

Image Recovery Using Partitioned-Separable Paraboloidal Surrogate Coordinate Ascent Algorithms

Saowapak Sotthivirat and Jeffrey A. Fessler

Dept. of Electrical Engineering and Computer Science, University of Michigan

4415 EECS Bldg., 1301 Beal Ave., Ann Arbor, MI 48109-2122

Email: ssothtiv@eecs.umich.edu, Phone: (734) 647-8390, Fax: (734) 764-8041

Abstract—Iterative coordinate ascent algorithms have been shown to be useful for image recovery, but are poorly suited to parallel computing due to their sequential nature. This paper presents a new fast converging parallelizable algorithm for image recovery that can be applied to a very broad class of objective functions. This method is based on paraboloidal surrogate functions and a concavity technique. The paraboloidal surrogates simplify the optimization problem. The idea of the concavity technique is to partition pixels into subsets that can be updated in parallel to reduce the computation time. For fast convergence, pixels within each subset are updated sequentially using a coordinate ascent algorithm. The proposed algorithm is guaranteed to monotonically increase the objective function and intrinsically accommodates nonnegativity constraints. A global convergence proof is summarized. Simulation results show that the proposed algorithm requires less elapsed time for convergence than iterative coordinate ascent algorithms. With four parallel processors, the proposed algorithm yields a speedup factor of 3.77 relative to single processor coordinate ascent algorithms for a 3-D confocal image restoration problem.

Keywords—Image restoration, maximum likelihood estimation, coordinate ascent algorithm, confocal microscopy.

I. INTRODUCTION

Statistical methods such as maximum likelihood (ML), penalized maximum likelihood (PML), and maximum *a posteriori* (MAP) estimation have been widely applied to recover degraded images. Because closed form solutions are usually unavailable, iterative maximization algorithms are needed. This paper describes a new fast monotonic algorithm for image recovery that is well suited to parallel computing.

Many algorithms for PML/MAP image recovery have been constructed; however, no existing algorithm has all the properties of an “ideal” algorithm such as fast convergence rate, quick computation time, stability, simplicity, parallelizability, etc. Expectation-maximization (EM) algorithms [1, 2] are widely used to compute ML estimates. Although EM algorithms are simple to implement and guaranteed to converge, they converge slowly since they simultaneously update all parameters. Fast converging algorithms are particularly desirable when large 3-D images are used or when time becomes an important issue such as in medical imaging and microscopy. Several algorithms have been proposed to improve the convergence rate. One example is the space-alternating generalized EM (SAGE) algorithm [3, 4] that converges quickly but is typically nonparallelizable. Similarly, the classical coordinate ascent algorithm, which updates parameters sequentially each iteration, is nonparallelizable, and furthermore does not have an explicit form for the update. To obtain a closed form for the update, one can use a coordinate ascent algorithm with Newton-Raphson

updates (CA-NR) [5]. However, the CA-NR algorithm is not guaranteed to converge if the objective function is nonquadratic. The paraboloidal surrogate coordinate ascent (PSCA) algorithm [6] solves the convergence problem of the CA-NR algorithm by maximizing paraboloidal surrogate functions instead of directly maximizing the original objective function. However, the PSCA algorithm is still not parallelizable. In summary, existing algorithms are either fast converging, as in the CA-NR or PSCA algorithms, or fully parallelizable, as in the EM algorithms, but not both.

This paper presents a new, fast converging, parallelizable algorithm called partitioned-separable paraboloidal surrogate coordinate ascent (PPCA). This new approach overcomes the convergence rate and parallelizability tradeoff of existing algorithms [7]. To provide parallelizability, we partition the set of pixels into subsets that are updated in parallel, usually by a different processor for each subset to reduce execution time. To provide fast convergence, each processor *sequentially* updates the pixels *within* each subset. This approach captures most of the rapid convergence of the CA algorithm, but remains parallelizable. A simplistic implementation of this idea would not ensure convergence; therefore, we derive the algorithm by applying optimization transfer principles. This approach guarantees that the proposed algorithm will monotonically increase the objective function. It also intrinsically accommodates the nonnegativity constraints. The PSCA algorithm of [6] is the special case of the PPCA algorithm when only one subset (and hence only one processor) is used.

The grouped coordinate ascent (GCA) algorithm derived in [8–10] represents an alternative approach to parallelization. The GCA algorithm simultaneously updates pixels within a group of spatially separated pixels and sequentially updates each group of pixels. This approach does not fully capture the fast convergence properties of CA, and thus the GCA algorithm converges slower than the proposed PPCA algorithm¹.

The parallel successive overrelaxation (PSOR) method [11] using domain decomposition techniques was proposed for solving the five-point and nine-point stencil approximation of Poisson’s equations, but it is inapplicable to the more general optimization problem of interest in imaging.

We implemented the proposed algorithm using direct con-

¹Matlab m-files for comparison are available at <http://www.eecs.umich.edu/~fessler>.

volution rather than fast Fourier transform (FFT) algorithms, so that it easily extends to problems where the space-invariant property is inapplicable or invalid such as in positron emission tomography (PET) and single photon emission computed tomography (SPECT) systems. In microscopy, many papers such as [12, 13] have assumed space-invariance of the microscope, and thus EM algorithms using FFT techniques have been applied to reduce the computation time. However, since our long-term interest is space-varying systems, we derived the proposed algorithm in image space rather than frequency space.

This paper is organized as follows. Section II describes the image recovery problem. Section III reviews some existing algorithms. Section IV presents the proposed algorithm in a general form suitable for many applications. Section V discusses convergence of this algorithm. Section VI compares the convergence rate of the proposed algorithm with other algorithms. In Section VII, the proposed algorithm is specifically applied to image restoration for confocal microscopy. Since a 3-D image from a confocal microscope has poor resolution, especially in the axial direction, due to out-of-focus contributions from other planes, image restoration techniques have been applied to remove the out-of-focus contributions and reduce elongation in the axial direction. Simulation results are presented in Section VIII and conclusions are given in Section IX. Since several algorithms are mentioned in this paper, for convenience, we summarize their acronyms and description in Table I.

II. THE PROBLEM

In image recovery 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 maximized. In this paper, we consider a very broad class of objective functions having the following form:

$$\Phi(x) = \sum_{i=1}^m \psi_i([Bx]_i), \quad (1)$$

where $x \in \mathbb{R}^p$ represents the true image and B is an $m \times p$ matrix that typically includes both an $m_L \times p$ system matrix and an $m_R \times p$ coefficient matrix of a roughness penalty function where $m = m_L + m_R$, m_L is the number of measurements, and m_R is roughly the number of neighbors of pixels. For $i = 1, \dots, m_L$, each ψ_i function characterizes the agreement between a noisy measurement and a linear function of the unknown image, namely $[Bx]_i = \sum_{j=1}^p b_{ij}x_j$. For $i = m_L + 1, \dots, m$, each ψ_i function corresponds to the roughness penalty function due to the ill-posed nature of the problem. Section VII shows a concrete example where some of the ψ_i functions correspond to a Poisson log-likelihood function, which describes fluorescent photons detected at a photodetector in a confocal microscope system, and the remaining ψ_i functions represent a nonquadratic penalty function. We assume that the objective function has a unique global maximum. Thus our goal is to estimate x by finding the maximizer of the objective function as follows:

$$\hat{x} \triangleq \arg \max_{x \geq 0} \Phi(x). \quad (2)$$

The ML, PML, and MAP estimators are all special cases of this maximization problem. We focus on the nonnegativity constrained problem, but of course all the algorithms are also applicable to the unconstrained case.

III. PREVIOUS ALGORITHMS

Many existing algorithms have been applied to obtain a maximizer of $\Phi(x)$ in (2). Generally there is a tradeoff between convergence rate and parallelizability. Although EM algorithms are guaranteed to converge to at least a local maximum, they converge very slowly. However, EM algorithms are usually fully parallelizable. At the other extreme, the CA algorithm, which updates the unknown parameters sequentially, converges much faster than the EM algorithms. However, the CA algorithm is not parallelizable.

Directly maximizing the objective function in (2) is difficult when ψ_i 's are nonquadratic, such as for the log-likelihood function of Poisson noise. To simplify the optimization problem and to assure monotonic increases in the objective function at each iteration, one can apply an optimization transfer approach by finding a ‘‘surrogate’’ function ϕ (Fig. 1) that lies below the objective function.

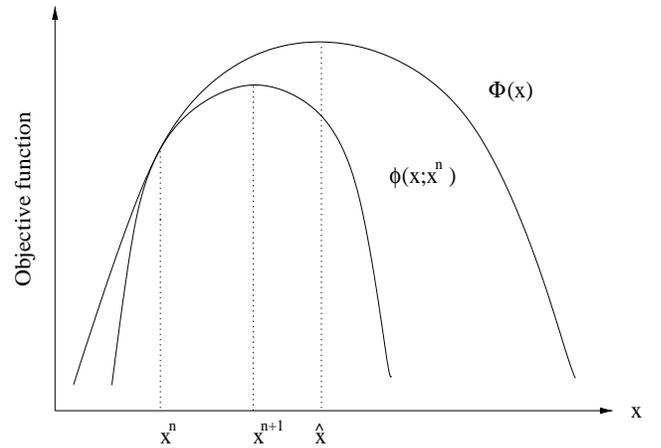


Fig. 1. Illustration of a surrogate function ϕ

As illustrated in Fig. 1, in the optimization transfer function approach, we obtain the next estimate by maximizing the surrogate function²:

$$x^{n+1} \triangleq \arg \max_{x \geq 0} \phi(x; x^n), \quad (3)$$

where x^n denotes the estimate at the n th iteration. Choosing a surrogate function ϕ that satisfies the following monotonicity condition ensures that the iterates x^n will monotonically increase the objective function Φ [6, 9, 10]:

$$\Phi(x) - \Phi(x^n) \geq \phi(x; x^n) - \phi(x^n; x^n), \quad \forall x \geq 0. \quad (4)$$

Rather than using (4), we choose surrogate functions $\phi(x; x^n)$ that satisfy the following sufficient conditions:

1. $\phi(x^n; x^n) = \Phi(x^n)$

²In practice, it usually suffices for the next iteration to simply increase the surrogate function rather than requiring a strict maximization of ϕ .

Acronym	Algorithm Name	Parallelizability	Speed	Guaranteed to Converge
EM	Expectation-maximization algorithm [1, 2]	Yes	Slow	Yes
SAGE	Space-alternating generalized expectation-maximization [3, 4]	No	Medium	Yes
SPS	Separable paraboloidal surrogates [2]	Yes	Slow	Yes
PSCA	Paraboloidal surrogate coordinate ascent [6]	No	Fast	Yes
CA-PS	Coordinate ascent with parabola surrogates	No	Slow	Yes
GCA	Grouped coordinate ascent [8–10]	Partially	Medium	Yes
PPCA	Partitioned-separable paraboloidal surrogate coordinate ascent [7]	Almost fully	Fast	Yes
CA-NR	Coordinate ascent with Newton-Raphson updates [5]	No	Fast	No
CA,P	Coordinate ascent with precomputed curvature	No	Fast	No
PSCA,P	PSCA with precomputed curvature [6]	No	Fast	No
PPCA,P	PPCA with precomputed curvature	Almost fully	Fastest	No

TABLE I

ACRONYMS AND DESCRIPTION OF DIFFERENT ALGORITHMS. “SPEED” REFERS QUALITATIVELY TO THE COMBINATION OF NUMBER OF ITERATIONS AND EXECUTION TIME PER ITERATION.

2. $\phi(x; x^n) \leq \Phi(x)$ for $x \geq 0$
3. $\frac{\partial}{\partial x_j} \phi(x; x^n) \Big|_{x=x^n} = \frac{\partial}{\partial x_j} \Phi(x) \Big|_{x=x^n}, \quad \forall j$.

The third condition follows from the other conditions for differentiable surrogate and objective functions.

The following subsections summarize some existing algorithms illustrating the convergence rate and parallelizability tradeoff.

A. Separable Paraboloidal Surrogates (SPS) Algorithm

To obtain a fully parallelizable algorithm, the surrogate function ϕ should be separable so that we can simultaneously update the unknown parameters. Like the EM algorithms, the SPS algorithm is fully parallelizable. It is derived by using the concavity technique developed by De Pierro [2].

To derive the SPS algorithm, we begin by considering the following quadratic surrogate function:

$$\Phi(x) \geq Q(x; x^n) \triangleq \sum_{i=1}^m q_i([Bx]_i; [Bx^n]_i). \quad (5)$$

The separable paraboloidal surrogate function Q lies below the objective function and is constructed from the parabola q_i having the following form:

$$q_i(t; t_i^n) \triangleq \psi_i(t_i^n) + \dot{\psi}_i(t_i^n)(t - t_i^n) - \frac{1}{2}c_i(t_i^n)(t - t_i^n)^2, \quad (6)$$

where $t_i^n \triangleq [Bx^n]_i = \sum_{j=1}^p b_{ij}x_j^n$, $\dot{\psi}_i$ denotes the first derivative of ψ_i , and $c_i(t_i^n)$ represents the curvature of the parabola $q_i(t; t_i^n)$. According to the sufficient conditions previously mentioned, we choose the parabola in (6) to satisfy the following conditions:

1. $q_i(t_i^n; t_i^n) = \psi_i(t_i^n)$
2. $q_i(t; t_i^n) \leq \psi_i(t), \quad \forall t \in \{[Bx]_i : x \geq 0\}$
3. $\dot{q}_i(t_i^n; t_i^n) = \dot{\psi}_i(t_i^n)$.

To obtain the fastest convergence rate while still guaranteeing monotonicity, we choose the optimal curvature [6] as follows:

$$c_i^{\text{opt}}(t_i^n) = \min\{c \geq 0 : \psi_i(t) \geq \psi_i(t_i^n) + \dot{\psi}_i(t_i^n)(t - t_i^n) - \frac{c}{2}(t - t_i^n)^2\}. \quad (7)$$

As shown in [6], if ψ_i is strictly concave and $\dot{\psi}_i(t)$ is strictly convex for $t \geq 0$, then a parabola q_i of form (6) exists and satisfies the above three conditions. Therefore, the optimal curvature

can be expressed as follows:

$$c_i^{\text{opt}}(t_i^n) = \begin{cases} \left[\frac{2}{(t_i^n)^2} \left(\psi_i(t_i^n) - \psi_i(0) - t_i^n \dot{\psi}_i(t_i^n) \right) \right]_+, & t_i^n > 0 \\ \left[-\ddot{\psi}_i(0) \right]_+, & t_i^n = 0. \end{cases} \quad (8)$$

The symbol $[g]_+$ represents g if $g \geq 0$ and 0 if $g \leq 0$. These conditions on ψ_i are only *sufficient* conditions. There are many other ψ_i 's for which q_i exists such as the broad family potential function discussed above (40) in Section VII.

To construct a separable surrogate function, we apply the additive concavity technique developed by De Pierro [2] to the quadratic surrogate functions. First, we rewrite the argument $[Bx]_i$ in (1) as follows:

$$[Bx]_i = \sum_{j=1}^p \pi_{ij} \left(\frac{b_{ij}}{\pi_{ij}}(x_j - x_j^n) + [Bx^n]_i \right), \quad (9)$$

where the π_{ij} 's are any nonnegative³ constants for which $\sum_{j=1}^p \pi_{ij} = 1, \forall i$. A simple choice is

$$\pi_{ij} = \frac{|b_{ij}|}{\sum_{j=1}^p |b_{ij}|}. \quad (10)$$

Since each q_i is concave,

$$q_i([Bx]_i; t_i^n) \geq \sum_{j=1}^p \pi_{ij} q_i \left(\frac{b_{ij}}{\pi_{ij}}(x_j - x_j^n) + t_i^n; t_i^n \right). \quad (11)$$

Thus from (5), the separable paraboloidal surrogate function ϕ is obtained as follows:

$$Q(x; x^n) \geq \phi(x; x^n) \triangleq \sum_{j=1}^p Q_j(x_j; x^n),$$

where

$$Q_j(x_j; x^n) = \sum_{i=1}^m \pi_{ij} q_i \left(\frac{b_{ij}}{\pi_{ij}}(x_j - x_j^n) + t_i^n; t_i^n \right).$$

Since Q_j is quadratic, we implement the maximization (3) by using Newton's method:

$$x_j^{n+1} \triangleq \arg \max_{x_j \geq 0} Q_j(x_j; x^n)$$

³ $\pi_{ij} = 0$ only if $b_{ij} = 0$.

$$= \left[x_j^n + \frac{\frac{d}{dx_j} Q_j(x_j; x^n) \Big|_{x_j=x_j^n}}{-\frac{d^2}{dx_j^2} Q_j(x_j; x^n)} \right]_+ \quad (12)$$

where

$$\frac{d}{dx_j} Q_j(x_j; x^n) \Big|_{x_j=x_j^n} = \sum_{i=1}^m b_{ij} \dot{q}_i(t_i^n; t_i^n) = \frac{\partial}{\partial x_j} \Phi(x) \Big|_{x=x^n} \quad (13)$$

$$-\frac{d^2}{dx_j^2} Q_j(x_j; x^n) = \sum_{i=1}^m b_{ij}^2 \frac{c_i^n}{\pi_{ij}}, \quad (14)$$

where $c_i^n \triangleq c_i^{\text{opt}}(t_i^n)$. The explicit form for the SPS algorithm for the choice (10) is thus as follows:

$$x_j^{n+1} = \left[x_j^n + \frac{\sum_{i=1}^m b_{ij} \dot{\psi}_i(t_i^n)}{\sum_{i=1}^m b_{ij} (\sum_l |b_{il}|) c_i^n} \right]_+, \quad j = 1, \dots, p. \quad (15)$$

As shown in Section V, for suitable ψ_i 's, this SPS algorithm is guaranteed to converge. However, since it simultaneously updates all the parameters, the convergence rate of this algorithm is usually very slow, much like the closely related EM algorithms.

The SPS algorithm is closely related to the ‘‘half-quadratic’’ optimization methods [14–17]. However, the quadratic surrogate (6) applies to a broader family of ψ_i 's than the half-quadratic approach, and the derivation of the paraboloidal surrogate Q is somewhat simpler than the corresponding derivation of half-quadratic algorithms.

B. Coordinate Ascent Algorithm with 1-D Newton-Raphson Step (CA-NR)

The CA algorithm updates one pixel at a time using the most recent values of all other pixels as follows:

$$x_j^{n+1} \triangleq \arg \max_{x_j \geq 0} \Phi(x_1^{n+1}, \dots, x_{j-1}^{n+1}, x_j, x_{j+1}^n, \dots, x_p^n).$$

In practice, it is usually impractical to perform an exact maximization. Using the one-dimensional Newton-Raphson algorithm, we obtain a closed-form approximate solution as follows:

$$x_j^{n+1} = \left[x_j^n + \frac{\frac{\partial}{\partial x_j} \Phi(x) \Big|_{x=\tilde{x}}}{-\frac{\partial^2}{\partial x_j^2} \Phi(x) \Big|_{x=\tilde{x}}} \right]_+, \quad (16)$$

where \tilde{x} denotes the current estimate, i.e., \tilde{x} is a shorthand for $[x_1^{n+1}, \dots, x_{j-1}^{n+1}, x_j^n, x_{j+1}^n, \dots, x_p^n]$, and

$$\frac{\partial}{\partial x_j} \Phi(x) \Big|_{x=\tilde{x}} = \sum_{i=1}^m b_{ij} \dot{\psi}_i([B\tilde{x}]_i) \quad (17)$$

$$\frac{\partial^2}{\partial x_j^2} \Phi(x) \Big|_{x=\tilde{x}} = \sum_{i=1}^m b_{ij}^2 \ddot{\psi}_i([B\tilde{x}]_i). \quad (18)$$

Being sequential, the CA-NR algorithm [5] converges rapidly if it converges. However, the curvature of the objective function in (18) does not ensure monotonic increases in the objective function, thus divergence is possible. The CA-NR algorithm is also poorly suited to parallel processing.

C. Coordinate Ascent Algorithm with Parabola Surrogates (CA-PS)

We can overcome the lack of monotonicity of the CA-NR algorithm by applying the optimization transfer principle using parabola surrogates. We call the resulting algorithm CA-PS. The CA-NR algorithm in (16) uses the curvature of the objective function $\Phi(x)$. This is equivalent to making a parabolic *approximation* to $\Phi(x)$, which will not satisfy the monotonicity condition in (4) in general. To guarantee monotonicity, we replace $\frac{\partial^2}{\partial x_j^2} \Phi(x)$ with the curvature of a parabola surrogate that satisfies the conditions discussed above. This approach leads to the following CA-PS iterative algorithm:

$$x_j^{n+1} = \left[x_j^n + \frac{\frac{\partial}{\partial x_j} \Phi(x) \Big|_{x=\tilde{x}}}{-\frac{\partial^2}{\partial x_j^2} Q(x; \tilde{x})} \right]_+, \quad (19)$$

where $Q(x; \tilde{x})$ is a parabola that lies below the objective function $\Phi(x)$ as in (5), defined here by

$$Q(x; \tilde{x}) = \sum_{i=1}^m q_i([Bx]_i; [B\tilde{x}]_i),$$

where q_i is similar to (6) but with the curvature $c_i([B\tilde{x}]_i)$. Thus the curvature of the surrogate function becomes:

$$-\frac{\partial^2}{\partial x_j^2} Q(x; \tilde{x}) = \sum_{i=1}^m b_{ij}^2 c_i([B\tilde{x}]_i). \quad (20)$$

The CA-PS algorithm is guaranteed to monotonically increase Φ . Furthermore, CA-PS is applicable to ψ_i 's like the Huber function [18], which is only once differentiable, whereas CA-NR requires twice differentiable ψ_i 's. However, CA-PS is still not parallelizable, and it is computationally expensive since the curvature $c_i([B\tilde{x}]_i)$ must be recomputed after *every* pixel update.

D. Paraboloidal Surrogates Coordinate Ascent (PSCA) Algorithm

In contrast to the CA-PS algorithm, the PSCA algorithm [6] is derived by first finding a paraboloidal surrogate function at each iteration and then using the CA algorithm to maximize that surrogate iteratively. Thus the next estimate is obtained as follows:

$$x_j^{n+1} = \left[x_j^n + \frac{\frac{\partial}{\partial x_j} Q(x; x^n) \Big|_{x=\tilde{x}}}{-\frac{\partial^2}{\partial x_j^2} Q(x; x^n)} \right]_+, \quad (21)$$

where $Q(x; x^n)$ is the same as in (5) and (6). The derivatives of $Q(x; x^n)$ are as follows:

$$\frac{\partial}{\partial x_j} Q(x; x^n) \Big|_{x=\tilde{x}} = \sum_{i=1}^m b_{ij} \dot{q}_i([B\tilde{x}]_i; t_i^n) \quad (22)$$

$$-\frac{\partial^2}{\partial x_j^2} Q(x; x^n) = \sum_{i=1}^m b_{ij}^2 c_i^n, \quad (23)$$

where c_i^n was defined below (14). Like the CA-PS algorithm, this algorithm will monotonically increase the objective function and is guaranteed to converge if Φ is strictly concave (Section V). Furthermore, the PSCA algorithm requires much less computation per iteration than the CA-PS algorithm, since we can precompute the curvature in (23) prior to cycling through the pixels, unlike the curvature in (20) which changes with every pixel update. However, the PSCA algorithm remains ill-suited to parallel processing since it sequentially updates each pixel.

E. “Naive” Parallel Coordinate Ascent Algorithm

The naive approach to parallelizing the CA algorithm would be to directly separate pixels into subsets and then assign one parallel processor to each subset to perform the CA algorithm. However, this technique is not guaranteed to increase the objective function at each iteration, and thus can diverge. To ensure convergence, we must somehow account for the “coupling” between pixels at the boundaries between subsets. The next section shows that the optimization transfer approach provides a suitable framework for deriving a monotonic parallelizable algorithm.

IV. PARTITIONED-SEPARABLE PARABOLOIDAL SURROGATE COORDINATE ASCENT ALGORITHM (PPCA)

This section describes a new algorithm that not only converges quickly, but is also well-suited to coarse-grain parallel processing. The partitioned-separable paraboloidal surrogate coordinate ascent (PPCA) algorithm is based on a concavity technique developed by De Pierro [2] and uses tangent parabolas. The idea is to sequentially update pixels within each subset while simultaneously updating all subsets.

A. Overview

To derive the PPCA algorithm, we first find a paraboloidal surrogate function for the original objective function, and then partition pixels into K subsets. Since the parabola is concave, we can derive a partitioned-separable surrogate function using a concavity technique. Finally, the CA algorithm is applied in parallel to each set of pixels. Here is an overview of the surrogates derived in this section:

$$\Phi(x) \geq Q(x; x^n) \geq \phi(x; x^n) \triangleq \sum_{k=1}^K Q_k(x_{J_k}; x^n), \quad (24)$$

where Q denotes the paraboloidal surrogate function, ϕ denotes the subset-separated paraboloidal surrogate function, and Q_k denotes the surrogate function for the k th subset. We let J_k denote the k th subset of pixels, and x_{J_k} denote the vector of length $|J_k|$ consisting of the elements of x indexed by J_k . The condition (24) ensures monotonicity in the objective function: $\Phi(x^{n+1}) \geq \Phi(x^n)$. To implement the update (3), we must obtain the next estimate x in each set by maximizing $Q_k(x_{J_k}; x^n)$ as follows:

$$x_{J_k}^{n+1} \triangleq \arg \max_{x_{J_k} \geq 0} Q_k(x_{J_k}; x^n), \quad k = 1, \dots, K. \quad (25)$$

For the example illustrated in Fig. 2, after obtaining the partitioned-separable paraboloidal surrogates, we divide the image into 4 subsets and then assign 4 parallel processors to perform the PPCA algorithm.

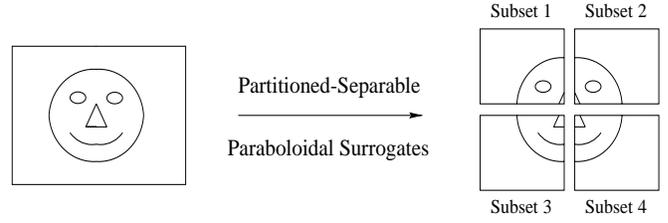


Fig. 2. Schematic of the PPCA algorithm

B. Derivation

First, we construct a paraboloidal surrogate function Q for the original objective function as in (5)-(7). After obtaining the paraboloidal surrogate function, we apply a form of the concavity technique to separate pixels into partitioned sets. Similarly to (9), we can rewrite $[Bx]_i$ as follows:

$$[Bx]_i = \sum_{k=1}^K \rho_{ik} \left(\frac{s_{ik}^n(x_{J_k})}{\rho_{ik}} + t_i^n \right),$$

where

$$s_{ik}^n(x_{J_k}) \triangleq [B_{J_k}(x_{J_k} - x_{J_k}^n)]_i = \sum_{j \in J_k} b_{ij}(x_j - x_j^n),$$

and the matrix B_{J_k} is formed by selecting the columns of B that are indexed by elements of J_k . To satisfy the constraint that $\sum_{k=1}^K \rho_{ik} = 1$ and $\rho_{ik} \geq 0$, which guarantees monotonicity of the algorithm, we choose

$$\rho_{ik} \triangleq \frac{\sum_{j \in J_k} |b_{ij}|}{\sum_{j=1}^p |b_{ij}|}.$$

Similar to (11), since q_i is concave, the following inequality is satisfied:

$$q_i([Bx]_i; t_i^n) \geq \sum_{k=1}^K \rho_{ik} q_i \left(\frac{s_{ik}^n(x_{J_k})}{\rho_{ik}} + t_i^n; t_i^n \right), \quad (26)$$

where from (6),

$$q_i(t + t_i^n; t_i^n) = \psi_i(t_i^n) + \dot{\psi}_i(t_i^n)t - \frac{1}{2}c_i^n t^2. \quad (27)$$

For simplicity, we define the following notation:

$$q_{ik}^n(s) \triangleq \rho_{ik} q_i \left(\frac{s}{\rho_{ik}} + t_i^n; t_i^n \right). \quad (28)$$

Thus from (26), we obtain the following partitioned-separable paraboloidal surrogate function:

$$\phi(x; x^n) = \sum_{i=1}^m \sum_{k=1}^K q_{ik}^n(s_{ik}^n(x_{J_k})), \quad (29)$$

where from (24), the k th subset surrogate function is

$$Q_k(x_{J_k}; x^n) \triangleq \sum_{i=1}^m q_{ik}^n(s_{ik}^n(x_{J_k})). \quad (30)$$

Now we update all the subsets simultaneously, while the pixels in each set are sequentially updated for fast convergence. One approach to implement the maximization in (25) is to apply the CA algorithm over each pixel of x_j by using the most recent values of other pixels of x_j in that set. Letting \tilde{x} denote the current estimates, we define the 1-D quadratic function for each pixel as follows:

$$\begin{aligned} Q_{kj}(x_j) &\triangleq Q_k([\dots, \tilde{x}_{j-1}, x_j, \tilde{x}_{j+1}, \dots, j \in J_k]; x^n) \\ &= Q_{kj}(\tilde{x}_j) + \dot{Q}_{kj}(\tilde{x}_j)(x_j - \tilde{x}_j) - \frac{1}{2}d_{kj}^n(x_j - \tilde{x}_j)^2, \end{aligned}$$

where the first derivative of $Q_{kj}(x_j)$ evaluated at $x_j = \tilde{x}_j$ is

$$\dot{Q}_{kj}(\tilde{x}_j) \triangleq \frac{d}{dx_j} Q_{kj}(x_j) \Big|_{x_j=\tilde{x}_j} = \sum_{i=1}^m b_{ij} \dot{q}_{ik}^n(s_{ik}^n(\tilde{x}_{J_k})),$$

and the curvature of the parabola $Q_{kj}(x_j)$ is given by:

$$d_{kj}^n \triangleq -\frac{d^2}{dx_j^2} Q_{kj}(x_j) = \sum_{i=1}^m w_{ik}^n b_{ij}^2,$$

where $w_{ik}^n \triangleq c_i^n / \rho_{ik}$. Thus we obtain the update x_j in each partitioned set by applying one or more CA cycles to maximize the surrogate function Q_k in (30). Sequentially, for each $j \in J_k$, we perform the following update:

$$\begin{aligned} x_j^{\text{new}} &= \arg \max_{x_j \geq 0} Q_{kj}(x_j) \\ &= \left[\tilde{x}_j + \frac{\dot{Q}_{kj}(\tilde{x}_j)}{d_{kj}^n} \right]_+. \end{aligned} \quad (31)$$

To minimize computation, we first differentiate q_{ik}^n in (28) with respect to s and evaluate at $s = \tilde{t}_{ik} - t_{ik}^n$ as follows:

$$\dot{q}_{ik}^n(\tilde{t}_{ik} - t_{ik}^n) = \dot{\psi}_i(t_{ik}^n) - w_{ik}^n(\tilde{t}_{ik} - t_{ik}^n), \quad (32)$$

where $\tilde{t}_{ik} \triangleq [B_{J_k} \tilde{x}_{J_k}]_i = \sum_{j \in J_k} b_{ij} \tilde{x}_j$. Then rearranging (32) yields

$$\tilde{t}_{ik} = t_{ik}^n + \frac{\dot{\psi}_i - \dot{q}_{ik}^n}{w_{ik}^n}, \quad \forall i \text{ s.t. } b_{ij} \neq 0, \forall j \in J_k,$$

This is a faster way to update \tilde{t}_{ik} . Table II shows the PPCA algorithm outline.

This algorithm generalizes the SPS and PSCA algorithms described in Section III. When $K = p$ and $J_k = \{k\}$, the PPCA algorithm is equivalent to the SPS algorithm. When $K = 1$ and $J_k = \{1, \dots, p\}$, the PPCA algorithm is equivalent to the PSCA algorithm. The most useful cases are when $2 \leq K \ll p$.

An alternate approach to deriving a parallelizable algorithm is to first separate pixels into subsets using De Pierro's concavity technique and then to fit the paraboloidal surrogates inside

the resulting surrogates. However, this approach only applies to concave objective functions, unlike the approach in (24).

```

 $\tilde{x}$  = initial image
 $\tilde{t}_{ik} = \sum_{j \in J_k} b_{ij} \tilde{x}_j, \quad k = 1, \dots, K$ 
 $\rho_{ik} = \frac{\sum_{j \in J_k} |b_{ij}|}{\sum_{j=1}^p |b_{ij}|}, \quad k = 1, \dots, K$ 
for  $n = 1, \dots, \text{Niters}$ 
   $\tilde{t}_i = \sum_k \tilde{t}_{ik}$  (inter-processor communication)
  compute  $c_i^n$  from  $\tilde{t}_i$  using (7)
  compute  $\psi_i$  at  $\tilde{t}_i$ 
  for each processor  $k$ 
     $w_{ik}^n = c_i^n / \rho_{ik}, \quad \forall i \text{ s.t. } b_{ij} \neq 0, \forall j \in J_k$ 
     $\dot{q}_{ik} = \dot{\psi}_i, \quad \forall i \text{ s.t. } b_{ij} \neq 0, \forall j \in J_k$ 
     $d_{kj}^n = \sum_{i=1}^m b_{ij}^2 w_{ik}^n, \quad j \in J_k$ 
    for  $j \in J_k$ 
       $x_j^{\text{old}} = \tilde{x}_j$ 
       $\dot{Q}_{kj} = \sum_{i=1}^m b_{ij} \dot{q}_{ik}$ 
       $\tilde{x}_j = \left[ \tilde{x}_j + \frac{\dot{Q}_{kj}}{d_{kj}^n} \right]_+$ 
       $\dot{q}_{ik} = \dot{q}_{ik} - w_{ik}^n b_{ij}(\tilde{x}_j - x_j^{\text{old}}), \quad \forall i \text{ s.t. } b_{ij} \neq 0$ 
    end
     $\tilde{t}_{ik} = \tilde{t}_{ik} + \frac{\dot{\psi}_i - \dot{q}_{ik}}{w_{ik}^n}, \quad \forall i \text{ s.t. } b_{ij} \neq 0, \forall j \in J_k$ 
  end
  combine  $\tilde{x}$  from all processors
end

```

TABLE II
PPCA ALGORITHM OUTLINE.

V. CONVERGENCE ANALYSIS

Based on the general sufficient conditions for convergence stated in [4], we prove convergence of the algorithm (25) by first stating some sufficient conditions for convergence and then verifying that the algorithm satisfies all the required conditions for convergence. We assume that the objective function, $\Phi(x)$, is strictly concave, continuous, and differentiable for $x \geq 0$. Moreover, the set $\{x \geq 0 : \Phi(x) \geq \Phi(x')\}$ is assumed to be bounded for any x' . We assume that each iteration is associated with disjoint index sets $\{J_1, \dots, J_K\}$ s.t. $\bigcup_{k=1}^K J_k = \{1, \dots, p\}$, and functionals $Q_k(x_{J_k}, x^n), k = 1, \dots, K$. The following conditions pertain to the functionals Q_k in (30).

Condition 1: For $k = 1, \dots, K$, the functionals Q_k satisfy

$$\Phi(x_{J_k}, x_{J_k}^n) - \Phi(x^n) \geq Q_k(x_{J_k}; x^n) - Q_k(x_{J_k}^n; x^n),$$

$\forall x_{J_k} \geq 0, \forall x_{J_k}^n \geq 0, \forall x^n \geq 0$, and $\forall x_{J_k}^n \geq 0$, where $x_{J_k}^n$ is the elements of x^n that are not in set J_k .

Condition 2: Each functional $Q_k(x_{J_k}; x^n)$ is strictly concave and twice differentiable on $x_{J_k} \geq 0, x^n \geq 0$, and each $Q_k(x_{J_k}; x^n)$ is jointly continuous on $x_{J_k} \geq 0, x^n \geq 0$.

Condition 3: The following derivatives match $\forall n$:

$$\frac{\partial}{\partial x_j} \Phi(x^n) = \frac{\partial}{\partial x_j} Q_k(x_{J_k}^n; x^n)$$

for any $x^n \geq 0$, $x_{J_k}^n \geq 0$, and $j \in J_k$.

Condition 4: For $x_{J_k}^{n+1} \geq 0$ and $x^n \geq 0$, the iterates satisfy the Karush-Kuhn-Tucker conditions, $\forall j \in J_k$:

$$\frac{\partial}{\partial x_j} Q_k(x_{J_k}^{n+1}; x^n) \begin{cases} = 0, & x_j^{n+1} > 0 \\ \leq 0, & x_j^{n+1} = 0. \end{cases}$$

Condition 4^{5}:* For any bounded set S , there exists a constant $C_S > 0$ such that $\forall v \neq 0$, $x_{J_k} \geq 0$, and $\forall x^n \in S$,

$$v' G(x_{J_k}; x^n) v \geq C_S \|v\|^2,$$

where $G(x_{J_k}; x^n)$ is the $|J_k| \times |J_k|$ matrix with the (i, j) th element representing $-\frac{\partial^2}{\partial x_i \partial x_j} Q_k(x_{J_k}; x^n)$, $\forall (i, j) \in J_k$.

Theorem: If the curvatures $c_i(\cdot)$ of the surrogates are continuous and have a positive lower bound, then any sequence $\{x^n\}$ generated by the algorithm (25) for penalized-maximum likelihood image recovery converges globally to the unique maximizer of a strictly concave objective function $\Phi(x)$ satisfying the assumptions given in the first paragraph above.

Proof:

- Condition 1 follows the second property of the surrogate function given in Section III-A or (24).
- Condition 2 is satisfied since $Q_k(x_{J_k}; x^n)$ is a concave quadratic function and thus differentiable and jointly continuous.
- Condition 3 follows the third property of the surrogate function in Section III-A.
- Condition 4 is inherent to the update (25).
- Condition 5* is satisfied due to the following proof.

Let $w_{ik}^n \geq \varepsilon$, $\forall i, k, n$, where ε is the positive lower bound of the curvature, then

$$\begin{aligned} v' G(x_{J_k}; x^n) v &= v' B'_{J_k} D(w_{ik}^n) B_{J_k} v \\ &\geq v' B'_{J_k} (\varepsilon I) B_{J_k} v \\ &\geq \varepsilon \lambda_{\min} \{B'_{J_k} B_{J_k}\} \|v\|^2, \end{aligned}$$

where $D(w_{ik}^n)$ is the diagonal matrix with diagonal elements belonging to w_{ik}^n , and $\lambda_{\min} \{B'_{J_k} B_{J_k}\}$ is the minimum eigenvalue of the matrix $B'_{J_k} B_{J_k}$. Thus $v' G(x_{J_k}; x^n) v \geq C_S \|v\|^2$ is satisfied, where $C_S = \varepsilon \lambda_{\min} \{B'_{J_k} B_{J_k}\}$.

Thus, all the conditions needed for the convergence proof in [4] are satisfied. \square

Theorem 1 shows that (25) converges to the global maximizer of a strictly concave objective function. In practice, we use one or more cycles of the CA update (31) rather than the exact maximization (25). However, we believe the proof in [4] can be generalized even to include the case (31).

VI. CONVERGENCE RATE

The curvature of the surrogate function strongly influences the convergence rate. Smaller curvatures generally yield faster convergence. The surrogate function curvatures for each of the algorithms described above are given as follows:

$$d_j^{\text{SPS}} = \sum_{i=1}^m b_{ij} c_i^n \cdot \left(\sum_{l=1}^p b_{il} \right)$$

⁴A sufficient condition for Condition 5 in [4]

$$\begin{aligned} d_j^{\text{PPCA}} &= \sum_{i=1}^m b_{ij} c_i^n \cdot \left(b_{ij} \frac{\sum_{l=1}^p b_{il}}{\sum_{l' \in J_k} b_{il'}} \right), \quad j \in J_k \\ d_j^{\text{PSCA}} &= \sum_{i=1}^m b_{ij} c_i^n \cdot (b_{ij}) \\ d_j^{\text{CA-NR}} &= \sum_{i=1}^m b_{ij} \ddot{\psi}_i([B\tilde{x}]_i) \cdot (b_{ij}). \end{aligned}$$

By comparing the arguments within parentheses of the above equations for the curvatures, we obtain the following inequalities:

$$d_j^{\text{SPS}} \geq d_j^{\text{PPCA}} \geq d_j^{\text{PSCA}} \geq d_j^{\text{CA-NR}} \geq 0,$$

assuming that $\ddot{\psi}_i([B\tilde{x}]_i) < c_i^n$. As expected, the SPS algorithm has the largest curvature, hence generally smallest step sizes and slowest convergence. On the other hand, the CA-NR algorithm has the smallest curvature, thus it has the biggest step sizes which yield the fastest convergence rate (when it converges).

VII. APPLICATION TO IMAGE RESTORATION FOR CONFOCAL MICROSCOPY

Confocal fluorescence microscopy is widely used in cell biology to image thick biological tissues in three dimensions. Unfortunately, most obtainable images contain out-of-focus signals from other planes and have poor resolution due to a resolution/signal-to-noise ratio tradeoff as the detector pinhole size is increased. Therefore, image restoration techniques have been applied to improve the resolution and SNR of the images. In confocal microscopy, the noisy measurement Y can be modeled as follows:

$$Y_i \sim \alpha_1 \text{Poisson}\{f_i[Ax]_i + b_i\}, \quad i = 1, \dots, m_L,$$

where the $m_L \times p$ system matrix A is assumed to be known⁵, x denotes the mean number of fluorescent photons per second, α_1 denotes a known measurement scaling factor, f_i denotes the product of the scan time and the detector efficiency, and b_i denotes the background noise and the dark current [19]. The corresponding log-likelihood function is given by:

$$L(x) = \sum_{i=1}^{m_L} \frac{y_i}{\alpha_1} \log(f_i[Ax]_i + b_i) - (f_i[Ax]_i + b_i) \quad (33)$$

(ignoring irrelevant constants independent of x), which is concave. Due to the ill-posed nature of image restoration problems, we modify the likelihood function by including a penalty function $R(x)$ to form the following penalized-likelihood objective function:

$$\Phi(x) = L(x) - \beta R(x), \quad (34)$$

where β controls the degree of smoothness in the restored image. Our goal is to estimate x by maximizing the objective function $\Phi(x)$:

$$\hat{x} \triangleq \arg \max_{x \geq 0} \Phi(x) = \arg \max_{x \geq 0} L(x) - \beta R(x). \quad (35)$$

⁵In practice, the point spread function (PSF) of a confocal microscope is not exactly known; however, one can measure the PSF by using very small microsphere beads.

For the penalty function $R(x)$ of interest here, the objective function Φ in (34) is a special case of the general form (1); therefore, the algorithms of sections III and IV are applicable. For clarification, we separately derive the surrogate functions for the likelihood part and the penalty part.

A. The Likelihood Part

The likelihood function in (33) can be expressed in the following form:

$$L(x) = \sum_{i=1}^{m_L} \psi_i^L([Bx]_i) \quad (36)$$

with $b_{ij} = f_i a_{ij}$, which is the (ij) th element of the matrix B , and

$$\psi_i^L(l) = \frac{y_i}{\alpha_1} \log(l + b_i) - (l + b_i). \quad (37)$$

Since $b_i > 0$, we can choose c_i^n to be the optimal curvature as derived in (8) [6, 20]:

$$c_i^{\text{opt}}(l) = \begin{cases} \left[\frac{2y_i}{\alpha_1 l^2} \left\{ \log\left(\frac{l+b_i}{b_i}\right) - \frac{l}{l+b_i} \right\} \right]_+, & l > 0 \\ \frac{y_i}{\alpha_1 b_i^2}, & l = 0. \end{cases} \quad (38)$$

B. The Penalty Part

The general form of the penalty function is given by:

$$R(x) = \sum_{i=1}^{m_R} \psi_i^R([Cx]_i), \quad (39)$$

where ψ_i^R is the potential function, and C is the weighting 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 $\psi_i^R(t)$ satisfies the following conditions [6, 10, 18]:

- ψ_i^R is symmetric.
- ψ_i^R is everywhere differentiable (and therefore continuous).
- $\dot{\psi}_i^R(t) = d/dt \psi_i^R(t)$ is convex.
- $\omega_i^R(t) = \frac{\dot{\psi}_i^R(t)}{t}$ is non-decreasing for $t \geq 0$.
- $\omega_i^R(0) = \lim_{t \rightarrow 0} \frac{\dot{\psi}_i^R(t)}{t}$ is finite and nonzero.

According to [18], the optimal curvature for such a symmetric nonquadratic penalty function is given by:

$$c_i^{\text{opt}}(v_i^n) \triangleq \frac{\dot{\psi}_i^R(v_i^n)}{v_i^n} = \omega_i^R(v_i^n), \quad (40)$$

where $\dot{\psi}_i^R$ is the first derivative of the potential penalty function at the i th element, and $v_i^n \triangleq [Cx^n]_i$.

Combining all the likelihood and penalty surrogate functions, we obtain the following update for each $j \in J_k$ by maximizing these surrogate functions as in (31):

$$\begin{aligned} x_j^{n+1} &= \arg \max_{x_j \geq 0} Q_{kj}^L(x_j) - \beta Q_{kj}^R(x_j) \\ &= \left[x_j^n + \frac{\dot{Q}_{kj}^L(\tilde{x}_j) - \beta \dot{Q}_{kj}^R(\tilde{x}_j)}{d_{kj}^L + \beta d_{kj}^R} \right]_+, \end{aligned} \quad (41)$$

where \dot{Q}_{kj}^L and d_{kj}^L are the first derivative and the curvature of the surrogate function for the likelihood part, and \dot{Q}_{kj}^R and d_{kj}^R are the first derivative and the curvature of the surrogate function for the penalty part.

VIII. RESULTS

A. 2-D Simulation Results

A 512×512 pepper image was degraded by a 15×15 Gaussian point spread function (PSF) with FWHM of 11.7 pixels (standard deviation = 5.0) and Poisson noise with PSNR⁶ of 25 dB as shown in Fig. 3b. Since our long-term interest is space-varying PSFs, we used convolution rather than FFT techniques for these algorithms. We used the following nonquadratic penalty function [21]:

$$\psi_i^R(t) = \delta^2 \left[\left| \frac{t}{\delta} \right| - \log \left(1 + \left| \frac{t}{\delta} \right| \right) \right], \quad (42)$$

where δ controls the degree of edge preservation. Fig. 3c shows the restoration using the 4-PPCA algorithm (with four parallel processors).

Table III compares wall times of monotonic algorithms (PSCA and PPCA), and nonmonotonic algorithms (CA-NR, CA-P, PSCA-P, PPCA-P). The letter ‘‘P’’ in nonmonotonic algorithms represents the precomputed curvature [6], where we replace $c_i^{\text{opt}}(\cdot)$ with $c_i^{\text{pre}}(\cdot) = -\dot{\psi}_i^L(l_i^{\text{max}})$ ⁷, and $l_i^{\text{max}} = \frac{y_i}{\alpha_1} - b_i$. The algorithms above were performed on the IBM SP2 parallel processors. Convergence in this table is defined as the smallest n such that $\Phi(x^n) - \Phi(x^0) > 0.999(\Phi(x^*) - \Phi(x^0))$, where $\Phi(x^0)$ is the objective value of the initial image, and $\Phi(x^*)$ is the largest objective value among all methods obtained in 50 iterations. Fig. 4 shows the subset partitions. Since the PSF is fairly small, interprocessor communication time becomes significant as the number of processors increases; therefore, speedup is less than the ideal inverse relationship, as predicted by Amdahl’s law [22]. Nevertheless, these results confirm that the PPCA algorithm is well suited for parallel processing.

Fig. 5 shows that the PPCA algorithms increase the objective function essentially as much per iteration as the PSCA algorithm. This effect implies that subset-separation technique barely slows the convergence rate of the PPCA algorithm compared with the PSCA algorithm, which is a one subset version

⁶The peak signal-to-noise ratio is defined as follows:

$$\text{PSNR} = 10 \log_{10} \frac{\max_i (y_i - \alpha_1 b_i)^2}{\frac{1}{N^2} \sum_i (y_i - \mathbb{E}[y_i])^2},$$

where $N^2 = \text{pixel size}$.

⁷This ad hoc modification loses the guarantee of monotonicity, but reduces computation and *usually* seems to converge.



(a) Original Image



(b) Degraded Image



(c) Restored Image

Fig. 3. 2-D simulation and restoration using a 4-PPCA algorithm with $\beta = 0.01$ and $\delta = 1.5$.

of the PPCA algorithm. Fig. 6 shows that the PPCA algorithms converge in less elapsed time than the PSCA algorithm. Using the precomputed curvatures, Fig. 7 illustrates that the CA-NR, CA,P, PSCA,P, and PPCA,P algorithms increase the objective function nearly at the same rate; however, Fig. 8 confirms that the PPCA,P algorithm converges in less time than other non-monotonic algorithms.

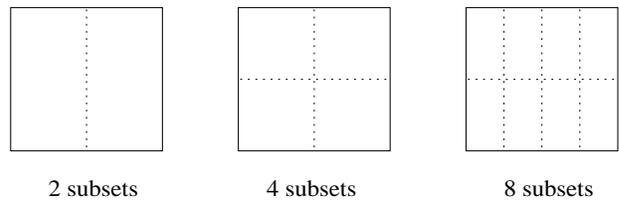


Fig. 4. Partitioned set patterns of a 2-D image.

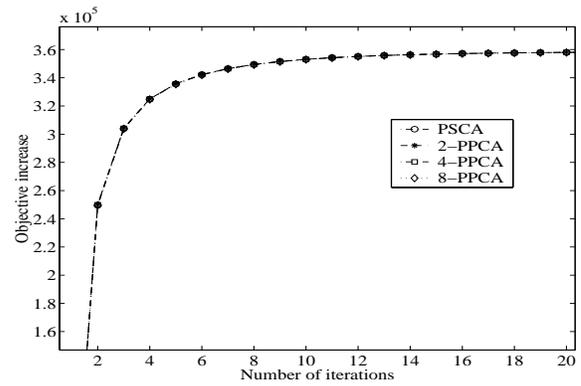


Fig. 5. Comparison of objective function increase versus number of iterations using monotonic algorithms on a 2-D image.

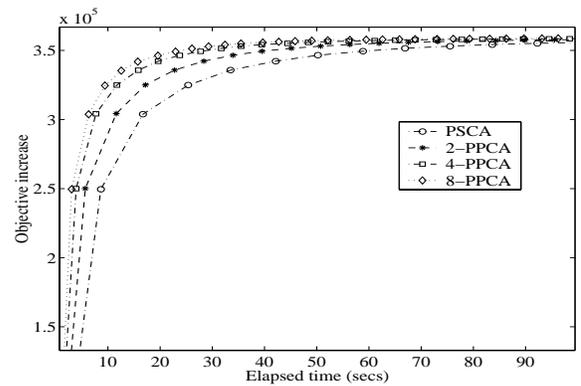


Fig. 6. Comparison of objective function increase versus elapsed time using monotonic algorithms on a 2-D image.

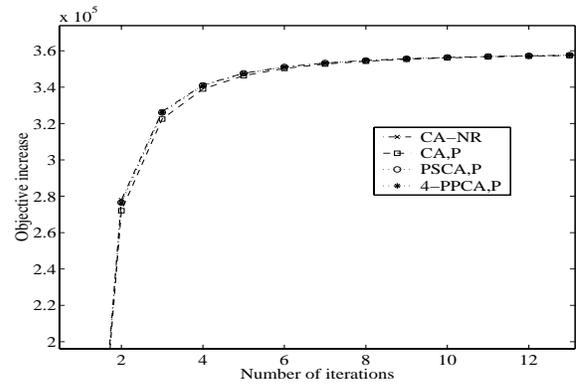


Fig. 7. Comparison of objective function increase versus number of iterations using nonmonotonic algorithms on a 2-D image.

Convergence	Monotonic Algorithms				Nonmonotonic algorithms					
	PSCA	2-PPCA	4-PPCA	8-PPCA	CA-NR	CA,P	PSCA,P	2-PPCA,P	4-PPCA,P	8-PPCA,P
#iters	38	39	39	41	28	29	27	28	28	28
Wall time (s)	318.20	213.85	156.20	132.34	612.5	517.91	195.89	123.48	89.33	80.55
Wall time/iter (s)	8.37	5.48	4.01s	3.23	21.87	17.86	7.26	4.41	3.19	2.88
Speedup factor	1	1.52	2.09	2.59	0.33	0.41	1	1.64	2.28	2.52

TABLE III

COMPARISON OF WALL TIMES AND NUMBER OF ITERATIONS TO CONVERGE USING MONOTONIC AND NONMONOTONIC ALGORITHMS FOR A 512×512 PEPPER IMAGE.

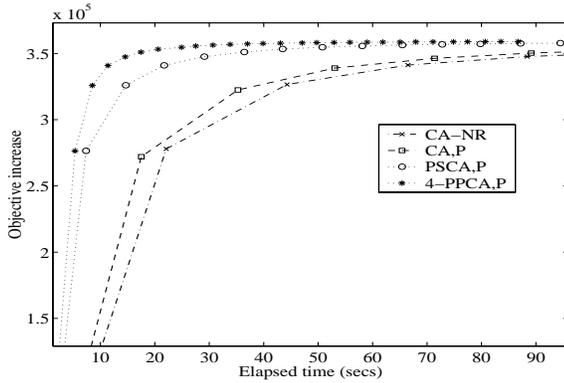


Fig. 8. Comparison of objective function increase versus elapsed time using nonmonotonic algorithms on a 2-D image.

B. 3-D Simulation Results for Confocal Microscopy

Following [13], a spherical shell test specimen was generated on a $256 \times 256 \times 64$ pixel grid and was degraded by a $15 \times 15 \times 15$ PSF created from the XCOSM package [23] having pixel sizes $\Delta x = \Delta y = \Delta z = 0.15 \mu\text{m}$, $40 \times /1.0$ NA oil-immersion objective, and a fluorescent wavelength of $0.63 \mu\text{m}$, and the Poisson noise with PSNR of 40 dB. This PSF is circularly symmetric in the $x - y$ plane but it has elongation in the z direction which causes a very poor resolution in the axial direction. Fig. 9 shows the lateral and axial medial sections through the original, degraded and restored images performed for 20 iterations. The images on the $x - z$ plane have been scaled up to the same scale as those in the $x - y$ plane for display purpose. As seen from the center slice of the $x - z$ plane of the restored image (Fig. 9c), the elongation in the z direction of the restored image have been dramatically reduced.

Fig. 10 shows that the total wall times for 3-D results are nearly inversely proportional to the number of processors. This is because a larger amount of work in 3-D data has been assigned to each processor which means less communication time relative to the total computation time.

Table IV shows the performance of the wall times of the PSCA, and PPCA algorithms for 20 iterations. Since we used convolution rather than the FFT algorithm, the total time requirement is quite large. Similarly to 2-D simulation results, Figs. 11-12 verify that the PPCA algorithm increases the objective function almost at the same rate as the PSCA algorithm but it requires much less total wall time for computation.

IX. CONCLUSIONS

We have presented a new fast converging parallelizable algorithm called the partitioned-separable paraboloidal surrogate co-

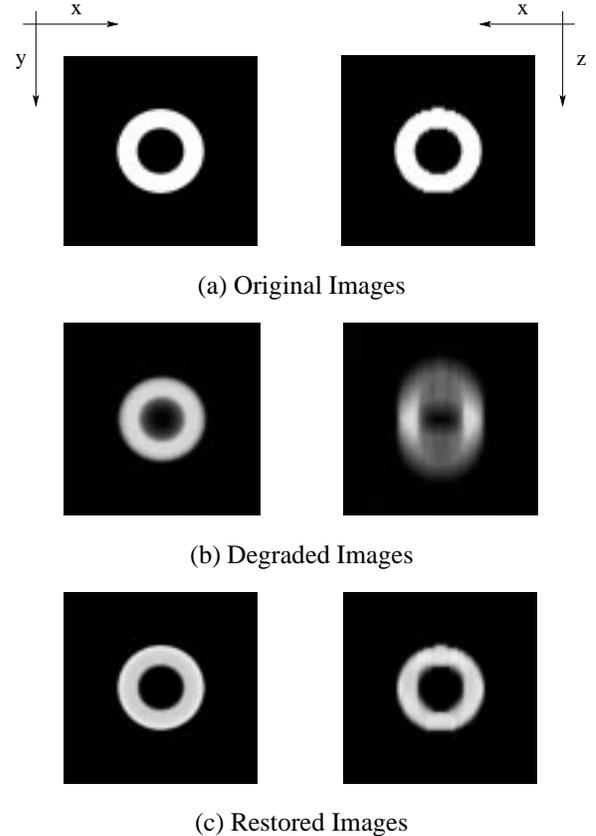


Fig. 9. Results for a 3-D simulated spherical shell using a 4-PPCA algorithm for 20 iterations with $\beta = 0.1$ and $\delta = 10$. Lateral and axial medial sections through the image are in the left and right, respectively. For display purpose, the axial sections were scaled in z to obtain a 1:1 aspect ratio.

20 iterations	PSCA	2-PPCA	4-PPCA	8-PPCA
Wall time (hrs)	11.54	5.93	3.06	1.64
Wall time/iter (mins)	34.62	17.79	9.18	4.92
Speedup factor	1	1.95	3.77	7.04

TABLE IV

COMPARISON OF WALL TIMES OF PSCA, AND PPCA ALGORITHMS FOR A 3-D IMAGE.

ordinate ascent algorithm. This approach overcomes the drawback of the ordinary coordinate ascent algorithm which is a non-parallelizable algorithm. Compared to completely simultaneous updates like EM and SPS algorithms, this proposed algorithm has a faster convergence rate due to larger updating step sizes. Unlike the PSCA [6] algorithm, the surrogate functions in the PPCA algorithm are separable between subsets to allow simultaneous update across pixel subsets. Thus the parallel processors

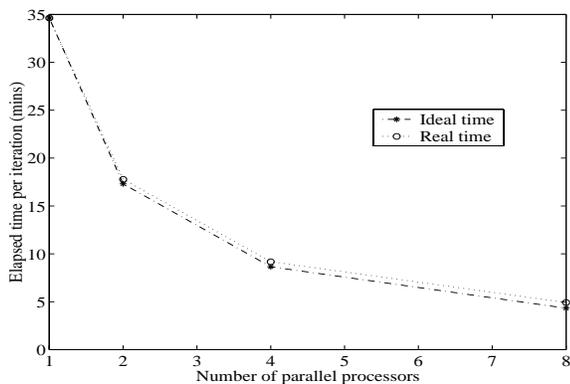


Fig. 10. Performance on parallel processors in elapsed time for the PPCA algorithms using a $256 \times 256 \times 64$ image and a $15 \times 15 \times 15$ PSF.

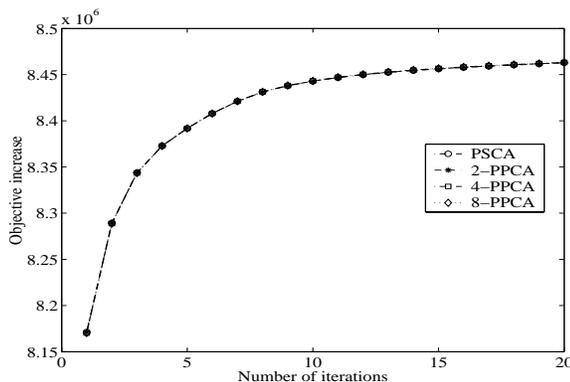


Fig. 11. Comparison of objective function increase versus number of iterations using PSCA and PPCA algorithms on a 3-D image.

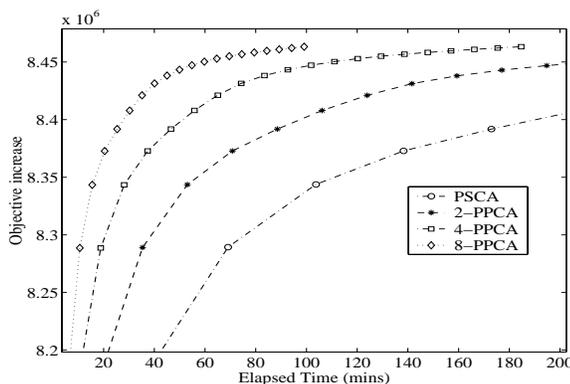


Fig. 12. Comparison of objective function increase versus elapsed time of PSCA and PPCA algorithms on a 3-D image.

can be assigned to each subset to reduce the total time requirement. Since the PPCA algorithm increases the objective function nearly at the same rate as the PSCA algorithm, parallelizability in the PPCA algorithm only slightly reduces the convergence rate. The PPCA algorithm using the precomputed curvature, which is a nonmonotonic algorithm, converges much faster than the CA-NR and CA,P algorithms. Thus the PPCA algorithm yields the fastest convergence among the monotonic and nonmonotonic algorithms tested. The PPCA algorithm seems mostly naturally suited to shift-variant system models where FFTs are inapplicable. It is an interesting open question whether

the parallelization associated with a subset-separable surrogate function could be combined with FFTs for faster computation in shift-invariant problems.

X. ACKNOWLEDGMENT

This work, which is also directly applicable to 3-D PET image reconstruction, was supported in part by NIH grant CA-60711.

REFERENCES

- [1] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Stat. Soc. B*, vol. 39, pp. 1–38, 1977.
- [2] A. R. De Pierro, "A Modified Expectation Maximization Algorithm for Penalized Likelihood Estimation in Emission Tomography," *IEEE Trans. Med. Imaging*, vol. 14, no. 1, pp. 132–137, March 1995.
- [3] J. A. Fessler and A. O. Hero, "Space-Alternating Generalized Expectation-Maximization Algorithm," *IEEE Trans. Signal Processing*, vol. 42, no. 10, pp. 2664–2677, October 1994.
- [4] J. A. Fessler and A. O. Hero, "Penalized Maximum-Likelihood Image Reconstruction Using Space-Alternating Generalized EM Algorithms," *IEEE Trans. Image Processing*, vol. 4, no. 10, pp. 1417–1429, October 1995.
- [5] C. A. Bouman and K. Sauer, "A unified approach statistical tomography using coordinate descent optimization," *IEEE Trans. Image Processing*, vol. 5, no. 3, pp. 480–492, March 1996.
- [6] H. Erdoğan and J. A. Fessler, "Monotonic Algorithms for Transmission Tomography," *IEEE Trans. Med. Imaging*, vol. 18, no. 9, pp. 801–814, September 1999.
- [7] S. Sothivirat and J. A. Fessler, "Partitioned Separable Paraboloidal Surrogate Coordinate Ascent Algorithm for Image Restoration," in *Proc. IEEE Int'l Conf. on Image Processing*, 2000, vol. 1, pp. 109–112.
- [8] S. Saquib, J. Zheng, C. A. Bouman, and K. D. Sauer, "Parallel Computation of Sequential Pixel Updates in Statistical Tomographic Reconstruction," in *Proc. IEEE Int'l Conf. on Image Processing*, 1995, vol. 2, pp. 93–96.
- [9] J. A. Fessler, "Grouped Coordinate Ascent Algorithms for Penalized-Likelihood Transmission Image Reconstruction," *IEEE Trans. Med. Imaging*, vol. 16, no. 2, pp. 166–75, April 1997.
- [10] J. A. Fessler, "Grouped Coordinate Descent Algorithms for Robust Edge-Preserving Image Restoration," *SPIE 97*, vol. 3170, pp. 184–194, July 1997.
- [11] D. Xie and L. Adams, "New parallel SOR method by domain partitioning," *Siam J. Sci. Comput.*, vol. 20, no. 6, pp. 2261–2281, 1999.
- [12] 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, 1996, pp. 199–208.
- [13] 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*, vol. 16, no. 10, pp. 2377–2391, October 1999.
- [14] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Patt. Anal. Mach. Int.*, vol. 14, no. 3, pp. 367–383, Mar. 1992.
- [15] D. Geman and C. Yang, "Nonlinear Image Recovery with Half-Quadratic Regularization," *IEEE Trans. Image Processing*, vol. 4, no. 7, pp. 932–46, July 1995.
- [16] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic Edge-Preserving Regularization in Computed Imaging," *IEEE Trans. Image Processing*, vol. 6, no. 2, February 1997.
- [17] A. H. Delaney and Y. Bresler, "Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography," *IEEE Trans. Image Processing*, vol. 7, no. 2, pp. 204–221, February 1998.
- [18] P. J. Huber, *Robust Statistics*, Wiley, New York, 1981.
- [19] 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*, vol. 10, no. 5, pp. 1014–1023, May 1993.
- [20] J. A. Fessler and H. Erdoğan, "A Paraboloidal Surrogates Algorithm for Convergent Penalized-Likelihood Emission Image Reconstruction," *IEEE Nuc. Sci. Symp. and Med. Im. Conf.*, vol. 2, pp. 1132–1135, 1998.
- [21] K. Lange, "Convergence of EM Image Reconstruction Algorithms with Gibbs Smoothing," *IEEE Trans. Med. Imaging*, vol. 9, no. 4, pp. 439–446, December 1990.
- [22] G. M. Amdahl, "Validity of the single-processor approach to achieving large scale computing capabilities," in *AFIPS Conference Proceedings*, 1967, vol. 30, pp. 483–485, AFIPS Press.

- [23] The XCOSM deconvolution package is available from URL <http://www.ibs.wustl.edu/bcl/xcosm/xcosm.html>.