THE PERIODIC STEP GRADIENT DESCENT ALGORITHM - GENERAL ANALYSIS AND APPLICATION TO THE SUPER RESOLUTION RECONSTRUCTION PROBLEM

Tamir Sagi[1], *Arie Feuer*[2] *and Michael Elad*[3]
[1,2]Dept. of Electrical Engineering, Technion - Israel Institute of Technology.
Haifa 32000, Israel.
[3]Hewlett Packard Laboratories - Israel, Technion - IIT.
Haifa 32000, Israel.
[1]e-mail: stamir@sayfan.technion.ac.il
[2]e-mail: feuer@ee.technion.ac.il
[3]e-mail: elad@hp.technion.ac.il

ABSTRACT

Solving image reconstruction problems, especially complex problems like Super Resolution reconstruction, is very demanding computationally. Iterative algorithms are the practical tool frequently used for this purpose. This paper reviews the *Periodic Step Gradient Descent* (PSGD) algorithm, suggested as a sub-optimal algorithm for solving reconstruction problems (with emphasis on Super Resolution reconstruction problems). The PSGD differs from well-known iterative algorithms in the way the data of the problem at hand is processed. Whereas iterative algorithms process the entire given data in order to update the result, the PSGD updates the result progressively. This paper provides an analysis of the PSGD. We show that the PSGD has an efficient implementation, easy to achieve convergence conditions and fast convergence speed when applied to a Super Resolution reconstruction problem. The performance of the PSGD when applied to a Super Resolution reconstruction problem, is demonstrated by simulations and compared to the performance of other well-known algorithms.

1 INTRODUCTION

One of the problems widely discussed in the image processing literature, is the problem of image reconstruction (restoration) [1]-[2]. Throughout the last decades as computer technology develops, there is a growing interest in reconstruction problems that are more demanding computationally. One such problem is the problem of Super Resolution reconstruction. In this problem a single improved resolution image is reconstructed from a set of geometrically warped, blurred, downsampled and noisy measured images.

The recent work by Elad and Feuer [3] presented a new approach toward the Super Resolution reconstruction problem. In [3] it is suggested to model the Super Resolution reconstruction problem using the well-known classical single image restoration equation ([1]-[2])

$$\underline{Y} = C\underline{X} + \underline{E} \tag{1}$$

where $\underline{Y}$ is a known $[L \times 1]$ vector of measurements, C is a known $[L \times N]$ matrix representing a linear distortion operator, $\underline{E}$ is a $[L \times 1]$ additive noise vector (assumed to be a white Gaussian noise with zero mean and a known covariance matrix) and $\underline{X}$ is a $[N \times 1]$ vector of unknowns. (Note that the images are represented using a columnwise lexicographic ordering). That way, methods associated with solving reconstruction problems may be used for solving the more complex Super Resolution reconstruction problem.

When facing a reconstruction problem our goal is to get an estimate of the unknown vector $\underline{X}$. The common solution associated with reconstruction problems, is the Least Squares (LS) solution ([2]). This solution is achieved by solving the quadratic minimization problem

$$\min_{\underline{X}}\left[(\underline{Y} - C\underline{X})^T(\underline{Y} - C\underline{X})\right]. \tag{2}$$

The solution of the minimization problem presented in (2) is

$$(C^T C)\hat{\underline{X}} = (C^T \underline{Y}). \tag{3}$$

In order to actually solve the equations set (3), the inverse of the matrix $C^T C$ (which for further reference will be denoted by R) must be calculated and stored in memory. The dimensions of R in a practical reconstruction problem are very large, thus making the invertion task computationally impossible and storage very demanding. These create the motivation to investigate indirect methods to solve reconstruction problems.

Iterative algorithms are usually suggested as the practical tool for solving reconstruction problems ([2]). Typically, iterative algorithms refer to the given data (the matrix C and the vector $\underline{Y}$) as a package. At each iteration this whole package is processed and a new estimate of the solution is calculated. In this paper we analyze an algorithm which processes the given data one equation at a time and not as one package, we refer to this algorithm as the *Periodic Step Gradient Descent* (PSGD).

The steady state solution of the PSGD is sub-optimal to the LS solution, however in this paper the PSGD is investigated as a stand-alone algorithm. We settle for a sub-optimal solution and investigate the PSGD as an algorithm for solving reconstruction problems.

Using the PSGD algorithm for Super Resolution reconstruction brings out the main advantages of the algorithm. Simulations we performed show that typically the PSGD converges to the steady state solution faster and with low computational cost when compared to well-known algorithms such as Steepest Descent (SD), Normalized Steepest Descent (NSD), Jacoby (J), Gauss-Siedel (GS), Successive Over Relaxation (SOR) and Conjugate Gradient (CG) (those algorithms are reported in detail in references [4]-[9]).

This paper is organized as follows: Section 2 presents the PSGD algorithm, the PSGD algorithm is analyzed and the main results are presented. Section 3 presents a comparison between the PSGD and other known algorithms, when applied to the Super Resolution reconstruction problem. Simulations results are presented in Section 4 and Section 5 concludes the paper.

2 THE PERIODIC STEP GRADIENT DESCENT

Consider a problem that may be formulated by equation (1). The PSGD algorithm, for some arbitrary initial solution, is given by

$$\underline{X}^{j+1} = \underline{X}^{j} - \mu \underline{C}(j \bmod L)^T [\underline{C}(j \bmod L)\underline{X}^{j} - y(j \bmod L)] \tag{4}$$

where j is the step index, $(j \bmod L)$ is a symbol represents the periodic 1 to L count, $\underline{C}(i)$ is the i'th row of the matrix C, $y(i)$ is the i'th element of the vector $\underline{Y}$, $\underline{X}^j$ is an estimate of $\underline{X}$ and μ is the stepsize parameter.

This algorithm is an implementation of an idea equivalent to that of the Stochastic Gradient algorithm - the LMS [6]-[7], to a non-stochastic problem. The data of the problem is fed to the algorithm one equation at a time and not as a whole package, as in the iterative algorithms mentioned above ([4]-[9]).

The PSGD is mentioned in the work of Bertsekas, reported in [5], [10]. There it is suggested to embed the PSGD (which is referred to as the *Incremental Gradient*) and the SD as well as other intermediate methods within a one parameter hybrid algorithm for LS problems, such as neural network training problems. This idea enables control over the degree of incrementalism of the algorithm via a non-negative scalar parameter. Hence it is possible to start updating the solution using the PSGD in order to gain fast convergence while far from the LS solution, and gradually lower the incrementalism degree of the algorithm in order to ensure convergence to the LS solution.

As for the PSGD itself, it is shown in [5], [10] that it converges to the LS solution if the stepseize parameter μ tends to zero and for the diminishing stepsize scheme.

In this paper however we consider the PSGD as a sub-optimal algorithm. We are interested in investigating the PSGD as a stand-alone algorithm, accepting the fact that the PSGD's steady state solution is not optimal in the LS sense.

In order to further investigate the PSGD algorithm it is written as an iterative algorithms in which one pass through the whole data is considered as one iteration, that way the PSGD algorithm takes the form

$$\underline{X}^{k+1} = A_R \underline{X}^k + B_R \underline{Y} \tag{5}$$

where A_R is a $[N \times N]$ matrix given by

$$A_R = \left[I - \mu \underline{C}(L)^T \underline{C}(L)\right]\left[I - \mu \underline{C}(L-1)^T \underline{C}(L-1)\right]\cdots \cdots\left[I - \mu \underline{C}(1)^T \underline{C}(1)\right] \tag{6}$$

and B_R is a $[N \times L]$ matrix given by

$$B_R = \left[A(L)\cdots A(3)A(2)\mu\underline{C}(1)^T \,|\, A(L)\cdots A(3)\mu\underline{C}(2)^T \,|\, \cdots \,|\, A(L)\mu\underline{C}(L-1)^T \,|\, \mu\underline{C}(L)^T\right] \tag{7}$$

where $A(i) = I - \mu\underline{C}(i)^T\underline{C}(i)$ and I is the identity matrix.

Writing the PSGD algorithm in the notation of equation (5) enables a straightforward analysis of the algorithm, through analyzing the properties and structure of the matrix A_R.

Note that when the PSGD converges to a steady state solution the solution will be

$$\underline{X}^{PSGD} = (I - A_R)^{-1} B_R \underline{Y} \tag{8}$$

2.1 Convergence Conditions of the PSGD

Clearly, for the PSGD algorithm to converge all the eigenvalues of A_R must be within the unit circle ([6]-[7]). Generally, the eigenvalues of A_R depend on the choice of the stepsize μ and on the matrix C.

We have the following condition guaranteeing the convergence of the PSGD:

THEOREM 1 [11]:

Let the parameter μ satisfy the condition

$$\mu \in \left(0, \frac{2}{\max_i \underline{C}(i)\underline{C}(i)^T}\right). \tag{9}$$

Then $\|A_R\|_2 < 1$ if and only if C is full rank.

PROOF:

To begin with, it should be noted that when the parameter μ is chosen according to (9), it is ensured that each matrix $\{[I - \mu\underline{C}(i)^T\underline{C}(i)]\}_{i=1}^{L}$ has one eigenvalue within the unit circle and $N-1$ eigenvalues equal to 1, as a result $\|A_R\|_2 \le 1$.

The proof requires two stages:

(*a*) Suppose that C is full rank (i.e. the rows of C span $\Re^N$) together with the fact that $\|A_R\|_2 = 1$ (in contradiction to the theorem).

Generally, there is a vector $\underline{v}$ that satisfies $\|\underline{v}\|_2 = 1$, therefore $\|A_R\underline{v}\|_2 = 1$. In that case it can be shown that

$$\left\| \left[I - \mu\underline{C}(1)^T\underline{C}(1)\right]\underline{v} \right\|_2 = 1, \tag{10}$$

writing the L_2 norm explicitly along with the fact that the parameter μ satisfies (9) yields

$$\underline{C}(1)\underline{v} = 0. \tag{11}$$

As a result of (11) it is easy to show that $\underline{v}$ is the eigenvector of the matrix $[I - \mu\underline{C}(1)^T\underline{C}(1)]$ associated with the eigenvalue 1.

Using the same methodology, one can show that $\underline{v}$ is orthogonal to each and every row of C. Since the rows of C span $\Re^N$ the vector $\underline{v}$ must satisfy $\underline{v} = 0$. As a result the assumption that $\|A_R\|_2 = 1$ is contradicted and the *if* part of the theorem proven.

(*b*) Assume now that $\|A_R\|_2 < 1$ together with the fact that the rows of C do not span $\Re^N$ (in contradiction to the theorem).

In this case one can choose a vector $\underline{v}$ that is orthogonal to each and every row of C and satisfies $\|\underline{v}\|_2 = 1$. Hence the equation $\|A_R\underline{v}\|_2 = 1$ is true, contradicting the assumption that $\|A_R\|_2 < 1$, that way the *only if* part of the theorem is proven.

A detailed proof can be found in [11]. ■

Clearly, from the results above, for a nonsingular restoration problem (if the rows of C span $\Re^N$ then R is positive definite), the choice of the stepsize parameter is an easy and practical task.

2.2 Convergence of the PSGD to the LS Solution

There are three cases in which the PSGD converges to the LS solution. The case where the stepsize tends to zero and the case of a diminishing stepsize scheme are discussed in [5], [10]. The third case in which the PSGD achieves the LS solution is presented in the following theorem:

THEOREM 2 [11]:

The PSGD converges to the LS solution if THEOREM 1 *is satisfied and* C *is a* $[N \times N]$ *matrix.*

PROOF:

It can be shown that $B_R C = I - A_R$. If C is a full rank $[N \times N]$ matrix, one can see that $I = C(I - A_R)^{-1} B_R$. Multiplying the previous equation by $(C^T C)^{-1} C^T$ from the left and by $\underline{Y}$ from the right proves the theorem. ■

2.3 Computer Resources Consumption

When evaluating an algorithm, computer resources consumption is a major factor that must be considered. Two aspects must be taken into account:

1. The number of mathematical operations (additions, subtractions, multiplications and divisions) required to implement an algorithm.
2. The number of memory cells required for implementation.

In Table1 we list computer resources required to implement various algorithms mentioned earlier. In order to present one number that represents the number of operations required to implement an algorithm we used Patterson and Hennessy's [12] Normalized Floating Point Operations measure (Note - multiplications by zero were not counted).

Table1 - Computer resources consumption.

Memory Requirements	Preliminary Calculations	Calc. per Iteration	Alg.
$2N$	---	$4Lq + 2N$	SD
$2N + L$	---	$L(6q + 1) + 4N$	NSD
N	---	$4Lq$	PSGD
$3N$	$2Lq + 4N$	$4Lq + 3N$	J
$Snnz(R) + 2N$	$2L(3N + nnz(R))/2 + 4N$	$2N^2$	GS
$Snnz(R) + 2N$	$2L(3N + nnz(R))/2 + 4N$	$N(2N + 3)$	SOR
$3N + L$	---	$L(6q + 5) + 6N$	CG

where q is the number of nonzero elements in a row of C (it is assumed that all rows of C have the same number of elements), $nnz(R)$ is the number of nonzero elements in R and S is a number that represents the memory overhead required in order to save a sparse matrix in memory [13].

2.4 Rate of Convergence

The rate of convergence of the PSGD algorithm is determined according to the maximal absolute eigenvalue of the matrix A_R. Simulations show ([11]) that in general, the larger the ratio L / N the faster the PSGD converges. Analytic investigation of the eigenvalues of A_R seems very difficult, since the eigenvalues of A_R depend on the stepsize parameter μ and the matrix C in a complicated way. This is not unusual when discussing rates of convergence of algorithms. Even in the case of the simple SD, the rate of convergence depends on the maximal eigenvalue of the matrix R, which is hard to calculate.

Moreover, rate of convergence strongly depend upon the problem to be solved. Therefore it is quite common to use a benchmark problem in order to compare rates of convergence. This is done in section 3.

3 APPLICATION OF THE PSGD TO SUPER RESOLUTION RECONSTRUCTION

In this section we consider the performance of the PSGD when applied to Super Resolution reconstruction. We attempt to present a performance envelope that takes into consideration the speed of convergence and computer resources consumption. The performance of the PSGD is compared to that of other well-known algorithms.

In order to compare speed of convergence and computer resources consumption, we synthesize a small-scale Super Resolution problem following the model suggested in [3]. We start with a 20×20 pixels ideal image, from which we create 20 samples of size 10×10 pixels. The distortion operator for each image includes affine motion (which parameters were randomly chosen), uniform blur (using a 3×3 kernel) and $1:2$ decimation in each axis. A random zero mean Gaussian noise ($\sigma = 3$) was added to each sample image. It is important to notice that the image $\underline{Y}$ itself has no bearing on the simulations results.

Figure 1 depicts the speed of convergence results. It can be seen that for Super Resolution reconstruction problems the PSGD converges to the steady state solution very rapidly.

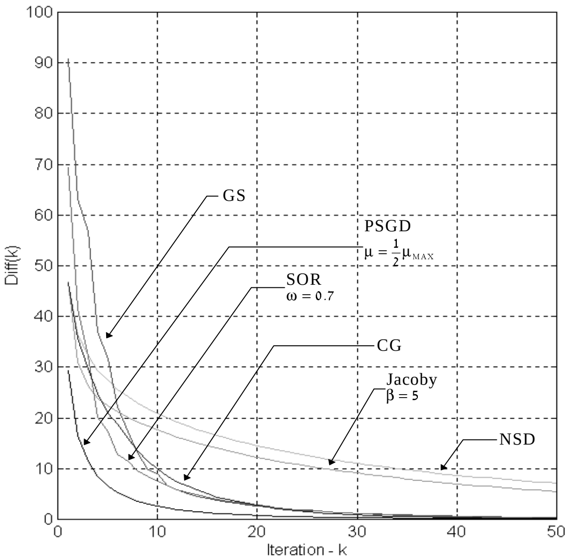

Figure 1 - Speed of Convergence.
The error percentage, $100\left\|\underline{X}^k - \underline{X}^{ss}\right\| / \left\|\underline{X}^{ss}\right\|$, as a function of the iterations.

Figure 2 presents the convergence percentage of each algorithm as a function of the number of operations required to implement the algorithm (calculated according to Table 1).

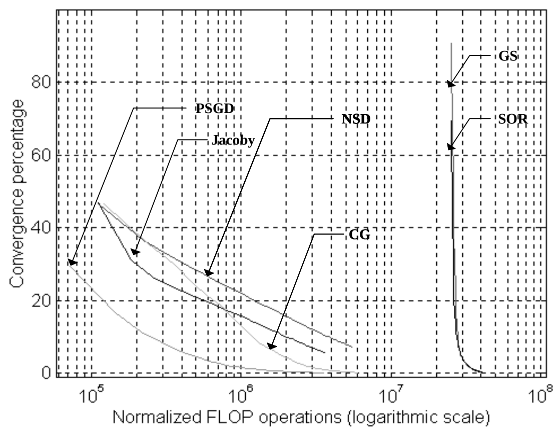

Figure 2 - Speed of convergence Vs. computational load.

One can see that when a performance envelope is considered, the fact that the PSGD has fast convergence speed and low computational requirements makes it a very attractive algorithm.

4 SIMULATIONS RESULTS

In this section we present some simulations results, in order to illustrate the bottom-line quality of the PSGD's result for Super Resolution reconstruction problems.

Again we follow the model suggested in [3]. We start with a 100×100 pixels ideal image, from which we create 25 samples of size 50×50 pixels. The distortion operator for each image includes affine motion (again, randomly chosen), uniform blur (using a 3×3 kernel) and $1:2$ decimation in each axis. A random zero mean Gaussian noise $(\sigma = 3)$ was added to each sample image.

In Figure 3 the LS solution and the PSGD solution are presented along with the original image and the best sample bilaterally interpolated to 100×100 pixels size.

Figure 3 - Top Left, the ideal image. Top Right, bilinear interpolation. Bottom Left, LS solution. Bottom Right, PSGD solution.

Super Resolution reconstruction from a real video sequence is presented in Figure 4.

Figure 4 - Super Resolution reconstruction from a real video sequence. Left - PSGE result. Right - Bilinear interpolation.

A video sequence was captured by a home video camera and saved on a computer disk. A set of 30 images of size 144×144 pixels was taken for the reconstruction process. The motion between the first image in the set and the other 29 images was compensated, using an affine motion model. The target was to reconstruct one image with 288×288 pixels. One pass of the PSGD algorithm, through the data, was applied to achieve the result presented in the left-hand side of Figure 4. The right-hand side image in Figure 4 is the best image from the original video sequence bilaterally interpolated to 288×288 size, for comparison.

It can be seen that the quality of the result achieved by the PSGD algorithm is very good.

5 CONCLUSIONS

In this paper we reviewed the properties and performance of a sub-optimal algorithm, referred to as the PSGD algorithm. Our goal was to show that the PSGD can be a powerful tool for solving computationally demanding image reconstruction problems. We have shown that from algorithmic point of view the PSGD is an efficient, fast converging and easy to implement algorithm. Through simulations we demonstrated that for large dimensional reconstruction problems, the PSGD's result is of good quality not inferior to the quality of the LS solution.

6 REFERENCES

[1] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[2] R. L. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images*, Kluwer Academic Publishers, 1991.

[3] M. Elad and A. Feuer, "Restoration of Single Superresolution Image From Several Blurred, Noisy and Under-Sampled Measured Images", IEEE Trans. Image Processing, Vol. 6, No. 12, December 1997.

[4] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley Publishing company, 1984.

[5] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[6] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, 1986.

[7] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1990.

[9] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, 1791.

[10] D. P. Bertsekas, "A New Class of Incremental Gradient Methods For Least Squares Problems", LIDS-P-2301, Lab. for Info. and Dec. Systems MIT, August 1996.

[11] T. Sagi, "Analysis of the PSGD Algorithm and it's Application to the Super Resolution Reconstruction Problem", Ms.c. Thesis, The Technion - Israel Institute of Technology, March 1998. (In Hebrew).

[12] D. A. Patterson and J. L. Hennessy *Computer Architecture A Quantitative Approach*, Second Edition, Morgan Kaufmann Publishers, 1996.

[13] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, 1981.