

BREGMAN SPLIT ALGORITHM AND APPLICATION TO IMAGE RECOVERY PROBLEM

Nguyen Dinh Dung¹, Vu Khac Hung^{2*}

¹TNU - University of Information and Communication Technology, ²Thai Binh University

ARTICLE INFO	ABSTRACT
Received: 13/01/2025 Revised: 19/03/2025 Published: 21/03/2025	The Split Bregman algorithm is a variation of the Bregman algorithm, which is an optimization method applied to non-smooth inverse problems in image reconstruction and restoration, particularly in total variation problems. Traditional methods often face challenges in handling non-differentiable problems and require significant computational effort. Therefore, this study aims to develop an improved algorithm based on Split Bregman to accelerate convergence and ensure the stability of the solution. The research method employs a splitting technique to separate non-smooth components, combined with a Bregman update step to solve the optimization problem for each component independently, thereby reducing computational complexity. The research results demonstrate that the improved algorithm achieves high performance in image reconstruction from noisy data, with significantly enhanced peak signal to noise ratio values and reduced mean squared error through iterations. Experimental computations illustrate that the improved Split Bregman method not only has high applicability but also opens new directions for research in optimizing parameters and processing more complex data in the future.
KEYWORDS	
Split Bregman Algorithm	
Image reconstruction	
Total variation	
Inverse problems	
Optimization methods	

THUẬT TOÁN SPLIT BREGMAN VÀ ỨNG DỤNG CHO BÀI TOÁN KHÔI PHỤC ẢNH

Nguyễn Đình Dũng¹, Vũ Khắc Hưng^{2*}

¹Trường Đại học Công nghệ thông tin và Truyền thông – ĐH Thái Nguyên, ²Trường Đại học Thái Bình

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 13/01/2025 Ngày hoàn thiện: 19/03/2025 Ngày đăng: 21/03/2025	Thuật toán Split Bregman là một biến thể của thuật toán Bregman, đây là một thuật toán tối ưu được áp dụng cho các bài toán ngược không trơn trong tái tạo và khôi phục hình ảnh, đặc biệt là các bài toán tổng biến phân. Hiện nay, các phương pháp truyền thống thường gặp khó khăn trong việc xử lý các bài toán mà hàm mục tiêu không khả vi trên toàn miền và yêu cầu tính toán lớn, do đó, nghiên cứu này nhằm phát triển một thuật toán cải tiến dựa trên Split Bregman giúp tăng tốc độ hội tụ và đảm bảo tính ổn định của nghiệm. Phương pháp nghiên cứu sử dụng kỹ thuật phân tách để tách rời các thành phần không trơn, kết hợp với bước cập nhật Bregman để giải quyết bài toán tối ưu hóa theo từng thành phần riêng biệt, từ đó giảm độ phức tạp tính toán. Kết quả nghiên cứu cho thấy thuật toán cải tiến đạt hiệu suất cao trong việc tái tạo hình ảnh từ dữ liệu bị nhiễu, với tỷ lệ giữa tín hiệu với độ nhiễu được cải thiện và sai số trung bình bình phương giảm qua các bước lặp. Các tính toán thử nghiệm minh họa cho thấy phương pháp Split Bregman cải tiến không chỉ có tính ứng dụng cao mà còn mở ra hướng nghiên cứu trong việc tối ưu hóa các tham số và xử lý dữ liệu phức tạp hơn trong tương lai.
TỪ KHÓA	
Thuật toán Split Bregman	
Khôi phục ảnh	
Tổng biến phân	
Bài toán ngược	
Các phương pháp tối ưu	

DOI: <https://doi.org/10.34238/tnu-jst.11870>

* Corresponding author. Email: vukhachung71@gmail.com

1. Giới thiệu

Thuật toán Split Bregman là một phương pháp tối ưu được thiết kế để giải các bài toán tối ưu hóa không trơn, thường xuất hiện trong các bài toán xử lý ảnh và tái tạo tín hiệu, đây là những bài toán thuộc lớp các bài toán ngược. Thuật toán này được xây dựng dựa trên phương pháp lặp Bregman và khả năng phân tách bài toán tối ưu thành các bài toán nhỏ hơn, dễ giải hơn [1].

Một trong những bài toán thuộc lớp các bài toán ngược là bài toán chụp cắt lớp vi tính (CT) [2], bài toán quy về việc tìm nghiệm của phương trình toán từ

$$\mathbf{Ax} = \mathbf{f} \quad (1)$$

Trong đó, $\mathbf{A}$ là toán tử Radon chùm tia hình nón, mô hình hóa cách tia X di chuyển và bị hấp thụ khi đi qua vật thể, $\mathbf{f}$ là ảnh đầu ra thu được từ các phép chiếu khi tia X đi qua vật thể và $\mathbf{x}$ là hình ảnh mong muốn cần tìm. Tuy nhiên, do dữ liệu $\mathbf{f}$ thường bị nhiễu hoặc không đầy đủ (do hạn chế số lượng góc chiếu), bài toán (1) trở thành bài toán đặt không chính, việc tìm lời giải của bài toán CT có thể dẫn đến nghiệm không ổn định hoặc không duy nhất. Để tìm lời giải cho bài toán đặt không chính, ta cần phải sử dụng các thuật toán hiệu chỉnh nhằm tìm nghiệm xấp xỉ hội tụ về một nghiệm tối thiểu duy nhất [3], [4]. Trong trường hợp các bài toán khôi phục ảnh khi được mô hình hóa về bài toán (1) là bài toán đặt không chính sẽ được xấp xỉ bởi bài toán tối ưu L_1 [5]. Dạng tổng quát của bài toán hiệu chỉnh L_1 được xác định là:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + F(\mathbf{x}) \quad (2)$$

Trong đó, $F(\mathbf{x})$ là phiếm hàm lỗi, nghiệm tìm được bởi dãy lặp

$$\mathbf{x}^{(k+1)} = \operatorname{argmin}_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{1}{2\beta} \left\| \mathbf{x} - \left(\mathbf{x}^{(k)} - \beta \nabla F(\mathbf{x}^{(k)}) \right) \right\|_2^2, k = 0, 1, 2, \dots \quad (3)$$

Trong các công thức (2) và (3), ta định nghĩa các chuẩn như sau: $\|\mathbf{x}\|_1 = \sum_i |x_i|$, $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$, $\mathbf{x}^{(k)}$ là nghiệm thu được ở lần lặp thứ k và β tham số hiệu chỉnh lặp. Vì $\mathbf{x}$ có thể phân tách theo nhiều thành phần, nên ta có thể sử dụng hàm shrink hoặc ngưỡng mềm để tìm lời giải cho bài toán tối ưu (3).

$$\mathbf{x}_i^{(k+1)} = \operatorname{shrink} \left(\left(\mathbf{x}^{(k)} - \beta \nabla F(\mathbf{x}^{(k)}) \right)_i, \beta \right), i = 1, 2, \dots, n. \quad (4)$$

Hàm Shrink theo $y, \beta \in \mathbf{R}$ được định nghĩa như sau [6]:

$$\operatorname{shrink}(y, \beta) = \frac{y}{|y|} \max(|y| - \beta, 0). \quad (5)$$

Trong các bài toán xấp xỉ cho bài toán (1), việc xây dựng hàm F sao cho bài toán trở thành bài toán đặt chính là một trong các yêu cầu đặt ra đối với mỗi thuật toán hiệu chỉnh. Trong đó, hiệu chỉnh tổng biến phân là một phương pháp được nhiều tác giả quan tâm. Phương pháp này lần đầu tiên được giới thiệu bởi Rudin, Osher và Fatemi khi giải quyết bài toán khử nhiễu hình ảnh [6], [7] và được phát triển để tìm lời giải cho các bài toán ngược [8], [9]. Trong bài báo này, gradient rời rạc được định nghĩa là sai phân hữu hạn tiến nên chuẩn tổng biến phân là chuẩn L_1 của gradient rời rạc ∇ trên ảnh, tức là $\|\mathbf{x}\|_{TV} \approx \|\nabla \mathbf{x}\|_1$ [10]. Thủ tục lặp (3) có thể được biểu diễn bởi tổng biến phân là [5]:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{TV} + F(\mathbf{x}). \quad (6)$$

Trong đó, $F(\mathbf{x}) = \frac{\mu}{2} \|\mathbf{Ax} - \mathbf{f}\|_2^2$. Vậy bài toán hiệu chỉnh có thể viết là

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{TV} + \frac{\mu}{2} \|\mathbf{Ax} - \mathbf{f}\|_2^2 \quad (7)$$

Bài toán tối ưu không ràng buộc (7) cũng có thể biểu diễn tương đương như bài toán tối ưu ràng buộc [11]

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{TV} \quad \text{thỏa mãn } \mathbf{Ax} = \mathbf{f} \quad (8)$$

Vì số hạng $\|\mathbf{x}\|_{TV}$ trong (7) không khả vi cũng như không thể tách rời nên không thể sử dụng trực tiếp các thuật toán như Gradient Descent truyền thống để giải quyết bài toán. Để khắc phục những khó khăn này, chúng tôi sử dụng thuật toán Split Bregman phân tách toán tử [12] để tìm lời giải cho bài toán tối ưu lồi không trơn và cải tiến thuật toán nhằm tăng tốc độ hội tụ của nghiệm lặp về nghiệm đúng của bài toán.

Nội dung bài báo được tổ chức thành 4 phần: Phần 1 là phần giới thiệu; Phần 2 trình bày phương pháp khôi phục ảnh được đề xuất; Phần 3 thảo luận về kết quả thực nghiệm và Phần 4 là kết luận.

2. Phương pháp

Theo các kết quả trong [6], ta xét bài toán (7), phương pháp Split Bregman thay thế phiếm hàm không trơn $\|\nabla \mathbf{x}\|_1$, bởi thành phần có thể phân tách, khi đó bài toán (7) được dẫn về tìm nghiệm của bài toán:

$$\arg \min_{\mathbf{x}, \mathbf{d}} \|\mathbf{d}\|_1 + \frac{\mu}{2} \|\mathbf{Ax} - \mathbf{f}\|_2^2, \quad \mathbf{d} = \nabla \mathbf{x}. \quad (9)$$

Trong đó, ∇ là toán tử gradient. Để tìm lời giải cho bài toán (9), ta xét bài toán

$$(\mathbf{x}^{(k+1)}, \mathbf{d}^{(k+1)}) = \arg \min_{\mathbf{x}, \mathbf{d}} \|\mathbf{d}^{(k)}\|_1 + \frac{\mu}{2} \|\mathbf{Ax}^{(k)} - \mathbf{f}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d}^{(k)} - \nabla \mathbf{x}^{(k)}\|_2^2 \quad (10)$$

Bằng cách áp dụng thuật toán lặp Bregman [13], ta có quá trình lặp:

$$(\mathbf{x}^{(k+1)}, \mathbf{d}^{(k+1)}) = \arg \min_{\mathbf{x}, \mathbf{d}} \|\mathbf{d}^{(k)}\|_1 + \frac{\mu}{2} \|\mathbf{Ax}^{(k)} - \mathbf{f}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d}^{(k)} - \nabla \mathbf{x}^{(k)} - \mathbf{b}^{(k)}\|_2^2 \quad (11)$$

$$\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + (\nabla \mathbf{x}^{(k+1)} - \mathbf{d}^{(k+1)}). \quad (12)$$

Trong đó, $\mathbf{b}$ là biến Bregman và λ là hằng số dương, $\|\mathbf{d}^{(k)}\|_1$ là thành phần hiệu chỉnh thường được sử dụng để thúc đẩy nghiệm thưa (sparse solution) hoặc trong tổng biến phân (TV), đây là đại lượng không trơn, theo kỹ thuật Split Bregman ta tách biến $\mathbf{d}$ ra khỏi biến $\mathbf{x}$, giúp bài toán tối ưu hóa dễ giải hơn bằng cách giải luân phiên các bài toán con. Thành phần $\frac{\mu}{2} \|\mathbf{Ax}^{(k)} - \mathbf{f}\|_2^2$ là hàm

khớp với dữ liệu đo lường $\mathbf{f}$, $\frac{\lambda}{2} \|\mathbf{d}^{(k)} - \nabla \mathbf{x}^{(k)} - \mathbf{b}^{(k)}\|_2^2$ là thành phần Bregman để điều chỉnh và đảm

bảo sự hội tụ. Xét thủ tục lặp (11), khi $\mathbf{L}_1$ và $\mathbf{L}_2$ là phân tách được, ta tìm nghiệm (11) là $\mathbf{x}$ và $\mathbf{d}$. Theo các kết quả trong [6], việc tìm lời giải được thực hiện qua hai bước như sau:

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} \frac{\mu}{2} \|\mathbf{Ax}^{(k)} - \mathbf{f}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d}^{(k)} - \nabla \mathbf{x}^{(k)} - \mathbf{b}^{(k)}\|_2^2 \quad (13)$$

$$\mathbf{d}^{(k+1)} = \arg \min_{\mathbf{x}, \mathbf{d}} \|\mathbf{d}^{(k)}\|_1 + \frac{\lambda}{2} \|\mathbf{d}^{(k)} - \nabla \mathbf{x}^{(k+1)} - \mathbf{b}^{(k)}\|_2^2 \quad (14)$$

Từ đó dễ thấy, hàm mục tiêu trong (13) là khả vi, vì vậy ta hoàn toàn có thể giải được bằng kỹ thuật tron, như phương pháp gradient liên hợp, Gauss-Seidel,... Trong bài báo này, chúng tôi áp dụng phương pháp gradient descent như sau:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \left(\mu \mathbf{A}^T (\mathbf{A} \mathbf{x}^{(k)} - \mathbf{f}) + \lambda \nabla (\nabla \mathbf{x}^{(k)} + \mathbf{b}^{(k)} - \mathbf{d}^{(k)}) \right) \quad (15)$$

Trong công thức (14), ta có thể tính toán giá trị tối ưu $\mathbf{d}$ bằng cách sử dụng toán tử shrink theo (5). Như vậy các bước tìm lời giải xấp xỉ cho bài toán (1) có thể dẫn về thuật toán sau:

Thuật toán 1: Split Bregman Algorithm

Input: $\mathbf{A}, \mathbf{f}, \alpha, \lambda, \varepsilon$

```

1:   Initialize:  $\mathbf{x}, \mathbf{b}, \mathbf{d}$ ,  $k=1$ 
2:   while  $\|\mathbf{A}\mathbf{x} - \mathbf{f}\|_2^2 > \varepsilon^2$  do
3:        $\mathbf{x} = \mathbf{x} - \alpha \left( \mu \mathbf{A}^T (\mathbf{A}\mathbf{x} - \mathbf{f}) + \lambda \nabla (\nabla \mathbf{x} + \mathbf{b} - \mathbf{d}) \right)$ 
5:        $\mathbf{d} = \text{shrink} \left( \nabla \mathbf{x} + \mathbf{b}, \frac{1}{\lambda} \right)$ 
7:        $\mathbf{b} = \mathbf{b} + (\nabla \mathbf{x} - \mathbf{d})$ 
9:        $k++$ ;
10:  End
```

Output: $\mathbf{x}$

Để cải tiến thuật toán lặp này nhằm đạt được hội tụ tốt hơn, chúng tôi cải tiến (15) bằng cách thêm một bước cập nhật trước khi lặp lại để tính nghiệm ở bước tiếp theo, cụ thể như sau:

$$\mathbf{y}^{(k)} = \mathbf{x}^{(k)} + \beta_k (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}) \quad (16)$$

β_k gọi là hệ số momentum thích nghi theo số bước lặp và trong trường hợp này, chúng tôi chọn

$\beta_k = \frac{k}{k+C}$, trong đó C là một hằng số dương, sau đó nghiệm của bài toán (13) sẽ được cập nhật theo công thức:

$$\mathbf{x}^{(k+1)} = \mathbf{y}^{(k)} - \alpha \nabla F(\mathbf{y}^{(k)}) \quad (17)$$

$$\nabla F(\mathbf{y}^{(k)}) = \mu \mathbf{A}^T (\mathbf{A} \mathbf{y}^{(k)} - \mathbf{f}) + \lambda \nabla (\nabla \mathbf{y}^{(k)} + \mathbf{b}^{(k)} - \mathbf{d}^{(k)}) \quad (18)$$

Để khẳng định dãy lặp (17) hội tụ về nghiệm đúng của bài toán (1), ta lần lượt xét các định lý sau:

Định lý 1: $F(\mathbf{y}) = \frac{\mu}{2} \|\mathbf{A}\mathbf{y} - \mathbf{f}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d} - \nabla \mathbf{y} - \mathbf{b}\|_2^2$ là phiếm hàm lồi mạnh và liên tục Lipschitz

với hằng số Lipschitz $L = \mu \|\mathbf{A}^T \mathbf{A}\|_2^2 + \lambda \|\nabla^T \nabla\|_2^2$.

Chứng minh

Đặt $F_1(\mathbf{y}) = \frac{\mu}{2} \|\mathbf{A}\mathbf{y} - \mathbf{f}\|_2^2$. Vì $\mathbf{A}$ là toán tử tuyến tính, nên $F_1(\mathbf{y})$ là hàm lồi mạnh với

$\nabla^2 F_1(\mathbf{y}) = \mu \mathbf{A}^T \mathbf{A} \geq \mu \sigma_{\min} \mathbf{I}$, $\sigma_{\min} > 0$ là giá trị riêng nhỏ nhất của $\mathbf{A}^T \mathbf{A}$. Đặt

$F_2(\mathbf{y}) = \frac{\lambda}{2} \|\mathbf{d} - \nabla \mathbf{y} - \mathbf{b}\|_2^2$, ta có $\nabla F_2(\mathbf{y}) = \lambda \nabla^T \nabla$ xác định dương với giá trị riêng nhỏ nhất là

$\lambda \rho_{\min} > 0$. Do đó, $F(\mathbf{y}) = F_1(\mathbf{y}) + F_2(\mathbf{y}) = \frac{\mu}{2} \|\mathbf{A}\mathbf{y}^{(k)} - \mathbf{f}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d} - \nabla \mathbf{y} - \mathbf{b}\|_2^2$ là hàm lồi mạnh với hằng số lồi mạnh là $\mu_F = \min(\mu \sigma_{\min}, \lambda \rho_{\min})$.

$\nabla F_1(\mathbf{y}) = \mu \mathbf{A}^T (\mathbf{A}\mathbf{y} - \mathbf{f})$, vì $\mathbf{A}^T \mathbf{A}$ xác định và liên tục Lipschitz, nên $F_1(\mathbf{y})$ liên tục Lipschitz với hằng số Lipschitz là $\mu \|\mathbf{A}^T \mathbf{A}\|_2^2$. $\nabla F_2(\mathbf{y}) = \lambda \nabla^T (\nabla \mathbf{y} + \mathbf{b} - \mathbf{d})$, toán tử Laplacian $\nabla^T \nabla$ xác định và liên tục với hằng số Lipschitz $\|\nabla^T \nabla\|_2^2$, nên $F_2(\mathbf{y})$ liên tục Lipschitz với hằng số Lipschitz $\lambda \|\nabla^T \nabla\|_2^2$.

Vậy $F(\mathbf{y}) = F_1(\mathbf{y}) + F_2(\mathbf{y})$ liên tục với hằng số Lipschitz $L = \mu \|\mathbf{A}^T \mathbf{A}\|_2^2 + \lambda \|\nabla^T \nabla\|_2^2$.

Định lý 2: Cho $L = \mu \|\mathbf{A}^T \mathbf{A}\|_2^2 + \lambda \|\nabla^T \nabla\|_2^2$ Nếu $0 < \alpha < \frac{2}{L}$ thì $\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2^2 = 0$, trong đó $\mathbf{x}^*$ là nghiệm của (1).

Chứng minh

Vì $F(\mathbf{x})$ lồi mạnh, nên $F(\mathbf{x}^{(k)}) \geq F(\mathbf{x}^*) + \nabla F(\mathbf{x}^*)^T (\mathbf{x}^{(k)} - \mathbf{x}^*) + \frac{m}{2} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2^2$, với m là tham số lồi mạnh. Mặt khác, $\nabla F(\mathbf{x}^*) = \mathbf{0}$, suy ra

$$F(\mathbf{x}^{(k)}) - F(\mathbf{x}^*) \geq \frac{m}{2} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2^2. \quad (19)$$

Theo định lý 1, $F(\mathbf{x})$ liên tục Lipschitz nên

$$\|\nabla F(\mathbf{x}^{(k+1)}) - \nabla F(\mathbf{y}^{(k)})\|_2^2 \leq L \|\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}\|_2^2. \quad (20)$$

Mặt khác, ta có

$$F(\mathbf{x}^{(k+1)}) = F(\mathbf{y}^{(k)}) + \int_0^1 \nabla F(\mathbf{y}^{(k)} + t(\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}))^T (\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}) dt$$

hay có thể viết

$$\begin{aligned} F(\mathbf{x}^{(k+1)}) &= F(\mathbf{y}^{(k)}) + \nabla F(\mathbf{y}^{(k)})^T (\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}) \\ &+ \int_0^1 \left(\nabla F(\mathbf{y}^{(k)} + t(\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)})) - \nabla F(\mathbf{y}^{(k)}) \right)^T (\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}) dt. \end{aligned} \quad (21)$$

Sử dụng tính chất Lipschitz, ta có

$$\|\nabla F(\mathbf{y}^{(k)} + t(\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)})) - \nabla F(\mathbf{y}^{(k)})\|_2^2 \leq tL \|\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}\|_2^2. \quad (22)$$

Từ (21), (22), ta có

$$\begin{aligned} F(\mathbf{x}^{(k+1)}) &\leq F(\mathbf{y}^{(k)}) + \nabla F(\mathbf{y}^{(k)})^T (\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}) + \int_0^1 L \|\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}\|_2^2 t dt \\ F(\mathbf{x}^{(k+1)}) &\leq F(\mathbf{y}^{(k)}) + \nabla F(\mathbf{y}^{(k)})^T (\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}) + \frac{L}{2} \|\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}\|_2^2 \end{aligned} \quad (23)$$

Từ (17), (23), ta có

$$F(\mathbf{x}^{(k+1)}) \leq F(\mathbf{y}^{(k)}) - \alpha \|\nabla F(\mathbf{y}^{(k)})\|_2^2 + \frac{L\alpha^2}{2} \|\nabla F(\mathbf{y}^{(k)})\|_2^2 \quad (24)$$

Để dãy lặp giảm thì $\frac{L\alpha^2}{2} - \alpha < 0$, tức là $0 < \alpha < \frac{2}{L}$, với cách chọn này thì

$$F(\mathbf{x}^{(k+1)}) < F(\mathbf{y}^{(k)}). \quad (25)$$

Vì $\mathbf{y}^{(k)} = \mathbf{x}^{(k)} + \beta_k(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$ và F là phiếm hàm lồi, nên ta có

$$F(\mathbf{y}^{(k)}) < F(\mathbf{x}^{(k)}) \quad (26)$$

Từ (25), (26), ta có $F(\mathbf{x}^{(k+1)}) < F(\mathbf{x}^{(k)})$, kết hợp với (19), ta có

$$\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2^2 \leq \frac{2}{m} (F(\mathbf{x}^{(k)}) - F(\mathbf{x}^*)). \quad (27)$$

Vậy $\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| = 0$.

Trên cơ sở Định lý 1 và Định lý 2, thuật toán 1 được cải tiến như sau:

Thuật toán 2: Modified Split Bregman Algorithm

Input: $\mathbf{A}, \mathbf{f}, \alpha, \lambda, \varepsilon, \mathbf{C}$

1: Initialize: $\mathbf{x}, \mathbf{b}, \mathbf{d}, k=1$

2: $\mathbf{z} = \mathbf{x} - \alpha (\mu \mathbf{A}^T (\mathbf{Ax} - \mathbf{f}) + \lambda \nabla (\nabla \mathbf{x} + \mathbf{b} - \mathbf{d}))$

3: while $\|\mathbf{Ax} - \mathbf{f}\|_2^2 > \varepsilon^2$ do

4: $\beta_k = \frac{k}{k + \mathbf{C}}$

5: $\mathbf{y} = \mathbf{z} + \beta_k(\mathbf{z} - \mathbf{x})$

6: $\mathbf{z} = \mathbf{x}$

7: $\mathbf{x} = \mathbf{y} - \alpha (\mu \mathbf{A}^T (\mathbf{Ay} - \mathbf{f}) + \lambda \nabla (\nabla \mathbf{y} + \mathbf{b} - \mathbf{d}))$

8: $\mathbf{d} = \text{shrink} \left(\nabla \mathbf{x} + \mathbf{b}, \frac{1}{\lambda} \right)$

9: $\mathbf{b} = \mathbf{b} + (\nabla \mathbf{x} - \mathbf{d})$

10: $k++;$

11: End

Output: $\mathbf{x}$

Trong mục 3, chúng tôi trình bày một số kết quả tính toán cho bài toán khôi phục ảnh sử dụng thuật toán 2, minh họa cho lý thuyết được trình bày.

3. Kết quả thực nghiệm

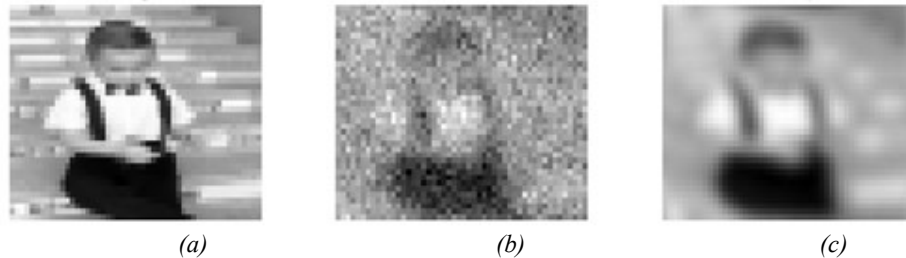
Thực hiện thuật toán 2, chúng tôi giải bài toán khôi phục ảnh với dữ liệu đầu vào là ảnh rõ nét, được sử dụng làm tiêu chuẩn so sánh, ảnh đầu vào có kích thước là 2048 x 1511 pixel và co nhỏ lại với kích thước 50 x 38 pixel. Ảnh bị làm mờ thông qua convolution với nhân Gaussian và thêm nhiễu Gaussian, phản ánh sự suy giảm chất lượng phổ biến trong thực tế. Chúng tôi kết hợp giữa kernel Gaussian (làm mờ) và nhiễu Gaussian tạo ra một ảnh bị suy giảm chất lượng, giúp mô phỏng chính xác các tình huống thực tế. Nhiễu làm giảm độ sắc nét, trong khi Gaussian blur làm giảm chi

tiết cục bộ trong ảnh. Các thông số về nhiễu được cho như sau:

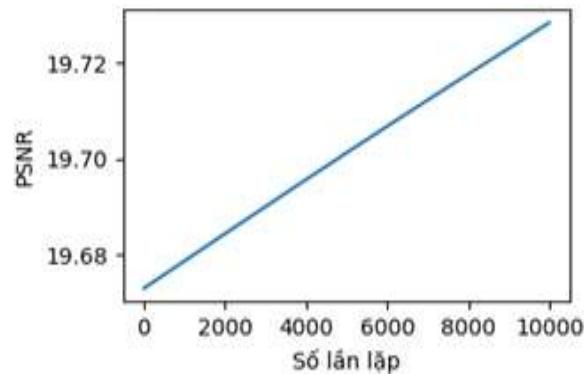
Đối với nhân làm mờ: Kích thước nhân là 7; độ lệch chuẩn (sigma): 1,5; nhân Kernel được sử dụng là một ma trận Gaussian, được chuẩn hóa để tổng các phần tử trong nhân bằng 1 và mô phỏng hiệu ứng làm mờ ảnh thực tế.

Đối với nhiễu Gaussian: Độ lệch chuẩn của nhiễu: 0,000005; nhiễu Gaussian được thêm trực tiếp vào ảnh đã làm mờ. Nhiễu có giá trị trung bình là 0 và độ lệch chuẩn nhỏ, nhằm mô phỏng các điều kiện nhiễu nhẹ thường gặp trong thực tế như cảm biến ảnh hoặc nhiễu truyền dẫn. Những thông số này được lựa chọn nhằm kiểm tra tính hiệu quả của thuật toán trong việc khôi phục ảnh từ một phiên bản bị làm mờ và nhiễu. Điều này giúp đảm bảo rằng thuật toán có thể xử lý tốt các tình huống suy giảm chất lượng phổ biến trong thực tế.

Để đánh giá tính đúng đắn của thuật toán, chúng tôi lựa chọn các thang đo là độ đo MSE (Mean Squared Error) và PSNR (Peak Signal-to-Noise Ratio). Kết quả khôi phục từ thuật toán 2. Chúng tôi thu được kết quả được thể hiện trong Hình 1 và Biểu đồ PSNR trong Hình 2 cho thấy giá trị PSNR tăng dần qua các lần lặp.



Hình 1. Kết quả khôi phục ảnh sau khi bị làm mờ và nhiễu:
(a) ảnh gốc, (b) ảnh nhiễu hoặc mờ, (c) ảnh khôi phục



Hình 2. Đồ thị PSNR qua các lần lặp

Giá trị PSNR cuối cùng đạt xấp xỉ **19,73** sau 10000 lần lặp, phản ánh một mức độ cải thiện chất lượng ảnh so với ảnh bị nhiễu/làm mờ ban đầu. Sai số bình phương trung bình (MSE) giảm dần qua các lần lặp, giá trị PSNR tăng từ **19,67** (lần lặp đầu tiên) lên **19,73** (lần lặp cuối cùng). Các kết quả cụ thể được minh họa trong Bảng 1.

Ảnh khôi phục có xu hướng rõ nét hơn so với ảnh nhiễu ban đầu, tuy nhiên vẫn tồn tại một số hiện tượng nhiễu và không đạt được độ sắc nét như ảnh gốc. Điều này là do số lần lặp chưa đủ lớn, hoặc các giá trị các tham số (α, λ, μ) , hoặc sự phụ thuộc của thuật toán vào độ chính xác của gradient và toán tử A, điều này mở ra một hướng tiếp theo cho sự phát triển của thuật toán về việc xác định tham số tối ưu nhằm nâng cao hơn chất lượng ảnh khôi phục.

Bảng 2 chỉ ra một số kết quả tính toán để thể hiện tính hiệu quả về thời gian tính toán của thuật toán 2 tốt hơn so với thuật toán 1. Chúng tôi chọn kích thước nhân là 21; độ lệch chuẩn (sigma) là

5; độ lệch chuẩn của nhiễu: 0,3. Sau 500 lần lặp, ta được kết quả như sau: Trong bảng 2, cột MSE1, MSE2 tương ứng là độ đo MSE của thuật toán 1 và thuật toán 2.

Bảng 1. Giá trị PSNR tăng dần qua các lần lặp khi thực hiện thuật toán 2

Lần lặp	MSE	PSNR	Lần lặp	MSE	PSNR
1	0,010782	19,67	5001	0,010712	19,70
501	0,010775	19,68	5501	0,010705	19,70
1001	0,010768	19,68	6001	0,010699	19,71
1501	0,010761	19,68	6501	0,010692	19,71
2001	0,010754	19,68	7001	0,010685	19,71
2501	0,010747	19,69	7501	0,010679	19,71
3001	0,010740	19,69	8001	0,010672	19,72
3501	0,010733	19,69	8501	0,010665	19,72
4001	0,010726	19,70	9001	0,010659	19,72
4501	0,010719	19,70	9501	0,010652	19,73

Bảng 2. So sánh thuật toán 1 và thuật toán 2

Lần lặp	MSE1	MSE2
1	0,029346	0,029346
51	0,029301	0,029300
101	0,029257	0,029255
151	0,029215	0,029214
201	0,029177	0,029176
251	0,029144	0,029143
301	0,029116	0,029115
351	0,029095	0,029094
401	0,029083	0,029082
451	0,029078	0,029077

4. Kết luận

Bài báo đã đề xuất thuật toán cải tiến Split Bregman nhằm tìm nghiệm cho bài toán đặt không chỉnh với toán tử tuyến tính giới nội và chứng minh được nghiệm xấp xỉ hội tụ và ổn định khi đặt ra các giới hạn về tham số hiệu chỉnh. Các kết quả thực nghiệm trên bài toán khôi phục ảnh từ ảnh nhiễu và bị mờ cho thấy giá trị PSNR tăng dần và sai số MSE giảm qua các lần lặp đã cho thấy tính hiệu quả của thuật toán trong việc tái tạo hình ảnh từ dữ liệu suy giảm chất lượng. Tuy nhiên, vẫn tồn tại những hạn chế như hiện tượng nhiễu và sự phụ thuộc vào tham số thuật toán, độ chính xác của thuật toán 2 chưa vượt trội so với thuật toán 1, điều này gợi mở hướng nghiên cứu tiếp theo nhằm tối ưu hóa các tham số và cải thiện chất lượng ảnh khôi phục. Kết quả nghiên cứu này sẽ góp phần vào việc ứng dụng các phương pháp hiệu chỉnh hiện đại trong các bài toán xử lý tín hiệu và xử lý ảnh.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] J.-F. Cai, S. Osher, and Z. Shen, "Convergence of the linearized Bregman iteration for ℓ_1 -norm minimization," *Mathematics of Computation*, vol. 78, no. 268, pp. 2127-2136, 2009.
- [2] T. Würfl, F. C. Ghesu, V. Christlein, and A. Maier, "Regularization of inverse problems in X-ray computed tomography with neural networks learned from imperfect data," *Medical Image Analysis*, vol. 54, pp. 68-82, 2019.
- [3] M. Benning and M. Burger, "Modern regularization methods for inverse problems," *Acta Numerica*, vol. 27, pp. 1-111, 2018.
- [4] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," *Theoretical Foundations and Numerical Methods for Sparse Recovery*, vol. 9, pp. 263-340, 2010.
- [5] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for L_1 -minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, pp. 143-168, 2008.

-
- [6] T. Goldstein and S. Osher, "The split Bregman method for L_1 -regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.
 - [7] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
 - [8] Y. Wang, X. Liu, and Z. Li, "A modified non-convex Cauchy total variation regularization model for image restoration," *Computational and Applied Mathematics*, vol. 43, no. 5, pp. 1–20, 2024.
 - [9] J. Zhang, L. Chen, and H. Sun, "Total variation image reconstruction algorithm based on non-convex regularization," *Signal, Image and Video Processing*, vol. 18, no. 2, pp. 263–275, 2024.
 - [10] G. Pascal, "Rudin-Osher-Fatemi total variation denoising using split Bregman," *Image Processing On Line*, vol. 2, pp. 74–95, 2012.
 - [11] C. Chen and G. Xu, "A new linearized split Bregman iterative algorithm for image reconstruction in sparse-view X-ray computed tomography," *Computers & Mathematics with Applications*, vol. 71, no. 8, pp. 1537–1559, 2016.
 - [12] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
 - [13] S. Hurault, U. Kamilov, A. Leclaire, and N. Papadakis, "Convergent Bregman Plug-and-Play image restoration for Poisson inverse problems," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 123–136, 2021.