PCA Reduced Gaussian Mixture Models with Applications in Superresolution

Johannes Hertrich, Lan Dang Phuong Nguyen, Jean-François Aujol, Dominique Bernard, Yannick Berthoumieu, Abdellatif Saadaldin, Gabriele Steidl

PCA Reduced Gaussian Mixture Models with Applications in Superresolution

Johannes Hertrich* Dang Phuong Lan Nguyen †‡
Jean-Francois Aujol † Dominique Bernard §
Yannick Berthoumieu ‡ Abdellatif Saadaldin § Gabriele Steidl*

September 16, 2020

Abstract

Despite the rapid development of computational hardware, the treatment of large and high dimensional data sets is still a challenging problem. This paper provides a twofold contribution to the topic. 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, called PCA-GMM. To learn the (low dimensional) parameters of the mixture model we propose an EM algorithm whose M-step requires the solution of constrained optimization problems. Fortunately, these constrained problems do not depend on the usually large number of samples and can be solved efficiently by an (inertial) proximal alternating linearized minimization algorithm. Second, we apply our PCA-GMM for the superresolution of 2D and 3D material images based on the approach of Sandeep and Jacob. Numerical results confirm the moderate influence of the dimensionality reduction on the overall superresolution result.

1 Introduction

The motivation for this work was superresolution of 3D material images taken within the project ITN MUMMERING (`https://www.mummering.eu`). Superresolution aims in reconstructing high resolution images from low resolution ones. Here a common assumption is that the low resolution image is generated by $y = Ax + \epsilon$, where ϵ is some noise

[1]TU Berlin, Straße des 17. Juni 136, D-10587 Berlin, Germany, {j.hertrich, steidl}@math.tu-berlin.de.
[2]Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 Talence, France, {lan.nguyen,jean-francois.aujol}@math.u-bordeaux.fr
[3]Univ. Bordeaux, Bordeaux INP, CNRS, IMS, UMR 5218, F-33400 Talence, France, yannick.berthoumieu@ims-bordeaux.fr
[4]CNRS, Univ. Bordeaux, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France, {dominique.bernard,abdellatif.saadaldin}@icmcb.cnrs.fr

and A is a possible unknown superresolution operator. Since this is an ill-posed inverse problem, methods addressing this task usually incorporate some prior information. One approach for solving the problem is based on Gaussian Mixture Models (GMMs). Usually, the GMM approximates the distribution of the patches of natural images and its parameters are learned from some given data, see also [6].

In literature, there exist several approaches to tackle superresolution by GMMs. Zoran and Weiss [25] proposed to use the negative log-likelihood function of a GMM as regularizer of the inverse problem by estimating the high resolution image x_H given the low resolution one x_L by solving

$$\operatorname*{argmin}_x \|Ax_H - x_L\| - \lambda \sum_{i \in I} \log p(x_{H,i}),$$

where p is the probability density function of the GMM and $(x_{H,i})_{i \in I}$ are the patches in x_H. For a special choice of λ, the solution of this problem can be interpreted as the maximum a postiori (MAP) estimator of x_H under the prior assumption that the distribution of the patches in x_H are given by the GMM. This method is called expected patch log likelihood (EPLL) and it can be applied for several inverse problems. Various accelerations and an efficient implementation of EPLL were developed in [19]. However, EPLL requires that the operator A is known, which is usually not the case for the superresolution task. Therefore, we prefer the alternative approach of Sandeep and Jacob [22], which does not require any knowledge about the operator A. While for EPLL the GMM describes only the distribution of the patches from the high resolution image, here the idea is to use a joint GMM, which describes the distribution of pairs of high and low resolution patches. Having learned a joint GMM, each high resolution patch is estimated separately from the low resolution patch and the joint GMM using the minimal mean squared error estimator.

However, any of these models requires the estimation of the parameters of a GMM using the patches of the images as data points. For this, the maximum likelihood (ML) estimator is used, which corresponds to minimizing the negative log likelihood function. The standard method, to find the ML estimator is the expectation maximization (EM) algorithm [3, 7]. Unfortunately, the EM algorithm for GMMs becomes very slow, as the number of data points becomes large and high dimensional, which is the case for our superresolution task, particularly if we have to deal with 3D images.

To overcome these performance issues, we reduce the dimension of the data points. The standard method for dimensionality reduction is the principle component analysis (PCA) [20]. The main assumption of the PCA is that the high dimensional data points are approximately located in a lower dimensional affine subspace. In this paper, we combine the GMM with a PCA by adding the minimization term of the PCA and the negative log likelihood function of the GMM on the dimensionality reduced data points. We rewrite this minimization problem again as the negative log likelihood function of a Gaussian mixture model which has additional constraints on the parameters. We call this model PCA-GMM. This representation allows in particular, to use a different PCA for each component of the GMM. We derive an EM algorithm with a special M-step for finding

a minimizer of our objective function. The M-step requires solutions of maximization problems with contraints on the Stiefel manifold. Fortunately, these problems do no longer depend on the large number of sampling points and they can be efficiently solved by the (inertial) proximal alternating linearized minimization algorithm (PALM) for which some convergence results can be ensured. The idea to couple parameter learning with dimension reduction is not new. So the authors of [12] propose directly a GMM with constraint covariance matrices. It is based on an extension of the PCA, which was proposed in [23] to replace the affine space in the PCA by the union of finitely many affine spaces using a mixture model of probabilistic PCAs. The relation to our approach is analysed in a remark in Section 3. Using our new PCA-GMM model within the superresolution model of Sandep and Jacob [22], we provide numerical examples of 2D and 3D material images.

The paper is organized as follows: in Section 2 we briefly review the two main ingredients for our model, namely GMMs and PCA. We derive our PCA-GMM model in Section 3. In Section 4, an EM algorithm with a special constrained optimization task in the M-step is proposed for minimizing the objective function. The solution of the constrained minimization problem via (inertial) PALM is investigated. The superresolution method using the PCA-GMM model is described in Section 5. Finally, Section 6 shows numerical examples of superresolution based on our PCA-GMM model on 2D and 3D images. Conclusions are drawn in Section 7.

2 Preliminaries

In this section, we briefly revisit the two building blocks of our approach, namely Gaussian mixture models and principal component analysis. We need the following notation. By $\mathrm{SPD}(n) \subset \mathbb{R}^{n,n}$ we denote the cone of *symmetric positive definite* $n \times n$ *matrices*, by $O(n)$ the group of orthogonal $n \times n$ matrices, by

$$\mathrm{St}(d,n) \coloneqq \left\{U \in \mathbb{R}^{n,d} : U^{\mathrm{T}}U = I_d\right\}, \quad n \geq d,$$

the *Stiefel manifold* and by $\Delta_K \coloneqq \{\alpha = (\alpha_k)_{k=1}^K \in \mathbb{R}^K_{\geq 0} : \sum_{k=1}^K \alpha_k = 1\}$ the *probability simplex*. We write 1_n for the vector with all n components equal to 1. Further, we denote by $\|\cdot\|_F$ the Frobenius norm.

Gaussian Mixture Models 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

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

Note, that not all multivariate normal distributions are absolutely continuous, in particular, the covariance matrix must not be invertible. A *Gaussian mixture model* (GMM)

is a random variable with probability density function

$$p(x) = \sum_{k=1}^{K} \alpha_k f(x|\mu_k, \Sigma_k), \qquad \alpha \in \Delta_K.$$

For samples $\mathcal{X} = \{x_1, ..., x_N\}$, the maximum likelihood (ML) estimator of the parameters $\alpha = (\alpha_k)_{k=1}^K$, $\mu = (\mu_k)_{k=1}^K$ and $\mathbf{\Sigma} = (\Sigma_k)_{k=1}^K$ of a GMM can be found by minimizing its *negative log-likelihood function*

$$L(\alpha, \mu, \mathbf{\Sigma}|\mathcal{X}) = -\sum_{i=1}^{N} \log\Big(\sum_{k=1}^{K} \alpha_k f(x_i|\mu_k, \Sigma_k)\Big)$$

for $\alpha \in \triangle_K$, $\mu_k \in \mathbb{R}^n$, and $\Sigma_k \in \mathrm{SPD}(n)$, $k = 1, \ldots, K$.

In the following, we use the notation $\vartheta \coloneqq (\mu, \Sigma)$ to address the parameters of a Gaussian distribution. A standard minimization algorithm for finding the ML estimator of the parameters α_k and $\vartheta_k = (\mu_k, \Sigma_k)$, $k = 1, \ldots, K$ of a GMM is the so-called *EM algorithm* [3, 7] detailed in Alg. 1.

Algorithm 1 EM Algorithm for Mixture Models

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

$$\beta_{i,k}^{(r)} = \frac{\alpha_k^{(r)} f(x_i|\vartheta_k^{(r)})}{\sum_{j=1}^{K} \alpha_j^{(r)} f(x_i|\vartheta_j^{(r)})}$$

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

$$\alpha_k^{(r+1)} = \frac{1}{N} \sum_{i=1}^{N} \beta_{i,k}^{(r)},$$
$$\vartheta_k^{(r+1)} = \operatorname*{argmax}_{\vartheta_k} \Big\{ \sum_{i=1}^{N} \beta_{i,k}^{(r)} \log(f(x_i|\vartheta_k)) \Big\}.$$

end for

For Gaussian density functions (1), the iterates $\vartheta_k^{(r+1)}$, $k = 1, \ldots, K$, i.e. the maximization in the M-Step of Alg. 1 can be simply computed by

$$\mu_k^{(r+1)} = \frac{1}{N\alpha_k^{(r+1)}} m^{(r)} \quad \text{and} \quad \Sigma_k^{(r+1)} = \frac{1}{N\alpha_k^{(r+1)}} C^{(r)}, \tag{2}$$

where

$$m^{(r)} = \sum_{i=1}^{N} \beta_{ik}^{(r)} x_i \quad \text{and} \quad C^{(r)} = \sum_{i=1}^{N} \beta_{ik}^{(r)} x_i x_i^{\mathrm{T}}.$$

Principal Component Analysis In many applications, the dimension of the data is huge such that dimensionality reduction methods become necessary. The working horse for dimensionality reduction is the principal component analysis (PCA). Given data samples $\mathcal{X} = \{x_1, ..., x_N\}$ in $\mathbb{R}^n$, the classical PCA finds the 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

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

for $b \in \mathbb{R}^n$ and $U \in \mathrm{St}(d, n)$. It is not hard to check that the affine subspace goes through 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., consider

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

3 PCA-GMM Model

In this section, we propose a GMM which incorporates a dimensionality reduction model via PCA. More precisely, we want to consider Gaussian distributions only on smaller subspaces of the original data space.

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

$$F(U, \alpha, \vartheta) = L\big(\alpha, \vartheta | \mathcal{X}_{\mathrm{low}}\big) + \frac{1}{2\sigma^2} P\,(U)\,, \quad \sigma > 0 \tag{3}$$

for $U \in \mathrm{St}(d, n)$, $\alpha \in \triangle_K$, $\mu_k \in \mathbb{R}^d$, and $\Sigma_k \in \mathrm{SPD}(d)$, $k = 1, \ldots, K$, where

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

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

$$\begin{aligned} F(U, \alpha, \vartheta) &= -\sum_{i=1}^{N} \left(\log\Big(\sum_{k=1}^{K} \alpha_k f\big(U^{\mathrm{T}} y_i | \vartheta_k\big) \Big) - \frac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I_n)y_i)\|^2 \right) \\ &= -\sum_{i=1}^{N} \left(\log\Big(\sum_{k=1}^{K} \alpha_k f\big(U^{\mathrm{T}} y_i | \vartheta_k\big)\, \exp\big(-\tfrac{1}{2\sigma^2} \|(UU^{\mathrm{T}} - I_n)y_i)\|^2 \big) \Big) \right). \end{aligned} \tag{4}$$

However, knowing that the samples were taken from K different Gaussian distributions it makes more sense to reduce the dimension according to the respective distribution. Based on the reformulation (4) 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**:

$$F(\mathbf{U}, \mathbf{b}, \alpha, \vartheta) \quad \text{subject to} \quad \alpha \in \Delta_K,\, U_k \in \mathrm{St}(d, n), \Sigma_k \in \mathrm{SPD}(d),\; k = 1, \ldots, K, \tag{5}$$

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

$$F(\mathbf{U}, \mathbf{b}, \alpha, \vartheta) \coloneqq -\sum_{i=1}^{N} \log\Big(\sum_{k=1}^{K} \alpha_k f(U_k^{\mathrm{T}} y_{ik} | \vartheta_k) \exp\big(-\tfrac{1}{2\sigma^2} \|(I_n - U_k U_k^{\mathrm{T}}) y_{ik}\|^2 \big) \Big), \quad (6)$$
$$y_{ik} \coloneqq x_i - b_k, \quad k = 1, \dots, K,\ i = 1, \dots, N.$$

Clearly, if $U_k = U$ and $b_k = \bar{x}$ for all $k = 1, \dots, K$, we get back to model (3).

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

Lemma 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* (1)*. Then the following relation holds true:*

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

where

$$\tilde{\Sigma} = \big(\tfrac{1}{\sigma^2}(I_n - UU^{\mathrm{T}}) + U\Sigma^{-1}U^{\mathrm{T}} \big)^{-1} \in \mathrm{SPD}(n), \quad (7)$$
$$\tilde{\mu} = \tilde{\Sigma} U \Sigma^{-1} \mu + b \in \mathbb{R}^n. \quad (8)$$

Proof. 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

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

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

$$V^{\mathrm{T}} \tilde{\Sigma}^{-1} V = \tfrac{1}{\sigma^2} \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{n-d} \end{array} \right) + \left(\begin{array}{c|c} \Sigma^{-1} & 0 \\ \hline 0 & 0 \end{array} \right) = \left(\begin{array}{c|c} \Sigma^{-1} & 0 \\ \hline 0 & \frac{1}{\sigma^2} I_{n-d} \end{array} \right) \quad (9)$$

and the last matrix is invertible.

2. We have to show that

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

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

$$\begin{aligned}
&\frac{1}{2\sigma^2}\|(I_n - UU^{\mathrm{T}})(x-b)\|^2 + \frac{1}{2}(U^{\mathrm{T}}(x-b)-\mu)^{\mathrm{T}}\Sigma^{-1}(U^{\mathrm{T}}(x-b)-\mu)\\
&= \frac{1}{2}x^{\mathrm{T}}\left(\tfrac{1}{\sigma^2}(I_n - UU^{\mathrm{T}}) + U\Sigma^{-1}U^{\mathrm{T}}\right)x - \left(\tfrac{1}{\sigma^2}b^{\mathrm{T}}(I_n - UU^{\mathrm{T}}) + (\mu^{\mathrm{T}} + b^{\mathrm{T}}U)\Sigma^{-1}U^{\mathrm{T}}\right)x\\
&\quad + \frac{1}{2}(U^{\mathrm{T}}b+\mu)^{\mathrm{T}}\Sigma^{-1}(U^{\mathrm{T}}b+\mu) + \tfrac{1}{2\sigma^2}b^{\mathrm{T}}(I_n - UU^{\mathrm{T}})b\\
&= \frac{1}{2}x^{\mathrm{T}}\tilde{\Sigma}^{-1}x - \tilde{\mu}^{\mathrm{T}}\tilde{\Sigma}^{-1}x + \frac{1}{2}\tilde{\mu}^{\mathrm{T}}\tilde{\Sigma}^{-1}\tilde{\mu}.
\end{aligned}$$

Finally, we see by (9) that $|\tilde{\Sigma}|^{-1} = \sigma^{-2(n-d)}|\Sigma|^{-1}$. □

By Lemma 3.1, we can rewrite our objective function F in (6) with $\tilde{\vartheta} = (\tilde{\mu}, \tilde{\Sigma})$ defined by (7) and (8) with corresponding indices as

$$\begin{aligned}
F(\mathbf{U}, \mathbf{b}, \alpha, \vartheta) &= -\sum_{i=1}^{N}\log\Big(\sum_{k=1}^{K}\alpha_k f(x_i|\tilde{\vartheta}_k)\Big) + (n-d)\log\sqrt{2\pi\sigma^2}\\
&= L(\alpha, \tilde{\vartheta}|\mathcal{X}) + (n-d)\log\sqrt{2\pi\sigma^2}.
\end{aligned}$$

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 (7) and (8) into account. More precisely, our model in (5) can be rewritten as **PCA-GMM model**:

$$\mathrm{F}(\mathbf{U}, \mathbf{b}, \alpha, \vartheta) \coloneqq L(\alpha, \tilde{\vartheta}|\mathcal{X}) \quad \text{subject to} \quad U_k \in \mathrm{St}(d,n),\ \alpha \in \Delta_K,\ \Sigma_k \in \mathrm{SPD}(d), \tag{10}$$

where

$$\tilde{\Sigma}_k = \left(\tfrac{1}{\sigma^2}(I_n - U_kU_k^{\mathrm{T}}) + U_k\Sigma_k^{-1}U_k^{\mathrm{T}}\right)^{-1},\ \tilde{\mu}_k = \tilde{\Sigma}_kU\Sigma_k^{-1}\mu_k + b_k, \quad k = 1, \dots, K. \tag{11}$$

The following remark shows the relation of our approach to a model proposed by Houdard, Bouveryron and Delon [12].

Remark 3.2. First, we note that we have as in (9) for matrices $\tilde{\Sigma}$ of the form (7) and an orthogonal matrix $V = (U|\tilde{U})$ that

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

so that

$$\begin{aligned}
&\left\{\tilde{\Sigma} = \left(\tfrac{1}{\sigma^2}(I_n - UU^{\mathrm{T}}) + U\Sigma^{-1}U^{\mathrm{T}}\right)^{-1} : U \in \mathrm{St}(d,n),\ \Sigma \in \mathrm{SPD}(d)\right\}\\
&\quad = \left\{Q^{\mathrm{T}}\left(\begin{array}{c|c} \mathrm{diag}(\lambda) & 0 \\ \hline 0 & \sigma^2 I_{n-d} \end{array}\right)Q : Q \in O(n),\ \lambda \in \mathbb{R}^d_{>0}\right\}.
\end{aligned} \tag{12}$$

The authors of [12] propose for $U_k \in \mathrm{St}(d,n)$ and $\lambda_k \in \mathbb{R}^d_{>0}$, $k = 1,\dots,K$ to minimize the functional

$$\tilde{\mathrm{F}}(\mathbf{U}, \alpha, \tilde{\mu}, \lambda) = -\sum_{i=1}^{N} \log\Big(\sum_{k=1}^{K} \alpha_k f(x_i|\tilde{\mu}_k, \tilde{\Sigma}_k)\Big),$$

where

$$\tilde{\Sigma}_k := U_k \operatorname{diag}(\lambda_k) U_k^{\mathrm{T}} + \sigma^2 I_n.$$

Hence, skipping the index, instead of minimizing over (12), they minimize over sets of the form

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

However, the authors algorithm only ensures that $\lambda > -\sigma^2 1_d$, so that the corresponding algorithm appears to be no longer an EM algorithm.

□

4 Minimization Algorithm

We propose to minimize F in (10) 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 (11) 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 4.1. In particular, we will see that the M-Step of the algorithm requires the minimization of functions $G_k(U,b)$, $k = 1,\dots,K$ 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, $i = 1,\dots,N$. 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 Subsection 4.2 and provide convergence results.

4.1 EM Algorithm for PCA-GMM

For our setting, we obtain a special EM algorithm described in Algorithm 2. Note that E-Step of Algorithm 2 requires only the mean and covariance matrix in $\vartheta_k^{(r)}$, $k = 1,\dots,K$ 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 [4, 5], see also [14] for a nice review. The authors showed that the objective function decreases for the iterates of the algorithm. Hence we obtain the following corollary.

Corollary 4.1. *For the iterates* $\left(\mathbf{U}^{(r)}, \mathbf{b}^{(r)}, \alpha^{(r)}, \vartheta^{(r)}\right)_r$ *generated by Algorithm 2 the objective function* F *is decreasing.*

Algorithm 2 EM Algorithm for PCA reduced Mixture Models

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

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

$$\begin{aligned}\beta_{i,k}^{(r)} &= \frac{\alpha_k^{(r)} f(x_i|\tilde{\vartheta}_k^{(r)})}{\sum_{j=1}^K \alpha_j^{(r)} f(x_i|\tilde{\vartheta}_j^{(r)})} \\ &= \frac{\alpha_k^{(r)} \exp\left(-\frac{1}{2\sigma^2}\|(I_n - U_k^{(r)}(U_k^{(r)})^{\mathrm{T}})y_{i,k}^{(r)}\|^2\right) f\left((U_k^{(r)})^{\mathrm{T}} y_{i,k}^{(r)}|\vartheta_k^{(r)}\right)}{\sum_{j=1}^K \alpha_j^{(r)} \exp\left(-\frac{1}{2\sigma^2}\|(I_n - U_j^{(r)}(U_j^{(r)})^{\mathrm{T}})y_{i,k}^{(r)}\|^2\right) f\left((U_j^{(r)})^{\mathrm{T}} y_{i,k}^{(r)}|\vartheta_j^{(r)}\right)}, \\ y_{i,k}^{(r)} &= x_i - b_k^{(r)}.\end{aligned}$$

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

$$\alpha_k^{(r+1)} = \frac{1}{N}\sum_{i=1}^N \beta_{i,k}^{(r)},$$

$$\begin{aligned}(U_k^{(r+1)}, b_k^{(r+1)}, \vartheta_k^{(r+1)}) = \underset{U,b,\mu,\Sigma}{\operatorname{argmax}} &\sum_{i=1}^N \beta_{ik}^{(r)} \log(f(x_i|\tilde{\vartheta}_k)) \\ &\text{subject to} \quad U_k \in \mathrm{St}(d,n), \Sigma_k \in \mathrm{SPD}(d) \\ &\text{with } \tilde{\vartheta}_k = (\tilde{\mu}_k, \tilde{\Sigma}_k) \text{ as in (11).}\end{aligned}$$

end for

The interesting step is the second M-Step which requires again the maximization of a function. Based on (2) we can prove the following proposition.

Proposition 4.2. *Let f be the Gaussian density function* (1) *and $\beta_i \in \mathbb{R}_{\geq 0}$, $i = 1, \dots, N$. Then a solution of*

$$\underset{U,b,\mu,\Sigma}{\operatorname{argmax}} \sum_{i=1}^N \beta_i \log(f(x_i|\tilde{\vartheta})) \tag{13}$$

with $\tilde{\vartheta} = (\tilde{\mu}, \tilde{\Sigma})$ of the form (8) *and* (7) *is given by*

$$\hat{\mu} = \frac{1}{\alpha}\hat{U}^{\mathrm{T}}\hat{d}, \quad \textit{and} \quad \hat{\Sigma} = \frac{1}{\alpha}\hat{U}^{\mathrm{T}}\hat{S}\hat{U},$$

where

$$\hat{d} = m - \alpha\hat{b}, \quad \hat{S} = C - m\hat{b}^{\mathrm{T}} - \hat{b}m^{\mathrm{T}} + \alpha\hat{b}\hat{b}^{\mathrm{T}},$$

$$m = \sum_{i=1}^N \beta_i x_i, \quad C = \sum_{i=1}^N \beta_i x_i x_i^{\mathrm{T}}, \quad \alpha = \sum_{i=1}^N \beta_i$$

and

$$(\hat{U}, \hat{b}) \in \operatorname*{argmin}_{U \in \mathrm{St}(d,n), b \in \mathbb{R}^n} G(U, b). \tag{14}$$

Here

$$G(U,b) := -\tfrac{1}{\sigma^2}\left(\mathrm{tr}(U^{\mathrm{T}}SU) - \tfrac{1}{\alpha}\|d\|^2\right) - d^{\mathrm{T}}U(U^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}d + \alpha\log(|U^{\mathrm{T}}SU|) \tag{15}$$

and $d = m - \alpha b$, $S = C - mb^{\mathrm{T}} - bm^{\mathrm{T}} + \alpha bb^{\mathrm{T}}$.

Note that α in the proposition is defined in another way than in the first M-step, more precisely, the factor $\frac{1}{N}$ is skipped.

Proof. For fixed U and b, we have as in the classical GMM, see (2), that the maximizer in (13) with respect to μ and Σ fulfills

$$\mu = \frac{1}{\alpha}\sum_{i=1}^{N}\beta_i U^{\mathrm{T}}(x_i - b) = \frac{1}{\alpha}(U^{\mathrm{T}}m - \alpha U^{\mathrm{T}}b),$$
$$\Sigma = \frac{1}{\alpha}\sum_{i=1}^{N}\beta_i U^{\mathrm{T}}(x_i - b)\left(U^{\mathrm{T}}(x_i - b)\right)^{\mathrm{T}} = \frac{1}{\alpha}U^{\mathrm{T}}SU.$$

By Lemma 3.1, the negative objective function in (13) is given by

$$2\tilde{G}(U,b) = G_1(U,b) + G_2(U,b) + \alpha\log(|\Sigma|) + \mathrm{const}, \tag{16}$$
$$G_1(U,b) = \sum_{i=1}^{N}\beta_i\tfrac{1}{\sigma^2}(x_i - b)^{\mathrm{T}}(I_n - UU^{\mathrm{T}})(x_i - b) \tag{17}$$
$$G_2(U,b) = \sum_{i=1}^{N}\beta_i\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b + \mu)\right)^{\mathrm{T}}\Sigma^{-1}\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b + \mu)\right). \tag{18}$$

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

$$G_1(U,b) = -\tfrac{1}{\sigma^2}\sum_{i=1}^{N}\beta_i\mathrm{tr}(U^{\mathrm{T}}x_i x_i^{\mathrm{T}}U) - \tfrac{1}{\sigma^2}2b^{\mathrm{T}}(I_n - UU^{\mathrm{T}})m + \tfrac{1}{\sigma^2}\alpha b^{\mathrm{T}}(I_n - UU^{\mathrm{T}})b + \mathrm{const}$$

$$= -\tfrac{1}{\sigma^2}\left(\mathrm{tr}(U^{\mathrm{T}}CU) + 2b^{\mathrm{T}}(I_n - UU^{\mathrm{T}})m - \alpha b^{\mathrm{T}}(I_n - UU^{\mathrm{T}})b\right) + \mathrm{const}$$
$$= -\tfrac{1}{\sigma^2}\left(\mathrm{tr}(U^{\mathrm{T}}SU) - \frac{1}{\alpha}\|d\|^2\right) + \mathrm{const}. \tag{19}$$

For the second function we get

$$\begin{aligned}
G_2(U,b) &= \sum_{i=1}^{N} \beta_i \mathrm{tr}\left(\Sigma^{-1}\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b+\mu)\right)\left(U^{\mathrm{T}}x_i - (U^{\mathrm{T}}b+\mu)\right)^{\mathrm{T}}\right) \\
&= \sum_{i=1}^{N} \beta_i \Big(\mathrm{tr}(\Sigma^{-1}U^{\mathrm{T}}x_i x_i^{\mathrm{T}}U) - \mathrm{tr}\left(\Sigma^{-1}(U^{\mathrm{T}}b+\mu)x_i^{\mathrm{T}}U\right) \\
&\quad -\mathrm{tr}\left(\Sigma^{-1}U^{\mathrm{T}}x_i(U^{\mathrm{T}}b+\mu)^{\mathrm{T}}\right) + \alpha\mathrm{tr}\left(\Sigma^{-1}(U^{\mathrm{T}}b+\mu)(U^{\mathrm{T}}b+\mu)^{\mathrm{T}}\right)\Big) \\
&= \mathrm{tr}(\Sigma^{-1}U^{\mathrm{T}}CU) - \mathrm{tr}\left(\Sigma^{-1}(U^{\mathrm{T}}b+\mu)m^{\mathrm{T}}U\right) \\
&\quad - \mathrm{tr}\left(\Sigma^{-1}U^{\mathrm{T}}m(U^{\mathrm{T}}b+\mu)^{\mathrm{T}}\right) + \alpha\mathrm{tr}\left(\Sigma^{-1}(U^{\mathrm{T}}b+\mu)(U^{\mathrm{T}}b+\mu)^{\mathrm{T}}\right).
\end{aligned}$$

Further, with $U^{\mathrm{T}}b+\mu = U^{\mathrm{T}}m/\alpha$ and $\Sigma = U^{\mathrm{T}}SU/\alpha$, we conclude

$$\begin{aligned}
G_2(U,b) &= \mathrm{tr}(\Sigma^{-1}U^{\mathrm{T}}CU) - \tfrac{1}{\alpha}\mathrm{tr}(\Sigma^{-1}U^{\mathrm{T}}mm^{\mathrm{T}}U) \\
&= \mathrm{tr}\big(\Sigma^{-1}U^{\mathrm{T}}SU\big) - \mathrm{tr}\big(\Sigma^{-1}U^{\mathrm{T}}(-mb^{\mathrm{T}} - bm^{\mathrm{T}} + \alpha bb^{\mathrm{T}})U\big) \\
&\quad - \tfrac{1}{\alpha}\mathrm{tr}(\Sigma^{-1}U^{\mathrm{T}}mm^{\mathrm{T}}U) \\
&= 2\alpha m^{\mathrm{T}}U(U^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}b - \alpha^2 b^{\mathrm{T}}U(U^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}b \\
&\quad - m^{\mathrm{T}}U(U^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}m + \mathrm{const} \\
&= -(m-\alpha b)^{\mathrm{T}}U(U^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}(m-\alpha b) + \mathrm{const} \\
&= -d^{\mathrm{T}}U(U^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}d + \mathrm{const} \qquad (20)
\end{aligned}$$

Combining (16), (19) and (20) we obtain the assertion. □

Remark 4.3. By definition of m and C we have for the matrix S in Proposition 4.2 that

$$\begin{aligned}
S &= C - mb^{\mathrm{T}} - bm^{\mathrm{T}} + \alpha bb^{\mathrm{T}} \\
&= \sum_{i=1}^{N} \beta_i (x_i x_i^{\mathrm{T}} - bx_i^{\mathrm{T}} - x_i b^{\mathrm{T}} + bb^{\mathrm{T}}) = \sum_{i=1}^{N} \beta_i (x_i - b)(x_i - b)^{\mathrm{T}}.
\end{aligned}$$

In particular, S is symmetric positive definite, if n of the vectors $x_i - b$, $i = 1, ..., N$ are linearly independent. In the case, that $n+1$ of the points x_i, $i = 1, ..., N$ are affinely independent, we have that S is symmetric positive definite for any $b \in \mathbb{R}^d$.

Lemma 4.4. *The function $G(U,b)$ in (15) has a global minimizer.*

Proof. First note that G_1 and G_2 defined in (16) are non negative. This yields that $g \coloneqq \inf_{U\in \mathrm{St}(d,n), b\in\mathbb{R}^n} G(U,b) > -\infty$. Now, let $(U^{(r)}, b^{(r)})_r$ be a minimizing sequence of G, i.e., $\lim_{r\to\infty} G(U^{(r)}, b^{(r)}) = g$. In the following, we show that the sequence $(U^{(r)}, b^{(r)})_r$ is bounded. Since the Stiefel manifold is compact, it is sufficient to show, that the sequence $b^{(r)}$ is bounded. Since $\mathrm{Ker}(U^{(r)})^{\mathrm{T}} \oplus \mathrm{Im}\, U^{(r)} = \mathbb{R}^n$, we have that $b^{(r)}$ can be

uniquely represented as $b^{(r)} = b_1^{(r)} + b_2^{(r)}$, where $b_1^{(r)} \in \mathrm{Ker}(U^{(r)})^{\mathrm{T}}$ and $b_2^{(r)} \in \mathrm{Im}\, U^{(r)}$.

Now, assume that $b^{(r)}$ is unbounded. By going over to a subsequence, we can assume that $\|b^{(r)}\|_2 \to \infty$ as $r \to \infty$. Up to a subsequence again, this implies that $\|b_1^{(r)}\|_2 \to \infty$ or $\|b_2^{(r)}\|_2 \to \infty$ as $r \to \infty$.

1. Assume that $b_1^{(r)} \to \infty$. Then it holds

$$\sigma^2 G_1(U^{(r)}, b^{(r)}) \geq \sum_{i=1}^{N} \beta_i \left((b^{(r)})^{\mathrm{T}} (I_n - U^{(r)}(U^{(r)})^{\mathrm{T}}) b^{(r)} - 2x_i^{\mathrm{T}} (I_n - U^{(r)}(U^{(r)})^{\mathrm{T}}) b^{(r)} \right) \tag{21}$$

   Now, since $b_1^{(r)} \in \mathrm{Ker}(U^{(r)})^T$ and $b_2 \in \mathrm{Im}\, U^{(r)}$ it holds

$$(I_n - U^{(r)}(U^{(r)})^{\mathrm{T}}) b^{(r)} = (I_n - U^{(r)}(U^{(r)})^{\mathrm{T}})(b_1^{(r)} + b_2^{(r)}) = (b_1^{(r)} + b_2^{(r)}) - b_2^{(r)} = b_1^{(r)}.$$

   Inserting this in (21) and using $(I_n - U^{(r)}(U^{(r)})^{\mathrm{T}}) = (I_n - U^{(r)}(U^{(r)})^{\mathrm{T}})^2$ yields

$$\begin{aligned} \sigma^2 G_1(U^{(r)}, b^{(r)}) &\geq \sum_{i=1}^{N} \beta_i \left((b_1^{(r)})^{\mathrm{T}} b_1^{(r)} - 2x_i^{\mathrm{T}} b_1^{(r)} \right) \\ &\geq \sum_{i=1}^{N} \beta_i \left(\|b_1^{(r)}\|_2^2 - 2\|x_i\|_2 \|b_1^{(r)}\|_2 \right) \to \infty \quad \text{as } r \to \infty. \end{aligned}$$

   Since G_2 is bounded from below, this contradicts the assumption that $(U^{(r)}, b^{(r)})_r$ is a minimizing sequence of G.

2. Assume that $b_2^{(r)} \to \infty$. Then we have for the smallest eigenvalue ϵ of Σ^{-1} that

$$\begin{aligned} G_2(U^{(r)}, b^{(r)}) &\geq \sum_{i=1}^{N} \beta_i \Big(((U^{(r)})^{\mathrm{T}} b^{(r)})^{\mathrm{T}} \Sigma^{-1} (U^{(r)})^{\mathrm{T}} b^{(r)} \\ &\quad - 2((U^{(r)})^{\mathrm{T}} x_i - \mu)^{\mathrm{T}} \Sigma^{-1} (U^{(r)})^{\mathrm{T}} b^{(r)} \Big) \\ &\geq \sum_{i=1}^{N} \beta_i \Big(\epsilon \|(U^{(r)})^{\mathrm{T}} b^{(r)}\|_2^2 \\ &\quad - 2\|\Sigma^{-1}((U^{(r)})^{\mathrm{T}} x_i - \mu)\|_2 \|(U^{(r)})^{\mathrm{T}} b^{(r)}\|_2 \Big). \end{aligned} \tag{22}$$

   Now the construction of $b_1^{(r)}$ and $b_2^{(r)}$ yields $\|(U^{(r)})^{\mathrm{T}} b^{(r)}\| = \|(U^{(r)})^{\mathrm{T}} b_2^{(r)}\| = \|b_2^{(r)}\|_2$. Inserting this in (22) yields

$$G_2(U^{(r)}, b^{(r)}) \geq \sum_{i=1}^{N} \beta_i \Big(\epsilon \|b_2^{(r)}\|_2^2 - 2\|\Sigma^{-1}((U^{(r)})^{\mathrm{T}} x_i - \mu)\|_2 \|b_2^{(r)}\|_2 \Big) \to \infty$$

   as $r \to \infty$. Since G_1 is bounded from below, this contradicts the assumption that $(U^{(r)}, b^{(r)})_r$ is a minimizing sequence of G.

Thus, the minimizing sequence $(U^{(r)}, b^{(r)})_r$ is bounded. Therefore it has a cluster point, which is a minimizer. □

By Proposition 4.2, the M-Step of Algorithm 2 reduces for $k = 1, ..., K$ to the computation

$$\begin{aligned}
\alpha_k^{(r+1)} &= \frac{1}{N}\sum_{i=1}^{N}\beta_{i,k}^{(r)},\\
m_k &= \sum_{i=1}^{N}\beta_{i,k}x_i,\ C_k = \sum_{i=1}^{N}\beta_{i,k}x_i x_i^{\mathrm{T}},\\
(U_k^{(r+1)}, b_k^{(r+1)}) &= \operatorname*{argmin}_{U\in\mathrm{SPD}(d,n),b\in\mathbb{R}^n} G_k(U,b) \qquad \text{with } G_k \text{ in (15)},\\
\mu_k^{(r+1)} &= \frac{1}{N\alpha_k^{(r+1)}}\big(U_k^{(r+1)}\big)^{\mathrm{T}}\left(m_k - N\alpha_k^{(r+1)}b_k^{(r+1)}\right),\\
S_k &= C_k - m_k\left(b_k^{(r+1)}\right)^{\mathrm{T}} - b_k^{(r+1)}m_k^{\mathrm{T}} + N\alpha_k^{(r+1)}b_k^{(r+1)}\left(b_k^{(r+1)}\right)^{\mathrm{T}}\\
\Sigma_k^{(r+1)} &= \frac{1}{N\alpha_k^{(r+1)}}\big(U_k^{(r+1)}\big)^{\mathrm{T}} S_k\, U_k^{(r+1)}
\end{aligned}$$

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.

4.2 PALM for Minimizing G

To minimize G in (15) we propose to use the Proximal alternating linearized minimization (PALM) [2], resp. its accelerated version iPALM [21], where the 'i' stands for inertial. The PALM algorithm can be applied to functions of the form

$$F(x_1, x_2) = H(x_1, x_2) + f(x_1) + g(x_2) \tag{23}$$

where $H \in C^1(\mathbb{R}^{d_1} \times \mathbb{R}^{d_2})$ and lower semi-continuous functions $f : \mathbb{R}^{d_1} \to (-\infty, \infty]$ and $g : \mathbb{R}^{d_2} \to (-\infty, \infty]$. It is based on the computation of 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

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

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

Starting with an arbitrary $x_1^{(0)}, x_2^{(0)}$ PALM performs the iterations

$$\begin{aligned}
x_1^{(r+1)} &\in \mathrm{prox}^f_{\tau_1^{(r)}}\left(x_1^{(r)} - \tfrac{1}{\tau_1^{(r)}}\nabla_{x_1}H(x_1^{(r)}, x_2^{(r)})\right),\\
x_2^{(r+1)} &\in \mathrm{prox}^g_{\tau_2^{(r)}}\left(x_2^{(r)} - \tfrac{1}{\tau_2^{(r)}}\nabla_{x_2}H(x_1^{(r+1)}, x_2^{(r)})\right).
\end{aligned}$$

Further, iPALM 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: $\alpha_i^{(r)}, \beta_i^{(r)}$, $i = 1, 2$ initialization $x_i^{(1)}$, $x_i^{(0)}$, $i = 1, 2$
for $r = 1, 2, ...$ **do** until a convergence criterion is reached

$$\begin{aligned}
y_1^{(r)} &= x_1^{(r)} + \alpha_1^{(r)}(x_1^{(r)} - x_1^{(r-1)}),\\
z_1^{(r)} &= x_1^{(r)} + \beta_1^{(r)}(x_1^{(r)} - x_1^{(r-1)}),\\
x_1^{(r+1)} &\in \operatorname{prox}^f_{\tau_1^{(r)}}(y_1^{(r)} - \tfrac{1}{\tau_1^{(r)}}\nabla_{x_1} H(z_1^{(r)}, x_2^{(r)})),\\
y_2^{(r)} &= x_2^{(r)} + \alpha_2^{(r)}(x_2^{(r)} - x_2^{(r-1)}),\\
z_2^{(r)} &= x_2^{(r)} + \beta_2^{(r)}(x_2^{(r)} - x_2^{(r-1)}),\\
x_2^{(r+1)} &\in \operatorname{prox}^f_{\tau_2^{(r)}}(y_2^{(r)} - \tfrac{1}{\tau_2^{(r)}}\nabla_{x_2} H(x_1^{(r+1)}, z_2^{(r)})).
\end{aligned}$$

end for

In the following, we give details on PALM for our setting. For our problem (14), we choose $f(U) \coloneqq \iota_{\mathrm{St}(d,n)}$, $g(b) \coloneqq 0$ and

$$H(U, b) \coloneqq G(U, b)\eta(\|I_d - U^T U\|_F^2), \tag{24}$$

where

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

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

$$U^{(r+1)} \in \Pi_{\mathrm{St}(d,n)}(U^{(r)} - \tfrac{1}{\tau_1^{(r)}}\nabla_U H(U^{(r)}, b^{(r)})) \tag{25}$$

$$b^{(r+1)} = b^{(r)} - \tfrac{1}{\tau_2^{(r)}}\nabla_b H(U^{(r+1)}, \tilde{b}^{(r)}), \tag{26}$$

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

Remark 4.5. (Projection onto Stiefel manifolds) Concerning this orthogonal projection, it is well known [10], 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 [10]. The authors of [11] 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 gradients with respect to G in (25) and (26). Then the iteration scheme reads as

$$U^{(r+1)} \in P_{\mathrm{St}(d,n)}(U^{(r)} - \tfrac{1}{\tau_1^{(r)}} \nabla_U G(U^{(r)}, b^{(r)})), \tag{27}$$

$$b^{(r+1)} = b^{(r)} - \tfrac{1}{\tau_2^{(r)}} \nabla_b G(U^{(r+1)}, \tilde{b}^{(r)}). \tag{28}$$

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

Lemma 4.6. *Let H be defined by (24). Then the functions $\nabla_U H(\cdot, b)$ and $\nabla_b H(U, \cdot)$ are globally Lipschitz continuous.*

Proof. The function $H(\cdot, b)$ 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. Using G_1 and G_2 in (17) and (18), resp., the derivative of H with respect to b is given by

$$\begin{aligned}
\nabla_b H(U, b) &= \eta(\|I_d - U^T U\|_F^2) \nabla_b G(U, b) \\
&= \tfrac{1}{2}\eta(\|I_d - U^T U\|_F^2) \left(\nabla_b G_1(U, b) + \nabla_b G_2(U, b)\right) \\
&= \eta(\|I_d - U^T U\|_F^2) \sum_{i=1}^{N} \beta_i \Big(\tfrac{1}{\sigma^2}(I_n - UU^{\mathrm{T}})(x_i - b) \\
&\quad - U\Sigma^{-1}(U^{\mathrm{T}} x_i - (U^{\mathrm{T}} b + \mu)) \Big).
\end{aligned}$$

This is a linear function. In particular, it is globally Lipschitz continuous. □

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

(i) $\phi(0) = 0$.

(ii) ϕ is continuously differentiable on $(0, \eta)$.

(iii) For all $s \in (0, \eta)$ it holds $\phi'(s) > 0$.

Definition 4.7 (Kurdyka-Łojasiewicz property)**.** A proper, lower semicontinuous function $\sigma\colon \mathbb{R}^d \to (-\infty, +\infty]$ has the Kurdyka-Łojasieweicz (KL) property at $\bar{u} \in \operatorname{dom} \partial\sigma = \{u \in \mathbb{R}^d : \partial\sigma \neq \emptyset\}$ if there exist $\eta \in (0, \infty]$, a neighborhood U of $\bar{u}$ and a function $\phi \in \Phi_\eta$, such that for all

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

it holds

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

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

Lemma 4.8. *The function H defined in (24) is a KL function.*

Proof. The functions G_1, G_2 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 [1, Remark 5] and [16, 17]. □

The convergence result of PALM in [2] requires the following assumption.

Assumption 4.9 (Assumptions on H)**.** *(i) For any $x_1 \in \mathbb{R}^{d_1}$, the function $\nabla_{x_2} H(x_1, \cdot)$ is globally Lipschitz continuous with Lipschitz constant $L_2(x_1)$. Similarly, for any $x_2 \in \mathbb{R}^{d_2}$, the function $\nabla_{x_2} H(\cdot, x_2)$ is globally Lipschitz continuous with Lipschitz constant $L_1(x_2)$.*

(ii) There exist $\lambda_1^-, \lambda_2^-, \lambda_1^+, \lambda_2^+ > 0$ such that

$$\inf\{L_1(x_2^{(r)}) : r \in \mathbb{N}\} \geq \lambda_1^- \quad \text{and} \quad \inf\{L_2(x_1^{(r)}) : r \in \mathbb{N}\} \geq \lambda_2^-$$
$$\sup\{L_1(x_2^{(r)}) : r \in \mathbb{N}\} \leq \lambda_1^+ \quad \text{and} \quad \sup\{L_2(x_1^{(r)}) : r \in \mathbb{N}\} \leq \lambda_2^+.$$

Remark 4.10. Assume that $H \in C^2(\mathbb{R}^{d_1, d_2})$ fulfills assumption 4.9(i). Then, the authors of [2] showed, that there are partial Lipschitz constants $L_1(x_2)$ and $L_2(x_1)$, such that Assumption 4.9(ii) is satisfied.

Using this assumption, the following theorem was proven in [2, Lemma 3, Theorem 1].

Theorem 4.11 (Convergence of PALM)**.** *Let $F\colon \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to (-\infty, \infty]$ by given by (23) fulfill Assumption 4.9 and let ∇H be Lipschitz continuous on bounded subsets of $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$. Let $(x_1^{(r)}, x_2^{(r)})_r$ be the sequence generated by PALM, where the step size parameters fulfill*

$$\tau_1^{(r)} \geq \gamma_1 L_1(x_2^{(r)}), \quad \tau_2^{(r)} \geq \gamma_2 L_2(x_1^{(r+1)})$$

for some $\gamma_1, \gamma_2 > 1$. Then, for $\eta := \min\{(\gamma_1 - 1)\lambda_1^-, (\gamma_2 - 1)\lambda_2^-\}$, the sequence $(F(x_1^{(r)}, x_2^{(r)}))_r$ is nonincreasing and

$$\tfrac{\eta}{2}\left\|(x_1^{(r+1)}, x_2^{(r+1)}) - (x_1^{(r)}, x_2^{(r)})\right\|_2^2 \leq F(x_1^{(r)}, x_2^{(r)}) - F(x_1^{(r+1)}, x_2^{(r+1)}).$$

If F is in addition a KL function and the sequence $(x_1^{(r)}, x_2^{(r)})_r$ is bounded, then it converges to a critical point of F.

By Lemma 4.6 and 4.8 and the fact that G coincides with H in a neighborhood of the Stiefel manifold we obtain the following corollary.

Corollary 4.12. *Let $(U^{(r)}, b^{(r)})_r$ be generated by (27) and (28) with $\tau_1^{(r)} \geq \gamma_1 L_1(b^{(r)})$ and $\tau_2^{(r)} \geq \gamma_2 L_2(U^{(r+1)})$, where $L_1(b)$ and $L_2(U)$ are the Lipschitz constants of $\nabla_U H(\cdot, b)$ and $\nabla_b H(U, \cdot)$, resp. and $\gamma_1, \gamma_2 > 0$. Consider the sequence generated by PALM with (27) and (28). Then, the sequence $(G(U^{(r)}, b^{(r)}))_r$ is monotone decreasing and the sequence $(U^{(r)}, b^{(r)})_r$ converges to a critical point of G.*

5 Superresolution

In this section, we adapt the superresolution method proposed by Sandeep and Jacob [22] to our PCA-GMM model. The method works in two steps.

1. Learning the PCA-GMM For given low resolution patches $x_{L,i} \in \mathbb{R}^{\tau^2}$ of an image and their higher resolution counterparts $x_{H,i} \in \mathbb{R}^{q^2\tau^2}$, $q \in \mathbb{N}$, $q > 2$, $i = 1, \ldots, N$ we learn a PCA-GMM based on the data $x_i = \begin{pmatrix} x_{H,i} \\ x_{L,i} \end{pmatrix} \in \mathbb{R}^n$, where $n = (q^2+1)\tau^2$, by Algorithm 2. This provides us with parameters $(\mathbf{U}, \mathbf{b}, \alpha, \mu, \mathbf{\Sigma})$ of the reduced d-dimensional GMM. Using these parameters, we compute the parameters of the corresponding high-dimensional mixture model $(\alpha, \tilde{\mu}_k, \tilde{\Sigma}_k)$, $k = 1, \ldots, K$, where μ_k and Σ_k are defined as in (8) and (7). In the following, we use the notations $\tilde{\mu}_k = \begin{pmatrix} \tilde{\mu}_{H,k} \\ \tilde{\mu}_{L,k} \end{pmatrix}$ and $\tilde{\Sigma}_k = \begin{pmatrix} \tilde{\Sigma}_{H,k} & \tilde{\Sigma}_{HL,k} \\ (\tilde{\Sigma}_{HL,k})^{\mathrm{T}} & \tilde{\Sigma}_{L,k} \end{pmatrix}$.

2. Estimation of high resolution patches by MMSE Now we want to improve the resolution of a given low resolution patch $x_L \in \mathbb{R}^{\tau^2}$. First, we select the component k^*, such that the likelihood that x_L belongs to the k^*-th component is maximal, i.e., we compute

$$k^* = \operatorname*{argmax}_{k=1,\ldots,K} \alpha_k f(x_L | \tilde{\mu}_{L,k}, \tilde{\Sigma}_{L,k}).$$

Then we estimate the high resolution patch $x_H \in \mathbb{R}^{q^2\tau^2}$ as the minimum mean-square estimator (MMSE). The following remark briefly review this estimator.

Remark 5.1. (MMSE) Given a random variable $Y : \Omega \to \mathbb{R}^d$ in a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, we wish to estimate a random variable $X : \Omega \to \mathbb{R}^d$, i.e., we seek an estimator $T \colon \mathbb{R}^d \to \mathbb{R}^d$ such that $\hat{X} = T(Y)$ approximates X. A common quality measure for this task is the *mean square error* $\mathbb{E}\big\|X - T(Y)\big\|_2^2$, which gives rise to the definition of the *minimum mean square estimator*

$$T_{\mathrm{MMSE}} \in \operatorname*{argmin}_T \mathbb{E}\|X - T(Y)\|_2^2. \tag{29}$$

Under weak additional regularity assumptions on the estimator T, the Lehmann-Scheffé theorem [15] states that the general solution of the minimization problem (29) is given by

$$T_{\mathrm{MMSE}}(Y) = \mathbb{E}(X|Y).$$

In general, it is not possible to give an analytical expression of the MMSE estimator T_{MMSE}. An exception are Gaussian random variables: if X and Y are jointly normally distributed, i.e.,

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix} \right),$$

then the conditional distribution of X given $Y = a$ is normally as well and reads as

$$(X|Y = a) \sim \mathcal{N}\big(\mu_{X|Y}, \Sigma_{X|Y}\big),$$

where

$$\mu_{X|Y} = \mu_X + \Sigma_{XY}\Sigma_Y^{-1}(a - \mu_Y), \qquad \Sigma_{X|Y} = \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}.$$

As a consequence we obtain for normally distributed random variables the MMSE estimator

$$T_{\mathrm{MMSE}}(Y) = \mathbb{E}(X|Y) = \mu_X + \Sigma_{XY}\Sigma_Y^{-1}(Y - \mu_Y). \tag{30}$$

In our superresolution task, we assume that the vector $\begin{pmatrix} x_H \\ x_L \end{pmatrix}$ is a realization of a random variable $\begin{pmatrix} X_H \\ X_L \end{pmatrix} \sim \mathcal{N}(\tilde{\mu}_{k^*}, \tilde{\Sigma}_{k^*})$. Then, by (30), the MMSE can be computed as

$$x_H = \tilde{\mu}_{H,k^*} + \tilde{\Sigma}_{HL,k^*}(\tilde{\Sigma}_{L,k^*})^{-1}(x_L - \tilde{\mu}_{L,k^*}).$$

3. Reconstruction of the high resolution image by patch averaging We estimate for any patch in the low resolution image the corresponding high resolution patch. Once, we have estimated the high resolution patches, we compute an estimate of the high resolution image in the following way:

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. Now, we assign to each pixel $x_{k,l}$ the weight

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

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.

6 Numerical Results

In this section, we demonstrate the performance of our algorithm by two- and three-dimensional examples, where we mainly focus on material data which provided the original motivation for this work. More precisely, in the frame of the ITN MUMMERING, a series of multi-scale 3D images has been acquired by synchrotron micro-tomography at the SLS beamline TOMCAT. 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 one is a composite ("SiC Diamonds") obtained by microwave sintering of silicon and diamonds, see [24].

Sections of the corresponding 3D images are given the first two columns of Figure 1.

All implementations 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.

For the implementation of PALM and iPALM, we use the implementation framework from [9][1]. As suggested in [21] 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 $\nabla_b G(U^{(r+1)}, \cdot)$.
We generate pairs of high and low resolution images using the following superresolution operator:

Generation of the test examples. For convenience, we describe the generation in 2D. The 3D setting is treated in a similar way. We use the operator A from the implementation of [19][2]. This operator consists of a blur operator H and a downsampling operator S. The blur operator is given by a convolution with a Gauss kernel with standard deviation 0.5. For the downsampling operator S we use the discrete Fourier transform (DFT). Given an image $x \in \mathbb{R}^{m,n}$ the two-dimensional DFT is defined by $\mathcal{F}_{m,n} \coloneqq \mathcal{F}_n \otimes \mathcal{F}_m$, where $\mathcal{F}_n = (\exp(-2\pi ikl/n))_{k,l=0}^{n-1}$. Now, the downsampling operator $S\colon \mathbb{R}^{m,n} \to \mathbb{R}^{m_2,n_2}$ is given by

$$S = \frac{m_2 n_2}{mn} \mathcal{F}_{m_2,n_2}^{-1} D \mathcal{F}_{m,n},$$

where for $x \in \mathbb{R}^{m,n}$ the (i,j)-th entry of $D(x)$ is given by

$$\begin{cases} x_{i,j}, & \text{if } i \le \frac{m_2}{2} \text{ and } j \le \frac{n_2}{2}, \\ x_{i+m-m_2,j}, & \text{if } i > \frac{m_2}{2} \text{ and } j \le \frac{n_2}{2}, \\ x_{i,j+n-n_2}, & \text{if } i \le \frac{m_2}{2} \text{ and } j > \frac{n_2}{2}, \\ x_{i+m-m_2,j+n-n_2}, & \text{if } i > \frac{m_2}{2} \text{ and } j > \frac{n_2}{2}. \end{cases}$$

[1] `https://github.com/johertrich/iSPRING`
[2] `https://github.com/pshibby/fepll_public`

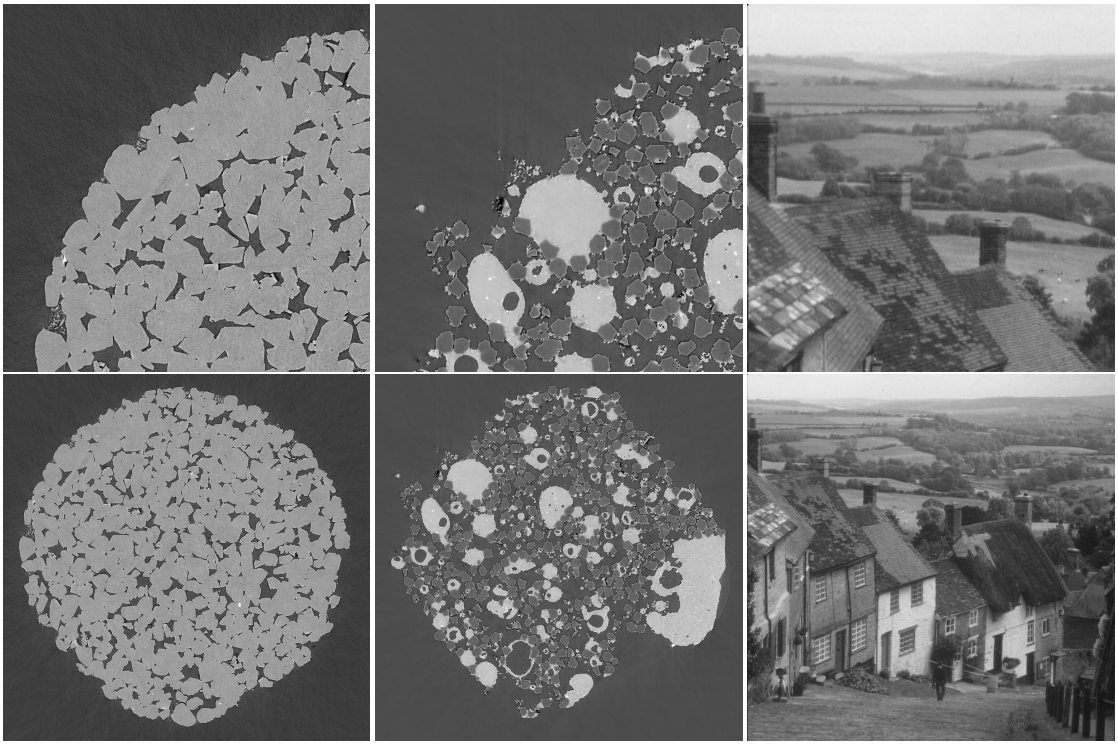


Figure 1: Top: Images for estimating the mixture models. Bottom: Ground truth for reconstruction. First column: Material "FS", second column: Material "SiC Diamonds", third column: `goldhill` image.

For a given high resolution image x, we now generate the low resolution image y by $y = Ax + \epsilon$, where epsilon is white Gaussian noise with standard deviation $\sigma = 0.02$.

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 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.

Now, we estimate the parameters of a GMM and for a GMM with PCA (PCA-GMM) as described in the previous sections. 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 [22] as described in the previous section to reconstruct the ground truth from the artificially downsampled images. The resulting PSNRs are given in Table 1. As a reference we also measure the PSNR of the bicubic interpolation. Figure 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).

		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	-	35.48	37.20	31.62	30.65	30.62	27.77
PCA-GMM	20	35.45	37.00	31.53	30.21	30.42	27.60
	16	35.41	36.88	31.48	29.91	30.14	27.63
	12	35.30	36.73	31.44	29.90	29.93	27.45
	8	35.18	36.61	31.19	30.09	30.07	27.35
	4	34.60	35.30	30.54	29.77	29.23	26.94

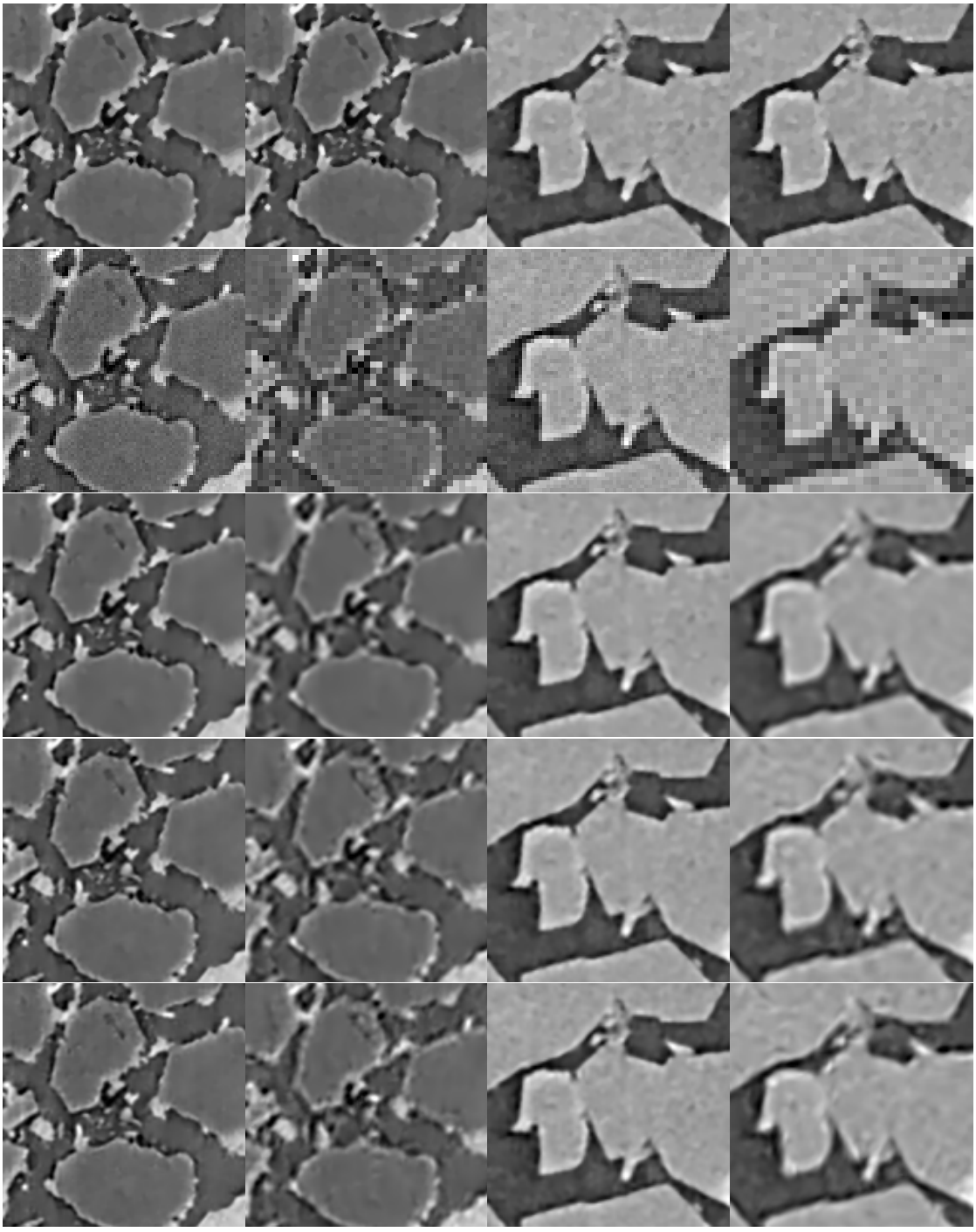
Table 1: PSNRs of the reconstructions of artificially downsampled 2D images using either bicubic interpolation, a GMM or PCA-GMM for different choices of d. The magnification factor is set to $q \in \{2, 4\}$. PCA-GMM produces results almost as good as GMM, with a much lower dimensionality.

3D-Data. In the following, we present the same experiments as in the 2D-case but with 3D-data. For this 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 as described in the previous sections. 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 [22] as described in the previous paragraph to reconstruct the ground truth from of the artificially downsampled images. The resulting PSNRs are given in Table 2. As a reference we also measure the PSNR of the nearest neighbor interpolation.

7 Conclusions

In this paper, we presented a new algorithm to perform image superresolution. Based on previous work by Sandeep and Jacob [22], we added a dimension reduction step within the GMM model using PCA on patches. The new variational model, called PCA-GMM is



(a) Diamonds, $q = 2$ (b) Diamonds, $q = 4$ (c) FS, $q = 2$ (d) FS, $q = 4$

Figure 2: Reconstructions of 2D low resolution images. 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.

	d	FS	Diamonds
Nearest neighbor	-	30.10	26.25
GMM	-	33.30	30.73
PCA-GMM	60	32.87	30.40
	40	33.00	30.49
	20	33.22	30.14

Table 2: 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.

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 4.1. However, our M-step requires to solve a non-convex constrained minimization problem. Here we propose a PALM algorithm and prove that all assumptions for the convergence of the sequence of iterates to a critical point required by [2] are fulfilled, see Corollary 4.12. 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 dimension reduction is of the utmost importance when dealing with 3D images, where the size of the data gets very large.

As future work, apart from the mathematical analysis of the EM algorithm with approximate M-step, we intend to work on the robustness of the method. This could be done by using a robust PCA [18], and also by making the model invariant to contrast changes, see, e.g. [8]. Further, we aim to deal with material examples, where we do not subsample the images in a synthetic way. In particular, we will not know the subsampling operator. Within ITN MUMMERING such measurements were taken, but require to undergo an advanced registration process.

Finally, we are aware of deep learning techniques for superresolution, e.g. [13]. We will consider such approaches in the future which would also benefit from dimensionality reduction, in particular in 3D.

Acknowledgment

Funding by the German Research Foundation (DFG) within the project STE 571/16-1 as well as by the French Agence Nationale de la Recherche (ANR) under reference ANR-18-CE92-0050 SUPREMATIM, is gratefully acknowledged. The EU Horizon 2020

Marie Sklodowska-Curie Actions Innovative Training Network MUMMERIN (MUltiscale, Multimodal and Multidimensional imaging for EngineeRING, Grant Number 765604) is also acknowledged.

References

[1] 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.

[2] 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.

[3] C. L. Byrne. *The EM Algorithm: Theory, Applications and Related Methods*. Lecture Notes, University of Massachusetts, 2017.

[4] S. Chrétien and A. O. Hero. Kullback proximal algorithms for maximum-likelihood estimation. *IEEE Transactions on Information Theory*, 46(5):1800–1810, 2000.

[5] S. Chrétien and A. O. Hero. On EM algorithms and their proximal generalizations. *ESAIM: Probability and Statistics*, 12:308–326, 2008.

[6] C. A. Deledalle, J. Salmon, and A. Dalalyan. Image denoising with patch based pca: local versus global. *Proceedings of the British Machine Vision Conference*, 25:1–25, 2011.

[7] 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.

[8] J. H. Fitschen, K. Losch, and G. Steidl. Unsupervised multi class segmentation of 3d images with intensity inhomogeneities. *Journal of Visual Communication and Image Representation*, 46:312–323, 2017.

[9] J. Hertrich and G. Steidl. Inertial stochastic PALM and its application for learning Student-t mixture models. *ArXiv preprint arXiv:2005.02204*, 2020.

[10] 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.

[11] 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.

[12] A. Houdard, C. Bouveyron, and J. Delon. High-dimensional mixture models for unsupervised image denoising (HDMI). *SIAM Journal on Imaging Sciences*, 11(4):2815–2846, 2018.

[13] T. Klatzer, D. Soukup, E. Kobler, K. Hammernik, and T. Pock. Trainable regularization for multi-frame superresolution. In V. Roth and T. Vetter, editors, *Pattern Recognition, GCPR 2017, Proceedings*, volume 10496 of *Lecture Notes in Computer Science*, pages 90–100. Springer, 2017.

[14] F. Laus. *Statistical Analysis and Optimal Transport for Euclidean and Manifold-Valued Data*. PhD Thesis, TU Kaiserslautern, 2019.

[15] E. Lehmann and H. Scheffé. Completeness, similar regions, and unbiased estimation: Part I. *Sankhyā: the Indian Journal of Statistics*, pages 305–340, 1950.

[16] 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.

[17] S. Łojasiewicz. Sur la géométrie semi- et sous-analytique. *Université de Grenoble. Annales de l'Institut Fourier*, 43(5):1575–1595, 1993.

[18] S. Neumayer, M. Nimmer, S. Setzer, and G. Steidl. On the rotational invariant l_1-norm PCA. *Linear Algebra and its Applications*, 587:243–270, 2020.

[19] S. Parameswaran, C. Deledalle, L. Denis, and T. Q. Nguyen. Accelerating gmm-based patch priors for image restoration: Three ingredients for a 100x speed-up. *IEEE Transactions on Image Processing*, 28(2):687–698, 2019.

[20] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[21] 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.

[22] P. Sandeep and T. Jacob. Single image super-resolution using a joint GMM method. *IEEE Transactions on Image Processing*, 25(9):4233–4244, 2016.

[23] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[24] 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.

[25] 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. IEEE, 2011.