Diagonal Variable Matrix Method in Solving Inverse Problem in Image Processing

Dick Mun Chang[1], *Hong Seng* Sim[1,*], *Yong Kheng* Goh[1], *Sing Yee* Chua[2], and *Wah June* Leong[3]

[1]Centre for Mathematical Sciences, Universiti Tunku Abdul Rahman, Malaysia.
[2]Centre for Advanced and Sustainable Materials Research, Universiti Tunku Abdul Rahman, Malaysia.
[3]Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Malaysia.

Abstract. In this paper, we introduce a new gradient method called the Diagonal Variable Matrix method. Our proposed method is aimed to minimize H_{k+1} over the log-determinant norm subject to weak secant relation. The derived diagonal matrix H_{k+1} is the approximation of the inverse Hessian matrix, which enables the calculation of the search direction, $d_k = -H_{k+1}g_k$, where g_k denotes the gradient of the objective function. The proposed method is coupled with the backtracking Armijo line search. The proposed method is specifically designed to reduce the number of iterations and training duration, particularly in the context of solving large-dimensional problems. Finally, as a practical illustration, the proposed method is applied to solve the image deblurring problem, and its performance is analyzed using image quality metrics. The results demonstrate that the proposed method outperforms various conjugate gradient (CG) methods and multiple damping gradient method.

1 Introduction

Optimization has been a popular research problem in various domains. It covers a wide range of fundamental concepts used in mathematics, engineering, computer science, economics, and more. Essentially, optimization requires identifying the best solution from a set of potential solutions, while often adhering to predefined constraints.

The initial step in optimization involves formulating an objective function, also known as a cost function. This function quantifies what one aims to minimize (e.g., cost, error) [8]. The primary objective is to determine the input or parameter values that yield the optimal value of this function. In many cases, optimization problems introduce constraint conditions that the solution must adhere to. Constraints can manifest as either equality constraints or inequality constraints (refer [8]).

Optimization has been commonly used in numerous fields of application such as neural networks, image processing, and computer science. Among the various applications considered, our main focus is on image processing, since it represents the practical implementation domain of the method proposed in this paper. Image deblurring is a fundamental task in image processing that aims to restore images affected by blur caused by factors such as motion, defocus, or optical aberrations.

Some challenges will be faced when we recover a blurred image, such as:

*e-mail: simhs@utar.edu.my

- Ill-posedness: The problem of recovering sharp images from blurred ones is often ill-posed, meaning it lacks unique solutions or is sensitive to small changes in the input data.
- Blur Kernel Estimation: In the cases of unknown or complex blur patterns, errors in blur kernel estimation can significantly affect the quality of the deblurred image, leading to inaccurate results.
- Trade-off between Sharpness and Noise: There is often a trade-off between recovering sharpness in the image and suppressing noise and artifacts introduced during the deblurring process.

In the field of image deblurring, many methods and algorithms have been published such as the blind deconvolution technique [3] and the iterative deconvolution [9]. Each method has its advantages and limitations. Blind deconvolution methods aim to recover sharp images and blur kernels without prior knowledge of the blurring process. Iterative deconvolution, or optimization-based methods such as steepest descent (SD) aim to iteratively refine the estimate of the original image by minimizing the difference between the blurred image and the estimated image. In this paper, we use iterative deconvolution as an application for image deblurring, due to it often comes with convergence guarantees, ensuring that the algorithm converges to a stable solution under certain conditions [19]. This provides confidence in the reliability and convergence properties of the deblurring process.

Numerous optimization algorithms, including gradient descent, Newton's method, and genetic algorithms, employ iterations to incrementally enhance the solution until a satisfactory optimum is achieved. The selection of an appropriate optimization algorithm and the specific details of the iterative method can vary depending on the nature of the problem(e.g., convexity vs. non-convexity, differentiability of the objective function) and the optimization objectives (e.g., minimizing or maximizing the objective function, discovering global or local optima)[14].

The SD method is an *iterative method* commonly integrated into the iterative process of optimization algorithms to minimize a function. The key idea behind the gradient descent method is to iteratively update the solution in the direction of the negative gradient, as this direction represents the steepest decrease in the objective function. By repeatedly moving in the direction of decreasing function values, the algorithm aims to converge to the optimal solution [8].

Quasi-Newton is another popular gradient method in optimization. However, the quasi-Newton method has a main drawback: it relies on the computation of the inverse Hessian matrix. The Hessian matrix is a full-rank matrix. Inverting a full rank matrix is computationally expensive. In this paper, we proposed a spectral gradient method with the spectral parameter approximating the inverse Hessian's matrix. The approximated Hessian matrix, in this case, takes the form of a diagonal matrix, offering computational efficiency and less memory storage. The proposed method is tested in blurred images and a comparison of its performance is conducted with state-of-the-art methods by using image quality metrics and number of iterations.

In the forthcoming sections, the paper will be structured as follows: in Section 2, we review some iterative methods in optimization. In Section 3, we derive the diagonal variable matrix (DVM) method and outline the algorithm of the DVM method and backtracking Armijo conditions. In Section 4, we apply the proposed method in an image deblurring system. It provides the result of the performance of the DVM method and compares it with some state-of-the-art methods presented in Section 2.

2 Iterative Method in Optimization

In this section, we consider the optimization problem

$$\min_{x \in \mathbf{R}^n} f(x)$$

where $f(x)$ is twice differentiable function. x is an $n \times 1$ decision vector. A generic formula for iterative updates in optimization is represented as:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1}$$

where x_k represents the approximated solution of the optimization problem at step k, α_k is the step size, and d_k is a direction vector. Various existing methods have been developed based on this updating formula.

In the following subsections, several iterative methods in optimization are discussed, including the SD method, the conjugate gradient (CG) method, the quasi-Newton method, and the spectral gradient family.

2.1 Steepest Descent

One of the oldest methods in the iterative method of optimization is the SD method, developed by Cauchy in 1847. SD is an iterative approach that starts with an initial guess and moves in the direction of the negative gradient of the function being minimized. It is a simple and widely used optimization method, moving in the direction of the negative gradient of the objective function [4]. SD converges slowly when the objective function is not well-conditioned.

This approach is vital to the development of optimization theory. It is a first-order method, simple to implement, requires less memory storage, and computationally inexpensive. However, due to zig-zagging behavior, it has slow convergence when it is near the minimum point. Therefore, increasingly complex methods such as the CG and quasi-Newton methods are frequently employed [17].

2.2 Conjugate Gradient

Moving forward, the Conjugate Gradient (CG) method is a commonly used iterative method for solving linear systems of equations and optimization problems. Hestenes and Stiefel (1952) introduced the CG approach for minimizing a large linear function. The CG method gained popularity due to its properties as an iterative method. Currently, the CG method is widely employed for non-linear challenges in large-scale systems. In the non-linear CG method, the search direction, d_k, in the iterative update equation is generated using this rule:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{2}$$

where g_{k+1} is the current gradient vector. The β_k represents an updating parameter to control the direction in the system. Table 1 shows some CG update parameters.

The CG method introduced by Fletcher and Reeves [10] is widely recognized as the first nonlinear CG algorithm, with a focus on nonlinear optimization. Subsequently, the CG method modified by Hestenes and Stiefel [13] primarily addresses symmetric, positive-definite linear systems. For large-scale problems, CG update parameters that circumvent the need for computing the Hessian matrix are typically preferred. When dealing with strongly

Update Formula	**Authors**
$\beta_k = \frac{g_{k+1}^T y_k}{d_k^T y_k}$	Hestenes and Stiefel (1952)
$\beta_k = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}$	Fletcher and Reeves (1964)
$\beta_k = \frac{g_{k+1}^T y_k}{\|g_k\|^2}$	Polak (1969) and Ribiere and by Polyak(1969)
$\beta_k = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$	Liu and Storey (1991)
$\beta_k = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$	Dai and Yuan (1999)
$\beta_k = \left(y_k - 2d\frac{\|y_k\|^2}{d_k^T y_k}\right)^T \frac{g_{k+1}}{d_k^T y_k}$	Hager and Zhang (2005)

Table 1. The update parameter, β_k for various CG methods. Here $y_k = g_{k+1} - g_k$.

convex quadratic functions, all the mentioned parameter choices in the list above are equivalent to an exact line search. However, for non-quadratic functions, the selection of each parameter leads to varying performance outcomes. Table 1 shows the incremental improvement on the CG methods over time.

All methods that include conjugate directions are quadratically convergence. This property allows one to find a quadratic function's minimum point in a maximum of n steps. Since a quadratic may fairly accurately approximate any general function as it approaches the optimum point, any quadratically convergent method should be able to identify the optimum point in a finite number of iterations. The convergence of the CG can be sensitive to the choice of the initial guess. Poor initializations may lead to slow convergence or convergence to suboptimal solutions. Furthermore, the approach might need more than n iterations for ill-conditioned quadratic problems due to the accumulation of rounding errors. The CG method performs better than the SD method in spite of this drawback [20]. As compared to Newton and quasi-Newton methods, the CG approach is less efficient but has the advantage of not requiring the storage of Hessian matrices [14].

2.3 Quasi-Newton Method

The quasi-Newton method encompasses a family of methods derived from Newton's method. Newton's method finds the solution to the optimization problem by searching for the roots of $\nabla f = 0$ [22]. Newton's method employs the actual Hessian matrix, while the quasi-Newton method uses an approximated version of the Hessian matrix. Table 2 presents the updating formulas used in the quasi-Newton family. This method has gained popularity in the field of optimization because it strikes a balance between the computational intensity of Newton's method and the simplicity of gradient-based methods.

The rank-one correction algorithm is effective for cases involving a constant Hessian matrix, such as the quadratic scenario. However, it tends to perform poorly in nonquadratic

Name	**Updating Formula**(B_{k+1})
Symmetric Rank One (SR1)	$B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$
Broyden Family	$B_k + \frac{s_k^T y_k - s_k^T B_k s_k}{(s_k^T B_k s_k)^2} B_k s_k s_k^T B_k$
Davidon, Fletcher, and Powell (DFP)	$\left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) B_k \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) + \frac{y_k y_k^T}{y_k^T s_k}$
Broyden, Fletcher, Goldfarb, and Shanno (BFGS)	$B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k^T}{s_k^T B_k s_k}$

Table 2. The update formula, B_k for various quasi-Newton methods. Here B_k is an approximation to the Hessian matrix, I is $n \times n$ identity matrix and $s_k = x_{k+1} - x_k$.

cases [5]. Next, initially developed by Davidon [7] and subsequently refined by Fletcher and Powell [11], the DFP algorithm provides an alternative. By replacing the rank-one correction formula with the DFP update B_{k+1}, the DFP algorithm can resemble the rank-one algorithm. Additionally, the BFGS algorithm inherits the positive definiteness property of the DFP algorithm. Notably, the BFGS update demonstrates robustness even when line searches are imprecise. Consequently, among the quasi-Newton methods, BFGS stands out as the most popular and potent technique due to its ability to maintain effectiveness in the presence of sloppy line searches.

However, the quasi-Newton method does come with certain drawbacks. It can be memory-intensive and computationally costly due to the computation of Hessian matrix[14]. Quasi-Newton methods require storing and updating an approximation of the Hessian matrix or its inverse. This can be memory-intensive, especially for large-scale optimization problems with many variables. Apart from this, one of the key drawbacks is the high computational cost. It can be computationally costly to update the Hessian matrix or its inverse approximation at each iteration, especially for high-dimensional situations.

2.4 Spectral Gradient Family

In the spectral gradient family, Barzilai and Browein proposed a two-point step size gradient method, named as Barzilai-Browein (BB) gradient method for the unconstrained minimization of a differentiation function $f : \mathbf{R}^n \to \mathbf{R}$ [2]. The method uses a two-point approximation of the secant equation based on quasi-Newton methods to determine the step size. In the BB gradient method, the gradient iteration form will be given as,

$$x_{k+1} = x_k - M_k g_k, \tag{3}$$

where $M_k = \lambda_k I$ and g_k is the gradient vector. In order to mimic the quasi-Newton method, the matrix M_k is derived from the secant equation and has the following λ_k,

$$\lambda_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \tag{4}$$

where $y_{k-1} = g_k - g_{k-1}$ and $s_{k-1} = x_k - x_{k-1}$. Note that, by symmetry, $\|M_k^{-1} s_{k-1} - y_{k-1}\|$ in relation to λ_k and obtain

$$\lambda_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}. \tag{5}$$

The BB method does not require the computation of full-rank matrices and does not require line search. Barzilai and Borwein demonstrated that their method is R-superlinearly convergence for the quadratic case [2]. Raydan and Svaiter combined the BB method with the SD method [21]. This method demonstrates that the limitations of the SD approach are not related to the choice of the search direction but instead stem from Cauchy's selection of the step length.

Sim et al. proposed the Multiple Damping Gradient (MDG) method by expressing M_k as B_k^{-1} [23]. MDG method has been developed to dampen the vector of gradient and decrease the function value simultaneously.

3 Diagonal Variable Matrix Method

In the previous section, we have discussed several iterative methods in optimization and their disadvantages. To overcome these shortcomings, we construct a new spectral gradient method, the DVM method, by constructing an updating formula to approximate the inverse Hessian matrix instead of the Hessian matrix using a diagonal approach. Our proposed method is anticipated to reduce computational cost and memory requirements in solving a large-scale unconstrained optimization problem.

3.1 Derivation

Sim et al. proposed to use the diagonal matrix to approximate the Hessian matrix [23]. The updating formula is shown as follows:

$$B_{k+1}^{(i)} = \frac{1}{1 + \omega(s_k^{(i)})^2}, \tag{6}$$

where i is the diagonal component of the matrix B, and the ω can be approximated by

$$\omega_k \approx \frac{s_k^T s_k - s_k^T y_k}{\sum_{i=0}^{n} (s_k^{(i)})^4}. \tag{7}$$

However, the updating formula of the MDG method needs to be inverted. Hence, the motivation of this paper is to extend a spectral gradient method that approximates the inverse Hessian matrix directly. Similar to the derivation by Sim et.al., we begin by establishing an updating formula for the approximation to the inverse Hessian matrix at step k, H_k. Consider the log-determinant norm as follows:

$$\Psi(H_k) = \operatorname{tr}(H_k) - \ln(\det(H_k)), \tag{8}$$

where $\operatorname{tr}(H_k)$ is the trace of H_k and the $\det(H_k)$ represent as the determinant of H_k.

The optimization problem can be formulated by minimizing the log-determinant norm subjected to the weak secant condition:

$$\min \Psi(H_{k+1}), \tag{9}$$

$$\text{s.t. } y_k^T H_{k+1} y_k = y_k^T s_k, \tag{10}$$

where H_{k+1} is a diagonal and positive-definite matrix which aim to approximate B_{k+1}^{-1}.

Assume that $H_{k+1} = \text{diag}(H_{k+1}^{(1)}, ..., H_{k+1}^{(n)})$ and $y_k = (y_k^{(1)}, ..., y_k^{(n)})$, the objective function (9) and constraint (10) can be rewritten as

$$\min \left(\sum_{i=1}^{n} H_{k+1}^{(i)} \right) - \ln \left(\prod_{i=1}^{n} H_{k+1}^{(i)} \right), \tag{11}$$

$$\text{s.t.} \left(\sum_{i=1}^{n} (y_k^{(i)})^2 H_{k+1}^{(i)} \right) - y_k^T s_k = 0. \tag{12}$$

Based on expressions (11) and (12), the Lagrangian is defined as

$$L(\lambda, \omega) = \omega \left[\left(\sum_{i=1}^{n} (y_k^{(i)})^2 H_{k+1}^{(i)} \right) - y_k^T s_k \right] + \left(\sum_{i=1}^{n} H_{k+1}^{(i)} \right) - \ln \left(\prod_{i=1}^{n} H_{k+1}^{(i)} \right), \tag{13}$$

where ω is the Lagrange multiplier.

Then, taking the derivative of Eqn.(13) and set to zero, one obtains

$$\frac{\partial L}{\partial H_{k+1}^{(i)}} = 1 - \frac{1}{H_{k+1}^{(i)}} + \omega \left(y_k^{(i)} \right)^2 = 0, \quad i = 1, 2, ..., n, \tag{14}$$

which yields

$$H_{k+1}^{(i)} = \frac{1}{1 + \omega \left(y_k^{(i)} \right)^2}, \quad i = 1, 2, ..., n. \tag{15}$$

By substituting the Eqn.(15) into the constraint (10), we get an expression

$$G(\omega) = \sum_{i=1}^{n} \left(\frac{\left(y_k^{(i)} \right)^2}{1 + \omega \left(y_k^{(i)} \right)^2} \right) - y_k^T s_k. \tag{16}$$

Note that $G'(\omega) < 0$ and G is monotonically decreasing since the $\omega \in [0, \infty)$. Hence, in the scenario $y_k^T y_k > y_k^T s_k$, Eqn.(16) has a unique positive solution. We could get the approximated value of ω by solving $G(\omega) = 0$. A reasonable approach approximating the solution is with a single step of Newton–Raphson iteration initiated at $\bar{\omega} = 0$. Subsequent Newton-Raphson iteration steps are unnecessary as we only require an approximate value of ω. Consequently, when $y_k^T y_k > y_k^T s_k$, the Lagrange multiplier ω_k is approximated by:

$$\omega_k \approx \bar{\omega} - \frac{G(\bar{\omega})}{G'(\bar{\omega})} = \frac{y_k^T y_k - y_k^T s_k}{\sum_{i=1}^{n} (y_k^{(i)})^4}. \tag{17}$$

We still need another updating rule for the case of $y_k^T y_k < y_k^T s_k$. We applied the Oren-Luenberger scaling [16], which is used to affect the direction of the SD for the majority of quasi-Newton methods. Therefore, the updating formula for H_{k+1} is given by the following when the two occurrences are combined:

$$H_{k+1} = \begin{cases} \text{diag}(H_{k+1}^{(i)}, ..., H_{k+1}^{(n)}), & \text{if } y_k^T y_k > y_k^T s_k \\ \frac{y_k^T s_k}{y_k^T y_k} I, & \text{otherwise,} \end{cases} \tag{18}$$

where Eqn. (15) defines $H_{k+1}^{(i)}$ and Eqn.(17) provides ω.

3.2 Algorithm

In addressing general nonlinear unconstrained optimization $\min_{x \in \mathbf{R}^n} f(x) = 0$, the iterative method is frequently used with a line search strategy. In this paper, we will apply a backtracking line search strategy. The steplength, α_k is computed to satisfy specific line search conditions, such as the Armijo condition [1]:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c\alpha_k d_k^T g_k, \tag{19}$$

where $0 < c < 1$.

Backtracking Armijo line search algorithm

Step 1: Set $c \in (0, 1)$, $\tau \in (0, 1)$, and initial steplength, $\alpha = 1$. In this paper, we use $c = 0.1$ and $\tau = 0.5$.
Step 2: Run the following relation,

$$f(x_k + \alpha d_k) \leq f(x_k) + c\alpha g_k^T d_k, \tag{20}$$

Step 3: If Eqn.(19) has been fulfilled, choose the $\alpha_k = \alpha$ and $x_{k+1} = x_k + \alpha_k d_k$. Otherwise, $\alpha_{k+1} = \tau\alpha_k$ and go to Step 2.

Next, for our proposed method, the algorithm is as follows:

Diagonal Variable Matrix Algorithm
Step 1: Set $k = 0$, given $x_0 \in \mathbf{R}^n$, $n \times n$ identity matrix as H_0, and convergence tolerance ϵ.
Step 2: Compute $g_0 = \nabla f(x_0)$ and $d_0 = -g_0$.
Step 3: If $\|g_k\| \leq \epsilon$, stop, else compute α by using a backtracking Armijo condition (19).
Step 4: Compute $x_{k+1} = x_k + \alpha d_k$.
Step 5: Calculate $g_{k+1} = \nabla f(x_{k+1})$
Step 6: Compute H_{k+1} by using the DVM method in Eqn. (18).
Step 7: Compute $d_{k+1} = -H_{k+1} g_{k+1}$.
Step 8: Set $k = k + 1$, continue from step 3.

3.3 Application

In this paper, we have implemented our proposed method in non-blinded image deblurring affected by linear motion blur. Figure 1 shows some examples of ground truth (left) and blurred images (right) that are affected by motion blur.

A drag kernel has been applied to the original images (Figure 1: left) to create the blurred images (Figure 1: right). The drag kernel, A, is given by

Figure 1. Examples of ground truth (left) and blurred image (right). Top to bottom: Lenna; Cat 1; Cat 2; Horses.

$$A = \begin{pmatrix} k_1 & \dots & k_n & 0 & 0 & 0 & 0 \\ 0 & k_1 & \dots & k_n & 0 & 0 & 0 \\ 0 & 0 & k_1 & \dots & k_n & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & k_1 & \dots & k_n \end{pmatrix}, \tag{21}$$

where $k_i = 1/n, 1 \leq i \leq n$, and n determines the "speed" of the drag. The relationship between the original image, X and blurred image, B is given by optimization problem, min $f(X)$,

$$f(X) = \frac{1}{2}\|B - AX\|_F^2. \tag{22}$$

Let x_k be a row in the original image X, and b is the corresponding blurred image B, the image deblurring algorithm is similar to the Diagonal Variable Matrix algorithm except that in Step 5, $g_{k+1} = \nabla f(x_{k+1})$ is replaced by $g_{k+1} = A^T(Ax_k - b)$.

4 Result and Discussion

In this paper, SD, MDG, and various CG methods are considered competitors to the DVM method. In the following sections, we will use some abbreviations for convenience:

1. DVM as Diagonal Variable Matrix method (Proposed method).
2. MDG as Multiple Damping Gradient method [23].
3. CG-FR as Fletcher-Reeves Conjugate Gradient method [10].
4. CG-PRP as Polak-Ribiere-Polyak Conjugate Gradient method [18].
5. CG-LS as Liu-Storey Conjugate Gradient method [15].
6. CG-DY as Dai-Yuan Conjugate Gradient method [6].
7. CG-HZ as Hager-Zhang Conjugate Gradient method [12].
8. SD as Steepest Descent method [4].

The performance of the mentioned methods is assessed using image quality metrics and the number of iterations. Less number of iterations indicates that the method converges faster. Image quality is assessed using three metrics: root mean square error (RMSE), peak signal-to-noise ratio (PSNR), and structural similarity (SSIM). RMSE measures the average difference between the restored image and the ground truth image. Lower values of RMSE indicate better performance. On the other hand, higher values of PSNR are more favorable. SSIM values range from 0 to 1, with higher values indicating better restoration.

MSE is defined as the difference between a noise-free image, F, and its noisy approximation R.

$$\mathrm{MSE} = \frac{1}{ab}\sum_{i=0}^{a-1}\sum_{j=0}^{b-1}[F(i,j) - R(i,j)]^2, \tag{23}$$

where $a \times b$ is the dimension of F and R. The RMSE is given as

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}. \tag{24}$$

The PSNR is represented as

$$\mathrm{PSNR} = 10\ \log_{10}\left(\frac{MAX_F^2}{MSE}\right), \tag{25}$$

where MAX_F is the maximum pixel value of the image.

The formula of SSIM is defined as

$$\mathrm{SSIM}(x,b) = \frac{(2\mu_x\mu_b + e_1)(2\sigma_{xb} + e_2)}{(\mu_x^2 + \mu_b^2 + e_1)(\sigma_x^2 + \sigma_b^2 + e_2)}, \tag{26}$$

where μ_x and μ_b denote as pixel sample mean of x and b, σ_x and σ_b denote as variance of x and b, σ_{xb} represent as covariance of x and b. e_1 and e_2 are two variables, used to stabilize the division with a weak denominator.

A total of four images were tested: Cat 1, Lenna, Cat 2, and Horses. Tables 3, 4, 5 and 6 show the results of the various gradient methods in restoring the images. Because computing the Hessian matrix is time-consuming and resource-intensive, the quasi-Newton method is not utilized as the benchmark for comparison.

Gradient Method	RMSE	PSNR	SSIM	No. Iteration
SD	0.070 7	22.453 4	0.944 2	150
CG-FR	**0.068 3**	22.748 8	**0.951 7**	84
CG-PRP	0.070 7	22.444 7	0.944 0	150
CG-LS	0.070 7	22.444 1	0.944 0	150
CG-DY	**0.068 3**	**22.949 4**	0.950 2	96
CG-HZ	0.070 6	22.456 8	0.944 3	150
MDG	0.069 2	22.631 8	0.951 5	150
DVM	**0.068 3**	22.745 2	**0.951 7**	**61**

Table 3. Image deblurring result of image "Cat 1"

Gradient Method	RMSE	PSNR	SSIM	No. Iteration
SD	0.104 0	17.723 7	0.826 3	150
CG-FR	**0.103 4**	17.769 1	0.823 9	101
CG-PRP	0.104 0	17.721 2	0.826 3	150
CG-LS	0.104 0	17.721 1	0.826 3	150
CG-DY	0.104 0	17.724 5	0.821 7	127
CG-HZ	0.103 9	17.725 5	0.826 3	150
MDG	0.103 6	17.753 0	**0.828 1**	150
DVM	**0.103 4**	**17.772 8**	0.824 1	**76**

Table 4. Image deblurring result of image "Lenna"

Gradient Method	RMSE	PSNR	SSIM	No. Iteration
SD	27.383 3	17.817 9	0.915 7	150
CG-FR	26.347 1	18.152 8	0.930 0	100
CG-PRP	27.382 8	17.818 0	0.915 7	150
CG-LS	27.414 3	17.808 0	0.915 3	150
CG-DY	26.365 5	18.146 9	0.931 6	122
CG-HZ	27.377 6	17.819 7	0.915 7	150
MDG	26.956 8	17.954 2	0.922 5	150
DVM	**26.3252**	**18.160 2**	**0.931 7**	**72**

Table 5. Image deblurring result of image "Cat 2"

As seen from Tables 3 – 6, all the gradient methods show comparable results in terms of image quality. It is worth mentioning that the DVM method shows the lowest number of iterations in achieving the optimal solution compared to the other methods. Hence, we can conclude that the DVM method outperforms the state-of-art methods. The images restored using the DVM method are shown in Figure 2.

Gradient Method	RMSE	PSNR	SSIM	No. Iteration
SD	10.662 5	27.505 3	0.879 8	150
CG-FR	10.025 9	28.039 9	0.831 5	150
CG-PRP	10.696 1	27.477 9	0.879 7	150
CG-LS	10.697 5	27.476 7	0.879 7	150
CG-DY	10.394 8	27.726 1	0.809 7	150
CG-HZ	10.661 5	27.506 0	0.879 8	150
MDG	10.628 0	27.533 4	**0.880 9**	150
DVM	**10.005 7**	**28.057 5**	0.832 6	**129**

Table 6. Image deblurring result of image "Horses"

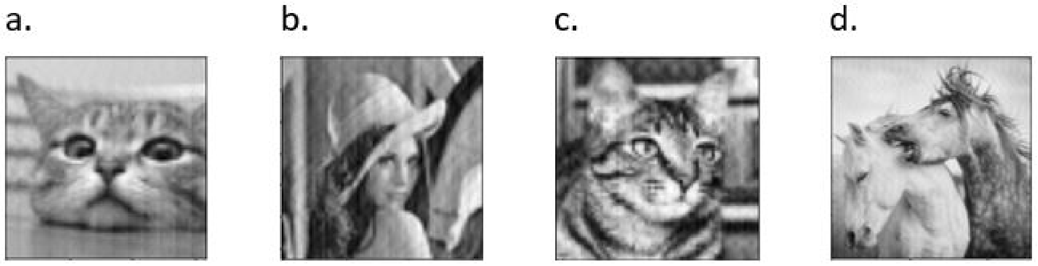

Figure 2. Restored images by DVM method: a. Cat 1 b. Lenna c. Cat 2 d. Horses

5 Conclusion

In this paper, we have introduced a gradient method named as DVM method. In the DVM method, we derived H_k as the approximation of the inverse Hessian matrix. The proposed method is coupled with the backtracking Armijo line search, aimed at reducing the number of iterations. From the theoretical perspective, this method does not require the inversion of the approximated Hessian matrix. As compared to the state-of-art methods, the DVM method shows comparable image quality results and requires the lowest number of iterations in achieving the optimal solution. Hence, it is expected that the proposed method would be capable of solving large-scale problems or recovering larger images in fewer iterations. Lastly, our proposed approach could be generalized to work with other line search methods. In the future, we may explore the application of our method in blind image deblurring and additionally incorporate noise factors into the image deblurring process. In addition to image deblurring, neural network systems, image denoising, and image segmentation are possible uses of this method.

Acknowledgement The work was supported by the Malaysia Fundamental Research Grant Scheme (FRGS/1/2022/STG06/UTAR/02/2).

References

[1] Armijo, Larry, Minimization of functions having Lipschitz continuous first partial derivatives, Pacific J. Math. **16 (1)**: 1-3 (1966).

[2] Barzilai, J., Borwein, J. , Two-Point Step Size Gradient Methods, IMA Journal of Numerical Analysis **8**, 141-148 (1988).

[3] Caron, J. N., Namazi, N. M., Rollins, C. J. Noniterative blind data restoration by use of an extracted filter function. Applied Optics **41**(32), 6884 (2002).

[4] Cauchy, A., Méthode générale pour la résolution des systemes d'équations simultanées [General method for solving systems of simultaneous equations], Comp. Rend. Sci. Paris **25**, 536-538 (1847).

[5] Conn, A. R., Gould, N. I. M., Toint, O. L., Convergence of quasi-Newton matrices generated by the symmetric rank one update. Mathematical Programming, **50**(1-3), 177-195. (1991).

[6] Dai, Y.H., Yuan, Y., A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization **10** (1): 177–182 (1999).

[7] Davidon, W.C., Variable metric method for minimization, United States: N. p., (1959).

[8] Edwin K.P. Chong, Stanislaw H. Zak, *An Introduction to Optimization*. 4th Edition, (A John Wiley & Sons, Inc., New Jersey, 2013) 81-93.

[9] Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T., Removing Camera Shake from a Single Photograph. ACM Transactions on Graphics **25** (3), 787 (2006).

[10] Fletcher, R., Reeves, C. M., Function minimization by conjugate gradients, The Computer Journal **7**(2): 149–154 (1964).

[11] Fletcher, R., Powell, M.J.D., A rapidly convergent descent method for minimization, The Computer Journal, **6**(2), 163-168. (1963).

[12] Hager, W. W., Zhang, H. A survey of nonlinear conjugate gradient methods. Pacific journal of Optimization **2**(1), 35-58 (2006).

[13] Hestenes, M. R., Stiefel, E. Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bereau of Standards **49**(6), 409-436 (1952).

[14] Jorge Nocedal, Stephen J. Wright, *Numerical Optimiation, (2nd ed)* (Springer, New York, 2006), 5-9.

[15] Liu, Y., Storey, C., Efficient generalized conjugate gradient algorithms, part 1: theory, J. Optim. Theory Appl. **69** (1), pp. 129-137 (1991).

[16] Luenberger, D.G., Linear and Nonlinear Programming, 2nd Edn., Addition-Wesley, Reading (1984).

[17] Poisel, R.A., Electronic Warfare Target Location Methods, Second Edition, Artech House (2012).

[18] Polak, E., Ribière, G., Note sur la convergence de méthodes de directions conjuguées, Revue Française d'Automatique, Informatique, Recherche Opérationnelle **3**(1), 35–43 (1969).

[19] Polyak, B., Gradient methods for minimizing functionals (in russian), Zh. Vychisl. Mat. Mat. Fiz., pp. 643-653 (1963).

[20] Rao, S.S., Engineering Optimization: Theory and Practice, John Wiley & Sons, Inc. (2009).

[21] Raydan, M., and Svaiter, B. F., Relaxed steepest descent and Cauchy-Barzilai-Borwein method., Computational Optimization and Application **21**(2), 155-167 (2002).

[22] Sim, H.S., Spectral Gradient Methods via variational technique for large scale unconstrained optimization, Doctoral dissertation, Universiti Putra Malaysia (2017).

[23] Sim, H.S., Leong, W.J., Chen, C.Y., Gradient Method with Multiple Damping for Large Scale Unconstrained Optimization, Optimization Letters **13**, 617-632 (2019).