# Binary Steering in Discrete Tomography Reconstruction with Sequential and Simultaneous Iterative Algorithms

Yair Censor
Department of Mathematics
University of Haifa, Mt. Carmel,
Haifa 31905, Israel (yair@math.haifa.ac.il).

December 10, 2000
Revised: May 15, 2001

Final Revision: July 17, 2001

**Abstract**

The binary steering process is a heuristic designed to intervene between consecutive steps of a nonbinary iterative image reconstruction algorithm in order to gradually steer the iterates towards a binary solution. We present computational results which show that a strongly overrelaxed simultaneous nonbinary iterative algorithm performs in our experiments better then a strongly underrelaxed sequential iterative algorithm. We also notice that faster binary steering gives better binary reconstructed images when the sequential iterative nonbinary algorithm is used.

## 1 Introduction

The fully discretized model of the two-dimensional image reconstruction problem of computerized tomography can be represented by a system of linear equations $Ax = b$ . Here $x \in R^n$ is the unknown *image vector* whose $j$-th component $x_j$ has the value of the uniform grayness level of the $j$-th pixel of the model, $b \in R^m$ is the *measurements vector* whose $i$-th component $b_i$ is the value of the $i$-th line integral through the unknown image. The *binary image reconstruction problem* assumes that the $m \times n$ *system matrix* $A$ is a zero–one matrix with its $i$-th row and $j$-th column element $a_j^i$ equal to zero if the path of the $i$-th line integral does not intersect the $j$-th pixel, and equal to one if it does. Further, it is assumed that the practically feasible values of the image vector are only zeros and ones. The problem then is to find a zero–one vector $x^*$ that approximates well enough a solution of the system $Ax = b$.

Recently, in [5], inspired by Herman's work [11], we proposed a process, called *binary steering*, designed to intervene between the iterative steps of any nonbinary algorithm for solving $Ax = b$ in a way that would gradually steer the iterates towards a binary solution. This heuristic process is applicable to a plethora of nonbinary iterative reconstruction algorithms which solve (asymptotically, depending on the relevant solution concept adopted) the system $Ax = b$. Some of these nonbinary iterative algorithms perform very well on nonbinary image reconstruction problems, efficiently generating acceptable reconstructed images (i.e., approximations to the solution vector $x^*$), some of them lend themselves to parallel computations, or have other favorable features such as guaranteed convergence even if the system $Ax = b$ is inconsistent, see, e.g., Herman [12], Censor and Zenios [6] or Bauschke and Borwein [1] and references therein.

In our preliminary work [5] we used as a nonbinary algorithm the simultaneous Cimmino method and, by running the binary steering process with it on three test images (phantoms) that were used earlier in this field, we showed that the heuristic binary steering process works. By this we mean that the additional operations introduced by the binary steering process between consecutive iterations of the nonbinary algorithm do not ruin the practical initial convergence of the overall process and that indeed the process results in binary images which reconstruct well (with some errors, of course) the original phantoms.

When we embarked on the project reported here we designed an experimental setup in which we could demonstrate the performance of the binary steering process on more test images and compare the results of applying the process with two different nonbinary iterative algorithms: the sequential Kaczmarz algorithm, also known in the image reconstruction literature by the name ART (for Algebraic Reconstruction Technique) and the, recently devised, fully simultaneous Component Averaging (CAV) algorithm of Censor, Gordon and Gordon [7]. We also improved upon the work in [5] by allowing the binary steering process progress in different speeds (the precise meaning of this becomes clear in the sequel). The experimental conclusions of the (still limited in scope) computational work done here (with two different phantoms and two different nonbinary iterative reconstruction algorithms) are as follows: (1) A strongly overrelaxed simultaneous nonbinary iterative algorithm performs in our experiments better then a strongly underrelaxed sequential iterative algorithm. (2) We found that binary steering has little effect on the simultaneous iterative nonbinary algorithm while in each of our experiments with the sequential iterative algorithm the quality of binary reconstruction is directly related to the speed by which the binary steering process is steered, i.e., better and less erroneous reconstructions are obtained for faster binary processes. More computational work is needed to verify whether these observations are universal or not and whether situations for which binary steering, with specific steering schemes and parameters, is advantageous can be clearly identified.

The results of our experiments are depicted by reconstructed images and by convergence-plots displaying error measures versus iteration index numbers in the reconstructions. Our computational work is of an exploratory nature and more work with carefully designed methodological sets of experiments is needed to refine it. The paper is laid out as follows. The binary steering process is defined and described in Section 2. Section 3 contains a precise description of the two nonbinary algorithms with which we experimented here and the computational results are presented in Section 4.

## 2 The Binary Steering Process

In this section we describe how the binary steering process works in conjunction with *any* nonbinary iterative reconstruction algorithm. We assume that the nonbinary algorithms, to which binary steering will be applied, have the following general form:

**Algorithm 1** ***(General Form of Nonbinary Algorithm).***

***Initialization****: $x^0 \in U$, where $U \subseteq R^n$ is the initialization set dictated by the specific nonbinary algorithm.*

***Iterative Step**: Given the k-th iterate $x^k$ and the data of the problem $d \in D$, where $D$ is the data space dictated by the specific nonbinary algorithm, calculate:*

***(1) Correction Calculation**: The k-th correction vector $c^k$ is calculated by a formula of the form $c^k = f_k(x^k, d)$, where the functions $f_k$ are dictated by the specific nonbinary algorithm.*

***(2) Correction Application**: The next iterate $x^{k+1}$ is calculated by a formula of the form $x^{k+1} = g_k(x^k, c^k)$, where the functions $g_k$ are dictated by the specific nonbinary algorithm.*

The term *data* ($d \in D$) in this (and the next) algorithm is meant to include not only the measured data (such as $b$ in $Ax = b$) but all measured as well as design data, i.e., both $A$ and $b$ in the case of linear equations.

Many of the algorithms in this field can be described in more detailed schematic forms as having the structure of *sequential, simultaneous, sequential block-iterative* or *simultaneous block-iterative* algorithms, see, e.g., [6, Section 1.3], another general algorithmic scheme of interest is that of *averaging sequential strings*, see Censor, Elfving and Herman [4]. But for the purpose of constructing the binary steering process it is enough to assume that the nonbinary iterative algorithms are of the form of Algorithm 1.

The next definition provides the tool with which we will sequentially binarize iterates generated by a nonbinary algorithm.

**Definition 2** *Let $\alpha = \{\alpha_k\}_{k \geq 0}$, $\beta = \{\beta_k\}_{k \geq 0}$ and $t = \{t_k\}_{k \geq 0}$ be three real sequences such that $0 \leq \alpha_k < t_k < \beta_k \leq 1$, and $\alpha_k < \alpha_{k+1}$ and $\beta_{k+1} < \beta_k$, for all $k \geq 0$. Given any sequence $\{x^k\}_{k \geq 0}$ of vectors $x^k = (x_j^k)_{j=1}^n \in R^n$, the sequence $\{\widetilde{x}^k\}_{k \geq 0}$, defined, for all $k \geq 0$ and $j = 1, 2, ..., n$, by*

$$\widetilde{x}_j^k = \begin{cases} 0, & \text{if } x_j^k \leq \alpha_k, \\ 1, & \text{if } x_j^k \geq \beta_k, \\ x_j^k, & \text{otherwise}, \end{cases} \tag{1}$$

*is called* ***the sequential binarization of $\{x^k\}_{k \geq 0}$ with respect to the triplet of sequences ($\alpha$,$\beta$,$t$).***

In the binary steering process, described below, the current iterate $x^k$ is undergoing a step of sequential binarization – the result of which is fed into the iterative step of the nonbinary Algorithm 1. The output obtained in this way might be in conflict with the nonbinarized previous iterate $x^k$

and, therefore, the following concept – from which the meaning of the term *conflict* becomes clear – is used.

**Definition 3** *Let $\alpha = \{\alpha_k\}_{k\geq 0}$, $\beta = \{\beta_k\}_{k\geq 0}$ and $t = \{t_k\}_{k\geq 0}$ be three real sequences as in Definition 2 and let $\epsilon$ be an arbitrarily small but fixed real number with $0 < \epsilon < 0.1$. Given any two vector sequences $\{x^k\}_{k\geq 0}$ and $\{y^k\}_{k\geq 0}$, the sequence $\{z^k\}_{k\geq 0}$, defined, for all $k \geq 0$ and $j = 1, 2, ..., n$, by*

$$z_j^k = \begin{cases} t_k - \epsilon, & \text{if } x_j^k \leq \alpha_k \text{ and } y_j^k \geq t_k, \\ t_k + \epsilon, & \text{if } x_j^k \geq \beta_k \text{ and } y_j^k \leq t_k, \\ y_j^k, & \text{otherwise,} \end{cases} \tag{2}$$

*is said* ***to settle sequentially the conflict between $\{x^k\}_{k\geq 0}$ and $\{y^k\}_{k\geq 0}$ with respect to the triplet of sequences ($\alpha$,$\beta$,$t$) and $\epsilon$.***

Using these definitions we formulate the binary steering process as follows.

**Algorithm 4** ***(The Binary Steering Process [5, Figure 12.2]).***

***Initialization****: $x^0 \in U$, where $U \subseteq R^n$ is the initialization set dictated by the specific nonbinary algorithm in use.*

***Iterative Step****: Given the (current) k-th iterate $x^k$ do the following:*

***(1) Sequential Binarization****: Use the sequences $(\alpha, \beta, t)$ of Definition 2 to perform a sequential binarization on $x^k$ to obtain $\widetilde{x}^k$ and then do:*

***(2) Nonbinary Algorithmic Step****: Use the $k$-th sequentially binarized iterate $\widetilde{x}^k$ and the data of the problem $d \in D$, where $D$ is the data space dictated by the specific nonbinary algorithm in use, to calculate:*

***(2a) Correction Calculation****: The $k$-th correction vector $c^k$ is calculated by a formula of the form $c^k = f_k(\widetilde{x}^k, d)$, where the functions $f_k$ are dictated by the specific nonbinary algorithm in use.*

***(2b) Correction Application****: The output iterate $y^k$ of the nonbinary algorithmic step is calculated by a formula of the form $y^k = g_k(\widetilde{x}^k, c^k)$, where the functions $g_k$ are dictated by the specific nonbinary algorithm in use.*

***(3) Conflict Resolution****: Use the sequences $(\alpha, \beta, t)$ and the parameter $\epsilon$ of Definition 3 to calculate the next iterate $x^{k+1}$ of the binary steering process by settling the conflict between $y^k$ and $x^k$, if any, according to Definition 3.*

As iterations of the binary steering process proceed, more and more components of the iteration vector take zero–one values because of the monotonicity of the sequences $\{\alpha_k\}$ and $\{\beta_k\}$, see Figure 8. Observe that the correction vector $c^k$ is based on the sequentially binarized vector $\widetilde{x}^k$ but it is applied to the vector $x^k$ itself. If the $j$-th component of the output vector of the nonbinary algorithmic step $y_j^k$ is larger or equal to the current value of $t_k$ but that component of the vector from which it was calculated $x_j^k$ was below $\alpha_k$ then we say that there is a *conflict* and we prefer not to make a decision about that component but rather let the conflict resolution step put that component back to $t_k - \epsilon$. A similar approach applies to the other case of conflict in Definition 3. At the end of each run of a binary process, i.e., at the iteration $K$ when the process is stopped, a simple thresholding step (with respect to the threshold value $t_K$) is always applied to force all remaining nonbinary values of the image vector to take either zero or one values. The final thresholding step is applied to derive from the iterate $x^K$ the approximate solution $x^*$ at this stage by

$$x_j^* = \begin{cases} 0, & \text{if } x_j^K \leq t_K, \\ 1, & \text{if } x_j^K > t_K. \end{cases} \tag{3}$$

# 3 The Nonbinary Algorithms in Our Experiments

We compare experimentally the behavior as discrete tomography solvers of two iterative nonbinary algorithms applied within the binary steering process. The first nonbinary algorithm is the sequential ART (Algebraic Reconstruction Technique), first introduced in the literature on image reconstruction from projections by Gordon, Bender and Herman [10], which is known as Kaczmarz's method [13], see, e.g., [6] and [12] for more information. This algorithm, designed to iteratively solve a linear system $Ax = b$ is as follows.

**Algorithm 5** ***(The Nonbinary ART Algorithm, [10]).***
***Intialization***: *$x^0 \in R^n$ is arbitrary.*
***Iterative Step***: *Given the k-th iterate $x^k$, choose a control index $i(k)$ from any repetitive control sequence (see definition below) and calculate:*
***(1) Correction Calculation***: *The k-th correction vector $c^k$ is calculated by*

$$c^k = \lambda_k \frac{b_{i(k)} - \langle a^{i(k)}, x^k \rangle}{\|a^{i(k)}\|^2} a^{i(k)}. \tag{4}$$

***(2) Correction Application**: The next iterate $x^{k+1}$ is calculated by*

$$x^{k+1} = x^k + c^k. \tag{5}$$

Here $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ are the standard inner product and Euclidean norm, respectively, and $a^{i(k)}$ and $b_{i(k)}$ are the $i(k)$-th column of the transposed matrix $A^T$ and the $i(k)$-th component of $b$, respectively. $\{\lambda_k\}_{k \geq 0}$ is a user-chosen sequence of *relaxation parameters*. To guarantee convergence of this algorithm to a solution of $Ax = b$ when the system is consistent, these parameters should be in the interval $0 < \delta \leq \lambda_k \leq 2 - \delta$, for all $k \geq 0$, where $\delta$ is an arbitrarily small but fixed real number, see, e.g., [6, Algorithm 5.4.3]. Well-documented computational experience shows that Algorithm 5 performs better as an image reconstruction problem solver if very small relaxation parameters are used, see, e.g., Herman [12] and references therein.

A *repetitive control sequence* is a sequence $\{i(k)\}_{k \geq 0}$ of indices $1 \leq i(k) \leq m$, for all $k \geq 0$, where $m$ is the number of rows in the matrix $A$, such that, for every $1 \leq l \leq m$ and every $k \geq 0$, there exists an index $q > k$ for which $i(q) = l$. The family of repetitive control sequences includes the *cyclic control,* defined by $i(k) = k \bmod m + 1$, for all $k \geq 0$, as well as many others such as the *almost cyclic control*, the *remotest set control* and the *approximately remotest set control*, see, e.g., [6, Definition 5.1.1].

The second nonbinary iterative algorithm that we use is the fully simultaneous *Component Averaging (CAV)* algorithm of Censor, Gordon and Gordon [7], which is a new, highly accelerated, modification of Cimmino's simultaneous algorithm [8], see also [6, Section 5.6]. The general form of an iterative step of the CAV algorithm is given by

$$x^{k+1} = x^k + \lambda_k \sum_{i=1}^{m} G_i(P_{H_i}^{G_i}(x^k) - x^k), \tag{6}$$

where $H_i$ is the hyperplane $H_i := \{x \in R^n \mid \langle a^i, x \rangle = b_i\}$, for every $i = 1, 2, ..., m$. The family $\{G_i\}_{i=1}^m$ of real nonnegative diagonal matrices $G_i = \operatorname{diag}\{g_{ij} \mid j = 1, 2, \ldots, n\}$, is such that $\sum_{i=1}^m G_i = I$, the unit matrix, and $g_{ij} = 0$, if and only if $a_j^i = 0$. This is called in [7] a *sparsity pattern oriented (SPO) family with respect to A*. The symbol $P_{H_i}^{G_i}(x^k)$ stands for the *generalized oblique projection* of $x^k$ onto $H_i$ with respect to $G_i$, as defined in [7, Definition 2.1], and $\{\lambda_k\}$ are relaxation parameters as in Algorithm 5. Following [7], we denote by $s_j$ the number of nonzero elements in the $j$-th column of $A$ and define

$$g_{ij} := \begin{cases} 1/s_j, & \text{if } a^i_j \neq 0, \\ 0, & \text{if } a^i_j = 0, \end{cases} \tag{7}$$

to obtain the following explicit form of the CAV algorithm for solving the system $Ax = b$.

**Algorithm 6** ***(The Nonbinary CAV Algorithm, [7]).***
***Initialization***: *$x^0 \in R^n$ is arbitrary.*
***Iterative Step***: *Given the k-th iterate $x^k$ do:*
***(1) Correction Calculation***: *The k-th correction vector $c^k = (c^k_j)$ is calculated by*

$$c^k_j = \lambda_k \sum_{i=1}^{m} \frac{b_i - \langle a^i, x^k \rangle}{\sum_{l=1}^{n} s_l (a^i_l)^2} a^i_j, \quad \textit{for all } j = 1, 2, ..., n. \tag{8}$$

***(2) Correction Application***: *The next iterate $x^{k+1}$ is calculated by*

$$x^{k+1} = x^k + c^k. \tag{9}$$

Again, the $\{\lambda_k\}_{k\geq 0}$ are user-chosen relaxation parameters. Although [7, Theorem 4.1] proves convergence only for the case $\lambda_k = 1$, for all $k \geq 0$, we apply the CAV algorithm with different relaxation parameters in the range $\delta \leq \lambda_k \leq 2 - \delta$, for some $\delta > 0$. This is justified by further progress that we made in studying the convergence of CAV in [3].

# 4 Experimental Results

In all the experiments reported below we used a fixed value $t_k = 0.5$, for all $k \geq 0$, in Definitions 2 and 3 and the value $\epsilon = 0.05$, in Definition 3. The progress of the sequences $\{\alpha_k\}_{k\geq 0}$ and $\{\beta_k\}_{k\geq 0}$ determines what we referred to earlier as the *speed* with which the binary steering process is steered. It is controlled by an integer *steering parameter* $S$, which is used in the definition of the following sequences

$$\alpha_k := (k/S)t_k \text{ and } \beta_k := 1 - (k/S)(1 - t_k), \text{ for all } k \geq 0, \tag{10}$$

We compared various values of the steering parameter and present here representative results for the three values $S = 100,000$, $S = 10,000$, and

$S = 1,000$, and we used $K = 1,000$ as our stopping iteration index. Figure 8 shows the behavior of the sequences $\{\alpha_k\}_{k\geq 0}$ and $\{\beta_k\}_{k\geq 0}$ for the first $1,000$ iterations with these three values of $S$. The smaller $S$ is the faster the process is steered and the high value $S = 100,000$, coupled with $K = 1,000$, practically amounts to no binary steering at all. In all cases the thresholding step (3) was applied with $t_K = 0.5$.

All runs were initialized with an initial uniform image $x^0 = (x_j^0)_{j=1}^n$ with the uniform value $x_j^0 = u$, for all $j = 1, 2, ..., n$, calculated by

$$u = \frac{\sum_{i\in I_h} b_i}{\sum_{i\in I_h}\sum_{j=1}^n a_j^i}, \tag{11}$$

where $I_h$ contains all indices $1 \leq i \leq m$ which represent rays in the horizontal view/projection. This means that the average uniform grayness of the initial iterate agrees with the average grayness of any solution, estimated from the measurements vector. Since we assume that the image is a square region subdivided into $n$ square pixels whose sides are equal to one unit of length, and that exactly one ray/line goes through each row of pixels in the horizontal view, the lengths of intersections of rays with pixels in this view are equal to one so the denominator of (11) is equal to $n$.

We present here the results of the binary steering process runs with the sequential ART (Algorithm 5) and the fully simultaneous CAV (Algorithm 6) with fixed relaxation parameters of $\lambda_k = \lambda = 0.05$, for all $k \geq 0$, and $\lambda_k = \lambda = 2.0$, for all $k \geq 0$. For other intermediate values of these parameters we observed a monotonic behavior of each algorithm, i.e., the various error plots for intermediate values lie in between those for these extreme values. In [5] only the fully simultaneous Cimmino algorithm was tested and with only one value of relaxation parameters which was $\lambda_k = \lambda = 1.0$, for all $k \geq 0$.

We used two different test images that were recently introduced in the literature. One is the third phantom of Vardi and Lee [14, Figure 3.1] (which appeared first in Fishburn *et al.* [9]). The second is one of the phantoms used by Carvalho *et al.* in [2] and which we nickname `Carvalho`, shown in Figure 9. Since the results that we obtained for the `Vardi` phantom are, in principle, similar to those for the `Carvalho` phantom – we present here only results for the latter.

The plots in Figures 1–6 describe the dependence of two types of reconstruction errors on the iteration index (iter). An iteration is considered one sweep through all equations of the system $Ax = b$, regardless of the nature (sequential or simultaneous) of the nonbinary algorithm in use. The *data error,* displayed in Figures 1–3, is denoted by *e-data*, and it is the sum, over all rays/lines, of the absolute values of the differences between the line-sums in the phantom and in the reconstructed image. The *image error,* displayed in Figures 4–6, is denoted by *e-image*, and it is the total number of locations (i.e., pixels) at which the reconstructed image value disagrees with the phantom value. There is a consistent correlation between these two errors by their similar behavior in the simulation tests. All errors are reported for the first $1,000$ iterations and for all three values of the steering parameter $S$ mentioned above. In Figures 1–6 the individual plots are marked with numbers (1)–(4) which represent the nonbinary algorithm and relaxation parameter used according to the following table:

| | Nonbinary Algorithm | Relaxation Parameter |
|---|---|---|
| (1) | ART | 2.0 |
| (2) | CAV | 0.05 |
| (3) | ART | 0.05 |
| (4) | CAV | 2.0 |

All the results demonstrated here were derived by using four views (projections): horizontal ($\longleftarrow$), vertical ($\downarrow$) and the two diagonal projections ($\swarrow$) and ($\searrow$) and no noise was introduced into the reconstruction model and measurements. Figure 7 displays all 12 reconstructed images after 1,000 iterations, with all algorithms, relaxation parameters and steering parameters mentioned above.

Our experiments with other phantoms (of the `Vardi` type) and with other values of the relaxation and steering parameters lead to the same conclusions as the reader will draw from examining the results presented here. In all cases the simultaneous algorithm (CAV) behaved with a large relaxation parameter $\lambda = 2.0$ better then with a small one $\lambda = 0.05$, while the sequential algorithm ART behaved better with the smaller relaxation parameter. The overall performance, from worst to best, was in the order (1)–(2)–(3)–(4) where the numbers refer to the table given above. This can be seen both in the error plots and in the reconstructed images in Figure 7.

Finally, observation shows that in our (still limited) experiments with the specific algorithms, parameters, phantoms, and, in particular, with the specific functions (10) for the steering process, the effect of the binary steering is hardly noticable when combined with the simultaneous nonbinary iterative algorithm, see the second and the last rows of Figure 7 and the appropriate error plots in Figures 1–6. In contrast with this, the binary process affects the sequential nonbinary iterative algorithm in a way that indicates its usefullness. Namely, for the smaller steering parameter $S = 1,000$, better results can be observed (compare right column images against left column images in first and third rows in Figure 7 and the appropriate error plots in Figures 1–6.)

As mentioned in the introduction, our binary steering process is based on the principles used by Herman in [11]. The experiments performed there lead to the conclusion that the method called there "BART" (for Binary ART) compares generally favorably with the other three techniques experimented with there. Specifically, compared with ART followed by thresholding at the end (called there "CART"), BART was better for nine data sets, worse for one data set, and the methods perform the same for the remaining eight date sets. We think that further work is necessary and warranted to more fully investigate the behavior of the binary steering process.

**Acknowledgements**

*We thank Samuel Matej for his collaboration on the first report on binary steering [5], and Gabor Herman and Attila Kuba for many useful discussions. We are indebted to Ronen Greenspan for his devoted work on the implementation and the computational research. We thank Bruno Carvalho and Gabor Herman for having shared with us their test and reconstructed images of [2] from where we took our second test image (the one we called* `Carvalho` *above) for our experiments. Two anonymous referees helped us greatly in bringing this paper to its final form and we are grateful to them for their comments. This work was supported by research grants 293/97 and 592/00 from the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities, and by grant HL-28438 of the National Institutes of Health (NIH).*

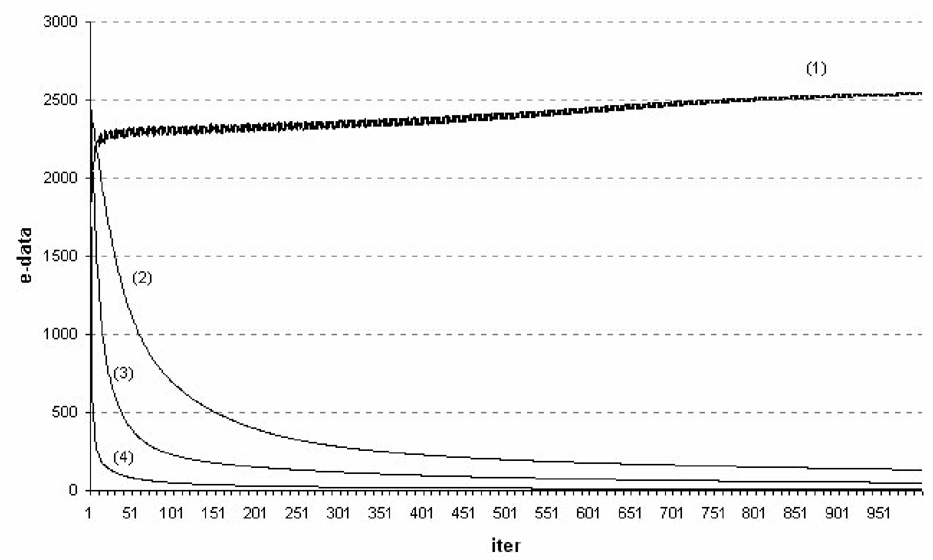

Figure 1. e–data versus iteration index (iter) plots for the `Carvalho` reconstructions obtained with steering parameter $S = 100,000$.

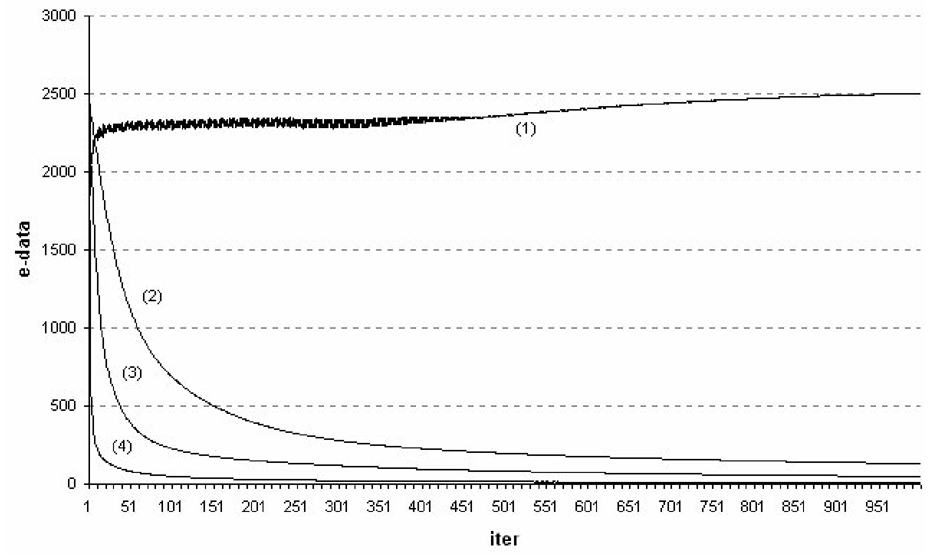

Figure 2. e–data versus iteration index (iter) plots for the `Carvalho` reconstructions obtained with steering parameter $S = 10,000$.

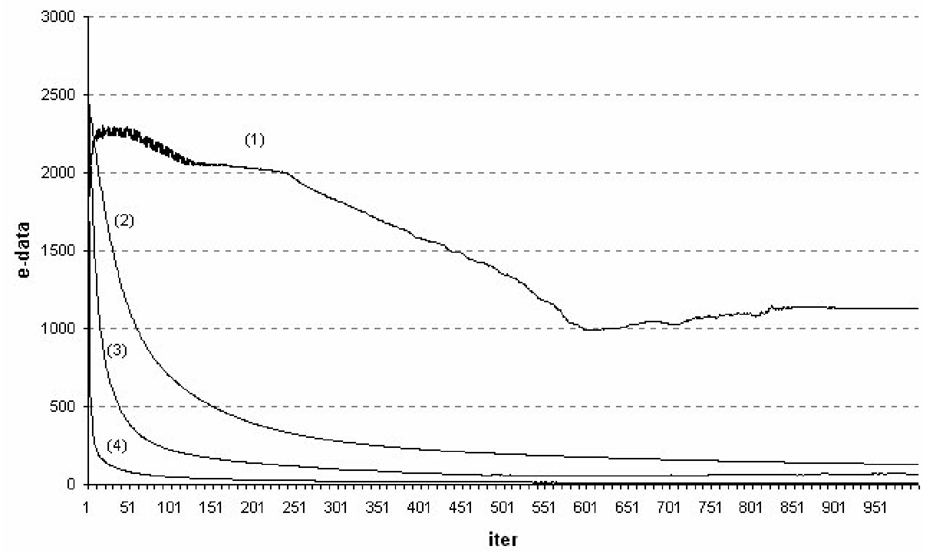

Figure 3. e–data versus iteration index (iter) plots for the `Carvalho` reconstructions obtained with steering parameter $S = 1,000$.

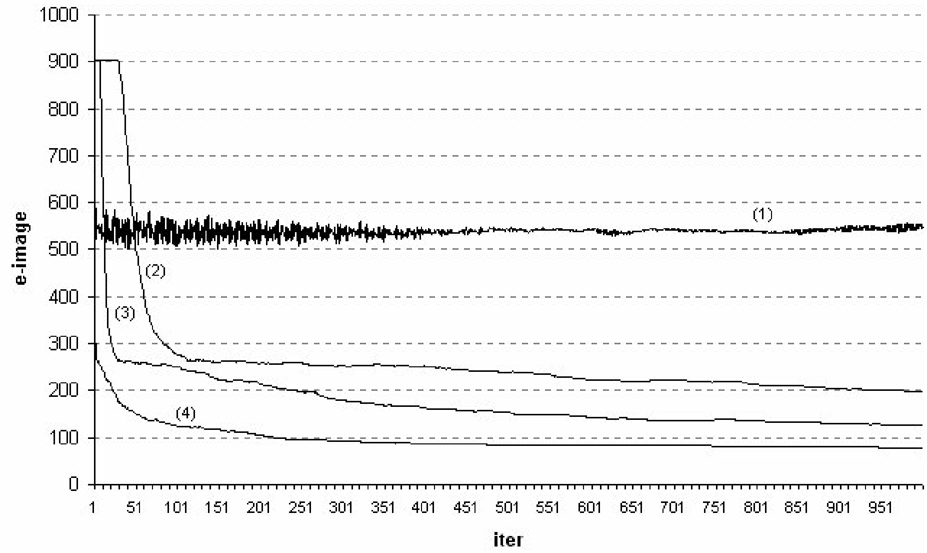

Figure 4. e–image versus iteration index (iter) plots for the `Carvalho` reconstructions obtained with steering parameter $S = 100,000$.

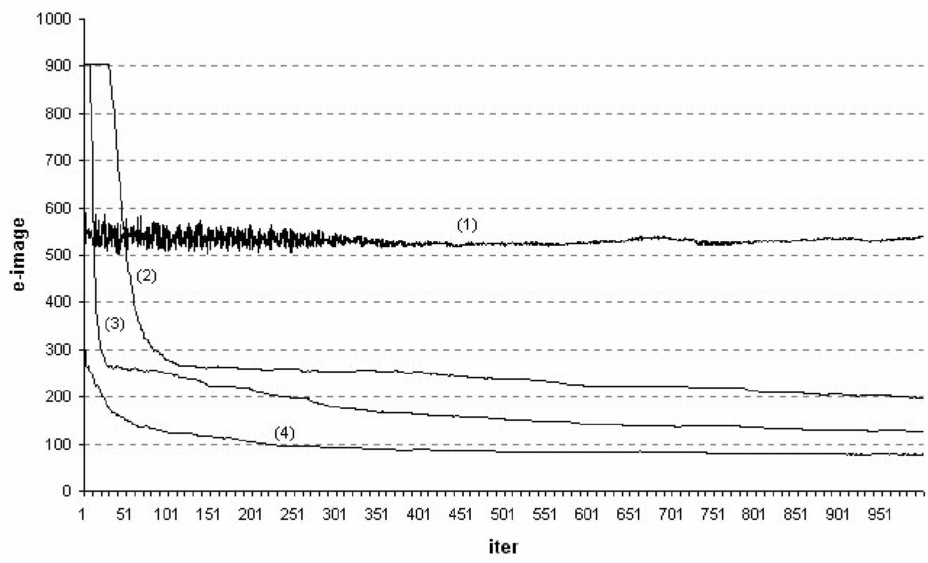

Figure 5. e–image versus iteration index (iter) plots for the `Carvalho` reconstructions obtained with steering parameter $S = 10,000$.

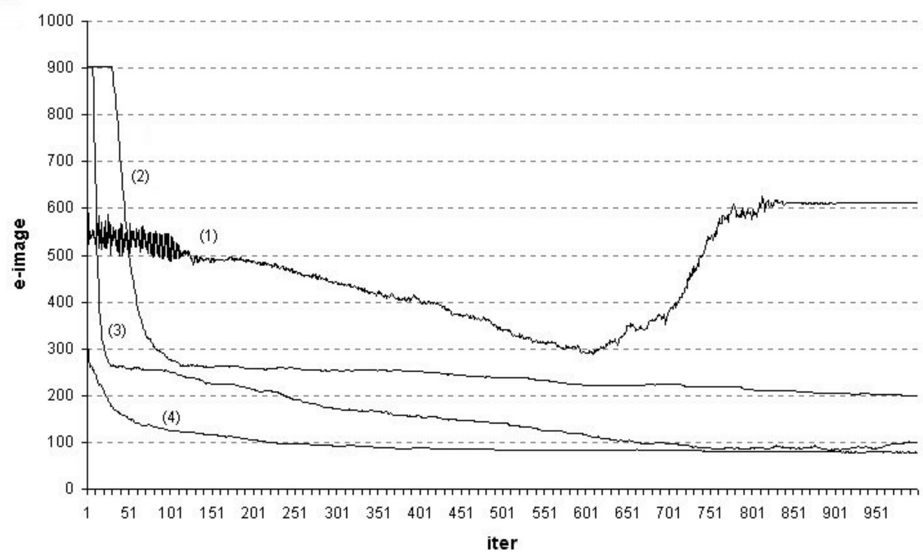

Figure 6. e–image versus iteration index (iter) plots for the `Carvalho` reconstructions obtained with steering parameter $S = 1,000$.

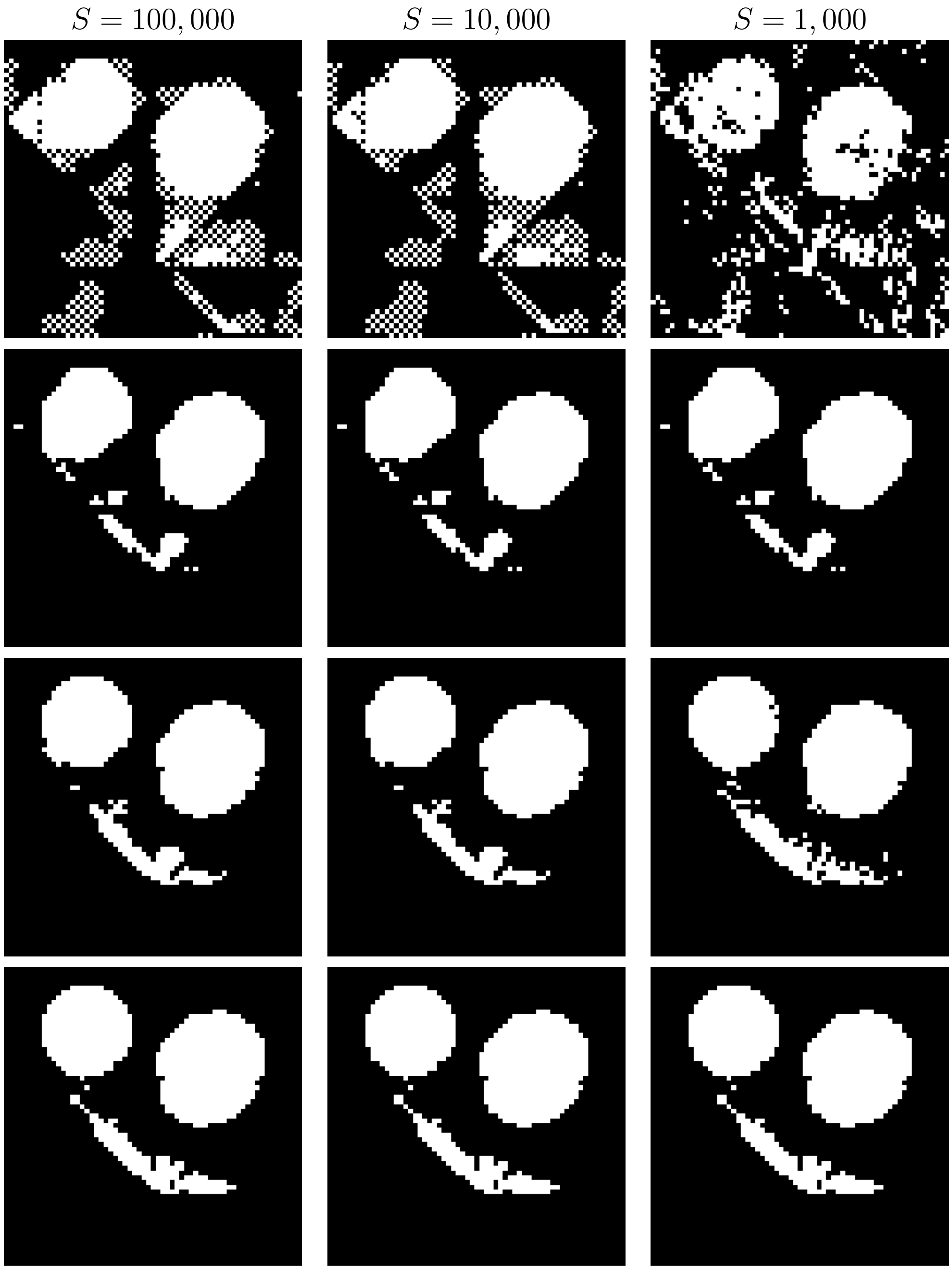

Figure 7. Reconstructions of the `Carvalho` phantom: $K = 1,000$, $S = 100,000, 10,000$ and $1,000$; algorithms (from top to bottom row): ART ($\lambda = 2.0$), CAV ($\lambda = 0.05$), ART ($\lambda = 0.05$) and CAV ($\lambda = 2.0$).

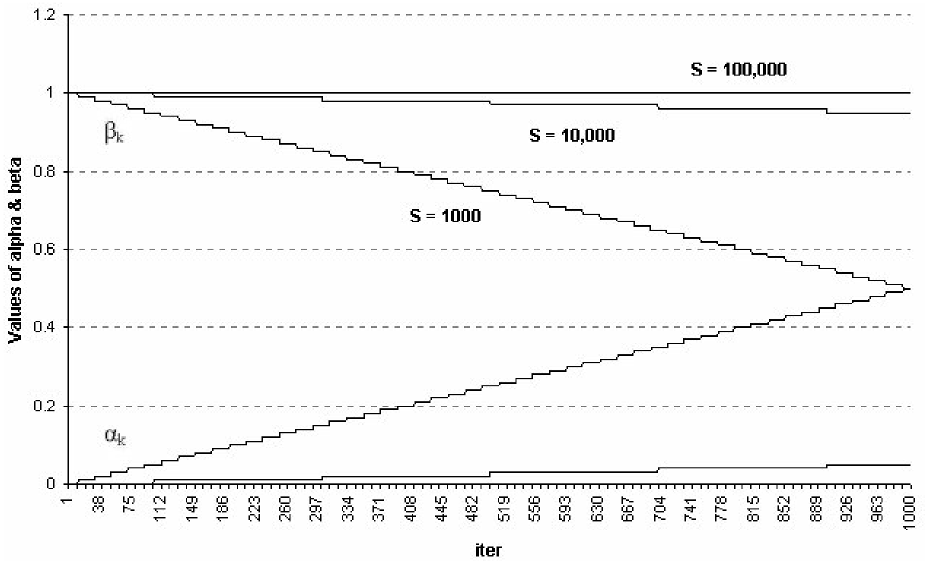

Figure 8. Plots of $\alpha_k$ and $\beta_k$ versus iteration index $k$ (iter) for steering parameter values $S = 1,000, 10,000$,and $100,000$.

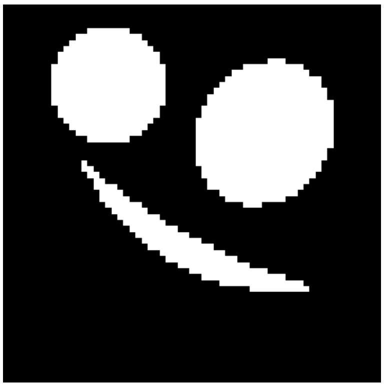

Figure 9. The `Carvalho` test image (phantom).

# References

[1] H.H. Bauschke and J.M. Borwein, On projection algorithms for solving convex feasibility problems, *SIAM Review* 38:367–426 (1996).

[2] B.M. Carvalho, G.T. Herman, S. Matej, C. Salzberg and E. Vardi, Binary tomography for triplane cardiography, in *Information Processing*

*in Medical Imaging*, (A. Kuba, M. Šámal and A. Todd-Pokropek, Eds.), Springer-Verlag, Berlin, Germany, 1999, pp. 29–41.

[3] Y. Censor and T. Elfving, Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem, *Technical Report*, April 2001.

[4] Y. Censor, T. Elfving and G.T. Herman, Averaging strings of sequential iterations for convex feasibility problems, in *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications* (D. Butnariu, Y. Censor and S. Reich, Eds.) Elsevier Science Publishers, Amsterdam, The Netherlands, 2001, pp. 101–114.

[5] Y. Censor and S. Matej, Binary steering of nonbinary iterative algorithms, in *Discrete Tomography: Foundations, Algorithms, and Applications* (G. T. Herman and A. Kuba, Eds.), Birkhäuser, Boston, MA, USA, 1999, pp. 285–296.

[6] Y. Censor and S.A. Zenios, *Parallel Algorithms: Theory, Algorithms, and Applications,* Oxford University Press, New York, USA, 1997.

[7] Y. Censor, D. Gordon and R. Gordon, Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems, *Parallel Computing* 27:777–808 (2001).

[8] G. Cimmino, Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari, La Ricerca Scientifica XVI, Series II, Anno IX 1:326–333.

[9] P. Fishburn, P. Schwander, L. Shepp and R. Vanderbei, The discrete Radon transform and its approximate inversion via linear programming, *Discrete Applied Mathematics* 75:39–61 (1997).

[10] R. Gordon, R. Bender and G.T. Herman, Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography, *Journal of Theoretical Biology* 29:471–481 (1970).

[11] G.T. Herman, Reconstruction of binary patterns from a few projections, in *International Computing Symposium 1973* (A. Günther, B. Levrat and H. Lipps, Eds.), North-Holland, Amsterdam, The Netherlands, 1974, pp. 371–378.

[12] G.T. Herman, Algebraic reconstruction techniques in medical imaging, in *Medical Imaging Systems Techniques and Applications: Computational Techniques* (C.T. Leondes, Ed.), Gordon and Breach, Amsterdam, The Netherlands, 1998, pp. 1–42.

[13] S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, *Bulletin de l'Académie Polonaise des Sciences at Lettres* A35:355–357 (1937).

[14] Y. Vardi and D. Lee, Discrete Radon transform and its approximate inversion via the EM algorithm, *International Journal of Imaging Systems and Technology* 9:155–173 (1998).