Đề xuất kỹ thuật thực thi nhanh bộ lọc trung bình không gian 2-chiều

A New Technique for Reducing the Computational Complexity for 2D Mean Filters

Nguyễn Hữu Tài

Trường Đại học Khoa học Huế, 77 Nguyễn Huệ, Tp. Huế, Việt Nam

Đến Tòa soạn: 18-7-2018; chấp nhận đăng: 18-01-2019

Tóm tắt

Bộ lọc trung bình không gian là một trong số các bộ lọc được sự dụng phổ biến trong lĩnh vực xử lý tín hiện số nói chung và xử lý ảnh nói riêng. Vì thế việc nghiên cứu và cải tiến kỹ thuật thực hiện bộ lọc sẽ mang lại ảnh hưởng tích cực xét trên cả khía cạnh phần cứng (Hardware) lẫn phần mềm (Software). Những nghiên cứu của chúng tôi trong bài báo này đã chỉ ra một giải pháp thực hiện mới, giúp cải thiện rất lớn về độ phức tạp tính toán, từ đó rút ngắn thời gian thực hiện một cách vượt trội khi so sánh với kỹ thuật gốc trong các kết quả thực nghiệm dưới dạng phần mềm. Đặc biệt, các đề xuất này có thể mở rộng một cách tự nhiên lên không gian n-chiều.

Từ khóa: Bộ lọc trung bình, Lọc nhiễu, Lọc làm mờ ảnh, Xử lý ảnh, Xử lý tín hiệu số

Abstract

Mean filters are among the most commonly used filters in the field of digital signal processing in general and image processing in particular. Therefore, the research and improvement of the filter technology will have a positive impact on both the hardware and the software. Our studies in this paper have shown a new implementation solution, which greatly improves computational complexity, thus reducing the time taken for implementation to be superior to that of the original technique. in experimental results in the form of software. In particular, these suggestions can naturally expand into the n-dimensional space.

Keywords: 2D mean filter, Noise filter, Blur filter, Digital image processing, Digital signal processing

1. Mở đầu

Bộ lọc trung bình trong không gian 2-chiều, hay còn được gọi với thuật ngữ là "2D Mean Filter", là một bộ lọc được sử dụng phổ biến trong lĩnh vực xử lý ảnh để thực hiện các tác vụ làm trơn ảnh (smoothing), làm mờ (blur), tăng cường các chi tiết (sharpen details), hay khử nhiễu (remove noise) [2-5] và nhiều ứng dụng khác trong lĩnh vực xử lý tín hiệu số nói chung. Do đó, việc nghiên cứu nhằm cải tiến kỹ thuật thực thi của bộ lọc luôn được quan tâm [6-8], bởi kết quả sẽ góp phần đơn giản hóa kiến trúc mạch xử lý tín hiệu số (DSP) đối với giải pháp phần cứng (Hardware), hay cải thiện thời gian thực hiện lọc cho các giải pháp phần mềm (Software). Từ đó, bài báo này tập trung nghiên cứu các kỹ thuật thực hiện bộ lọc trung bình đã có, và đề xuất cải tiến nhằm mang lại một kết quả khả dĩ tốt hơn các kỹ thuật hiện tại.

2. Các kiến thức liên quan

2.1. Nhân chập 2-chiều và bộ lọc chia tách được [1]

Trong lĩnh vực xử lý tín hiệu số, một ảnh số X có kích thước M×N (M cột và N dòng) được xem là một tín hiệu 2 chiều $x(n_1, n_2)$, trong đó:

$$x(n_1,n_2) = \begin{cases} X(n_1,n_2) & \text{với } \begin{matrix} 0 \le n_1 \le M \\ 0 \le n_2 \le N \end{matrix} \\ 0 & \text{nếu ngược lại} \end{cases}$$

và bộ lọc số 2-chiều H được xác định qua các giá trị đáp ứng xung $h(n_1, n_2)$ của nó. Bộ lọc 2-chiều H được gọi là có đáp ứng xung hữu hạn khi và chỉ khi:

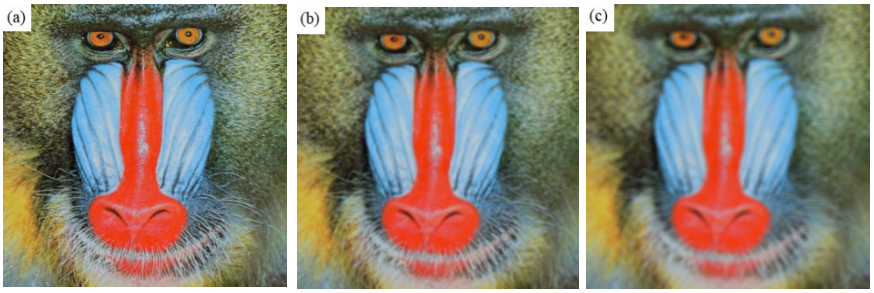
$$\begin{cases} h(n_1,n_2) \ge 0 \text{ với } 0 \le n_1 \le m \text{ và } 0 \le n_2 \le n \\ h(n_1,n_2) = 0 \text{ nếu ngược lại} \end{cases}$$

m×n được gọi là kích thước bộ lọc. Khi m = n ta có bộ lọc hình vuông.

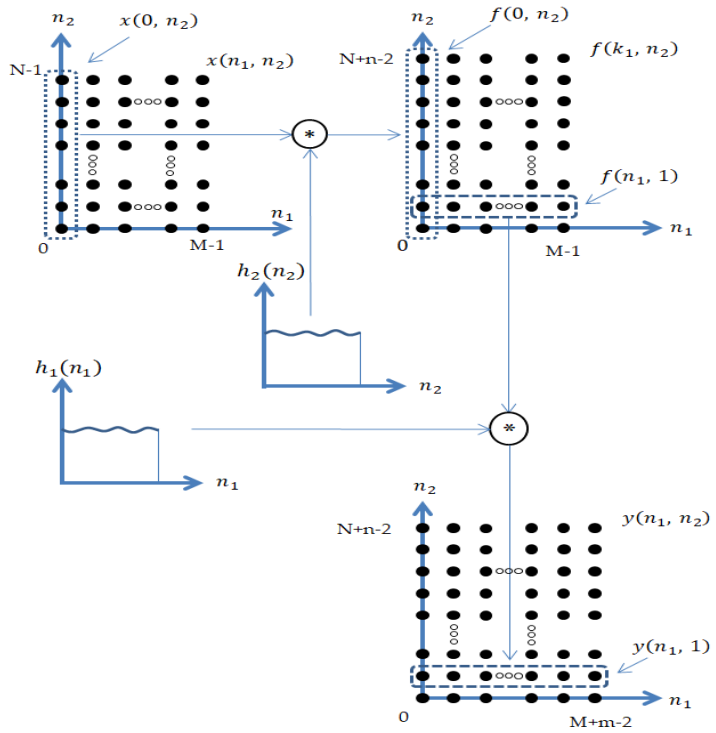
Lọc ảnh đầu vào X bởi bộ lọc H được thực hiện qua thao tác nhân cuộn (convolution) 2-chiều, hay còn được gọi là nhân chập, cho bởi công thức sau:

$$y(n_1,n_2) = x(n_1,n_2) * h(n_1,n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1,k_2) h(n_1-k_1, n_2-k_2) \quad (1)$$

* Địa chỉ liên hệ: Tel.: (+84) 905.103.928
Email: nhtai2004@gmail.com



Hình. 1. Lọc trung bình trong không gian 2-chiều. (a) Ảnh gốc mandril_color kích thước 512x512, (b) Kết quả lọc với kích thước bộ lọc 5×5, (c) Kết quả lọc với kích thước bộ lọc 9×9.



Hình. 2. Nhân cuộn của $x(n_1, n_2)$ với đáp ứng xung 2-chiều $h(n_1, n_2)$ chia tách được.

Kết quả đầu ra chúng ta thu được ảnh Y được biểu diễn qua tín hiệu 2-chiều $y(n_1, n_2)$ có giá trị xác định là:

$$\begin{cases} y(n_1, n_2) \geq 0 & với \quad 0 \leq n_1 \leq M + m - 1 \\ & \quad\quad 0 \leq n_2 \leq N + n - 1 \\ y(n_1, n_2) = 0 & \text{nếu ngược lại} \end{cases}$$

Từ đó, kích thước ảnh đầu ra Y sẽ là: *(M+m-1)×(N+n-1)*, và cần thực hiện *(M+m-1)×(N+n-1)×(m×n)* phép xử lý toán học, với mỗi phép xử lý toán học ở đây gồm 1 phép nhân và một phép cộng.

Khi kích thước ảnh là rất lớn so với kích thước bộ lọc, *hay M,N≫m,n*, thì số phép xử lý toán học cần xử lý là xấp xỉ:

$$(N \times M) \times (n \times m) \qquad (2)$$

Trong trường hợp $h(n_1, n_2)$ là chuỗi chia tách được (separable sequence), nó có thể được biểu diễn dưới dạng:

$$h(n_1, n_2) = h_1(n_1) h_2(n_2) \qquad (3)$$

Trong đó: $h_1(n_1) = 0$ với n_1 ngoài $[0, m]$

$h_2(n_2) = 0$ với n_2 ngoài $[0, n]$

Từ (2) và (3) ta có:

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$$

$$= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_1(n_1 - k_1) h_2(n_2 - k_2) \qquad (4)$$

$$= \sum_{k_1=-\infty}^{\infty} h_1(n_1 - k_1) \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_2(n_2 - k_2)$$

Với mỗi giá trị k_1 cố định,

$$\sum_{k_2=-\infty}^{\infty} x(k_1,k_2)h_2(n_2-k_2)$$

trong (4) là một phép nhân cuộn 1-chiều (1-D convolution) của $x(k_1,n_2)$ và $h_2(n_2)$. Nếu chúng ta đặt:

$$f(k_1,n_2)=\sum_{k_2=-\infty}^{\infty} x(k_1,k_2)h_2(n_2-k_2) \qquad (5)$$

thì $f(0,n_2)$ là kết quả nhân cuộn 1-chiều giữa $x(0,n_2)$ và $h_2(n_2)$ như minh họa trong Hình 2. Ta thấy, N giá trị ứng với cột k_1 của $x(n_1,n_2)$ sẽ nhân cuộn 1-chiều với n giá trị của $h_2(n_2)$, thao tác này cần thực hiện M lần ứng với M giá trị khác nhau của k_1. Vì vậy, chúng ta cần xấp xỉ *(N+n-1)×n×M* phép xử lý toán học.

Từ (4) và (5) ta có:

$$y(n_1,n_2)=\sum_{k_1=-\infty}^{\infty} h_1(n_1-k_1)\,f(k_1,n_2) \qquad (6)$$

Với mỗi giá trị n_2 cố định, $y(n_1,n_2)$ có được bằng cách nhân cuộn 1-chiều $h_1(n_1)$ với $f(n_1,n_2)$. Ví dụ, $y(n_1,1)$ có được bằng cách nhân cuộn 1-chiều $h_1(n_1)$ với $f(n_1,1)$ như trong Hình 2. Ở đây $f(n_1,n_2)$ chính là $f(k_1,n_2)$ nhưng được đổi tên biến. Thao tác nhân cuộn 1-chiều $h_1(n_1)$ với 1 hàng của $f(n_1,n_2)$ cần được thực hiện (N+n-1) lần với (N+n-1) hàng khác nhau, vì vậy ở gian đoạn này chúng ta sẽ cần (M+m-1)×m×(N+n-1) phép xử lý toán học.

Như vậy, khi đáp ứng xung 2-chiều $h(n_1,n_2)$ là chia tách được thành tích của 2 đáp ứng xung 1-chiều là $h_1(n_1)$ và $h_2(n_2)$ thì độ phức tạp tính toán của thao tác lọc thông qua 2 phép nhân cuộn 1-chiều sẽ là:

$$[(N+n-1)\times n\times M] \\ +[(M+m-1)\times m\times(N+n-1)] \qquad (7)$$

Khi kích thước ảnh là rất lớn so với kích thước bộ lọc, *hay M,N≫m,n*, thì số phép xử lý toán học cần thực hiện là xấp xỉ:

$$N\times M\times(n+m) \qquad (8)$$

2.2. Bộ lọc trung bình 2-chiều

Một bộ lọc trung bình 2-chiều kích thước m×n được xác định bởi các giá trị đáp ứng xung cho bởi công thức sau:

$$h(n_1,n_2)=\frac{1}{m\times n} \text{ với } 0\le n_1\le m, 0\le n_2\le n \\ h(n_1,n_2)=0 \text{ ngược lại} \qquad (9)$$

Hình 3(a) minh họa cho chúng ta các giá trị của đáp ứng xung $h(n_1,n_2)$ có kích thước 3×3 được biểu diễn dưới dạng ma trận (hay còn gọi là mặt nạ lọc), và dạng biểu diễn tương đương của nó với hệ số nhân 1/9 (còn được gọi là Gain của bộ lọc) và các giá trị bên trong đều bằng 1. Dạng chuyển đổi về số nguyên này mang lại khả năng thực hiện bộ lọc trên các hệ thống xử lý số nguyên.

$H_{3\times3}=$ | 1/9 | 1/9 | 1/9 |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

$=\frac{1}{9}$ | 1 | 1 | 1 |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |

(a)

$H_{5\times5}=\frac{1}{25}$ | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

(b)

Hình. 3. Một số mặt nạ lọc của bộ lọc trung bình 2D.

Vì toàn bộ các hệ số của mặt nạ lọc trung bình đều được chuyển đổi về giá trị 1, nên trong thao tác nhân cuộn để thực hiện lọc sẽ không cần thực hiện phép nhân $x(k_1,k_2)h(n_1-k_1,n_2-k_2)$. Do đó, số phép toán để thực hiện bộ lọc trung bình 2-chiều trên một tín hiệu ảnh đầu vào theo công thức (2) sẽ cho giá trị xấp xỉ:

$(N\times M)\times(n\times m)$ phép toán cộng và $(N\times M)$ phép toán chia (10)

3. Đề xuất kỹ thuật chia tách

Rõ ràng, bộ lọc trung bình 2-chiều là chia tách được thành 2 bộ lọc 1-chiều theo các định nghĩa tại (3) và (9). Cụ thể:

$$h(n_1,n_2)=\frac{1}{m\times n}=\frac{1}{m\times n}.1=h_1(n_1)h_2(n_2) \qquad (11)$$

Trong đó:

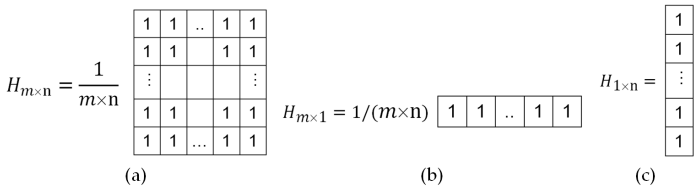
$$h_1(n_1)=\tfrac{1}{m\times n} \text{ với } 0\le n_1\le m \\ h_1(n_1)=0 \text{ ngược lại} \qquad (12)$$

và

$$h_2(n_2)=1 \text{ với } 0\le n_2\le n \\ h_2(n_2)=0 \text{ ngược lại} \qquad (13)$$

$h(n_1,n_2)$, $h_1(n_1)$ và $h_2(n_2)$ được xác định bởi các mặt nạ lọc như trong Hình 3.

Như vậy, bộ lọc trung bình 2-chiều ở mọi kích thước đều có thể được thực hiện một cách nhanh chóng thông qua 2 bộ lọc 1-chiều theo quy trình được mô tả ở Hình 2. Theo cách thực hiện này, và từ các công thức (8) và (10), chúng ta suy ra số phép toán cần thực hiện sẽ giảm từ:



Hình. 4. Mặt nạ lọc của bộ lọc trung bình 2-chiều kích thước m×n và 2 thành phần chia tách của nó. (a) Mặt nạ lọc trung bình 2-chiều , (b) & (c) Mặt nạ lọc của các chia tách $h_1(n_1)$ theo dòng và $h_2(n_2)$ theo cột..

(N×M)×(n×m) phép toán cộng và (N×M) phép toán chia ở phương pháp gốc,

xuống còn:

N×M×(n+m) phép toán cộng và (N×M) phép toán chia. (14)

Hay nói cách khác, khi m = n, sẽ giảm được (N×M×m)/2 số phép toán cộng.

4. Đề xuất kỹ thuật tối ưu hóa quy trình thực hiện bộ lọc trung bình 2-chiều.

Trong phần này, chúng tôi tập trung phân tích một số đặc điểm quan trọng trong quy trình thực hiện nhân cuộn tín hiệu ảnh đầu vào với 2 bộ lọc thành phần $h_1(n_1)$ và $h_2(n_2)$, như đã được trình bày trong mục 2.

Trước tiên, chúng ta tiến hành nhân cuộn các cột của tín hiệu ảnh 2-chiều $x(n_1, n_2)$ với đáp ứng xung $h_2(n_2)$ để thu được tín hiệu trung gian $f(k_1, n_2)$ theo công thức (5) sẽ trở thành:

$$f(k_1,n_2) = \sum_{k_2=-\infty}^{\infty} x(k_1,k_2)h_2(n_2-k_2) = \sum_{k_2=n_2-n+1}^{n_2} x(k_1,k_2).1 = \sum_{k_2=n_2-n+1}^{n_2} x(k_1,k_2) \quad (15)$$

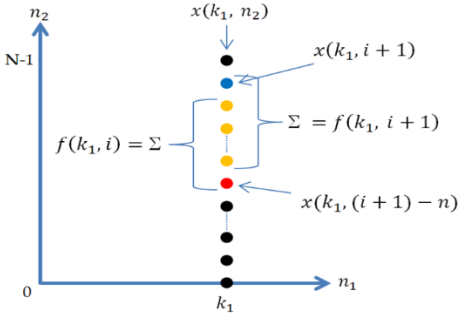
Với mọi giá trị i thuộc biến số n_2, chúng ta sẽ tìm cách biểu diễn đầu ra $f(k_1, i+1)$ dựa vào $f(k_1, i)$ đã có trước đó như sau:

$$f(k_1,i) = \sum_{k_2=i-n+1}^{i} x(k_1,k_2) \quad (16)$$

$$f(k_1,i+1) = \sum_{k_2=i-n+2}^{i+1} x(k_1,k_2) = \sum_{k_2=i-n+1}^{i} x(k_1,k_2) + x(k_1,i+1) - x(k_1,i-n+1) = f(k_1,i) + x(k_1,i+1) - x(k_1,(i+1)-n) \quad (17)$$

Như vậy, với mỗi giá trị k_1 cố định, giá trị đầu ra của f tại vị trí $(k_1, i+1)$ được tính bằng chính đầu ra của f tại vị trí trước đó là (k_1, i) cộng với giá trị đầu vào của x tại $(k_1, i+1)$ và trừ đi giá trị x tại $(k_1, (i+1)-n)$.

Như vậy, với phương pháp tính truy hồi qua công thức (17), để có một giá trị đầu ra f ở cột k_1 chúng ta cần mất một chi phí là 2 phép toán cộng, ngoại trừ giá trị đầu tiên của cột ứng với $n_2 = 0$ thì vẫn phải thực hiện n phép toán cộng.



Hình. 5. Minh họa cách tính nhanh các giá trị f qua phương pháp tính truy hồi.

Lập luận tương tự với giai đoạn tiếp theo, chúng ta sử dụng tín hiệu đầu ra của giai đoạn trước đó là f nhân cuộn 1-chiều với $h_1(n_1)$ để thu được đầu ra y, như sau:

$$y(n_1,n_2) = \sum_{k_1=-\infty}^{\infty} h_1(n_1-k_1)\, f(k_1,n_2) = \frac{1}{m\times n}\sum_{k_1=n_1-m+1}^{n_1} 1.f(k_1,n_2) = \frac{1}{m\times n}\sum_{k_1=n_1-m+1}^{n_1} f(k_1,n_2) \quad (18)$$

Với mọi giá trị i thuộc biến số n_1, chúng ta sẽ tìm cách biểu diễn đầu ra $y(i+1, n_1)$ dựa vào $y(i, n_1)$ đã có trước đó như sau:

$$y(i,n_2) = \frac{1}{m\times n}\sum_{k_1=i-m+1}^{i} f(k_1,n_2) = \frac{S_i}{m\times n} \quad (19)$$

Trong đó:

$$S_i \overset{\text{def}}{=} \sum_{k_1=i-m+1}^{i} f(k_1, n_2)$$

$$y(i+1, n_2) = \frac{1}{m \times n} \sum_{k_1=i-m+2}^{i+1} f(k_1, n_2)$$

$$= \frac{1}{m \times n} \left[\sum_{k_1=i-m+1}^{i} f(k_1, n_2) + f(i+1, n_2) - f(i+1-m, n_2) \right] \quad (20)$$

$$= \frac{1}{m \times n} [S_i + f(i+1, n_2) - f(i+1-m, n_2)]$$

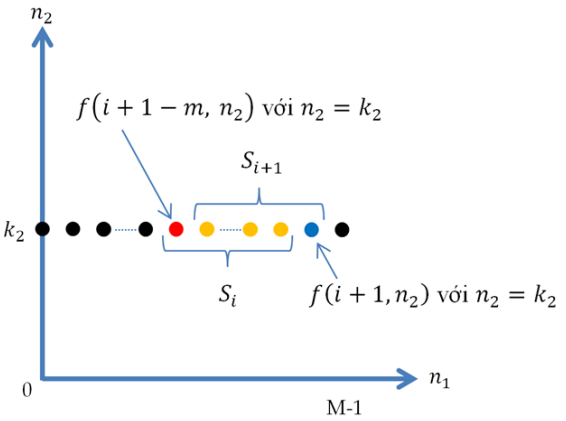
Thực hiện tính truy hồi cho các giá trị S_i qua công thức sau:

$$S_{i+1} = S_i + f(i+1, n_2) - f(i+1-m, n_2) \quad (21)$$

Từ (20) và (21) chúng ta có:

$$y(i+1, n_2) = \frac{S_{i+1}}{m \times n} \quad (22)$$

Như vậy, với phương pháp tính truy hồi qua công thức (21) và (22), để có mỗi giá trị đầu ra y ở dòng n_2, chúng ta cần mất một chi phí là 2 phép toán cộng và một phép chia với hằng số nguyên m×n, ngoại trừ giá trị đầu tiên của dòng ứng với $n_1 = 0$ thì vẫn phải thực hiện m phép toán cộng và một phép chia nguyên. Mặt khác, để có thể thực hiện tính truy hồi, chúng ta cần phải có một biến nhớ để lưu trữ giá trị tổng S_i thu được ở mỗi bước để phục vụ cho bước tính sau. Hình 6 minh họa cho phương pháp tính các giá trị S dựa vào truy hồi.



Hình. 6. Minh họa cách tính nhanh các giá trị y qua phương pháp tính truy hồi dựa trên các công thức (21) và (22).

Tóm lại, tổng số phép toán cần thực hiện để thu được toàn bộ tín hiệu ảnh đầu ra y dựa vào tín hiệu ảnh đầu vào x theo đề xuất mới sẽ là:

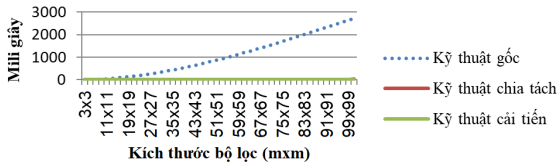
$[(2\times M\times(N+n-1)) + M\times n] + [2\times(M+m-1)\times(N+n-1) + (N+n-1)\times m]$ phép cộng và $(M+m-1)\times(N+n-1)$ phép chia nguyên (23)

Khi M,N ≫m,n thì số phép toán xấp xỉ là:

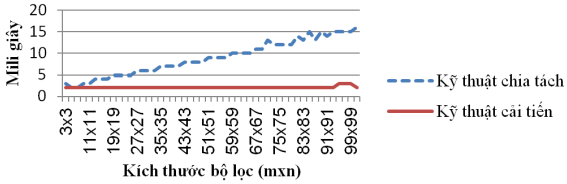
$(4\times M\times N)$ phép cộng và $M\times N$ phép chia nguyên (24)

5. Kết quả thực nghiệm và thảo luận

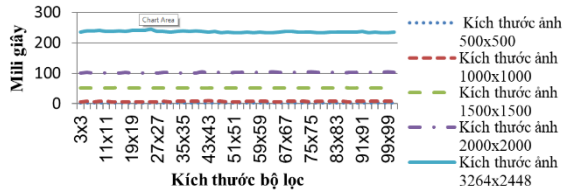
Chúng tôi đã tiến hành thực nghiệm cài đặt các đề xuất trên môi trường lập trình Microsoft Visual C++ MFC, kết quả thực nghiệm được thể hiện qua các hình dưới đây:



Hình. 7. So sánh thời gian thực hiện bộ lọc trung bình theo các kỹ thuật khác nhau dựa trên ảnh đầu vào có kích thước 500x500, thực nghiệm trên cùng một máy tính HĐH Windows 10.



Hình. 8. So sánh thời gian thực hiện bộ lọc trung bình giữa kỹ thuật chia tách và kỹ thuật cải tiến, thực nghiệm trên cùng một máy tính HĐH Windows 10.



Hình. 9. So sánh thời gian thực hiện bộ lọc trung bình theo kỹ thuật cải tiến với các kích thước ảnh đầu vào khác nhau, thực nghiệm trên cùng một máy tính HĐH Windows 10.

Kết quả thể hiện trong Hình 7 cho thấy kỹ thuật chia tách và kỹ thuật cải tiến đem lại sự cải thiện rất lớn về tốc độ khi kích thước của bộ lọc tăng trưởng. Và nó cũng cho thấy kỹ thuật gốc có chi phí thời gian tăng nhanh theo kích thước bộ lọc, chi phí thời gian này dễ dàng tiến tới ngưỡng khó chấp nhận được với các ứng dụng đòi hỏi tốc độ đáp ứng cao mà điển

hình trong số này là các ứng dụng thời gian thực (real time).

Hình 8 cho thấy sự khác biệt lớn về thời gian thực hiện theo kỹ thuật chia tách và kỹ thuật cải tiến theo sự tăng trưởng của kích thước bộ lọc.

Hình 9 thể hiện chi phí thời gian thực hiện bộ lọc theo kỹ thuật cải tiến với kích thước bộ lọc và kích thước ảnh đầu vào thay đổi. Nó cũng cho thấy thời gian thực hiện theo kỹ thuật cải tiến gần như là bất biến với sự thay đổi kích thước của bộ lọc.

Tham khảo thêm video thực nghiệm so sánh tốc độ, giữa giải pháp cải tiến cài đặt trong phần mềm ImPro của chúng tôi với giải pháp gốc được cài đặt trong chức năng lọc làm mờ (Box Blur) của phần mềm Adobe Photoshop tại [5].

6. Kết luận

Qua các phân tích về mặt lý thuyết cũng như các kết quả thực nghiệm, đã thể hiện rõ sự tối ưu của kỹ thuật mà chúng tôi đề xuất trong bài báo này. Việc áp dụng kỹ thuật này dưới dạng phần mềm (software) sẽ giúp đẩy nhanh tốc độ tính toán, mang lại tính khả thi cao cho các ứng dụng phần mềm đòi hỏi xử lý thời gian thực. Nếu triển khai dưới dạng mạch xử lý (hardware) sẽ giúp giảm các phần tử (nút) tính toán, từ đó ảnh hưởng tích cực đến chi phí thiết kế và sản xuất.

Xa hơn nữa, kỹ thuật cải tiến mà chúng tôi đề xuất có thể mở rộng một cách tự nhiên cho các bài toán lọc tín hiệu nhiều chiều bằng bộ lọc trung bình không gian, điều mà các kỹ thuật [7] và [8] không thể làm được bởi cách tiếp cận đơn giản và bó buộc trong không gian 2 chiều của nó.

Tài liệu tham khảo

[1] JAE S. LIM, Two-Dimensional Signal and Image Processing, Prentice Hall Ptr, USA (1990), page 16-20.

[2] A.K.Jain, Fundamentals of Digital Image Processing, Prentice Hall (1989), page 246.

[3] Gonzalez RC, RE Woods, Digital image processing, 2nd edn. Prentice Hall (2002), Upper Saddle River.

[4] Qingkun Song et al, Image Denoising Based on Mean Filter and Wavelet Transform, 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS), Harbin, China.

[5] A. Kundu, S. Mitra, P. Vaidyanathan, Application of two-dimensional generalized mean filtering for removal of impulse noises from images, IEEE Transactions on Acoustics, Speech, and Signal Processing (1984), Vol. 32, Iss. 3.

[6] Songyot Nakariyakul, Fast spatial averaging: an efficient algorithm for 2D mean filtering, Springer Science+Business Media (2011), Vol. 65, pp 262–273.

[7] Rajesh Reddy Datla, Adaptive fast spatial averaging filter, 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, DOI: 10.1109/ICCCNT.2017.8203929.

[8] Archer et al., Two dimentional moving average filter, United States Patent, Patent No.: US 6,389,441 B1, Date of Patent: May 14, 2002.

[9] Video thực nghiệm so sánh với kỹ thuật gốc trong phần mềm Adobe Photoshop. Link: https://youtu.be/U3KJ9mI_J6M.