Blind Image Deconvolution Using Machine Learning For Three-Dimensional Microscopy

Tal Kenig, Zvi Kam and Arie Feuer, Fellow, IEEE

Abstract—In this work, we propose a novel method for the regularization of blind deconvolution algorithms. The proposed method employs example-based machine learning techniques for modeling the space of point spread functions. During an iterative blind deconvolution process, a prior term attracts the point spread function estimates to the learned point spread function space. We demonstrate the usage of this regularizer within Bayesian blind deconvolution framework, and also integrate into the latter a method for noise reduction, thus creating a complete blind deconvolution method. The application of the proposed algorithm is demonstrated on synthetic and real world three-dimensional images acquired by a wide-field fluorescence microscope, where the need for blind deconvolution algorithms is indispensable, yielding excellent results.

Index Terms—blind deconvolution, deblurring, machine learning, PCA, kernel PCA, microscopy.

1 INTRODUCTION

An image acquired by an imaging system may be degraded due to numerous reasons. Two of the most common reasons for image degradation are blur and statistical noise. Sometimes, the blurring operation can be assumed to be linear and shift invariant and therefore, the acquired image can be modeled as a convolution of the imaged object with some convolution kernel, representing the imaging system. This convolution kernel is often referred to as the point spread function (PSF) of the imaging system. Neglecting the effect of statistical noise, the blurring is mathematically modeled as a convolution, and therefore the process of inverting this operation is termed deconvolution. In addition, there are situations where explicit knowledge of the convolution kernel cannot be assumed. Therefore, inverting the convolution operation under those circumstances is termed blind deconvolution (BD).

In general, the assumed image degradation model can be formulated as

\[ g = n(h \ast f) \]  

and in the discrete case, we formulate (1) explicitly as:

\[ g(x) = n \left( \sum_{k \in \Omega} f(k) h(x - k) \right) \]

where:

- \( x \in \Omega \) is a \( p \)-tuple of discrete spatial coordinates corresponding to the spatial locations sampled by the imaging system
- \( f : \Omega \rightarrow \mathbb{R} \) denotes the imaged object
- \( h : \Omega \rightarrow \mathbb{R} \) denotes the imaging system PSF
- \( g \) denotes the acquired image
- \( n \) denotes a pixel-wise noise function
- \( \ast \) denotes the \( p \)-dimensional convolution operator

The goal of BD algorithms is to estimate \( f \) and \( h \) from \( g \) and any additional prior knowledge available.

In what follows, we assume that the dominant source of statistical noise in the imaging system is photon noise. Photon noise is common to many imaging systems, as it originates from the stochastic nature of the photon detection process at the sensor and is distributed according to Poisson distribution [5], with its parameter being the noise-free grey level value at each pixel location. The photon noise can be characterized as follows:

\[ p(g(x)|f, h) = \frac{(f \ast h)(x)g(x)e^{-(f \ast h)(x)}}{g(x)!} \]

It can be easily observed that this type of noise is highly correlated with the signal, which makes it much more difficult to handle than additive noise.

It is not difficult to realize that the problem (1) is ill-posed, in the sense that numerous solutions exist. This can be easily shown even without examining any specific method for solution. First, we note that the trivial solution \( f \equiv g, h \equiv \delta \), where \( \delta \) denotes the three-dimensional Dirac function, always exists. Second, if we examine (1), even without noise, from an information quantity point of view we can see that we have one image \( g \) as input, from which we are required to estimate both \( f \) and \( h \). Hence, the problem at hand is underdetermined, i.e. there are more unknowns than inputs. Due to this ill-posedness, all BD methods use some sort of regularization. Many algorithms regularize the object of interest, usually by applying some sort of smoothness...
constraint see, e.g. [18], [30], [34], [36]. However, in general, the object we wish to reconstruct is unknown and regularizing it may have an adverse effect, especially in cases where it is known to have high frequency structures.

Contrary to that, the PSF we seek is usually directly related to the imaging system, thus providing us with some prior knowledge of its form. We note that in BD problems, although the PSF is not exactly known, it typically belongs to some distinct class. Therefore, we wish to integrate into the BD process as much accurate and specific information about the PSF as possible. Sometimes, this information is theoretically available, since we know the general form of the PSF, e.g. in the cases of linear motion blur, atmospheric turbulence blur and out-of-focus blur [6]. Even when this is not the case, there are many situations where the PSF can be experimentally measured under varying acquisition conditions, thus providing us a sampling of the class of PSFs.

We propose to regularize the PSF by attracting it to a space of admissible PSFs, denoted as PSF-space, known to represent the imaging system at hand. This is done by first performing example based learning of the PSF-space by using linear principal component analysis (PCA) [49] or kernel PCA (KPCA) [57]. We then introduce into the deconvolution algorithm a term that attracts the PSF towards PSF-space.

This approach has never been proposed in the context of blind deconvolution, to the best of our knowledge. The inspiration for the proposed approach stems from the prolific activity in the field of statistical shape priors for segmentation using active contours in the past decade [13], [19], [20], [38], [54], [64].

The algorithm presented in this work has been developed for the specific application of restoring three-dimensional (3D) microscopic imagery. However, we present it in a general enough form to facilitate the application of the proposed algorithm for other uses of BD, as well.

We now outline the contents of this paper. In section 2 we briefly review related work in the field of BD. Section 3 contains the main results of the Bayesian BD formulation used as a framework for the proposed method. Section 4 contains the main contribution of our work, which is a novel PSF regularization method, utilizing learning based algorithms. Section 5 briefly discusses the convergence of the proposed algorithm. In section 6 we provide the necessary details about wide field fluorescence microscopic imaging, which is required for the understanding of the numerical experiments depicted in section 7, which are conducted in order to validate the proposed algorithm. Finally, we conclude our work and suggest some further research topics in section 8.

2 PREVIOUS WORK

In general, BD algorithms can be divided into two categories: A-priori blur identification methods and joint blur identification and image restoration methods [6]. The first category consists of algorithms which separate BD into two stages. The first stage consists of identification of the PSF. Once the PSF estimate is available, a non-blind deconvolution is carried out.

The earliest methods in this field are [7], [61], which assume that the PSF is of known parametric form and is completely characterized by its frequency domain zeros, which is frequently not the case. Another popular BD method in this field is zero sheet separation [37], in which the PSF is identified by separating the two-dimensional (2D) Z-transform of the corrupted image into two convolutive factors. In [56], the authors suggest calculating several restorations of the acquired images, using different PSFs. Those PSFs are obtained by acquiring in advance small, point-like objects under different imaging conditions. Then, the best image is selected either by some numerical measure, or by visual inspection. Two other related methods for blur identification include generalized cross validation (GCV) [52] and kurtosis minimization [40]. Both methods assume an underlying parametric model for the PSF, perform multiple deconvolutions with different PSF parameters, and choose the PSF which leads to minimization of the GCV prediction error in [52], or the kurtosis of the restored image in [40]. Due to the need for performing multiple restorations, the deconvolution methods reported in [56], [52] and [40] are limited to computationally inexpensive algorithms. The authors of [56] use a fast, semi 3D deconvolution algorithm, despite the fact their problem is fully 3D, and the authors of [52] and [40] use linear, Wiener-like inverse filters for image restoration. Another recent method in the class of a-priori blur identification methods, which has been used in the context of microscopy, relies on the assumption that the imaged object contains small point-like structures [16]. The latter are identified, cropped from the acquired image and serve as a PSF estimate.

Most recent work in the field of BD is related to the class of joint image restoration and blur identification methods. The methods in this class are numerous and diverse, and we do not pertain to cover them all. For a thorough review of BD methods the reader is referred to [6], [8], [34] and the references therein. In essence, the methods in this class can be categorized according to the deconvolution framework itself and the applied regularization.

One of the most popular BD frameworks is Bayesian BD, which is also used in our work. In Bayesian framework, the observed image and sometimes also the object and PSF are considered to be a single realization of a stochastic process. Some underlying probability distribution of those stochastic processes is assumed and a cost functional which aims at optimizing some statistical quantity is formulated and optimized. Usually, in creating this cost functional, one of the versions of the famous Bayes rule is utilized. Especially popular Bayesian methods are the maximum likelihood and maximum a-posteriori methods, which are discussed in some detail.
in section 3. Examples for the use of such methods can be found in [18], [22], [29], [30], [33], [65], [68] to name a few.

Due to the ill-posed nature of the BD problem, essentially all BD algorithms utilize prior knowledge in the deconvolution process, which is required in order to reduce the degrees of freedom of the problem. One class of regularization methods is the technique of projection onto convex sets (POCS). In this technique, some knowledge of the object or the PSF is forced, by projecting the intermediate object and/or PSF estimates onto some convex sets. For example, popular constraints used for POCS regularization are non-negativity, spatial and frequency domain supports and phase information of the object. POCS methods can be used as regularizers in conjunction with other iterative methods, as in [1], [33], [34], [65], or as a BD method of itself as reported in [46], [66].

Another form of regularization used in BD is parametrization of the PSF. In these methods the PSF is either of known parametric form, or is assumed to be well approximated by a simple function dependent on a few parameters. Then, during the reconstruction process, only the PSF parameters need to be estimated, thus dramatically reducing the degrees of freedom in the problem. Examples for methods that use PSF parametrization are available in [43], [48].

Some regularization methods used for BD induce smoothness of the solution. Those methods usually incorporate into the BD process a term that attempts to minimize some differential quantity of the estimated object and/or PSF, such as the Laplacian, or an anisotropic regularization term, such as total variation [55]. Examples for this type of regularization in BD have been previously reported in [30], [67], and in [18] in the context of non-blind deconvolution.

The adaptive image deconvolution algorithm (AIDA) [30] contains a PSF regularization term which attracts the PSF estimate towards a given PSF. This approach can be viewed as a particular case of the approach we propose, where the PSF-space is degenerate and “learned” by a single example. This algorithm has been tested on 2D and 3D datasets, including 3D microscopy imagery and we use it for comparison purposes in the sequel.

Another class of methods for regularization of BD algorithms are learning based algorithms. Not much research has been conducted in this field, and we note the recent work in [45], [47], [39]. Similar methods are also used for example-based super-resolution algorithms [23], [15]. In those works, the basic idea is that the prior knowledge required for solving the restoration problem can be learned from training data, i.e. a set of prototype images representing the same class of images as the ones processed. The training images are first corrupted in a similar manner to the expected degradation, thus creating image pairs. Then, the training images are cropped into small patches. During the reconstruction process, the surrounding patch of each image pixel is extracted, and similar degraded patches are found within the training data. The high quality patches paired with the found degraded patches are then used within the restoration process. While all those methods share a similar idea, they differ in the way the training data is pre-processed, organized and used.

We mention these methods since they are somewhat related to the method we propose, in the sense that they use example based learning. However, we stress that this approach is fundamentally different from the one we propose in this paper. First, we do not make any assumption regarding the class of objects, whereas the above mentioned methods use very strong assumptions regarding the images to be processed. Second, we use machine learning methods in order to learn the parametrization of the PSF-space, and the objects which the learning process is employed upon are whole PSFs, rather than small image patches.

Finally, we note that much attention has been given to the problem of blind deblurring in the graphics and computer vision literature in recent years. Most of this activity focuses on the important problem of removing motion blur from natural images, see e.g. [68], [31], [21], [32], [58], [50], [11]. In this context, we mention that our proposed method does not pertain to solve the problem of motion blur. This limitation stems from the learning methods we employ, which are incapable of capturing the properties of the motion blur PSF, since it is usually very diverse and unpredictable. In addition, some of this work addresses spatially variant blur [50], [11], involves acquisition of more than one photograph in special settings [68], or using dedicated acquisition devices [50], all of which are outside the scope of this work.

3 BD FRAMEWORK

The algorithm we propose in this work is based on the Maximum Likelihood Expectation Maximization (MLEM) and maximum a-posteriori (MAP) algorithms. The MLEM algorithm was first introduced in the context of positron emission tomography reconstruction [59], and is similar to the Richardson-Lucy (RL) algorithm [41], [53].

3.1 Maximum a-posteriori

Maximum a-posteriori (MAP), is an algorithm which optimizes the following cost function:

\[ \hat{f}, \hat{h} = \arg \max_{f,h} \{ p(f, h | g) \} \]

According to the Bayes rule \( p(f, h | g) = \frac{p(g | f, h) p(f, h)}{p(g)} \). We can assume with high confidence that \( f \) and \( h \) are statistically independent, so we get \( p(f, h | g) = \frac{p(g | f, h) p(f)p(h)}{p(g)} \).

1. The derivations of MLEM and RL stem from different assumptions, yet result in the same iterative process. See [8] for further details.
The log likelihood of this expression is
\[
\log (p(f, h|g)) = \log (p(g|f, h)) + \log (p(f)) + \log (p(h)) - \log (p(g))
\] (3)

We note that \(\log (p(g))\) is not dependent upon \(f\) or \(h\).
The term \(\log (p(g|f, h))\) is governed by Poisson statistics, according to (2). The terms \(\log (p(f))\) and \(\log (p(h))\) can be interpreted as prior information regarding the object \(f\) and the PSF \(h\), respectively. At this point, we assume uniform distributions for \(f\) and \(h\), thus dropping the prior terms, which reduces the algorithm to the MLEM, or RL. However, we will show in Sections 3.2 and 4 how we regularize the object and PSF estimates, respectively. By substituting (2) into (3) and differentiating, we arrive at the following iterative scheme:
\[
\begin{align*}
f_{k+1} &= f_k \cdot \left[ \left( \frac{g}{f_k + h_k} \right) * \bar{h}_k \right] \\
h_{k+1} &= \frac{h_k}{\sum_{x \in \Omega} f_k(x)} \left[ \left( \frac{g}{f_k + h_k} \right) * f_k \right]
\end{align*}
\] (4)

where the superscript \(s\) denotes symmetrical reflection: \(\xi^s(x) = \xi(-x)\) and \(k\) is the iteration index. Detailed derivation of the MLEM formulation can be found in [26], [8).

3.2 MLEM with integrated denoising

We neither possess any specific knowledge of our object \(f\), nor wish to impose any general prior which assumes smoothness of any kind, since it may be inadequate for certain objects. However, it is implied by (3) that some regularization of the object estimate is required.

In order to overcome the noise, we use a generalization of a scheme from the field of astronomical imaging, of incorporating denoising within the MLEM framework [60]. This method is based on the observation that at each MLEM iteration \(k\), the acquired image can be decomposed into two components: the estimated image at iteration \(k\) and a residual: \(g = f_k * h_k + R_k\). It can be noticed that the first term \(f_k * h_k\) is a smooth term (since \(h_k\) is a blurring kernel and as such is low-pass in nature), and therefore the residual \(R_k\) must contain the noise. This method suggests denoising \(R_k\) at each iteration: \(\bar{R}_k = \text{Denoise}(R_k)\).

The authors of the original work [60] use wavelet based denoising for the function \(\text{Denoise}()\). However, in general there is no restriction regarding the denoising function. Using the aforementioned denoising scheme, the residual - denoised MLEM (RD-MLEM) iteration becomes:
\[
\begin{align*}
R_k &= g - f_k * h_k \\
\bar{R}_k &= \text{Denoise}(R_k) \\
f_{k+1} &= f_k \cdot \left[ \left( \frac{f_k * h_k + \bar{R}_k}{f_k * h_k + R_k} \right) * \bar{h}_k \right] \\
h_{k+1} &= \frac{h_k}{\sum_{x \in \Omega} f_k(x)} \left[ \left( \frac{f_k * h_k + \bar{R}_k}{f_k * h_k + R_k} \right) * f_k \right]
\end{align*}
\] (5)

In our experiments, the RD-MLEM was implemented using a 3x3x3 median filter [28] for the \(\text{Denoise}()\) function, yielding satisfactory results.

We would like to briefly discuss our choice of the median filter. First, we note that the use of a linear filter is not desirable for this denoising task. If we denote \(\bar{g}_k = f_k * h_k + \bar{R}_k\) the denoised estimate of \(g\) at iteration \(k\) and assume that \(\text{Denoise}()\) is a linear shift-invariant filter characterized by the convolution kernel \(m\), we get
\[
\bar{g}_k = f_k * h_k + m * \left( g - f_k * h_k \right) = (\delta - m) * (f_k * h_k) + m * g
\]

Since \(f_k * h_k\) is a smooth term and \(\delta - m\) is a high-pass filter\(^2\) we expect the term \((\delta - m) * (f_k * h_k)\) to be negligible. This means that \(\bar{g}_k\) is essentially a low-pass filtered version of \(g\). Thus, we further blur the image \(g\), which is in contrast to the goal of the deconvolution process. Another important detail is that we denoise \(g - f_k * h_k\), which ideally contains only noise, and in practice, we do not expect to be of any specific structure. Therefore, we refrain from using complicated denoising methods that assume the presence of edges or any other kind of structure in the denoised image. We choose to use the median filter, which is non-linear, efficient and does not make any assumption regarding the underlying structure of the denoised image.

Finally, we note that although this scheme for noise reduction is very simple, in practice we found it to be very efficient in suppressing noise, without compromising fine detail in the images. This property of the denoising scheme stems from the fact it does not impose any constraints regarding the restored object, but instead, heavily relies on the convolutive imaging model.

4 PSF regularization by example-based learning

In this section, we introduce the main novelty of this work. We propose to regularize the PSF by attracting it to a space of admissible PSFs, denoted as PSF-space. This is done by first performing example based learning of the PSF-space by using linear principal components analysis (PCA) [49] or kernel PCA (KPCA) [57]. We then introduce into the deconvolution algorithm a term that attracts the PSF towards PSF-space.

4.1 The inspiration: segmentation with shape priors

The inspiration for the proposed approach stems from the prolific activity in the field of statistical shape priors for segmentation using active contours in the last decade [13], [14], [19], [20], [38], [54], [64] and specifically, our work follows closely the methods presented in [12], [13], [14]. In those methods of segmentation, a contour is evolved over the image domain under the influence of an iterative process, usually aimed to optimize some cost functional. The latter is designed in such a manner that

\(^2\) Since \(m\) is a low-pass filter
at convergence, the contour will delineate an object of interest within the image.

Due to noise, clutter and occlusions, it is not always sufficient to use the given imagery data for the contour evolution. Therefore, a significant amount of research has been conducted in order to include into the contour evolution process prior information regarding the shape of the object of interest.

All of the above referenced methods operate in a similar outline: First, the space of shapes is learned from a training set of examples. Then, during the iterative process, a regularization term which attracts the contour under evolution towards the previously learned shape-space is included within the contour evolution process.

We make a direct analogy between those methods and the BD problem by replacing the shapes of interest in the active contour framework with the functions which are of our interest - the PSFs. In the sequel we will show how we adapt and apply the methods previously developed for constructing shape priors for segmentation methods as PSF priors for BD.

Finally, we would like to note that the BD problem is inherently free of a problem related to shape priors in the active contours framework, which is the problem of geometrical shape alignment and scaling. In the BD problem, we regularize functions and not shapes. Therefore, the PSFs are inherently aligned and scaled with respect to a global set of axes.

### 4.2 Integration as a prior

As we show later, we use learning methods in order to learn the PSF-space, prior to the deconvolution process. Then, at the end of each iteration $k$, we find the best approximation of the PSF estimate $\hat{h}_k$ within PSF-space. This approximation is denoted as $A(\hat{h}_k)$.

In order to use $A(\hat{h}_k)$ for PSF regularization, we propose a soft projection stage to be carried after each iteration:

$$P(h) = \eta A(h) + (1-\eta)h$$

where $0 \leq \eta \leq 1$. The soft projection operation can be regarded as a weighted average of $h$ and $A(h)$, which draws $h$ towards its approximation.

The complete iterative scheme we use, including the integrated denoising stage, is summarized as follows:

$$R_k = g - \hat{f}_k \ast \hat{h}_k$$

$$\overline{R}_k = \text{Denoise}(R_k)$$

$$\hat{f}_{k+1} = \hat{f}_k \ast \left( \left( \frac{\hat{f}_k \ast \hat{h}_k + \overline{R}_k}{\hat{f}_k \ast \hat{h}_k} \right) \ast \hat{h}_k \right)$$

$$\hat{h}_{k+1}^\text{temp} = \hat{h}_k \ast \left( \sum_{x \in \Omega} \left( \frac{\hat{f}_k \ast \hat{h}_k + \overline{R}_k}{\hat{f}_k \ast \hat{h}_k} \right) \ast \hat{f}_k \right)$$

$$\hat{h}_{k+1} = \eta A(\hat{h}_{k+1}^\text{temp}) + (1-\eta)\hat{h}_{k+1}^\text{temp}$$

### 4.3 Statistical learning methods

In the following section, we review two learning methods, which we use in order to construct the PSF-space and the PSF approximation $A(h)$. We use PCA [49] and KPCA [57] in order to learn the PSF-space. We note that PCA can be considered as a particular case of KPCA. However, for the sake of clarity, we present those methods separately.

#### 4.3.1 Principal Components Analysis

PCA is a vector space transform which, for a given dataset, finds a transformation of the axes in such a manner that the greatest variance by any projection of the data comes to lie on the first axis (called the first principal component), the second greatest variance on the second axis, and so on. Once this transform is calculated, the data is represented by the first $l$ principal directions, which are assumed to capture most of the variance in the data. It can be shown [49] that the principal directions are in fact the eigenvectors of the data covariance matrix. The number $l$ of principal directions to be used is usually determined by inspecting the eigen-spectrum of the covariance matrix.

This approach is especially useful when the data is of very high dimension, but is known to have only a few degrees of freedom. This is exactly the case in the problem we have at hand. Our sampled PSFs have a very large number of voxels we need to estimate, and we can view them as very high dimensional vectors. However, we know that the PSF has no more than a few degrees of freedom, as PSFs representing physical imaging systems are usually determined by a small number of parameters. By using PCA, we assume that the few degrees of freedom of the PSFs can be represented by a subspace of this high dimensional vector space. We note that this assumption is in general not true, as we cannot assure that an arbitrary PSF can be represented as a linear combination of other PSFs. We further discuss this assumption, and propose a method of relaxing it in the sequel.

In what follows, we assume that we have a training set of $N$ sampled PSFs $\{h_1, h_2, \ldots, h_N\}$, each containing $n_v$ voxels. Those training samples are assumed to represent PSFs of the imaging system used to acquire the images we wish to restore.

First, the mean PSF $\bar{h}$ is computed by taking the mean of the training PSFs $\bar{h} = \frac{1}{N} \sum_{k=1}^{N} h_k$. Next, we compute the covariance matrix $C$ as follows. The mean PSF is subtracted from all training PSFs to create the centered training PSFs $h_k = h_k - \bar{h}$, and the latter are lexicographically ordered as column vectors. Each centered training sample is placed as the $k$-th column of an $n_v \times N$ matrix $A = [h_1^T, h_2^T, \ldots, h_N^T]$. The $n_v \times n_v$ symmetric covariance matrix is defined as $C = \frac{1}{N} A A^T$. Then, an eigen-decomposition is performed on $C$, $C = U \Sigma U^T$, where $U$ is a matrix whose columns are the eigenvectors of $C$ (the principal components) and $\Sigma$ is a diagonal matrix containing the eigenvalues of $C$. The eigenvectors of $C$ are the principal components of $A$. The number of principal components to be used is usually determined by selecting the number of principal components with the largest eigenvalues. In practice, the number of principal components is determined by visual inspection of the eigen-spectrum of $C$.
matrix with the corresponding eigenvalues of C on its diagonal, in descending order.

We denote by \( u_k \) the \( k \)-th principal component. The principal components are sampled eigen-PSFs, which can be used as a basis for representation of other PSFs. We denote by \( U_l \) the \( n_i \times l \) matrix composed of the first \( l \) columns of \( U \). The coordinates of the projection of an arbitrary test function \( h \) onto the space of PSFs spanned by the first \( l \) principal components can be computed as

\[
\alpha^l = U_l^T (h - \bar{h})
\]

And the projection itself can be calculated as

\[
P^l(h) = \sum_{k=1}^{l} \alpha^l_k u_k + \bar{h}
\]

(9)

where \( \alpha^l_k \) is the \( k \)-th component of the vector \( \alpha^l \). It can be shown [49] that the error of representing any of the training set examples \( h_k \) by their projection \( P^l(h_k) \) is minimal in the least square sense.

It is important to point out that during the derivation of PCA it is assumed that any PSF within the dataset, and essentially within the PSF-space, can be represented as a linear combination of some basis (a vector space assumption). As previously noted, this assumption is in general not true, as a PSF cannot be represented in general as a linear combination of the sample PSFs. In spite of this, this assumption can be made as a rough approximation, especially when an arbitrary PSF is close to the training set in the sense of small variations around its mean.

Finally, we take the approximation \( A(\cdot) \) in (7) simply to be the projection \( P^l(\cdot) \), i.e.

\[
A(\tilde{h}_{k+1}^{\text{temp}}) = P^l(\tilde{h}_{k+1}^{\text{temp}})
\]

(10)

### 4.3.2 Kernel PCA

In order to relax the vector space assumption, which is made when using the PCA method, we propose to use Kernel PCA. Kernel PCA can be considered as a non-linear extension of PCA. KPCA was first introduced in [44], [57], where it was proven to be a powerful technique for extracting the non-linear structure of a dataset.

In KPCA, the data is first mapped from the original input space \( \mathcal{I} \), where it resides, to some feature space \( \Gamma \) via a non-linear function \( \varphi \). Then, linear PCA is employed in feature space to find the feature space principal components, which correspond to the largest variations of the dataset in feature space. Similarly to the linear PCA case, we use the first \( l \) principal components, which account for the largest variation possible within the dataset using \( l \) directions.

Due to the non-linearity of the mapping \( \varphi \), this technique enables us to capture non-linear variations within the dataset. In essence, it relaxes the assumption that an arbitrary PSF can be represented as a linear combination of the training set PSFs. More specifically, by using KPCA, we assume that the above assumption holds in feature space, thus allowing the representation of an arbitrary PSF as some non-linear function of the training set PSFs.

In general, the feature space can be of very high dimension [44], and in practice, we do not necessarily know which transformation \( \varphi \) will provide useful results. In order to overcome those limitations, the mapping \( \varphi \) is not carried out in explicit form. Moreover, \( \varphi \) is usually not known and all calculations are carried out by the use of Mercer kernels. A Mercer kernel is a function \( k : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R} \), which for all points in the dataset \( k(h_i, h_j) \) is positive and symmetric with respect to its arguments, i.e. \( k(h_i, h_j) = k(h_j, h_i) \). It has been shown [4] that applying the kernel in input space is equivalent to the calculation of an inner product in feature space:

\[
k(h_i, h_j) = \langle \varphi(h_i), \varphi(h_j) \rangle
\]

(11)

Therefore, all calculations that can be formulated in terms of inner products in feature space, can be carried out in input space without explicitly using the mapping \( \varphi \) and only the kernel \( k(\cdot, \cdot) \) has to be known.

The choice of the kernel can be arbitrary, as long as it satisfies the conditions of symmetry and positivity. One especially popular choice for a kernel, which was also proven to be useful [44], [13], [14], is the exponential kernel \( k(h_i, h_j) = \exp \left(-\frac{\|h_i - h_j\|^2}{2\sigma^2}\right) \), where \( \sigma \) is a parameter of the kernel function. Due to its successful application for different problems, we choose to use the exponential kernel in what follows. Intuitively, it can be understood that this type of kernel provides some kind of adjacency measure between two PSFs, i.e. it measures the similarity between PSFs. Therefore, it is not surprising that the analysis of such a kernel over the training set is capable of capturing information regarding the relations between the PSFs in the training set. This property of the kernel function also explains the advantage of choosing the kernel function over choosing the mapping \( \varphi \), which is more difficult to obtain intuition about.

In the following, we briefly describe the KPCA method [33], [44], while closely following the notations in [44]. As in the linear PCA formulation, we assume to have a training set of \( N \) sampled PSFs \( \{h_1, h_2, \ldots, h_N\} \), each one containing \( n_i \) voxels. The symmetric positive definite kernel matrix is defined as \( K_{ij} = k(h_i, h_j) \). We define the mean map of the training dataset in feature space \( \bar{\varphi} = \frac{1}{N} \sum_{k=1}^{N} \varphi(h_k) \) and the centered maps of the dataset in feature space, as \( \tilde{\varphi}(h_k) = \varphi(h_k) - \bar{\varphi} \). Then, the centered kernel function is defined as \( \tilde{k}(h_i, h_j) = \langle \tilde{\varphi}(h_i), \tilde{\varphi}(h_j) \rangle \), and the centered kernel matrix as \( \tilde{K}_{ij} = \tilde{k}(h_i, h_j) \). It can be easily shown that the matrix \( K \) can be centered as follows: \( \tilde{K} = HKH \), where \( H = I_N - \frac{1}{N} 1_N 1_N^T \). \( I_N \) Denotes the \( N \times N \) unit matrix and \( 1_N \) is an \( N \times 1 \) column vector of ones.

---

3. Which is assumed to be sampled in the same scheme the training PSFs were sampled.
Eigenvalue decomposition is performed for $\tilde{K}$: $\tilde{K} = V \Lambda V^T$ where $V$ is a matrix whose columns are the eigenvectors of $\tilde{K}$ (the principal components) and $\Lambda$ is a diagonal matrix with the corresponding eigenvalues of $\tilde{K}$, denoted as $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ on its diagonal, in descending order. We denote by $v_k$ the $k$-th principal component of the kernel matrix.

We note that it follows from (11) that the kernel matrix $\tilde{K}$ is closely related to the covariance matrix in feature space, and it can be shown [57] that the kernel matrix $\tilde{K}$ is given in detail in [51], produces the following scheme for the calculation of the pre-image of $P_l(\varphi(h))$:

1. 
$$\mu_i = \sum_{k=1}^{l} v_{k,i} \hat{\varphi}(h) + \frac{1}{N} (1 - \sum_{j=1}^{N} \mu_j) \quad (17)$$

2. 
$$\frac{\|P_l(\varphi(h)) - \varphi(h_i)\|^2}{\sum_{i=1}^{N} \hat{\mu}_i \left(1 - \frac{1}{2} \|P_l(\varphi(h)) - \varphi(h_i)\|^2\right)} \approx h_{PI}$$

3. 
$$h_{PI} = \sum_{i=1}^{N} \hat{\mu}_i h_i \left(1 - \frac{1}{2} \|P_l(\varphi(h)) - \varphi(h_i)\|^2\right) \sum_{i=1}^{N} \hat{\mu}_i \left(1 - \frac{1}{2} \|P_l(\varphi(h)) - \varphi(h_i)\|^2\right) \quad (19)$$

We summarize the overall process of obtaining a pre-image of the feature space projection of an arbitrary input space test function using the KPCA method for an exponential kernel:

**Learning stage**
1. Calculate the kernel matrix $K_{ij} = k(h_i, h_j) = \exp \left(-\frac{\|h_i - h_j\|^2}{2\sigma^2}\right)$
2. Center the kernel matrix $\tilde{K} = HKH$
3. Preform eigenvalue decomposition $\tilde{K} = V \Lambda V^T$

**Projection pre-image**
1. Calculate the projection coefficients $\gamma_k$, by (13)
2. Calculate the coefficients $\mu_i$ according to (17)
3. Calculate $\|P_l(\varphi(h_i)) - \varphi(h_i)\|^2$ by (18)
4. Calculate the estimated pre-image $h_{PI}$ by (19)

We take the approximation $A(\cdot)$ in (7) simply to be the pre-image estimate $h_{PI}$, i.e.

$$A\left(h_{temp}\right) = h_{PI}$$

Finally, we would like to note that the approximation $A(\cdot)$ both for the PCA and KPCA methods, requires no more than a few matrix and vector products and thus the projection stage is very efficient.

### 5 Convergence properties

It is a well known result that each step of the alternating minimization scheme (4) by itself, converges to a global optimum [10]. Therefore, it is guaranteed that this iterative process yields a sequence of estimates with non-decreasing log-likelihood. In addition, other researchers provide convergence proofs for specific classes of priors, see e.g. [17].
However, since we use the formulation in (7), which includes both the residual denoising step and our prior term, there is no guarantee that the iterative scheme we propose brings (3) even to a local optimum.

In practice, we observed that the iterative scheme (7) produces sequences of estimates with non-decreasing likelihood, except for the first few iterations, in some datasets. Since the calculation of the log-likelihood requires additional computations at each iteration, we define the relative change of estimates at iteration $k$ as:

$$E_k = \frac{1}{2} \left( \frac{\| \hat{h}_k - \hat{h}_{k-1} \|}{\| \hat{h}_{k-1} \|} + \frac{\| \hat{f}_k - \hat{f}_{k-1} \|}{\| \hat{f}_{k-1} \|} \right)$$

and stop the iterative scheme when $E_k$ drops below a predetermined threshold. Typical plots of the log-likelihood and the relative change of estimates along the iterative process are displayed in Fig. 1.

### 6 APPLICATION FOR WIDE-FIELD FLUORESCENCE MICROSCOPY

#### 6.1 Background

Wide-field fluorescence microscopy (WFFM) is a modality used for imaging 3D biological specimens. The microscope objective lens is immersed in some oily medium, denoted as the immersion medium. A cover slide of the same refractive index as the immersion medium is mounted below the layer of oil, and covers the imaged specimen. The microscope optics are focused at a plane of certain depth within the imaged sample, denoted as the in focus plane, and produce a 2D image, of a section within the sample. By consecutively focusing the microscope at different depths within the sample, a collection of 2D sections is formed, overall creating a 3D image. The horizontal and vertical directions spanning the 2D in-focus planes are denoted as $x$ and $y$, respectively. The coordinate which varies along the optical axis of the microscope (the depth coordinate) is denoted as $z$. This process is named optical sectioning [42]. The detector array receives a significant amount of radiation originating from out of focus planes, in addition to the radiation coming from the in focus plane. The latter phenomenon results in significant blurring of the acquired images and thus deteriorates image quality. In addition, WFFM imagery is usually contaminated by photon noise.

In many imaging systems, the PSF is determined solely by the properties of the imaging system. Therefore, under this assumption knowledge of the imaging system implies exact knowledge of the PSF, or at least a good approximation of it. In microscopy, however, the imaged sample itself influences the PSF and therefore exact a-priori knowledge of the PSF is not available, hence the necessity for blind deconvolution algorithms. Moreover, as we will see next, the WFFM PSF possesses some properties which make the restoration process especially difficult.

#### 6.2 PSF model

Several analytical models for the WFFM PSF have been proposed and validated [3], [24], [25], [27], [62], [63]. A comprehensive discussion of the WFFM optics is out of the scope of this work, and the interested reader is referred to the above references and the references therein. However, a PSF model was required for the proposed BD algorithm for the cause of simulating the training set of PSFs. The PSF model used for this work follows [62],[2]:

$$h(r, z) = I_0 \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{pupil}(k_r) \cdot \text{defocus}(k_r, z) \cdot \text{depth}(k_r) \cdot e^{-2\pi i (k_x x + k_y y)} dk_x dk_y \right)^2$$

$$\text{pupil}(k_r) = \begin{cases} 1, & k_r \leq \frac{NA}{\lambda_0} \\ 0, & \text{else} \end{cases}$$

$$\text{defocus}(k_r, z) = \exp \left\{ i \cdot k_0 \cdot z \cdot \cos (\theta_0(k_r)) \right\}$$

$$\text{depth}(k_r) = \exp \left\{ i \cdot k_0 \cdot d \cdot |n_p \cos (\theta_p(k_r)) - n_0 \cos (\theta_0(k_r))| \right\}$$

Where $x, y, z$ are spatial Cartesian coordinates, $k_r = \sqrt{k_x^2 + k_y^2}$, $r = \sqrt{x^2 + y^2}$, $\theta_0(k_r) = \arcsin \left( \frac{\lambda_0}{n_0 k_r} \right)$, $\theta_p(k_r) = \arcsin \left( \frac{\lambda_p}{n_p k_r} \right)$, $k_0 = \frac{2\pi n_0}{\lambda_0}$ is the wave frequency in the medium, $\lambda_0$ is the light wavelength, $NA$ is the numerical aperture of the microscope, $n_0$ is the refractive index of the immersion medium, $n_p$ is the refractive index of the imaged sample, $d$ is the depth of the in-focus plane below the cover glass and $I_0$ is a scaling factor that was chosen so that $h$ will integrate to unity.

One important property of the WFFM PSF is that it is radially symmetric in the $x$-$y$ plane:

$$h(x, y, z) = h_{rad}(\sqrt{x^2 + y^2}, z) = h_{rad}(r, z)$$

The formulation in (21) also reveals the fact that in WFFM, the degradation model is shift variant, as it is dependent upon the depth of the in-focus plane within the image.

4. This layer of oil enhances the optical properties of the microscope.

5. This is a good assumption since in-plane aberrations are practically fully corrected for within the field imaged by the microscope, in all high quality objective lenses.
immersion medium. However, the convolutive model is a very good approximation for the degradation induced by the microscope, and is often assumed in literature. By adopting this assumption, \( d \) becomes the average depth of the imaged sample below the cover slide. Examples for various simulated PSFs can be observed in Fig. 2.

6.3 Depth aberrations

The WFFM PSF is dependent on a few parameters, most of which relate to the imaging system itself. However, the differences between the refractive indices of the sample and the immersion oil affect the PSF as well. This effect manifests in the form of asymmetry of the PSF along the \( z \) direction, as can be observed in Fig. 2. We term depth aberration the spherical aberration caused by imaging into a specific depth \( d \) below the cover slide for live biological specimens with a refractive index different than the immersion oil. Since the depth aberrations are governed by parameters related to the imaged sample itself, they are the root cause for the inability of obtaining exact a-priori knowledge of the PSF. In practice, all of the PSF parameters, with the exception of the sample depth \( d \) and the refractive index of the sample \( n_p \) are usually known. In the sequel, we will show how the known parameters can be utilized, and the unknown depth aberration can be estimated using our proposed method.

The WFFM PSF does not have a compact spatial domain support (see Fig. 2), which essentially means that many PSF voxels are required to be estimated, thus increasing the number of degrees of freedom within the system. In addition, the optimization over the PSF cannot be easily carried out in parametric form, due to its complexity.

An approach that has the potential of overcoming those difficulties is to approximate the PSF with simple functions. Such an approach is proposed in [48] for blind deconvolution of confocal fluorescence microscopy images, where the PSF is approximated by a Gaussian function. However, it has been previously shown that in 3D WFFM the PSF cannot be well approximated by Gaussians [69], even in the case of no depth aberrations.

From all the above we conclude that first, in WFFM it is essential to use blind deconvolution methods, since the PSF is affected by the imaged sample itself and second, that using example based learning for studying the class of PSFs seems to be advantageous in WFFM, since the PSF cannot be easily and accurately parametrized.

7 Numerical Experiments

In order to validate the proposed BD scheme, we tested it on five datasets. The first two datasets were created by simulating a \( 128 \times 128 \times 128 \) synthetic image depicting three tori of different sizes and intensities (Fig. 6(a)). The synthetic image was blurred by two different PSFs, one free of depth aberrations and one containing depth aberrations (\( d = \mu m \)). Then, each blurred image was scaled to a maximum intensity value of 1000, and corrupted by Poisson noise. In addition, three images of biological specimens were acquired by a WFFM. Two of the acquired images were known to have depth aberrations and one was known to be free of depth aberrations, or to have very little depth aberration.

The original images were of size \( 1024 \times 1024 \) in the \( x - y \) plane and contained 42 to 46 sections in the \( z \) direction. The pixel size is 0.0063\( \mu m \) in the \( x - y \) plane and 0.2\( \mu m \) along the \( z \) axis. In order to facilitate the processing of such large datasets, a \( 256 \times 256 \) section was cropped from each image. The images were not cropped in the \( z \) direction.

Each of the images was processed using three different methods: BD using (7) with the PCA and KPCA priors, and the AIDA algorithm, depicted in [30]. In addition, the synthetic images were processed by non-blind deconvolution using (5), excluding the last step, with the true PSF kept fixed.

7. The scaling of the blurred image determines the noise magnitude.
8. The degree of depth aberrations was controlled in this experiment by replacing the objective immersion oil with media of different refractive indices.
7.1 Implementation details

In the following experiments, it was assumed that the microscope parameters are known, and the only unknown parameter is the sample depth $d$. As previously discussed in section 6.3, in practice neither $n_p$ nor $d$ are exactly known, but since they both control the depth aberrations in a similar manner, for the following experiments it is sufficient to modify only one of them. This is a practical assumption and we stress that it does not reduce the generality of the proposed method.

In general, it is possible to create the training set of PSFs by imaging small fluorescent beads, see e.g. [30], [56]. However, since it is beneficial for our method to use a large number of PSFs for the training set, and the acquisition of a large number of samples under varying acquisition conditions is a tedious and cumbersome task, we used simulated PSFs for our experiments. The training set contained a total of 201 PSFs which were simulated according to the formulation (21) with $d$ varying from -4.6$\mu$m to 4.6$\mu$m, which covers the range of expected depth aberrations. A sample of the training set PSFs is displayed in Fig. 2, along with their best approximations by PCA and KPCA. As a measure of approximation error we calculated the root mean squared error (RMSE). We define the RMSE as

$$\text{RMSE}(f, \hat{f}) = \left( \frac{1}{|\Omega|} \sum_{x \in \Omega} (f(x) - c \cdot \hat{f}(x))^2 \right)^{\frac{1}{2}} \quad (23)$$

where $|\Omega|$ is the number of voxels in the image and $c = \frac{\sum_{x \in \Omega} f(x)}{\sum_{x \in \Omega} \hat{f}(x)}$ accounts for different scaling between the images. It can be observed both visually and by examining the RMSE values, that the PSF-space approximation error is low for both PCA and KPCA methods. It can also be seen in Fig. 2 that the KPCA method provides smoother approximations than PCA, with higher RMSE values. This is due to the choice of the KPCA parameter $\sigma$, which is set to a value sufficiently large for providing good approximations for PSFs which do not belong to the training set. The influence of $\sigma$ is further discussed in the following.

In addition, since the implementation of the learning stage depicted in section 4.3 requires the storage of the entire training set, in general it requires a very large amount of memory space. However, in WFFM, we can take advantage of our prior knowledge of the PSF radial symmetry, and perform the entire statistical learning stage on a 2D half section of the $r-z$ plane, thus significantly reducing memory requirements by an order of magnitude. Exploiting the PSF radial symmetry also facilitates the diagonalization of the covariance matrix in the PCA case, which would be otherwise infeasible.

Another implementation issue is the choice of parameters, namely $\eta$ and $\sigma$. The regularization parameter $\eta$ controls the degree in which the PSF estimates are drawn to the PSF-space. It can be understood from (6) that $\eta = 1$ implies a hard constraint, where the PSF is projected onto PSF-space at the end of each iteration. Since the MLEM iterative process is very sensitive to local maxima, we found this approach not to be useful. During our experimentation, it was empirically found that many PSFs in the training set correspond to local optima of the iterative process. At the beginning of the iterative process, the PSF estimates might be far from the optimal solution. The hard constraint forces all PSF estimates to reside within the learned PSF-space, and might cause the iterative process to be attracted to a local maximum. We demonstrate this property of the deconvolution process in Fig. 3. Non-blind deconvolution was carried out on the synthetic image shown in Fig. 6(g), using 45 different PSFs with $d$ varying from -4$\mu$m to 4$\mu$m. The log-likelihood values at convergence were recorded and displayed against the value of $d$. It can be observed that although the global maximum is obtained very close to the true solution $d = 1\mu$m, a number of other local maxima exist as well. Setting $\eta < 1$ allows the PSF estimates to approach this global maximum on a different path, which could potentially avoid the local maxima. On the other hand, setting $\eta$ to a very small value might lead to insufficient regularization. We found the adequate values of $\eta$ to reside between 0.01 and 0.1, and that the results do not vary significantly when $\eta$ is set to a value within this range.

The KPCA parameter $\sigma$ controls the kernel width, and ultimately determines the degree of mixing allowed between the training set PSFs when creating the PSF-space approximation $\mathcal{A}(\hat{h})$ [13], [14]. As the value of $\sigma$ grows, more training samples are weighted into the PSF approximation, causing the PSF approximation to become blurry, and less plausible. Contrary to that, setting $\sigma$ to extremely low values causes the regularization scheme to degenerate to a nearest-neighbor search within the training set. This is usually not desirable, as the true PSF is not expected to reside within the training set. We suggest using the values of the distances between the training set samples $\|h_i - h_j\|$ as a guide to the value of $\sigma$.

All convolution operations were carried out using the fast Fourier transform (FFT). Running on the $128 \times 128 \times 128$ dataset using the KPCA regularization, a single iteration (requiring three convolution operations) took approximately 14 seconds on an Intel Q6600 2.4 GHz processor (utilizing a single core), and exhibited peak memory usage of 750 Mb. In our experiments, the algorithm typically converged in 500-600 iterations, yielding a total processing time of around two hours. Out of the total processing time, the PSF regularization code took about 20%. The FFT code took most of the remaining processing time (~70% of total processing time).

9. It can be easily shown that this choice of the scaling factor $c$ minimizes the mean square error.
The images were processed according to (7), with the PSF regularization implemented according to (6) with $\eta = 0.05$, for both PCA and KPCA methods. In the KPCA case, $\sigma$ was set to the 0.2 quantile of the distances $\|h_i - h_j\|$ within the training set. The value of $l$, which is the number of principal components used, was set to 10 for both the PCA and the KPCA methods. These values were determined by observing the eigen-spectrum of the covariance matrix in the PCA case, and the centered kernel matrix in the KPCA case, as can be observed in Fig. 4. The PCA eigenvalues corresponding to the 10 largest magnitude eigenvalues are displayed in Fig. 5.

We note that the observed reconstruction results were found not to be highly dependent on the value of $l$ and in practice, any number between 5 and 30 performed well, for both methods. The algorithm was initialized with the acquired image as the object estimate and a symmetrical PSF support as the PSF estimate (see Figs. 7-9).

As previously mentioned, the AIDA algorithm, which we compare our results to, uses a PSF prior given by the user. In the following experiments, we supplied AIDA with a symmetrical PSF as the prior.

As a measure of fidelity to the ground truth image, we calculated the RMSE between the object estimates and the original image according to (23). We note that the maximal intensity value in the ground truth image is 1000. In addition, the RMSE between the estimated PSFs and the true PSFs was calculated.

Results for the synthetic object are shown in Fig. 6.

---

**Fig. 4:** Selection of the number of used eigenvalues for PCA and KPCA methods. (a) Eigen-spectrum of covariance matrix. (b) Eigen-spectrum of centered kernel matrix.

**Fig. 5:** The PCA eigenvectors corresponding to the ten largest magnitude eigenvalues. The eigenvectors are displayed in descending order of the corresponding eigenvalues, from top left to bottom right.

**7.2 Results**

The images were processed according to (7), with the PSF regularization implemented according to (6) with $\eta = 0.05$, for both PCA and KPCA methods. In the KPCA case, $\sigma$ was set to the 0.2 quantile of the distances $\|h_i - h_j\|$ within the training set. The value of $l$, which is the number of principal components used, was set to 10 for both the PCA and the KPCA methods. These values were determined by observing the eigen-spectrum of the covariance matrix in the PCA case, and the centered kernel matrix in the KPCA case, as can be observed in Fig. 4. The PCA eigenvalues corresponding to the 10 largest magnitude eigenvalues are displayed in Fig. 5.

We note that the observed reconstruction results were found not to be highly dependent on the value of $l$ and in practice, any number between 5 and 30 performed well,
Results for the real world depth aberrated images are shown in Fig. 7 and Fig. 8. Results for the real world aberration free image are shown in Fig. 9.

It can be seen that the proposed algorithm significantly enhances image resolution, as many new details can be resolved in the reconstructed images. In addition, the reconstructed images seem free of noise. We further observe that the proposed algorithm succeeds in estimating the depth aberrations of the PSF, as the PSFs estimated by the algorithm are indeed asymmetric w.r.t the z axis in the images acquired in the setting which contained the aberrations (Figs. 7 and 8), and symmetric for the aberration free setting (Fig. 9). We note that once again, AIDA fails to detect the depth aberrations. Finally, we note that the results obtained using the KPCA regularization method outperform the ones obtained by PCA, which seems to over estimate the depth aberrations.

The superiority of the KPCA method is expected, since it is able to extract non-linear structures in the training set of PSFs, and is not limited by the vector space assumption discussed in section 4.3.1. Another noteworthy detail is the repeatability of the proposed BD method to yield consistent results. This can be observed when comparing Fig. 7 and 8, which show that similar PSF estimates are obtained when processing different images acquired under similar conditions.

**8 Conclusion and further research**

In this work, we presented a new method for blind image deconvolution. We have shown that this method is capable of reconstructing synthetic and real world acquired WFFM thick specimen images, and produce reconstruction results of high resolution and low noise levels. In addition, the proposed BD method is capable of reliably providing accurate PSF estimates. This property of the algorithm is demonstrated by overcoming the depth aberration problem in WFFM.

The proposed method compares favorably to a recent BD algorithm [30], especially designed for the reconstruction of 3D microscopic imagery.

The proposed algorithm generalizes a scheme for integrated denoising within the iterative process, which was originally proposed for non blind deconvolution. We show the potency of this method for BD.

Moreover, we propose a novel method for PSF regularization in BD. This regularization scheme uses machine learning methods, which extract the statistical properties of a family of PSFs by example based learning. This prior is integrated into the iterative process using a simple and efficient formulation and is shown to produce excellent reconstruction results for real world data.
We propose further research directions in related topics. One interesting research path would be exploring other learning methods for the creation of PSF priors. One example for such recent learning method is diffusion maps [9], which is a state of the art method for extracting structures within a given dataset. This method has been successfully employed for introducing shape priors for segmentation using active contours [19], [20], and is a natural extension to the method proposed in this work. Another research direction is the application of the proposed algorithm to BD in different fields, other than microscopy, such as astronomical imaging, medical imaging, optics and photography applications.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the Israeli Ministry of Science and Technology.

REFERENCES


Tal Kenig received his B.Sc. from the Biomedical Engineering faculty at the Technion in 2005 and his M.Sc. from the faculty of Electrical Engineering at the Technion in 2009. From 2004 to 2007 he has been with General Electric Healthcare, Functional Imaging, working on the development of medical image processing and analysis algorithms. He is currently with Mediscan Vision Brain Technologies, where he is a senior algorithm developer, in the field of medical imaging. His research interests include medical and biological image processing and analysis, computer vision, inverse problems and machine learning.

Zvi Kam received his Ph.D. in solid state physics in 1971 from the Technion, Israel Institute of Technology. He continued his postdoc in biophysics at UCSF, and in 1975 he joined the Weizmann Institute of Science, Rehovot, Israel, where he is a Professor of Biophysics at the department of Molecular Cell Biology. He is applying light microscopy to cell biology. Recent work is focused on cell-based screens as a way to probe complex multi-molecular mechanisms. To better relate microscope imaging to cellular functions high-resolution automated screening microscope was built, quantitative image analysis pipeline for high-resolution imaging. His current research interests include resolution enhancement of digital images and videos, 3D video and multi-view data, sampling and combined representations of signals and images, and adaptive systems in signal processing and control.

Arie Feuer has been with the Electrical Engineering department at the Technion - Israel Institute of Technology, since 1983 where he is currently a chaired professor and head of the control and robotics lab. He received his B.Sc. and M.Sc. from the Technion in 1967 and 1973 respectively and his Ph.D. from Yale University in 1978. From 1967 to 1970, he was in industry working on automation design and between 1978 and 1983 with Bell Labs in Holmdel, N.J. Between the years 1994 and 2002 he served as the president of the Israel Association of Automatic Control and was a member of the IFAC Council during the years 2002-2005. Arie Feuer is a Fellow of the IEEE. In the last 20 years he has been regularly visiting the EE&S department at the University of Newcastle. His current research interests include resolution enhancement of digital images and videos, 3D video and multi-view data, sampling and combined representations of signals and images, and adaptive systems in signal processing and control.