaussian Mixture Models (GGMMs) have shown to be a suitable tool for many image processing problems due to the flexible shape parameter. In the first place, we propose using a joint GGMM learned from concatenated vectors of high and low resolution training patches to do super-resolution image based on the minimum mean square error (MMSE) method.

Unfortunately, the dimension of the concatenated samples is too high for the learning of the mixture models such as Gaussian mixture model and GGMM. Then we propose these two models Gaussian mixture model (GMM) and GGMM with a reduction of the dimensionality of the data in each component of the model by principal component analysis. These models are called to as PCA-GMM and PCA-GGMM, respectively. To learn the (low dimensional) parameters of the mixture model we propose two different expectation-maximization (EM) algorithms whose M-step requires the solution of constrained optimization problems. Then we apply our PCA-GMM and PCA-GGMM for the super-resolution of 2D and 3D material images based on the MMSE method for the GGMM model.

In addition, the study of the material data becomes difficult because the HR and LR images have a different contrast. In our last contribution, we study a deep learning approach considering the problem of contrast change in material images. Indeed, we propose a generative adversarial network (GAN) within two generators, each responding to a different task. The first generator deals with the problem of contrast changes, and the second one focuses on the reconstruction of the high frequencies of the SR.

All the proposed methods lead to convincing results, both quantitative and visual. Especially the numerical results of the dimensionality reduction methods confirm a moderate influence of the size reduction on the overall SR results.

Keywords: Image super-resolution, materials science, Gaussian mixture model, generalized Gaussian mixture model, high-dimensional data, dimensionality reduction, deep learning, generative adversarial network (GAN).

Adresse:
L'Institut de Mathématiques de Bordeaux (IMB)
UMR 5251 Université de Bordeaux, CNRS, Bordeaux INP, 33400 Talence, France.
Laboratoire de l'Intégration du Matériau au Système (IMS)
UMR 5218 Université de Bordeaux, CNRS, Bordeaux INP, 33400 Talence, France.

?contentsname?

?listfigurename?

?listtablename?

List of Acronyms

Introduction (Français)

Contents

Cette thèse a été rédigée à l'Institut de Mathématiques de Bordeaux (IMB), UMR 5251 et au laboratoire IMS du CNRS (Laboratoire de l'Intégration du Matériau au Système, UMR 5218) - Université de Bordeaux - Bordeaux INP). Cette thèse a été financée par l'Agence Nationale de la Recherche (ANR) dans le cadre du projet ANR-18-CE92-0050 SUPREMATIM[1] (SUPerREsolution of 3d MATerials IMages). L'objectif de ce projet est de développer des méthodes de super-résolution (SR) pour des images de matériaux en 3D à partir d'une image à basse résolution (LR) et d'une image locale guidée à haute résolution (HR). Les images 3D de matériaux sont prises dans le cadre du projet MUMMERING[2] (MUltiscale, Multimodal and Multidimensional imaging for EngineeRING).

1.1 Contexte

La super-résolution (SR), qui consiste à obtenir une ou plusieurs images à haute résolution (HR) à partir d'une ou plusieurs observations à basse résolution (LR), a trouvé des applications dans de nombreux problèmes du monde réel. Au cours des deux dernières décennies, un grand nombre d'articles de recherche et de livres traitant d'objectifs pratiques spécifiques ont été rédigés. Pour un très bon aperçu des résultats obtenus jusqu'en 2014, nous vous renvoyons à l'étude exhaustive [68].

En ce qui concerne les algorithmes de super-résolution image unique (SISR), la plupart des travaux sont basés sur l'interpolation ou sur les statistiques de la paire d'images HR-LR. Diverses méthodes basées sur l'interpolation ont été proposées à partir d'interpolateurs linéaires ou non linéaires incorporant des noyaux de reconstruction adaptatifs plus ou moins sophistiqués à partir de structures locales

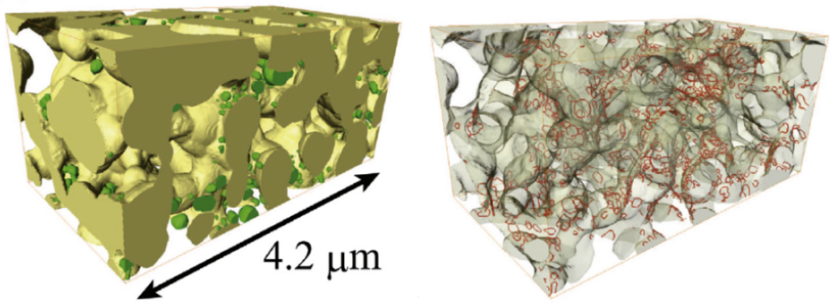
[1] https://www.math.u-bordeaux.fr/~jaujol/suprematim/
[2] http://www.mummering.eu/index

des images telles que les bords [75]. Récemment, compte tenu de la définition mal conditionnée du problème de SR, des approches variationnelles ou bayésiennes ont été envisagées [33, 42]. Les a priori appris sont exploités dans les algorithmes SR pour concevoir un mappage statistique adapté entre l'image HR et l'image LR. Des modèles d'image populaires ont été étudiés, allant de la variation totale [42] aux décompositions de type ondelettes, en passant par les hypothèses de sparsité [1, 42]. Plus spécifiquement, dans le contexte des problèmes inverses généraux, le travail de Buades et al. [12] a montré un intérêt dans l'utilisation des patches et de l'hypothèse d'auto-similarité pour apprendre une mise en correspondance statistique efficace entre les images observées et restaurées. Dans un travail récent, [33] a développé une méthode variationnelle de SR pour les images texturées qui vise à super-résoudre la texture LR en prenant explicitement en compte les patchs HR dans un schéma de régularisation non-local couplé à plusieurs a priori d'histogrammes de caractéristiques locales de l'image [33]. Une autre approche de la SR basée sur l'apprentissage est celle des méthodes basées sur des exemples qui exploitent la structure récurrente de l'image des patchs à l'intérieur et à travers différentes échelles [75]. L'objectif principal de cette classe de méthodes est d'une part d'apprendre un modèle conjoint de paires de patchs HR et LR et d'autre part d'estimer un patch HR à partir d'un patch LR donné.

Zoran et Weiss [92, 93] ont montré que, malgré leur simplicité, les modèles de mélanges gaussiens (GMM) sont remarquablement efficaces pour modéliser les statistiques des patchs d'images. Dans les tâches de restauration d'images, ils améliorent d'autres méthodes préalables génériques comme, par exemple, le codage parcimonieux.

Récemment, Sandeep et Jacob [75] ont montré le potentiel de la modélisation paramétrique pour apprendre la cartographie statistique en SR dans le cas où le noyau de flou est inconnu. En supposant un grand ensemble de patchs de paires d'images exemples HR et LR sous le modèle SR, les auteurs ont démontré que le GMM conjoint est une proposition très compétitive pour les méthodes basées sur l'apprentissage en termes de performance SR et de charge de calcul. Ils proposent une étude comparative avec un large panel de méthodes qui tirent parti de techniques d'apprentissage bien connues consacrées à l'appariement des dictionnaires HR-LR.

Dans le contexte du traitement d'images à partir de l'extraction de caractéristiques de patchs, des travaux récents [21] tirent parti du modèle de mélange gaussien généralisé (GGMM) pour apprendre les statistiques des patchs d'images. C. Deledalle et al. ont montré que le GGMM obtient de meilleures distributions de patchs qu'un GMM et surpasse le GMM lorsqu'il est utilisé dans le même cadre que [92]. Divers travaux [2, 10, 71] ont montré la possibilité de considérer des modèles à fonction de densité de probabilité, qui ont plus de flexibilité pour s'adapter à la forme des données et moins de sensibilité à la sur-adaptivité du nombre de classes que le GMM.



?figurename? 1.1: Rendu 3D d'une électrode de pile à combustible avec, sur la partie gauche, la porosité (transparente), le conducteur électronique (vert) et le conducteur ionique (jaune). Sur la partie droite, les lignes rouges représentent les Triple-Phase-Boundaries où les réactions électrochimiques ont lieu.

1.2 Motivations et objectifs

1.2.1 Motivation en sciences des matériaux

Les développements récents et en cours dans les techniques d'imagerie et l'analyse computationnelle modifient profondément la façon dont la science et l'ingénierie des matériaux considèrent leurs sujets de recherche. Notre projet contribue à cette direction de recherche en développant de nouvelles méthodes de SR guidées par une sous-image locale de HR des données de matériaux 3D. L'imagerie 3D est passée d'une résolution à micro-échelle à un véritable régime nano-échelle. Cela permet le passage de l'analyse statistique à l'analyse des modulations des défauts et des détails aux interfaces. Les interfaces entre les phases des matériaux sont des caractéristiques importantes et déterminantes pour les performances de la science des matériaux. Elles sont décrites comme dans la Figure 1.1. Les méthodes d'imagerie multi-échelle constituent le pont entre la compréhension à nano-échelle et les modèles de performance à micro-échelle, non seulement pour les systèmes statiques, mais aussi pour mesurer les paramètres structurels clés localement et en fonction du temps.

Le projet MUMMERING travaille sur le transport réactif dans les milieux poreux: une approche d'imagerie et de modélisation multi-échelle. Des approches d'imagerie et de modélisation multi-échelles ont été utilisées pour étudier le transport réactif dans les milieux poreux où les réactions hétérogènes sont fondamentales, ce qui signifie que les frontières solides doivent être décrites avec précision. D'autre part, la distribution spatiale globale du solide doit également être caractérisée avec précision, car elle détermine les propriétés moyennes de transport. La méthode SR d'image nous permet de compléter l'image LR globale avec des informations locales extraites d'une région donnée de l'image HR et LR. Cela permet d'affiner de manière itérative une simulation numérique des propriétés de transport dans l'espace global avec des images HR. À cette fin, notre projet (SUPREMATIM) a été construit pour explorer le modèle statistique reliant les images LR et HR en collaboration avec le groupe allemand du professeur Gabriele Steidl à la Technische Universität Berlin et

du professeur Claudia Redenbach à la Technische Universität Kaiserslautern.

Dans le projet SUPREMATIM, nous examinons deux problèmes transitoires de la science des matériaux (défaillance des composites en aluminium et réactions électrochimiques dans les électrodes poreuses), pour lesquels l'imagerie à nano-échelle et à micro-échelle est nécessaire. Les perspectives de développement de la SR comme méthode de propagation de l'information géométrique de nano-échelle à micro-échelle sont encore plus stimulantes ici, puisque l'imagerie à micro-échelle (tomographie aux rayons X) est non destructive, mais que l'imagerie à nano-échelle pourrait être destructive (tomographie au microscope électronique à balayage à faisceau d'ions focalisé (FIB-SEM)). En fait, la caractérisation 3D des matériaux considérés ici serait facilitée si les acquisitions HR destructives étaient limitées à des zones suffisamment petites pour maintenir la représentativité de l'échantillon, lorsque les acquisitions LR et HR sont réalisées.

1.2.2 Objectifs

L'objectif de cette thèse est le développement de méthodes de SR pour des images 3D uniques provenant d'images LR et HR de la même sonde de matériaux. Sur la base de nos motivations, nous avons décidé de développer de nouvelles méthodes de SR guidées par des sous-images HR locales à partir d'images LR globales.

Pour cela, le premier objectif de cette thèse est d'étendre la super-résolution d'image utilisant la méthode GMM des travaux précédents [75, 92] au cas GGMM. Nous avons l'intention de capturer les statistiques HR de l'image LR d'une manière plus appropriée en utilisant cette modélisation plus riche. Le deuxième objectif de cette thèse est d'étudier les techniques de réduction de la dimensionnalité qui incorporent un GMM ou un GGMM. Enfin, nous considérons les techniques d'apprentissage profond pour la super-résolution.

1.3 Résumé de nos contributions

Sur la base de nos objectifs, cette thèse est divisée en quatre contributions principales. Tout d'abord, pour aborder la SR d'images dans des images matérielles 3D guidées par des sous-images HR locales issues d'images LR globales, nous avons choisi de nous placer dans le cadre d'une approche basée sur l'exemple. Cette approche s'appuie sur des jeux de données spécifiques fournis à partir de sous-images HR locales et d'images LR correspondantes issues d'images LR globales. Elle étend la méthode SR de Sandeep et Jacob [75] au modèle de mélange gaussien généralisé. À cette fin, nous proposons un algorithme comme première contribution pour apprendre un modèle de mélange gaussien conjoint basé sur l'algorithme espérance-maximisation (EM) à partir d'une paire de patchs LR et des patchs HR correspondants provenant des données de référence. Les paramètres du GGMM sont estimés en utilisant les itérations d'un algorithme de point fixe (FP) pour l'estimation du maximum de vraisemblance pondéré dans l'algorithme EM. Cet algorithme EM incorporant des itérations FP pour le modèle GGMM est appelé

algorithme FP-EM. Nous reconstruisons ensuite l'image HR en utilisant l'estimateur de l'erreur quadratique moyenne minimale (MMSE) pour le modèle GGMM. Cette proposition est appelée la méthode MMSE-GGMM. En outre, l'image du matériau est passée d'une résolution à l'échelle du micron à une véritable échelle nanométrique. L'étude des données matérielles est très difficile car les images HR et LR ont des contrastes différents. De plus, la connaissance de l'opérateur SR n'est pas connue. Dans cette contribution, nous démontrons que notre méthode gère l'opérateur inconnu et est invariante au problème de changement de contraste affine.

La deuxième contribution consiste en une nouvelle approche pour la réduction de la dimension des données avec des applications en super-résolution. En effet, la réduction de dimensionnalité devient importante dans l'apprentissage de modèles statistiques lorsque la dimension des données est élevée. Tout d'abord, nous proposons un modèle de mélanges gaussiens en conjonction avec une réduction de la dimensionnalité des données dans chaque composante du modèle par analyse en composantes principales (PCA), appelée PCA-GMM. Ce modèle est généré en ajoutant le terme de minimisation de la PCA et la fonction de log-vraisemblance négative du GMM sur les points de données de dimension réduite. Ensuite, pour apprendre les paramètres (de faible dimension) du modèle de mélange, nous proposons un algorithme EM dont l'étape M nécessite la résolution de problèmes d'optimisation sous contrainte. Deuxièmement, nous appliquons notre PCA-GMM pour la SR d'images de matériaux 2D et 3D en nous basant sur l'approche de Sandeep et Jacob [75]. Les résultats numériques confirment l'influence modérée de la réduction de la dimensionnalité sur le résultat du SR.

La troisième contribution étend la contribution précédente au modèle GGMM pour le problème de réduction de la dimensionnalité. Dans ce travail, nous proposons de combiner un GGMM avec une PCA et de dériver un algorithme EM pour estimer les paramètres du modèle qui en résulte. Le nouveau modèle est appelé modèle PCA-GGMM. Pour gérer la complexité de l'algorithme EM pour le modèle PCA-GGMM, la technique de séparation semi-quadratique est utilisée dans l'étape d'estimation du maximum de vraisemblance. Ensuite, le modèle PCA-GGMM peut être appliqué au SR en utilisant la méthode MMSE-GGMM.

Dans la dernière contribution, cette thèse propose une nouvelle méthode SR basée sur l'apprentissage profond pour les images de matériaux avec le problème de changement de contraste. Parmi tous les types de réseaux de neurones convolutifs qui existent, nous avons choisi de nous placer dans le contexte des réseaux antagonistes génératifs (GAN). En effet, ces dernières années, les GAN ont commencé à émerger pour le problème de la super-résolution, notamment suite à la méthode proposée par Ledig et al. [56]. Cependant, ils manquent encore d'efficacité lorsque les contrastes des images HR et LR changent. Dans notre première contribution, nous avons montré que le MMSE-GGMM peut être invariant avec le problème de changement de contraste sous une transformation affine. Mais cette méthode n'est pas garantie pour le cas non affine. Pour surmonter ce problème avec une fonction de transformation non affine, nous proposons un nouveau cadre GAN dans lequel le générateur G est divisé en deux sous-réseaux. Le premier sous-

réseau traite du problème des changements de contraste, et le second sous-réseau se concentre sur la reconstruction des hautes fréquences du SR.

1.4 Organisation du manuscrit

Cette thèse traite du traitement de super-résolution de données matérielles en utilisant des méthodes basées sur l'apprentissage, en particulier, basées sur des modèles de mélanges gaussiens généralisés locaux et globaux. Cette thèse apporte quatre contributions à ce sujet. Notre première contribution vise à prendre en compte une nouvelle méthode de super-résolution qui utilise l'estimateur MMSE pour le modèle de mélanges gaussiens généralisés GGMM. Les deuxième et troisième contributions de cette thèse consistent à introduire deux modèles qui combinent la méthode PCA de réduction de dimensionnalité avec le modèle de mélanges gaussiens et le modèle de mélanges gaussiens généralisés. La dernière contribution est de concevoir l'architecture d'un GAN pour nos données matérielles avec les défis de changement de contraste et d'opérateur de corruption inconnu.

Dans le chapitre 3, nous donnons une vue d'ensemble des données sur les matériaux et un état-de-l'art général des méthodes existantes pour aborder et résoudre le problème de la super-résolution des images de matériaux. Les défis de la super-résolution pour notre image matérielle sont mentionnés dans ce chapitre. Il s'agit des changements de contraste, de luminosité de l'image, et des problèmes de super-résolution avec un opérateur de corruption inconnu. Il existe une grande variété de méthodes pour résoudre le problème de super-résolution d'image. En effet, il existe des approches basées sur l'interpolation, des approches basées sur la reconstruction ou des approches basées sur l'exemple. Par conséquent, ce chapitre vise à souligner les forces et les faiblesses de chaque type de méthodes.

Le chapitre 4 est consacré à la méthode de super-résolution guidée par la sous-image HR locale basée sur l'approche par exemple. Tout d'abord, nous présentons un état de l'art détaillé des approches basées sur les patchs pour le problème de la super-résolution. Il s'agit notamment de la méthode de super-résolution via le modèle de mélanges gaussiens conjoints [75] et de la méthode EPLL (Expected patch log-likelihood) pour la super-résolution [92]. Nous terminons ce chapitre en nous concentrant sur notre première contribution concernant l'approche MMSE-GGMM du problème de la SR. Cette contribution a fait l'objet d'une publication lors de la conférence GRETSI 2022 [69].

Le chapitre 5 traite du défi de la haute dimension des données matérielles. Pour résoudre ce problème, nous étudions quelques solutions de pointe au problème de la réduction de la dimensionnalité. Cela inclut le modèle PCA, l'analyse probabiliste en composantes principales (PPCA) pour la distribution gaussienne, le modèle MPPCA (Mixture of Probabilistic Principal Component Analyzers) et la méthode HDDC (High-Dimensional Data Clustering). Ensuite, nos deuxième et troisième contributions sont présentées dans le contexte de leurs applications en super-résolution. La deuxième contribution a été publiée dans la revue *Inverse*

Problems in Imaging [41] en 2022. La troisième contribution a été soumise à la revue *Inverse Problems in Imaging* [70] en 2022.

Ensuite, le chapitre 6 est orienté vers les méthodes basées sur les GAN pour le problème de changement de contraste de nos images de matériaux. Ce chapitre commence par une brève introduction aux réseau de neurones convolutif (CNN) et aux GAN pour le problème des SR. Ensuite, la dernière contribution sur les réseaux GAN avec deux sous-réseaux dans le générateur est mentionnée.

Le chapitre 7 conclut cette thèse et propose des pistes d'amélioration et des perspectives pour la tâche SR pour les images de matériaux.

?CHAPTERNAME? 2

Introduction

Contents

This thesis was written at the Institute of Mathematics of Bordeaux (IMB), UMR 5251 and the IMS laboratory (Laboratoire de l'Intégration du Matériau au Système, UMR 5218) in CNRS - Université de Bordeaux - Bordeaux INP). This thesis was funded through the French Agence Nationale de la Recherche (ANR) under the reference project ANR-18-CE92-0050 SUPREMATIM[1] (SUPerREsolution of 3d MATerials IMages). The aim of this project is the development of super-resolution (SR) methods for 3D materials images originating from a low-resolution (LR) image and a guided local high-resolution (HR) sub-image. The 3D materials images are taken within the MUMMERING[2] project (MUltiscale, Multimodal and Multidimensional imaging for EngineeRING).

2.1 Context

Super-resolution (SR), the process of obtaining one or more high-resolution (HR) images from one or more low-resolution (LR) observation(s) has found applications in many real-world problems. Over the past two decades a large number of research papers and books addressing specific practical purposes have been written. For a very good overview of the results until 2014 we refer to the comprehensive survey [68].

Considering single image super-resolution (SISR) algorithms, most of the works are interpolation-based or based on the statistics of the pair HR-LR images. Various interpolation-based methods have been proposed from linear or nonlinear interpolators incorporating more or less sophisticated adaptive reconstruction kernels from local image structures such as edges [75]. Recently, taking into account

[1] https://www.math.u-bordeaux.fr/~jaujol/suprematim/
[2] http://www.mummering.eu/index

the underlying ill-conditioned definition of the SR problem, variational or Bayesian approaches have been considered [33, 42]. The learned priors are exploited in the SR algorithms to design an adapted statistical mapping between the HR image and the LR image. Popular image models have been studied ranging from total variation [42], wavelet-like decompositions, or sparsity assumptions [1, 42]. More specifically, in the context of general inverse problems, the special work of Buades et al. [12] has shown an interest in using patches and self-similarity hypothesis to learn efficient statistical mapping between observed and restored images. In a recent work, [33] has developed a variational SR method for textured images that aims to super-resolve the LR texture by explicitly taking into account HR patches in a non-local regularization scheme coupled with several histogram priors of local image features [33]. Another approach to learning-based SR is example-based methods that exploit the recurrent image structure of patches within and across different scales [75]. The main purpose of this class of methods is on the one hand to learn a joint model of HR and LR patch pairs and on the other hand to estimate a HR patch from a given input LR patch.

Zoran and Weiss [92, 93] showed that, despite their simplicity, Gaussian mixture model (GMM) is remarkably good at modeling the statistics of natural image patches. In image restoration tasks they improve other generic prior methods such as for instance sparse coding.

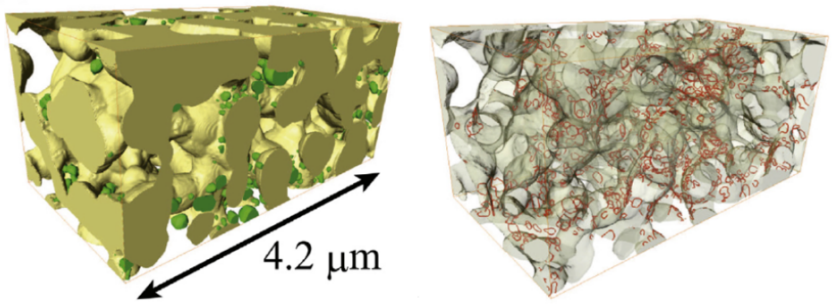
Recently, Sandeep and Jacob [75] have shown the potential of parametric modeling to learn statistical mapping in SR in the case where the blur kernel is unknown. Assuming a large patch set of HR and LR pairs of example images under the SR model, the authors demonstrated that joint GMM is a very competitive proposal for learning based methods in terms of SR performance and computational load. They propose a comparative study with a large panel of methods that take advantage of well-known learning techniques devoted to pairing HR-LR dictionaries.

In the context of image processing from patch feature extraction, recent work [21] takes advantage of generalized Gaussian mixture model (GGMM) to learn the statistics of image patches. C. Deledalle et al. showed that the GGMM obtains better patch distributions than a GMM and outperforms the GMM when used in the same framework with [92]. Various works [2, 10, 71] have shown the possibility of considering probability density function models, which have more flexibility to adapt the shape of data and less sensibility to overfitting the number of classes than the GMM.

2.2 Motivations and objectives

2.2.1 Motivation from Materials Sciences

Recent and ongoing developments in imaging techniques and computational analysis deeply modify the way materials science and engineering consider their research subjects. Our project contributes to this direction of research by developing new SR methods guided by a local HR sub-images of 3D materials data. 3D imaging



?figurename? 2.1: 3D rendering of a fuel cell electrode with, on the left part, porosity (transparent), electronic conductor (green) and ionic conductor (yellow). On the right part the red lines represent the Triple-Phase-Boundaries where electrochemical reactions take place.

has moved from microscale resolution to true nanoscale regime. This allows the passage from statistical analysis to the analysis of the modulations of the defects and details at the interfaces. The interfaces between the phases of materials are important and determining characteristics for the performance of materials science. They are described as in Figure 2.1. Multiscale imaging methods form the bridge between the nanoscale understanding and the microscale performance models, not only for static systems, but also to measure key structural parameters locally and as a function of time.

The MUMMERING project is working on reactive transport in porous media: A multiscale imaging and modelling approach. Multiscale imaging and modeling approaches were used to investigate reactive transport in porous media where heterogeneous reactions are fundamental, meaning that the solid boundaries must be precisely described. On the other hand, the global spatial distribution of the solid must also be precisely characterized, because it determines the average transport properties. The image SR method allows us to supplement the global LR image with local information extracted from a given HR and LR image region. This allows for the iterative refinement of a numerical simulation of transport properties in global space with HR images. To this end, our project (SUPREMATIM) was built to explore the statistical model linking LR and HR images in collaboration with the German group of Professor Gabriele Steidl at Technische Universität Berlin and Professor Claudia Redenbach at Technische Universität Kaiserslautern.

In the SUPREMATIM project, we consider two transitory problems of materials science (failure of Aluminum composites and electrochemical reactions in porous electrodes), where nanoscale and microscale imaging are necessary. The perspectives of the development of SR as a method to propagate the geometrical information from the nanoscale to the microscale are even more stimulating here, since microscale imaging (X-ray tomography) is non-destructive, but nanoscale imaging could be destructive (Focused Ion Beam Scanning Electron Microscope (FIB-SEM) tomography). In fact, 3D characterization of the materials considered here would

be facilitated if destructive HR acquisitions were limited to zones small enough to maintain the representativeness of the sample, when LR and HR are acquired.

2.2.2 Objectives

The aim of this thesis is the development of SR methods for single 3D images originating from LR images and HR images from the same material probe. Based on our motivations, we decided to develop new SR methods guided by local HR sub-images from global LR images.

For this, the first goal of this thesis is to extend the image super-resolution using the GMM method of previous work [75, 92] to the GGMM case. We intend to capture the HR statistics from the LR image in a more appropriate way by using this richer modeling. The second goal of this thesis is to study the dimensionality reduction techniques which incorporate a GMM or GGMM. Finally, we consider deep learning techniques for super-resolution.

2.3 Summary of our contributions

Based on our objectives, this thesis is divided into four main contributions. First of all, to address image SR in 3D material images guided by local HR sub-images from global LR images, we have chosen to place ourselves in the framework of an example-based approach. This approach relies on specific datasets provided from local HR sub-images and corresponding LR images from global LR images. It extends the SR method of Sandeep and Jacob [75] to the generalized Gaussian mixture model. To this end, we propose an algorithm as a first contribution to learn a joint GGMM based on the Expectation-Maximization (EM) algorithm from a pair of LR patches and the corresponding HR patches from the reference data. The parameters of the GGMM are estimated by using the Fixed-Point (FP) iterations for the weighted maximum likelihood estimation in the EM algorithm. This EM algorithm incorporated with FP iterations for the GGMM model is called the Fixed-Point Expectation-Maximization (FP-EM) algorithm. We then reconstruct the HR image using the Minimum mean square estimator (MMSE) estimator for the GGMM model. This proposal is called the MMSE estimator for joint GGMM (MMSE-GGMM) method. Furthermore, the material image has moved from micronscale resolution to true nanoscale. Studying the material data is very challenging because the HR and LR images have different contrasts. In addition, the knowledge of the SR operator is not known. In this contribution, we demonstrate that our method handles the unknown operator and is invariant to the affine contrast change problem.

The second contribution consists of a new approach for data dimensionality reduction with applications in super-resolution. In fact, dimensionality reduction becomes important in learning statistical models when the dimension of the data is high. First, we propose a Gaussian Mixture Model in conjunction with a reduction of the dimensionality of the data in each component of the model by Principal

Component Analysis (PCA), called PCA - Gaussian mixture model (PCA-GMM). This model is generated by adding the minimization term of the PCA and the negative log-likelihood function of the GMM on the reduced-dimensionality data points. Then, to learn the (low-dimensional) parameters of the mixture model, we propose an expectation–maximization (EM) algorithm whose M-step requires the solution of constrained optimization problems. Second, we apply our PCA-GMM for the SR of 2D and 3D material images based on the approach of Sandeep and Jacob [75]. Numerical results confirm the moderate influence of the dimensionality reduction on the overall SR result.

The third contribution extends the previous contribution to the GGMM model for the dimensionality reduction problem. In this work, we propose to combine a GGMM with a PCA and derive an EM algorithm for estimating the parameters of the arising model. The new model is called the PCA - generalized Gaussian mixture model (PCA-GGMM) model. To handle the complexity of the EM algorithm for the PCA-GGMM model, the half-quadratic splitting technique is used in the maximum likelihood estimation step. Then the PCA-GGMM model can be applied to SR by using the MMSE-GGMM SR method.

In the last contribution, this thesis proposes a new SR method based on deep learning for material images with the contrast change problem. Among all the types of convolutional neural networks that exist, we have chosen to place ourselves in a framework of Generative adversarial network (GAN). Indeed, in recent years, GAN has begun to emerge for the super-resolution problem, in particular following the method proposed by Ledig et al. [56]. However, they still lack effectiveness when the contrasts of the HR and LR images change. In our first contribution, we showed that MMSE-GGMM can be invariant with the contrast change problem under an affine transformation. But this method is not guaranteed for the non-affine case. To overcome this problem with a non-affine transformation function, we propose a new GAN framework in which the generator G is divided into two sub-networks. The first sub-network deals with the problem of contrast changes, and the second sub-network focuses on the reconstruction of the high frequencies of the SR.

2.4 Organisation of the manuscript

This thesis discusses super-resolution processing of material data using learning-based methods, in particular, based on local and global generalized Gaussian mixture models. This thesis provides four contributions to the topic. Our first contribution aims to take into account a new method for super-resolution that uses the MMSE estimator for the generalized Gaussian mixture model. The second and third contributions in this thesis are to introduce two models that combine the PCA method of dimensionality reduction with the Gaussian mixture model and the generalized Gaussian mixture model. The last one is to design the architecture of a generative adversarial network (GAN) for our material data with the contrast change and unknown corruption operator challenges.

In Chapter 3, we give an overview of material data and a general state-of-the-art on existing methods to address and solve the image super-resolution problem for material image. The challenges of super-resolution for our material image are mentioned in this chapter. These are the changes in contrast, brightness of image, and super-resolution problems with unknown corruption operator. There are a wide variety of methods to address the image super-resolution problem. Indeed, there are interpolation-based approach, reconstruction-based approach, and example-based approach. Therefore, this chapter aims to outline the strengths and weaknesses of each type of method.

Chapter 4 is devoted to the super-resolution method guided by local HR sub-image based on the example-based approach. First, we present a detailed state-of-the-art of patch-based approaches for super-resolution problem. It includes the super-resolution method via the joint Gaussian mixture model [75] and the Expected patch log-likelihood (EPLL) method for super-resolution [92]. We finish this chapter by focusing on our first contribution concerning the MMSE-GGMM approach to the SR problem. This contribution was the subject of a publication at the GRETSI conference 2022 [69].

Chapter 5 deals with the high dimension challenge of material data. To address this problem, we study some state-of-the-art solutions of the dimensionality reduction problem. This includes the PCA model, the Probabilistic Principal Component Analysis (PPCA) for Gaussian distribution, the Mixture of Probabilistic Principal Component Analyzers (MPPCA) model, and the High-Dimensional Data Clustering (HDDC) method. Then, our second and third contributions are presented in the context of their applications in super-resolution. The second contribution was published in the *Inverse Problems in Imaging* journal [41] in 2022. The third contribution was submitted to the *Inverse Problems in Imaging* journal [70] in 2022.

Then, Chapter 6 is oriented towards methods based on generative adversarial network (GAN) for the contrast change problem of our material images. This chapter begins with a brief introduction to the Convolutional neural network (CNN) and GAN for the SR problem. Then the last contribution to the GAN network with two sub-networks in the generator is mentioned.

Chapter 7 concludes this thesis and proposes avenues for improvement and perspectives for the SR task for the material images.

?CHAPTERNAME? 3

Super-resolution for 3D images from materials science

Contents

In this chapter, we present the fundamentals of SR, by providing a general definition in Section 3.1. Section 3.2 discusses our materials data of this thesis. First, we present the material images on which most of the work in this thesis is based. After that, some challenges of super-resolution for materials data will be presented. Finally, we describe some main categories of approaches for SISR, and the SR imaging model used throughout this work.

3.1 Super-resolution definition

In this section, we provide some brief definitions of the digital image and related definitions of super-resolution.

Digital image A digital image can be considered as a discrete representation of data that contains spatial (layout) and intensity (color) information. The discrete two-dimensional (2D) digital image I is represented by a 2D array. Each element of the array is called pixel, derived from the term "picture element". Thus, a $M \times N$ digital image I looks like this

$$I = \begin{bmatrix} I(1,1) & I(1,2) & \cdots & I(1,N) \\ I(2,1) & I(2,2) & \cdots & I(2,N) \\ \vdots & \vdots & \vdots & \vdots \\ I(M,1) & I(M,2) & \cdots & I(M,N) \end{bmatrix} \tag{3.1.1}$$

where $I(m,n)$ represents the fixed pixel (m,n) of image I ($m = 1,2,...,M$ and $n = 1,2,...,N$). Image quality is a complex concept, largely subjective, and highly application-dependent. Basically, a good quality image is an image that has the following properties:

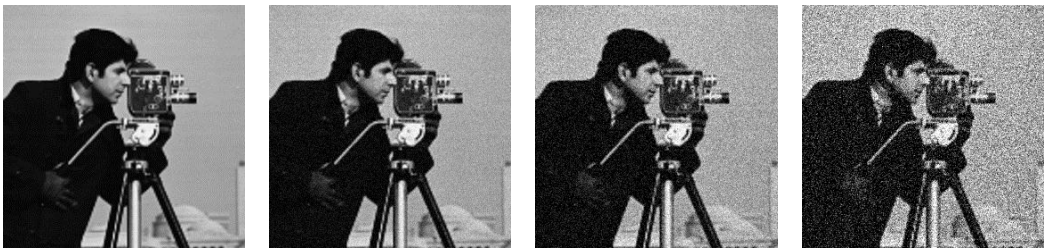
- it has a high resolution,
- it is not noisy,
- it has good contrast.

We discuss these definitions in the following of this subsection.

Image resolution In general terms, image resolution means the amount of detail that an image holds. The resolution of an image can be described in many different ways. According to the particular aspect taken into account, it can be mentioned as: pixel count, spatial resolution, spectral resolution, temporal resolution, radiometric resolution, etc. In particular, spatial resolution refers to the level of visual details discernible in an image. Higher resolutions mean that there are more pixels per inch (PPI), resulting in more pixel information and creating a high-quality, crisp image. As for digital images, where the pixel is the base unit used, there is a connection between spatial resolution and the total number of pixels. In practice, we often take into account simply the total counts of pixels, horizontally and vertically, which define what is more precisely called pixel resolution, and serves as upper bounds on spatial resolution. In this manuscript, the word resolution is intended as a spatial resolution. The resolution of a 2D image is commonly quoted as $C \times R$ (column by row), e.g. 640×480, 512×512, 2512×2512, etc. An example of quality of "cameraman image" with different resolutions is given in Figure 3.1. In this example, we keep the level of contrast and blur of the image constant, and reduce the resolution from 272×272 to 34×34. The three last images are generated using the downsampling operator as shown in equation (3.2.3) with magnification factors $q = 2, 4, 8$, respectively.

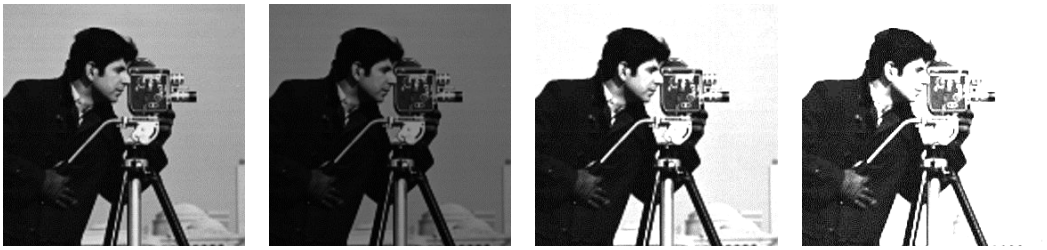
?figurename? 3.1: Cameraman image with different resolution from the left to the right: 272×272, 136×136, 68×68, and 34×34 (keep noise and contrast level constant).

Image noise Image noise is a random variation of brightness or color information in an image. It is caused by the image sensor and the circuits of a camera. Figure 3.2 shows the effect of noise when we change the level of noise.



?figurename? 3.2: Cameraman image size 272×272 with different level of noise (keep resolution and contrast level constant).

Contrast image Contrast is the difference in luminance or colour that represents in an image. Good contrast means that the grey values present in the image range from black to white, making use of the full range of brightness to which the human vision system is sensitive. The different contrasts of the image are illustrated in Figure 3.3.



?figurename? 3.3: Cameraman image with different contrasts (keep resolution constant and no noise).

Spatial Super-resolution Super-resolution (SR) is a class of image restoration that increases the resolution of image. According to the number of input LR images, SR can be classified into single-image super-resolution (SISR) and multi-image super-resolution (MISR). In the context of this thesis, we refer to SISR, which allows to increase the spatial resolution of the image from a single LR image. It means that for a given low-resolution image input, our task is to estimate the corresponding high-resolution image output and it must also provide an accurate image signal without distortion.

3.2 Super-resolution for materials data

As we discussed in Chapter 2, our motivation is to develop new super-resolution methods for data from 3D material images. In the frame of the ITN MUMMERING project, a series of multiscale 3D images has been acquired by synchrotron micro-tomography at the TOmographic Microscopy and Coherent rAdiology experimentTs (TOMCAT[1]) beamline of the Swiss Light Source (SLS[2]) at the Paul Scherrer Institute in Switzerland.

The series of multi-scale 3D images of two different samples were acquired following a protocol conceived for testing and validating the SR process by the following steps:

- Firstly, we selected samples with micro-geometries of different levels of complexity.

- Secondly, we acquired four tomograms for each sample for three magnifications $2, 4$, and 10 following the scheme given Figure 3.4. Two different regions were scanned at a magnification of 10. The first part is used for training the SR model and the other for comparing the real and reconstructed micro-geometry.

Materials of two samples were selected to provide 3D images having diverse levels of complexity:

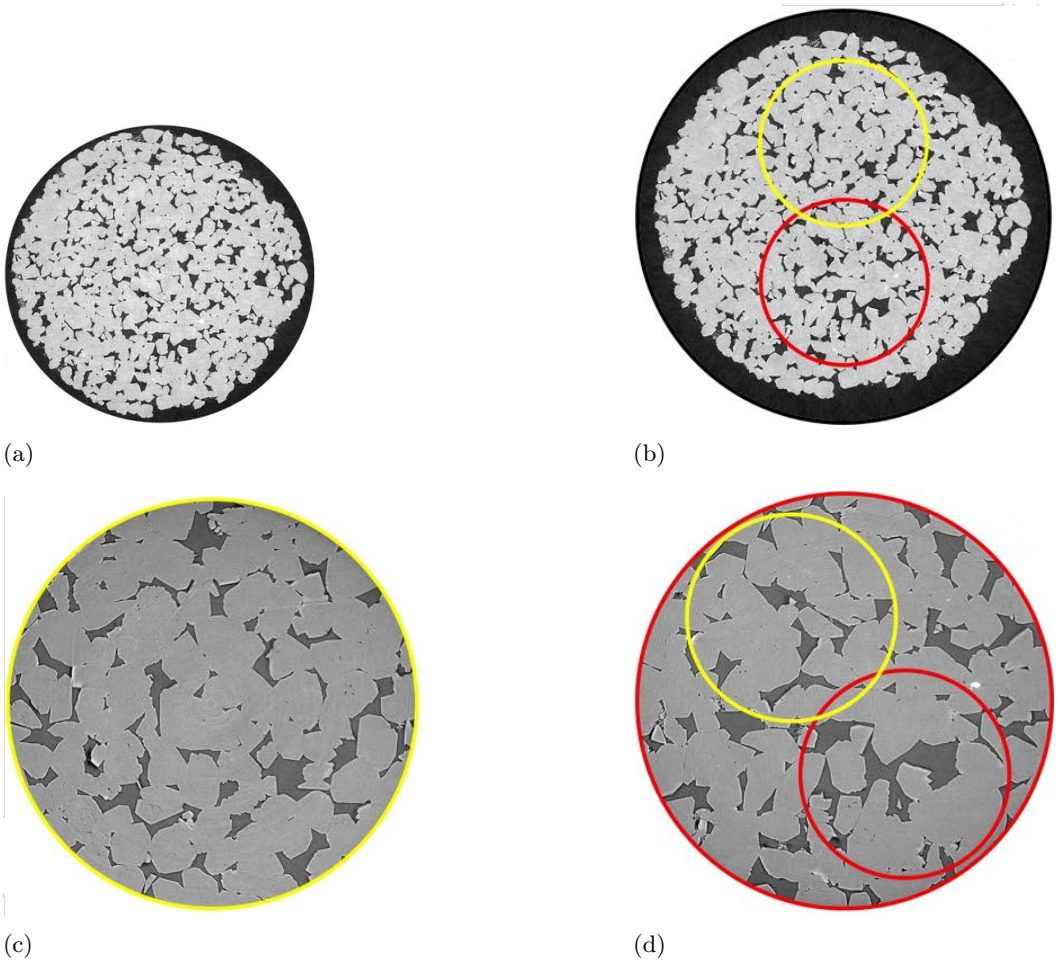
- The first one is a sample of Fontainebleau sandstone ("FS"), a natural rock rather homogeneous and commonly used in the oil industry for flow experiments.

- The second is a composite ("SiC Diamonds") obtained by microwave sintering of silicon and diamonds, see [81].

Each sample has been taken with different Magnification (MAG) of 2x, 4x, and 10x. The 3D data with 2x and 4x magnification were collected for the entire sample with $1600\times1600\times962$ and $2560\times2560\times2160$ voxels, respectively. With magnification 10x, two regions of sample were obtained by perform local microtomography resulting $3D$ volumes with $2560\times2560\times2160$ voxels. Figure 3.4 illustrates the multiscale imaging experiment with cross sections of the Fontainebleau sandstone sample. The two FS images with MAG of 10x were taken from the yellow and red circles of the FS image with MAG of 4x. This means that on 10x 3D data we only have 2 private local data, not the whole sample. We assume that 2x 3D data is a 3D LR image, and 10x 3D local parts of the data are 3D HR images. Thus we have 2 parts of HR image, not the entire HR image. This leads to the idea that we study an approach to perform super-resolution image guided by HR local subimage of 3D materials data. From the HR local subimage, the LR local subimage can be retrieved. These HR and LR sub-images are to be used as reference images for training step of super-resolution.

[1] https://www.psi.ch/en/sls/tomcat
[2] https://www.psi.ch/en/sls

By extracting these subimages into small patches, we obtain a set of HR patches and the corresponding LR patches as the dictionaries. From these points of view, this thesis focuses on the research and development of example-based SR methods. In this thesis, we propose new super-resolution methods based on Generalized Gaussian Mixture Models. These methods will be presented in Chapter 4 and Chapter 5.



(a) (b) (c) (d)

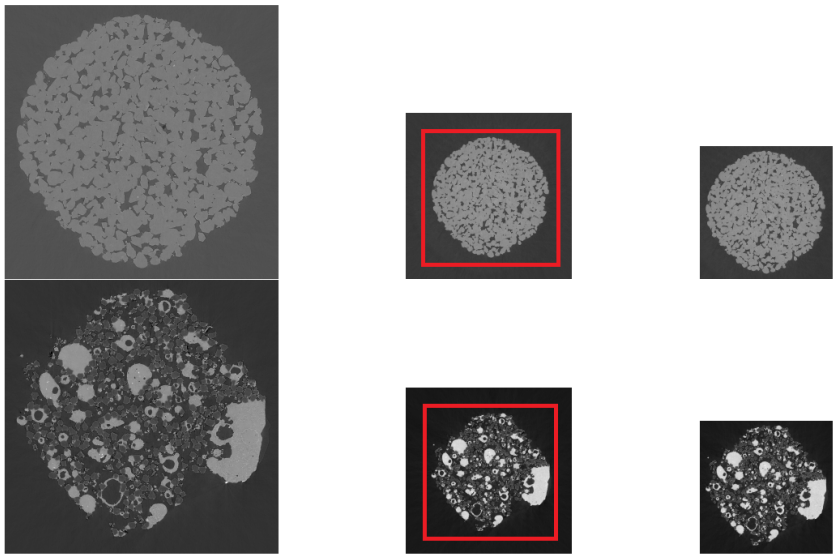
?figurename? 3.4: Cross sections of Fontainebleau sandstone with different resolutions. a) 2x MAG with pixel size of $3.25\mu m$, b) 4x MAG with pixel size of $1:63\mu m$, c) and d) 10x MAG with pixel size of $0.65\mu m$

In this section, we investigate some of the challenges and solutions of super-resolution 3D materials data.

3.2.1 Detecting local HR and LR images via Scale-invariant feature transform (SIFT) algorithm

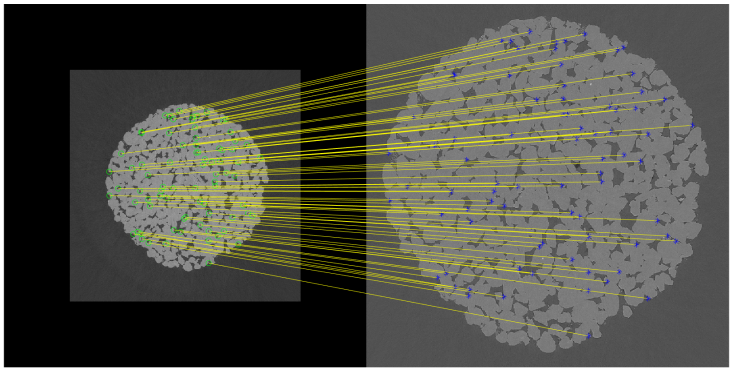
In materials science, 3D data is acquired using a synchrotron microtomography for different magnifications at different times. Therefore, the positions of the subject are

changed between the HR and LR images. This is illustrated in detail in Figure 3.5. Assume that we consider a cross section with 2x MAG as a 2D LR image and a cross section with 4x MAG as a 2D HR image. In Figure 3.5, the first column shows the HR images of FS and Sic-Diamond, and the second column displays the LR images, which we acquired from synchrotron microtomography. One can easily notice that the LR image has more black areas than the HR image in the bounder of the sample. Additionally, the 10x MAG data is obtained with 2 separate local parts, not the whole data as shown in Figure 3.4. These two arguments lead us to the important task before image super-resolution, which is to find the position corresponding to the HR image in the LR image in case of 4x MAG or to detect the HR parts in the LR image in case of 10x MAG. To detect HR and LR images, we refer to the Scale-invariant feature transform (SIFT) algorithm [61].



?figurename? 3.5: Top: Material "FS", bottom: Material "SiC Diamonds". First column: Ground-truth HR image, second column: Input LR image, third column: Match LR images. The red lines in the second column are the boundaries of the LR image that correspond to the HR image.

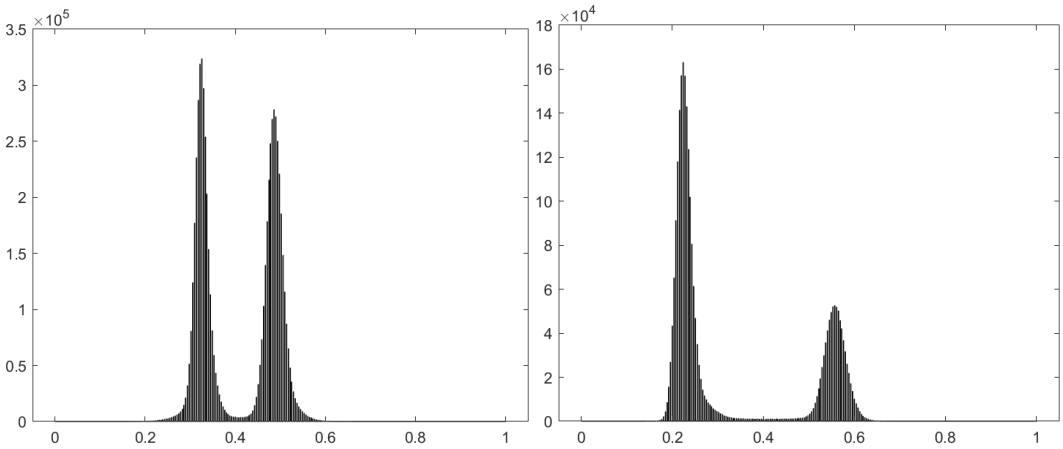
For more than a decade, the SIFT [61] has arguably been the most popular keypoint detection and matching method. The SIFT algorithm proposed by David Lowe consists of a keypoint detection which is independent towards image rotation, scale change, affine transformation, intensity variation, and viewpoint change. Using the SIFT algorithm, we determine all the match points between the HR and LR images as shown in Figure 3.6. Then, the boundary of the LR image will be found by selecting the minimum and maximum coordinates of the keypoints. Finally, we obtain the 2D LR image which matches with the 2D HR image. The matched LR image is given in the third column in Figure 3.5.



?figurename? 3.6: Matched points between the HR and LR image by using SIFT

3.2.2 Modeling contrast change parameters

In practice, the high-resolution image and its low-resolution correspondence were acquired under different conditions. Therefore, they may have different brightness and contrast. In fact, the LR and HR images of the FS and sic Diamond data in Figure 3.5 clearly show this comment. In addition to that, Figure 3.7 shows the histogram of the HR and LR images of FS material data. It shows that the contrasts of the HR and LR images are different.



?figurename? 3.7: Histogram of the images of the FS material. The image on the left is the histogram of the high-resolution image, and the image on the right is its low-resolution correspondence.

Notably, there exists a celebrated approach which has been explored, for instance, in midway image equalization [22, 36]. The idea of mid-way equalization [22] is to give a pair of images the same intermediate histogram. However, in our project, our aim is to reconstruct the HR image based on the guided HR part of the

LR image. Therefore, the basic idea is to determine an affine transformation such that the LR image applied to this transformation has the same contrast as the HR image.

Let $C(t) = \alpha t + \beta$ with $\alpha, \beta \in \mathbb{R}$, $\alpha > 0$ be an affine contrast change function such that $\tilde{Y} = C(Y)$ has the same contrast as the HR image X, where Y denotes the LR image. In the case of data with two gray-level regions, such as our material data, we can find the maximum values in the two regions of the histogram. We denote these two values of HR image by m_H, M_H, and LR image by m_L, M_L respectively. Thanks to the property of the affine function C, we get the following equation:

$$m_H = \alpha m_L + \beta$$
$$M_H = \alpha M_L + \beta.$$

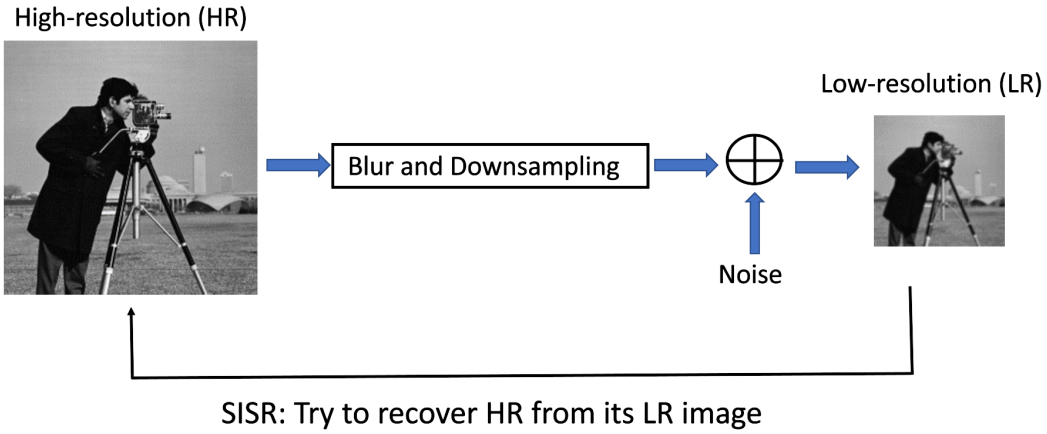
The affine contrast change function can be easily found by solving the above system of linear equations. In this manuscript, we propose a method to reconstruct the HR image without estimating the parameters α, β of the affine contrast change function C as a pre-processing step. This statement is discussed in chapter 4.

3.2.3 Modeling super-resolution operator

The last challenge of super-resolution materials image processing is the SISR with unknown corruption, the major challenge in the SR community. The SISR aims to reconstruct an HR image from a single LR input image. In general, the LR image is generated by

$$X_L = AX_H + \varepsilon \tag{3.2.1}$$

where X_L is the low-resolution image, X_H is the high-resolution image, ε is a white Gaussian noise with the standard derivation σ, and A is an unknown super-resolution operator.



?figurename? 3.8: Sketch of the overall framework of SR.

In the following, we describe the estimation process of the forward operator A

for super-resolution. For estimation, we assume that we have given a registered pair $(\tilde{X}_H, \tilde{X}_L)$ of a high-resolution and a low-resolution image. We assume that our forward operator is given by

$$A = SH. \tag{3.2.2}$$

Where H is a blur kernel and S is a downsampling operator. The downsampling operator $S : \mathbb{R}^{m_1,n_1} \to \mathbb{R}^{m_2,n_2}$ ($m_1 > m_2$ and $n_1 > n_2$) is given by

$$S = \frac{m_2 n_2}{m_1 n_1} \mathcal{F}^{-1}_{m_2,n_2} D \mathcal{F}_{m_1,n_1}, \tag{3.2.3}$$

where $D : \mathbb{R}^{m_1,n_1} \to \mathbb{R}^{m_2,n_2}$ and for $x \in \mathbb{R}^{m_1,n_1}$ the (i,j)-th entry of $D(x)$ is given by

$$\begin{cases} x_{i,j}, & \text{if } i \leq \frac{m_2}{2} \text{ and } j \leq \frac{n_2}{2}, \\ x_{i+m_1-m_2,j}, & \text{if } i > \frac{m_2}{2} \text{ and } j \leq \frac{n_2}{2}, \\ x_{i,j+n_1-n_2}, & \text{if } i \leq \frac{m_2}{2} \text{ and } j > \frac{n_2}{2}, \\ x_{i+m_1-m_2,j+n_1-n_2}, & \text{if } i > \frac{m_2}{2} \text{ and } j > \frac{n_2}{2}. \end{cases}$$

$\mathcal{F}_{m_1,n_1}$ is the discrete two-dimensional Fourier transform (DFT) defined by $\mathcal{F}_{m_1,n_1} := \mathcal{F}_{n_1} \otimes \mathcal{F}_{m_1}$, where $\mathcal{F}_{m_1} = \left(\exp\left(-2\pi i k l / m_1\right)\right)_{k,l=0}^{m_1-1}$.

The question now is how to estimate the blur kernel H. We assume that we have given a reference HR image $\tilde{X}_H \in \mathbb{R}^{m_1,n_1}$ and a corresponding LR image $\tilde{X}_L \in \mathbb{R}^{m_2,n_2}$ related by $\tilde{X}_L = A\tilde{X}_H = S(H * \tilde{X}_H))$, where the blur kernel H is unknown. By the definition of downsampling operator S, we apply the DFT on both sides of $\tilde{X}_L = S(H * \tilde{X}_H))$, we obtain the following

$$\mathcal{F}_{m_2,n_2}\left(\tilde{X}_L\right) = \frac{m_2 n_2}{m_1 n_1} D\left(\mathcal{F}_{m_1,n_1}\left(H * \tilde{X}_H\right)\right) = \frac{m_2 n_2}{m_1 n_1} D\left(\mathcal{F}_{m_1,n_1}(H)\right) \odot D\left(\mathcal{F}_{m_1,n_1}\left(\tilde{X}_H\right)\right), \tag{3.2.4}$$

where $\odot$ indicates the element-wise product. Now, we can conclude that

$$D\left(\mathcal{F}_{m_1,n_1}(H)\right) = \frac{m_1 n_1}{m_2 n_2} \mathcal{F}_{m_2,n_2}\left(\tilde{X}_L\right) \oslash D\left(\mathcal{F}_{m_1,n_1}\left(\tilde{X}_H\right)\right), \tag{3.2.5}$$

where $\oslash$ is the element-wise quotient. In practice, we stabilize this quotient by increasing the absolute value of $D\left(\mathcal{F}_{m_1,n_1}\left(\tilde{X}_H\right)\right)$ by 10^{-5} while retaining the phase. Thus, assuming that the high-frequency part of H is negligible (i.e., that $D^{\mathrm{T}} D H = H$), we can approximate $\mathcal{F}_{m_1,n_1}(H)$ by

$$\mathcal{F}_{m_1,n_1}(H) \approx \frac{m_1 n_1}{m_2 n_2} D^{-1}\left(\mathcal{F}_{m_2,n_2}\left(\tilde{X}_L\right) \oslash D\left(\mathcal{F}_{m_1,n_1}\left(\tilde{X}_H\right)\right)\right) \tag{3.2.6}$$

Therefore, the blur kernel H can be estimated by applying the inverse DFT

$$H \approx \frac{m_1 n_1}{m_2 n_2} \mathcal{F}^{-1}_{m_1,n_1}\left(D^{-1}\left(\mathcal{F}_{m_2,n_2}\left(\tilde{X}_L\right) \oslash D\left(\mathcal{F}_{m_1,n_1}\left(\tilde{X}_H\right)\right)\right)\right) \tag{3.2.7}$$

where $\mathcal{F}^{-1}_{m_1,n_1}$ is the inverse of $\mathcal{F}_{m_1,n_1}$.

3.3 State-of-the-art of Super-resolution method

To date, mainstream algorithms of SISR are mainly divided into three families: the interpolation-based approach, the reconstruction-based approach, and the example-based approach. We will briefly summarize their principles in the following.

Interpolation-based approach The most widely used interpolation methods in image processing and super-resolution are generally referred to as nearest neighbors, bilinear, bicubic [50], and Lanczos [27]. These methods compute the missing pixels in the HR image as averages of known pixels. However, some objects often show discontinuities, such as edges and corners. Thus, learning missing pixels leads to ringing, jagged edges, and blurred artifacts in the output image. Therefore, interpolation-based SISR methods are fast and straightforward, but suffer from accuracy deficiencies.

Reconstruction-based approach Reconstruction-based SR methods often use prior knowledge to improve the quality of reconstruction. This approach includes a variety of methods that focus on sharpening edges such as [19, 20, 28, 4, 62]. These methods motivate to design an edge smoothness prior and suppress the ringing, jagged edge artifact of the interpolation-based methods. In [19, 20], S. Dai et al. proposed methods that based on soft edge smoothness prior to extract the edges and corner points of LR image. Then they perform the continuity of these edges and combine with the interpolation results to yield the final HR image. Similarly, R. Fattal proposed a method in [28] in which missing information is learned from edge statistics. Beside that [4, 62] extend to new formulation to solve the SR problem through regularization term. They use the total variation (TV) regularizer that specifies the requirements of edge-directed filtering. These reconstruction-based SR methods provide improved results, yielding sharp edges without ringing or other artifacts. However when the magnification factor increases, the performance of these methods degrade rapidly and they are often time consuming.

Example-based approach This third approach is arguably the most successful because of its fast computation and outstanding performance. The example-based SR methods use machine learning techniques to analyze statistical relationships between the LR and corresponding HR image from an image database. These methods can be typically achieved by dividing images into small (overlapping) patches. In the upsampling procedure, the LR input image is divided into a set of patches. Then for each LR input patch, a HR patch is estimated by using the statistical parameters, that are learned from the examples contained in the dictionary.

The first study on this approach was introduced by Freeman et al. [29], based on Markov Random Field (MRF) framework. For the input LR patch, by searching the nearest neighbors in the training set, several paired HR candidate patches will be selected. These HR candidate patches subsequently are used in the MRF to reconstruct output HR patch. A disadvantage of this method is its high computational cost, due to the complex solution and the necessity of having large dictionaries, including a large variety of image patches.

To address the problem of dictionary size, many example-based SR algorithms consisting of different procedures have been proposed. The neighbor embedding method [15] proposed by Chang et al. utilized the local geometric similarities of LR and HR to restore HR patches. The general idea of this method is that each observed LR patch can be described as a linear combination of its nearest neighbors in the dictionary of LR patches. In the dictionary, we can select the HR patches corresponding to the nearest neighbors in the LR patches. Then the output HR patch can be reconstructed by a linear combination of the HR neighbors with the same linear weights obtained in LR. In this way, since patch combinations are enabled, the number of required image patch exemplars can ideally be reduced, while maintaining the same size of the dictionary. However, the requirement to determine the nearest neighbors for each patch from a large example dictionary might render the computation too expensive if the dictionary size is large and the zoom factor is high.

Another important line of example-based SR methods was initiated in sparse-coding SR (ScSR) method [84] using sparse representation of image patches over the joint dictionaries. These joint dictionaries $\{D_h, D_l\}$ include pairs of HR and LR patches such that the sparse representation α of the HR patch is the same as the sparse representation of the corresponding LR patch. Finally, for a given input LR patch y_i, its sparse representation α_i is first computed using D_l and the corresponding HR patch is estimated as $\hat{x}_i = D_h\alpha_i$. Due to joint learning, the estimated HR and LR dictionaries in ScSR are not well adapted to their respective feature spaces. Some SR algorithms were proposed in [89, 83, 85] by modifying ScSR, using coupled dictionaries. These sparse coding methods are an improvement over neighbor embedding and provide impressive results. Although the dictionary size is significantly reduced, the execution time of the ScSR method is still long.

In recent years, many deep learning (DL)-based SISR methods have been proposed in the literature. In fact, the use of neural networks has demonstrated their effectiveness in producing state-of-the-art results. Dong et al. proposed a SR algorithm named SRCNN in [24, 25] that uses a three-layered convolutional neural network (CNN) to learn end-to-end mapping between high and low-resolution images. The input of this network is a HR image interpolated from the LR image based on Bicubic interpolation method. This leads to erroneous estimates of the image structure when the inputs have been smoothed by interpolation. To overcome this issue, some research uses the deconvolution network, which was introduced by Zeiler et al. [88], to learn an upsampling operator. Fast SRCNN (FSRCNN) method proposed in [26] by Dong et al. is the pioneering work using the deconvolution

layer to reconstruct HR images directly from LR images. Using the deconvolution layer at the end of the network, the mapping is learned directly from the original low-resolution image (without interpolation) to the high-resolution one. This leads to a significant reduction in computation of the SRCNN proposal. However, the deconvolution layer enlarges feature maps by padding the images with zeros and then does the convolution, which may not be good for the results. Besides that, Shi et al. proposed an efficient sub-pixel convolution layer in [76], known as efficient sub-pixel convolutional neural network (ESPCN). ESPCN expands the multiple-channel feature maps to store the extra points to increase resolution and then rearranges these points to obtain the HR output. Thus compared with the deconvolution layer, ESPCN is capable of super-resolving LR to HR images with low computational cost at the time of training. More recently, inspired by recent work on Generative Adversarial Neural Networks (GANs) [35], C. Ledig et al. proposed a SR scheme with outstanding results in [56], called SRGAN. A GAN is a generative model in which two networks compete each other. The first network is the generator G, which generates a sample. While its adversary, the discriminator network D tries to detect if a sample is real or if it is the result of the generator. With all these adversarial networks, SRGAN significantly improves the overall visual quality of reconstruction over previous methods. Along with the promising performance that deep learning algorithms have achieved in SISR, there remain several important challenges and drawbacks such as the expensive computation of the depth architecture, the vagueness of the optimization objectives, the unknown corruption and large upsampling factor of SR.

3.4 Conclusion

This chapter provides a general overview of super-resolution processing as well as issues of super-resolution for 3D materials data. Firstly, this chapter introduces our material data and gives some challenges when performing a super-resolution for 3D materials data. The three main challenges are detecting the locality of the HR image in the corresponding LR image, changing the contrast between the HR and LR images, and modeling of an unknown downsampling operator. Second, we introduce and classify the methods that can be used to perform super-resolution. Three main groups have been presented as interpolation-based approaches, reconstruction-based approaches, and example-based approaches. Interpolation methods allow one to resample a low-resolution image at different sampling rates. However, the high-resolution image intensity values are a fixed linear function of the low-resolution image. Therefore, only low spatial frequencies are reconstructed in the reconstructed image. This leads to many artifacts in the output image. Reconstruction-based SR methods provide results without ringing or other artifacts and improve interpolation methods significantly. However, the execution time of these method is too large when the size of image and the magnification factor increases. To improve the results and execution time, example-based methods were first introduced by Freeman et al. [29]

using some machine learning techniques. In example-based approaches, the deep learning for SISR allows end-to-end reconstruction of HR image with successful results such as SRCNN, ESPCN, and SRGAN. However, these methods still have some limitations, such as high computation time, large database requirements, and convergence uncertainty. Besides that in this thesis, we focus on SR guided by high-resolution local sub-images of 3D materials data. This means that the training data is taken from the guided HR and LR part pair. Thus, the database is not large enough to train the network. Moreover we do not know the corruption downsampling operator when we only know the HR and LR image. This is also a great challenge for deep learning approaches. So in the first research of material data, we start to learn SR without deep learning approaches. The example-based approaches have been selected for research in this thesis, especially the SR approach based on joint dictionaries and sparse representations of image patches. The next chapter details some super-resolution approaches with Gaussian mixture distribution which based on patch-based approach. Then, we propose a new SR method with a generalized Gaussian mixture model. This method addresses all the requirements of super-resolution processing for materials data, such as estimating the affine contrast change function C and the unknown super-resolution operator A. In particular, this method does not need to estimate the parameters of C, A and can still achieve good results.

?CHAPTERNAME? 4

GMM and GGMM for Super-resolution

Contents

This chapter aims to present the modeling of the super-resolution problem guided by high-resolution local sub-images from an example-based point of view.

First, a brief overview of the example-based approaches used is provided. A SR method based on sparse representation of image patches over the joint dictionaries and particularly of a joint Gaussian mixture model (GMM) is followed by a state-of-the-art in order to describe the framework of this thesis.

Then an extension method is proposed using the MMSE with the GGMM to reconstruct HR image.

This chapter is organized as follows. A short summary of patch-based SR approaches is provided in Section 4.1. The MMSE estimator with GGMM [75] and EPLL method [92] are mentioned in this section. Section 4.2 proposes a method that uses the MMSE estimator for the joint GGMM, called the MMSE-GGMM method. This is a main contribution in this chapter, and it adapts the FP-EM algorithm to learn the joint GGMM. Finally, the conclusion is given in Sections 4.3.

4.1 State of the art of patch-based approach

Super-resolution is the task of reconstructing the estimate $\hat{X}_{HR}$ of a HR image X_H based on a LR observation X_L. The observed LR image is generated by

$$X_L = AX_H + \varepsilon \tag{4.1.1}$$

where ε is a white Gaussian noise with the standard derivation σ, and A is an unknown downsampling operator.

In recent years, various patch-based super-resolution image algorithms have been presented, and their efficiency with respect to image super resolution has been studied. Zoran and Weiss [92] proposed using the negative log-likelihood function of a GMM as a regularizer of the inverse problem. The estimated HR image $\hat{X}_{HR}$ is computed by solving

$$\arg\min_{X_H} \|AX_H - X_L\|^2 - \lambda \sum_{i \in I} \log p\left(X_{H,i}\right) \tag{4.1.2}$$

where p is the probability density function of the GMM and $(X_{H,i})_{i \in I}$ are the patches in the HR image. This method is called EPLL. However, EPLL requires knowledge of the operator A, which is not the case in some real applications. Therefore, we investigate the alternative approach of P. Sandeep et al. [75], which uses a joint GMM of the concatenated vectors of HR and corresponding LR patches. Each HR patch is estimated from the LR patch by using the Minimum Mean Squared Error MMSE estimator as:

$$\hat{X}_H \in \arg\min_T \mathbb{E}\|X_{HR} - T\left(X_L\right)\|_2^2 \tag{4.1.3}$$

$$= \mu_H + \Sigma_{HL}\Sigma_L^{-1}\left(X_L - \mu_L\right). \tag{4.1.4}$$

In the following, we detail the SR method using EPLL method and joint GMM with MMSE.

4.1.1 Super-resolution via Joint Mixture Models based on MMSE [75]

In this section, we briefly revisit the super-resolution by Sandeep wt al. [75] using joint Gaussian mixture models based on MMSE. Let $\{P_\theta : \theta \in \Theta\}$ be a parametric family of probability distributions with density functions $p(\cdot|\theta)$. Then, a mixture model is a probability distribution defined by the probability density function

$$F(x|\mathbf{w}, \mathbf{\Theta}) = \sum_{k=1}^{K} w_k p(x|\theta_k), \quad \mathbf{w} = (w_1, ..., w_K), \mathbf{\Theta} = (\theta_1, ..., \theta_K), \tag{4.1.5}$$

where the weights w_k are non-negative and sum up to one. If $\theta = (\mu, \Sigma)$ and $P_\theta = \mathcal{N}(\mu, \Sigma)$, we speak of a Gaussian mixture model (GMM). The (absolutely

continuous) *d-dimensional normal distribution* $\mathcal{N}(\mu, \Sigma)$ with mean $\mu \in \mathbb{R}^n$ and positive semi-definite covariance matrix $\Sigma \in \mathrm{SPD}(n)$ has the density

$$f(x|\mu, \Sigma) = (2\pi)^{-\frac{n}{2}} \operatorname{abs} \Sigma^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right). \tag{4.1.6}$$

Throughout this work, we aim to reconstruct the unknown high-resolution image X_H based on a low resolution observation X_L. Here, we assume that we have given a reference image, where the high-resolution image $\tilde{x}_H$ as well as the low-resolution observation $\tilde{x}_L$ are both known.

In this setting, Sandeep and Jacob proposed in [75] to reconstruct X_H using the following steps.

1. **Learning a joint mixture model.** In a first step, for given low-resolution patches $\tilde{x}_{L,i} \in \mathbb{R}^{\tau^2}$ of an image and their high-resolution counterparts $\tilde{x}_{H,i} \in \mathbb{R}^{q^2\tau^2}$, $q \in \mathbb{N}$, $q \geq 2$, $i = 1, ..., N$, we approximate the distribution of the joint high- and low-resolution patches $x_i = \begin{pmatrix} \tilde{x}_{H,i} \\ \tilde{x}_{L,i} \end{pmatrix} \in \mathbb{R}^n$, $n = (q^2+1)\tau^2$ by a mixture model. Here, Sandeep and Jacob use GMMs such that the resulting approximation is characterized by the parameters

$$w = (w_k)_k, \quad \mu = (\mu_k)_k, \quad \Sigma = (\Sigma_k)_k$$

with

$$\mu_k = \begin{pmatrix} \mu_{H,k} \\ \mu_{L,k} \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{H,k} & \Sigma_{HL,k} \\ \Sigma_{HL,k}^{\mathrm{T}} & \Sigma_{L,k} \end{pmatrix}.$$

2. **Estimation of the high-resolution patches using the MMSE estimator.** In the second step, we estimate the high-resolution correspondence of a given low-resolution patch $x_L \in \mathbb{R}^{\tau^2}$. For this purpose, we first select the component $k*$, such that the likelihood that x_L belongs to the k^*-th component is maximal, i.e., we compute

$$k^* = \underset{k=1,...,K}{\arg\max}\, w_k p(x_L|\mu_{L,k}, \Sigma_{L,k})$$

Afterwards, we estimate the high-resolution patch $x_H \in \mathbb{R}^{q^2\tau^2}$ as the minimum mean square estimator (MMSE) of X_H given $X_L = x_L$ for a random variable $X = (X_H, X_L) \sim P_{\theta_k}$. For Gaussian distributions, this corresponds to

$$\hat{x}_H = \mu_{H,k} + \Sigma_{HL,k}\Sigma_L^{-1}(x_L - \mu_{L,k}).$$

We give a more detailed explanation on MMSE estimators in Section 4.2.2.

3. **Reconstruction of the high-resolution image by patch averaging.** Finally, we estimate for any patch within the low-resolution image, the high-resolution correspondence. Afterwards, we reconstruct the high-resolution

image as follows: Let $x_H = (x_{k,l})_{k,l=1}^{q\tau} \in \mathbb{R}^{q\tau,q\tau}$ be a two-dimensional high-resolution patch. Then, we assign to each pixel $x_{k,l}$ the weight

$$\rho_{k,l} \coloneqq \exp\Big(-\tfrac{\gamma}{2}\big((k-\tfrac{q\tau+1}{2})^2+(l-\tfrac{q\tau+1}{2})^2\big)\Big).$$

After that, we add up for each pixel in the high-resolution image the corresponding weighted pixel values and normalize the result by dividing by the sum of the weights.

For readability, due to the combination of MMSE estimator with GMM, in the following of this thesis, we call this method is MMSE-GMM. This method outperforms the ScSR [84] algorithm with a significant margin. Instead of learning the pair of dictionaries from the reference HR-LR patch pairs as ScSR algorithm, the MMSE-GMM method learns the joint GMM from the reference images. This can improve the speed and performance of the MMSE-GMM algorithm.

4.1.2 Expected patch log-Likelihood (EPLL) method for super-resolution [92]

The second approach of patch-based SR approaches, which we would like to mention in this thesis, is the Expected patch log Likelihood (EPLL) method by Zoran and Weiss [92]. The EPLL of a given image X_H is defined under prior p as:

$$EPLL_p\left(X_H\right) = \sum_{i\in\mathcal{I}} \log p\left(\mathcal{P}_i X_H\right), \tag{4.1.7}$$

where $\mathcal{I} = \{1, ...N\}$ is a set of pixel indices; $\mathcal{P}_i : \mathbb{R}^N \to \mathbb{R}^P$ is a linear operator which extracts the i-th patch from the image X_H, and $\log p\left(\mathcal{P}_i X_H\right)$ is the likelihood of the i-th patch under the prior p. The basic idea of this method is to maximize the EPLL while the AX_H is still being close to the low-resolution image X_L. Thus the EPLL method defines the restored image as the maximum a posteriori estimate, corresponding to the following minimization problem:

$$\arg\min_{X_H} \frac{\lambda}{2}\left\|AX_H - X_L\right\|^2 - \sum_{i\in\mathcal{I}} \log p\left(\mathcal{P}_i X_H\right). \tag{4.1.8}$$

This optimization is a non-convex problem and depends on prior of the operator $\mathcal{P}_i\left(x\right)$. Thus the problem therefore becomes difficult to calculate. In order to avoid the optimization of the prior, Zoran et al. use a classical method, known as "Half Quadratic Splitting" [32, 53]. This technique introduces a set of patches $\{z_i\}_{i=1}^N \subset \mathbb{R}^P$, one for each overlapping patch $\mathcal{P}_i X_H$ in the image, and consider instead the penalized optimization problem that reads, for $\beta > 0$, as:

$$\arg\min_{X_H, z_1, ...z_N} \frac{\lambda}{2}\left\|AX_H - X_L\right\|^2 + \sum_{i\in\mathcal{I}}\left[\frac{\beta}{2}\left\|\mathcal{P}_i X_H - z_i\right\|^2 - \log p\left(z_i\right)\right]. \tag{4.1.9}$$

Equation (4.1.9) for a fixed β value is solved by an alternating optimization scheme:

- Solving X_H while keeping $\{z_i\}_{i\in\mathcal{I}}$ constant:

$$\hat{X}_H \in \arg\min_{X_H} \frac{\lambda}{2} \|AX_H - X_L\|^2 + \frac{\beta}{2} \sum_{i\in\mathcal{I}} \|\mathcal{P}_i X_H - z_i\|^2 \tag{4.1.10}$$

- Solving z_i for all $i = 1, ..., N$ given X_H:

$$\hat{z}_i \in \arg\min_{z_i} \frac{\beta}{2} \|\mathcal{P}_i X_H - z_i\|^2 - \log p\,(z_i) \tag{4.1.11}$$

Sub-problem (4.1.10) corresponds to solving a linear inverse problem with a Tikhonov regularization, and has an explicit solution often referred to as Wiener filtering:

$$\hat{X}_H = \left(\lambda A^{\mathrm{T}} A + \beta \sum_{i\in\mathcal{I}} \mathcal{P}_i^{\mathrm{T}} \mathcal{P}_i\right)^{-1} \left(\lambda A X_L + \beta \sum_{i\in\mathcal{I}} \mathcal{P}_i^{\mathrm{T}} z_i\right). \tag{4.1.12}$$

The solution to Sub-problem (4.1.11) depends on the prior p and the sub-problem is considered as a Maximum A-Posteriori estimator (MAP). In the case of Gaussian mixture model, [92] keeps only 1 component k^* of Gaussian mixture model maximizing the likelihood for the given i-th patch. With this assumption, the solution of (4.1.11) is also given by Wiener filtering:

$$z_i = \left(\Sigma_{z_i}^{k^*} + \frac{1}{\beta} Id_P\right)^{-1} \left(\Sigma_{z_i}^{k^*} \tilde{z}_i + \frac{1}{\beta} Id_P \mu_{z_i}^{k^*}\right). \tag{4.1.13}$$

In the same setting reference images and testing images of MMSE-GMM method, the EPLL method can reconstruct X_H with GMM as follows:

1. **Learning Gaussian mixture model.** In a first step, for given high-resolution patches $\tilde{x}_{H,i} \in \mathbb{R}^{q^2\tau^2}$, $q \in \mathbb{N}$, $q \geq 2$, $i = 1, ..., N$, we approximate the distribution of the joint high-resolution patches by Gaussian mixture model such that the resulting approximation is characterized by the parameters

$$w = (w_k)_k, \quad \mu_H = (\mu_{H,k})_k, \quad \Sigma = (\Sigma_{H,k})_k.$$

2. **Estimation of the patches $\{z_i\}_{i\in\mathcal{I}}$ using the MAP estimator.** In the second step, we first select the component $k*$, such that the likelihood that x_L belongs to the k^*-th component is maximal, i.e., we compute

$$k^* = \arg\max_{k=1,...,K} w_k p(x_L|\mu_{L,k}, \Sigma_{L,k}).$$

We estimate z_i for all $i \in \mathcal{I}$ while give the HR image X_H by using Equation (4.1.13).

3. **Reconstruction of the high-resolution image.** Finally, we estimate for the high-resolution image X_H by using Equation (4.1.12).

4.1.3 Conclusion

In both MMSE-GMM and EPLL-GMM methods, the GMM is used to estimate the high-resolution patches. While the MMSE learns the joint GMM of the concatenated vector of high- and low- resolution patches, the EPLL only needs to learn the GMM of the high-resolution patches. By using the knowledge of A, the EPLL can achieve better results than MMSE. However, in some cases, such as our material data, we have no knowledge of operator A. Thus we have to learn the operator A from the reference high- and low- resolution images, as discussed in Subsection 3.2.3. This means that the MMSE method can reconstruct HR image without the knowledge of operator A. Therefore, we decide to learn the SR method by extending the MMSE estimator with the joint mixture model. In the following section, we will propose a method that is based on MMSE estimator with joint generalized Gaussian mixture model. In this section, we will demonstrate that our method does not need the knowledge of operator A and is invariant with the contrast change problem.

4.2 Contribution: MMSE estimator with generalized Gaussian mixture model for super-resolution [69]

In the second step of MMSE-GMM method, the high-resolution patches is estimated thanks to the parameters of the joint GMM model. GMM can be learned by the expectation–maximization algorithm (EM) [13, 23]. In the past decade various works [2] [10] [71] have shown that generalized Gaussian mixture model (GGMM) has more flexibility to adapt to the shape of data probability density function and less sensibility for over fitting the number of classes than the GMM. In [21], C. Deledalle et al. proposed a method that uses the generalized Gaussian mixture model (GGMM) for the EPLL algorithm (EPLL-GGMM). They showed that the GGMM gets the distribution of patches better than a GMM and that it outperforms the GMM when used in the EPLL framework. However, the EPLL-GGMM model learns the parameters of the mixture model by estimating the covariance matrix with the following formula:

$$\Sigma_k = \frac{\sum_{i=1}^{N} \xi_{i,k} \tilde{x}_{H,i} \tilde{x}_{H,i}^T}{\sum_{i=1}^{N} \xi_{i,k}}. \tag{4.2.1}$$

Here $\{\tilde{x}_{H,i}\}_{i=1}^{N}$ is a set of all training HR patches and $\xi_{i,k}$ is the conditional distribution of the component k given the patch $\tilde{x}_{H,i}$. Unfortunately, this formula is not true in the GGMM setting.

In [10], Z. Boukouvalas et al. introduced a Riemannian averaged Fixed-point (RA-FP) algorithm for multivariate generalized Gaussian distribution (MGGD) parameter estimation. The RA-FP algorithm can effectively estimate the scatter matrix for any value of the shape parameter: for instance, $\beta = 1$ is the Gaussian distribution and $\beta = 0.5$ is the Laplacian distribution.

In the case of the generalized Gaussian mixture model, we generalize the algorithm from [82] for the weighted maximum likelihood estimation using the EM algorithm based on a fixed-point algorithm. This algorithm estimates the parameters of the mixture model, including the shape parameters of each component. This algorithm is called FP-EM algorithm for generalized Gaussian mixture model. In our work, we provide a method that uses the MMSE estimator for the super-resolution based on GGMM for joint HR-LR modeling, called MMSE-GGMM.

4.2.1 Parameter Estimation of Mixture Models

In this subsection, we focus on the parameter estimation of mixture models. For this purpose, we aim to employ the expectation-maximization EM algorithm. First, in Subsection 4.2.1.1, we revisit the generic form of the EM algorithm for mixture models. This algorithm contains as a subproblem the weighted maximum likelihood estimation for the corresponding parametric distribution family. Thus, we consider this subproblem in Subsection 4.2.1.2 for some elliptical distributions, namely, the Gaussian distribution, the generalized Gaussian distribution.

4.2.1.1 EM algorithm for Mixture Models

In the following, we consider the EM algorithm to estimate the parameters of the mixture models. EM algorithms were introduced in [23] and can be used for various probability distributions. We refer to [13] for an overview. Given samples $x_1, ..., x_N$ aims to minimize the negative log-likelihood function

$$\mathcal{L}(\mathbf{w}, \mathbf{\Theta}) = -\frac{1}{N} \sum_{i=1}^{N} \log(F(x_i|\mathbf{w}, \mathbf{\Theta}),$$

where F is defined as in (4.1.5). Then, the EM algorithm for mixture models reads as Algorithm 1, see e.g. [63].

As outlined in the algorithm, we have to compute the weighted maximum likelihood estimator as a sub-problem of the EM algorithm, i.e., we need to maximize

$$\ell(\theta) = \sum_{i=1}^{N} \alpha_i \log(p(x_i|\theta)) \tag{4.2.5}$$

for the underlying parametric distribution family. For readability, in the following, we denote $\alpha_{i,k}$ by α_i and θ_k by θ. In the following sections, we show how the EM algorithm can be done for various distributions.

Algorithm 1 EM Algorithm for Mixture Models

Input: $x = (x_1, ..., x_N) \in \mathbb{R}^{n \times N}$, initial estimate $\mathbf{w}^{(0)}, \mathbf{\Theta}^{(0)}$.
for $r = 0, 1, ...$ **do**
E-Step: For $k = 1, ..., K$ and $i = 1, \dots, N$ compute

$$\alpha_{i,k}^{(r)} = \frac{w_k^{(r)} p(x_i|\theta_k^{(r)})}{\sum_{j=1}^{K} w_j^{(r)} p(x_i|\theta_j^{(r)})} \tag{4.2.2}$$

M-Step: For $k = 1, ..., K$ compute

$$w_k^{(r+1)} = \frac{1}{N} \sum_{i=1}^{N} \alpha_{i,k}^{(r)}, \tag{4.2.3}$$

$$\theta_k^{(r+1)} = \arg\max_{\theta_k} \Big\{ \sum_{i=1}^{N} \alpha_{i,k}^{(r)} \log(p(x_i|\theta_k)) \Big\}. \tag{4.2.4}$$

end for

4.2.1.2 Weighted Maximum Likelihood Estimation of generalized Gaussian Distribution

In order to solve subproblem (4.2.5) within the EM algorithm for mixture models, we consider in the following the weighted maximum likelihood estimation for several elliptical distributions. Let $\mathrm{SPD}(n)$ be the set of positive-definite symmetric matrices. A probability distribution $\mathcal{E}_n(\mu, \Sigma, g)$ with $\mu \in \mathbb{R}^n$ and $\Sigma \in \mathrm{SPD}(n)$ is called an elliptical distribution if it admits a density function

$$p(x|\mu, \Sigma, g) \propto \Sigma^{-1/2} g((x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu)),$$

where $g \colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is Lebesgue measurable with

$$\int_0^\infty t^{n/2-1} g(t) \mathrm{d}t < \infty.$$

In this work, we consider Gaussian, generalized Gaussian as examples of elliptical distributions.

Gaussian distribution. A probability distribution is a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathrm{SPD}(n)$ if it has the probability density function

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\big(-\tfrac{1}{2} (x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu) \big).$$

The Gaussian distribution is an elliptical distribution with $g(t) = \exp(-\frac{1}{2}t)$. For samples $x_1, ..., x_N$ with weights $\alpha_1, ..., \alpha_N$, we can calculate the weighted maximum

likelihood estimator of μ and Σ, i.e., the solution of (4.2.5) by setting the gradients of the objective function to zero. This leads to estimators

$$\hat{\mu} = \frac{1}{\sum_{i=1}^N \alpha_i} \sum_{i=1}^N \alpha_i x_i \tag{4.2.6}$$

$$\hat{\Sigma} = \frac{1}{\sum_{i=1}^N \alpha_i} \sum_{i=1}^N \alpha_i (x_i - \hat{\mu})(x_i - \hat{\mu})^{\mathrm{T}} = \frac{1}{\sum_{i=1}^N \alpha_i} \sum_{i=1}^N \alpha_i x_i x_i^{\mathrm{T}} - \hat{\mu}\hat{\mu}^{\mathrm{T}}. \tag{4.2.7}$$

Generalized Gaussian distribution. We call a probability distribution a generalized Gaussian distribution $\mathrm{GG}(\mu, \Sigma, \beta)$ with expected value $\mu \in \mathbb{R}^n$, scatter matrix $\Sigma \in \mathrm{SPD}(n)$ and shape parameter $\beta \in \mathbb{R}_{>0}$, if it has the probability density function

$$p(x|\mu, \Sigma, \beta) = \frac{C(\beta)}{|\Sigma|^{1/2}} \exp\big(-\tfrac{1}{2}\big((x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\big)^{\beta}\big), \tag{4.2.8}$$

where the normalizing constant C is given by

$$C(\beta) = \frac{\beta\Gamma(\frac{n}{2})}{2^{n/(2\beta)}\pi^{n/2}\Gamma(\frac{n}{2\beta})}.$$

Here, Γ denotes the gamma function. For small values of the shape parameter β, we obtain heavy-tailed distributions, while large values of β mean that the corresponding generalized Gaussian distribution is flat-tailed. The generalized Gaussian distribution includes the Gaussian distribution for $\beta = 1$ and the Laplace distribution for $\beta = 1/2$. The generalized Gaussian distribution is an elliptical distribution with $g(t) = \exp(-\frac{1}{2}t^{\beta})$, $\beta > 0$.

In the literature, several algorithms have been proposed to estimate the parameters of a generalized Gaussian distribution of samples $x_1, ..., x_N$. Most of these methods are based on fixed point (FP) iterations [71, 67] or first-order computations on the Riemannian manifold of parameters [10]. For this work, we generalize the algorithm from [82] for weighted maximum likelihood estimation (4.2.5), i.e., for maximizing

$$\ell(\mu, \Sigma, \beta) = \sum_{i=1}^N \alpha_i \log(p(x_i|\mu, \Sigma, \beta)),$$

for some weights $\alpha_1, ..., \alpha_N > 0$. The following proposition gives the computations for estimating parameters of generalized Gaussian distribution.

Proposition 1. *Let* $\mathrm{GG}(\mu, \Sigma, \beta)$ *be a generalized Gaussian distribution with a density function* $q(\cdot|\mu, \Sigma, \beta)$. *Any maximizer* (μ, Σ, β) *of the function* $\ell(\mu, \Sigma, \beta)$

should satisfy the following equations:

$$\mu = \frac{\sum_{i=1}^{N} \alpha_i \delta_i^{\beta-1} x_i}{\sum_{i=1}^{N} \alpha_i \delta_i^{\beta-1}}, \tag{4.2.9}$$

$$\Sigma = \frac{\sum_{i=1}^{N} \alpha_i \delta_i^{\beta-1} (x_i - \mu)(x_i - \mu)^{\mathrm{T}}}{\sum_{i=1}^{N} \alpha_i}, \tag{4.2.10}$$

$$\beta = \beta + \rho \nabla_\beta \ell(\mu, \Sigma, \beta), \tag{4.2.11}$$

where $\delta_i = (x_i - \mu)\Sigma^{-1}(x_i - \mu)$ *and* $\rho > 0$.

?proofname? In the following, we compute the derivatives of

$$\ell(\mu, \Sigma, \beta) \coloneqq \sum_{i=1}^{N} \alpha_i \log f_\beta(x_i|\mu, \Sigma), \tag{4.2.12}$$

which is up to a constant equal to

$$\sum_{i=1}^{N} \alpha_i \log \frac{C_p(\beta)}{|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}\big((x_i - \mu)^T \Sigma^{-1}(x_i - \mu)\big)^\beta \Big) \tag{4.2.13}$$

$$\tag{4.2.14}$$

Using the notation $\delta_i = (x_i - \mu)^{\mathrm{T}} \Sigma^{-1}(x_i - \mu)$, the gradient with respect to μ is given by

$$\nabla_\mu \ell(\mu, \Sigma, \beta) = \sum_{i=1}^{N} \alpha_i \beta \Sigma^{-1}(x_i - \mu)\delta_i^{\beta-1} \tag{4.2.15}$$

Setting the gradient to zero yields the following result.

$$\mu = \frac{\sum_{i=1}^{N} \alpha_i x_i \delta_i^{\beta-1}}{\sum_{i=1}^{N} \alpha_i \delta_i^{\beta-1}}. \tag{4.2.16}$$

Plugging in the formulas

$$\frac{\partial a^{\mathrm{T}} \Sigma^{-1} b}{\partial \Sigma} = \Sigma^{-\mathrm{T}} a b^{\mathrm{T}} \Sigma^{-\mathrm{T}} \quad \text{and} \quad \frac{\partial \log|\Sigma^{-1}|}{\partial \Sigma} = \frac{\partial \log|\Sigma^{-1}|}{\partial \Sigma^{-1}} \frac{\partial \Sigma^{-1}}{\partial \Sigma} = \Sigma\Sigma^{-2} = \Sigma^{-1}$$

from [73], we obtain that

$$\nabla_\Sigma \ell(\mu, \Sigma, \beta) = \sum_{i=1}^{N} \alpha_i \Big(\tfrac{1}{2}\Sigma^{-1} - \frac{\beta}{2}\delta_i^{\beta-1} \Sigma^{-1}(x_i - \mu)(x_i - \mu)^{\mathrm{T}} \Sigma^{-1} \Big).$$

Setting the gradient to zero and multiplying by Σ from the left and from the right

yields that

$$\Sigma = \frac{\sum_{i=1}^{N} \alpha_i \delta_i^{\beta-1} (x_i - \mu)(x_i - \mu)^{\mathrm{T}}}{\sum_{i=1}^{N} \alpha_i}.$$

Finally, the gradient with respect to β is given by

$$\nabla_\beta \ell(\mu, \Sigma, \beta) = \sum_{i=1}^{N} \alpha_i \big(\nabla_\beta \log(C_n(\beta)) - \tfrac{1}{2} \nabla_\beta \delta_i^\beta \big)$$

Here, we have that $\nabla_\beta \delta_i^\beta = \delta_i^\beta \log(\delta_i)$ and $\nabla_\beta \log(C_n(\beta)) = \frac{1}{\beta} + \frac{n}{2\beta^2} \big(\psi(\frac{n}{2\beta}) + \log 2 \big)$, where ψ is the digamma function. Hence, we obtain

$$\nabla_\beta \ell(\mu, \Sigma, \beta) = \Big(\frac{1}{\beta} + \frac{n}{2\beta^2} \big(\psi(\tfrac{n}{2\beta}) + \log 2 \big) \Big) \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \alpha_i \delta_i^\beta \log(\delta_i).$$

□

The expression of maximizer (μ, Σ, β) motivate us to maximize $\ell\,(\mu, \Sigma, \beta)$ by generating a sequence $(\mu^{(r)}, \Sigma^{(r)}, \beta^{(r)})$ by fixed-point (FP) iteration.

$$\mu^{(r+1)} = \frac{\sum_{i=1}^{N} \alpha_i (\delta_i^{(r)})^{\beta^{(r)}-1} x_i}{\sum_{i=1}^{N} \alpha_i (\delta_i^{(r)})^{\beta^{(r)}-1}}, \tag{4.2.17}$$

$$\Sigma^{(r+1)} = \frac{\sum_{i=1}^{N} \alpha_i (\delta_i^{(r)})^{\beta^{(r)}-1} (x_i - \mu^{(r)})(x_i - \mu^{(r)})^{\mathrm{T}}}{\sum_{i=1}^{N} \alpha_i}, \tag{4.2.18}$$

$$\beta^{(r+1)} = \beta^{(r)} + \rho \nabla_\beta \ell(\mu^{(r)}, \Sigma^{(r)}, \beta^{(r)}), \tag{4.2.19}$$

where $\delta_i^{(r)} = (x_i - \mu^{(r)})(\Sigma^{(r)})^{-1}(x_i - \mu^{(r)})$. Note that $\nabla_\beta \ell(\mu, \Sigma, \beta)$ can be explicitly derived as

$$\nabla_\beta \ell(\mu, \Sigma, \beta) = \Big(\frac{1}{\beta} + \frac{n}{2\beta^2} \big(\psi(\tfrac{n}{2\beta}) + \log 2 \big) \Big) \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \alpha_i \delta_i^\beta \log(\delta_i).$$

The EM algorithm with the FP iteration for GGMM is called FP-EM algorithm.

Remark 1. In the unweighted case, i.e., $\alpha_1 = \ldots = \alpha_N$, the authors of [82] show convergence of the above iteration.

Remark 2. Note that for the estimation of β, several other approaches could be considered such as the Newton-Raphson method [86].

4.2.2 MMSE Estimator for Generalized Gaussian Distributions

In the following, we consider the computation of a high-resolution patch x_H given the corresponding low-resolution patch x_L and a mixture model (4.1.5), where the distribution P_{θ_k} is a generalized Gaussian distribution. In particular, we have that

the parameters are given by $\theta_k = (\mu_k, \Sigma_k, \beta_k)$ for some location $\mu_k \in \mathbb{R}^n$, a scatter matrix $\Sigma_k \in \mathrm{SPD}(n)$ and $\beta > 0$.

As in [75], we first split the parameters

$$\mu_k = \begin{pmatrix} \mu_{H,k} \\ \mu_{L,k} \end{pmatrix}, \quad \Sigma_k = \begin{pmatrix} \Sigma_{H,k} & \Sigma_{HL,k} \\ \Sigma_{HL,k}^{\mathrm{T}} & \Sigma_{L,k} \end{pmatrix}$$

into components belonging to the high- and low-resolution part of the distribution $p(\cdot|\theta_k)$. Then we select the component k^*, which best matches the low-resolution patch x_L by

$$k^* := \operatorname*{arg\,max}_{k=1,\dots,K} p(x_L|\mu_{L,k}, \Sigma_{L,k}, \beta_k).$$

Subsequently, we compute x_H as the minimum mean square estimator (MMSE) of X_H given $X_L = x_L$, where $X = (X_H, X_L) \sim P_{\theta_k}$.

Remark 3. Given a random variable $X_L \colon \Omega \to \mathbb{R}^d$ in a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, we wish to estimate a random variable $X_H \colon \Omega \to \mathbb{R}^D$, i.e., we seek an estimator $T \colon \mathbb{R}^d \to \mathbb{R}^D$ such that $\hat{X}_H = T(X_L)$ approximates X_H. A common quality measure for this task is the *mean square error* $\mathbb{E}\|X_H - T(X_L)\|_2^2$, which gives rise to the definition of the *minimum mean square estimator*

$$T_{\mathrm{MMSE}} \in \operatorname*{arg\,min}_T \mathbb{E}\|X_H - T(X_L)\|_2^2. \tag{4.2.20}$$

Under weak additional regularity assumptions on the estimator T, the Lehmann-Scheffé theorem [57] states that the general solution of the minimization problem (4.2.20) is given by

$$T_{\mathrm{MMSE}}(X_L) = \mathbb{E}(X_H|X_L).$$

In general, it is not possible to give an analytical expression of the MMSE estimator T_{MMSE}. In the case of elliptical distributions, the following theorem can be found, e.g., in [34, Theorem 8].

Theorem 1. *Assume that $X = (X_H, X_L) \colon \Omega \to \mathbb{R}^n$ has an elliptical distribution P_θ with parameters $\theta = (\mu, \Sigma, g)$, where*

$$\mu = \begin{pmatrix} \mu_H \\ \mu_L \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_H & \Sigma_{HL} \\ \Sigma_{HL}^{\mathrm{T}} & \Sigma_L \end{pmatrix}$$

Then, for each P_{X_L}-almost every x_L, we have that the conditional distribution $P_{X_H|X_L=x_L}$ is given by the elliptical distribution $P_{\hat{\theta}}$, where the parameters $\hat{\theta} = (\hat{\mu}, \hat{\Sigma}, \hat{g})$ are given by

$$\hat{\mu} = \mu_H + \Sigma_{HL}\Sigma_L^{-1}(x_L - \mu_L), \quad \hat{\Sigma} = \Sigma_H - \Sigma_{HL}\Sigma_L^{-1}\Sigma_{HL}^{\mathrm{T}}, \quad \hat{g}(t) = g(t + t_0)$$

with $t_0 = (x_L - \mu_L)^{\mathrm{T}}\Sigma_L^{-1}(x_L - \mu_L)$.

Since a generalized Gaussian distribution is also an elliptical distribution, the

MMSE estimator T_{MMSE} for the generalized Gaussian distribution is expressed as

$$T_{\mathrm{MMSE}}(X_L) = \mathbb{E}(X_H|X_L) = \mu_H + \Sigma_{HL}\Sigma_L^{-1}(x_L - \mu_L). \tag{4.2.21}$$

In our super-resolution task, we assume that the vector $\begin{pmatrix} x_H \\ x_L \end{pmatrix}$ is a realization of the random variable $\begin{pmatrix} X_H \\ X_L \end{pmatrix} \sim \mathcal{N}(\mu_{k^*}, \Sigma_{k^*})$. Then, by (4.2.21), the MMSE can be computed as

$$x_H = \mu_{H,k^*} + \Sigma_{HL,k^*}\Sigma_{L,k^*}^{-1}(x_{L,k^*} - \mu_{L,k^*}). \tag{4.2.22}$$

Finally, we can apply the reconstruction of the high-resolution image step by patch averaging as in 3, Subection 4.1.1.

In some applications, the high-resolution image and its low-resolution correspondence have different brightness and contrast. The following theorem shows that the MMSE estimator is invariant under affine transformations of the observation. In particular, the MMSE estimator compensates for the change in contrast and brightness.

Theorem 2. *Assume that $X = (X_H, X_L)\colon \Omega \to \mathbb{R}^n$ has a generalized Gaussian distribution $GG(\mu, \Sigma, \beta)$, where*

$$\mu = \begin{pmatrix} \mu_H \\ \mu_L \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_H & \Sigma_{HL} \\ \Sigma_{HL}^{\mathrm{T}} & \Sigma_L \end{pmatrix}$$

Further define the random variable $\tilde{X}_L \coloneqq \alpha_1 X_L + \alpha_2$ with $\alpha_1 > 0$, $\alpha_2 \in \mathbb{R}$. Then it holds

$$\mathbb{E}(X_H|\tilde{X}_L) = \mathbb{E}(X_H|X_L) = \mu_H + \Sigma_{HL}\Sigma_L^{-1}(X_L - \mu_L). \tag{4.2.23}$$

?proofname? The first equality holds true as it holds

$$\sigma(\tilde{X}_L) = \sigma(\{\tilde{X}_L^{-1}(]-\infty, x]) : x \in \mathbb{R}^d\}) = \sigma(\{\tilde{X}_L^{-1}(]-\infty, \alpha_1 x + \alpha_2]) : x \in \mathbb{R}^d\}) \tag{4.2.24}$$

$$= \sigma(\{X_L^{-1}(]-\infty, x]) : x \in \mathbb{R}^d\}) = \sigma(X_L). \tag{4.2.25}$$

The second equality follows directly from Theorem 1. □

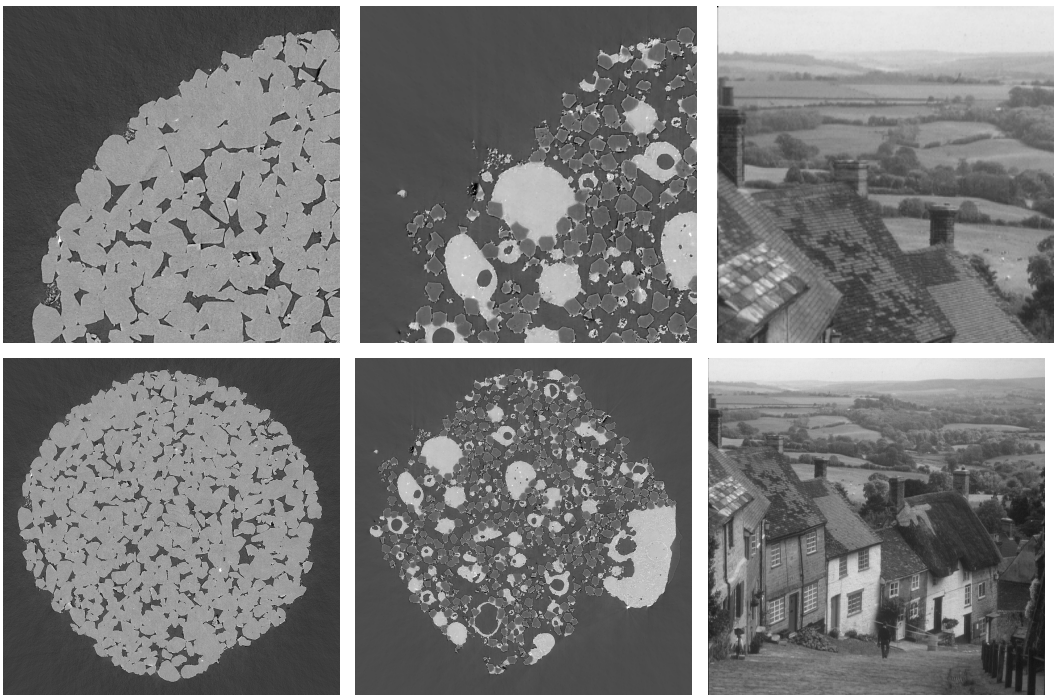
4.2.3 Numerical Results

In this section, we demonstrate the performance on some 2D images such as Gold-hill, Barbara, Camera-man, and our material data, which was presented in Chapter. 3. All the implementations were done in Matlab. We run all experiments on PlaFRIM [1] with 32 cores (4 Go/core) AMD EPYC 7452.

[1] https://www.plafrim.fr/

4.2.3.1 Dataset and quality evaluation

Generation of the dataset. In the experiment, for a given high-resolution image X_H, we now generate the low-resolution image X_L by $X_L = AX_H + \varepsilon$, where epsilon is a realization of the white Gaussian noise with standard deviation $\sigma = 0.02$. For the downsampling operator A, we use the definition in Equation (3.2.2). The blur operator H is given by a convolution with a Gaussian kernel with standard deviation 0.5.



?figurename? 4.1: Top: Images for estimating the mixture models. Bottom: Ground truth for reconstruction. First column: Material "FS", second column: Material "SiC Diamonds", third column: `goldhill` image.

Generation of the training dataset. To estimate the parameters of the mixture model, we used a part of the original HR image, such as the top left quarter as the top row in Figure 4.1 and the corresponding part of the LR image. Slices of the 3D images and the basic image for training step are given Figure 4.1. From the training part of the LR image X_L, we define a set of overlapping patches of size $\tau \times \tau$: $\{\tilde{x}_{L,i}\}_{i=1}^{N}$ with $\tilde{x}_{L,i} \in \mathbb{R}^{\tau^2}$. For the HR image X_H, we can also have a set of $q\tau \times q\tau$ HR patches $\{\tilde{x}_{H,i}\}_{i=1}^{N}$ where $\tilde{x}_{H,i} \in \mathbb{R}^{q^2\tau^2}$ corresponds to the LR patch $\tilde{x}_{L,i}$. Finally, we define a set $\{x_i\}$ with a vector $x_i \in \mathbb{R}^{\tau^2(q^2+1)}$ by concatenating the HR and LR patches as

$$x_i = \begin{pmatrix} \tilde{x}_{H,i} \\ \tilde{x}_{L,i} \end{pmatrix} \in \mathbb{R}^n. \tag{4.2.26}$$

Quality evaluation To evaluate the results, we use the Peak Signal to Noise Ratio (PSNR) value of two images X and Y. It measures the reconstruction quality of the fused image,

$$PSNR(X,Y) = 10\log_{10} 1\left(\frac{d^2}{MSE(X,Y)}\right) \tag{4.2.27}$$

where $MSE(X,Y)$ is the mean square error between X and Y with d the peak value, corresponding to the maximum fluctuation in the image.

4.2.3.2 MMSE-GGMM

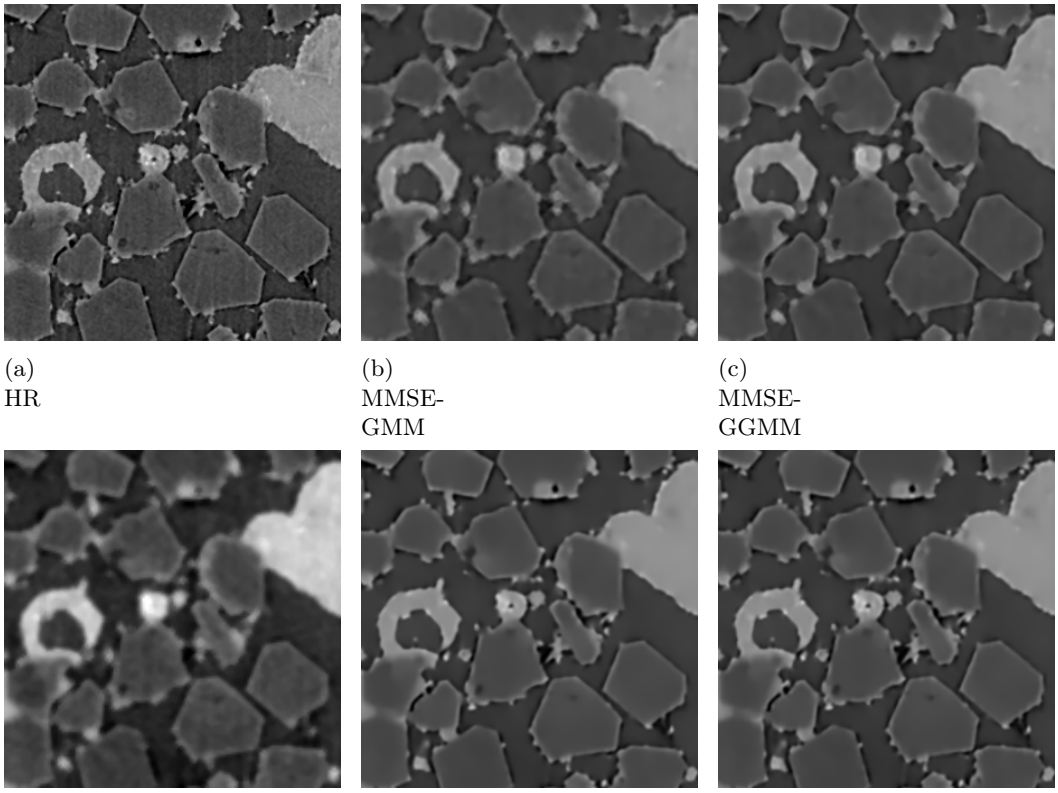
First, to underline the advantages of the proposed method MMSE-GGMM, we compare our method with the MMSE-GMM [75] and the EPLL-GMM [92] as well as EPLL-GGMM [21]. In this subsection, we use all models in standard images of size 512×512, such as Gold Hill, Camera-Man, Barbara, and material images, for the magnification factor $q = 2$. Low-resolution images are created using the operator A in (3.2.2). To estimate the parameters of the mixture model, we extract the low-resolution image with the size of the overlap patches $\tau = 4$. Thus, the corresponding high-resolution patch has size $q\tau = 8$. Then we get $N \approx 1500$ patches for the training data to learn the mixture model. Finally, high-resolution images can be reconstructed from the learned mixture model and LR images based on the MMSE as described in Subsection 4.1.1. To implement the EPLL method, we estimate the mixture models using the EM algorithm, which is introduced in [21]. Once again, EPLL requires the knowledge of the operator A in the image estimation step. For the standard image experiments, we reconstruct the HR image with the operator A defined by equation (3.2.2), called ("*EPLL with given A*"). Besides that, we implement the EPLL method with the operator A estimated as in [3] from the reference HR and LR images, called ("*EPLL with learned A*"). Since the operator A is learned from the quarter-known image, it is the same size as the reference image. Thus, we upsample the operator A by a zero padding technique to have the same size as the whole image.

Table 4.1 shows the performance of the MMSE and EPLL methods in synthetic images. Both of these methods are compared based on the GMM, LMM, GGMM with fixed shape parameter β (e.g. $\beta = 0.8, 0.4, 0.25$) and the GGMM model which learns the shape parameter based on FP-EM algorithm. The PSNR values of MMSE-GGMM with the FP algorithm are slightly higher than those of the MMSE-GMM and LMM models. To compare our EM algorithm for the GGMM model with the EM algorithm proposed by Deledalle et al. [21], we compare the MMSE method with the FP algorithm (MMSE-GGMM FP) and Deledalle's approach (MMSE-GGMM Deledalle) in Table 4.1. The MMSE-GGMM FP gives better results than the MMSE-GGMM Deledalle. Furthermore, we observed that our method can obtain results similar to those of EPLL with given A, while we do not have to learn the super-resolution operator A. Besides the results of our method are 1.5 dB higher than EPLL with the learned operator A.

		Hill	Camera	Barbara
MMSE	GMM	31.60	32.75	25.27
	LMM	31.69	32.82	25.31
	GGMM $\beta = 0.8$	31.70	32.81	25.29
	GGMM $\beta = 0.4$	31.68	32.84	25.20
	GGMM $\beta = 0.25$	31.60	32.80	25.29
	GGMM FP	**31.70**	32.86	25.33
	GGMM Deledalle	31.52	32.56	25.20
EPLL with given A	GMM	31.62	32.91	**25.39**
	LMM	31.46	32.85	25.30
	GGMM $\beta = 0.8$	31.58	32.86	25.28
	GGMM $\beta = 0.4$	31.43	32.83	25.30
	GGMM $\beta = 0.25$	31.26	32.65	25.21
	GGMM	31.58	**32.94**	25.33
EPLL with learned A	GMM	30.64	30.74	24.32
	LMM	30.57	30.88	24.29
	GGMM $\beta = 0.8$	30.56	30.79	24.25
	GGMM $\beta = 0.4$	30.42	30.76	24.31
	GGMM $\beta = 0.25$	30.40	30.67	24.19
	GGMM	30.67	30.99	24.36

?tablename? 4.1: PSNRs of the reconstructions of synthetically downsampled 2D images using either MMSE and EPLL approaches for GMM and LMM, GGMM with different shape parameter β. The magnification factor is $q = 2$.

In practice, LR and HR images were acquired under different conditions. Therefore, they may have different contrasts. Thus, we have to estimate the contrast change parameters between the LR and HR images. However, Theorem 2 proves that MMSE-GGMM is invariant for an affine contrast change. Therefore, MMSE-GGMM does not require learning these parameters and knowledge of the operator A, while the EPLL method does. To demonstrate these arguments, we perform the MMSE and the EPLL approach for our material data. Table 4.2 gives the PSNR value of the HR reconstructions of real material images using the MMSE and EPLL approach. The second column group gives the PSNR values of the EPLL approach with a learned operator A, which is estimated from the known part of the HR image. The PSNR values in the third column group are obtained from the EPLL method with the learning contrast parameters and the operator A. This operator is learned as in [3] from the observed LR image and the whole HR image of the ground truth. We call this approach is "*EPLL with given A*". The PSNRs in Table 4.2 show that our method achieves significantly better results than the EPLL method with the learning A from the known HR part. Our method gives PSNRs 1 dB lower than the EPLL method with learned A, while knowledge of the operator A is not required in our method.



(a) HR (b) MMSE-GMM (c) MMSE-GGMM

(d) LR (e) EPLL-GMM (f) EPLL-GGMM

?figurename? 4.2: Reconstructions of 2D Sic Diamonds image with magnification factor $q = 2$ by using MMSE and EPLL method.

4.3 Conclusion

This chapter proposed a new algorithm to perform image super-resolution. We extended the image super-resolution using the GMM method provided by Sandeep and Jacob [75] to the GGMM model, which is learned by the FP algorithm. We also derived a new method based on a Fixed Point approach to estimate the parameters of GGMM model, which is called FP-EM algorithm. Experiments on synthetic and material images demonstrate that our method is a promising solution for image SR.

However, the EM algorithm for these models (GMM or GGMM) becomes very slow as the number of data points becomes large and the 3D images lead to an increase in the dimensionality of the training data. Therefore, one perspective to improve this method would be to use a dimensionality reduction method.

With this in mind, the next chapter takes into account that we incorporate a dimensionality reduction within mixture models of generalized Gaussian distributions and derive an algorithm for estimating its parameters.

	MMSE			*EPLL with learned A*			*EPLL with given A*		
	GMM	LMM	GGMM	GMM	LMM	GGMM	GMM	LMM	GGMM
FS	33.09	33.32	33.35	32.25	32.31	32.39	33.61	33.86	**34.22**
Sic Diamonds	28.00	28.07	28.08	26.75	26.78	26.82	29.11	29.29	**29.45**

?tablename? 4.2: PSNRs of the reconstructions of material 2D images with contrast change problem using either MMSE estimator and EPLL approach for GMM, LMM, and GGMM. The magnification factor is $q = 2$.

?CHAPTERNAME? 5

Dimensionality reduction

Contents

5.1 Introduction

This chapter is devoted to the dimensionality reduction problem and presents our second contribution.

First, a brief overview of the dimensionality reduction problem is provided. A presentation of the principal component analysis PCA method and related state-of-the-art of dimentionality reduction are followed in Section 5.2 in order to describe more precisely the framework of this Chapter.

Next, we present our main contributions of this chapter. The two dimensionality reduction models related to GMM and GGMM will be presented, respectively, in Section 5.3 and Section 5.4. These models are proposed to combine a GMM or GGMM with a principal component analysis (PCA) and derive an EM algorithm for estimating the parameters of the arising model. Finally, these dimensionality reduction models are applied to the SR image using the MMSE estimator with the GMM and GGMM models. The results of the SR methods are presented in Section 5.5.

The conclusion is given in Sections 5.6.

5.2 State-of-the-art of dimensionality reduced data

An important motivation for dimensionality reduction is that many datasets have the property that all data points lie close to a manifold. This manifold has a much lower dimensionality than the original data space. Dimensionality reduction is therefore the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data. The standard method for dimensionality reduction is the principal component analysis (PCA) [72], which was introduced in 1901 by Karl Pearson. Based on this method, [79] introduced the probabilistic principal component analysis (PPCA) for the Gaussian distribution. It was extended to the Gaussian Mixture model in [80] with the mixture of probabilistic principal component analysers (MPPCA) model. Beside that, the authors of [11] proposed a dimensionality reduction method for Gaussian mixture model, called High-Dimensional Data Clustering (HDDC) method. It is based on the MPPCA model to replace the affine space in the PCA by the union of finitely many affine spaces using a mixture model of probabilistic PCA.

5.2.1 Principal Component Analysis (PCA) [72]

PCA is a well-known technique for dimensionality reduction. PCA can be defined as the linear projection that minimizes the mean squared distance between the data points and their projection. Based on this definition, given data samples $\mathcal{X} = \{x_1, ..., x_N\}$ in $\mathbb{R}^n$ with dimensionality n, the classical PCA finds the latent d-dimensional affine space $\{U\,t + b : t \in \mathbb{R}^d\}$, $1 \leq d \ll n$ having smallest squared distance from the samples by minimizing

$$P(U, b) = \sum_{i=1}^{N} \|(UU^{\mathrm{T}} - I_n)(x_i - b)\|^2 \tag{5.2.1}$$

for $b \in \mathbb{R}^n$ and $U \in \mathrm{St}(d, n)$, where $\mathrm{St}(d, n) \coloneqq \{U \in \mathbb{R}^{n,d} : U^{\mathrm{T}}U = I\}$ is the Stiefel manifold. By setting the derivative of P with respect to b equal to zero, the *offset* (*bias*) $b = \bar{x} \coloneqq \frac{1}{N}(x_1 + \ldots + x_N)$. So that we can reduce our attention to the

minimization with respect to $U \in \mathrm{St}(d,n)$, i.e., to the consideration of

$$P(U) = \sum_{i=1}^{N} \|(UU^{\mathrm{T}} - I_n)y_i)\|^2, \quad y_i = x_i - \bar{x}.$$

Now we can estimate matrix U with constraint $U \in \mathrm{St}(d,n)$. To enforce this constraint, we introduce a Lagrange multiplier that we will denote by $\Lambda \in \mathbb{R}^{d\times d}$. The minimization problem to $P(U)$ with constraint $U \in \mathrm{St}(d,n)$ become an unconstrained minimization of

$$\mathcal{L}(U,\Lambda) = \sum_{i=1}^{N} \|(UU^{\mathrm{T}} - I_n)y_i)\|^2 + \mathrm{tr}\left(U^{\mathrm{T}}U - I_d\right).$$

By setting the derivative of $\mathcal{L}(U,\Lambda)$ with respect to U equal to zero, this quantity will have a stationary point when

$$SU = \Lambda U$$

where $S = \sum_{i=1}^{N} y_i y_i^{\mathrm{T}}$. This implies that a minimizer can be derived explicitly as the matrix $\hat{U}$, whose columns are given by the eigenvectors corresponding to the d largest eigenvalues of the empirical covariance matrix S. This minimizer is not unique, since it holds $P(UV) = P(U)$ for any orthogonal matrix $V \in O(d)$.

In summary, the formulation of PCA is based on a linear projection of the data onto a subspace of lower dimensionality than the original data space. To apply the PCA for dimensionality reduction, we first find eigenvectors and eigenvalues of matrix S, and then compute the low-dimentional data from original data space using $\left\{U^{\mathrm{T}}(x-b) : x \in \mathbb{R}^n\right\}$. In the next subsection, we consider a reformulation of PCA, known as probabilistic PCA (PPCA). This formulat shows that PCA can also be expressed as the maximum likelihood solution of a probabilistic latent variable model.

5.2.2 Probabilistic Principal Component Analysis PPCA [79]

The probabilistic PCA model was proposed by Tipping and Bishop in [79]. This model is considered as a Gaussian model. Based on PCA's goals, the mapping of latent data to original data can be defined as a linear function of t:

$$x = Ut + b + \varepsilon, \tag{5.2.2}$$

where the latent variables are defined to be independent and Gaussian with a zero-mean unit-covariance $t \sim \mathcal{N}(0, I)$. The noise of the model is also Gaussian such that $\varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$. Thus the conditional probability distribution over x-space for a given t is $x \mid t \sim \mathcal{N}\left(Ut + b, \sigma^2 I\right)$. The probability density function of this

distribution is written as

$$p\left(x \mid t\right) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{1}{2\sigma^2}\left\|x - Ut - b\right\|^2\right\}. \tag{5.2.3}$$

To determine the parameters U, b, σ^2 by using maximum likelihood, we need an expression for the marginal distribution of original variables $p\left(x\right)$. The expression of marginal distribution $p\left(x\right)$ can be obtained thanks to the sum and product rules of probability, in the form:

$$\begin{aligned} p\left(x\right) &= \int p\left(x \mid t\right) p\left(t\right) dt \\ &= \left(2\pi\right)^{-n/2} \left|\tilde{\Sigma}\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(x - b\right)^{\mathrm{T}} \tilde{\Sigma}^{-1} \left(x - b\right)\right\}, \end{aligned} \tag{5.2.4}$$

where $\tilde{\Sigma} = UU^{\mathrm{T}} + \sigma^2 I$ is the covariance matrix. Now the parameters U, b, σ^2 can be estimated by using maximum likelihood method. Given a data set $\mathcal{X} = \{x_1, ..., x_N\}$ in $\mathbb{R}^n$ with n-dimensional, the corresponding log-likelihood function is given by

$$\begin{aligned} \mathcal{L}_P\left(U, b, \sigma^2\right) &= \sum_{i=1}^{N} \log\left(p\left(x\right)\right) \\ &\quad - \frac{N}{2}\left(n \log\left(2\pi\right) + \log\left|\tilde{\Sigma}\right| + \mathrm{tr}\left(\tilde{\Sigma}^{-1} S\right)\right), \end{aligned} \tag{5.2.5}$$

where

$$S = \frac{1}{N} \sum_{i=1}^{N} \left(x_i - b\right)\left(x_i - b\right)^{\mathrm{T}} \tag{5.2.6}$$

is the sample covariance matrix of the observed $\{x_i\}_{i=1,...,N}$. Setting the derivative of the log-likelihood function $\mathcal{L}$ with respect to b equal to zero, the bias $b = \overline{x} = \frac{1}{N}\left(x_1 + ... + x_N\right)$. The maximization of the log-likelihood is more complex with respect to U and σ^2 than b. In [79], Tipping and Bishop showed that all of the stationary points of the log-likelihood function can be written as

$$U_{ML} = W\left(\Lambda - \sigma^2 I\right)^{1/2} R, \tag{5.2.7}$$

where W is a $n \times d$ matrix whose d column vectors are principal eigenvectors of S, with the d largest corresponding eigenvalues $\lambda_1, ..., \lambda_d$ in the $d \times d$ diagonal matrix Λ in descending order of magnitude and R is an arbitrary $d \times d$ orthogonal rotation matrix. Maximizing the log-likelihood function $\mathcal{L}_P$ with respect to σ^2 gives

$$\sigma^2_{ML} = \frac{1}{n - d} \sum_{j=d+1}^{n} \lambda_j \tag{5.2.8}$$

where $\lambda_{d+1}, ..., \lambda_n$ are the smallest eigenvalues of S. So that σ^2_{ML} is the average variance associated over the lost dimensions. In practice, to implement probabilistic

PCA, we first compute the eigen-decomposition of S. Then we estimate σ^2_{ML} from (5.2.8), and finally the matrix U_{ML} from equation (5.2.7).

5.2.3 Mixture of probabilistic principal component analysers [80]

The association of a probabilistic PCA model offers attractive prospects for application to dimensionality reduction. In [80], Tipping and Bishop introduced a model that consider a mixture distribution in which the components are probabilistic PCA. This is called Mixtures of Probabilistic PCA (MPPCA) model. By using the Gaussian mixture models (GMMs) with the density function , the log-likelihood of observing the data set for such a Gaussian mixture model is:

$$\mathcal{L} = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} w_k p\left(x_n \mid k\right) \right\}, \tag{5.2.9}$$

where $p\left(x \mid k\right)$ is a single PPCA model (5.2.4) and w_k is the corresponding cluster weight, with w_k are non-negative and sum up to one for all k cluster. As a consequence, the parameters U_k, b_k, σ^2_k are now associated with each of the K mixture components. This means that the generative model (5.2.2) now requires the weight w_k, and the appropriate parameters U_K, b_k, σ^2_k of each mixture component. These parameters can be estimated by using an iterative EM Algorithm 1 for the mixture model to maximize the log-likelihood function (5.2.9) of all of the model parameters $\left\{w_k, U_k, b_k, \sigma^2_k\right\}$. The parameters θ_k in equation (4.2.4) now are considered as $\theta_k = \left\{U_k, b_k, \sigma^2_k\right\}$. At iteration r, the posterior responsibility of mixture k of x in the E-Step is given by

$$\alpha^{(r)}_{i,k} = \frac{w_k p\left(x_i \mid \theta_k\right)}{\sum_{l=1}^{K} w_l p\left(x_i \mid \theta_l\right)}, \tag{5.2.10}$$

and the mixture component weight w_k is estimated as in equation (4.2.3). The parameter b_k is updated as

$$b^{(r+1)}_k = \frac{\sum_{i=1}^{N} \alpha^{(r)}_{i,k} x_i}{\sum_{i=1}^{N} \alpha^{(r)}_{i,k}} \tag{5.2.11}$$

The transformation matrix U_k, and the noise σ^2_k can be obtained from matrix

$$S^{(r)}_k = \frac{1}{w^{(r+1)}_k} \sum_{i=1}^{N} \alpha^{(r)}_{i,k} \left(x_i - b^{(r+1)}_k\right) \left(x_i - b^{(r+1)}_k\right)^{\mathrm{T}} \tag{5.2.12}$$

in the same way as for a single PPCA model and the covariance matrices of the high-dimensional data are $\tilde{\Sigma}_k = U_k U^{\mathrm{T}}_k + \sigma^2 I$ for all $k = 1, ..., K$.

5.2.4 High-Dimensional Data Clustering (HDDC) [11]

In [11], C. Bouveyron et al. proposed a Gaussian mixture models designed for high-dimensional data, which is called High-Dimensional Data Clustering (HDDC). This model combines the ideas of subspace clustering and parsimonious modeling based on the Expectation-Maximization (EM) algorithm [23]. The EM algorithm is used to estimate the parameter of the specific subspace and the intrinsic dimension of each mixture component. Given a high-dimensional dataset $\{x_i\}_{i=1,...,N}$ in $\mathbb{R}^n$, we assume that the variables have Gaussian mixture distribution with means $\tilde{\mu}_k$ and covariance matrices $\tilde{\Sigma}_k$ for $k = 1, ..., K$. Then the covariance matrix $\tilde{\Sigma}_k$ can be factorized as:

$$\tilde{\Sigma}_k = Q_k \Lambda_k Q_k^{\mathrm{T}} \tag{5.2.13}$$

where Q_k is an orthogonal matrix containing the eigenvectors of $\tilde{\Sigma}_k$ and Λ_k is a $n \times n$ diagonal matrix containing the eigenvalues of $\tilde{\Sigma}_k$. It is further assumed that λ_k has the following form:

$$\Lambda_k = \left(\begin{array}{ccc|cc} a_{k1} & & 0 & & \\ & \ddots & & & \mathbf{0} \\ 0 & & a_{kd_k} & & \\ \hline & & & b_k & 0 \\ & \mathbf{0} & & & \ddots \\ & & & 0 & b_k \end{array}\right), \tag{5.2.14}$$

with $a_{kj} \geq b_k, j = 1, ..., d_k$ and $d_k \in \{1, ..., n-1\}$ for $k = 1, ..., K$.

We consider the affine subspace $\mathbb{E}_k$ is defined by the d_k eigenvectors associated to the eigenvalues a_{kj} and such that $\tilde{\mu}_k \in \mathbb{E}_k$. Similarly, the affine subspace $\mathbb{E}_k^{\perp}$ is such that $\mathbb{E}_k \oplus \mathbb{E}_k^{\perp} = \mathbb{R}^n$ and $\tilde{\mu}_k \in \mathbb{E}_k^{\perp}$. Let $P_k(x) = \tilde{Q}_k \tilde{Q}_k^{\mathrm{T}}(x - \tilde{\mu}_k) + \tilde{\mu}_k$ and $P_k(x)^{\perp} = \overline{Q}_k \overline{Q}_k^{\mathrm{T}}(x - \tilde{\mu}_k) + \tilde{\mu}_k$ be the projection of x on $\mathbb{E}_k$ and $\mathbb{E}_k^{\perp}$ respectively, where $\tilde{Q}_k$ is made of the d_k first columns of Q_k supplemented by $n - d_k$ zero columns and $\overline{Q}_k = Q_k - \tilde{Q}_k$.

We thus obtain a re-parameterization of the Gaussian mixture model which depends on $\tilde{\theta}_k = \{\tilde{\mu}_k, a_{kj}, b_k, Q_k, d_k\}$ for all $k = 1, ..., K$. This mixture model is denoted by $[a_{kj}, b_k, Q_k, d_k]$ and its probability density function of the k mixture component is written as:

$$\begin{aligned} p\left(x \mid \tilde{\theta}_k\right) &= \frac{1}{(2\pi)^{n/2}|Q_k \Lambda_k Q_k^{\mathrm{T}}|^{1/2}} \exp\left(-\tfrac{1}{2}(x - \tilde{\mu}_k)^{\mathrm{T}} \left(Q_k \Lambda_k Q_k^{\mathrm{T}}\right)^{-1} (x - \tilde{\mu}_k)\right) \\ &= \frac{1}{(2\pi)^{n/2}|\Lambda_k|^{1/2}} \exp\left(-\tfrac{1}{2}(x - \tilde{\mu}_k)^{\mathrm{T}} \left(Q_k \Lambda_k Q_k^{\mathrm{T}}\right)^{-1} (x - \tilde{\mu}_k)\right). \end{aligned} \tag{5.2.15}$$

To estimate this new Gaussian mixture model for the high-dimensional data x, the High-Dimensional Data Clustering (HDDC) method was introduced based on the EM Algorithm 1. At iteration r and for each $k = 1, ..., K$ and $i = 1, ..., N$ the responsibility $\alpha_{i,k}^{(r)}$ and the mixture component weight $w_k^{(r+1)}$ can be obtained as in

equation (4.2.2) and (4.2.3) respectively with $p\left(x \mid \tilde{\theta}_k\right)$. The optimization equation (4.2.4) for the new parameters can be solved by using the method of Lagrange multipliers. We refer to [11] for a demonstration of the following results for each mixture component $k = 1, ..., K$:

- The mean is estimated by:

$$\mu_k^{(r+1)} = \frac{1}{\sum_{i=1}^{N} \alpha_{i,k}^{(r)}} \sum_{i=1}^{N} \alpha_{i,k}^{(r)} x_i.$$

- The empirical covariance matrix is estimated by:

$$S_k^{(r+1)} = \frac{1}{\sum_{i=1}^{N} \alpha_{i,k}^{(r)}} \sum_{i=1}^{N} \alpha_{i,k}^{(r)} \left(x_i - \mu_k^{(r+1)}\right) \left(x_i - \mu_k^{(r+1)}\right)^{\mathrm{T}}.$$

- Sub-space $\mathbb{E}_k$: the d_k first columns of Q_k are estimated by the eigenvectors associated to the d_k largest eigenvalues λ_{kj} of the empirical covariance matrix $S_k^{(r+1)}$.

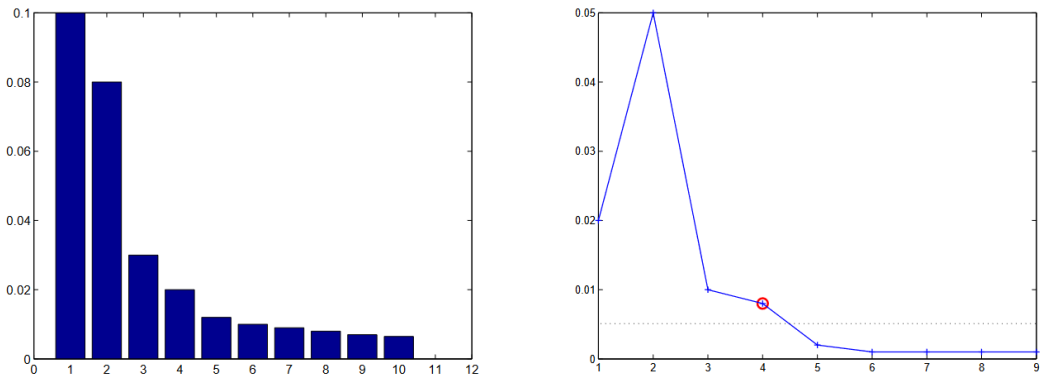
- The parameters a_{kj} are estimated by the d_k largest eigenvalues λ_{kj} of $S_k^{(r+1)}$.

- The parameter of b_k is estimated by:

$$b_k^{(r+1)} = \frac{1}{n - d_k} \left(\mathrm{tr}\left(S_k^{(r+1)}\right) - \sum_{j=1}^{d_k} \lambda_{kj} \right).$$

The main difference between the HDDC method and the others is that it estimates the d_k dimension for each cluster of the model. Within the M step of EM algorithm, the intrinsic dimensions $d_k, k = 1, ..., K$k are estimated through the scree-test of Cattell [14] which is illustrated in Figure 5.1. The selected d_k is the one for which the subsequent eigenvalues differences are smaller than the threshold (dashed line). In this paper, new Gaussian mixture models designed for high-dimensional data are introduced. It is assumed that the intrinsic dimension of each mixture component is much smaller than the one of the original space. In addition, outside the specific subspace of each group, the noise variance is modeled by a single parameter. Additional constraints can be imposed on the parameters within or between the groups in order to obtain further regularized models. This parameterization in the eigenspaces of the mixture components gives rise to an EM-based clustering method, called High-Dimensional Data Clustering (HDDC). Experiments on artificial and real datasets demonstrated the effectiveness of the different models of HDDC compared to classical Gaussian mixture models.

We will analyze and compare these methods with ours in the following section 5.3.2.



?figurename? 5.1: Plot of ordered eigenvalues of S_k (left) and differences between eigenvalues (right).

5.3 Contribution: Dimensionality-Reduced Gaussian Distribution: PCA-GMM Model [41]

The main purpose of this section is to propose our contribution in the subject of dimensionality reduction data. The first contribution is a dimensionality reduction method that combines the PCA model with the Gaussian mixture model, called PCA-GMM model. This contribution was published in the *Inverse Problems in Imaging* journal [41] in 2022.

5.3.1 PCA-Gaussian mixture model

A first idea would be to couple the GMM and the PCA model in an additive way and to minimize for data samples $\mathcal{X} = \{x_1, ..., x_N\}$ in $\mathbb{R}^n$ the function

$$F(U, \mathbf{w}, \Theta) = L\big(\mathbf{w}, \Theta | \mathcal{X}_{\text{low}}\big) + \frac{1}{2\sigma^2} P(U), \quad \sigma > 0 \tag{5.3.1}$$

for $U \in \text{St}(d, n)$, $\mathbf{w} \in \triangle_K$, $\mu_k \in \mathbb{R}^d$, $\Theta = (\theta_1, \ldots, \theta_K)$ with $\theta_k = \{\mu_k, \Sigma_k\}$ and $\Sigma_k \in \text{SPD}(d)$, $k = 1, \ldots, K$, where $\triangle_K := \left\{\mathbf{w} = (w_k)_{k=1}^K \in \mathbb{R}^K_{\geq 0} : \sum_{k=1}^K w_k = 1\right\}$ and

$$\mathcal{X}_{\text{low}} := \{U^{\text{T}} y_1, \ldots, U^{\text{T}} y_N\}, \quad y_i = x_i - \bar{x}.$$

It is important that the negative log-likelihood function L acts with respect to θ only on the lower dimensional space $\mathbb{R}^d$. The function F can be rewritten as

$$F(U, \mathbf{w}, \Theta) = -\sum_{i=1}^{N} \left(\log \left(\sum_{k=1}^{K} w_k f\big(U^{\mathrm{T}} y_i | \theta_k\big) \right) - \frac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I_n) y_i)\|^2 \right) \tag{5.3.2}$$

$$= -\sum_{i=1}^{N} \left(\log \left(\sum_{k=1}^{K} w_k f\big(U^{\mathrm{T}} y_i | \theta_k\big) \exp\big(-\tfrac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I_n) y_i)\|^2 \big) \right) \right). \tag{5.3.3}$$

However, knowing that the samples were taken from K different Gaussian distributions it makes more sense to reduce the dimension individually for each distribution. Based on the reformulation (5.3.3) and using the notation $\mathbf{U} = (U_k)_{k=1}^K$ and $\mathbf{b} = (b_k)_{k=1}^K$, we propose to minimize the following **PCA-GMM model**:

$$F(\mathbf{U}, \mathbf{b}, \mathbf{w}, \boldsymbol{\Theta}) \quad \text{subject to} \quad \mathbf{w} \in \triangle_K,\ U_k \in \mathrm{St}(d, n), \Sigma_k \in \mathrm{SPD}(d),\ k = 1, \ldots, K, \tag{5.3.4}$$

where $b_k \in \mathbb{R}^n$, $\mu_k \in \mathbb{R}^d$ and

$$F(\mathbf{U}, \mathbf{b}, \mathbf{w}, \boldsymbol{\Theta}) \coloneqq -\sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} w_k f(U_k^{\mathrm{T}} y_{ik} | \theta_k) \exp\big(-\tfrac{1}{2\sigma^2} \|(I_n - U_k U_k^{\mathrm{T}}) y_{ik}\|^2 \big) \right), \tag{5.3.5}$$

$$y_{ik} \coloneqq x_i - b_k, \quad k = 1, \ldots, K,\ i = 1, \ldots, N. \tag{5.3.6}$$

Clearly, if $U_k = U$ and $b_k = \bar{x}$ for all $k = 1, \ldots, K$, we get back to model (5.3.1).

The next lemma shows that our PCA-GMM model can be rewritten as a GMM model whose parameters incorporate those of the PCA.

Lemma 5.3.1. Let $\mu \in \mathbb{R}^d$, $\Sigma \in \mathrm{SPD}(d)$, $U \in \mathrm{St}(d, n)$, $b \in \mathbb{R}^n$ and let f be the Gaussian density function (4.1.6). Then the following relation holds true:

$$f\big(U^{\mathrm{T}}(x - b) | \mu, \Sigma\big) \exp\big(-\tfrac{1}{2\sigma^2} \|(I_n - UU^{\mathrm{T}})(x - b)\|^2 \big) = (2\pi\sigma^2)^{\frac{n-d}{2}} f(x | \tilde{\mu}, \tilde{\Sigma}), \tag{5.3.7}$$

where

$$\tilde{\mu} = \tilde{\Sigma} U \Sigma^{-1} \mu + b \in \mathbb{R}^n, \tag{5.3.8}$$

$$\tilde{\Sigma} = \big(\tfrac{1}{\sigma^2} (I_n - UU^{\mathrm{T}}) + U \Sigma^{-1} U^{\mathrm{T}} \big)^{-1} \in \mathrm{SPD}(n). \tag{5.3.9}$$

?proofname? 1. First of all, we verify that the matrices $\tilde{\Sigma}$ are well defined, i.e. that $\frac{1}{\sigma^2}(I_n - UU^{\mathrm{T}}) + U(\Sigma)^{-1}U^{\mathrm{T}}$ is invertible. Let $\tilde{U} \in \mathbb{R}^{n,(n-d)}$ such that $V \coloneqq (U|\tilde{U})$

is an orthogonal matrix. Then we obtain

$$V^{\mathrm{T}}\tilde{\Sigma}^{-1}V = V^{\mathrm{T}}(\tfrac{1}{\sigma^2}(I_n - UU^{\mathrm{T}}) + U\Sigma^{-1}U^{\mathrm{T}})V \tag{5.3.10}$$
$$= \tfrac{1}{\sigma^2}(I_n - V^{\mathrm{T}}UU^{\mathrm{T}}V) + V^{\mathrm{T}}U\Sigma^{-1}U^{\mathrm{T}}V. \tag{5.3.11}$$

Since $(V^{\mathrm{T}}U)^{\mathrm{T}} = U^{\mathrm{T}}V = (I_d|0)$, this is equal to

$$V^{\mathrm{T}}\tilde{\Sigma}^{-1}V = \tfrac{1}{\sigma^2}\left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{n-d} \end{array}\right) + \left(\begin{array}{c|c} \Sigma^{-1} & 0 \\ \hline 0 & 0 \end{array}\right) = \left(\begin{array}{c|c} \Sigma^{-1} & 0 \\ \hline 0 & \frac{1}{\sigma^2}I_{n-d} \end{array}\right) \tag{5.3.12}$$

and the last matrix is invertible.

2. We have to show that

$$\begin{aligned}
&(2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}\exp\Big(-\frac{1}{2\sigma^2}\|(I_n - UU^{\mathrm{T}})(x-b)\|^2 \\
&\quad -\frac{1}{2}(U^{\mathrm{T}}(x-b)-\mu)^{\mathrm{T}}\Sigma^{-1}(U^{\mathrm{T}}(x-b)-\mu)\Big) \\
&= (2\pi)^{-\frac{n}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\exp\Big(-\frac{1}{2}(x-\tilde{\mu})^{\mathrm{T}}\tilde{\Sigma}^{-1}(x-\tilde{\mu})\Big) \\
&= (2\pi)^{-\frac{n}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\exp\Big(-\frac{1}{2}x^{\mathrm{T}}\tilde{\Sigma}^{-1}x + \tilde{\mu}^{\mathrm{T}}\tilde{\Sigma}^{-1}x - \frac{1}{2}\tilde{\mu}^{\mathrm{T}}\tilde{\Sigma}^{-1}\tilde{\mu}\Big).
\end{aligned}$$

Straightforward calculation together with the observation that $U^{\mathrm{T}}\tilde{\Sigma}U = \Sigma$ and hence $U^{\mathrm{T}}\tilde{\Sigma}^{-1}U = \Sigma^{-1}$ gives

$$\frac{1}{2\sigma^2}\|(I_n - UU^{\mathrm{T}})(x-b)\|^2 + \frac{1}{2}(U^{\mathrm{T}}(x-b)-\mu)^{\mathrm{T}}\Sigma^{-1}(U^{\mathrm{T}}(x-b)-\mu) \tag{5.3.13}$$
$$= \frac{1}{2}x^{\mathrm{T}}\left(\tfrac{1}{\sigma^2}(I_n - UU^{\mathrm{T}}) + U\Sigma^{-1}U^{\mathrm{T}}\right)x \tag{5.3.14}$$
$$-\left(\tfrac{1}{\sigma^2}b^{\mathrm{T}}(I_n - UU^{\mathrm{T}}) + (\mu^{\mathrm{T}} + b^{\mathrm{T}}U)\Sigma^{-1}U^{\mathrm{T}}\right)x \tag{5.3.15}$$
$$+\frac{1}{2}(U^{\mathrm{T}}b+\mu)^{\mathrm{T}}\Sigma^{-1}(U^{\mathrm{T}}b+\mu) + \tfrac{1}{2\sigma^2}b^{\mathrm{T}}(I_n - UU^{\mathrm{T}})b \tag{5.3.16}$$
$$= \frac{1}{2}x^{\mathrm{T}}\tilde{\Sigma}^{-1}x - \tilde{\mu}^{\mathrm{T}}\tilde{\Sigma}^{-1}x + \frac{1}{2}\tilde{\mu}^{\mathrm{T}}\tilde{\Sigma}^{-1}\tilde{\mu}. \tag{5.3.17}$$

Finally, we see by (5.3.12) that $|\tilde{\Sigma}|^{-1} = \sigma^{-2(n-d)}|\Sigma|^{-1}$. □

By Lemma 5.3.1, we can rewrite our objective function F in (5.3.5) with $\tilde{\Theta} = (\tilde{\mu}, \tilde{\Sigma})$ defined by (5.3.8) and (5.3.9) with corresponding indices as

$$F(\mathbf{U}, \mathbf{b}, \mathbf{w}, \boldsymbol{\Theta}) = -\sum_{i=1}^{N}\log\Big(\sum_{k=1}^{K} w_k f(x_i|\tilde{\theta}_k)\Big) + (n-d)\log(\sqrt{2\pi\sigma^2}) \tag{5.3.18}$$
$$= L(\mathbf{w}, \tilde{\Theta}|\mathcal{X}) + (n-d)\log(\sqrt{2\pi\sigma^2}). \tag{5.3.19}$$

Up to the constant this is a negative log-likelihood function of a GMM. However, when minimizing this function, we have to take the constraints (5.3.9) and (5.3.8) into account. More precisely, our model in (5.3.4) can be rewritten as **PCA-GMM**

model:

$$\mathrm{F}(\mathbf{U}, \mathbf{b}, \mathbf{w}, \mathbf{\Theta}) \coloneqq L(\mathbf{w}, \tilde{\mathbf{\Theta}}|\mathcal{X}) \quad \text{subject to} \quad U_k \in \mathrm{St}(d, n),\ \mathbf{w} \in \Delta_K,\ \Sigma_k \in \mathrm{SPD}(d), \tag{5.3.20}$$

where

$$\tilde{\Sigma}_k = \left(\tfrac{1}{\sigma^2}(I_n - U_k U_k^{\mathrm{T}}) + U_k \Sigma_k^{-1} U_k^{\mathrm{T}}\right)^{-1},\ \tilde{\mu}_k = \tilde{\Sigma}_k U \Sigma_k^{-1} \mu_k + b_k, \quad k = 1, \ldots, K. \tag{5.3.21}$$

The choice of μ_k and b_k is redundant. This can be seen as follows, for any μ_k and b_k, define $\hat{\mu}_k = 0$ and $\hat{b}_k = \tilde{\mu}_k$. Then, it holds

$$\tilde{\Sigma}_k = \left(\tfrac{1}{\sigma^2}(I_n - U_k U_k^{\mathrm{T}}) + U_k \Sigma_k^{-1} U_k^{\mathrm{T}}\right)^{-1},\ \tilde{\mu}_k = \tilde{\Sigma}_k U \Sigma_k^{-1} \hat{\mu}_k + \hat{b}_k, \quad k = 1, \ldots, K \tag{5.3.22}$$

such that $\mathrm{F}(\mathbf{U}, \hat{\mathbf{b}}, \mathbf{w}, \hat{\mathbf{\Theta}}) = \mathrm{F}(\mathbf{U}, \mathbf{b}, \mathbf{w}, \mathbf{\Theta})$. Consequently, in the M-step of the EM algorithm in Section 5.3.3.1 we obtain that the update for the mean μ is given by $\mu_k = 0$.

Remark 4 (Different component dimensions)**.** So far the dimension d is the same for all components $k = 1, ..., K$. But by some simple adjustments, the PCA-GMM model can also be rewritten with $U_k \in \mathrm{St}(d_k, n)$, $\mu_k \in \mathbb{R}^{d_k}$ and $\Sigma_k \in \mathrm{SPD}(d_k)$, where the d_k are not necessarily equal for all k. However, to keep the notations as simple as possible, we will restrict our analysis to the case that $d_k = d$ for $k = 1, ..., K$. Nevertheless, all the results of this chapter can be derived analogously for other choices of d_k.

Remark 5 (Learning σ)**.** The function F in (5.3.1), resp. (5.3.5), (5.3.19) is strictly decreasing in σ. Thus it does not make sense to minimize F with respect to σ.

However, the function $\mathrm{F} = F - \frac{n-d}{2}\log(2\pi\sigma^2)$ in (5.3.20) can be optimized with respect to σ. To keep the M-step of the EM algorithm simple, we associate to each sum and in the mixture model an own σ_k, $k = 1, \ldots, K$ such that $\tilde{\Sigma}$ in (5.3.21) becomes

$$\tilde{\Sigma}_k = \left(\tfrac{1}{\sigma_k^2}(I_n - U_k U_k^{\mathrm{T}}) + U_k \Sigma_k^{-1} U_k^{\mathrm{T}}\right)^{-1}. \tag{5.3.23}$$

In this case, we use the notation $\sigma = (\sigma_k)_{k=1}^K$.

5.3.2 Comparison with state-of-the-art of dimensionality reduced data

There are several relations of the PCA-GMM model to other models proposed in the literature, in particular to mixtures of probabilistic PCAs (MPPCA) [80], high dimensional data clustering (HDDC) [11] and high-dimensional mixture models for unsupervised image denoising (HDMI) [45]. In the following, we shortly review

these methods and comment on similarities and differences to the PCA-GMM model.

For understanding the relation to other models, we first need the following reformulation of the covariance matrices $\tilde{\Sigma}$ from the PCA-GMM model. We have as in (5.3.12) for matrices $\tilde{\Sigma}$ of the form (5.3.9) and an orthogonal matrix $V = (U|\tilde{U})$ that

$$V^{\mathrm{T}}\tilde{\Sigma}V = \left(\begin{array}{c|c} \Sigma & 0 \\ \hline 0 & \sigma^2 I_{n-d} \end{array}\right), \tag{5.3.24}$$

so that

$$\left\{\tilde{\Sigma} = \left(\tfrac{1}{\sigma^2}(I_n - UU^{\mathrm{T}}) + U\Sigma^{-1}U^{\mathrm{T}}\right)^{-1} : U \in \mathrm{St}(d,n),\, \Sigma \in \mathrm{SPD}(d)\right\} \tag{5.3.25}$$

$$= \left\{Q^{\mathrm{T}}\left(\begin{array}{c|c} \mathrm{diag}(\lambda) & 0 \\ \hline 0 & \sigma^2 I_{n-d} \end{array}\right) Q :\, Q \in O(n),\, \lambda \in \mathbb{R}^d_{>0}\right\}. \tag{5.3.26}$$

As outlined in Remark 5, the σ can either be fixed a priori, or optimized within the EM algorithm, as later outlined in Section 5.3.3.1, simultaneously with the other parameters.

In [80], Tipping and Bishop propose mixture models of probabilistic PCAs (MPPCA), which are GMMs of the form

$$p(x) = \sum_{k=1}^{K} w_k f(x_i|\tilde{\mu}_k, \tilde{\Sigma}_k), \tag{5.3.27}$$

where

$$\tilde{\Sigma}_k = U_k U_k^{\mathrm{T}} + \sigma_k^2 I_n, \quad U_k \in \mathrm{St}(d_k, n).$$

Here, the parameters σ_k are optimized simultaneously with the w_k and U_k via the EM algorithm. Hence, skipping the index, instead of minimizing over (5.3.26), they minimize over sets of the form

$$\left\{Q^{\mathrm{T}}\left(\begin{array}{c|c} (1+\sigma^2)I_d & 0 \\ \hline 0 & \sigma^2 I_{n-d} \end{array}\right) Q : Q \in O(n)\right\}. \tag{5.3.28}$$

Since this form of the covariance matrices is very restrictive, Bouveyron, Girard and Schmid generalized MPPCA in [11] to a model called high dimensional data clustering (HDDC). Again, they minimize a special GMM (5.3.27), but here the covariances are given by

$$\tilde{\Sigma}_k = U_k \mathrm{diag}(\lambda_k) U_k^{\mathrm{T}} + \sigma_k^2 I_n, \quad U_k \in \mathrm{St}(d_k, n), \lambda_k \in \mathbb{R}^{d_k}_{>0}.$$

As for the MPPCA, the parameters are optimized via the EM algorithm. For deriving it, it is important that the parameters σ_k are not fixed a priori but are

optimized within the EM algorithm. Skipping the index, instead of minimizing over (5.3.26) or (5.3.28), this corresponds to a minimization over

$$\left\{ Q^{\mathrm{T}} \left(\begin{array}{c|c} \operatorname{diag}(\lambda) + \sigma^2 I_d & 0 \\ \hline 0 & \sigma^2 I_{n-d} \end{array} \right) Q : Q \in O(n), \lambda \in \mathbb{R}^d_{>0} \right\}. \tag{5.3.29}$$

In contrast to (5.3.26), where the diagonal values λ are required to be strictly greater than 0, the diagonal values $\lambda + \sigma^2$ in (5.3.29) are automatically strictly greater than σ^2. Consequently, the PCA-GMM model is more general than HDDC. Note that HDDC model contains the so-called mixture factor analysis [64] as a special case. Here also the alternating expectation conditional maximization algorithm [66] is applicable [91], which is an improved version of the EM algorithm.

Finally, Houdard, Bouveyron and Delon proposed in [45] a model selection algorithm for the dimensions d_k. For this, they propose a model called HDMI, where the only difference to HDDC is, that σ is a priori fixed. They derive as an intermediate step a corresponding EM algorithm in [45, Proposition 2]. Unfortunately, the M-step only ensures that $\lambda > -\sigma^2 1_d$ and not $\lambda > 0$, such that the calculations appear to be not fully correct. However, the final model selection algorithm again ensures that $\lambda > 0$ such that this seems not to be a problem in [45].

5.3.3 Minimization Algorithm

We propose to minimize F in (5.3.20) based on the EM algorithm, where we have to take the special structure of $\tilde{\mu}_k \in \mathbb{R}^n$ and $\tilde{\Sigma}_k \in \mathrm{SPD}(n)$ in (5.3.21) into account to work indeed in the lower d-dimensional space. This requires the solution of a special inner minimization problem within the M-Step of the EM algorithm. We describe the EM algorithm for our PCA-GMM model in Subsection 5.3.3.1. In particular, we will see that the M-Step of the algorithm requires the minimization of functions $G_k(U, b)$, $k = 1, \ldots, K$, of the same structure. We prove that these functions have indeed a global minimizer. In particular, these functions do not depend on the large number of input data x_i, $i = 1, \ldots, N$. Therefore it turns out that the E-step of the algorithm is the most time consuming one. We propose to find at least a local minimizer of G by the (inertial) proximal alternating linearized minimization (PALM) in Section 5.3.3.2 and provide convergence results.

5.3.3.1 EM Algorithm for PCA-GMM

For our setting, we obtain a special EM algorithm described in Algorithm 2. Note that the E-Step of Algorithm 2 requires only the mean and covariance matrix in $\theta_k^{(r)}$, $k = 1, \ldots, K$ with respect to the smaller space $\mathbb{R}^d$.

A convergence analysis of the EM algorithm via Kullback-Leibler proximal point algorithms was given in [17, 18], see also [54] for a nice review. The authors showed that the objective function decreases for the iterates of the algorithm. Hence we obtain the following corollary.

Algorithm 2 EM Algorithm for PCA reduced Mixture Models

Input: $\mathcal{X} = (x_1, ..., x_N) \in \mathbb{R}^{n,N}$, initialization $\mathbf{U}^{(0)}$, $\mathbf{b}^{(0)}$, $\mathbf{w}^{(0)}$, $\theta^{(0)} = (\mu^{(0)}, \mathbf{\Sigma}^{(0)})$.
for $r = 0, 1, ...$ **do**

E-Step: For $k = 1, ..., K$ and $i = 1, \dots, N$ compute

$$\alpha_{i,k}^{(r)} = \frac{w_k^{(r)} f(x_i|\tilde{\theta}_k^{(r)})}{\sum_{j=1}^K w_j^{(r)} f(x_i|\tilde{\theta}_j^{(r)})} \tag{5.3.30}$$

$$= \frac{\frac{w_k^{(r)}}{(\sigma_k^{(r)})^{n-d}} \exp\left(-\frac{1}{2(\sigma_k^{(r)})^2}\|(I_n - U_k^{(r)}(U_k^{(r)})^{\mathrm{T}})y_{i,k}^{(r)}\|^2\right) f\left((U_k^{(r)})^{\mathrm{T}} y_{i,k}^{(r)}|\theta_k^{(r)}\right)}{\sum_{j=1}^K \frac{w_j^{(r)}}{(\sigma_j^{(r)})^{n-d}} \exp\left(-\frac{1}{2(\sigma_j^{(r)})^2}\|(I_n - U_j^{(r)}(U_j^{(r)})^{\mathrm{T}})y_{i,k}^{(r)}\|^2\right) f\left((U_j^{(r)})^{\mathrm{T}} y_{i,k}^{(r)}|\theta_j^{(r)}\right)}, \tag{5.3.31}$$

$$y_{i,k}^{(r)} = x_i - b_k^{(r)}. \tag{5.3.32}$$

M-Step: For $k = 1, ..., K$ compute

$$w_k^{(r+1)} = \frac{1}{N} \sum_{i=1}^N \alpha_{i,k}^{(r)}, \tag{5.3.33}$$

$$(U_k^{(r+1)}, b_k^{(r+1)}, \sigma_k^{(r+1)}, \theta_k^{(r+1)}) = \underset{U,b,\mu,\Sigma}{\arg\max} \sum_{i=1}^N \alpha_{ik}^{(r)} \log(f(x_i|\tilde{\theta}_k)) \tag{5.3.34}$$

$$\text{subject to} \quad U_k \in \mathrm{St}(d,n), \Sigma_k \in \mathrm{SPD}(d) \tag{5.3.35}$$

$$\text{with } \tilde{\theta}_k = (\tilde{\mu}_k, \tilde{\Sigma}_k) \text{ as in (5.3.21).} \tag{5.3.36}$$

end for

Corollary 1. For the iterates $\left(\mathbf{U}^{(r)}, \mathbf{b}^{(r)}, w^{(r)}, \theta^{(r)}\right)_r$ generated by Algorithm 2 the objective function F is decreasing.

The interesting step is the second M-Step which requires again the maximization of a function. Based on (4.2.6) and (4.2.7) we can prove the following proposition.

Proposition 1. Assume that $n + 1$ of the points x_i, $i = 1, ..., N$ are affinely independent.

Further, let f be the Gaussian density function (4.1.6) and $\alpha_i \in \mathbb{R}_{\geq 0}$, $i = 1, \dots, N$.
i) For fixed σ^2, a solution of

$$\underset{U,b,\mu,\Sigma}{\arg\max} \sum_{i=1}^N \alpha_i \log(f(x_i|\tilde{\theta})) \tag{5.3.37}$$

with $\tilde{\theta} = (\tilde{\mu}, \tilde{\Sigma})$ of the form (5.3.8) and (5.3.9) is given by

$$\hat{\mu} = 0, \quad \hat{\Sigma} = \frac{1}{w}\hat{U}^{\mathrm{T}} S \hat{U}, \quad \text{and} \quad \hat{b} = \frac{1}{w} m, \tag{5.3.38}$$

where

$$m = \sum_{i=1}^{N} \alpha_i x_i, \quad C = \sum_{i=1}^{N} \alpha_i x_i x_i^{\mathrm{T}}, \quad w = \sum_{i=1}^{N} \alpha_i, \quad S = C - \frac{1}{w} m m^{\mathrm{T}}, \tag{5.3.39}$$

and

$$\hat{U} \in \underset{U \in \mathrm{St}(d,n)}{\arg\min}\, G(U). \tag{5.3.40}$$

Here

$$G(U) := -\frac{1}{\sigma^2} \mathrm{tr}(U^{\mathrm{T}} S U) + w \log(|U^{\mathrm{T}} S U|). \tag{5.3.41}$$

ii) If σ is learned, we have

$$\hat{\sigma}^2 = \tfrac{1}{w(n-d)} \left(\mathrm{tr}(S) - \mathrm{tr}(\hat{U}^{\mathrm{T}} S \hat{U}) \right), \tag{5.3.42}$$

and G from (5.3.41) is replaced by

$$G(U) := (n-d) \log \left(\mathrm{tr}(S) - \mathrm{tr}(U^{\mathrm{T}} S U) \right) + \log(|U^{\mathrm{T}} S U|). \tag{5.3.43}$$

Note that w in the proposition is defined in another way than in the first M-step, more precisely, the factor $\frac{1}{N}$ is skipped. Before presenting the proof of the proposition, we give the following remark.

Remark 6. By definition of C in Proposition 1 we have that

$$S = \sum_{i=1}^{N} \alpha_i (x_i - \tfrac{1}{w} m)(x_i - \tfrac{1}{w} m)^{\mathrm{T}}. \tag{5.3.44}$$

Since $n+1$ of the points x_i, $i = 1, ..., N$, are affinely independent, S is symmetric positive definite. In particular, it holds for G from (5.3.41) or (5.3.43) that $G(U) > -\infty$ for any $U \in \mathrm{St}(d,n)$. Further, since the function G is continuous and the Stiefel manifold is compact, we can conclude, that G has a global minimizer.

Proof of Proposition 1. i) Let σ be fixed. Using (5.3.7), we have for fixed U and b, as in the classical GMM, see (4.2.6) and (4.2.7), that the maximizer in (5.3.37) with respect to μ and Σ fulfills

$$\mu = \frac{1}{w} \sum_{i=1}^{N} \alpha_i U^{\mathrm{T}} (x_i - b) = \frac{1}{w} (U^{\mathrm{T}} m - w U^{\mathrm{T}} b), \tag{5.3.45}$$

$$\Sigma = \frac{1}{w} \sum_{i=1}^{N} \alpha_i \left(U^{\mathrm{T}} (x_i - b) - \mu \right) \left(U^{\mathrm{T}} (x_i - b) - \mu \right)^{\mathrm{T}} \tag{5.3.46}$$

$$= \frac{1}{w} \sum_{i=1}^{N} \alpha_i \left(U^{\mathrm{T}} (x_i - \frac{1}{w} m) \right) \left(U^{\mathrm{T}} (x_i - \frac{1}{w} m) \right)^{\mathrm{T}} = \frac{1}{w} U^{\mathrm{T}} S U. \tag{5.3.47}$$

By Lemma 5.3.1, the negative objective function in (5.3.37) is given by

$$2\tilde{G}(U,b) = G_1(U,b) + G_2(U,b) + w\log(|\Sigma|) + \text{const}, \tag{5.3.48}$$

$$G_1(U,b) = \tfrac{1}{\sigma^2}\sum_{i=1}^{N}\alpha_i(x_i-b)^{\mathrm{T}}(I_n - UU^{\mathrm{T}})(x_i-b) + w(n-d)\log(\sigma^2) \tag{5.3.49}$$

$$G_2(U,b) = \sum_{i=1}^{N}\alpha_i\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b+\mu)\right)^{\mathrm{T}}\Sigma^{-1}\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b+\mu)\right). \tag{5.3.50}$$

In the following, we use const as a generic constant which has values independent of μ, Σ, U and b. The linear trace operator $\mathrm{tr} : \mathbb{R}^{d,d} \to \mathbb{R}$ fulfills $x^{\mathrm{T}}Ay = \mathrm{tr}(Axy^{\mathrm{T}})$ and in particular $x^{\mathrm{T}}UU^{\mathrm{T}}x = \mathrm{tr}(U^{\mathrm{T}}xx^{\mathrm{T}}U)$. Using this property we obtain

$$G_2(U,b) = \mathrm{tr}\Big(\Sigma^{-1}\underbrace{\sum_{i=1}^{N}\alpha_i\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b+\mu)\right)\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b+\mu)\right)^{\mathrm{T}}}_{=\Sigma}\Big) = \mathrm{tr}(I). \tag{5.3.51}$$

Thus, the only term in (5.3.48) which depends on b and U is G_1. Further, minimizing G_1 is equivalent to minimizing

$$g_1(U,b) \coloneqq \sum_{i=1}^{N}\alpha_i(x_i-b)^{\mathrm{T}}(I_n - UU^{\mathrm{T}})(x_i-b).$$

For fixed U, we can minimize g_1 with respect to b by setting the gradient to 0. Since g_1 is convex in b this is equivalent for being a global minimizer. This yields

$$0 = \sum_{i=1}^{N}\alpha_i(I_n - UU^{\mathrm{T}})(b-x_i)$$

which is equivalent to

$$0 = (I_n - UU^{\mathrm{T}})(wb - m).$$

In particular, $b = \frac{1}{w}m$ is a global minimizer of g_1 resp. G_1, and it is independent of U. Using this, we get

$$\mu = \frac{1}{w}(U^{\mathrm{T}}m - wU^{\mathrm{T}}b) = 0.$$

Minimizing G_1 with respect to U for $b = \frac{1}{w}m$ is equivalent to minimizing

$$G_1(U, \frac{1}{w}m) = -\tfrac{1}{\sigma^2}\mathrm{tr}\left(U^{\mathrm{T}}SU\right) + \text{const}. \tag{5.3.52}$$

Further we have $\log\left(\left|\frac{1}{w}U^{\mathrm{T}}SU\right|\right) = \log(|U^{\mathrm{T}}SU|) + \text{const}$. Thus, by combining the above computations, we get that minimizing (5.3.48) with respect to U is equivalent

to minimizing

$$G(U) = -\frac{1}{\sigma^2}\mathrm{tr}\left(U^{\mathrm{T}}SU\right) + w\log(|U^{\mathrm{T}}SU|). \tag{5.3.53}$$

ii) Now consider the case, where σ is learned. Again by (5.3.7), the maximizer in (5.3.37) with respect to σ is given by the maximizer of

$$\sum_{i=1}^{N}\alpha_i\big(-\tfrac{1}{2\sigma^2}\|(I_n - UU^{\mathrm{T}})(x_i - b)\|^2 - (n-d)\log(\sigma)\big).$$

By setting the derivative to zero, one obtains, that

$$\sigma^2 = \tfrac{1}{w(n-d)}\sum_{i=1}^{N}\alpha_i(x_i - b)^{\mathrm{T}}(I_n - UU^{\mathrm{T}})(x_i - b).$$

Then the function in (5.3.49) modifies to

$$G_1(U,b) = w(n-d)\log\Big(\sum_{i=1}^{N}\alpha_i(x_i - b)^{\mathrm{T}}(I_n - UU^{\mathrm{T}})(x_i - b)\Big) + \text{const.} \tag{5.3.54}$$

Now the monotonicity of the logarithm implies that minimizing G_1 is again equivalent to minimizing g_1. Hence, as in case i) we get $b = \frac{1}{w}m$ is a global minimizer of g_1 resp. G_1, and it is independent of U. Using this, we obtain

$$\mu = \frac{1}{w}(U^{\mathrm{T}}m - wU^{\mathrm{T}}b) = 0 \quad \text{and} \quad \sigma^2 = \tfrac{1}{w(n-d)}\left(\mathrm{tr}(S) - \mathrm{tr}(U^{\mathrm{T}}SU)\right).$$

By (5.3.54), minimizing G_1 with respect to U for $b = \frac{1}{w}m$ is equivalent to minimizing

$$G_1(U, \frac{1}{w}m) = w(n-d)\log\left(\mathrm{tr}(S) - \mathrm{tr}(U^{\mathrm{T}}SU)\right) + \text{const}, \tag{5.3.55}$$

such that minimizing (5.3.48) with respect to U is equivalent to minimizing

$$G(U) = (n-d)\log\left(\mathrm{tr}(S) - \mathrm{tr}(U^{\mathrm{T}}SU)\right) + \log(|U^{\mathrm{T}}SU|). \tag{5.3.56}$$

□

By Proposition 1, the M-Step of Algorithm 2 reduces for $k = 1, ..., K$ to the

computation of

$$w_k^{(r+1)} = \frac{1}{N}\sum_{i=1}^{N} \alpha_{i,k}^{(r)}, \tag{5.3.57}$$

$$m_k = \sum_{i=1}^{N} \alpha_{i,k} x_i,\ C_k = \sum_{i=1}^{N} \alpha_{i,k} x_i x_i^{\mathrm{T}}, \tag{5.3.58}$$

$$(U_k^{(r+1)}, b_k^{(r+1)}) \in \underset{U\in \mathrm{SPD}(d,n), b\in\mathbb{R}^n}{\arg\min} G_k(U,b) \qquad \text{with } G_k \text{ in (5.3.41)}, \tag{5.3.59}$$

$$\mu_k^{(r+1)} = \frac{1}{N w_k^{(r+1)}} \left(U_k^{(r+1)}\right)^{\mathrm{T}} \left(m_k - N w_k^{(r+1)} b_k^{(r+1)}\right), \tag{5.3.60}$$

$$S_k = C_k - m_k \left(b_k^{(r+1)}\right)^{\mathrm{T}} - b_k^{(r+1)} m_k^{\mathrm{T}} + N w_k^{(r+1)} b_k^{(r+1)} \left(b_k^{(r+1)}\right)^{\mathrm{T}} \tag{5.3.61}$$

$$\Sigma_k^{(r+1)} = \frac{1}{N w_k^{(r+1)}} \left(U_k^{(r+1)}\right)^{\mathrm{T}} S_k \, U_k^{(r+1)} \tag{5.3.62}$$

Note that the large data set $\mathcal{X}$ is involved in the computation of m_k and C_k, but it does not influence the computational time for minimizing the G_k, $k = 1, \ldots, K$. Indeed, the E-Step of Algorithm 2 will be the most time consuming one.

5.3.3.2 PALM for Minimizing G

To minimize G in (5.3.41) we propose to use the Proximal alternating linearized minimization (PALM) [9], resp. its accelerated version iPALM [74], where the 'i' stands for inertial. As a special case the PALM algorithm can be applied to functions of the form

$$F(x) = H(x) + f(x) \tag{5.3.63}$$

where $H \in C^1(\mathbb{R}^d)$ and a lower semi-continuous function $f\colon \mathbb{R}^d \to (-\infty, \infty]$. It is based on the computation of the so-called proximal operators. For a proper and lower semi-continuous function $f\colon \mathbb{R}^d \to (-\infty, \infty]$ and $\tau > 0$ the *proximal mapping* $\mathrm{prox}_\tau^f\colon \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ is defined by

$$\mathrm{prox}_\tau^f(x) = \underset{y\in\mathbb{R}^d}{\arg\min} \left\{ \tfrac{\tau}{2}\|x-y\|^2 + f(y) \right\},$$

where $\mathcal{P}(\mathbb{R}^d)$ denotes the power set of $\mathbb{R}^d$.

Starting with an arbitrary $x^{(0)}$ PALM performs the iterations

$$x^{(r+1)} \in \mathrm{prox}_{\tau^{(r)}}^f \left(x^{(r)} - \tfrac{1}{\tau^{(r)}} \nabla H(x^{(r)}) \right). \tag{5.3.64}$$

Remark 7. In fact, this special case of the PALM algorithm with one variable U for the optimization problem (5.3.63) is just the classical Forward-Backward algorithm.

The Forward-Backwatd algorithm is also supposed to converge in the non-convex case as soon as the function satisfies the Polyak-Lojasiewics condition (see Theorem 5.1 in [6]). Thus, we can use the Forward-Backwatd algorithm with equation (5.3.64) as iterations of the algorithm. In this thesis, to get a faster scheme, we propose to use the iPALM algorithm to minimize G, which is detailed in Algorithm 3.

Indeed, we have applied the iPALM algorithm in our numerical examples. However, although we observed convergence of the iterates numerically, we have not proved convergence theoretically so far. Alternatively, we could apply the PALM algorithm which is slightly slower. Note again, that the E-Step of the algorithm is the most time consuming one.

Algorithm 3 iPALM

Input: $\alpha^{(r)}, \beta^{(r)}$ initialization $x^{(1)}$, $x^{(0)}$
for $r = 1, 2, ...$ **do** until a convergence criterion is reached

$$y^{(r)} = x^{(r)} + \alpha^{(r)}(x^{(r)} - x^{(r-1)}), \tag{5.3.65}$$
$$z^{(r)} = x^{(r)} + \beta^{(r)}(x^{(r)} - x^{(r-1)}), \tag{5.3.66}$$
$$x^{(r+1)} \in \operatorname{prox}^{f}_{\tau^{(r)}}(y^{(r)} - \tfrac{1}{\tau^{(r)}}\nabla H(z^{(r)})). \tag{5.3.67}$$

end for

In the following, we give details on PALM for our setting. For our problem (5.3.40), we choose $f(U) \coloneqq \iota_{\mathrm{St}(d,n)}$ and

$$H(U) \coloneqq G(U)\eta(\|I_d - U^T U\|_F^2), \tag{5.3.68}$$

where

$$\eta(x) \coloneqq \begin{cases} 1, & \text{if } x \in (-\rho, \rho), \\ \exp(-\frac{\rho}{\rho - (|x| - \rho)^2}), & \text{if } x \in (-2\rho, -\rho] \cup [\rho, 2\rho), \\ 0, & \text{otherwise} \end{cases}$$

is a smooth cutoff function of the interval $(-\rho, \rho)$ for some $\rho > 0$. Then, the iteration scheme reads as

$$U^{(r+1)} \in \Pi_{\mathrm{St}(d,n)}(U^{(r)} - \tfrac{1}{\tau^{(r)}}\nabla H(U^{(r)})) \tag{5.3.69}$$

where $\Pi_{\mathrm{St}(d,n)}$ denotes the orthogonal projection onto the Stiefel manifold.

Remark 8. (Projection onto Stiefel manifolds) Concerning this orthogonal projection, it is well known [43], that for a matrix $A \in \mathbb{R}^{n,d}$, the projection $\Pi_{\mathrm{St}(d,n)}(A)$ is given by the orthonormal polar factor W from the polar decomposition

$$A = WM, \quad W \in \mathrm{St}(d,n), \quad M \in \mathrm{SPD}(d).$$

Further, this orthonormal polar factor can be computed by $W = UV$, where $A = U\Sigma V$ is the singular value decomposition of A, see [43]. The authors of [44] propose to use the so-called Schulz-iteration

$$X_{k+1} = X_k(I + \tfrac{1}{2}(I - X_k^{\mathrm{T}} X_k))$$

with $X_0 = A$ for computing the orthonormal polar factor of a full rank matrix A. Unfortunately, the convergence of this iteration requires that $\|I - A^{\mathrm{T}}A\|_F < 1$, which is usually not fulfilled in our case.

Note that for any $r \in \mathbb{N}$, the matrix $U^{(r)}$ belongs to the Stiefel manifold, such that $\eta(\|I_d - U^{\mathrm{T}}U\|_F) = 1$ in a neighborhood of $U^{(r)}$. Thus, we can replace the gradient with respect to H by the gradient with respect to G in (5.3.69). Then the iteration scheme reads as

$$U^{(r+1)} \in P_{\mathrm{St}(d,n)}(U^{(r)} - \tfrac{1}{\tau^{(r)}} \nabla G(U^{(r)})), \tag{5.3.70}$$

In particular, we do not need to choose the ρ explicitly within our algorithm.

To show convergence of the algorithm, we need the following two lemmas.

Lemma 5.3.2. Let H be defined by (5.3.68). Then the function ∇H is globally Lipschitz continuous.

?proofname? The function H is twice continuously differentiable and zero outside of a compact set. Hence the second order derivative is bounded and $\nabla_U H(\cdot, b)$ is globally Lipschitz continuous. □

Further, let us recall the notation of Kurdyka-Łojasiewicz functions. For $\delta \in (0, \infty]$, we denote by Φ_δ the set of all concave continuous functions $\phi\colon [0, \delta) \to \mathbb{R}_{\geq 0}$ which fulfill the following properties:

1. $\phi(0) = 0$.
2. ϕ is continuously differentiable on $(0, \delta)$.
3. For all $s \in (0, \delta)$ it holds $\phi'(s) > 0$.

For a proper and lower semicontinuous function $\gamma\colon \mathbb{R}^d \to (-\infty, +\infty]$ denote by $\partial\gamma$ the subdifferntial of γ.

Definition 1 (Kurdyka-Łojasiewicz property)**.** *A proper, lower semicontinuous function $\gamma\colon \mathbb{R}^d \to (-\infty, +\infty]$ has the Kurdyka-Łojasieweicz (KL) property at $\bar{u} \in \operatorname{dom} \partial\gamma = \{u \in \mathbb{R}^d : \partial\gamma \neq \emptyset\}$ if there exist $\delta \in (0, \infty]$, a neighborhood U of $\bar{u}$ and a function $\phi \in \Phi_\delta$, such that for all*

$$u \in U \cap \{v \in \mathbb{R}^d : \gamma(\bar{u}) < \gamma(v) < \gamma(\bar{u}) + \delta\},$$

it holds

$$\phi'(\gamma(u) - \gamma(\bar{u}))\operatorname{dist}(0, \partial\gamma(u)) \geq 1.$$

We say that γ is a KL function, if it satisfies the KL property in each point $u \in \operatorname{dom} \partial\gamma$.

Lemma 5.3.3. The function H defined in (5.3.68) is a KL function.

?proofname? The functions G and η are sums, products, quotients and concatenations of real analytic functions. Thus, also H is a real analytic function. This implies that it is a KL function, see [5, Remark 5] and [59, 60]. □

The following theorem follows directly from [9, Lemma 3, Theorem 1].

Theorem 5.3.4 (Convergence of PALM). Let $F\colon \mathbb{R}^d \to (-\infty, \infty]$ be given by (5.3.63) and let ∇H be globally L-Lipschitz continuous. Let $(x^{(r)})_r$ be the sequence generated by PALM, where the step size parameters fulfill

$$\tau^{(r)} \geq \gamma L$$

for some $\gamma > 1$. Then, for $\eta := (\gamma - 1)L$, the sequence $(F(x^{(r)}))_r$ is nonincreasing and

$$\tfrac{\eta}{2}\|x^{(r+1)} - x^{(r)})\|_2^2 \leq F(x(r)) - F(x^{(r+1)}).$$

If F is in addition a KL function and the sequence $(x^{(r)})_r$ is bounded, then it converges to a critical point of F.

By Lemma 5.3.2 and 5.3.3 and the fact that G coincides with H in a neighborhood of the Stiefel manifold we obtain the following corollary.

Corollary 2. Let $(U^{(r)})_r$ be generated by (5.3.70) with $\tau^{(r)} \geq \gamma L$, where L is the Lipschitz constant of ∇H and $\gamma > 1$. Consider the sequence generated by PALM with (5.3.70). Then, the sequence $(G(U^{(r)}))_r$ is monotone decreasing and the sequence $(U^{(r)})_r$ converges to a critical point of G.

5.3.4 PCA-GMM model with application in super-resolution

In this subsection, we adapt the super-resolution method proposed by Sandeep and Jacob [75] to our PCA-GMM model. The method follows 3 steps as described in subsection 4.1.1. The highlight of our method is the use of PCA-GMM model to learn the parameters of the mixture model instead of directly learning the original high-dimensional data.

For given low-resolution patches $\tilde{x}_{L,i} \in \mathbb{R}^{\tau^2}$ of an image and their high-resolution counterparts $\tilde{x}_{H,i} \in \mathbb{R}^{q^2\tau^2}$, $q \in \mathbb{N}$, $q \geq 2$, $i = 1, ..., N$, we learn a PCA-GMM model based on the data $x_i = \begin{pmatrix} \tilde{x}_{H,i} \\ \tilde{x}_{L,i} \end{pmatrix} \in \mathbb{R}^n$, $n = (q^2 + 1)\tau^2$ by Algorithm 2. This provides us with parameters $\{\mathbf{U}, \mathbf{b}, \mathbf{w}, \mu, \mathbf{\Sigma}\}$ of the reduced d-dimensional GMM. Using these parameters, we compute the parameters of the corresponding high-dimensional mixture model $\left\{w_k, \tilde{\mu}_k, \tilde{\Sigma}_k\right\}$, $k = 1, \ldots, K$, where μ_k and Σ_k are defined as in (5.3.8) and (5.3.9). In the following, we use the notations $\tilde{\mu}_k = \begin{pmatrix} \tilde{\mu}_{H,k} \\ \tilde{\mu}_{L,k} \end{pmatrix}$ and

$\tilde{\Sigma}_k = \begin{pmatrix} \tilde{\Sigma}_{H,k} & \tilde{\Sigma}_{HL,k} \\ (\tilde{\Sigma}_{HL,k})^{\mathrm{T}} & \tilde{\Sigma}_{L,k} \end{pmatrix}$. With parameters $\left\{w_k, \tilde{\mu}_k, \tilde{\Sigma}_k\right\}$, $k = 1, \ldots, K$, the high-resolution patches can be estimated as in equation (4.2.22). Finally, the high-resolution image is reconstructed by applying the reconstruction of high-resolution image step by patch averaging as in 3, Subsection 4.1.1. We will demonstrate the performance of PCA-GMM method by two- and three- dimensional examples in Subsection 5.5.1.

5.4 Contribution: Dimensionality-Reduced Generalized Gaussian Distribution: PCA-GGMM Model [70]

This section aims to present our second contribution for dimensionality reduction topic. This contribution develops the PCA-GMM model by incorporate a dimensionality reduction within mixture models of generalized Gaussian distributions and derive an algorithm for estimating its parameters. Additionally to these methodical improvements, we apply our method to three-dimensional real-world images showing material microstructures. This contribution was submitted for the *Inverse Problems in Imaging* journal [70] in 2022. We will discuss it in the following.

5.4.1 Combining PCA with generalized Gaussian distribution

In the following, we consider the samples $x_1, ..., x_N \in \mathbb{R}^n$, which are approximately located in a d-dimensional subspace of $\mathbb{R}^n$. Then, our objective is to find simultaneously the subspace $\{Ut + b : t \in \mathbb{R}^d\}$, $U \in \mathrm{St}(d, n)$, $b \in \mathbb{R}^n$ that contains the samples x_i and a generalized Gaussian distribution $\mathrm{GG}(\mu, \Sigma, \beta)$ with density function $q(\cdot|\mu, \Sigma, \beta)$. For this, we propose a model that combines the PCA and the generalized Gaussian model of the low-dimensional space. Based on the idea of the PCA-GMM model, we are motivated to minimize the following function

$$\ell_{\mathrm{PCA}}(U, b, \mu, \Sigma, \beta) \coloneqq \sum_{i=1}^{N} \frac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I)(x_i - b)\|^2 - \log(q(U^{\mathrm{T}}(x_i - b)|\mu, \Sigma, \beta)), \tag{5.4.1}$$

which is a weighted sum of a PCA term and the negative log-likelihood function of q within the lower dimensional subspace.

The following proposition gives a general property of the function ℓ_{PCA} for the generalized Gaussian distribution. The function ℓ_{PCA} is up to a constant the negative log-likelihood function of a distribution P_θ with $\theta = (U, b, \mu, \Sigma, \beta)$.

Proposition 2. *Let* $\mathrm{GG}(\mu, \Sigma, \beta)$ *be a generalized Gaussian distribution with a density function* $q(\cdot|\mu, \Sigma, \beta)$. *The function* $\ell_{\mathrm{PCA}}(U, b, \mu, \Sigma, \beta)$ *is up to a constant*

the negative log-likelihood function of a probability density

$$p(x|U,b,\mu,\Sigma,\beta) \coloneqq \frac{1}{(2\pi\sigma^2)^{(n-d)/2}} \exp(-\tfrac{1}{2\sigma^2}\|(UU^{\mathrm{T}}-I)(x-b)\|^2) q(U^{\mathrm{T}}(x_i-b)|\mu,\Sigma,\beta). \tag{5.4.2}$$

The distribution P_θ with $\theta = (U,b,\mu,\Sigma,\beta)$ corresponding to this density is given by

$$P_\theta = r_{\#}(GG(\mu,\Sigma,\beta) \otimes \mathcal{N}(0,\sigma^2 I_{n-d})), \quad r(x) = Vx + b,$$

where $V = (U|\tilde{U})$ is an orthogonal matrix, $\otimes$ denotes the product measure and $r_{\#}P$ is the push-forward measure of P under r.

?proofname? Using that $U^{\mathrm{T}}U = I_d$ and $\tilde{U}^{\mathrm{T}}\tilde{U} = I_{n-d}$, we have that the distribution $\mathrm{GG}(\mu,\Sigma,\beta) \otimes \mathcal{N}(0,\sigma^2 I_{n-d})$ has at $x = (x_1,x_2) \in \mathbb{R}^n$ the density

$$\frac{1}{(2\pi\sigma^2)^{(n-d)/2}} \exp(-\tfrac{1}{2\sigma^2}\|\tilde{U}^{\mathrm{T}}\tilde{U}x_2\|^2) q(U^{\mathrm{T}}Ux_1|\mu,\Sigma,\beta)$$

As $U^{\mathrm{T}}\tilde{U} = 0$ and $\tilde{U}^{\mathrm{T}}U = 0$, we obtain, that this is equal to

$$\frac{1}{(2\pi\sigma^2)^{(n-d)/2}} \exp(-\tfrac{1}{2\sigma^2}\|\tilde{U}^{\mathrm{T}}(Ux_1+\tilde{U}x_2)\|^2) q(U^{\mathrm{T}}(Ux_1+\tilde{U}x_2)|\mu,\Sigma,\beta) \tag{5.4.3}$$

$$= \frac{1}{(2\pi\sigma^2)^{(n-d)/2}} \exp(-\tfrac{1}{2\sigma^2}\|\tilde{U}^{\mathrm{T}}Vx\|^2) q(U^{\mathrm{T}}Vx|\mu,\Sigma,\beta) \tag{5.4.4}$$

Using the change of variables formula for push-forward measures, we obtain that P_θ has the density

$$\frac{1}{(2\pi\sigma^2)^{(n-d)/2}} \exp(-\tfrac{1}{2\sigma^2}\|\tilde{U}^{\mathrm{T}}Vr^{-1}(x)\|^2) q(U^{\mathrm{T}}Vr^{-1}(x)|\mu,\Sigma,\beta)|\det(\nabla r^{-1}(x))|$$

As it holds $r^{-1}(x) = V^{\mathrm{T}}(x-b)$ and as $\nabla r^{-1}(x) = V^{\mathrm{T}}$ is an orthogonal matrix, we obtain that this is equal to

$$\frac{1}{(2\pi\sigma^2)^{(n-d)/2}} \exp(-\tfrac{1}{2\sigma^2}\|\tilde{U}^{\mathrm{T}}(x-b)\|^2) q(U^{\mathrm{T}}(x-b)|\mu,\Sigma,\beta) \tag{5.4.5}$$

where we used $VV^{\mathrm{T}} = I_n$. Finally, the fact that the mapping $x \mapsto \tilde{U}x$ is an isometry and that $I - UU^{\mathrm{T}} = \tilde{U}\tilde{U}^{\mathrm{T}}$ imply that

$$\|\tilde{U}^{\mathrm{T}}(x-b)\|^2 = \|\tilde{U}\tilde{U}^{\mathrm{T}}(x-b)\|^2 = \|(UU^{\mathrm{T}}-I)(x-b)\|^2$$

such that the density (5.4.5) coincides with $p(x|U,b,\mu,\Sigma,\beta)$. Taking the negative logarithm and subtracting $\frac{n-d}{2}\log(2\pi\sigma^2)$ shows that the negative log-likelihood function of $p(\cdot|U,b,\mu,\Sigma,\beta)$ coincides up to a constant with $\ell_{\mathrm{PCA}}(U,b,\mu,\Sigma,\beta)$. □

Due to its relationship to the PCA, we call the probability distribution $p(\cdot|U,b,\mu,\Sigma,\beta)$ a *PCA-reduced generalized Gaussian distribution.*

Remark 9 (PCA-GMM Model [41])**.** In particular, our PCA-reduced Gaussian distribution (PCA-GMM model in the case of mixture model) is again a Gaussian distribution with a special structured covariance matrix. Using this result, we have derived an EM algorithm for PCA-reduced Gaussian mixture models, which showed a significant speed-up compared to a high-dimensional GMM. However, for generalized Gaussian distribution, the PCA-reduced counterpart is not a generalized Gaussian distribution such that the analysis of [41] is no longer applicable.

In the following subsection, we introduce a method for estimating the parameters of mixture models for the PCA-reduced generalized Gaussian distribution. We call this mixture model a PCA-GGMM model, which has density function as

$$F\left(x \mid w, \Theta\right) = \sum_{k=1}^{K} w_k p\left(x \mid \theta_k\right) \tag{5.4.6}$$

where the weights w_k are non-negative and sum up to one, $\Theta = \{\theta_k\}_{k=1,\dots,K}$ with $\theta_k = (U_k, b_k, \mu_k, \Sigma_k, \beta_k)$ and $p\left(x \mid \theta_k\right)$ is defined as (5.4.2).

5.4.2 Weighted Maximum Likelihood Estimation

The PCA-GGMM model is estimated based on EM Algorithm 1. For each k mixture cluster of the PCA-GGMM model, the weighted maximum likelihood estimation (4.2.5) is estimated for PCA-reduced generalized Gaussian distribution. For readability, we consider the estimation for the the weighted maximum likelihood estimation with parameter $\theta = (U, b, \mu, \Sigma, \beta)$ instead of θ_k.

To estimate the parameters of a PCA-reduced generalized Gaussian distribution from samples $x_1, ..., x_N$ with weights $\alpha_1, ..., \alpha_N$, our objective is to minimize the weighted negative log-likelihood function

$$\sum_{i=1}^{N} \alpha_i \log(p(x_i|U, b, \mu, \Sigma, \beta)).$$

By Proposition 2 this is equivalent to minimizing the function

$$\ell_{\mathrm{PCA},\alpha}(U, b, \mu, \Sigma, \beta) \coloneqq \sum_{i=1}^{N} \alpha_i \Big(\frac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I)(x_i - b)\|^2 - \log(q(U^{\mathrm{T}}(x_i - b)|\mu, \Sigma, \beta)) \Big),$$

which is up to the weights α the same function as ℓ_{PCA} from (5.4.1). Unfortunately, we cannot minimize $\ell_{\mathrm{PCA},\alpha}$ directly. Instead, we apply a technique called half-quadratic splitting [32]. That is, instead of considering $\ell_{\mathrm{PCA},\alpha}(U, b, \mu, \Sigma, \beta)$, we minimize for some large $\eta > 0$ the function

$$H_{\sigma,\eta}(z, U, b, \mu, \Sigma, \beta) \coloneqq \sum_{i=1}^{N} \alpha_i \Big(\frac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I)(x_i - b)\|^2 + \eta \|U^{\mathrm{T}}(x_i - b) - z_i\|^2 - \log(q(z_i|\mu, \Sigma, \beta)) \Big),$$

i.e., we replace $U^{\mathrm{T}}(x_i - b)$ in the second summand by z_i and penalize the squared distance of $U^{\mathrm{T}}(x_i - b)$ and z_i. Thus, for $\eta \to \infty$, the minimizers of $H_{\sigma,\eta}$ converge to the minimizers of $\ell_{\mathrm{PCA},\alpha}$. Now, we minimize $H_{\sigma,\eta}$ iteratively with respect to (U, b), z and (μ, Σ, β), i.e., we generate a sequence $(U^{(r)}, b^{(r)}, \mu^{(r)}, \Sigma^{(r)}, \beta^{(r)})_r$ by iteration

$$U^{(r)} \in \operatorname*{arg\,min}_{U \in \mathrm{St}(d,n)} \sum_{i=1}^{N} \alpha_i \Big(\frac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I)(x_i - b^{(r-1)})\|^2 + \eta \|U^{\mathrm{T}}(x_i - b^{(r-1)}) - z_i^{(r-1)}\|^2 \Big), \tag{5.4.7}$$

$$b^{(r)} \in \operatorname*{arg\,min}_{b \in \mathbb{R}^n} \sum_{i=1}^{N} \alpha_i \Big(\frac{1}{2\sigma^2} \|(U^{(r)}(U^{(r)})^{\mathrm{T}} - I)(x_i - b)\|^2 + \eta \|(U^{(r)})^{\mathrm{T}}(x_i - b) - z_i^{(r-1)}\|^2 \Big), \tag{5.4.8}$$

$$z^{(r)} \in \operatorname*{arg\,min}_{z=(z_1,\dots,z_N) \in \mathbb{R}^{d,N}} \sum_{i=1}^{N} \alpha_i \Big(\eta \|(U^{(r)})^{\mathrm{T}}(x_i - b^{(r)}) - z_i\|^2 - \log(q(z_i | \mu^{(r-1)}, \Sigma^{(r-1)}, \beta^{(r-1)})) \Big) \tag{5.4.9}$$

and finally

$$(\mu^{(r)}, \Sigma^{(r)}, \beta^{(r)}) \in \operatorname*{arg\,max}_{\mu \in \mathbb{R}^d, \Sigma \in \mathrm{SPD}(d), \beta > 0} \sum_{i=1}^{N} \alpha_i \log(q(z_i^{(r)} | \mu, \Sigma, \beta)) \tag{5.4.10}$$

The final step (5.4.10) is the weighted maximum likelihood estimation of the generalized Gaussian distribution q, which was discussed in the previous section. It remains to solve the first three steps.

Solving (5.4.7) by Uzawas' Algorithm. Using that for any $y \in \mathbb{R}^n$ and $U \in \mathrm{St}(d, n)$ it holds that

$$\|(UU^{\mathrm{T}} - I)y\|^2 = y^{\mathrm{T}}(UU^{\mathrm{T}} - I)^2 y = y^{\mathrm{T}}(UU^{\mathrm{T}}UU^{\mathrm{T}} - 2UU^{\mathrm{T}} + I)y = y^{\mathrm{T}}(UU^{\mathrm{T}} - 2UU^{\mathrm{T}} + I)y \tag{5.4.11}$$

$$= y^{\mathrm{T}}(I - UU^{\mathrm{T}})y = \|y\|^2 - \|U^{\mathrm{T}}y\|^2, \tag{5.4.12}$$

the optimization problem in (5.4.7) reads as

$$\operatorname*{arg\,min}_{U \in \mathrm{St}(d,n)} \sum_{i=1}^{N} \alpha_i \Big(-\frac{1}{2\sigma^2} \|U^{\mathrm{T}}(x_i - b)\|^2 + \eta \|U^{\mathrm{T}}(x_i - b) - z_i\|^2 \Big)$$

Now, we transform the problem in an unconstrained problem by considering the Lagrangian function

$$\mathcal{L}(U, \Lambda) = \sum_{i=1}^{N} \alpha_i \Big(-\frac{1}{2\sigma^2} \|U^{\mathrm{T}}(x_i - b)\|^2 + \eta \|U^{\mathrm{T}}(x_i - b) - z_i\|^2 \Big) + \mathrm{tr}(\Lambda(U^{\mathrm{T}}U - I)).$$

Then, the solution of (5.4.7) is equivalent to solving the saddle-point problem

$$\arg\min_{U\in\mathbb{R}^{n,d}} \max_{\Lambda\in\mathbb{R}^{d,d}} \mathcal{L}(U,\Lambda).$$

To solve this saddle-point problem, we use the Uzawas' algorithm [7], which consists of the following two steps: First, we minimize $\mathcal{L}$ with respect to U. Second, we perform a gradient ascent step with respect to Λ.

The second step is given by

$$\Lambda^{(r+1)} = \Lambda^{(r)} + \rho((U^{(r+1)})^{\mathrm{T}}U^{(r+1)} - I)$$

In the case of our specific Lagrangian $\mathcal{L}$, the second step is given by the following proposition.

Proposition 3. *Let Λ be fixed. Then, any minimizer of $\mathcal{L}(U,\Lambda)$ solves the Sylvester equation*

$$\Big((\eta - \tfrac{1}{2\sigma^2})\sum_{i=1}^{N}\alpha_i(x_i-b)(x_i-b)^{\mathrm{T}}\Big)U + U\Lambda = \eta\sum_{i=1}^{N}\alpha_i(x_i-b)z_i^{\mathrm{T}}.$$

?proofname? We aim to minimize the Lagrangian function

$$\mathcal{L}(U,\Lambda) = \sum_{i=1}^{N}\alpha_i\Big(-\frac{1}{2\sigma^2}\|U^{\mathrm{T}}(x_i-b)\|^2 + \eta\|U^{\mathrm{T}}(x_i-b)-z_i\|^2\Big) + \mathrm{tr}(\Lambda(U^{\mathrm{T}}U-I)).$$

for fixed Λ with respect to U by setting the gradient to zero. Since it holds by [73] that

$$\frac{\partial a^{\mathrm{T}}UU^{\mathrm{T}}b}{\partial U} = U^{\mathrm{T}}(ab^{\mathrm{T}}+ba^{\mathrm{T}}) \quad \text{and} \quad \frac{\partial \mathrm{tr}(\Lambda U^{\mathrm{T}}U)}{\partial U} = \frac{\partial \mathrm{tr}(\Lambda U^{\mathrm{T}}U)}{\partial (U^{\mathrm{T}}U)}\frac{\partial U^{\mathrm{T}}U)}{\partial U} = 2\Lambda^{\mathrm{T}}U^{\mathrm{T}},$$

Thus, it holds that

$$\nabla_U\mathcal{L}(U,\Lambda) = 2(\eta-\tfrac{1}{2\sigma^2})\sum_{i=1}^{N}\alpha_i(x_i-b)(x_i-b)^{\mathrm{T}}U - 2\eta\sum_{i=1}^{N}\alpha_i(x_i-b)z_i^{\mathrm{T}} + 2U\Lambda.$$

Setting the gradient to zero, it shows that U is a solution of

$$\Big((\eta - \tfrac{1}{2\sigma^2})\sum_{i=1}^{N}\alpha_i(x_i-b)(x_i-b)^{\mathrm{T}}\Big)U + U\Lambda = \eta\sum_{i=1}^{N}\alpha_i(x_i-b)z_i^{\mathrm{T}}.$$

This finishes the proof. □

Solving (5.4.8). Setting the gradient of the objective function to zero leads to

$$\sum_{i=1}^{N} \alpha_i \left[\left(\frac{1}{2\sigma^2} \left(I - UU^{\mathrm{T}} \right) + \eta UU^{\mathrm{T}} \right) (b - x_i) - \eta U z_i \right] = 0 \tag{5.4.13}$$

which is equivalent to

$$b = \frac{\sum_{i=1}^{N} \alpha_i x_i + \left[\frac{1}{2\sigma^2} \left(I - UU^{\mathrm{T}} \right) + \eta UU^{\mathrm{T}} \right]^{-1} \eta \sum_{i=1}^{N} \alpha_i U z_i}{\sum_{i=1}^{N} \alpha_i}. \tag{5.4.14}$$

Solving (5.4.9) using a Gradient Descent. Due to the sum-structure of problem (5.4.9), the solution decouples into

$$z_i^{(r)} \in \underset{z_i \in \mathbb{R}^d}{\arg\min}\, \eta \| (U^{(r)})^{\mathrm{T}} (x_i - b^{(r)}) - z_i \|^2 - \log(q(z_i | \mu^{(r-1)}, \Sigma^{(r-1)}, \beta^{(r-1)})). \tag{5.4.15}$$

This is now a d-dimensional optimization problem, which (owing to its low dimension) can be solved efficiently via a gradient descent scheme.

Remark 10 (Differentiability of the objective)**.** In the case of generalized Gaussian distributions, the function g is given by $g(x) = \exp(-\frac{1}{2}x^{\beta})$, which is differentiable in $\{0, +\infty\}$. Thus, the objective function in (5.4.15) is differentiable for any $z_i \neq \mu^{(r-1)}$. In the case $\beta \geq 1$, the derivative of g can be continuously extended to 0. In this case, the objective function in (5.4.15) is differentiable.

The reduction in dimensionality of the PCA-GGMM model reduces the execution time of the E-step, but the M-step is slower than the GGMM due to its higher complexity. However, all computations in the M-step are on d-dimensional optimization problems, and the FP iteration is implemented on d-dimensional data. This limits the numerical instability problem to estimate the generalized Gaussian mixture model of the original data using the FP-EM algorithm. Moreover, the PCA-reduced generalized Gaussian distribution can be generalized to the PCA-reduced elliptical distribution by combining PCA with the elliptical distribution $\mathcal{E}_n(\mu, \Sigma, g)$ with the density function $q(\cdot|\mu, \Sigma, g)$. We denote $p(\cdot|U, b, \mu, \Sigma, g)$ as a probability density function of the PCA-reduced elliptical distribution. By using the half-quadratic splitting technique, we can apply the iterations (5.4.7), (5.4.8), (5.4.9) and (5.4.10) with $q(\cdot|\mu, \Sigma, g)$ instead of $q(\cdot|\mu, \Sigma, \beta)$ to estimate the parameters of a PCA-reduced elliptical distribution $p(\cdot|U, b, \mu, \Sigma, g)$.

5.4.3 PCA-GGMM model with application in super-resolution

In the PCA-GGMM model, the latent data $U^{\mathrm{T}}(x_i - b)$ have a d-dimensional generalized Gaussian mixture distribution with parameters $\theta_k = \{\mu_k, \Sigma_k, \beta_k\}$ for all $k = 1, ..., K$. Due to the proposition 4 and the definition of the affine subspace $\left\{Ut + b : t \in \mathbb{R}^d\right\}$, the high-dimensional variables $x_1, ..., x_N \in \mathbb{R}^n$ have a generalized Gaussian mixture distribution with parameters $\overline{\theta}_k = \left\{\overline{\mu}_k, \overline{\Sigma}_k, \overline{\beta}_k\right\}$ for

all $k = 1, ..., K$. At each k mixture component of the mixture model, the parameters are given by:

$$\overline{\mu}_k = b_k + U_k \mu_k, \tag{5.4.16}$$
$$\overline{\Sigma}_k = U_k \Sigma_k U_k^{\mathrm{T}}, \tag{5.4.17}$$
$$\overline{\beta}_k = \beta_k. \tag{5.4.18}$$

Proposition 4. *An absolutely continuous random vector Y has a n-dimensional elliptical distribution with parameters μ, Σ and g $Y \sim \mathcal{E}_d(\mu, \Sigma, g)$. Let B be a $n \times d$ matrix, and let $b \in \mathbb{R}^n$. Then*

$$b + BY \sim \mathcal{E}_n(b + B\mu, B\Sigma B^T, g). \tag{5.4.19}$$

?proofname? Lemma 3.1, p. 5 [47]. □

The PCA-GGMM model can be used for super-resolution based on MMSE estimator approach as in Section 4.2.2. We can apply the MMSE estimator in equation (4.2.22) presented in Section 4.2.2 to the generalized Gaussian mixture model with parameters $\{\overline{\theta}_k\}$ of the high dimensional data. Finally, the high-resolution image is reconstructed by applying the reconstruction of high-resolution image step by patch averaging as in 3, Subection 4.1.1.

5.5 Numerical results

In this section, we demonstrate the performance of PCA-GMM model and PCA-GGMM model with applications in super-resolution by two- and three-dimensional examples, where we mainly focus on material data which introduced in Chapter 3.

Initialization of the EM algorithms. Since the negative log-likelihood function is non-convex and admits many critical points, EM algorithms for GMMs are very sensitive with respect to the initialization. For example this can be seen by considering the case that $\theta_k^{(r)} = \theta_l^{(r)}$, $k, l = 1, ..., K$ for some $r \in \mathbb{N}$. Then we obtain that $\alpha_{i,k}^{(r)} = w_k$ and consequently $\theta_k^{(r+1)} = \theta_l^{(r+1)}$, $k, l = 1, ..., K$. The same effect appears for PCA-GMMs, PCA-GGMMs and HDDC. Consequently the initialization of the EM algorithms is of great importance. For our numerical examples, we initialize the GMMs as follows. We set $w_k^{(0)} = \frac{1}{K}$, for $k = 1, ..., K$. For initializing the means, we choose randomly K distinct data points $\mu_1, ..., \mu_K$ from our training data $x_{H,1}, ..., x_{H,N}$. Finally, we choose for each $k = 1, ..., K$ the M points $x_{L,1}, ..., x_{L,M}$ from $x_{H,1}, ..., x_{H,N}$ which are the closest ones to μ_k and initialize the covariances by $\Sigma_k = \frac{1}{M} \sum_{i=1}^{M} x_{L,i} x_{L,i}^{\mathrm{T}}$. The number M is chosen according to the dimension n of the data. In our examples, we use $M = 2n$.

We initialize the PCA-GMMs, PCA-GGMMs and HDDC by taking the initialization for GMMs, running one E-Step from the EM algorithm for GMMs followed by the M-step of the PCA-GMMs, PCA-GGMMs or HDDC, respectively.

Choice of σ and K. The PCA-GMM model and PCA-GGMM model depend heavily on the choice of the parameter σ. As pointed out in Subsection 5.3.3.1, this parameter could be learned from the data. However, the forward model for the low-resolution images $X_L = AX_H + \varepsilon$ for some (unknown) super-resolution operator A, the high-resolution image X_H and noise ε suggest to choose the σ according to the standard deviation of the noise ε. Note, that in our experiments, the low-resolution images are artificially generated by applying a downsampling operator and adding some noise. Consequently, the standard deviation of ε is known. Nevertheless, if the noise level is unknown, it could be estimated very accurately from the data based on homogeneous area detection as done, e.g., in [37, 78].

In practice, it can be unstable to estimate the standard deviation of the noise within the optimization of the mixture model, since this requires that the image patches belong exactly (and not only approximately) to a dimensionality reduced GMM and GGMM with K components, which is an unrealistic assumption. Therefore, it can be beneficial and quite more accurate to estimate the standard deviation of the noise a priori. In particular, if the standard deviation of the noise is known, fixing σ can be the better approach.

Note that the noise with standard deviation σ within the super-resolution model does not necessarily imply that the eigenvalues of the covariance matrices in the mixture model are greater than or equal to σ^2 (which is assumed for HDDC), since the noise is only applied to the low-resolution images.

Also the number of components K of the mixture models can have a large impact on the results. For super-resolution, a detailed comparison of the prediction quality for different choices of K was done by Sandeep and Jacob in [75]. They observed that the benefit of taking more than 100 components in the GMM is usually very small. Therefore, we take $K = 100$ components for all mixture model in our numerical examples.

5.5.1 PCA-GMM model

All implementations in this subsection were done in Python and Tensorflow and they can be parallelized on a GPU. We run all our experiments on a Lenovo ThinkStation with Intel i7-8700 6-Core processor with 32GB RAM and NVIDIA GeForce GTX-2060 Super GPU. The code is available online[1].

For the implementation of PALM and iPALM, we use the implementation framework from [40][2]. As suggested in [74] we set the extrapolation factors $\gamma_1^{(r)} = \gamma_2^{(r)} = \frac{r-1}{r+2}$ and choose $\tau_1^{(r)} = \frac{1}{\tilde{L}_1(b^{(r)})}$ and $\tau_2^{(r)} = \frac{1}{\tilde{L}_2(U^{(r+1)})}$, where $\tilde{L}_1(b^{(r)})$ and $\tilde{L}_2(U^{(r+1)})$ are estimates of the Lipschitz constant of $\nabla_U G(\cdot, b^{(r)})$ and $\nabla_b G(U^{(r+1)}, \cdot)$.

[1] https://github.com/johertrich/PCA_GMMs
[2] https://github.com/johertrich/Inertial-Stochastic-PALM

Comparison of the computation times. Note that there already exist implementations of HDDC by some of the authors of [11]. However, to provide a fair comparison of the execution times, we reimplement the EM algorithm for HDDC in Python and Tensorflow, such that it supports GPU parallelization. Further, note that we compute the updates of α, m and C simultaneously to the E-step such that the corresponding execution time is contained in the E-step, even though the updates technically belong to the M-step. This has the advantage that we have to iterate only once over the whole data set and enables a better parallelization. We implemented this optimization of the order of computation for all of the models (GMM, PCA-GMM and HDDC) analogously.

2D-Data. For estimating the parameters of the mixture models, we use the upper left quarter of the image as in the top row of Figure 4.1. As ground truth for the reconstruction we use the whole images as in the bottom row. The images in the left and middle columns are the middle slices of the material data "FS" and "SiC Diamonds". The high resolution images have a size of 2560×2560. The right column contains the `goldhill` image, which has the size 512×512.

We estimate the parameters of a GMM and of our PCA-GMM as described in the previous sections. First, we fix the parameter σ in Algorithm 2 as the standard deviation of the noise on the low dimensional image (i.e. $\sigma = 0.02$). Second, we consider the case when σ is learned from the data and finally we compare our results with HDDC [11]. Each mixture model has $K = 100$ classes. We use the magnification factors $q \in \{2, 4\}$ and the patch size $\tau = 4$ for the low resolution patches. This corresponds to a patch size of $q\tau = 8$ or $q\tau = 16$ respectively for the high resolution images. For the material images, this leads to $N \approx 400000$ patches for $q = 2$ and $N \approx 100000$ for $q = 4$. Using the `goldhill` image, we get $N \approx 15000$ patches for $q = 2$ and $N \approx 3700$ patches for $q = 4$. We reduce the dimension of the pairs of high and low resolution patches from $n = (q^2+1)\tau^2 = 80$ or $n = (q^2+1)\tau^2 = 272$ respectively to d for $d \in \{4, 8, 12, 16, 20\}$. After estimating the mixture models, we use the reconstruction method from [75] as described in the previous section to reconstruct the ground truth from the artificially downsampled images. The resulting PSNRs are given in Table 5.1. As a reference we also measure the PSNR of the bicubic interpolation. The average execution times for one E-step and one M-step are given in Table 5.2. Figure 5.2 shows some small areas of the high resolution images, low resolution images and the corresponding reconstructions for GMMs and PCA-GMM with $d = 12$ and $d = 20$. The result with $d = 12$ for PCA-GMM is already almost as good as GMM, whereas the dimension of the patches was reduced by a factor between 4 and 22 (depending on the case). Further, we observed that the dimensionality reduction reduces the execution time of the E-step significantly. On the other hand, the execution time of the M-step is larger than those in the GMM for all dimension reduced models due its higher complexity. Comparing the different dimensionality reduced models, we observe that the PCA-GMM with fixed σ gives significantly better results than the other models, while HDDC achieves the

fastest M-step due to the closed-form updates. However, compared to the execution time of the E-step, this advantage seems to be negligible for large data sets as, e.g., the patches from the FS and SiC Diamonds image.

		Magnification factor $q = 2$			Magnification factor $q = 4$		
	d	FS	Diamonds	Goldhill	FS	Diamonds	Goldhill
bicubic	-	30.57	30.67	28.99	25.27	25.19	24.66
GMM	-	35.49	37.21	31.63	30.69	30.74	27.80
PCA-GMM, $\sigma = 0.02$	20	35.44	37.24	31.25	30.75	30.74	27.64
	16	35.42	37.22	31.25	30.74	30.62	27.59
	12	35.47	37.13	31.18	30.67	30.48	27.55
	8	35.32	36.69	31.00	30.46	30.16	27.38
	4	34.69	35.23	30.42	29.78	29.24	26.89
PCA-GMM, learned σ	20	35.22	37.06	31.27	30.43	30.51	27.66
	16	35.14	37.01	31.14	30.34	30.31	27.51
	12	34.95	36.54	30.94	30.13	29.84	27.33
	8	34.43	35.47	30.54	29.62	29.08	26.88
	4	32.74	33.41	29.69	28.51	27.75	26.16
HDDC [11]	20	35.35	37.12	31.35	30.54	30.63	27.73
	16	35.31	37.10	31.25	30.47	30.48	27.62
	12	35.24	36.64	31.08	30.27	30.08	27.40
	8	34.76	35.66	30.76	29.80	29.34	27.00
	4	33.46	33.86	29.93	28.61	27.99	26.37

?tablename? 5.1: PSNRs of the reconstructions of artificially downsampled 2D images using either bicubic interpolation, a GMM, PCA-GMM for different choices of d or HDDC. The magnification factor is set to $q \in \{2, 4\}$. PCA-GMM produces results almost as good as GMM, with a much lower dimensionality.

Figure 5.3 shows a histogram of the eigenvalues of the covariance matrices Σ_k, $k = 1, ..., K$ of the PCA-GMM model with fixed $\sigma = 0.02$ for the FS and SiC Diamonds image with magnification $q = 4$. We observe, that for the SiC Diamonds image a significant amount of eigenvalues are smaller than $\sigma^2 = 4 \cdot 10^{-4}$ which is not possible within a HDDC model [11]. For the FS image, the eigenvalues are mostly greater than σ^2.

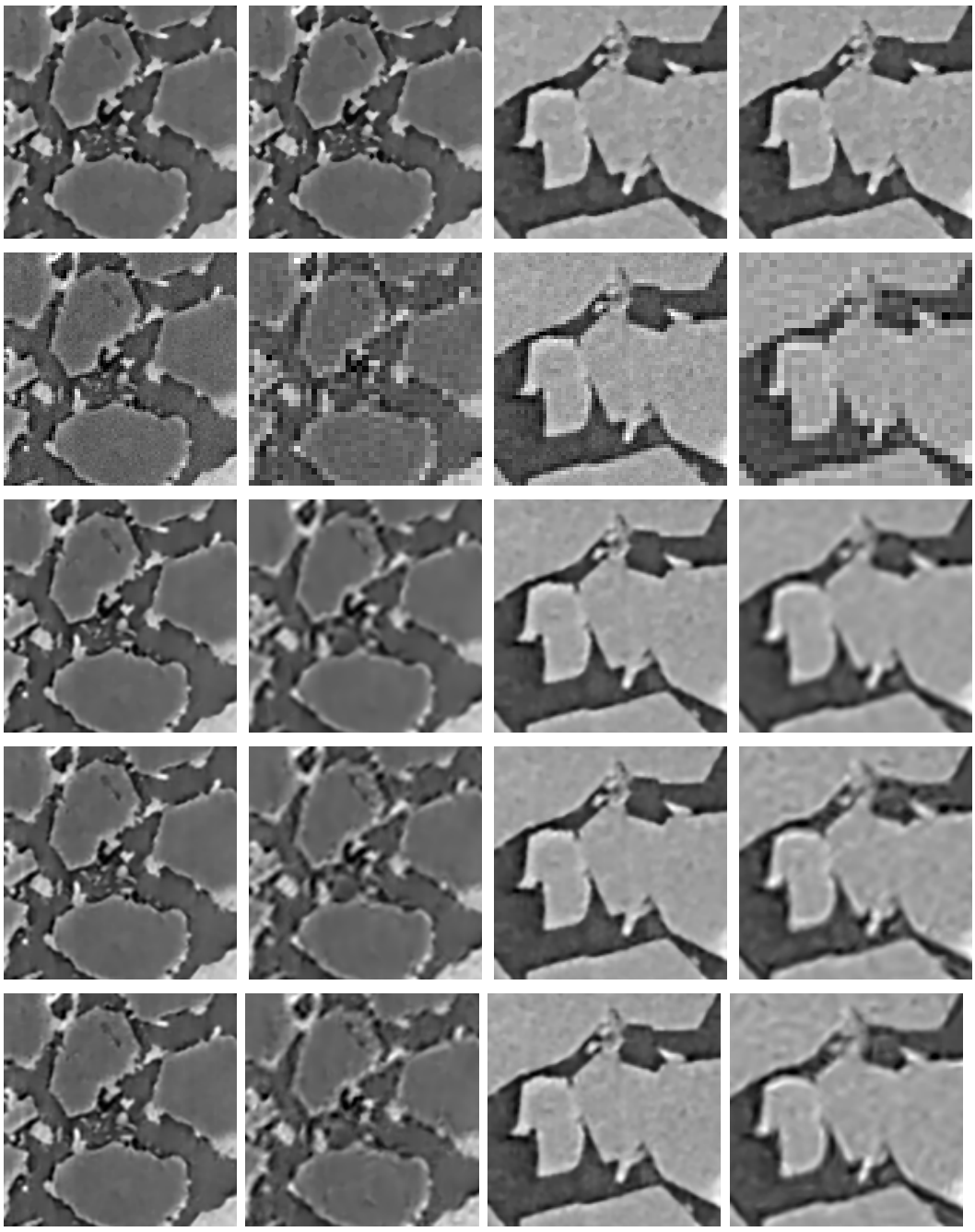
3D-Data. In the following, we present the same experiments as in the 2D-case but with 3D-data. For this experiment, we crop a $600 \times 600 \times 600$ image from the material images "FS" and "SiC Diamonds". For the estimation of the mixture model, we use the upper front left $300 \times 300 \times 300$ part of the images and crop randomly $N = 1000000$ patches.

Again, we estimate the parameters of a GMM and a PCA-GMM with $K = 100$ classes and fixed $\sigma = 0.02$ as described in the previous sections. Since we have seen in the 2D examples that the results of PCA-GMMs with learned σ and HDDC are

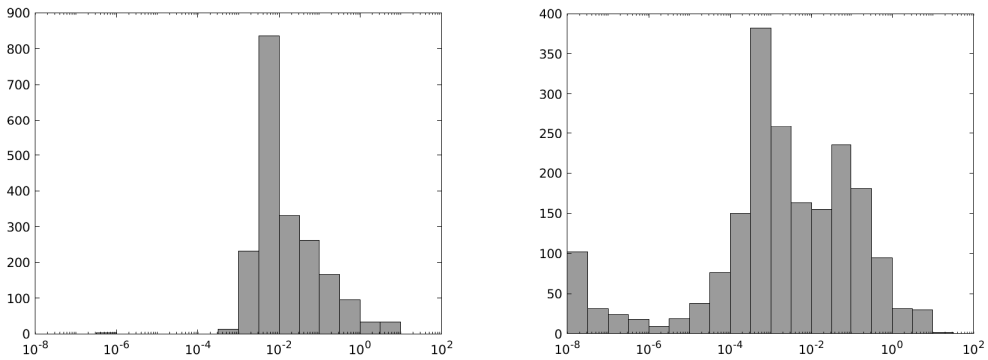
		Magnification factor $q = 2$, i.e. dimension $n = 80$					
		FS, $N = 405769$		Diamonds, $N = 405769$		Goldhill, $N = 15625$	
	d	E-step	M-step	E-step	M-step	E-step	M-step
GMM	-	10.91	0.06	10.91	0.06	0.44	0.06
PCA-GMM, $\sigma = 0.02$	20	7.25	0.74	7.42	0.57	0.28	0.54
	12	6.58	0.59	6.53	0.51	0.25	0.46
	4	6.18	0.56	6.17	0.52	0.24	0.48
PCA-GMM, learned σ	20	7.28	0.54	7.41	0.54	0.28	0.54
	12	6.59	0.47	6.53	0.45	0.25	0.47
	4	6.20	0.47	6.17	0.44	0.24	0.51
HDDC [11]	20	7.27	0.27	7.44	0.27	0.28	0.26
	12	6.64	0.26	6.64	0.26	0.25	0.26
	4	6.27	0.27	6.23	0.26	0.24	0.26

		Magnification factor $q = 4$, i.e. dimension $n = 272$					
		FS, $N = 100489$		Diamonds, $N = 100489$		Goldhill, $N = 3721$	
	d	E-step	M-step	E-step	M-step	E-step	M-step
GMM	-	17.15	0.06	17.11	0.06	0.90	0.06
PCA-GMM, $\sigma = 0.02$	20	8.65	3.54	8.68	2.03	0.44	1.83
	12	8.17	2.73	8.15	1.99	0.42	1.75
	4	7.95	2.10	7.94	2.49	0.41	1.91
PCA-GMM, learned σ	20	8.65	1.92	8.70	1.83	0.44	1.87
	12	8.17	1.99	8.17	1.74	0.42	1.74
	4	7.94	2.17	7.93	1.76	0.41	1.72
HDDC [11]	20	8.65	1.53	8.71	1.54	0.44	1.52
	12	8.16	1.54	8.14	1.53	0.42	1.52
	4	7.95	1.54	7.96	1.54	0.41	1.52

?tablename? 5.2: Average execution time (in seconds) for the E-step and M-step in the EM algorithm for estimating the parameters of the mixture models.



?figurename? 5.2: Reconstructions of 2D low resolution images. The columns from left to right correspond to the following materials images: Diamond with $q = 2$, Diamond with $q = 4$, FS with $q = 2$, FS with $q = 4$. First row: ground truth, second row: low resolution, third row: reconstruction with GMM, fourth row: reconstruction with PCA-GMM and $d = 20$, fifth row: reconstruction with PCA-GMM and $d = 12$. The larger of d, the closer is the result of PCA-GMM to GMM.



(a) (b)

?figurename? 5.3: Histograms of the eigenvalues of Σ_k, $k = 1, ..., K$ for the PCA-GMM with fixed $\sigma = 0.02$ for $d = 20$. a)FS with magnification $q = 4$, b) SiC Diamonds with magnification $q = 4$

similar, we compare our 3D results just with HDDC. As magnification factor, we use $q = 2$. For the low resolution image we use $\tau \times \tau \times \tau$-patches with patch size $\tau = 4$ and for the high resolution image we use a patch size of $q\tau = 8$. We reduce the dimension of the pairs of high and low resolution patches from $n = (q^3 + 1)\tau^3 = 576$ to d for $d \in \{20, 40, 60\}$. After estimating the mixture models, we use the reconstruction method from [75] as described in the previous paragraph to reconstruct the ground truth from of the artificially downsampled images. The resulting PSNRs are given in Table 5.3 and the average execution times of one E-step and one M-step are given in Table 5.4. As a reference we also measure the PSNR of the nearest neighbor interpolation.

	d	FS	Diamonds
Nearest neighbor	-	30.10	26.25
GMM	-	33.32	30.71
PCA-GMM, $\sigma = 0.02$	60	33.38	30.83
	40	33.36	30.75
	20	33.25	30.17
HDDC [11]	60	33.23	30.49
	40	33.24	30.29
	20	33.02	29.47

?tablename? 5.3: PSNRs of the reconstructions of artificially downsampled 3D images using either nearest neighbor interpolation, GMM or PCA-GMM for different choices of d. The magnification factor is set to $q = 2$. As in the 2D case, PCA-GMM with small d produces results almost as good as GMM, but with a much lower dimensionality.

		FS		Diamonds	
	d	E-step	M-step	E-step	M-step
GMM	-	717.91	0.07	718.13	0.07
PCA-GMM, $\sigma = 0.02$	60	338.22	12.29	337.44	17.49
	40	327.34	9.73	324.93	13.87
	20	320.00	7.85	319.46	9.80
HDDC [11]	60	337.29	4.15	337.42	4.16
	40	327.11	4.19	324.95	4.15
	20	320.03	4.20	319.07	4.15

?tablename? 5.4: Average execution time (in seconds) of the E-step and M-step in the EM algorithm for estimating the parameters of the mixture models.

5.5.2 PCA-GGMM

In the second subsection, we focus on the dimensionality reduction task with our PCA-GGMM model for application in super-resolution. We compare our model with the PCA-GMM [41] on 2D artificially images as in the subsection 4.2.3 and the two- and three-dimensional material images: FS", "SiC Diamonds" with the zooming ratio $q = 2, 4$.

As in the previous experiments with MMSE-GGMM, we use the patch size $\tau = 4$ for the low resolution and $q\tau = 8$ (or $q\tau = 16$) for high resolution images. For the material images with size 2500×2500 of HR, this leads to $N \approx 400000$ patches for $q = 2$ and $N \approx 100000$ patches for $q = 4$. These patches will be used for the training step of the PCA-GGMM model. In this step, we use $K = 100$ components for the mixture model. Each training patch V of size $D = \tau^2 \left(q^2 + 1\right) = 80$ or $D = \tau^2 \left(q^2 + 1\right) = 272$ is reduced to d low dimensional for $d \in \{4, 8, 12, 16, 20\}$. The PSNR values in Tables 5.5, 5.6 and 5.7 demonstrate that our PCA-GGMM model produces results almost as good as MMSE-GGMM and even slightly better than MMSE-GMM.

For the 3D material data, we crop a $600 \times 600 \times 600$ image from the material images FS and SiC Diamonds. Training data are taken from the upper left part of the 3D image $300 \times 300 \times 300$. We use the magnification factor $q = 2$, the low-resolution patch size $\tau = 4$, and the high-resolution patch size $q\tau = 8$. Thus, the dimension of the pair high and low resolution patches is $D = \tau^3 \left(q^3 + 1\right) = 576$. These high-dimensional data are reduced to d for $d \in \{20, 40, 60\}$ in the PCA-GGMM model. Table 5.8 shows the PSNR values of the MMSE method for the GMM, LMM, GGMM model and the PCA-GMM, PCA-GGMM model.

5.6 Conclusion

In this chapter, we presented two new algorithms to perform image super-resolution. In the first contribution, we added a dimensionality reduction step within the

	d	Hill	Camera	Barbara	FS	Diamonds
MMSE-GMM	80	31.60	32.75	25.27	35.48	37.23
MMSE-LMM	80	31.69	32.82	25.31	35.53	37.34
MMSE-GGMM	80	**31.70**	**32.86**	**25.33**	**35.57**	**37.39**
PCA-GMM	20	31.59	32.69	25.26	35.45	36.22
	16	31.46	32.61	25.23	35.43	36.20
	12	31.23	32.58	25.11	35.46	36.15
	8	30.83	32.36	24.82	35.29	36.67
	4	30.71	32.19	24.75	34.71	35.41
PCA-LMM	20	31.61	32.72	25.29	35.51	37.32
	16	31.60	32.64	25.28	35.49	37.29
	12	31.49	32.61	25.20	35.48	37.23
	8	31.09	32.46	25.02	35.36	36.88
	4	30.80	32.32	24.93	34.85	35.57
PCA-GGMM	20	31.63	32.74	25.30	35.56	37.38
	16	31.60	32.70	25.28	35.52	37.35
	12	31.54	32.65	25.25	35.50	37.30
	8	31.18	32.53	25.11	35.41	36.97
	4	31.02	32.40	24.94	34.97	35.71

?tablename? 5.5: PSNRs of the reconstructions of 2D images using MMSE method with GMM, LMM and GGMM (with learned shape parameter β) models, and the PCA with PCA-GMM, PCA-LMM, PCA-GGMM. The magnification factor is set to $q = 2$.

		d	GMM	LMM	GGMM
	MMSE		33.09	33.32	33.35
FS	PCA	20	33.03	33.25	33.29
		16	32.99	33.21	33.25
		12	32.86	33.17	33.20
		8	32.41	32.68	32.91
		4	32.10	32.43	32.44
	MMSE		28.00	28.07	28.08
Diamonds	PCA	20	27.99	28.02	28.05
		16	27.98	28.00	28.02
		12	27.85	27.91	27.94
		8	27.67	27.70	27.70
		4	27.14	27.21	27.23

?tablename? 5.6: PSNRs of the reconstructions of real material 2D image for magnification factor is set to $q = 2$ using MMSE method with GMM, LMM and GGMM (with learned shape parameter β) models, and the PCA with PCA-GMM, PCA-LMM, PCA-GGMM.

	d	Hill	Camera	Barbara	FS	Diamonds
MMSE-GMM	-	27.15	26.19	23.64	30.72	30.74
MMSE-LMM	-	27.16	26.21	23.67	30.76	30.79
MMSE-GGMM	-	**27.18**	**26.23**	**23.72**	**30.83**	**30.81**
PCA-GMM	20	26.89	26.15	23.47	30.73	30.73
	16	26.81	26.07	23.32	30.71	30.61
	12	26.73	25.96	23.27	30.68	30.46
	8	26.41	25.69	23.01	30.47	30.18
	4	26.22	25.19	22.86	29.80	29.23
PCA-LMM	20	27.14	26.18	23.50	30.74	30.76
	16	27.05	26.12	23.41	30.71	30.68
	12	26.84	26.03	23.29	30.70	30.53
	8	26.60	25.87	23.12	30.52	30.24
	4	26.46	25.34	22.94	29.84	29.41
PCA-GGMM	20	27.15	26.20	23.65	30.81	30.78
	16	27.01	26.14	23.58	30.78	30.73
	12	26.83	26.03	23.43	30.74	30.59
	8	26.71	25.89	23.21	30.59	30.30
	4	26.57	25.40	23.02	29.93	29.48

?tablename? 5.7: PSNRs of the reconstructions of 2D images for magnification factor is set to $q = 4$ using MMSE method with GMM, LMM and GGMM (with learned shape parameter β) models, and the PCA with PCA-GMM, PCA-LMM, PCA-GGMM.

		d	GMM	LMM	GGMM
	MMSE		33.34	33.36	33.41
FS	PCA	60	33.28	33.31	33.36
		40	33.25	33.27	33.35
		20	33.17	33.19	33.23
	MMSE		30.68	30.70	30.75
Diamonds	PCA	60	30.63	30.64	30.67
		40	30.55	30.57	30.62
		20	30.36	30.42	30.48

?tablename? 5.8: PSNRs of the reconstructions of 3D images using MMSE method and PCA-GGMM with $q = 2$.

GMM model using PCA on patches. The new variational model, called PCA-GMM is of interest on its own, and can be also applied for other tasks. We solved our PCA-GMM model by an EM algorithm with the usual decreasing guarantees for the objective if the E-step and M-step can be performed exactly, see Corollary 1. However, our M-step requires to solve a non-convex constrained minimization problem. Here we propose to use the PALM algorithm and prove that all assumptions for the convergence of the sequence of iterates to a critical point required by [9] are fulfilled, see Corollary 2. Our algorithm has the advantage that the M-step is cheap in relation to the E-step since it does not rely on the large numbers of samples in the inner iterations. We have demonstrated the efficiency of the new model by numerical examples, in the case of 2D and 3D images. They confirm that PCA-GMM is an efficient way of reducing the dimension of the patches, while keeping almost the same quality of the results than with a GMM algorithm. This dimensionality reduction is of the utmost importance when dealing with 3D images, where the size of the data gets very large.

The second contribution of this chapter, called PCA-GGMM, incorporates the PCA model with the GGMM model on the low-dimensional data. Experiments with 2D, 3D synthetic, and real material images demonstrated the effectiveness of the PCA-GGMM models. They showed that our model gives a similar quality for the results as MMSE-GGMM and is better than the PCA-GMM model of [41]. Furthermore, our method, which estimates the parameters of the PCA-reduced generalized model, could be extended to the case of elliptical distributions such as the Student-t distribution or any other one.

In recent years, many deep learning (DL)-based SISR methods have enabled many advances in super-resolution topic. We are aware of the effectiveness of deep learning approaches for super-resolution, e.g. [25, 58, 90, 56]. In the next chapter, we will consider some deep learning approaches for super-resolution with high magnification factor and high-dimensional data.

?CHAPTERNAME? 6

A generative adversarial network with contrast change challenge

Contents

This chapter focuses on the SR method for the contrast change problem of the material image in a framework of generative adversarial networks (GANs), which has made many breakthroughs in image processing recently.

In particular, Sections 6.1, 6.2 provide a brief overview of the tools used. A presentation of convolutional neural networks (CNNs), and particularly of GANs, is followed by a state-of-the-art on GANs in order to describe more precisely the framework of this chapter.

Then in Section 6.3, a RDGAN-contrast method is proposed considering a generator network including two sub-networks in a GAN context. The first generator sub-network deals with the contrast change problem, and the second one focuses on the reconstruction of the HR image.

6.1 Convolutional neural network (CNN) for SISR

Recently, convolutional neural network (CNN) based SR methods have achieved significant improvements. Convolutional neural networks are a particular class of neural network designed for image processing. The first convolutional network

was proposed by LeCun et al. in 1989 [55]. The authors successfully applied backpropagation to train the convolution kernel coefficients to identify and recognize patterns within a series of handwritten zip codes. Specifically, the CNN architectures became popular following the success of the work by Krizhevsky et al. [52].

6.1.1 Convolutional neural network (CNN)

Convolutional neural networks (CNNs) are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. CNNs allow the relationship between input and output variables to be modeled efficiently by composing several layers. They have three main types of layers, which are *convolutional layers*, *pooling layers* and *fully-connected (FC) layers*. In one layer, each neuron is connected to all neurons in the next layer. The output of the neuron is defined by a weighted sum of all inputs received from the previous layer. This weighted sum is then passed through an *activation function* to produce the output for the next layer. In this section, we will detail the three main types of layers and the activation functions.

6.1.1.1 Activation functions

Activation function is a node that is placed at the end or between neural networks. This function aim to decide whether a neuron's response would fire or not.This means that if the output of any individual neuron exceeds a threshold value, that node will be activated, sending data to the next layer of the network. Otherwise, no data is transmitted to the next layer. There is a wide variety of different activation functions that can be used in neural networks. For the CNNs, we mention three main activation functions: sigmoid function, softmax function and rectified linear unit (ReLU) function.

Sigmoid function The logistic sigmoid function for $x \in \mathbb{R}$ is defined by

$$g(x) = \frac{1}{1 + \exp(-x)}.$$

Sigmoid function has a return value in the range 0 to 1 and is differentiable, which is useful when training a network. However, this function becomes bad for a reliable training neural network because of the problem of gradient degeneration to zero if the network is too deep. Therefore, the sigmoid function can cause a neural network to be blocked at training time.

Softmax function The softmax function is known as the normalized exponential function and can be regarded as a multiclass generalization of the logistic sigmoid function. The softmax function for a vector $x = (x_1, x_2, ..., x_K) \in \mathbb{R}^K$ is defined

when $K \geq 1$ by the formula

$$g(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{K} \exp(x)} \text{for} i = 1, ..., K.$$

Rectified Linear Unit (ReLU) function ReLU function is the most widely used activation function in neural networks today. ReLU is simply a max operation between an input and 0

$$g(x) = \max(0, x).$$

This function has less vanishing gradient problem than the sigmoid activation function, since the gradients will not become zero at least in one direction.

6.1.1.2 Convolutional layers

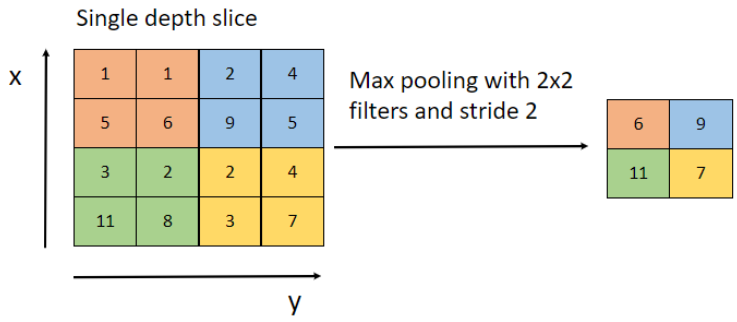
The convolutional network starts with the convolutional layers. These can be followed by additional convolutional layers or pooling layers, with the fully-connected layer being the final layer.

Convolutional layer The convolutional layer is the core building block of CNN and it accounts for the majority of the computation. The main objective of convolution is to extract features of different complexity from the input image. The extraction of these features is adapted to the problem considered by learning the weights of each convolutional layer. It is the automatic learning of weights that makes convolutional networks very efficient. The objective of these layers is to search for all the features of the input images by convolutional filtering. Thus, the resulting feature map can be seen as a filter that indicates where the features of interest are located in the image.

Let us assume that the input will be a gray image, which is made up of a matrix of pixels in two-dimensional (2D). This means that the input will have two dimensions - height, width. The feature detector is a 2D array of weights that represents part of the image. The filter is then applied to an area of the image, and a dot product between the filter and the input pixels is calculated. This dot product is then passed into an output array. The filter then shifts by a stride, repeating the process until the kernel has scanned the entire image.

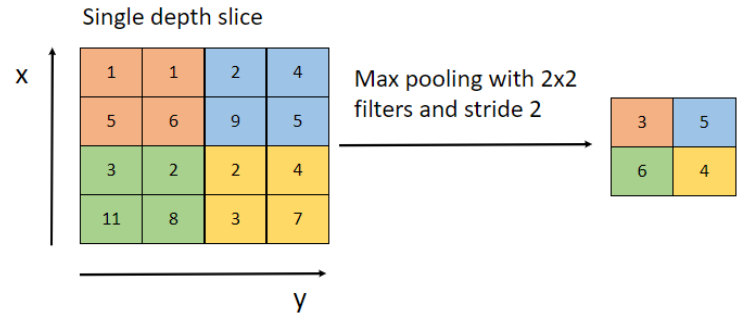
Pooling layer The pooling layer is used to reduce the spatial dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Two types of pooling used are:

- **Max pooling:** It selects the pixel with the maximum value for each neighbourhood to send to the output array (Figure 6.1). In this way, it keeps the strongest features. This approach tends to be used more often compared to average pooling.



?figurename? 6.1: Scheme of max pooling

- **Average pooling:** It calculates the average value in each neighborhood and sends to the output array (Figure 6.2).



?figurename? 6.2: Scheme of average pooling

Although a lot of information is lost in the pooling layer, it reduces complexity, improves efficiency and reduces the risk of overfitting.

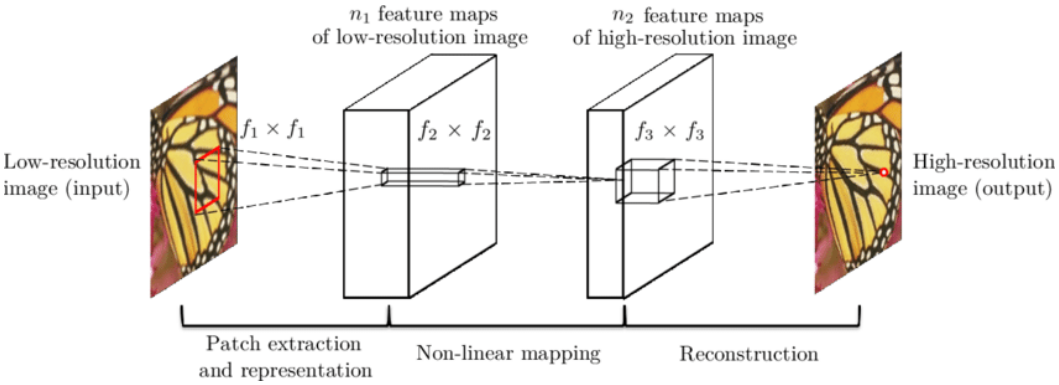
Fully-connected (FC) layer Fully connected layers connect every neuron in one layer to every neuron in another layer. This layer performs the classification task based on the features extracted by the previous layers and their different filters.

While convolutional and pooling layers tend to use ReLU functions, FC layers usually use the softmax activation function to calculate the probability distribution from 0 to 1.

6.1.2 Super-resolution CNN method (SRCNN) [24, 25]

In this section, we mainly discuss the network architecture of super-resolution CNN (SRCNN), which was proposed by C.Dong et al. [24, 25]. SRCNN has three main parts: patch extraction and representation, non-linear mapping, and reconstruction, as shown in Figure 6.3. The input of this network is a bicubic-interpolated image, which is upscaled from the low-resolution image and has the same resolution of the high-resolution image. We denote the bicubic-interpolated image as X_L and

the ground truth high-resolution image as X_H. The bicubic-interpolated image is convolved by learned non-linear filters in different layers, merged to reconstruct the high-resolution image.



?figurename? 6.3: Sketch of the SRCNN architecture [24].

Patch extraction and representation This step extracts the overlapping patches from the interpolated image X_L and represents each patch as a high-dimensional vector. The first layer performs a standard convolution with a ReLU activation function on the filter responses and is expressed as an operation $F1$.:

$$F_1(X_L) = \max(0, W_1 * X_L + B_1) \tag{6.1.1}$$

where W_1 and B_1 represent the filters and biases, respectively, and $*$ denotes the convolution operation. Here, W_1 corresponds to n_1 filters of support $c \times f_1 \times f_1$, where c is the number of channels in the input image, f_1 is the spatial size of a filter. B_1 is an n_1-dimensional vector, each element of which is associated with a filter. Intuitively, W_1 applies n_1 convolutions on the image, then combines with the n_1-dimensional bias vector B_1 to increase degrees of freedom by 1. Finally, the output is composed of n_1 feature maps and will be passed to the second step.

Non-linear mapping The purpose of this step is to increase the resolution of each patch to obtain a high-resolution patch. In the second operation, we map each of the n_1-dimensional vectors from the first layer to a n_2-dimensional one. The operation of the second layer is as follows:

$$F_2(X_L) = \max(0, W_2 * F_1(X_L) + B2). \tag{6.1.2}$$

W_2 contains n_2 filters of size $n_1 \times f_2 \times f_2$, and B_2 is n_2-dimensional bias vector. Each n_2-dimensional output vector is a representation of a HR patch that will be used for the reconstruction step.

Reconstruction The reconstruction step aggregates all HR patches to restore the HR image by using the following convolutional layer

$$F_3(X_L) = \max(0, W_3 * F_2(X_L) + B3). \tag{6.1.3}$$

W_3 and B_3 here correspond to c filters of size $n_2 \times f_3 \times f_3$ and c-dimensional bias vector, respectively. This layer therefore reconstructs the HR image by analogy with the weighted average for high-resolution patches.

Although the three operations of the SRCNN architecture are motivated by different objectives, they all lead to the same form of convolutional layer with a ReLU activation function. In this model, all filter weights and biases $\Theta = \{W_1, W_2, W_3, B_1, B_2, B_3\}$ are learned by minimizing an objective function related to the mean squared error (MSE) between the HR training image and the corresponding reconstructed HR image:

$$\mathcal{L}(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \|F_3(X_{L,i}) - X_{H,i}\|^2 , \tag{6.1.4}$$

where $X_{L,i}, X_{H,i}$ are given LR images and HR images for $i = 1, ..., n$ training samples. The formulation of SRCNN is relatively simple and is superior to concurrent traditional methods.

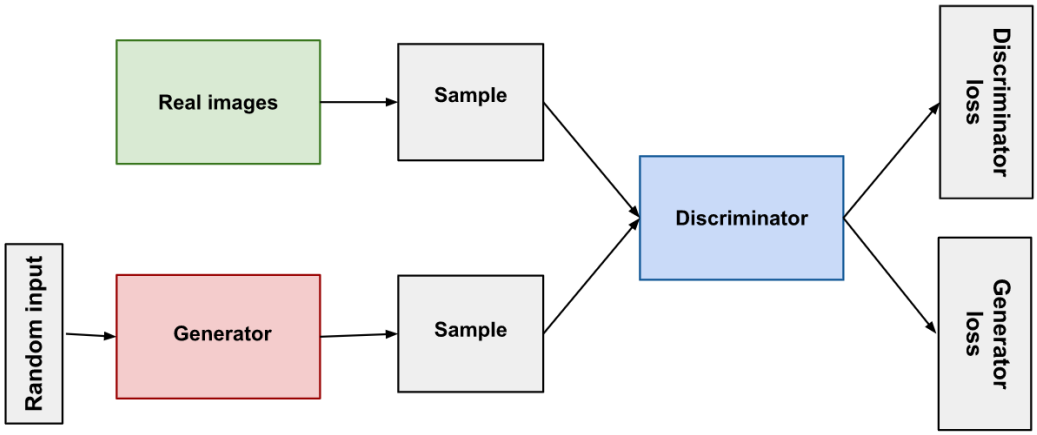
6.2 Generative Adversarial Neural Network (GAN)

6.2.1 Generative adversarial neural network

Generative Adversarial Network (GANs) is a class of unsupervised learning algorithms, introduced by Goodfellow et al [35] in 2014. In general, a GAN is a generative model in which two networks are put in competition, as shown in Figure 6.4. A generator model for generating new examples, and a discriminator model for classifying whether the generated examples are real, or fake, generated by the generator model. Two neural networks compete in a zero-sum game where one agent gains while the other loses.

Generator The generator (G) learns to generate plausible data. The input of this network is a random sample, and the output is a generated sample which will be used in the discriminator network. The objective of the generator is to fool the discriminator network into thinking that the generated sample is real.

Discriminator The discriminator (D) learns to distinguish the fake data of the generator from the real data. This network is a basic classification network that returns the probability that the input sample belongs to the database (that it is real). Ideally, we consider a sample y. $D(y) = 1$ if y belongs to the database and $D(y) = 0$ if y is a generated sample.



?figurename? 6.4: Sketch of the GAN architecture [35].

Loss function To define the generator G and discriminator D, we assume that x is ground truth data and that z is observed data. The objective of D is to properly classify the real and generated data, this amounts to maximizing equation

$$\max_D \log(D(x)) + \log(1 - D(G(z))). \tag{6.2.1}$$

At the same time, the objective of G is to deceive the discriminator, that is, G is trained to minimize the probability that the discriminator makes the correct prediction for a generated data set, so we want $D(G(z)) = 1$. This amounts to having to minimize $\log(1-D(G(z)))$ with respect to the parameters of G. Therefore, D and G play the following two-player minimax game with the value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \min_G \max_D \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[\log(1 - D(G(z)))]. \tag{6.2.2}$$

The optimization of the weights of the generator θ_G and the discriminator θ_D is carried out in an alternative way, optimizing in D and then in G from equation (6.2.2) as in Algorithm 4 [35]. In this algorithm, the parameters are updated using the stochastic gradient descent method with their stochastic gradient:

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))) = \frac{1}{m} \sum_{i=1}^{m} \left[-\frac{1}{1 - D(G(z^{(i)}))} D'(G(z^{(i)})) \nabla G(z^{(i)}) \right], \tag{6.2.3}$$

$$\begin{aligned} &\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)}))) \right] \\ &= \frac{1}{m} \sum_{i=1}^{m} \left[\frac{1}{D(x^{(i)})} \nabla D(x^{(i)}) - \frac{1}{1 - D(G(z^{(i)}))} \nabla D(G(z^{(i)})) \right], \end{aligned} \tag{6.2.4}$$

where $\{z^{(1)}, \ldots, z^{(m)}\}$ are the observed samples and $\{x^{(1)}, \ldots, x^{(m)}\}$ are real examples.

Algorithm 4 Minibatch stochastic gradient descent training of generative adversarial nets [35]

Input: noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ and examples $\{x^{(1)}, \ldots, x^{(m)}\}$
for number of training iterations **do**
for k steps **do**

- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}(x)}$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)}))) \right].$$

end for

- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))).$$

end for

In practice, the hyperparameter k is chosen equal to 1 and the Stochastic Gradient Descent SGD step is done using the ADAptive Moment estimation (ADAM) algorithm [51]. ADAM algorithm is an adaptive stochastic gradient descent optimization algorithm specifically designed for training deep neural networks. It is an algorithm in which the gradient used in each iteration is updated from the previous one using a moment-based technique.

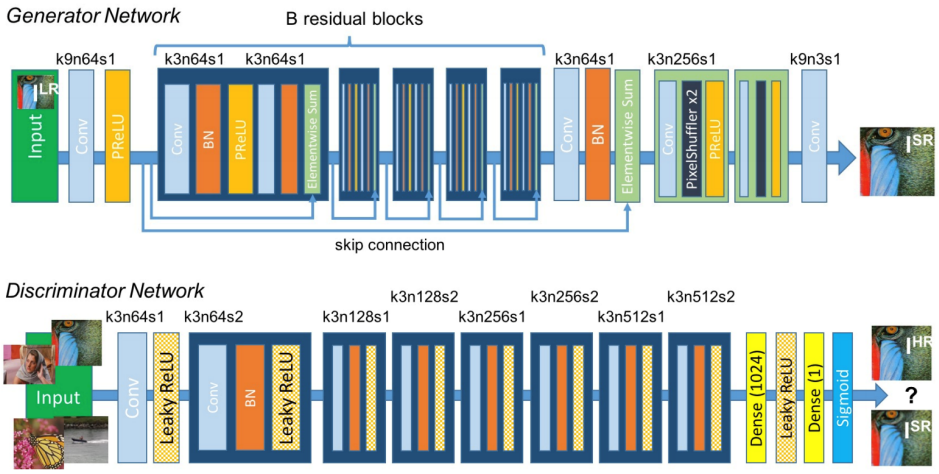
6.2.2 SRGAN method [56]

For the super-resolution task, Ledig et al. [56] pioneered a super-resolution method using GAN. This method is called SRGAN. The core concept of GAN is retained in SRGAN, i.e. the generator G generates a estimated HR image (conditionally for certain LR images) and the discriminant operator classifies it as a ground truth HR image or fake HR image.

6.2.2.1 SRGAN Architecture

In order to obtain generator G and the discriminator D, the authors use a GAN architecture (as shown in Figure 6.5) for the generator and a classification network

combining dense and convolutional layers for the discriminator.



?figurename? 6.5: Architecture of the GAN [56] with the corresponding kernel size k, the number of feature maps n and the stride s indicated for each convolution layer.

Generator Architecture The generator architecture contains B residual blocks with 16 identical layouts, created by a residual neural network (ResNet). Specifically, within the residual block, two convolutional layers are used, with small 3×3 kernels and 64 feature maps followed by batch-normalization (BN) layers [48] and ParametricReLU (PReLU) [38] as the activation function. The PReLU adaptively learns rectifier parameters and improves accuracy with negligible computational cost. Then, the LR image is up-resolution using two trained sub-pixel convolutional layers [76]. Finally, the output of the generator architecture is passed into the discriminator.

Discriminator Architecture The task of the discriminator is to discriminate between real HR images and generated SR images. This architecture contains 8 convolutional layers with an increasing number of 3×3 filter kernels, increasing by a factor of 2 from 64 to 512 kernels. In this network, the LeakyReLU activation function is used. To reduce the resolution of the image for distinguishing step, we use the strided convolutions each time the number of features is doubled. Then two dense layers with a ReLU function are used in between. Finally, a sigmoid activation function is added at the end of the network to obtain a probability for sample classification.

6.2.2.2 Loss function

The SRGAN uses the perpetual loss function l^{SR} for the generator network. The perceptual term is motivated by the fact that a perceptual loss function gives visually

better results than a loss function in pixel space. The authors then propose the loss function, which is the weighted sum of two loss components: content loss and adversarial loss.

$$l^{SR} = l_X^{SR} + 10^{-3} l_{Gen}^{SR}, \tag{6.2.5}$$

where l_X^{SR} is a content loss and l_{Gen}^{SR} is a adversarial loss.

Content loss Let the LR image X_L that serves as input to the generator network G and the reference image X_H, the content loss function is a VGG loss based on the ReLU activation function of the pre-trained 19-layer VGG network [77].

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{m=1}^{W_{i,j}} \sum_{n=1}^{H_{i,j}} [\phi_{i,j}(X_H)_{m,n} - \phi_{i,j}(G(X_L))_{m,n}]^2 \tag{6.2.6}$$

$W_{i,j}$ and $H_{i,j}$ correspond to the size of the image at the output of the layer $\phi_{i,j}$.

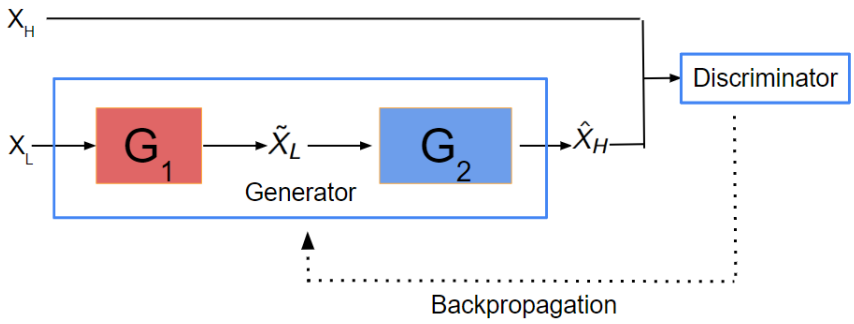
Adversarial loss The adversarial loss is the loss function that forces the generator to an image more similar to HR image by using the trained discriminator network to differentiate realistic images from unrealistic ones. This loss function is expressed as

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log(D(G(X_L))) \tag{6.2.7}$$

where N corresponds to the batch size.

6.3 Contribution: Residual dense GAN (RDGAN) network for contrast change problem of super-resolution

The problem of changing contrast is one of the important challenges of SR for materials images. In the case where the transformation between LR and HR image is an affine transformation, we can estimate the contrast change parameters of this transformation as described in Chapter 3. Moreover, in Chapter 4 we proposed an SR method, the MMSE-GGMM method, which is invariant under affine transformations of the observation. However, we are not sure that the change of contrast is always characterized by a simple affine transformation. It would be interesting to show some examples of non-affine transformations. To deal with the non-affine transformation, we propose an SR method based on GAN considering a bi-generator network that includes two generator sub-networks. The first generator deals with the contrast change problem, and the second is to reconstruct the high frequencies. The architecture can be summarized as in Figure 6.6:



?figurename? 6.6: Scheme of the proposed architecture, where X_H is the HR image, X_L the LR image, $\tilde{X}_L$ the LR image with the good contrast and $\hat{X}_H$ the SR image.

6.3.1 Generator

6.3.1.1 First generator G_1

In the SR training step for material images, we have a data set of HR and LR pair images, which have different contrast and brightness. The first generator of our GAN network aims to transfer the contrast of the LR image X_L to a contrast as good as the HR image X_H. It means, it will be necessary to associate each pixel of LR image with the pixels of the HR image of which it is closest in terms of distance. To do this, we will use optimal transport for the LR image and downsampling of the HR image so that its resolution is equal to the resolution of the LR image. We denote the downsampling of the HR image as $\downarrow X_H$. The way to achieve the optimal overall cost is to change the range of the LR image to be the same as the HR image. Therefore, we minimize the distance between the sorted value in pixels of two images X_L and $\downarrow X_H$. The first loss function corresponding to G_1 is:

$$\mathcal{L}(G_1) = \sum_{n=1}^{N} ||sort(\downarrow X_H) - sort(\tilde{X}_L)||_1 \tag{6.3.1}$$

6.3.1.2 Second generator G_2

The second generator sub-network takes care of the high frequency reconstruction and returns the final HR image $\hat{X}_H$. As the goal of SR for materials images is to understand textures and interfaces in materials, we consider that the SR approach focuses on the image geometry. To enhance the geometric reconstruction of the images, we propose to consider a regularization term in the generator loss function. This term was initially proposed by Ballester et al. [8] in a variational framework. We then use the loss function, which is proposed by Gastineau et al. [30], for this sub-network with three terms of the loss function such as adversarial loss, content loss, and geometric loss. The loss function for the second sub-network G_2 is expressed as

follows:

$$\mathcal{L}(G_2) = \sum_{n=1}^{N} -log(D(\hat{X}_H)) + \alpha_{l1}||\hat{X}_H - X_H||_1 + \alpha_g \sum_{x \in \Omega} |\langle \nabla X_H(x)^{\perp}, \nabla \hat{X}_H(x) \rangle| \tag{6.3.2}$$

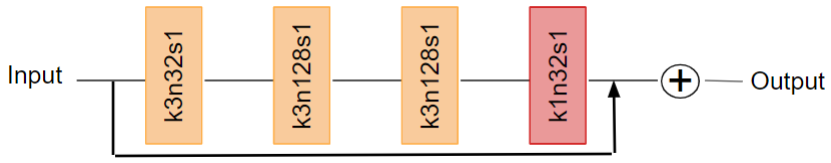
where N corresponds to the batch size, $\nabla(\cdot)$ is the gradient, $\perp$ the orthogonal vector, Ω the image domain, and α_{L_1}, α_g are the content weight and the geometric weight respectively. The first term is the adversarial loss that is associated with the discriminator. The second term is the l_1 norm between the target and the reconstructed image. This third term forces the alignment of the gradient of the reconstructed image with the gradient of the target image. This allows the geometry of the reference image to be transferred to that of the one we are trying to reconstruct. Then the scalar product is used for the orthogonal gradient vector of the target image and the gradient vector of the reconstructed image at any point in the image domain. Indeed, a zero scalar product indicates that the vectors are collinear and therefore the direction of the gradients is preserved.

6.3.1.3 Architecture of generator

First generator The architecture of the first generator, presented in Figure 6.7, is composed of two convolutional layers and $RB_i, i = 1, \cdots, p$ residual blocks.

?figurename? 6.7: Architecture of the the first generator, where blocks RB_i, $i = 1, \cdots, p$, are p residual blocks, k is the kernel size, n the number of filters and s the stride of the convolutional layers.

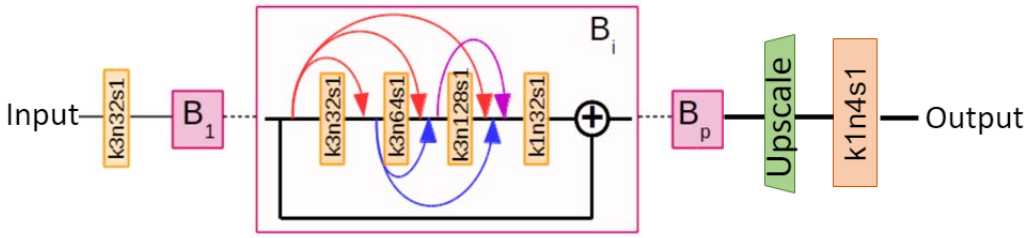
The architecture starts with a convolutional layer with 32 3×3 filter kernels, 1 stride, and ReLU activation function. This is followed by p residual blocks RB_i as in Figure 6.8. In each residual block, we use 3 convolutional layers with an increasing number of 3×3 filter kernels from 32 to 128, stride 1, and ReLU as the activation function. This block is terminated by a convolutional layer with 32 1×1 filter kernel, stride 1. The idea of the residual network is that each layer of the neural network is only responsible for fine-tuning the output of a previous layer by simply adding the residual learned at the previous layer to the input of that layer.



?figurename? 6.8: Architecture of each residual block $RB_i, i = 1, \cdots p$.

Finally, the architecture of our first generator is completed with a convolution layer with one 1×1 filter kernel and stride 1.

Second generator We consider the same architecture as in the RDGAN method [30], a residual dense architecture as in Figure 6.9.

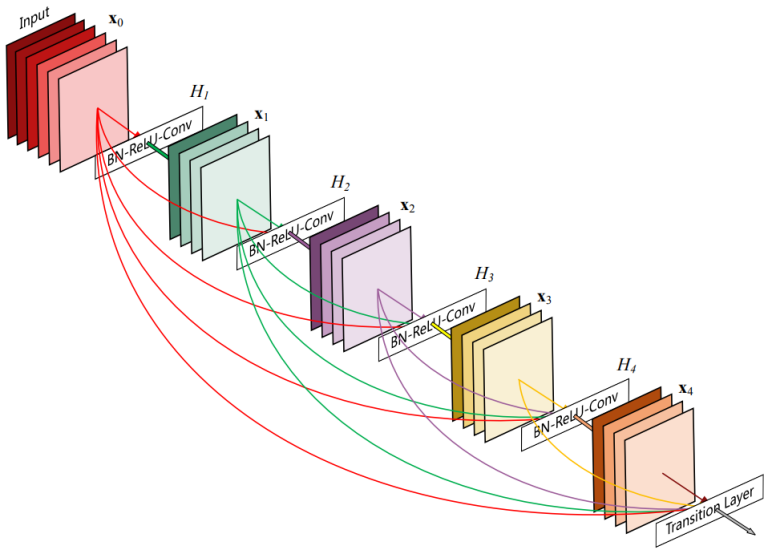


?figurename? 6.9: Architecture of the second generator, where blocks B_i, $i = 1, \cdots, p$, are p residual dense blocks, k is the kernel size, n the number of filters and s the stride of the convolutional layers. Arrows represent the dense connections and the $\oplus$ residual connection.

The architecture starts with a convolutional layer with 32 3×3 filter kernels and a ReLU activation function. This is followed by p residual dense blocks $B_i, i = 1, \cdots p$, which retain the advantages of Densenet [46] and residual [39] architectures. Thus, this architecture can avoid the problem of vanishing gradients that is often encountered during the training step. Each residual dense block is composed of four convolutional layers. In each block, the input to the l^{th} layer consists of the feature maps of all previous layers. It means that if we note H_l the non-linear transformation given by layer l of the network and x_l the output of this layer, the l^{th} layer receives the feature maps of all the previous layers, $x_0, x-1, \cdots, x_{l-1}$, as input:

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]) \tag{6.3.3}$$

where $[\cdot]$ represents the concatenation, and $[x_0, x-1, \cdots, x_{l-1}]$ refers to the concatenation of the feature-maps produced in layers $0, \cdots, l-1$.



?figurename? 6.10: Architecture of the Dense network. A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

In addition, each of these blocks consists of a residual connection. This connection is represented by adding the input of each block with the output of the last layer. Therefore, the input information for each block is transmitted without modification.

After extracting the features in the LR space, we up-sample these features into the HR space by using the sub-pixel convolutional layer as proposed by Shi et al. [76]. Finally, the architecture of our second generator is completed with one convolution layer with one 1×1 filter kernel and stride 1 as in the first generator.

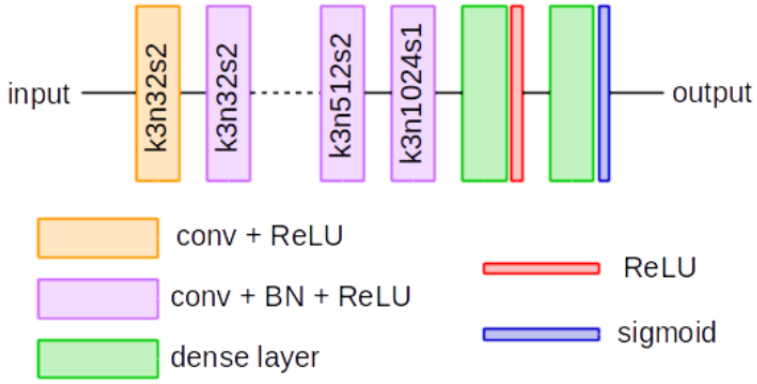
6.3.2 Discriminator

Loss function For the discriminator, we maximize the loss function as follows:

$$\mathcal{L}(D) = \sum_{n=1}^{N} log(1 - D(G(X_L))) + log(D(X_H)). \tag{6.3.4}$$

Architecture of discriminator We consider the same architecture of the discriminator network as in the MDSSC-GAN SAM method [31]. The first convolutional layer of the architecture performs a standard convolution with 32 feature maps, 1 stride, and a ReLU activation function. This is followed by an architecture containing 6 convolutional layers with an increasing number of 3×3 filter kernels, increasing by a factor of 2 from 32 to 1024 kernels. Each layer is followed by the BN layer and the ReLU activation function with stride 2, except the last layer with 1 stride. These convolutional layers are used to extract sufficient features and capture the representation of data in a space. Then, two dense layers

are used to classify whether the generated image is real or fake as in the SRGAN architecture.



?figurename? 6.11: Architecture of the discriminator, where k is the kernel size, n the number of filters and s the stride of the convolutional layers.

6.3.3 Numerical results

Database In this contribution, we implement the RDGAN-contrast method for our material images: FS and Sic diamonds. Due to the complexity of the network, we cannot train the network from overlapping patches of size 4×4 as in the MMSE-GGMM experiment. Therefore, we train the network with a patch size of 64×64 for LR patches. This leads to the use of the training database from a 2D image for training is not sufficient. Fortunately, our 3D image was acquired with several slices of the 2D image. For instance, we obtain a 3D HR image which has a size of $1652 \times 1652 \times 40$. This gives us 40 2D HR images of size 1652×1652. Then, for each 2D HR image, we have a corresponding 2D LR image of size 826×826. To generate the database for the training step, we use the upper left quarter of 40 pairs of HR and LR images as in the top row of Figure 4.1. Finally, we extract the quarter of these HR images into a set of HR patches with patch size 128×128 without overlapping and LR patches with size 64×64. Therefore, we can obtain a training set with 1960 pairs of HR and LR images.

For the test process, we reconstruct the HR image for the entire image with a pixel size of 1652×1652 from one of the 40 2D images that we obtained from the original 3D image. We again extract this image and the corresponding LR image into a set of HR patches of size 128×128 and LR patches of size 64×64 to feed to the generator network. Thus, we will apply our network to these 225 pairs of HR LR images for testing.

Quality evaluation To evaluate the results, we use the Peak Signal to Noise Ratio (PSNR) value in equation (4.2.27) of two images X and Y. In addition, in order to compare the geometric similarities of two images X and Y, we consider the

metric of total variation. The total variation (TV) is given as

$$TV\,(X,Y) = \frac{1}{|\Omega|}\sum_{\Omega}||\nabla X - \nabla Y||_2 = \frac{1}{|\Omega|}\sum_{\Omega}\sqrt{|\nabla_x X - \nabla_x Y|^2 + |\nabla_y X - \nabla_y Y|^2}, \tag{6.3.5}$$

where $|\Omega|$ is the number of pixels. The l_2 norm

$$||X - Y||_2^2 = \frac{1}{N^2}\sum_{i}^{N}\sum_{j}^{N}|X_{ij} - Y_{ij}|^2. \tag{6.3.6}$$

The metric of total variation allows us to compare the edges of two images.

Details of implementation The proposed method is implemented with Tensorflow 1.2 and uses the ADAM algorithm to optimize the weights of the networks with an initial learning rate of 0.0002 and a momentum of 0.5. Finally, the batch size is adjusted to 10 for our database. Parameters α_g, α_{l1} are optimized to get the best balance between all the metrics.

Results We compare the proposed method with some state-of-the-art SR methods. The SR methods in comparison are the bicubic interpolate method, MMSE-GGMM method, RDGAN and our proposal RDGAN-contrast method. For a better comparison, we train both networks on the FS and Sic Diamonds images, which have a contrast change between the LR and HR images. We know that the MMSE-GGMM and RDGAN-contrast methods can directly solve the contrast change problem in the case of an affine transformation and a non-affine transformation, respectively. But the bicubic and RDGAN methods cannot. Therefore, to make the comparison fair for the bicubic and RDGAN methods, we first estimate the contrast change parameters using the estimation method as in Section 3.2.2 for all pairs of HR and LR images from the training data set. Thus, we apply the affine transformation within these parameters to the reconstructed HR images, which were obtained by the bicubic and RDGAN method.

	FS		Sic Diamond	
Quality evaluation	PSNR	Total variation	PSNR	Total variation
ideal value	max	min	max	min
Bicubic	32.01	0.0168	24.94	0.0299
MMSE-GGMM	33.68	0.0152	27.22	0.0266
RDGAN	**33.76**	0.0153	27.33	0.0261
RDGAN-Contrast	33.71	**0.0151**	**27.62**	**0.0259**

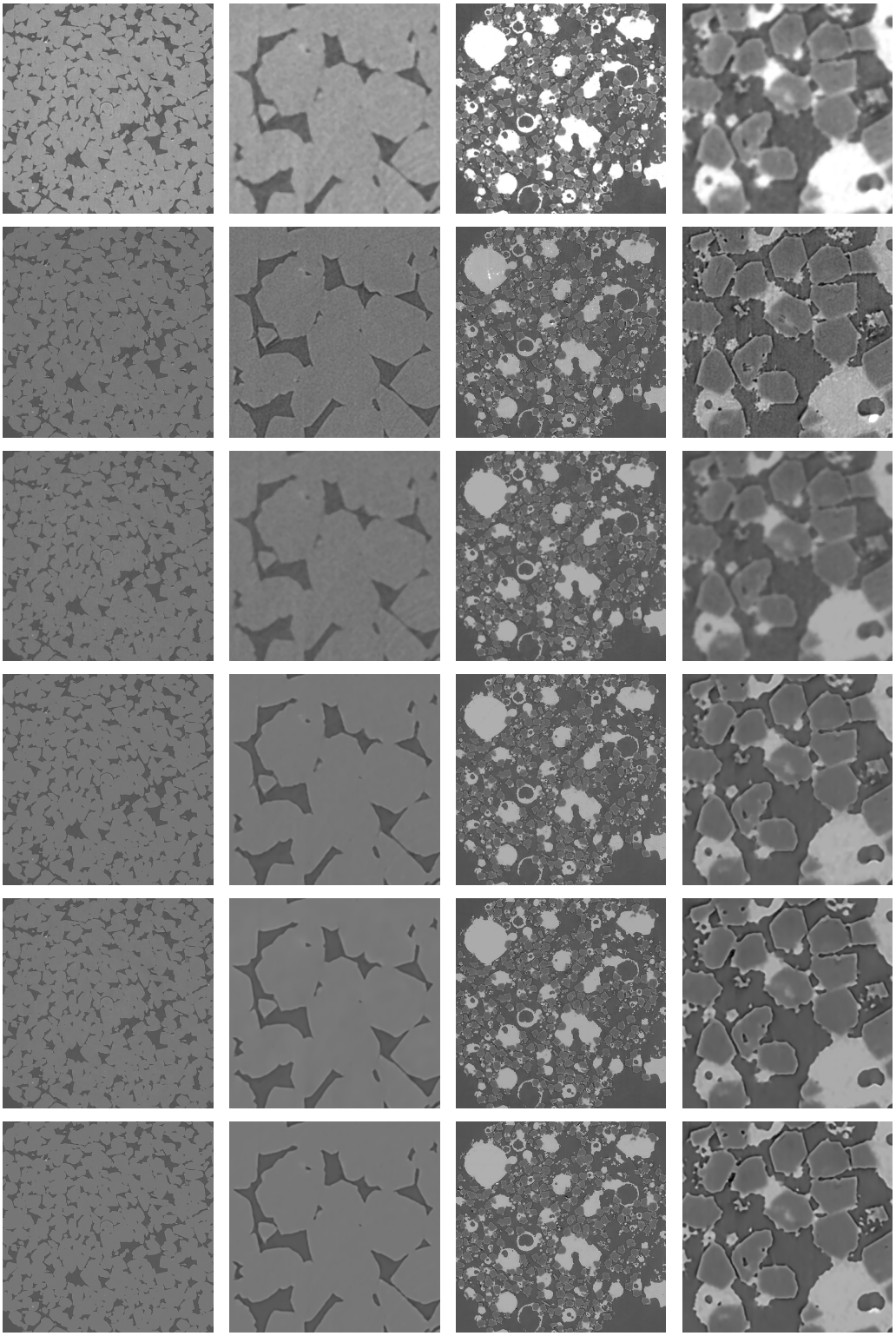
?tablename? 6.1: Quantitative results obtained on FS and Sic Diamonds images, the best results are in bold.

Table 6.1 shows the PSNR and the metric total variation values of the

reconstruction of the FS and Sic Diamonds images using either bicubic interpolation, MMSE-GGMM, RDGAN, and RDGAN-contrast with magnification factor $q = 2$. For the bicubic and RDGAN methods, we first estimate the contrast parameters from the training data. Then we apply the affine transformation to the recontructed HR images. In the FS image, the PSNR value of the RDGAN method is slightly higher than our proposal RDGAN-contrast. Its TV value is lower than that of RDGAN-contrast. However, both the PSNR and TV values of our RDGAN-contrast method are superior to the other methods for the Sic Diamond image. This means that when the contrast transformation is affine, the RDGAN-contrast method can obtain results similar to those of the MMSE-GGMM and RDGAN methods. But when the contrast transformation is a non-affine, the RDGAN-contrast is more efficient than the other methods both in contrast and geometry. This can be seen in more detail in Figure 6.12. In addition, in the case of FS image, our MMSE-GGMM method results in only 0.1 dB lower than the RDGAN method. However, for model learning, the MMSE-GGMM method only needs to extract the database from a 2D image. As for the RDGAN and RDGAN-contrast methods, they need to use the overlapping patches from 40 2D images.

6.4 Conclusion

In this chapter, we have studied the deep learning approach for super-resolution. We present a new SR method based on a residual dense generative adversarial neural network (RDGAN) with the contrast change problem, called RDGAN-contrast. To solve the contrast change problem, our method considers two generator sub-networks. The first sub-network focuses on the contrast change problem and the second one reconstructs the high frequencies. The results show a visual improvement with our proposal RDGAN-contrast over the RDGAN method and the MMSE-GGMM method in the case of non-affine transformation. However, in the case of the affine transformation, the MMSE-GGMM method should be preferred to the RDGAN-contrast method.



(a) FS (b) FS part (c) Sic Diamonds (d) Sic Diamonds part

?figurename? 6.12: Reconstructions of 2D low-resolution images with a magnification factor $q = 2$. First row: LR image, second row: ground truth, third row: reconstruction with Bicubic, fourth row: reconstruction with MMSE-GGMM, fifth row: reconstruction with RDGAN, sixth row reconstruction with RDGAN-contrast.

?CHAPTERNAME? 7

Conclusions and perspectives

Contents

7.1 Conclusions

In this thesis, we are interested in solving the super-resolution problem for material images using learning methods.The three main challenges of the SR problem for material images are the contrast change problem, the problem of undefined corruption operators, and the high dimensionality of the data. We have proposed four contributions based on the MMSE estimator for GGMM, the PCA method for GMM and GGMM, and the GAN method to solve these challenges.

The first proposed method [69] mainly addresses our first dual challenge of material images by considering the MMSE estimator with the GGMM model. The results obtained for the FS and Sic Diamonds images have shown that the MMSE-GGMM method is invariant with the contrast change problem under affine transformations of the observation. We also proved that our SR method using the MMSE estimator with the GGMM model improves the results over the GMM model in both synthetic and material images.

The second contribution introduced a method to deal with the last challenge of our material images. This is the problem of high dimensionality. Our contribution [41] combined the GMM model with a reduction in the dimensionality of the data in each component of the model using the principal component analysis method, called PCA-GMM. Next, this PCA-GMM model is applied to the super-resolution of 2D and 3D material images based on the MMSE-GMM method. The numerical results of this second contribution confirm the gain of the PCA-GGMM method in reducing the dimension of the patches when we learn the mixture model. In addition, they maintain almost the same quality of results as with a GMM algorithm for the super-resolution problem.

Following these second results, our third contribution [70] is an extension of the second one by focusing on the GGMM model with the PCA method. Due to the complexity of the GGMM model, it is not easy to consider a GGMM in conjunction

with PCA in the same way as in the second contribution. This PCA-GGMM takes advantage of the flexibility of the "Half-Quadratic splitting" technique to tackle the minimization problem in the M-step of the EM algorithm. Then the PCA-GGMM model can be used for the super-resolution model based on the MMSE method for GGMM as in the first contribution. The PCA-GGMM model produces results almost as good as the MMSE-GGMM model and even slightly better than the MMSE-GMM model.

Finally, the last proposed method mainly focuses on SR method for material images with the contrast change problem based on the deep learning approach, specifically the SRGAN method. Indeed, this contribution considers two generators, each responding to one of the two problems: contrast change and the super-resolution task. The results of the RDGAN-contrast method obtained on the material images are encouraging because we can observe a quantitative gain confirmed by the visual results in both contrast and texture. However, in the case of the affine transformation, the RDGAN-contrast method requires a larger database and is more computationally complex than the MMSE-GGMM method, but it is slightly better than the MMSE-GGMM method. Thus, if the contrast change is a non-affine transformation or if we have a large database, we prefer to use the RDGAN-contrast method. Conversely, if the contrast change is an affine transformation or if we have little data, the MMSE-GGMM method should be preferred.

7.2 Perspectives

This thesis provides some proposals for the super-resolution problem with several challenges to material images. However, some other possibilities remain unexplored such as the SR method with the elliptical mixture model, the super-resolution problem with a high magnification factor, and the convergence of the GANs.

A first perspective could be to consider a dimensionality reduction method within mixture models of elliptical distributions. Indeed, throughout this thesis, we have been interested in dimensionality reduction methods for the GMM and GGMM models. The generalized Gaussian distribution is a special case of the elliptical distribution. Thus, the PCA-GGMM model can be generalized to the PCA-reduced elliptical mixture model by combining PCA with the elliptical mixture model. Then the PCA-reduced elliptical mixture model can be used for super-resolution task by using the MMSE estimator following the same methodology as the MMSE-GGMM method.

The second perspective would be to study the convergence of the GANs. In the last contribution, we have proposed an SR method based on GANs for the contrast change problem. On the other hand, GANs are rather unstable algorithms and may have difficulty in converging. The reason for these problems is that one of the networks can overtake the adversarial network. In this case, the less efficient network will not be able to progress. Thus, the training will become unstable. In

this thesis, we try to solve this issue by using the residual dense network for the second generator. Recently, a new method [87] using adaptive weighting for the discriminator has been developed to solve the convergence problem. The author has shown that this method successfully remedies the problem of instability and convergence of GANs. Indeed, our RDGAN-contrast network can be improved by using the discriminator loss, which consists of two different weighted parts: the real and the fake terms.

From the material imaging point of view, the magnification factor can be up to 5, 10. As the scale factor increases, some essential details may be lost, which has a serious impact on the material image. The large scale factor is also one of the major challenge in the SR community and there is still a lack of effective remedies. Recently, some methods for super-resolution with large scale factor were proposed such as StyleGAN [49], PULSE [65] method, and GLEAN [16] method. These methods significantly improve the restoration quality of large-factor image super-resolution by using the latent space. These methods have achieved the improvements in fidelity and more realistic textures than existing methods for SR with a large scale factor. These contributions open up a very promising direction for image SR with a high scale factor, especially for material images.

7.3 List of publications

International journal

- J. Hertrich, D-P-L. Nguyen, J-F. Aujol, D. Bernard, Y. Berthoumieu, A. Saadaldin, and G. Steidl. PCA Reduced Gaussian Mixture Models with Applications in Superresolution. *Inverse Problems in Imaging*, 16 (2) : 341-366. doi: 10.3934/ipi.2021053, 2022.

Submitted journal

- D-P-L. Nguyen, J. Hertrich, J-F. Aujol and Y. Berthoumieu. Image super-resolution with PCA Reduced generalized Gaussian mixture models. *HAL Preprint hal-03664839*, 2022.

Conference paper

- D-P-L. Nguyen, J-F. Aujol and Y. Berthoumieu. Image Super Resolution with a Joint Generalized Gaussian Mixture Model. *XXVIIIe Colloque Francophone de Traitement du Signal et des Images (GRETSI)*, September, 2022.

?bibname?

[1] C. Aguerrebere, A. Almansa, J. Delon, Y. Gousseau, and P. Muse. A Bayesian hyperprior approach for joint image denoising and interpolation with an application to HDR imaging. *IEEE Transactions on Computational Imaging,* 633-646, 2017. (Cited on pages 2 and 10.)

[2] M. S. Allili. Wavelet modeling using finite mixtures of generalized Gaussian distributions: Application to texture discrimination and retrieval. *IEEE Transactions on Image Processing*, 21(4):1452-1464, 2012. (Cited on pages 2, 10 and 34.)

[3] F. Altekruger, J. Hertrich. WPPNets: Unsupervised CNN Training with Wasserstein Patch Priors for Image Superresolution. *arXiv preprint arXiv:2201.08157*, 2022. (Cited on pages 43 and 44.)

[4] H. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1647-1659, Oct. 2005. (Cited on page 24.)

[5] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming. A Publication of the Mathematical Programming Society*, 116(1-2, Ser. B):5–16, 2009. (Cited on page 67.)

[6] H. Attouch, J. Bolte, and B.F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming* 137.1, pp. 91-129, 2013. (Cited on page 65.)

[7] K. J. Arrow, L. Hurwicz, and H. Uzawa. Studies in linear and nonlinear programming. *In Stanford Mathematical Studies in the Social Sciences*, volume II. Stanford University Press, Stanford, 1958. (Cited on page 72.)

[8] C. Ballester, V. Caselles, L. Igual, J. Verdera, and B. Rougé. A Variational Model for P+XS Image Fusion. *IJCV*, vol. 69, no. 1, pp 43-59, 2006. (Cited on page 95.)

[9] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2, Ser. A):459–494, 2014. (Cited on pages 64, 67 and 84.)

[10] Z. Boukouvalas, S. Said, L. Bombrun, Y. Berthoumieu, and T. Adali. A new Riemannian averaged fixed-point algorithm for MGGD parameter estimation. *IEEE Signal Processing Letters*, 22(12):2314-2318, Dec 2015. (Cited on pages 2, 10, 34, 35 and 37.)

[11] C. Bouveyron, S. Girard, and C. Schmid. High-dimensional data clustering. *Computational Statistics & Data Analysis*, 52(1):502–519, 2007. (Cited on pages vi, 47, 48, 52, 53, 57, 58, 76, 77, 78, 80 and 81.)

[12] A. Buades, B. Coll, and J-M. Morel. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation*, 4, 2005. (Cited on pages 2 and 10.)

[13] C. L. Byrne. The EM Algorithm: Theory, Applications and Related Methods. *Lecture Notes, University of Massachusetts*, 2017. (Cited on pages 34 and 35.)

[14] R. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2), pp. 245–276, 1966. (Cited on page 53.)

[15] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 275–282 , 2004. (Cited on page 25.)

[16] K. C. K. Chan, X. Wang, X. Xu, j. Gu, c. c. Loy. Glean: Generative latent bank for large-factor image super-resolution. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 14245-14254, 2021. (Cited on page 105.)

[17] S. Chrétien and A. O. Hero. Kullback proximal algorithms for maximum-likelihood estimation. *IEEE Transactions on Information Theory*, 46(5):1800–1810, 2000. (Cited on page 59.)

[18] S. Chrétien and A. O. Hero. On EM algorithms and their proximal generalizations. *ESAIM: Probability and Statistics*, 12:308–326, 2008. (Cited on page 59.)

[19] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft Edge Smoothness Prior for Alpha Channel Super Resolution *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, Jun. 2007. (Cited on page 24.)

[20] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos. Softcuts: a soft edge smoothness prior for color image super-resolution. *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 969–981, 2009. (Cited on page 24.)

[21] C.A. Deledalle, S Parameswaran, and TQ Nguyen. Image denoising with generalized Gaussian mixture model patch priors. *SIAM Journal on Imaging Sciences*, 11(4):2568-2609. (Cited on pages 2, 10, 34 and 43.)

[22] Julie Delon. Midway image equalization. *Journal of Mathematical Imaging and Vision*, 21.2, pp. 119-134, 2004. (Cited on page 21.)

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1-38, 1977. (Cited on pages 34, 35 and 52.)

[24] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. *Proceedings of the European Conference on Computer Vision*, pp. 184–199, 2014. (Cited on pages vi, viii, 25, 85, 88 and 89.)

[25] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016. (Cited on pages vi, 25, 84, 85 and 88.)

[26] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. *Proceedings of the European Conference on Computer Vision*, pp. 391–407, 2016. (Cited on page 25.)

[27] C. E. Duchon. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979. (Cited on page 24.)

[28] R. Fattal. Image Upsampling via Imposed Edge Statistics *ACM Transactions on Graphpics*, vol. 26, no. 3, pp. 1-8, Jul. 2007. (Cited on page 24.)

[29] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002. (Cited on pages 25 and 26.)

[30] A. Gastineau, J.-F. Aujol, Y. Berthoumieu, and C. Germain. A Residual Dense Generative Adversarial Network for Pansharpening with Geometrical Constraints. *IEEE International Conference on Image Processing (ICIP)*, 2020. (Cited on pages 95 and 97.)

[31] A Gastineau, J.-F Aujol, Y Berthoumieu, C Germain. Generative adversarial network for pansharpening with spectral and spatial discriminators. *IEEE Transactions on Geoscience and Remote Sensing*, 60, pp. 1-11, 2021. (Cited on page 98.)

[32] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 5(7):932–946, 1995. (Cited on pages 32 and 70.)

[33] M. El Gheche, J-F Aujol, Y. Berthoumieu, and C-A Deledalle. Texture reconstruction guided by the histogram of a high-resolution patch. *IEEE Transactions on Image Processing*, 26, 2017 (Cited on pages 2 and 10.)

[34] E. Gomez, M.A Gomez-Villegas and J.M. Marin. A survey on continuous Elliptical vector distributions. (Cited on page 40.)

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. (Cited on pages viii, 26, 90, 91 and 92.)

[36] T. Guillemot, and J. Delon. Implementation of the midway image equalization. *Image Processing On Line*, pp. 114-129, 2016. (Cited on page 21.)

[37] M. Hasannasab, J. Hertrich, F. Laus, and G. Steidl. Alternatives to the EM algorithm for ML estimation of location, scatter matrix, and degree of freedom of the student t distribution. *Numerical Algorithms*, 87(1):77–118, 2021. (Cited on page 75.)

[38] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision (ICCV)*, pp 1026–1034, 2015. (Cited on page 93.)

[39] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. (Cited on page 97.)

[40] J. Hertrich and G. Steidl. Inertial stochastic PALM (iSPALM) and applications in machine learning. *ArXiv preprint arXiv:2005.02204*, 2020. (Cited on page 75.)

[41] J. Hertrich, D-P-L. Nguyen, J-F. Aujol, D. Bernard, Y. Berthoumieu, A. Saadaldin, and G. Steidl. PCA Reduced Gaussian Mixture Models with Applications in Superresolution. Inverse Problems in Imaging, 16 (2) : 341-366. doi: 10.3934/ipi.2021053, 2022. (Cited on pages vi, 7, 14, 47, 54, 55, 57, 59, 61, 63, 65, 67, 70, 81, 84 and 103.)

[42] M. Hidane, M. El Gueche, J-F Aujol, Y. Berthoumieu, and C-A. Deledalle. Image zoom completion. *IEEE Transactions on Image Processing*, 25, 2016. (Cited on pages 2 and 10.)

[43] N. J. Higham. Matrix nearness problems and applications. In *Applications of matrix theory (Bradford, 1988)*, volume 22 of *The Institute of Mathematics and its Applications Conference Series. New Series.*, pages 1–27. Oxford University Press, New York, 1989. (Cited on pages 65 and 66.)

[44] N. J. Higham and R. S. Schreiber. Fast polar decomposition of an arbitrary matrix. *Society for Industrial and Applied Mathematics. Journal on Scientific and Statistical Computing*, 11(4):648–655, 1990. (Cited on page 66.)

[45] A. Houdard, C. Bouveyron, and J. Delon. High-dimensional mixture model for unsupervised image denoising (HDMI). *SIAM Journal on Imaging Sciences*, 11(4):2815-2846, 2018. (Cited on pages 57 and 59.)

[46] G. Huang, Z. Liu, L. van der Maaten, and K. Weinberer Densely connected convolutional networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700-4708, 2017. (Cited on page 97.)

[47] H. Hult and L. Filip. Multivariate extremes, aggregation and dependence in elliptical distributions. *Advances in Applied probability* 34.3 (2002): 587-608. (Cited on page 74.)

[48] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In Proceedings of The 32nd International Conference on Machine Learning (ICML)*, pp 448–456, 2015. (Cited on page 93.)

[49] T. Karras, S. Laine, T. Aila. A stylebased generator architecture for generative adversarial networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019. (Cited on page 105.)

[50] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981. (Cited on page 24.)

[51] D. P. Kingma and J. Ba. ADAM : A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 2015. (Cited on page 92.)

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems (NIPS)* 25, pp.1097-1105, 2012. (Cited on page 86.)

[53] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in Neural Information Processing Systems,* 22, pp. 1033–1041, 2009. (Cited on page 32.)

[54] F. Laus. Statistical Analysis and Optimal Transport for Euclidean and Manifold-Valued Data. *PhD Thesis, TU Kaiserslautern*, 2019. (Cited on page 59.)

[55] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. (Cited on page 86.)

[56] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang et al. Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition (CVPR)*, pp. 4681–4690, 2017. (Cited on pages vi, viii, 5, 13, 26, 84, 85, 92 and 93.)

[57] E. Lehmann and H. Scheffé. Completeness, similar regions, and unbiased estimation: Part I. *Sankhyā: the Indian Journal of Statistics*, pages 305–340, 1950. (Cited on page 40.)

[58] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 136–144, 2017. (Cited on page 84.)

[59] S. Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In *Les Équations aux Dérivées Partielles (Paris, 1962)*, pages 87–89. Éditions du Centre National de la Recherche Scientifique, Paris, 1963. (Cited on page 67.)

[60] S. Łojasiewicz. Sur la géométrie semi- et sous-analytique. *Université de Grenoble. Annales de l'Institut Fourier*, 43(5):1575–1595, 1993. (Cited on page 67.)

[61] D. G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, 2. pp. 1150–1157, 1999. (Cited on page 20.)

[62] A. Marquina and S. J. Osher. Image super-resolution by TV regularization and Bregman iteration. *Journal of Scientific Computing*, vol. 37, no. 3, pp. 367–382, 2008. (Cited on page 24.)

[63] G. J. McLachlan, D. Peel. Finite Mixture Models. *Wiley-Interscience*, New York, 2000. (Cited on page 35.)

[64] G. J. McLachlan, D. Peel, and R. Bean. Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis*, 41(3-4):379–388, 2003. (Cited on page 59.)

[65] S. Menon, A. Damian, S. Hu, N. Ravi, C. Rudin. PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2434-2442, 2020. (Cited on page 105.)

[66] X.-L. Meng and D. Van Dyk. The EM algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):511–567, 1997. (Cited on page 59.)

[67] F. Najar, S. Bourouis, N. Bouguila and S. Belghith. A Fixed-point Estimation Algorithm for Learning The Multivariate GGMM: Application to Human Action Recognition. *IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, 2018. (Cited on page 37.)

[68] K. Nasrollahi and T.B. Moeslund. Super-resolution: a comprehensive survey. *Machine Vision and Application*, 25:1423–1468, 2014. (Cited on pages 1 and 9.)

[69] D-P-L. Nguyen, J-F. Aujol and Y. Berthoumieu. Image Super Resolution with a Joint Generalized Gaussian Mixture Model. *XXVIIIe Colloque Francophone de Traitement du Signal et des Images,* 2022. (Cited on pages v, 6, 14, 29, 34, 35, 37, 39, 41, 43 and 103.)

[70] D-P-L. Nguyen, J. Hertrich, J-F. Aujol and Y. Berthoumieu. Image super-resolution with PCA Reduced generalized Gaussian mixture models. *HAL Preprint hal-03664839,* 2022. (Cited on pages vi, 7, 14, 47, 68, 69, 71, 73 and 103.)

[71] F. Pascal, L. Bombrun, J.-Y. Tourneret, and Y. Berthoumieu. Parameter estimation for multivariate generalized Gaussian distributions. *IEEE Transactions on Image Processing*, 61(23):5960-5971, 2013. (Cited on pages 2, 10, 34 and 37.)

[72] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559-572, 1901. (Cited on pages vi, 47 and 48.)

[73] K.B. Petersen, M.S. Pedersen. The Matrix Cookbook. *Technical University of Denmark*, Lecture Notes (2008). (Cited on pages 38 and 72.)

[74] T. Pock and S. Sabach. Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, 9(4):1756–1787, 2016. (Cited on pages 64 and 75.)

[75] P. Sandeep and T. Jacob. Single image super-resolution using a joint GMM method. *IEEE Transactions on Image Processing*, 25(9): 4233-4244, 2016. (Cited on pages v, 2, 4, 5, 6, 9, 10, 12, 13, 14, 29, 30, 31, 40, 43, 45, 67, 75, 76 and 80.)

[76] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video superresolution using an efficient sub-pixel convolutional neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016. (Cited on pages 26, 93 and 98.)

[77] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In International Conference on Learning Representations (ICLR)*, 2015. (Cited on page 94.)

[78] C. Sutour, C.-A. Deledalle, and J.-F. Aujol. Estimation of the noise level function based on a nonparametric detection of homogeneous image regions. *SIAM Journal on Imaging Sciences*, 8(4):2622–2661, 2015. (Cited on page 75.)

[79] M. E. Tipping , C. M. Bishop. Probabilistic Principal Component Analysis. *Journal of the royal statistical society*, Series B, Vol. 61, No. 3 (1999), 611-622. (Cited on pages vi, 47, 48, 49 and 50.)

[80] M. E. Tipping and C. M. Bishop. Mixture of probabilistic principal component analyzers. *Neural Computation*, 11(2):443-482, 1999. (Cited on pages vi, 47, 48, 51, 57 and 58.)

[81] S. Vaucher, P. Unifantowicz, C. Ricard, L. Dubois, M. Kuball, J.-M. Catala-Civera, D. Bernard, M. Stampanoni, and R. Nicula. On-line tools for microscopic and macroscopic monitoring of microwave processing. *Physica B: Condensed Matter*, 398(2):191–195, 2007. (Cited on page 18.)

[82] B. Wang, H. Zhang, Z. Zhao and Y. Sun. Globally Convergent Algorithms for Learning Multivariate Generalized Gaussian Distributions. *IEEE Statistical*

Signal Processing Workshop (SSP), 2021, pp. 336-340, doi: 10.1109/SSP 49050.2021.9513857. (Cited on pages 35, 37 and 39.)

[83] S. Wang, D. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 2216–2223, 2012. (Cited on page 25.)

[84] J. Yang, J. Wright, T. Huang, and Y. Ma. Image Super-Resolution Via Sparse Representation. *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861-2873, 2010. (Cited on pages 25 and 32.)

[85] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, Aug 2012. (Cited on page 25.)

[86] T. J. Ypma. Historical development of the Newton–Raphson method. *SIAM* Review 37 (4), 531–551, 1995. (Cited on page 39.)

[87] V. Zadorozhnyy, Q. Cheng, Q. Ye. Adaptive weighted discriminator for training generative adversarial networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4781-4790, 2021. (Cited on page 105.)

[88] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2018–2025, 2011. (Cited on page 25.)

[89] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. *Proceedings of the International Conference on Curves and Surfaces*, pp. 711–730, 2010. (Cited on page 25.)

[90] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2472–2481, 2018. (Cited on page 84.)

[91] J.-H. Zhao and L. Philip. Fast ML estimation for the mixture of factor analyzers via an ECM algorithm. *IEEE transactions on neural networks*, 19(11):1956–1961, 2008. (Cited on page 59.)

[92] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. *In Computer Vision (ICCV)*, 2011 IEEE International Conference on, pages 479–486, 2011. (Cited on pages v, 2, 4, 6, 10, 12, 14, 29, 30, 32, 33 and 43.)

[93] D. Zoran and Y. Weiss. Natural images, Gaussian mixtures and dead leaves. *Advances in Neural Information Processing Systems*, pages 1736–1744, 2012 (Cited on pages 2 and 10.)

Super-resolution of multiscale 3D images from material sciences

Abstract:

Recent developments in imaging techniques and computational analysis deeply modify the way materials sciences. The materials image has moved from microscale resolution to true nanoscale to analysis the defects and details at the interfaces of the materials. Thus, this thesis deals with the super-resolution (SR) problem in order to reconstruct the materials images in the high-resolution for instance the nanoscale resolution. To reach this goal, this thesis explores several SR approaches for materials images.

Single Image Super-Resolution (SISR) algorithms based on patch-based have been noticed and widely used over the past decade. Recently, Generalized Gaussian Mixture Models (GGMMs) have shown to be a suitable tool for many image processing problems due to the flexible shape parameter. In the first place, we propose using a joint GGMM learned from concatenated vectors of high and low resolution training patches to do super-resolution image based on the minimum mean square error (MMSE) method.

Unfortunately, the dimension of the concatenated samples is too high for the learning of the mixture models such as Gaussian mixture model and GGMM. Then we propose these two models Gaussian mixture model (GMM) and GGMM with a reduction of the dimensionality of the data in each component of the model by principal component analysis. These models are called to as PCA-GMM and PCA-GGMM, respectively. To learn the (low dimensional) parameters of the mixture model we propose two different EM algorithms whose M-step requires the solution of constrained optimization problems. Then we apply our PCA-GMM and PCA-GGMM for the super-resolution of 2D and 3D material images based on the MMSE method for the GGMM model.

In addition, the study of the material data becomes difficult because the HR and LR images have a different contrast. In our last contribution, we study a deep learning approach considering the problem of contrast change in material images. Indeed, we propose a generative adversarial network (GAN) within two generators, each responding to a different task. The first generator deals with the problem of contrast changes, and the second one focuses on the reconstruction of the high frequencies of the SR.

All the proposed methods lead to convincing results, both quantitative and visual. Especially the numerical results of the dimensionality reduction methods confirm a moderate influence of the size reduction on the overall SR results.

Keywords: Image super-resolution, materials science, Gaussian mixture model, generalized Gaussian mixture model, high-dimensional data, dimensionality reduction, deep learning, generative adversarial network (GAN).