Journal of Microscopy, Vol. 217, Pt 1 January 2005, pp. 93–108
Received 5 September 2003; accepted 28 October 2004

Noise suppression of point spread functions and its influence on deconvolution of three-dimensional fluorescence microscopy image sets

X. LAI*†‡, ZHIPING LIN†, E. S. WARD* & R. J. OBER*‡
**Center for Immunology, University of Texas Southwestern Medical Center, Dallas, TX 75390, U.S.A.*
†School of Electrical and Electronic Engineering, Nanyang Technological University, Block S2, Nanyang Avenue, Singapore 639798, Republic of Singapore
‡Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75083, U.S.A.

Key words. Deconvolution, fluorescent microscopy, image restoration, noise suppression, point spread function.

Summary

The point spread function (PSF) is of central importance in the image restoration of three-dimensional image sets acquired by an epifluorescent microscope. Even though it is well known that an experimental PSF is typically more accurate than a theoretical one, the noise content of the experimental PSF is often an obstacle to its use in deconvolution algorithms. In this paper we apply a recently introduced noise suppression method to achieve an effective noise reduction in experimental PSFs. We show with both simulated and experimental three-dimensional image sets that a PSF that is smoothed with this method leads to a significant improvement in the performance of deconvolution algorithms, such as the regularized least-squares algorithm and the accelerated Richardson–Lucy algorithm.

1. Introduction

Deconvolution of three-dimensional (3D) microscopy image sets is important for removing undesired blurring due to out-of-focus fluorescence, which can obscure details of the in-focus image. The point spread function (PSF) is of central importance in any deconvolution algorithm. In blind deconvolution algorithms the PSF is estimated as part of the algorithm whereas in non-blind deconvolution algorithms the PSF is assumed to be known. In non-blind algorithms either an experimentally acquired PSF or a theoretically calculated PSF is used. It is well known that a theoretical PSF is often an imprecise model for the actual PSF (Gibson & Lanni, 1991; McNally *et al.*, 1994). Because it is known that an inaccurate PSF can affect the performance of deconvolution algorithms (Preza *et al.*, 1992) the experimental PSF would appear to be the preferred choice of a PSF model for a deconvolution algorithm. However, experimental PSFs typically exhibit strong randomness due to the Poisson nature of the photon counting process and other noise sources in the acquisition electronics (Castleman, 1996; Inoue & Spring, 1997). These effects are serious obstacles to the use of experimental PSFs in deconvolution algorithms, because it is known that a noisy PSF can compromise the results of deconvolution algorithms.

It therefore appears that a promising approach is to suppress noise or random components in an experimental PSF and to use a smoothed PSF in deconvolution. As a standard method, a Gaussian filter is often used to smooth 3D image sets, including PSFs (Chen *et al.*, 1995; van Kempen & van Vliet, 1997). Because the approach is based on a weighted average of neighboring pixels it typically results in the loss of sharp details in the PSF. Another approach to suppress noise is to average the acquired data from 20–30 beads and then to average the resulting PSF cylindrically about the optical axis (Shaw & Rawlins, 1991; Shaw, 1993). The problems of this approach include that it is time consuming, that the alignment of the different PSFs for averaging is not without problems and that the radial averaging removes any potential asymmetry in the PSF.

In Ober *et al.* (2005) we have introduced an algorithm that is capable of effectively suppressing noise in 3D microscopy image sets. Here we apply this algorithm to smooth PSFs and compare it with the use of a Gaussian filter. The smoothed PSFs are then used in deconvolution algorithms: the regularized least-squares algorithm (see e.g. Jain, 1989), and the accelerated Richardson–Lucy algorithm (Biggs & Andrews, 1997). The performance of the deconvolution algorithms using noisy PSFs and smoothed PSFs is compared. We observe

Correspondence to: Raimund J. Ober. Fax: +1 214 648 1259; e-mail: ober@utdallas.edu.

that the recovered object is significantly enhanced by using a smoothed PSF.

The rest of the paper is organized as follows. In Section 2 we discuss materials and methods. Noise suppression of experimental PSFs and PSFs with simulated random components is investigated in Section 3. The influence of noise suppression on the deconvolution algorithms is discussed in Section 4. We draw conclusions in Section 5.

2. Materials and methods

2.1. Microscope system, image acquisition and image processing

Sample specimens were analysed using a Zeiss Axiovert 200M inverted microscope, equipped with a Mercury (Hg) arc lamp and a Plan-Aprochromat 100× oil objective (NA = 1.4). Images were acquired at 100-nm increments using the inbuilt *z*-focus system. The following filter sets were used, exciter: HQ470/40, emitter: HQ525/50, beam splitter: Q495IP (Chroma Technology Corp. Brattleboro, VT, U.S.A.).

The acquisition system was a high-resolution Hamamatsu Orca-100 cooled 12-bit CCD camera, (Hamamatsu Photonics K.K., Japan) whose pixel size is 6.7 μm × 6.7 μm. Based on the manufacturer's specification, the conversion factor of the CCD camera is 3.2 electrons per analog to 12-bit digital unit (*ADU*) for a 12-bit scale. Images were processed in double precision with customized software written in the Matlab programming language. The displays of all images are on a 16-bit scale. Therefore, data from the 12-bit camera are multiplied by 16 for display purposes. Based on the camera specifications this implies that one increment in the display units corresponds to 0.2 (= 3.2/16) photons.

2.2. Experimental PSFs

PSFs were acquired by imaging (yellow:green) 50-nm-diameter fluorescent microspheres (Fluoresbrite YG Microspheres, Polysciences, Inc. Warrington, PA, U.S.A.) and 20-nm diameter (yellow:green) fluorescent microspheres (Molecular Probes, Eugene, OR, U.S.A.). A drop of the diluted liquid of microspheres was placed in the middle of an air-cleaned cover-slip (No. 1.5) pretreated with poly-L-lysine (Sigma Chemical Co., St. Louis, MO, U.S.A.). After drying, the cover-slip was mounted in the middle of a slide with a small drop of Prolong (P7481, Molecular Probes). Both cropped image sets P_{exp} and P_{exp200} of the 50- and 200-nm-diameter beads, respectively, consist of 43 images acquired at 100-nm increments with each image being a 128 × 128-pixel array.

2.3. Description of the algorithm and notation

We give here a brief description of the algorithm that will be used to reduce the noise level of PSFs (see Appendix and Ober *et al.*, 2005, for details). The algorithm consists of two parts. In the first part a 3D discrete image set $P(k_1, k_2, k_3)$, $k_i = 1, 2, \ldots, N_i$, $i = 1, 2, 3$, is decomposed into three 1D sequences $P_i(k_i)$, $k_i = 1, 2, \ldots, N_i$, $i = 1, 2, 3$, i.e.

$$P(k_1, k_2, k_3) = P_1(k_1)P_2(k_2)P_3(k_3), \quad k_i = 1, 2, 3, \ldots, i = 1, 2, 3,$$

where P_i, $i = 1, 2, 3$, are vector/matrix-valued sequences such that $P_1(k_1) \in \mathbb{R}^{1\times m_1}$, $P_2(k_2) \in \mathbb{R}^{m_1\times m_2}$, $P_3(k_3) \in \mathbb{R}^{m_2\times 1}$ for some integer values m_1, m_2 that are determined as part of the algorithm. Note that we use discrete parameters to denote the fact that we are dealing with a pixelated image. In the second part a parameter set (A_i, B_i, C_i) is obtained for each sequence P_i such that

$$P_i(k_i) = C_i A_i^{k_i-1} B_i, \quad k_i = 1, 2, 3, \ldots, \quad i = 1, 2, 3.$$

Therefore, the 3D image set P can be represented as

$$P(k_1, k_2, k_3) = C_1 A_1^{k_1-1} B_1 C_2 A_2^{k_2-1} B_2 C_3 A_3^{k_3-1} B_3, \quad k_i = 1, 2, 3, \ldots, \quad i = 1, 2, 3$$

with $C_1 \in \mathbb{R}^{1\times n_1}$, $A_1 \in \mathbb{R}^{n_1\times n_1}$, $B_1 \in \mathbb{R}^{n_1\times m_1}$, $C_2 \in \mathbb{R}^{m_1\times n_2}$, $A_2 \in \mathbb{R}^{n_2\times n_2}$, $B_2 \in \mathbb{R}^{n_2\times m_2}$, $C_3 \in \mathbb{R}^{m_2\times n_3}$, $A_3 \in \mathbb{R}^{n_3\times n_3}$, $B_3 \in \mathbb{R}^{n_3\times 1}$. The integer values n_1, n_2 and n_3 are again determined as part of the algorithm. Note that the identification of the middle index of the pixel array with the optical axis has proved to be numerically favourable in our algorithm (Ober *et al.*, 2005). Therefore, in all the data presented in this paper the middle index k_2 of an image set is assumed to correspond to the optical axis, i.e. the *z* axis.

The key methods used in the algorithm to suppress noise are approximations that are based on the singular value decomposition. Two singular value decompositions (one in step 3, one in step 5) are used to decompose the 3D image set into three 1D sequences while another three singular value decompositions (in step 6) are used to calculate the parameter sets for the three 1D sequences.

In each singular value decomposition, we will obtain a series of singular values. The corresponding approximation is essentially based on splitting the set of singular values in two subsets. One, typically the larger values, that will be 'retained' and two, typically the smaller values, that will be 'discarded' to provide the approximation. The number of retained singular values in the algorithm determines the accuracy of the approximation. As the larger a singular value is the more important it is for the approximation and vice versa, only the large, i.e. dominant, singular values are important for obtaining an accurate approximation. In addition, due to the presence of noise in the 3D image set, small singular values are corrupted by noise. By excluding those singular values, we effectively suppress noise and obtain a smoothed approximation of the noisy image set.

Let r_1, r_2 denote the numbers of the dropped singular values in the first two singular value decompositions used to decompose the image set P and let s_1, s_2, s_3 be the numbers of the discarded singular values in the other three singular value decompositions used to calculate the parameter sets. The

smoothed estimate of the 3D image set P obtained is then denoted as P^{rs}, $rs = \{r_1, r_2; s_1, s_2, s_3\}$. Note that r_1, r_2, s_1, s_2 and s_3 are typically experimentally determined values (see Section 3.1 for more details).

This algorithm was applied to the experimentally acquired PSF P_{exp} and P_{exp200} to obtain the smoothed PSFs $P_{exp}^{rs_1}$ and $P_{exp200}^{rs_2}$, respectively.

2.4. PSFs with simulated random components

To assess the effectiveness of our algorithm to suppress random and noise components we simulated PSFs as samples of spatial Poisson processes, whose intensities were given by known deterministic PSFs. The noise suppression algorithm was then applied to these simulated PSFs and the resulting smoothed estimates were compared with the deterministic PSFs. This was done for two different deterministic PSFs. In the first case we used the smoothed PSF $P_{exp}^{rs_1}$ (Section 2.3) that was obtained by smoothing the experimental PSF P_{exp}. In the second case we used a theoretical PSF, which was generated as described below.

In fact for the first case several different experimental PSFs were simulated to model different levels of randomness. The image set P_n^λ of the simulated PSF was obtained by first defining

$$\mu_P^\lambda(k_1, k_2, k_3) = \lambda(cP_{exp}^{rs_1}(k_1, k_2, k_3) - cb_{offset}),$$
$$k_1 = 1, \ldots, 128, k_2 = 1, \ldots, 43, k_3 = 1, \ldots, 128,$$

where the conversion factor c was set to be $c = 0.2$ (Section 2.1). The factor λ was varied to control the level of the randomness (Section 3.2), with $\lambda = 1, 3, 6, 9$. The offset b_{offset} for the simulation was set to be $b_{offset} = 3050$, which is the offset measured for our camera system. The photon count number $C_P^\lambda(k_1, k_2, k_3)$ of the (k_1, k_2, k_3) pixel was simulated as a Poisson random variable with mean $\mu_P^\lambda(k_1, k_2, k_3)$. Finally, the image set for the PSF with simulated random components was obtained as $P_n^\lambda(k_1, k_2, k_3) = C_P(k_1, k_2, k_3)/(c\lambda) + b_{offset}$, $k_1 = 1, \ldots, 128$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$. Hence, we consider $P_d^\lambda = \mu_P^\lambda/(c\lambda) + b_{offset}$ as the deterministic version of the PSF P_n^λ. Note that for all λ we have that $P_d^\lambda = P_{exp}^{rs_1}$. By offset we mean the signal level that is acquired by the CCD camera with closed shutter and negligible exposure time. This is in contrast to the background of an image that also includes scattered photons, autofluorescence from the sample, etc. The background level b_P for the experimental PSF is $b_P = 3350$.

In the second case, the PSF P_{dt} based on Li and Wolf's model (Li & Wolf, 1984) was generated. All photons falling in the area of one pixel were summed together to mimic the integration over a CCD well. The randomness was introduced in the same way as discussed above for the parameter $\lambda = 1$ to yield the simulated PSF P_{nt} with the simulated random components. The simulated PSF was scaled so that in the focal plane the integral matched that of the experimental PSF, adjusted for the background level.

The smoothing algorithm (Section 2.3) was applied to the simulated PSF P_n^λ and P_{nt} to obtain the smoothed PSFs P_n^{λ, rs_e} and $P_{nt}^{rs_t}$, respectively.

2.5. Biological samples

Jurkat cells transfected with an FcRn–green fluorescent protein (GFP) plasmid have been described previously (Ober *et al.*, 2001). Cells were grown in complete RPM1 medium containing 1 mg mL^{-1} geneticin. For microscopy, cells were washed twice by centrifugation for 5 min (5 °C and 1500 rpm) with cold Dulbeco's phosphate-buffered saline (DPBS) (BioWhittaker, Walkersville, MD, U.S.A.). Cells were fixed for 15 min with 3% paraformaldehyde (PFA) (Electron Microscopy Sciences, Fort Washington, PA, U.S.A.). The cells were washed twice with DPBS and allowed to settle on polylysine-treated cover-slips (No. 1.5) (Fisher Scientific, Allentown, PA, U.S.A.). Cells were mounted on slides using Prolong (P7481, Molecular Probes).

2.6. Simulated image sets

A simulated object O_s consisting of four balls (one of 300 nm radius and three of 200 nm radius) was generated using an analytical description of the ball in the frequency domain (van Vliet, 1993):

$$H_3(w_1, w_2, w_3) = \frac{-2\pi R\sqrt{w_1^2 + w_2^2 + w_3^2}\cos(2\pi R\sqrt{w_1^2 + w_2^2 + w_3^2}) + \sin(2\pi R\sqrt{w_1^2 + w_2^2 + w_3^2})}{\pi^2\sqrt{w_1^2 + w_2^2 + w_3^2}^3},$$

$w_1, w_2, w_3 \in \mathbb{R}$, where R is the radius of the ball. To avoid Gibbs effects the data array of the object was convolved with a 3D Gaussian filter with standard deviation of 1. The image set of the simulated object consists of 80 2D frames (images) each being a 148×148-pixel array. The sampling distances of the simulated data in the object space were assumed to be 67 nm along the lateral axes and 100 nm along the optical axis. Following the convention described in Section 2.3 the second index k_2 of O_s corresponds to the optical axis, i.e. the z axis. Therefore, the data array O_s has $148 \times 80 \times 148$ data points.

In the noise-free case the ideal image set Im_{id} was obtained as the sum of the background b_{Im} of 3350 and the result of the convolution between the data array O_s and the PSF P_{id}, i.e. $Im_{id} = O_s * P_{id} + b_{Im}$, where $*$ denotes the convolution and the PSF P_{id} was obtained as $P_{id}(k_1,k_2,k_3) = (P_{exp}^{rs_1}(k_1,k_2,k_3) - b_P)/\sum_{k_1=1}^{128}\sum_{k_2=1}^{43}\sum_{k_3=1}^{128}(P_{exp}^{rs_1}(k_1,k_2,k_3) - b_P)$, $k_1 = 1, \ldots, 128$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$ with $b_P = 3350$. A simulated image set was obtained by first defining $\mu_{Im}(k_1, k_2, k_3) = c(Im_{id}(k_1, k_2, k_3) - b_{offset})$, $k_1 = 1, \ldots, 148$, $k_2 = 1, \ldots, 80$, $k_3 = 1, \ldots, 148$, where the offset b_{offset} was set to be 3050 counts. Then the photon count number $C_{Im}(k_1, k_2, k_3)$ of the (k_1, k_2, k_3) pixel was simulated as a Poisson random variable with mean $\mu_{Im}(k_1, k_2, k_3)$. Finally, the simulated image set was

obtained as $Im_n(k_1, k_2, k_3) = C_{Im}(k_1, k_2, k_3)/c + b_{offset}$, $k_1 = 1, \ldots, 148, k_2 = 1, \ldots, 80, k_3 = 1, \ldots, 148$.

2.7. Image restoration algorithms

Two image deconvolution algorithms were used: the accelerated Richardson–Lucy algorithm and the regularized least-squares algorithm. To use both algorithms estimated background levels were subtracted from the PSFs. For the accelerated Richardson–Lucy algorithm, negative pixel values in the resulting PSFs were replaced by zeros. The PSFs were then normalized such that the sum of all data points in each resulting PSF is one. For the accelerated Richardson–Lucy algorithm initial estimates in the iteration were taken as the image sets with background estimate subtracted.

3. Noise suppression of PSFs

In this section we apply the smoothing algorithm (Section 2.3 and Appendix) to reduce noise and randomness levels in experimental and simulated PSFs.

3.1. Smoothing of an experimental point spread function

To demonstrate the use of the proposed algorithm the acquired PSF P_{exp} (see Section 2.2) was smoothed. Figure 1 shows

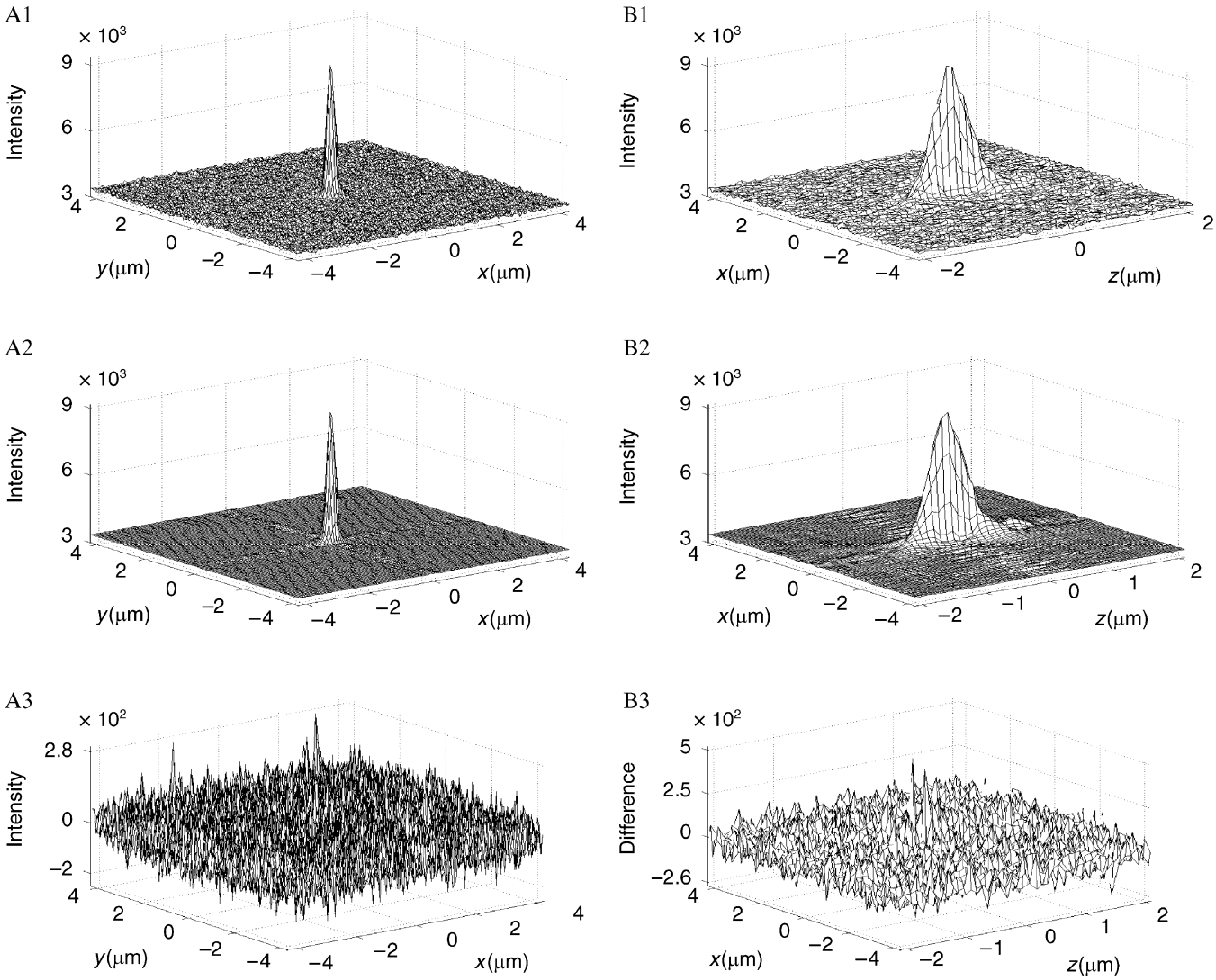

Fig. 1. Comparison of the experimental PSF P_{exp} (A1, B1) and its smoothed estimate $P_{exp}^{rs_1}$, $rs_1 = \{120, 120; 49, 72, 49\}$ (A2, B2) calculated via the noise suppression algorithm (Appendix). A1, A2 and A3 show cross-sections in the x–y plane at the focal level, the 22nd frame ($P_{exp}(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$) of the experimental PSF, the 22nd frame ($P_{exp}^{rs_1}(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$) of the estimate and their difference, respectively. B1, B2 and B3 show cross-sections in the x–z plane, the cross-section $P_{exp}(64, k_2, k_3)$, $k_2 = 1, \ldots, 43, k_3 = 1, \ldots, 128$, the cross-section $P_{exp}^{rs_1}(64, k_2, k_3)$, $k_2 = 1, \ldots, 43, k_3 = 1, \ldots, 128$, and their difference, respectively. The coordinate system shown in the plots is such that the y-axis (x-axis, z-axis/optical axis) coincides with the first (third, second) component of the PSFs P_{exp} and $P_{exp}^{rs_1}$, respectively.

cross-sections of the acquired PSF P_{exp}. As described in Section 2.3, we smooth the PSF P_{exp} by excluding small singular values of the five singular value decompositions used in the algorithm. The key question is how to determine the number of the retained singular values in each singular value decomposition in order to generate an estimate of the experimental PSF that is smooth enough for its use in, for example, deconvolution algorithms, but still exhibits the important features of the PSF.

As an example we will show how to determine the number of the retained singular values in the first singular value decomposition in detail (see Step 3 in the Algorithm). The same approach can be used to determine the numbers of the retained singular values of the other singular value decompositions. Figure 2.A1 shows the singular values of the first singular value decomposition used in the algorithm. We observe that the sizes of the singular values vary significantly. Only a few singular values are of significant size, while the remaining singular values are relatively small. To evaluate the importance of each singular value in the first singular value decomposition in the approximation, we generate smoothed estimates of P_{exp} for different numbers of the retained singular values in the first singular value decomposition. The numbers of the retained singular values in the other singular value decompositions are kept to a maximum. To reduce the computational complexity it is also possible to use relatively large numbers of singular values in these singular value decompositions, if these numbers are large enough to guarantee that no significant approximation takes place.

Then we calculate the root-mean-square errors between the corresponding smoothed estimates and P_{exp} and plot them in Fig. 2.B1. For two 3D sequences $P(k_1, k_2, k_3)$, $\hat{P}(k_1, k_2, k_3) \in \mathbb{R}$, $k_i = 1, 2, \ldots, N_i$, $i = 1, 2, 3$, the root-mean-square error is defined by

$$\epsilon = \left(\frac{1}{N_1 N_2 N_3} \sum_{k_1=1}^{N_1} \sum_{k_1=1}^{N_1} \sum_{k_3=1}^{N_3} (P(k_1, k_2, k_3) - \hat{P}(k_1, k_2, k_3))^2 \right)^{1/2}.$$

As the number of the retained singular values decreases, the root-mean-square error increases gradually because more noise components are suppressed. When the number of the retained singular values is six or less, the root-mean-square error increases rapidly. That implies the main features of the experimental PSF are severely distorted in the approximation due to the loss of dominant singular values. There is no precise criterion for how many singular values should be retained as this is essentially a trade-off between smoothing on the one hand and accuracy on the other hand. In this particular case we found that retaining the first eight singular values appears to be a good compromise.

Figure 2.A2, A3, A4 and A5 show the singular values of the other singular value decompositions used in the algorithm, when eight singular values are retained in the first and second singular value decompositions. We observe that the singular values of the last three singular value decompositions change much more slowly than the singular values of the first two singular value decompositions, because the noise components in the experimental PSF have already been largely suppressed in the first two singular value decompositions. Therefore, more singular values are expected to be retained in the last three singular value decompositions. By following the analogous procedure to that above we decided to retain 8, 80, 280 and 80 singular values of the second, third, fourth and fifth singular value decompositions, respectively.

In this case, the total numbers of singular values for each singular value decomposition were 128, 128, 129, 352 and 129 and we discarded 120, 120, 49, 72 and 49 singular values in these five singular value decompositions. Following the notation introduced in Section 2.3 the smoothed PSF is denoted as $P_{exp}^{rs_1}$, where the superscript rs_1 denotes the numbers of discarded singular values in the five singular value decompositions, i.e. $rs_1 = \{120, 120; 49, 72, 49\}$.

Figure 1 shows cross-sections of the smoothed PSF $P_{exp}^{rs_1}$. The cross-sections appear to be fairly smooth and the difference between the smoothed PSF $P_{exp}^{rs_1}$ and the experimental PSF P_{exp} appears as random noise. It seems that the random components and noise in the experimental PSF are suppressed effectively. Based on the relative uniformity of the difference between the experimental PSF and its smoothed version it appears that no significant artefacts were introduced during the smoothing procedure.

However, in the smoothed PSF ripples of very low amplitude can be seen in the cross-sections (Fig. 1.A2 and B2) extending from the main peak of the PSF to the boundaries of the cropped images parallel to the image axes. The level of these ripples is well below the level of the noise and the random components in the experimental PSF, with a maximum of about 80 in the 16-bit display units. This corresponds to a maximum error of about 16 photons, based on the manufacturer's specifications for our camera.

The experimental PSF P_{exp} has a relatively low signal level. We chose such an experimental PSF to illustrate that the algorithm can perform well in the presence of high variance of the random components in the data. To evaluate the algorithm for an experimental PSF with higher signal level we imaged a 200-nm bead and acquired the experimental PSF P_{exp200} (see Section 2.2 and Fig. 3.A1 and B1).

Using the same approach as above we obtained a smoothed estimate $P_{exp200}^{rs_2}$ whose cross-sections are shown in Fig. 3.A2, B2. We retained 14, 16, 85, 450 and 85 singular values for the five singular value decompositions. The total numbers of singular values for each singular value decomposition were 128, 128, 129, 616 and 129 in this case. As part of the approximation step we discarded 114, 112, 44, 166 and 44 singular values in the five singular value decompositions, i.e. $rs_2 = \{114, 112; 44, 166, 44\}$. In this case, more singular values are retained in the first two singular value decompositions because fewer singular values are corrupted due to the low

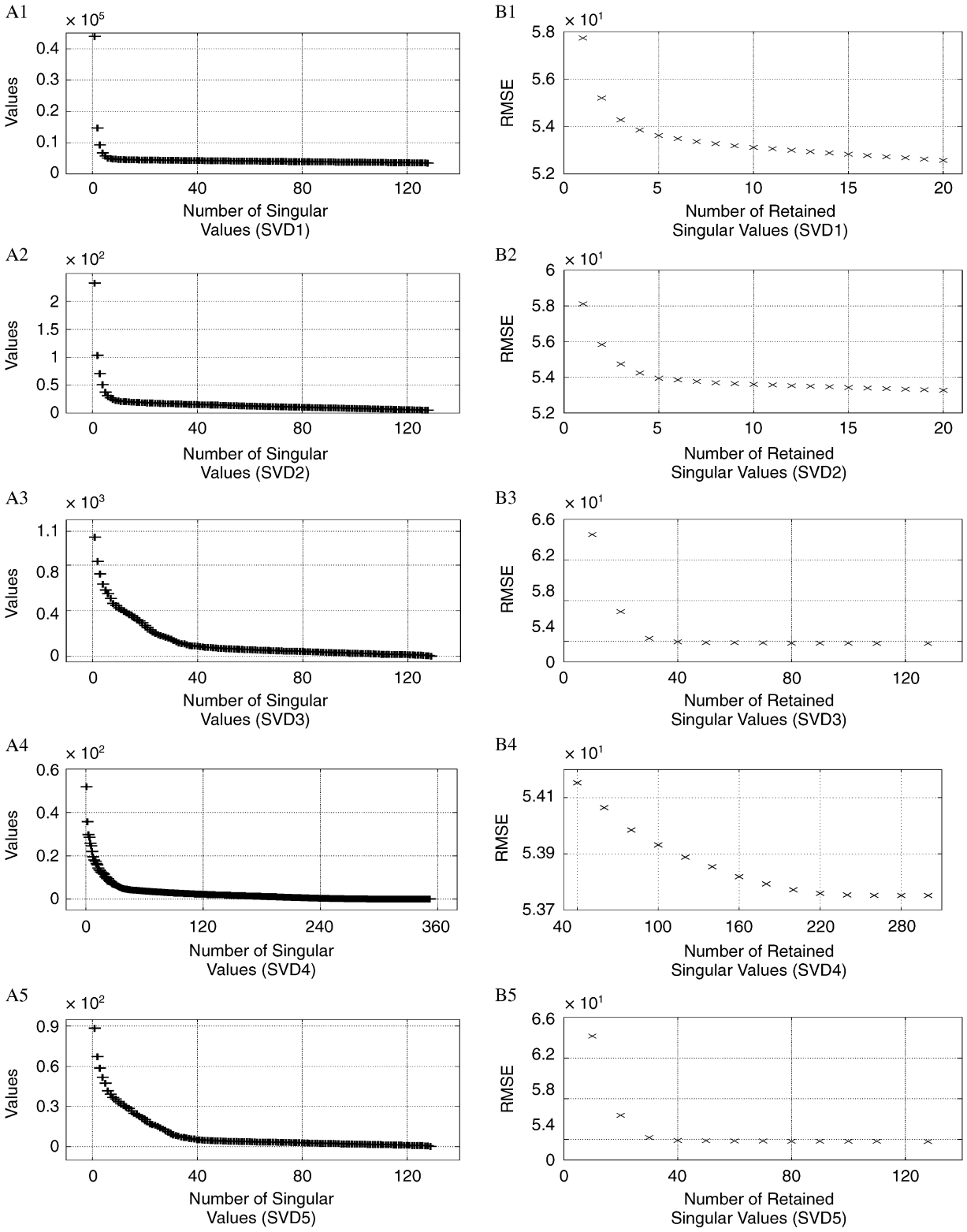

Fig. 2. The singular values of the five singular value decompositions and their importance in the approximation. A1, A2, A3, A4 and A5 show the singular values of the first, section, third, fourth and fifth singular value decompositions, respectively, when eight singular values are retained in the first and second singular value decompositions. B1, B2, B3, B4 and B5 show the root-mean-square errors between the experimental PSF P_{exp} (Fig. 1) and the smoothed PSFs obtained by varying the numbers of the retained singular values in the first, second, third, fourth or fifth singular value decomposition, respectively, while keeping the same numbers of the retained singular values in the other singular value decompositions (see Section 3.1 for details).

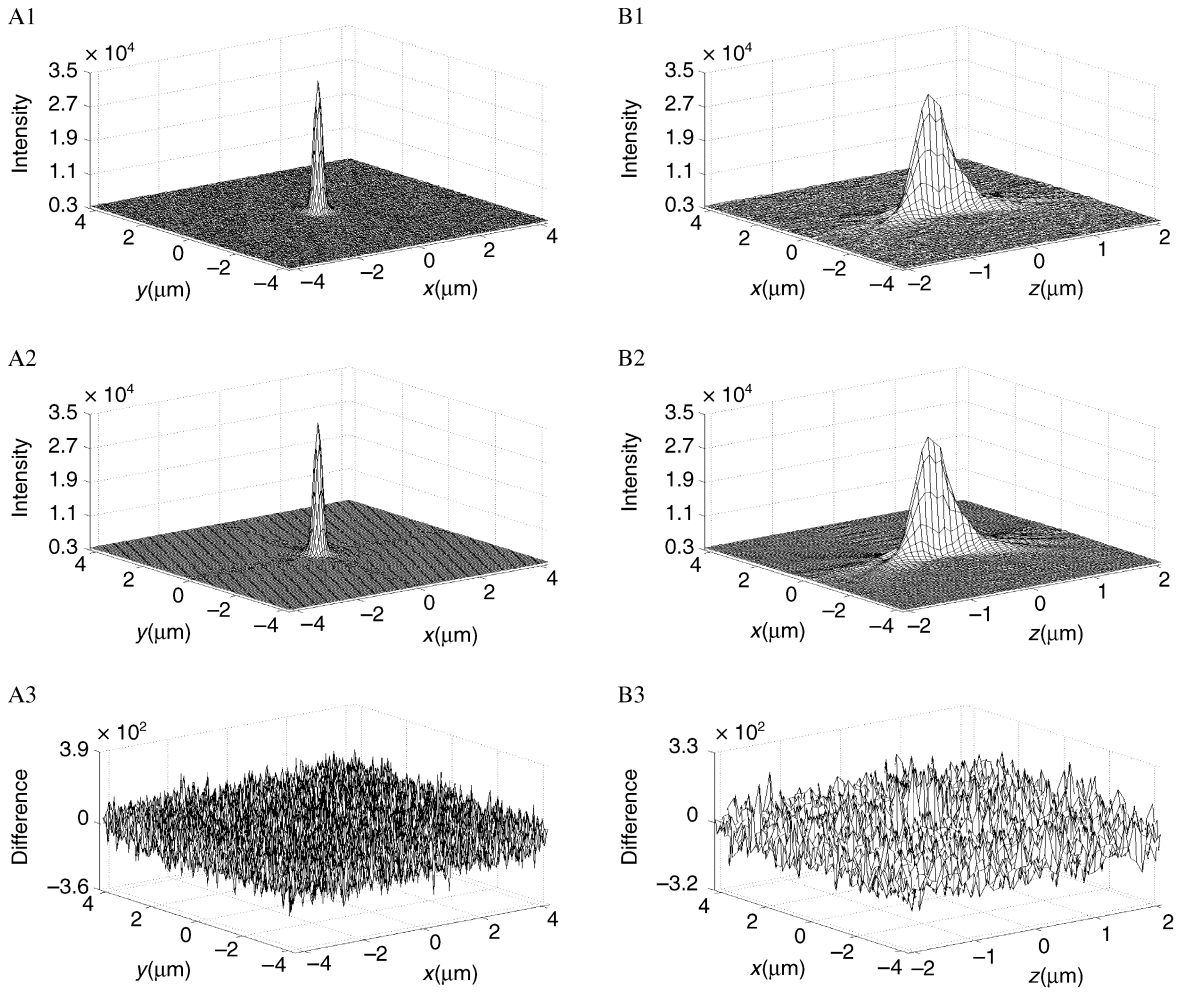

Fig. 3. Comparison of the experimental PSF P_{exp200} (A1, B1) and its smoothed estimate $P_{exp200}^{rs_2}$, $rs_2 = \{114, 112; 44, 166, 44\}$ (A2, B2) calculated via the noise suppression algorithm (Appendix). A1, A2 and A3 show cross-sections in the x–y plane at the focal level, the 22nd frame ($P_{exp200}(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$), of the experimental PSF, the 22nd frame ($P_{exp200}^{rs_2}(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$) of the estimate and their difference, respectively, B1, B2 and B3 show cross-sections in the x–z plane, the cross-section $P_{exp200}(64, k_2, k_3)$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$, the cross-section $P_{exp200}^{rs_2}(64, k_2, k_3)$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$, and their difference, respectively. The coordinate system is defined as in Fig. 1.

noise level in contrast to the PSF P_{exp} that has much lower signal level. As in the earlier analysis the difference between the experimental PSF and its smoothed version indicates that the smoothing did not introduce significant distortions. The ripples that could be seen for the low signal are hardly visible in the smoothed version for this PSF (Fig. 3). Zooming into the plots (data not shown) they are, however, revealed at an average level of 90 counts in the 16-bit display units. This shows that the ripples do not completely disappear with the higher signal level but their sizes become much lower is comparison with the signal level.

3.2. Smoothing of PSFs with simulated random components

In the previous section we applied the algorithm to the experimental PSF P_{exp} and obtained a smoothed estimate $P_{exp}^{rs_1}$. The plots of the differences between the experimental PSF and its smoothed version show no appreciable distortions. To be able to assess more precisely the performance of the smoothing algorithm we also investigated its behaviour for simulated data, because then we have the possibility to compare the results of the smoothing process with the deterministic PSFs that we used to generate the simulated PSFs.

We re-introduced randomness to the smoothed estimate $P_{exp}^{rs_1}$ to obtain a simulated PSF P_n^{λ} by simulating a Poisson distribution for each pixel (see Section 2.4). This simulated PSF was then processed again with our smoothing algorithm to investigate its effectiveness by comparing the resulting PSF P_n^{λ, rs_e} with the PSF $P_{exp}^{rs_1}$.

It is instructive to see the effects of the randomness introduced to $P_{exp}^{rs_1}$ on the singular values of the five singular value decompositions that are at the core of the algorithm when it is

applied to the randomized version of $P_{exp}^{rs_1}$, i.e. P_n^1 with $\lambda = 1$ (Fig. 4.A1 and A3). From Fig. 5.A1 and A2 we observe that the singular values of the first two singular value decompositions for the PSF P_n^1 are different from those for the data array $P_{exp}^{rs_1}$, especially for the small singular values, because the small singular values are corrupted by the randomness introduced in the PSF function P_n^1. From Fig. 5.A3, A4 and A5 the difference between the singular values of the last three singular value decompositions is not as significant as that shown in Fig. 5.A1 and A2 because the randomness in P_n^1 is partially suppressed by the approximations that take place in the first and second singular value decompositions.

Using the same approach as that described in Section 3.1, we decided to retain 9, 9, 100, 300 and 100 singular values in the first, second, third, fourth and fifth singular value decompositions, respectively. In this case, the total numbers of singular values of each singular value decomposition were 128, 128, 129, 396 and 129. Therefore, we discarded 119, 119, 29, 96 and 29 singular values in the five singular value decompositions to obtain the smoothed PSF P_n^{1,rs_e} with truncation parameter $rs_e = \{119, 119; 29, 96, 29\}$.

Cross-sections of the estimate P_n^{1,rs_e} are shown in Fig. 4.B1 and B3. It appears that the randomness in the PSF P_n^1 is effectively suppressed. Difference plots (Fig. 4.B2 and B4) between the deterministic PSF of the simulation, i.e. $P_{exp}^{rs_1}$, and the smoothed PSF P_n^{1,rs_e} show that the smoothing algorithm can recover the starting PSF quite well. The only potential problem is the small ripples that were seen earlier. Based on the scales used here, the ripples have sizes that correspond to a maximum of about 12 photons per pixel, which is low in comparison with the noise level.

We also smoothed the PSF P_n^1 via a standard method by applying a Gaussian filter with standard deviation one (Chen *et al.*, 1995; van Kempen & van Vliet, 1997). Comparing P_n^{1,rs_e} (Fig. 4.B1, B2, B3, B4) with the result P_n^g of the application of the Gaussian filter (Fig. 4.C1, C2, C3, C4), it becomes clear that using the Gaussian filter results in much larger errors than using the new smoothing algorithm. This is especially the case at the peak of the PSF along the optical axis. For example, the maximum error of the smoothed PSF P_n^g is 1330, which is about eight times the maximum error of 168 introduced by the new smoothing algorithm. In addition, the root-mean-square error (11) introduced by the use of the Gaussian filter is more than twice the root-mean-square error (4.5) introduced by the new smoothing algorithm.

It is instructive to examine whether the proposed smoothing algorithm can also be applied to smooth the randomness of a simulated PSF whose intensity is given by a theoretical PSF. To this end a theoretical 3D PSF was simulated using Li & Wolf's (1984) model, shown in Fig. 6.A1 and A2. The simulated acquired PSF P_{nt} was then simulated by simulating a spatial Poisson process whose intensity is given by the theoretical PSF (see Section 2.4 and Fig. 6.B1 and B2). The smoothed PSF $P_{nt}^{rs_t}$ was obtained with the reduction parameter $rs_t = \{121, 120; 29, 18, 29\}$ and its cross-sections are shown in Fig. 6.C1 and C2. Figure 6 shows that the smoothing process is relatively effective, as can be seen by examining the residuals. It appears that the algorithm performs equally well on experimental PSFs as on theoretical ones.

To study the behaviour of the smoothing algorithm for different levels of 'randomness' in the experimental PSFs, we also simulated PSFs based on $P_{exp}^{rs_1}$ with different values for the variance of the Poisson statistics. This was done by varying the parameter λ that linearly scales the intensity of the Poisson random variables (Section 2.4). The analysis was carried out analogously to the simulation that was discussed above.

We introduce a randomness factor α to measure quantitatively the randomness of the simulated PSFs as

$$\alpha = \frac{\sum_{k_1=1}^{N_1}\sum_{k_2=1}^{N_2}\sum_{k_3=1}^{N_3}(P_{exp}^{rs_1}(k_1,k_2,k_3) - b_P)^2}{\sum_{k_1=1}^{N_1}\sum_{k_2=1}^{N_2}\sum_{k_3=1}^{N_3}(P_n^{\lambda}(k_1,k_2,k_3) - P_{exp}^{rs_1}(k_1,k_2,k_3))^2},$$
$$k_i = 1, 2, \ldots, N_i, i = 1, 2, 3.$$

Note that the randomness factor α is calculated in the same way as the signal-to-noise ratio used in van Kempen (1998). We prefer to use the term randomness factor instead of signal-to-noise ratio because we want to distinguish the randomness of the data resulting from a Poisson statistics of the photon emission process from noise that is extraneously added to the data through, for example, the measurement process in the CCD chip.

The smoothing algorithm is applied to each simulated PSF P_n^{λ}. When the smoothed estimates of P_n^{λ} are calculated, the numbers of the retained singular values for each smoothed estimate are determined separately, as discussed in Section 3.1. The root-mean-square error (RMSE1) between the PSFs P_n^{λ} and $P_{exp}^{rs_1}$, and the root-mean-square error (RMSE2) between the smoothed estimates and $P_{exp}^{rs_1}$ are calculated and summarized in Table 1. It is observed from Table 1 that the randomness (indicated by RMSE1) increases as the randomness factor α decreases as λ gets smaller. When the randomness increases, more singular values are corrupted and fewer singular values are retained to calculate the smoothed PSF. As a result, only the rough features of $P_{exp}^{rs_1}$ can be recovered and we obtain a corresponding increase in RMSE2. However, the ratio of RMSE2 over RMSE1 is small and decreases as the level of randomness increases. This suggests that the algorithm suppresses randomness effectively, even when the pixels of simulated PSF have large variances.

Table 1. The effectiveness of the algorithm to smooth PSFs with different levels of simulated random components. RMSE1 is the root-mean-square error between P_n^{λ} and the data array $P_{exp}^{rs_1}$. RMSE2 is the root-mean-square error between the smoothed PSFs and the data array $P_{exp}^{rs_1}$.

λ	1	3	6	9
α	2.0	5.9	11.8	17.8
RMSE1	39.5	22.8	16.2	13.2
RMSE2	4.5	2.9	2.4	2.2

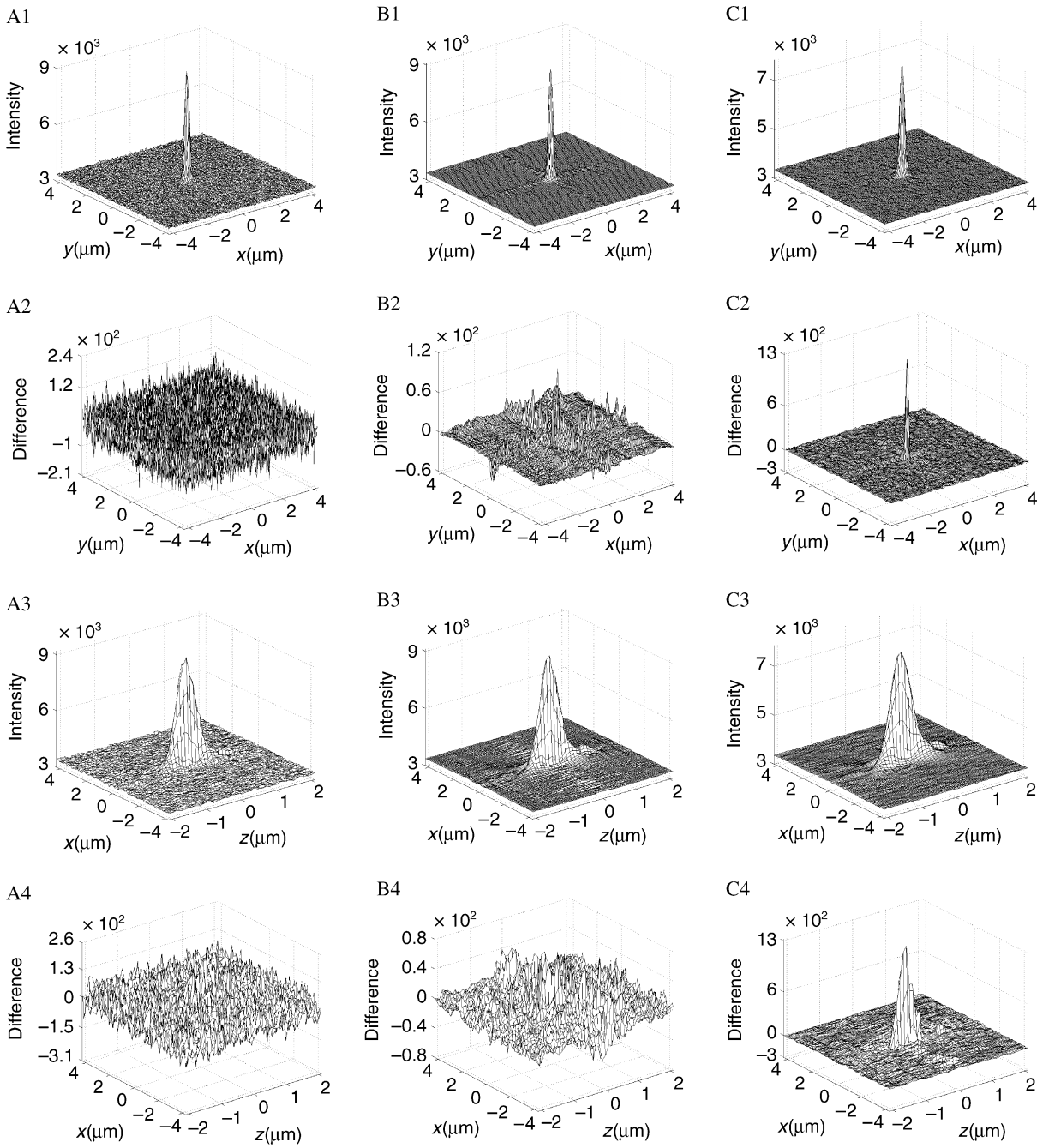

Fig. 4. Comparison of the smoothed estimates P_n^{1,rs_ϵ}, $rs = \{119, 119; 29, 96, 29\}$ (B1, B2, B3, B4) and P_n^g(C1, C2, C3, C4) of the PSF P_n^1(A1, A2, A3, A4), discussed in Section 3.2. The estimate P_n^g is obtained using a Gaussian filter with standard deviation one while the estimate P_n^{1,rs_ϵ} is calculated via the noise suppression algorithm (Appendix). A1 and A2 show cross-sections in the x–y plane at the focal level, the 22nd frame ($P_n^1(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$) of the simulated PSF and the difference between this frame and the 22nd frame of $P_{exp}^{rs_1}$ (see Section 2.4 and Fig. 1). A3 and A4 show cross-sections in the x–z plane, the cross-section $P_n^1(64, k_2, k_3)$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$ and the difference between it and the cross-section $P_{exp}^{rs_1}(64, k_2, k_3)$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$, respectively. B1 and B2 show frames in the x–y plane at the focal level, the 22nd frame ($P_n^{1,rs_\epsilon}(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$) of the smoothed estimate and its error, respectively. B3 and B4 show the cross-sections in the x–z plane $P_n^{1,rs_\epsilon}(64, k_2, k_3)$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$, and its error, respectively. C1 and C2 show frames in the x–y plane at the focal level, the 22nd frame ($P_n^g(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$) of the smoothed estimate and its error, respectively. C3 and C4 show the cross-sections in the x–z plane $P_n^g(64, k_2, k_3)$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$, and its error, respectively. The coordinate system is defined as in Fig. 1.

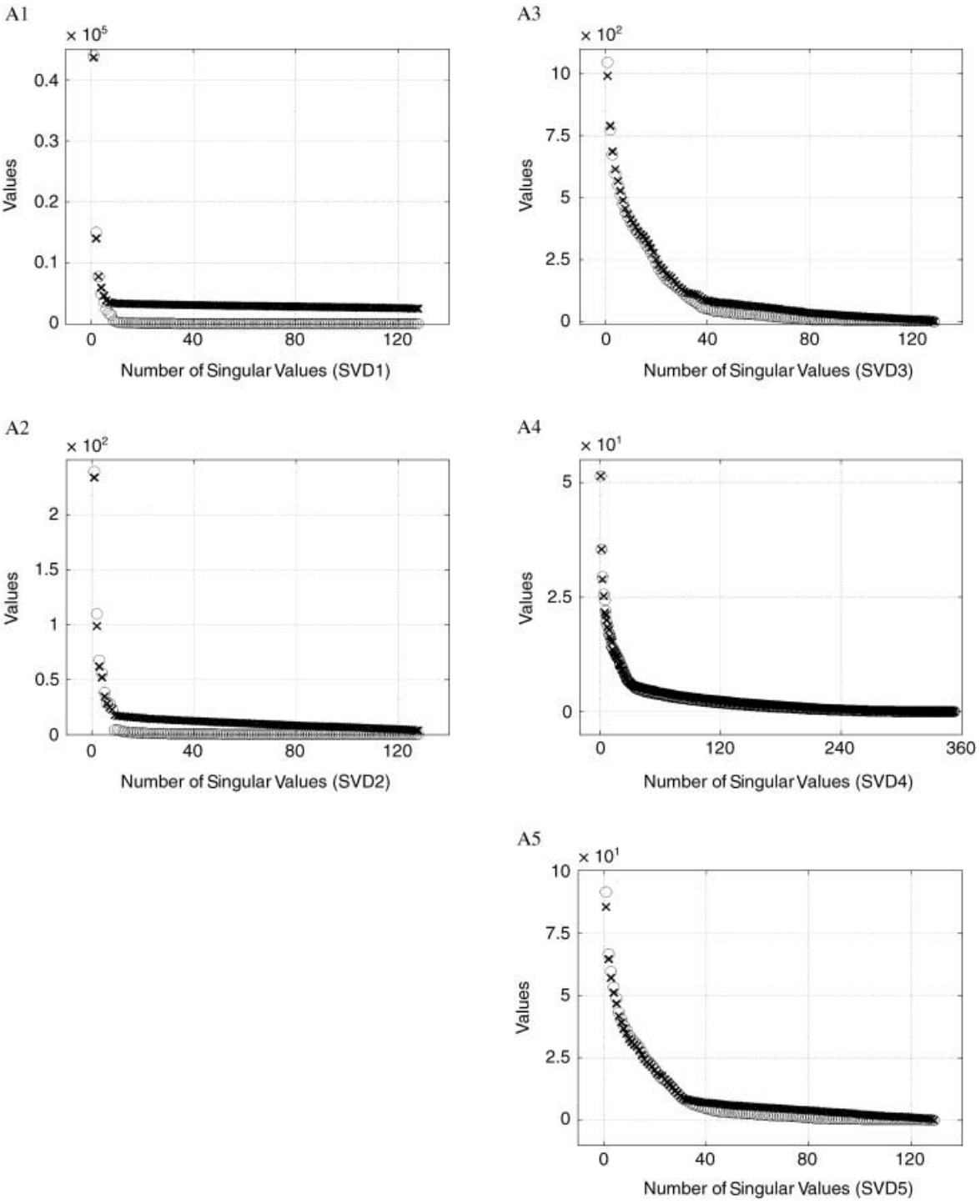

Fig. 5. Comparison of the singular values of the five singular value decompositions (A1, A2, A3, A4, A5) used in the noise suppression algorithm (Appendix), when the algorithm is applied to the noisy simulated PSF P_n^1 and its deterministic version, the data array $P_{exp}^{rs_1}$. The crosses (×) correspond to the PSF P_n^1, while the circles (O) correspond to the data array $P_{exp}^{rs_1}$.

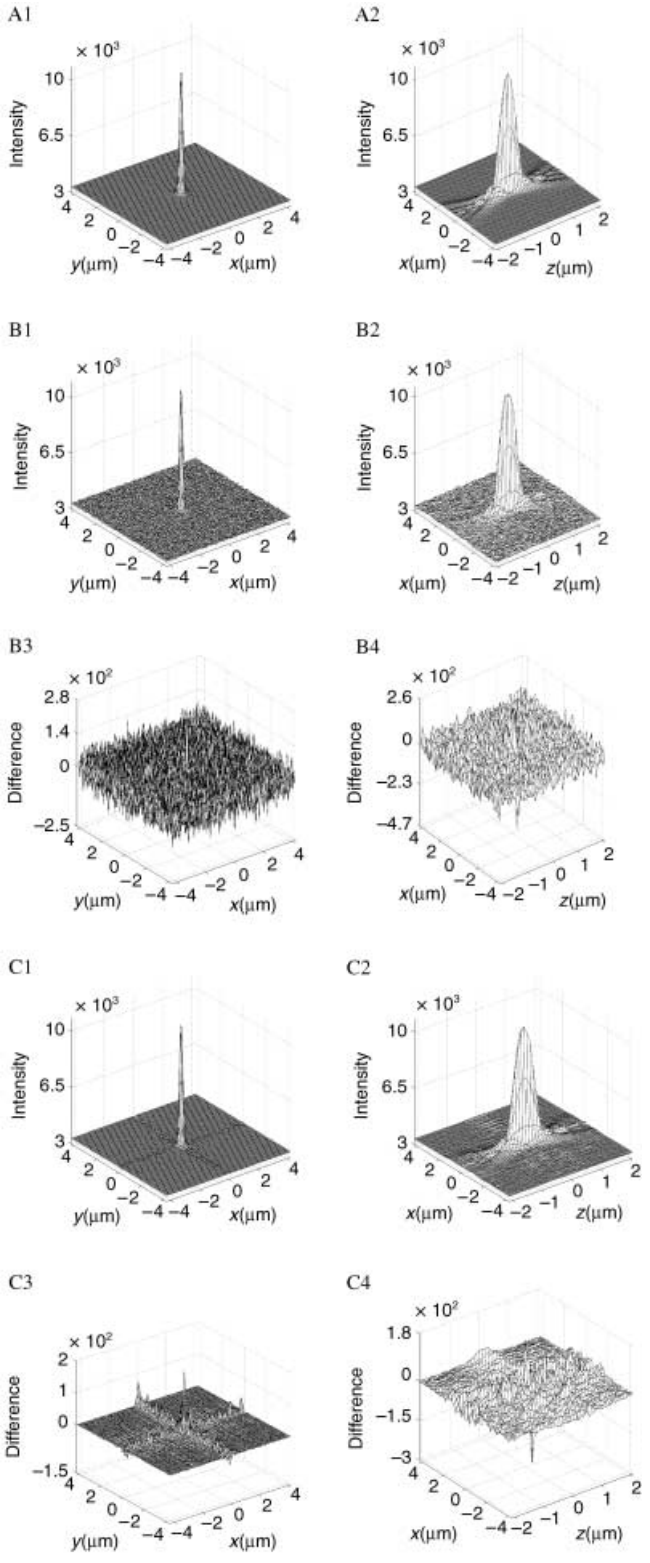

Fig. 6. Comparison of the simulated PSF P_{dt} (A1, A2) and its smoothed estimate $P_{nt}^{rs_t}$, $rs_t = \{121, 120; 29, 18, 29\}$ (C1, C2) calculated via the noise suppression algorithm. A1, B1 and C1 show cross-sections in the x–y plane at the focal level, the 22nd frame ($P_{dt}(k_1, 22, k_3)$, $k_1, k_3 = 1, \ldots, 128$) of the theoretical PSF, the corresponding frame of the simulated acquired PSF P_{nt} and the smoothed PSF $P_{nt}^{rs_t}$, respectively. B3 and C3 show the difference between the data shown in A1 and B1 and the difference between the data shown in A1 and C1, respectively. A2, B2 and C2 show cross-sections in the x–z plane, the cross-section $P_{dt}(64, k_2, k_3)$, $k_2 = 1, \ldots, 43$, $k_3 = 1, \ldots, 128$, and the corresponding cross-sections in P_{nt} and $P_{nt}^{rs_t}$, respectively. B4 and C4 show the difference between the data shown in A2 and B2 and the difference between the data shown in A2 and C2, respectively. The coordinate system is defined as in Fig. 1.

4. Enhanced performance of deconvolution algorithms

In this section we investigate the influence of the smoothing of the PSF on the performance of the accelerated Richardson–Lucy algorithm and the regularized least-squares algorithm.

4.1. Simulated data

A 3D image set of four balls was simulated as discussed in Section 2.6. This image set, shown in Fig. 7, was then deconvolved using the accelerated Richardson–Lucy algorithm and the regularized least-squares algorithm with both the noisy PSF P_n^1 and its smoothed version P_n^{1,rs_e}.

Figure 8 shows the root-mean-square errors between the original object and the recovered objects for different numbers of iterations in the accelerated Richardson–Lucy algorithm. When the PSF P_n^1 is used the best recovered object in terms of the root-mean-square error is obtained after 155 iterations. The corresponding root-mean-square error is 1306, which is about three times the error of 578 for the recovered object after 320 iterations, when the smoothed PSF P_n^{1,rs_e} (Section 3.2) is used. From Fig. 8 we also observe that as the number of iterations increases the recovered object diverges away from the original object in both cases, but the pace of divergence is much slower when the smoothed PSF is used. For example, the root-mean-square error between the recovered object using the smoothed PSF after 1200 iterations and the original object is 667, which is about one-half of 1306 for the best recovered object using the non-smoothed PSF P_n^1.

This analysis shows that the performance of the accelerated Richardson–Lucy algorithm is very sensitive to the amount of randomness in the PSF. Moreover, the recovered objects are improved significantly by using the smoothed PSF, as shown in Fig. 9.A1, A2, B1 and B2.

Figure 9.C1 and C2 show cross-sections of the recovered object when the PSF P_n^1 is used in the regularized least-squares algorithm with the regularization term $\gamma = 5 \times 10^{-6}$. The recovered object appears very noisy, especially at the background area. To suppress this noisy reconstruction a larger regularization term is often used at the cost of losing more details of the recovered object. If we replace the PSF P_n^1 by its smoothed estimate P_n^{1,rs_e} in the regularized least-squares algorithm and keep the regularization parameter γ unchanged, the noisy appearance of the background area of the recovered object (Fig. 9.D1 and D2) is significantly reduced and the recovered balls become smoother. The root-mean-square

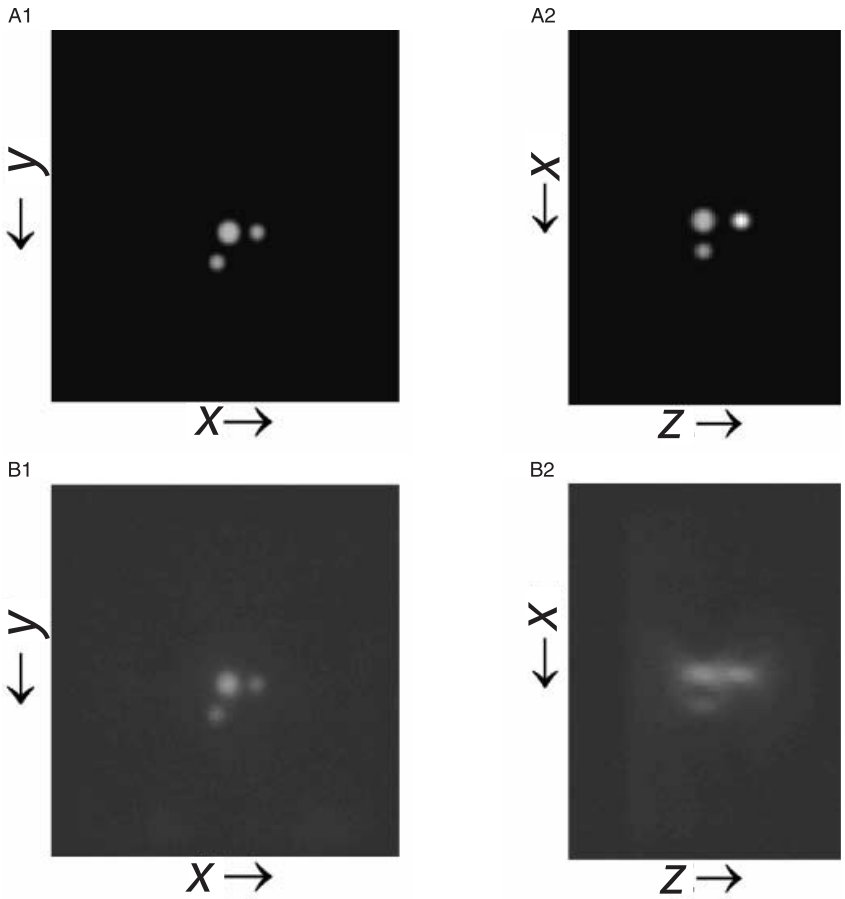

Fig. 7. Cross-sections of the simulated object O_s (A1, A2) and the simulated image set Im_n (B1, B2), discussed in Section 2.6. A1 and B1 show cross-sections in the x–y plane, the 40*th* frame ($O_s(k_1, 40, k_3)$, $k_1, k_3 = 1, \ldots, 148$) of the simulated object and the 40*th* frame ($Im_n(k_1, 40, k_3)$, $k_1, k_3 = 1, \ldots, 148$) of the simulated image set, respectively. A2 and B2 show cross-sections in the x–z plane, the cross-section $O_s(82, k_2, k_3)$, $k_2 = 1, \ldots, 80$, $k_3 = 1, \ldots, 148$ and the cross-section $Im_n(82, k_2, k_3)$, $k_2 = 1, \ldots, 80$, $k_3 = 1, \ldots, 148$, respectively. The coordinate system is defined as in Fig. 1.

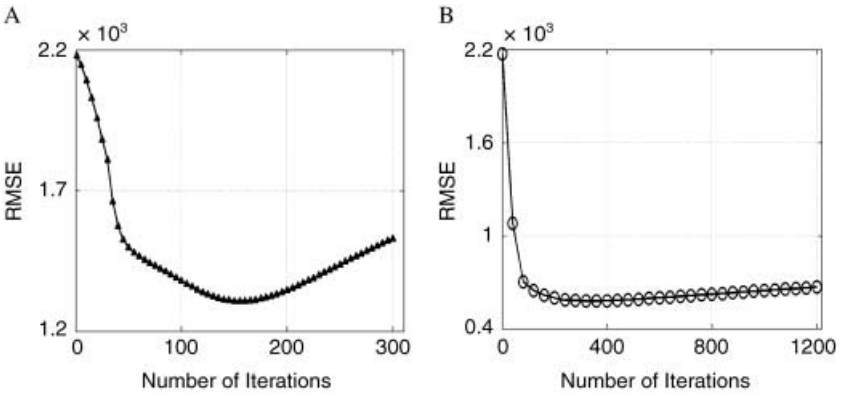

Fig. 8. Root-mean-square errors of the recovered objects calculated from the simulated image set Im_n (Fig. 7) via the accelerated Richardson–Lucy algorithm for different numbers of iterations. (A) Triangles (▲) represent the root-mean-square errors between the recovered objects using the PSF P_n^1 (Fig. 4.A1, A3) and the original object O_s (Fig. 7). (B) Circles (○) represent the root-mean-square errors between the recovered objects using the smoothed PSF P_n^{1,rs_e} (Fig. 4.B1, B3) and the original object.

error between the recovered object and the simulated object is 4505, which is significantly smaller than that of 6367 for the recovered object using the non-smoothed PSF P_n^1.

Therefore as for the accelerated Richardson–Lucy algorithm the use of a smoothed PSF significantly improves the performance of the regularized least-squares algorithm.

Noise suppression in deconvolution algorithms has been discussed in the literature in some detail (e.g. Verveer *et al.*, 1999). It is, however, important to note that this literature typically addresses the problem of noise suppression that originates from noisy images rather than from non-smooth PSFs.

4.2. Experimental data

In this section, we deconvolve the image set Im_c (Fig. 10.A1 and A2) of a Jurkat cell transfected with FcRn-GFP. The MHC Class I related receptor, FcRn, plays a role in the transfer of gammaglobulin (IgG) from mother to young and also regulates the serum levels of IgG (reviewed in Ghetie & Ward, 2000). We compare the recovered objects calculated using the experimental PSF P_{exp} (Fig. 1) and the smoothed PSF $P_{exp}^{rs_1}$ (Fig. 1) in the accelerated Richardson–Lucy algorithm and the regularized least-squares algorithm.

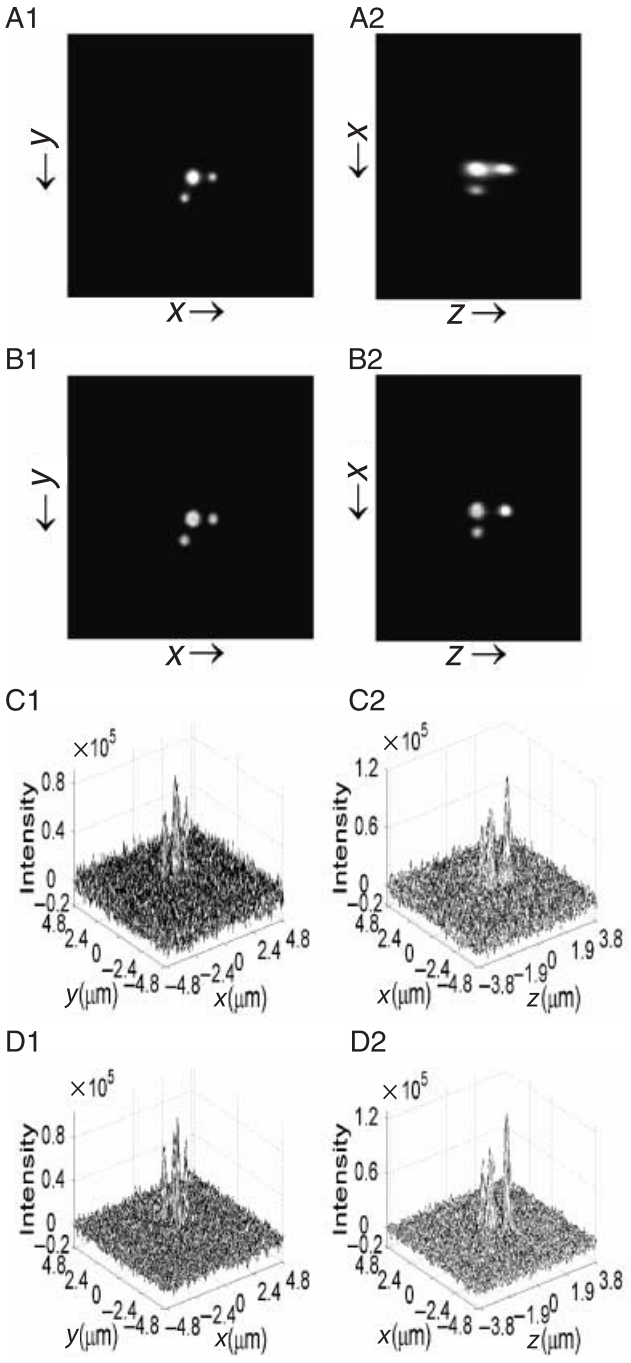

Fig. 9. Cross-sections of the recovered objects calculated from the simulated image set Im_n (Fig. 7) using the accelerated Richardson–Lucy algorithm (A1, A2, B1, B2) and the regularized least-squares algorithm (C1, C2, D1, D2). A1 and A2 show the 40*th* frame of the recovered object in the *x*–*y* plane and the cross-section of the recovered object in the *x*–*z* plane, corresponding to the cross-section $Im_n(82, k_2, k_3)$, $k_2 = 1, \ldots, 80$, $k_3 = 1, \ldots, 148$, respectively, when the PSF P_n^1 (Fig. 4.A1, A3) is used and 155 iterations are performed. B1 and B2 show the corresponding frames, when the smoothed PSF P_n^{1,rs_e} (Fig. 4.B1, B3) is used and 320 iterations are performed. C1 and C2 show the corresponding frames, when the PSF P_n^1 is used and the regularization term is set to $\gamma = 5 \times 10^{-6}$. D1 and D2 show the corresponding frames, when the smoothed PSF P_n^{1,rs_e} and the same regularization term is used. Three-dimensional plots are used in C1, C2, D1 and D2 to show the background noise. The coordinate system is defined as in Fig. 1.

Figure 10.B1, B2, C1 and C2 show cross-sections of the recovered objects using the experimental and smoothed PSF P_{exp} and $P_{exp}^{rs_1}$, respectively, in the accelerated Richardson–Lucy algorithm after 200 iterations. We observe that the recovered objects differ significantly. More details can be seen in the recovered object using the smoothed PSF.

Figure 10.D1 and D2 show cross-sections of the recovered object calculated using the experimental PSF P_{exp} in the regularized least-squares algorithm, when the regularization term γ equals 5×10^{-5}. The recovered object has a very noisy appearance. When the smoothed PSF $P_{exp}^{rs_1}$ is used with the same regularization term γ the noisy appearance is much reduced, although it is still present in the deconvolved image set (Fig. 10.E1 and E2).

5. Conclusions

In this paper we proposed to apply a novel algorithm to the smoothing of experimental PSFs. The analysis of both simulated data and experimental data has shown that this algorithm is capable of effectively suppressing randomness and noise in experimental PSFs. The algorithm performs favourably in comparison with the use of a Gaussian filter because it does not lead to the large errors at the peak of the PSF that are characteristic of algorithms that are based on averaging of neighbouring pixels.

We have also used the smoothed PSFs in the accelerated Richardson–Lucy algorithm and the regularized least-squares algorithm with both simulated data and experimental data. For the analysis of the simulated data, we used the root-mean-square error criterion to compare the results of the deconvolution. The recovered objects using the smoothed PSF have much smaller root-mean-square errors for both algorithms. In addition, for the accelerated Richardson–Lucy algorithm, if the smoothed PSF is used the recovered object diverges very slowly as the number of the iterations increases. For the regularized least-squares algorithm the recovered object has a less noisy appearance if the smoothed PSF is used.

For experimental data the use of the smoothed PSF reveals more details than the use of the non-smoothed PSF in the accelerated Richardson–Lucy algorithm. For the regularized least-squares algorithm the recovered object has much less background noise when the smoothed PSF is used.

Acknowledgements

We thank Carlos Vaccaro for preparing the biological sample. This work was supported in part by grants from the National Institutes of Health (R01 AI50747 and R01 AI39167).

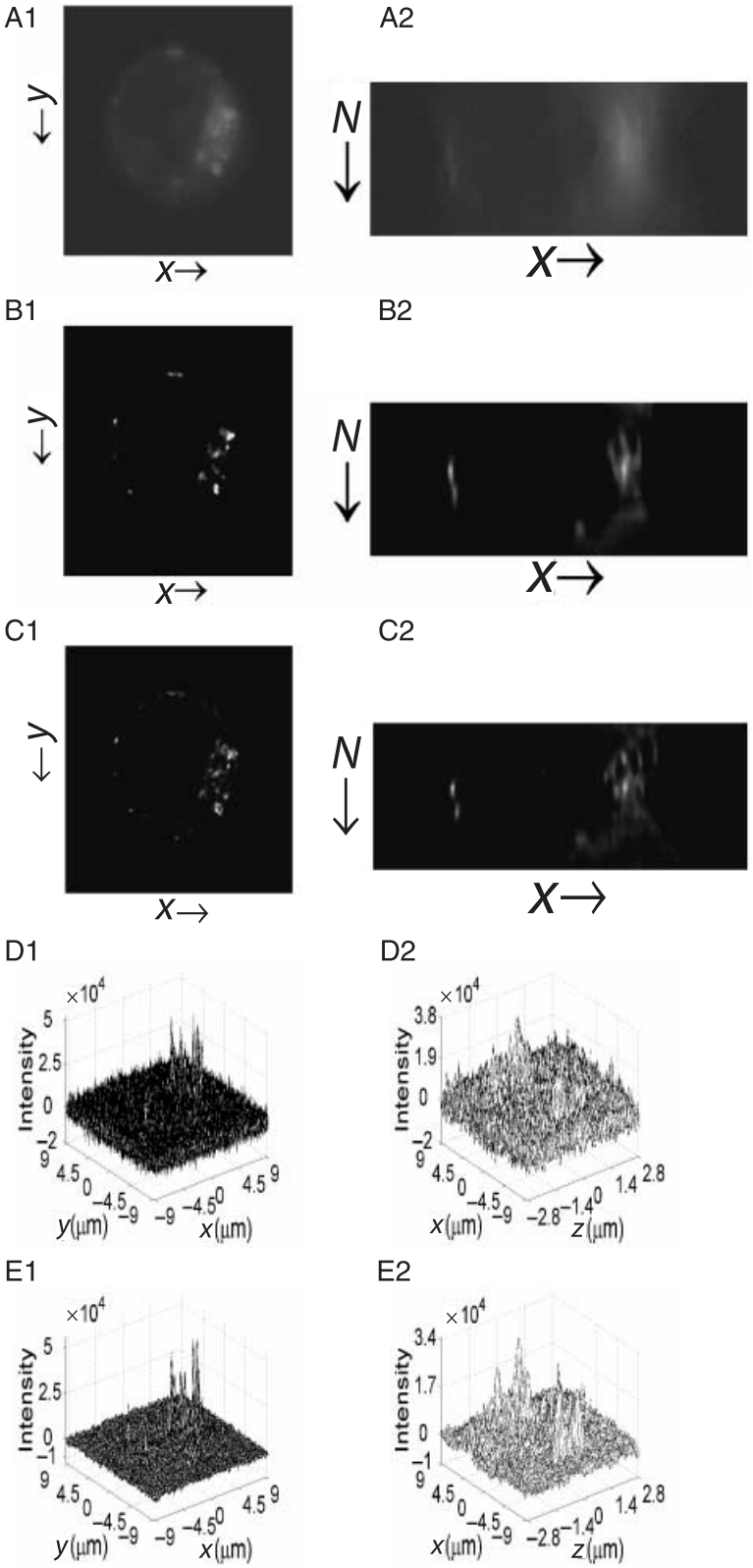

Fig. 10. Comparison of the recovered objects calculated from the image set Im_c (A1, A2) of a Jurkat cell transfected with FcRn-GFP (Section 4.2) via the accelerated Richardson–Lucy algorithm (B1, B2, C1, C2) and the regularized least-squares algorithm (D1, D2, E1, E2). The 3D image set Im_c consists of 58 images each being a 280 × 280-pixel array. A1 and A2 show the 18*th* frame ($Im_c(k_1, 18, k_3)$, $k_1 = 1, \ldots, 280$, $k_3 = 1, \ldots, 280$) of the image set in the x–y plane and the cross-section $Im_c(140, k_2, k_3)$, $k_2 = 1, \ldots, 58$, $k_3 = 1, \ldots, 280$ in the x–z plane. B1 and B2 show the corresponding cross-sections, when the PSF P_{exp} (Fig. 1.A1, B1) is used and 200 iterations are performed. C1 and C2 show the corresponding cross-sections, when the smoothed PSF $P_{exp}^{rs_1}$ (Fig. 1.A2, B2) is used and 200 iterations are carried out. D1 and D2 show the corresponding cross-sections, when P_{exp} is used and the regularization term is set to $\gamma = 5 \times 10^{-5}$. E1 and E2 show the corresponding cross-sections, when $P_{exp}^{rs_1}$ and the same regularization term are used. Three-dimensional plots are used in D1, D2, E1 and E2 to show the background noise. The coordinate system is defined as in Fig. 1.

References

Biggs, D.S.C. & Andrews, M. (1997) Acceleration of iterative image restoration algorithms. *Applied Opt.* **36**, 1766–1775.

Castleman, K.R. (1996) *Digital Image Processing*. Prentice Hall, Inc., New Jersey.

Chen, H., Swedlow, J.R., Grote, M., Sedat, J.W. & Agard, D.A. (1995) The collection, processing, and display of digital three-dimensional images of biological specimens. *Handbook of Biological Confocal Microscopy*, 2nd edn. (ed. by J.B. Pawley), pp. 197–210. Plenum Press, New York.

Ghetie, V. & Ward, E.S. (2000) Multiple roles for the major histocompatibility complex class I related receptor, FcRn. *Ann. Rev. Immunol.* **18**, 739–766.

Gibson, S.F. & Lanni, F. (1991) Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy. *J. Op. Soc. Am.* **8**, 1601–1613.

Inoue, S. & Spring, K.R. (1997) *Video Microscopy: The Fundamentals*. Plenum Publishing Corp., New York.

Jain, A.K. (1989) *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., New Jersey.

van Kempen, G.M.P. (1998) *Image restoration in fluorescence microscopy*. PhD thesis, Technische Universiteit Delft.

van Kempen, G.M.P. & van Vliet, L.J. (1997) Improving the restoration of textured objects with prefiltering. *Proceeding of 3rd Annual Conference of the Advanced School for Computing and Imaging (ASCI'97)*, pp. 174–179.

Li, Y. & Wolf, E. (1984) Three-dimensional intensity distribution near the focus in systems of different Fresnel numbers. *J. Opt. Soc. Am.* **1**, 801–808.

McNally, J.G., Preza, C., Conchello, J.A. & Thomas, L.J. (1994) Artifacts in computational optical-sectioning microscopy. *J. Opt. Soc. Am. A*, **11**, 1056–1067.

Ober, R.J., Lai, X., Lin, Z. & Ward, E.S. (2005) State space realization of a three-dimensional image set with application to noise reducion of fluorescent microscopy images of cells. *Multidimensional Systems and Signal Processing*. **16**, 7–48.

Ober, R.J., Radu, C.G., Ghetie, V. & Ward, E.S. (2001) Differences in promiscuity for antibody–FcRn iterations across species: implications for therapeutic antibodies. *Int. Immunol.* **13**, 1551–1559.

Preza, C., Miller, M.I., Thomas, L.J., Jr & McNally, J.G. (1992) Regularized linear method for reconstruction of three-dimensional microscopic objects from optical sections. *J. Opt. Soc. Am. A*, **9**, 219–228.

Shaw, P.J. (1993) Computer reconstruction in three-dimensional fluorescence microscopy. *Electronic Light Microscopy: The Principles and Practice of Video-Enhanced Contrast, Digital Intensified Fluorescence, and Confocal Scanning Light Microscopy* (ed. by D.M. Shotton), pp. 211–230. Wiley-Liss, Inc., New York.

Shaw, P.J. & Rawlins, D.J. (1991) The point-spread functions of a confocal microscope: its measurement and use in deconvolution of 3-D data. *J. Microsc.* **163**, 151–165.

Verveer, P.J., Gemkow, M.J. & Jovin, T.M. (1999) A comparison of image restoration approaches applied to three-dimensional confocal and wide-field fluorescence microscopy. *J. Microsc.* **193**, 50–61.

van Vliet, L.J. (1993) *Grey-scale measurements in multi-dimensional digitized images*. PhD thesis, University of Delft.

Appendix

Algorithm

Let $P(k_1, k_2, k_3), k_i = 1, 2, \ldots, N_i, i = 1, 2, 3$, represent a 3D data array.

1. Subtract an estimated background level $\hat{b}$ (e.g. the mean of the data points near the boundary of the data array P) from data array P, and define $\Phi(k_1, k_2, k_3) := P(k_1, k_2, k_3) - \hat{b}$, $k_i = 1, 2, \ldots, N_i$, $i = 1, 2, 3$ (if no approximation is carried out, take $\hat{b} = 0$).
2. Arrange the entries of the resulting data array Φ to form a matrix Q_3 as

$$Q_3 = \begin{bmatrix} \Phi(1,1,1) & \Phi(1,1,2) & \cdots & \Phi(1,2,1) & \Phi(1,2,2) \\ \Phi(2,1,1) & \Phi(2,1,2) & \cdots & \Phi(2,2,1) & \Phi(2,2,2) \\ \Phi(3,1,1) & \Phi(3,1,2) & \cdots & \Phi(3,2,1) & \Phi(3,2,2) \\ \vdots & \vdots & & \vdots & \vdots \\ \Phi(N_1,1,1) & \Phi(N_1,1,2) & \cdots & \Phi(N_1,2,1) & \Phi(N_1,2,2) \end{bmatrix}$$
$$\begin{bmatrix} \cdots & \Phi(1,3,1) & \Phi(1,3,2) & \cdots & \Phi(1,N_2,N_3) \\ \cdots & \Phi(2,3,1) & \Phi(2,3,2) & \cdots & \Phi(2,N_2,N_3) \\ \cdots & \Phi(3,3,1) & \Phi(3,3,2) & \cdots & \Phi(3,N_2,N_3) \\ & \vdots & \vdots & & \vdots \\ \cdots & \Phi(N_1,3,1) & \Phi(N_1,3,2) & \cdots & \Phi(N_1,N_2,N_3) \end{bmatrix}.$$

3. (Approximation step 1) Decompose Q_3 via the singular value decomposition* as $Q_3 = U_1 \Sigma_1 V_1$. Partition $\Sigma_1 = diag(\hat{\Sigma}_1, \hat{\hat{\Sigma}}_1)$, $U_1 = [\hat{U}_1, \hat{\hat{U}}_1]$, and

$$V_1 = \begin{bmatrix} \hat{V}_1 \\ \hat{\hat{V}}_1 \end{bmatrix}$$

conformally, where $\hat{\Sigma}_1 \in \mathbb{R}^{K_1 \times K_1}$ and $\hat{\hat{\Sigma}}_1 \in \mathbb{R}^{r_1 \times r_1}$. Define $\Phi_1^{r_1}(i) \in \mathbb{R}^{1 \times K_1}$, $i = 1, \ldots, N_1$, such that

$$\begin{bmatrix} \Phi_1^{r_1}(1) \\ \Phi_1^{r_1}(2) \\ \Phi_1^{r_1}(3) \\ \vdots \\ \Phi_1^{r_1}(N_1) \end{bmatrix} := \hat{U}_1 \hat{\Sigma}_1^{1/2}$$

(for a diagonal matrix $D = diag(d_1, d_2, \ldots, d_n)$, $d_i \geq 0$, $i = 1, 2, \ldots, n$, we define $D^{1/2} = diag(d_1^{1/2}, d_2^{1/2}, \ldots, d_n^{1/2})$) and define $R_3^{r_1}(i) \in R^{K_1 \times 1}$, $i = 1, \ldots, N_2 N_3$, such that $[R_3^{r_1}(1), \ldots, R_3^{r_1}(N_2 N_3)] := \hat{\Sigma}_1^{1/2} \hat{V}_1$. Note that r_1 denotes the number of discarded singular values in this step. *A singular value decomposition of a 2D $M \times L$ matrix Q is defined by the factorization $Q = U \Sigma V$, where U, V are matrices of sizes M-by-K and K-by-L, respectively, such that $U^* U = I$ and $V V^* = I$, in which I denotes an identity matrix, and

$$\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_K) = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_K \end{bmatrix}$$

with singular values $\sigma_i > 0$, $i = 1, \ldots, K$.

4. Arrange $R_3^{r_1}(1), \ldots, R_3^{r_1}(N_2 N_3)$ to form Q_2 as

$$Q_2 := \begin{bmatrix} R_3^{r_1}(1) & R_3^{r_1}(2) \\ R_3^{r_1}(N_3+1) & R_3^{r_1}(N_3+2) \\ R_3^{r_1}(2N_3+1) & R_3^{r_1}(2N_3+2) \\ \vdots & \vdots \\ R_3^{r_1}((N_2-1)N_3+1) & R_3^{r_1}((N_2-1)N_3+2) \end{bmatrix}$$

$$\begin{matrix} R_3^{r_1}(3) & \cdots & R_3^{r_1}(N_3) \\ R_3^{r_1}(N_3+3) & \cdots & R_3^{r_1}(2N_3) \\ R_3^{r_1}(2N_3+3) & \cdots & R_3^{r_1}(3N_3) \\ \vdots & \cdots & \vdots \\ R_3^{r_1}((N_2-1)N_3+3) & \cdots & R_3^{r_1}(N_2N_3) \end{matrix} \Bigg],$$

where $Q_2 \in \mathbb{R}^{N_2K_1 \times N_3}$.

5. (Approximation step 2) Decompose Q_2 via the singular value decomposition as $Q_2 = U_2\Sigma_2V_2$. Partition $\Sigma_2 = diag(\hat{\Sigma}_2, \hat{\hat{\Sigma}}_2)$, $U_2 = [\hat{U}_2, \hat{\hat{U}}_2]$, and

$$V_2 = \begin{bmatrix} \hat{V}_2 \\ \hat{\hat{V}}_2 \end{bmatrix}$$

conformally, where $\hat{\Sigma}_2 \in \mathbb{R}^{K_2 \times K_2}$ and $\hat{\hat{\Sigma}}_2 \in \mathbb{R}^{r_2 \times r_2}$. Define $\Phi_2^{r_2}(i) \in \mathbb{R}^{K_1 \times K_2}$, $i = 1, \ldots, N_2$, such that

$$\begin{bmatrix} \Phi_2^{r_2}(1) \\ \Phi_2^{r_2}(2) \\ \Phi_2^{r_2}(3) \\ \vdots \\ \Phi_2^{r_2}(N_2) \end{bmatrix} := \hat{U}_2\hat{\Sigma}_2^{1/2},$$

and define $\Phi_3^{r_2}(i) \in \mathbb{R}^{K_2 \times 1}$, $i = 1, \ldots, N_3$, such that $[\Phi_3^{r_2}(1), \Phi_3^{r_2}(2), \ldots, \Phi_3^{r_2}(N_3)] := \hat{\Sigma}_2^{1/2}\hat{V}_2$. Note that r_2 denotes the number of discarded singular values in this step.

6. (Approximation steps 3–5) Calculate the realizations $(A_1^{r_1;s_1}, B_1^{r_1;s_1}, C_1^{r_1;s_1})$, $(A_2^{r_2;s_2}, B_2^{r_2;s_2}, C_2^{r_2;s_2})$ and $(A_3^{r_2;s_3}, B_3^{r_2;s_3}, C_3^{r_2;s_3})$ of $\Phi_1^{r_1}$, $\Phi_2^{r_2}$ and $\Phi_3^{r_2}$, respectively, via the realization step (see below) for some $s_1, s_2, s_3 \geq 0$, where $\Phi_1^{r_1}(i) \in \mathbb{R}^{1 \times K_1}$, $i = 1, \ldots, N_1$, $\Phi_2^{r_2}(i) \in R^{K_1 \times K_2}$, $i = 1, \ldots, N_2$, and $\Phi_3^{r_2}(i) \in \mathbb{R}^{K_2 \times 1}$, $i = 1, \ldots, N_3$, from step 3 and step 5. Note that s_1, s_2 and s_3 are the numbers of singular values discarded in the corresponding realization step.
7. Calculate the estimate $\Phi^{r_1,r_2;s_1,s_2,s_3}$ as

$$\begin{aligned} \Phi^{r_1,r_2;s_1,s_2,s_3}(k_1,k_2,k_3) &= C_1^{r_1;s_1}(A_1^{r_1;s_1})^{k_1-1}B_1^{r_1;s_1}C_2^{r_2;s_2}(A_2^{r_2;s_2})^{k_2-1} \\ &\quad B_2^{r_2;s_2}C_3^{r_2;s_3}(A_3^{r_2;s_3})^{k_3-1}B_3^{r_2;s_3}, \\ k_i &= 1, 2, \ldots, N_i, i = 1, 2, 3. \end{aligned} \tag{1}$$

8. Add the estimated background level $\hat{b}$ to the estimation $\Phi^{r_1,r_2;s_1,s_2,s_3}$ such that

$$\begin{aligned} P(k_1,k_2,k_3) &\approx P^{r_1,r_2;s_1,s_2,s_3}(k_1,k_2,k_3) := \Phi^{r_1,r_2;s_1,s_2,s_3}(k_1,k_2,k_3) + \hat{b}, \\ k_i &= 1, 2, \ldots, N_i, i = 1, 2, 3. \end{aligned} \tag{2}$$

Realization step

Let $P(i) \in \mathbb{R}^{p \times m}$, $i = 1, 2, \ldots, N$, be a finite 1D sequence.

1. Construct the $(N+1)p \times (N+1)m$ Hankel matrix

$$H = \begin{bmatrix} P(1) & P(2) & \cdots & P(N-1) & P(N) & 0 \\ P(2) & P(3) & \cdots & P(N) & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ P(N) & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix},$$

where 0 denotes a block of zeros of size $p \times m$.

2. Let $H = U\Sigma V$ be a singular value decomposition.
3. Partition $\Sigma = diag(\Sigma_1, \Sigma_2)$, $\Sigma_1 \in \mathbb{R}^{n \times n}$, $\Sigma_2 \in \mathbb{R}^{s \times s}$, $U = [U_1, U_2]$, $U_1 \in \mathbb{R}^{(N+1)p \times n}$, $U_2 \in \mathbb{R}^{(N+1)p \times s}$, and

$$V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix},$$

$V_1 \in \mathbb{R}^{n \times (N+1)m}$, $V_2 \in \mathbb{R}^{s \times (N+1)m}$, conformally. We also allow for the trivial partition in which the second components are empty, i.e. $s = 0$.

4. Let $C^s \in \mathbb{R}^{p \times n}$ be the first p rows of $U_1\Sigma_1^{1/2}$.
5. Let $B^s \in \mathbb{R}^{n \times m}$ be the first m columns of $\Sigma_1^{1/2}V_1$.
6. Let

$$U_1 = \begin{bmatrix} \bar{U}_{11}^1 \\ \vdots \\ \bar{U}_{N1}^1 \\ \bar{U}_{(N+1)1}^1 \end{bmatrix},$$

where $\bar{U}_{t_1 1}^1 \in \mathbb{R}^{p \times n}$ for all $t_1 = 1, \ldots, N+1$, and define

$$U_1^{\uparrow} = \begin{bmatrix} \bar{U}_{21}^1 \\ \vdots \\ \bar{U}_{(N+1)1}^1 \end{bmatrix} \quad \text{and} \quad U_1^{\downarrow} = \begin{bmatrix} \bar{U}_{11}^1 \\ \vdots \\ \bar{U}_{N1}^1 \end{bmatrix}.$$

Then let $A^s = \Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} \in \mathbb{R}^{n \times n}$.

Note that the symbols r_1, r_2, s_1, s_2 and s_3 denote the numbers of dropped singular values in each singular value decomposition. When $r_1 = r_2 = s_1 = s_2 = s_3 = 0$ we have $P = P^{0,0;0,0,0}$, i.e. a perfect reconstruction (Ober *et al.*, 2005). A detailed discussion of the algorithm is available in Ober *et al.* (2005). For the sake of simplifying the notation, we write $P^{rs} := P^{r_1,r_2;s_1,s_2,s_3}$, with $rs := \{r_1, r_2; s_1, s_2, s_3\}$.