Relaxed Ordered-Subsets Algorithm for Penalized-Likelihood Image Restoration

Saowapak Sotthivirat and Jeffrey A. Fessler

Department of Electrical Engineering and Computer Science,
University of Michigan, Ann Arbor, MI 48109

ssotthiv@eecs.umich.edu, fessler@eecs.umich.edu

The expectation-maximization (EM) algorithm for maximum-likelihood image recovery, although it provides the guarantee of convergence, has a slow convergence rate. Its ordered subset version (OS-EM), widely used in image reconstruction for tomography due to an order-of-magnitude acceleration over the EM algorithm, however, is not able to guarantee convergence. The recently proposed ordered subsets, separable paraboloidal surrogates (OS-SPS) algorithm with relaxation has been shown to converge to the optimal point while providing fast convergence. In this paper, we adapt the relaxed OS-SPS algorithm to the problem of image restoration. Because data acquisition is different in image restoration than in tomography, we employ a different strategy for choosing subsets using pixel location rather than projection angles. Simulation results show that the relaxed OS-SPS algorithm can achieve the same order-of-magnitude acceleration in restoration as in tomography. This new algorithm now provides image restoration with the speed and guaranteed convergence necessary for efficiency and improved resolution in image restoration.

OCIS codes: 100.0100,100.1830,100.2000,100.3020,100.3190,180.1790

1. Introduction

Statistical techniques have been shown to improve image quality in image restoration. Statistical methods can incorporate physical models of imaging systems, thus improving restoration. Moreover, object constraints, such as nonnegativity, can be easily enforced. Since closed form solutions for these techniques are usually unobtainable, iterative algorithms are needed.[1–5] Fast converging algorithms are desirable to quickly recover the original image. However, there are some drawbacks of existing algorithms, such as convergence, computation time, and parallelizability.

Expectation-maximization (EM) algorithms[6,7] and their ordered subset (OS) versions[8] are

among the most commonly used algorithms; however, they have limitations either of speed or convergence. The EM algorithms are guaranteed to converge; however, they converge very slowly. The OS-EM algorithm[8] has become very attractive to image reconstruction in tomography due to its fast convergence rate compared with the EM algorithms. It "converges" approximately M times faster than the EM algorithms, where M is the number of subsets. However, the OS-EM algorithm is not guaranteed to converge. After many iterations, the OS-EM algorithm oscillates rather than converges to an ML solution. Therefore, many approaches have been proposed to solve the convergence problem of the OS algorithm, such as the row-action maximum likelihood algorithm (RAMLA)[9] and its regularized version, the block sequential regularized EM (BSREM) algorithm.[10] Although the RAMLA and BSREM algorithms were proved to converge, they require a strong assumption that the objective sequence is convergent.

Recently, the relaxed ordered subsets separable paraboloidal surrogates (OS-SPS) algorithm[11] has been shown to converge under practical assumptions. This algorithm is derived from the separable paraboloidal surrogates (SPS) algorithm,[12,13] which is closely related to the EM algorithms. Like the EM algorithms, the OS version of the SPS (OS-SPS) algorithm[14] was introduced for transmission tomography. Even though the OS-SPS algorithm converges very fast, it is not guaranteed to converge. To fix the convergence problem of the OS-SPS algorithm, the relaxed OS-SPS algorithm[11] was proposed by introducing the relaxation parameter into the algorithm. This algorithm not only retains the fast convergence rate of the OS-SPS algorithm but is guaranteed to globally converge as well. Unlike the relaxed OS-SPS algorithm, the relaxed version of the OS-EM algorithm is not guaranteed to converge to the optimal point. Therefore, in this paper, we will implement the relaxed OS-SPS algorithm for image restoration.[15] Most existing OS methods have been applied to image reconstruction in tomography only, but not to image restoration; therefore, a different strategy for choosing subsets must be developed.

Since data acquisition necessary for image reconstruction in tomography is based on projection angles, the subsets obtained are not suitable for the pixel-based image restoration. Bertero and Boccacci applied the OS-EM method to the restoration of the large binocular telescope (LBT) images.[16] However, the structure of the LBT imaging is similar to that of the computed tomography (CT): multiple views of the same object have been observed at different angles. Thus, this technique cannot be applied to typical image restoration problems. In this paper, we focus on the more traditional image restoration problem of recovering a scene from a single blurred, noisy measured image under the simplifying assumption that the point spread function (PSF) is known. Instead of choosing subsets by downsampling projection angles as in tomography, for restoration, we choose subsets by downsampling pixels.

This paper is organized as follows. Section **2.** describes the measurement model and the objective function. The relaxed OS-SPS algorithm for image restoration is presented in Section **3.**. Subset design for restoration problems is discussed in Section **4.**. In Section **5.**, we develop some efficient implementation strategies, and quantify the computational complexity for the relaxed OS-SPS algorithm. Simulation results and performance of subset designs are presented in Section **6.**. Finally, conclusions are given in Section **7.**.

2. Measurement Model

In image restoration problems, the measurements are usually degraded by blur and noise. To recover the original image, one can use the statistical characteristics of the measurement system to specify an objective function that is to be maximized. Since image restoration is an ill-posed problem, we focus on penalized likelihood (PL) estimation using an objective function of the following form:

$$\Phi(x) = L(x) - \beta R(x), \tag{1}$$

where x denotes the image parameter vector to be estimated, L denotes the log-likelihood function of the measurement, R denotes the roughness penalty function, and β denotes a parameter that controls the degree of smoothness in the restored image.

For photon limited imaging (such as confocal microscopy), the noisy measurement Y can be modeled (approximately[17,18]) as follows:

$$Y_i \sim \text{Poisson}\{[Ax]_i + b_i\}, \quad i = 1, \ldots, N,$$

where A is the system matrix which is assumed to be known, b_i represents the background noise and dark current, and N is the number of pixels. The corresponding log-likelihood function is given by

$$L(x) = \sum_{i=1}^{N} \psi_i(l_i), \tag{2}$$

where $l_i = \sum_{j=1}^{P} a_{ij} x_j$ and $\psi_i(l) = y_i \log(l + b_i) - (l + b_i)$, ignoring irrelevant constants independent of x.

To reduce noise, we penalize the differences between neighboring pixels using a roughness penalty function of the form

$$R(x) = \sum_{i=1}^{r} \psi^R([Cx]_i),$$

where ψ^R is the potential function and C is the penalty matrix. For the first-order neighborhood, the matrix C consists of horizontal and vertical cliques. For example, with a 2×2 image, the matrix C can be written as follows:

$$Cx = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ x_4 - x_3 \\ x_3 - x_1 \\ x_4 - x_2 \end{bmatrix}$$

We assume that each potential penalty function $\psi^R(t)$ satisfies the following conditions:[13,19,20]

- ψ^R is symmetric.
- ψ^R is everywhere differentiable (and therefore continuous).
- $\dot{\psi}^R(t) = d/dt\psi^R(t)$ is convex.

- $\omega(t) = \frac{\dot{\psi}^R(t)}{t}$ is non-decreasing for $t \geq 0$.
- $\omega(0) = \lim_{t \to 0} \frac{\dot{\psi}^R(t)}{t}$ is finite and nonzero.

With proper regularization, the objective function has a unique global maximum. Thus, our goal is to estimate x by finding the maximizer of the objective function:

$$\hat{x} \stackrel{\triangle}{=} \arg\max_{x \geq 0} \Phi(x).$$

Because closed form solutions are unavailable for the maximizer, iterative algorithms are needed.

3. The Algorithms

Due to an order-of-magnitude acceleration over the EM algorithms, the OS-EM algorithm has been widely used in image reconstruction for tomography; however, the application to conventional image restoration has not been explored yet to our knowledge. The OS-EM algorithm is not guaranteed to converge even if relaxed.[11] Since the OS-SPS algorithm with relaxation is guaranteed to converge, our ordered subsets algorithm for image restoration will be based on the relaxed OS-SPS algorithm.

A. OS Technique

The objective function in (1) can be decomposed into sub-objective functions f_m as follows:

$$\Phi(x) = \sum_{m=1}^{M} f_m(x),$$

where M is the total number of subsets. Let $\{S_m\}_{m=1}^{M}$ be a disjoint partition of $\{1, \ldots, N\}$. Then f_m's are obtained by replacing a sum over all pixel indices in the likelihood function of (2) with a sum over a subset of data S_m as follows:

$$f_m(x) \stackrel{\triangle}{=} \sum_{i \in S_m} \psi_i(l_i(x)) - \frac{\beta}{M} R(x).$$

Suppose the "subset-balance"-like conditions[8] hold for the gradient of each sub-objective function:

$$\nabla f_1(x) \cong \nabla f_2(x) \cong \cdots \cong \nabla f_M(x).$$

Then the gradient of the objective function $\Phi(x)$ can be approximated as follows:

$$\nabla \Phi(x) \cong M \nabla f_m(x), \quad \forall m. \tag{3}$$

Using (3), one can replace $\nabla \Phi(x)$ with $M \nabla f_m(x)$ in any gradient-based algorithm to construct an OS-version of that algorithm. As previously discussed, the OS-EM is not guaranteed to converge even in the relaxed version.

B. OS-SPS

The separable paraboloidal surrogates (SPS) algorithm is based on paraboloidal surrogate functions[12,13] and the concavity technique developed by De Pierro.[7] The pixel update for the SPS algorithm can be summarized as follows:

$$x_j^{n+1} = \left[x_j^n + \frac{\nabla_j \Phi(x^n)}{\sum_{i=1}^m a_{ij}\gamma_i c_i^n + \beta p_j} \right]_+, \tag{4}$$

where, in the PL estimation,

$$\nabla_j \Phi(x) = \sum_{i=1}^N a_{ij} \dot{\psi}_i(l_i(x)) - \beta \sum_{i=1}^r c_{ij} \dot{\psi^R}([Cx]_i). \tag{5}$$

In (4), $\gamma_i = \sum_{j=1}^P a_{ij}$, and c_i^n is the following optimal curvature which guarantees convergence of SPS[13]

$$c_i^n = \begin{cases} \left[\frac{2}{(l_i^n)^2} \left(\psi_i(l_i^n) - \psi_i(0) - l_i^n \dot{\psi}_i(l_i^n) \right) \right]_+, & l_i^n > 0 \\ \left[-\ddot{h}_i(0) \right]_+, & l_i^n = 0. \end{cases}$$

By using the nonquadratic penalty function, its curvature p_j in (4) is given by

$$p_j \;=\; \sum_{i=1}^r c_{ij} \nu_i \omega(0),$$

where $\nu_i = \sum_{j=1}^p c_{ij}$, and $\omega(t) = \frac{\dot{\psi^R}(t)}{t}$.

Erdoğan and Fessler[14] introduced to the OS version of the SPS algorithm for transmission tomography. Based on (3), the gradient of the objective function in (5) is replaced by the sub-objective function multiplied by the number of subsets. We define $x^{(n+1,0)} = x^{(n,M)}$. Then the pixel update x_j for the OS-SPS algorithm is

$$x_j^{(n,m)} = \left[x_j^{(n,m-1)} + M \frac{\nabla_j f_m(x^{(n,m-1)})}{d_j + \beta p_j} \right]_+, \; m = 1, \ldots, M, \tag{6}$$

where

$$\nabla_j f_m(x) = \sum_{i \in S_m} a_{ij} \dot{\psi}_i(l_i(x)) - \frac{\beta}{M} \sum_{i=1}^r c_{ij} \dot{\psi^R}([Cx]_i).$$

In (6), the curvature of the likelihood d_j is precomputed as follows:

$$d_j \;=\; \sum_{i=1}^N a_{ij} \gamma_i c_i,$$

where $c_i = -\ddot{\psi}_i(y_i - b_i)$. Although the OS-SPS algorithm converges faster than SPS in early iterations, it is not guaranteed to converge.

C. Relaxed OS-SPS Algorithm

To guarantee the convergence of the OS-SPS algorithm, Ahn and Fessler[11] modified the OS-SPS algorithm to include relaxation. From (6), the pixel update of the relaxed OS-SPS algorithm becomes

$$x_j^{(n,m)} = \left[x_j^{(n,m-1)} + \alpha_n M \frac{\nabla_j f_m(x^{n,m-1})}{d_j + \beta p_j} \right]_+, \; m = 1, \ldots, M,$$

where a positive relaxation parameter α_n is chosen such that $\sum_n \alpha_n = \infty$ and $\sum_n \alpha_n^2 < \infty$. We use $\alpha_n = \frac{\xi}{(\xi-1)+n}$, where ξ is a positive constant, a "tuning parameter" that affects the rate of convergence and is chosen empirically. This algorithm is globally convergent.[11] The algorithm outline for relaxed OS-SPS algorithm is shown in Table 1.

D. Blind Restoration

Many blind restoration techniques have been applied to simultaneously restore the image and estimate the PSF.[21–24] The relaxed OS-SPS algorithm is applicable to blind restoration as well. In blind restoration technique, the image can be estimated by using the relaxed OS-SPS algorithm, whereas the PSF can be simultaneously estimated by using the ordinary SPS or EM algorithms due to a small number of parameters in the PSF.

4. Subset Design

Since most OS algorithms have been used for image reconstruction to date, a different strategy for choosing subsets in image restoration needs to be considered because of difference in data acquisition. A good choice of subsets should satisfy the "subset-balance" condition stated in (3). In tomography, the subsets are chosen from downsampling the projection angles. One possible approach for choosing the subsets in restoration problem is to downsample the pixels in the image. This approach seems to satisfy the "subset-balance" condition. Possible choices of four subsets for a 2D image are shown in Fig. 1. "4×1" OS-SPS stands for a downsampling design with 4 subsets along the row and 1 subset along the column as shown in Fig. 1a. We have found that dividing the image into large contiguous blocks (Fig. 2) tends to be a poor choice of subsets since it violates the "subset-balance" condition. "4×1B" OS-SPS stands for a block design with 4 subblocks along the row and 1 subblock along the column as shown in Fig. 2a. The analysis of how the possible choices of subsets may effect the convergence rate will be discussed in Section **6.**.

5. Implementation Techniques and Complexity

Because most computation time in the OS-SPS algorithm takes place in (11) and (12), in this section, we discuss how to efficiently implement these two expressions in the code for both space-variant and space-invariant systems.

A. Space-Variant Systems

A literal implementation of (11) and (12) in Table 1 would be appropriate for a shift-variant imaging system whose collection of PSFs is tabulated as a sparse set of a_{ij} values. By using this technique,

the computational complexity in the OS-SPS algorithm is indifferent from that in the nonordered subsets (nonOS) algorithm.

B. Space-Invariant Systems Using Convolution

For shift-invariant systems, however, one would typically implement (11) and (12) using convolution or FFT-based convolution in the conventional single-subset type of gradient-based iteration. Since these operations dominate the algorithm, it is essential to formulate efficient implementation of these two expressions. Computing all values of $\hat{l}$ using ordinary convolution would be inefficient when only some values of $\hat{l}$ are being used. Therefore, we introduced the following technique for computing (11) and (12) efficiently with convolution.

By assuming a space-invariant system, we rewrite (11) in the convolution form as follows:

$$\hat{l}_i = \sum_{j=1}^{P} h_{i-j} x_j, \quad \forall i \in S_m, \tag{7}$$

where h is the PSF. For illustration, 1D convolution is considered. Extension to 2D and 3D is straightforward. To efficiently compute some values of $\hat{l}$, we rewrite (7) using two summations:

$$\hat{l}_i = \sum_{m=1}^{M} \sum_{j \in S_m} h_{i-j} x_j, \quad \forall i \in S_m. \tag{8}$$

Using this expression, we can compute $\hat{l}_i$ for $i \in S_m$ by convolving the downsampled image and the PSF according to subset S_m, and then summing all the subsets (Fig. 3).

Likewise, to compute (12) efficiently by convolution, we can rewrite that expression as follows:

$$\dot{L}_j = \sum_{i \in S_m} h_{i-j} \dot{\psi}_i, \tag{9}$$

For each j, $\dot{L}_j$ can be computed by using $\dot{\psi}_i$ and a downsampled PSF. Different j's require a different downsampling of the PSF, but use the same $\dot{\psi}_i$'s (Fig. 4).

If implemented carefully, computational complexity for this convolution technique does not increase as the number of subsets increases.

C. Space-Invariant Systems Using FFT

For simultaneous update methods, such as the EM algorithms for image restoration, one can use FFT to reduce computation, especially for large 3D problems. Similarly, a strategy for using FFTs in the OS-SPS algorithm would be desirable to efficiently compute $\dot{L}_j$ and $\hat{l}_i$. One possible solution is to implement the partial FFT algorithm,[25] where only a small number of frequencies are evaluated. Since there is a specific pattern for computing $\hat{l}_i$ and $\dot{L}_j$ in each subset, rather than adapting and implementing this partial FFT technique into our algorithm, we develop the following technique based on the ordinary FFT algorithm, which should yield the same complexity but avoids implementing a new FFT code.

For illustration of our technique, 1D data and 2 subsets ($M = 2$) are considered. Let spatial indices be replaced by η to avoid confusion, $H(k)$ be an N-point FFT of h, and $X(k)$ be an N-point FFT of x. To compute $\hat{l}$ for 2 subsets using FFT, we reformulate the following expression:

$$\hat{l}(\eta) \;=\; \frac{1}{N}\sum_{k=0}^{N-1} H(k)X(k)\exp\left(j\frac{2\pi\eta k}{N}\right), \qquad \eta \in S_m.$$

Let $\eta = 0, \ldots, N/2-1$. Then the even indices of $\hat{l}$ belonging to subset 1 and the odd indices belonging to subset 2 are computed as follows:

$$m=1; \qquad \hat{l}(2\eta) \;=\; \frac{1}{N}\sum_{k=0}^{N/2-1} [H(k)X(k) + H(k+N/2)X(k+N/2)]\exp\left(j\frac{2\pi\eta k}{N/2}\right)$$

$$m=2; \quad \hat{l}(2\eta+1) \;=\; \frac{1}{N}\sum_{k=0}^{N/2-1} [H(k)X(k) - H(k+N/2)X(k+N/2)]\exp\left(j\frac{2\pi k}{N}\right)\exp\left(j\frac{2\pi\eta k}{N/2}\right)$$

In this technique, full N-point FFT is performed for h and x, but N/M-point inverse FFT (IFFT) is performed on $\hat{l}$ for each subset. Therefore, the number of complex multiplications in the OS-SPS using FFT for M subsets is given as follows:

$$\frac{MN}{2}\log_2 N + \frac{N}{2}\log_2\left(\frac{N}{M}\right) + MN + N - \frac{N}{M}, \tag{10}$$

whereas the number of complex multiplications in the nonOS algorithm is $N\log_2(2N)$ for computing $\hat{l}$. Table 2 shows the numerical comparison of complex multiplication between the OS and nonOS algorithms using FFT.

Although the number of complex multiplications increases as the number of subsets increases, it increases less rapidly than the number of subsets. Since the convergence rate increases roughly by a factor of number of subsets,[8,11,15] there is still an advantage for using FFTs in the OS-SPS algorithm, especially when N is large.

Similarly to $\hat{l}$, to compute $\dot{L}_j$ in (9) efficiently using FFT, we perform the following technique:

$$\dot{L}(\eta) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)\Psi(k)\exp\left(j\frac{2\pi\eta k}{N}\right), \qquad \forall\eta.$$

$\dot{L}$ is obtained by performing an N-point IFFT of the product of $H(k)$ and $\Psi(k)$; however, $H(k)$ and $\Psi(k)$ are computed from the reduced data given in each subset, i.e., even and odd sets of data for a 2-subset case. Thus, for $k = 0, \ldots, N/2-1$, N-point $H(k)$ and N-point $\Psi(k)$ for both subsets are computed as follows:

$$m=1; \qquad H(k) \;=\; \sum_{\eta=0}^{N/2-1} h(2\eta)\exp\left(-j\frac{2\pi\eta k}{N/2}\right) = H(k+N/2)$$

$$\Psi(k) \;=\; \sum_{\eta=0}^{N/2-1} \dot{\psi}(2\eta)\exp\left(-j\frac{2\pi\eta k}{N/2}\right) = \Psi(k+N/2)$$

$$m=2; \quad H(k) = \exp\left(-j\frac{2\pi k}{N}\right)\sum_{\eta=0}^{N/2-1} h(2\eta+1)\exp\left(-j\frac{2\pi\eta k}{N/2}\right) = -H(k+N/2)$$

$$\Psi(k) = \exp\left(-j\frac{2\pi k}{N}\right)\sum_{\eta=0}^{N/2-1} \dot{\psi}(2\eta+1)\exp\left(-j\frac{2\pi\eta k}{N/2}\right) = -\Psi(k+N/2).$$

Thus $N/2$-point FFTs are performed to obtain the first halves of $H(k)$ and $\Psi(k)$. In this case, the multiplication complexity for computing $\dot{L}$ is the same as the complexity for computing $\hat{l}$.

In the FFT technique described above, we only illustrate the techniques for radix-2 FFT. If the data sizes are not powers of 2, then zero-padding should be applied to avoid large prime factors.[26]

6. Simulation Results

This section illustrates the proposed algorithm using a 2D simulated data in comparison with existing algorithms. It also reports the characteristics of various subset choices as discussed in Section **4.**.

A. 2D Results

A 256×256 cell image (Fig. 5a) was degraded by a 15×15 PSF, created from the XCOSM package[1] (only the central xz plane is used to clearly show elongation of the PSF in the z direction),[27] and Poisson noise with PSNR[2] of 40 dB, as shown in Fig. 5b. We assigned the relaxation parameter $\alpha_n = 11/(10+n)$ and, for edge-preserving,[28] we used the nonquadratic roughness penalty function $\psi^R(t) = \delta^2\left[\left|\frac{t}{\delta}\right| - \log\left(1+\left|\frac{t}{\delta}\right|\right)\right]$, where δ controls the degree of edge preservation. Fig. 5c shows the restoration with the relaxed OS-SPS algorithm (8 subsets) performed for 50 iterations.

Table 3 compares the elapsed time per iteration of different algorithms: De Pierro's modified EM (DPEM),[7] SPS (with optimal curvature), and relaxed OS-SPS (with precomputed curvature) algorithms. Theoretically, different subsets of the relaxed OS-SPS algorithm (using the convolution technique described in Section **5.B.**) should yield approximately the same computation time per iteration as the nonOS version. Although, we were not able to achieve that due to MATLAB overhead, the computation time per iteration increases less rapidly than the number of subsets. A better way to compare the complexity of the OS-SPS algorithm with the nonOS version is to calculate the number of floating point operations (FLOPs). Table 3 shows that the numbers of FLOPs required in the OS-SPS algorithms are slightly different from the number of FLOPs required in the SPS algorithm. This agrees with our discussion given in Section **5.B.**.

Fig. 6 shows the objective increase, $\Phi(x^n) - \Phi(x^0)$, at each iteration of DPEM, SPS, ordinary OS-SPS (8 subsets), and relaxed OS-SPS (8 subsets). In this figure, both ordinary OS-SPS and relaxed OS-SPS algorithms increase the objective function faster than the DPEM algorithm roughly

[1]pixel sizes $\triangle x = \triangle y = \triangle z = 0.15\mu$m, 40× /1.0 NA oil-immersion objective, and a fluorescent wavelength of 0.63 μm.

[2]The peak signal-to-noise ratio is defined as follows:

$$\text{PSNR} = 10\log_{10}\left(\frac{\max_i (y_i - b_i)^2}{\frac{1}{N}\sum_i (y_i - \text{E}[y_i])^2}\right).$$

by the number of subsets. However, the relaxed OS-SPS algorithm is guaranteed to eventually converge to the optimal point, unlike the original OS-SPS algorithm. Fig. 7 compares the convergence rates for different number of subsets. The relaxed OS-SPS-16 yields the fastest convergence rate.

B. Sebset Design

One's choice of subsets can effect the convergence rate of the algorithm. Fig. 8 shows the plots of objective increase versus number of iterations for different choices of subsets (Figs. 1-2) using the relaxed OS-SPS algorithm. In this figure, the subsets with the subblock approach show a poor unpredictable behavior at the beginning of iterations; however, the relaxed OS-SPS algorithm using these subsets eventually converge to the optimal point. Without relaxation, they might get stuck at some suboptimal point. This unpredictable behavior is due to the violation of the "subset-balance" condition and it does not appear in the case of the downsampling approach. Therefore, the downsampling approach is more desirable for obtaining a "well-behaved" objective function. With the downsampling approach, different designs of subsets provide almost the same convergence rate and similar number of FLOPs. Thus, the choice of subsets does not affect the convergence rate much as long as the downsampling approach is used.

7. Conclusion

We demonstrated that the relaxed OS-SPS algorithm, conventionally used for tomography, can be adapted to use in image restoration by choosing appropriate subsets. Essentially, we based this choice on the pixel location. As long as the subsets are chosen from downsampling the pixels, different choices of subsets hardly effect the convergence rate of the algorithm. Similarly to tomography, we are able to achieve order-of-magnitude acceleration over the nonOS version algorithm. The computational complexity of the OS-SPS algorithm using the convolution approach described in Section **5.B.** is theoretically the same for all the numbers of subsets. Although the FFT approach described in Section **5.C.** increases the computational complexity of the algorithm when the number of subset increases, the overall convergence rate is still faster than the nonOS algorithm.

References

1. R. Richardson, "Bayesian-based iterative method of image restoration," J. Opt. Soc. Am. **62**, 55–9 (1972).

2. T. J. Holmes, "Maximum-likelihood image restoration adapted for noncoherent optical imaging," J. Opt. Soc. Am. A **5**, 666–673 (1988).

3. S. Joshi and M. I. Miller, "Maximum a posteriori estimation with Good's roughness for three-dimensional optical-sectioning microscopy," J. Opt. Soc. Am. A **10**, 1078–85 (1993).

4. R. G. Lane, "Methods for maximum-likelihood deconvolution," J. Opt. Soc. Am. A **13**, 1992–8 (1996).

5. J. A. Conchello and J. G. McNally, "Fast regularization technique for expectation maximization algorithm for computational optical sectioning microscopy," in *Three-dimensional and multidimensional microscopy: Image Acquisition and Processing*, C. J. Cogswell, G. S. Kino, and T. Wilson, Eds. Proc. SPIE **2655**, 199–208 (1996).

6. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," J. Roy. Stat. Soc. B **39**, 1–38 (1977).

7. A. R. De Pierro, "A Modified Expectation Maximization Algorithm for Penalized Likelihood Estimation in Emission Tomography," IEEE Trans. Med. Imaging **14**, 132–137 (1995).

8. H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," IEEE Trans. Med. Imaging **13**, 601–609 (1994).

9. J. A. Browne and A. R. De Pierro, "A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography," IEEE Trans. Med. Imaging **15**, 687–699 (1996).

10. A. R. De Pierro and M. E. B. Yamagishi, "Fast EM-like methods for maximum 'a posteriori' estimates in emission tomography," IEEE Trans. Med. Imaging **20**, 280–288 (2001).

11. S. Ahn and J. A. Fessler, "Globally Convergent Ordered Subsets Algorithms: Application to Tomography," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.* (2001), pp. M1–2.

12. J. A. Fessler and H. Erdoğan, "A Paraboloidal Surrogates Algorithm for Convergent Penalized-Likelihood Emission Image Reconstruction," in *Proc. IEEE Nuc. Sci. Symp. and Med. Im. Conf.* **2**

(1998), pp. 1132–1135.

13. H. Erdoğan and J. A. Fessler, "Monotonic Algorithms for Transmission Tomography," IEEE Trans. Med. Imaging **18**, 801–814 (1999).

14. H. Erdoğan, G. Gualtieri, and J. A. Fessler, "An Ordered Subsets Algorithm for Transmission Tomography," Phys. Med. Biol. **44**, 2835–51 (1999).

15. S. Sotthivirat and J. A. Fessler, "Relaxed ordered subsets algorithm for image restoration of confocal microscopy," in *Proc. IEEE Int'l Symp. on Biomedical Imaging* (2002), To appear.

16. M. Bertero and P. Boccacci, "Application of the OS-EM method to the restoration of LBT images," Astron. Astrophys. Suppl. Ser. **144**, 181–186 (2000).

17. D. L. Snyder, A. M. Hammoud, and R. L. White, "Image recovery from data acquired with a charge-coupled-device camera," J. Opt. Soc. Am. A **10**, 1014–1023 (1993).

18. D. L. Snyder, C. W. Helstrom, A. D. Lanterman, M. Faisal, and R. L. White, "Compensation for readout noise in CCD images," J. Opt. Soc. Am. A **12**, 272–283 (1995).

19. P. J. Huber, *Robust Statistics*, (Wiley, New York, 1981).

20. J. A. Fessler, "Grouped Coordinate Descent Algorithms for Robust Edge-Preserving Image Restoration," in *Image reconstruction and restoration II*, T. J. Schulz, Eds. Proc. SPIE **3170**, 184–194 (1997).

21. T. J. Holmes, "Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach," J. Opt. Soc. Am. A **9**, 1052–1061 (1992).

22. E Thiébaut and J. M. Conan, "Strict a priori constraints for maximum-likelihood blind deconvolution," J. Opt. Soc. Am. A **12**, 485–92 (1995).

23. J. Markham and J. A. Conchello, "Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur," J. Opt. Soc. Am. A **16**, 2377–2391 (1999).

24. E. Y. Lam and J. W. Goodman, "Iterative statistical approach to blind image deconvolution," J. Opt. Soc. Am. A **17**, 1177–84 (2000).

25. J. C. D. de Melo, "Partial FFT Evaluation," in Int'l Conf. on Signal Processing Application and Technology **1**, 134–141 (1996), Available: http://www.icspat.com/papers/234amfi.pdf.

26. S. K. Mitra, *Digital signal procesing: A computer-based approach*, (McGraw-Hill, New York, 2001,2nd edition).

27. The XCOSM deconvolution package [Online]. Available: http://3dmicroscopy.wustl.edu/∼xcosm/.

28. K. Lange, "Convergence of EM Image Reconstruction Algorithms with Gibbs Smoothing," IEEE Trans. M. Imaging **9**, 439–446 (1990).

Fig. 1. These subsets tend to satisfy the "subset-balance" condition. The first number in the quotation marks is the number of subsets along the row and the second number is the number of subsets along the column. The total number of subsets is the product of these two numbers. The pixel label m belongs to the respective set S_m.

Fig. 2. These subsets tend to violate the "subset-balance" condition. The first number in the quotation marks is the number of subblocks along the row and the second number is the number of subblocks along the column.

Fig. 3. Compute $\hat{l}_i, \forall i \in S_m$, using all information of x and h. The symbol * represents convolution. The white blocks denote elements of x belonging to subset $m = 1$ and the stripe blocks denote elements of x belonging to subset $m = 2$.

Fig. 4. Compute $\dot{L}_j, \forall j$, using some information of $\dot{\psi}_i$.

Fig. 5. Simulated images and restoration using the relaxed OS-SPS algorithm with $\beta = 10^{-6}$ and $\delta = 100$. The PSF in the noisy blurry image was simulated from the 2D PSF of the confocal microscope in the xz direction, where x is along the horizontal axis and z is along the vertical axis, to show elongation in the z direction. This elongation disappears in the restored image.

Fig. 6. Comparison of objective function increase of DPEM, SPS, OS-SPS, and relaxed OS-SPS algorithms. The nonrelaxed and relaxed OS-SPS algorithms have the order-of-magnitude acceleration over the DPEM and SPS algorithms.

Fig. 7. Comparison of objective function increase versus elapsed time of relaxed OS-SPS algorithms with different numbers of subsets. The 16-subset OS-SPS yield the fastest convergence rate.

Fig. 8. Comparison of different choices of subsets using relaxed OS-SPS algorithm. The subset unbalance of the relaxed OS-SPS with the subblock design causes an unpredictable behavior of the objective function increase at the beginning of iterations but the algorithm eventually converges due to relaxation. The relaxed OS-SPS algorithms with downsampling design converge almost at the same rate for different choices of subsets.

Possible Choices for Four Subsets by a Downsampling Approach

1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4

a) "4x1" OS−SPS

1	2	3	4	1	2	3	4
1	2	3	4	1	2	3	4
1	2	3	4	1	2	3	4
1	2	3	4	1	2	3	4
1	2	3	4	1	2	3	4
1	2	3	4	1	2	3	4
1	2	3	4	1	2	3	4
1	2	3	4	1	2	3	4

a) "1x4" OS−SPS

1	3	1	3	1	3	1	3
2	4	2	4	2	4	2	4
1	3	1	3	1	3	1	3
2	4	2	4	2	4	2	4
1	3	1	3	1	3	1	3
2	4	2	4	2	4	2	4
1	3	1	3	1	3	1	3
2	4	2	4	2	4	2	4

a) "2x2" OS−SPS

Figure 1, Sotthivirat and Fessler

Possible Choices for Four Subsets by a Block Approach

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4

a) "4x1B" OS−SPS

1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4

a) "1x4B" OS−SPS

1	1	1	1	3	3	3	3
1	1	1	1	3	3	3	3
1	1	1	1	3	3	3	3
1	1	1	1	3	3	3	3
2	2	2	2	4	4	4	4
2	2	2	2	4	4	4	4
2	2	2	2	4	4	4	4
2	2	2	2	4	4	4	4

a) "2x2B" OS−SPS

Figure 2, Sotthivirat and Fessler

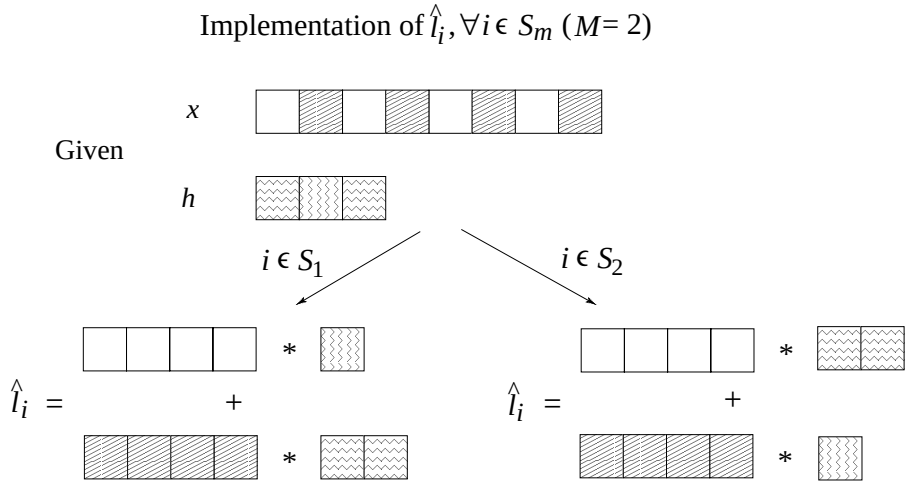


Figure 3, Sotthivirat and Fessler

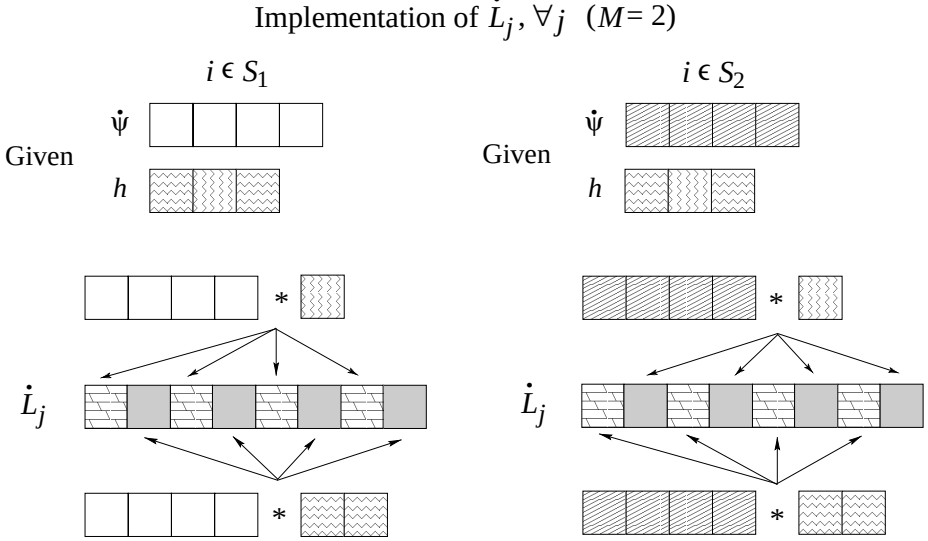
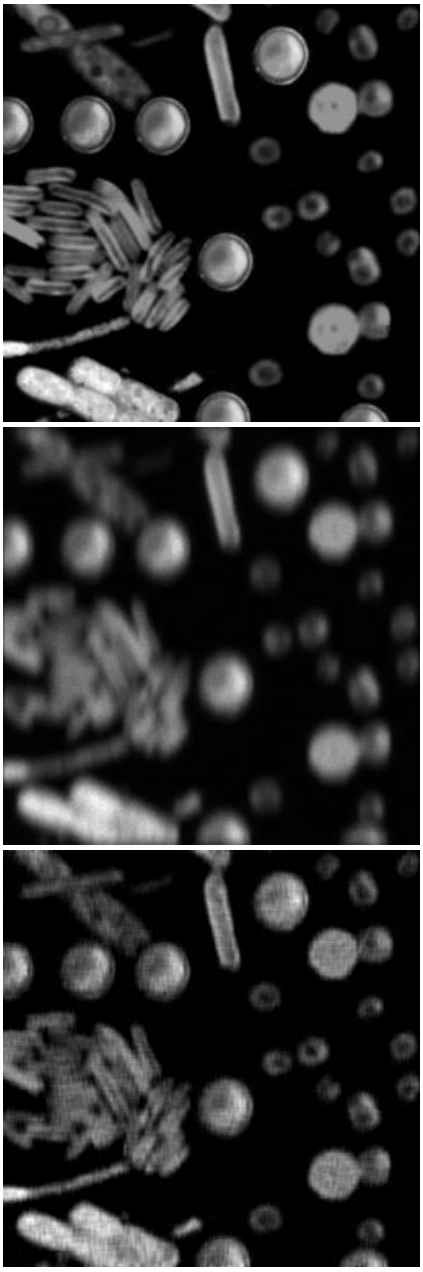


Figure 4, Sotthivirat and Fessler



a. Original image
b. Noisy blurry image
c. Restored image

Figure 5, Sotthivirat and Fessler

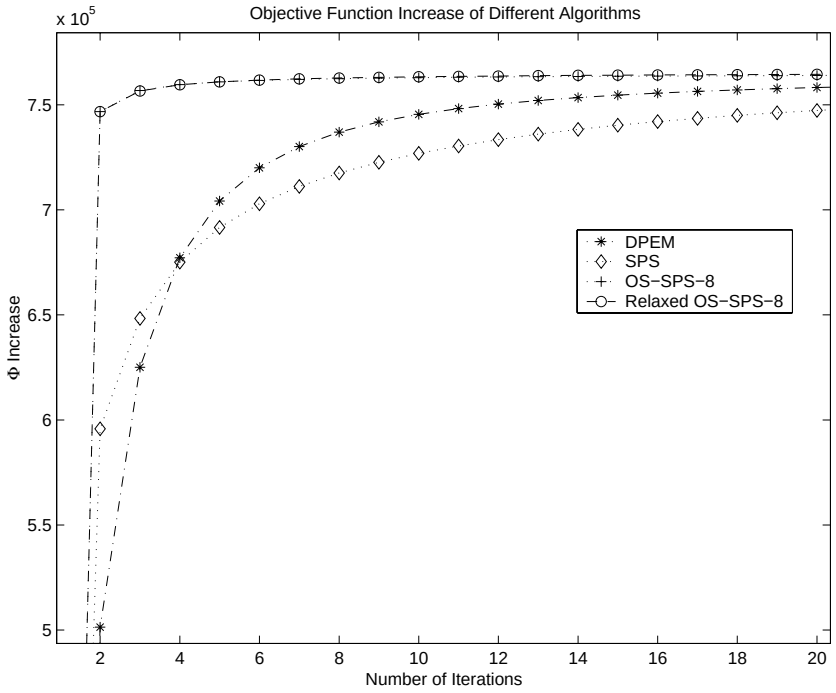


Figure 6, Sotthivirat and Fessler

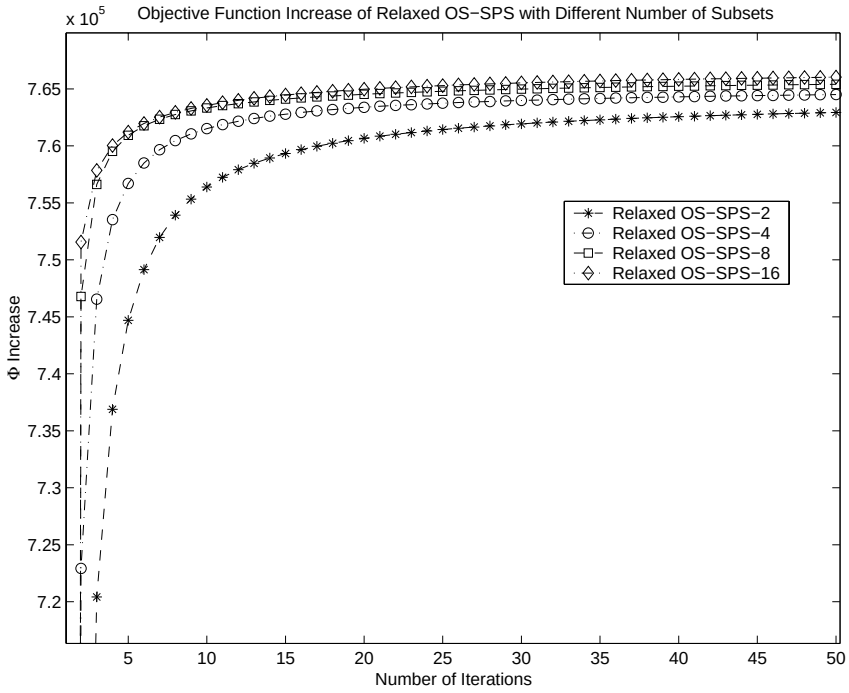


Figure 7, Sotthivirat and Fessler

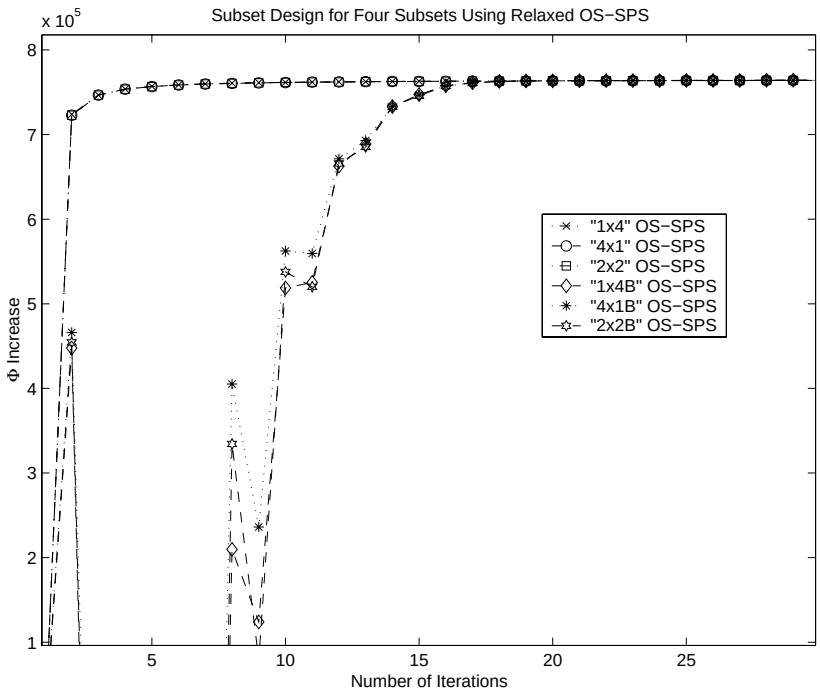


Figure 8, Sotthivirat and Fessler

Table 1. Relaxed OS-SPS Algorithm Outline.

Precompute:

$d_j = \sum_{i=1}^{N} a_{ij}\gamma_i c_i$

$p_j = \sum_{i=1}^{r} c_{ij}\nu_i\omega(0)$

for $n = 1, \ldots,$Niters

$\alpha_n = \frac{\xi}{(\xi-1)+n}$

for $m = 1, \ldots, M$

$$\hat{l}_i = \sum_{j=1}^{P} a_{ij} x_j^{(n,m-1)}, \quad \forall i \in S_m \tag{11}$$

$\dot{\psi}_i = \frac{y_i}{\hat{l}_i + b_i} - 1, \quad \forall i \in S_m$

for $j = 1, \ldots, P$

$$\dot{L}_j = \sum_{i \in S_m} a_{ij}\dot{\psi}_i \tag{12}$$

$\dot{R}_j = \sum_{i=1}^{r} c_{ij}\dot{\psi}^R([Cx^{(n,m-1)}]_i)$

$x_j^{(n,m)} = \left[x_j^{(n,m-1)} + \alpha_n M \frac{\dot{L}_j - \frac{\beta}{M}\dot{R}_j}{d_j + \beta p_j} \right]_+$

end

end

end

Table 2. Comparison of multiplication complexity ratio between OS-SPS and nonOS algorithms in computing $\hat{l}_i$ (using FFT)

Number of data points	Number of subsets	Complexity ratio between OS and nonOS
64	2	1.57
	4	2.68
	8	4.91
512	2	1.55
	4	2.62
	8	4.79

Table 3. Comparison of elapsed times per iteration and number of FLOPs for DPEM, SPS, and OS-SPS algorithms.

	Time/iter (s)	Time comparison	Number of FLOPs	FLOPs comparison
DPEM	1.03	0.92	84,937,142	0.92
SPS	1.12	1	92,406,026	1
OS-SPS-2	1.23	1.10	92,522,010	1.00
OS-SPS-4	1.86	1.66	95,944,812	1.04
OS-SPS-8	3.65	3.26	102,919,258	1.11
OS-SPS-16	6.83	6.10	116,976,572	1.27