Tạp Chí Tin Học và Điều Khiển Học
Tập IX(1993), Số 4, 10-19

Một Vài Khía Cạnh Thuật Toán Của Các Kỹ Thuật Lọc và Làm Rõ Đường Biên Trong Xử Lý Ảnh

Hoàng Kiếm & Trần Vĩnh Phước
TP HCM,

I. Mở đầu

Việc khôi phục, nâng cao chất lượng ảnh của các lọc nhiễu ảnh và làm rõ nét các chi tiết cần thiết trong ảnh là công đoạn quan trọng trong quá trình xử lý ảnh. Đến nay đã có nhiều phương pháp và thuật toán lọc nhiễu ảnh [2, 3, 6, 11]. Bài này nhằm phân tích ưu nhược điểm và độ phức tính toán của từng thuật toán lọc và đưa ra một thuật toán lọc nhiễu mới dựa trên phương pháp k-cực trị.

Những thuật toán trình bày dưới đây áp dụng trên ảnh được lấy mẫu (đúng điều kiện lấy mẫu không bị trùng lặp) thành những pixel, cường độ mỗi pixel được lượng tử hóa bằng 8 bit (giá trị thập phân từ 0 đến 255)

2. Lọc nhiễu ảnh

Ảnh số bị nhiễu có thể lọc bằng các phương pháp sau:

Lọc trung bình: [2, 6, 11]

Trên cơ sở cho rằng pixel sai có giá trị khác xa với giá trị các pixel lân cận. Do đó để loại nhiễu, thuật toán lọc trung bình tính lại giá trị pixel theo trung bình cộng các giá trị lân cận. Thí dụ: để $v(m, n)$ tại vị trí X trong cửa sổ 3×3

01	02	03
08	X	04
07	06	05

ta tính $y(m,n)$ là trung bình cộng 8 điểm lân cận:

$$\text{nếu} \quad y(m,n) - \tfrac{1}{8}\textstyle\sum_{i=1}^{8} 0_i \leq \varepsilon : \; v(m,n) = y(m,n).$$

$$\text{nếu} \quad y(m,n) - \tfrac{1}{8}\textstyle\sum_{i=1}^{8} 0_i \geq \varepsilon : \; v(m,n) = \tfrac{1}{8}\textstyle\sum_{i=1}^{8} 0_i.$$

với ε chọn trước.

Tổng quát, lọc trung bình có trọng số được biểu diễn:

$$v(m,n) = \sum_{(k,l)\in W} a(k,l)u(m-k, n-1)$$

với $W = (k,l)$: cửa sổ, $a(k,l)$: trọng số của thuật toán lọc.

Trường hợp đặc biệt $a(k,l)$ = hằng số, $V(k,l)$ - tất cả trọng số bằng nhau, $a(k,l) = 1/N_w$ (N_w : số điểm trong W):

$$v(m,n) = 1/overN_w \sum_{(k,l)\in W} u(m-1, k-1).$$

Áp dụng thuật toán trên, có thể khái quát bằng giả thuật chập:

Chập mặt nạ $Lm \times Ln$ gồm các phần tử $k(i,j)$, $i = -(Lm-1)/2, \ldots, 0, \ldots (Lm-1)/2$; $j = -(Ln-1)/2, \ldots 0, \ldots, (ln-1)/2$, lên mặt phẳng để tính:

$$v(m,n) = \sum_{i=-(Lm-1)/2}^{(Lm-1)/2} \; \sum_{i=-(Ln-1)/2}^{(Ln-1)/2} u(m+i, n+j)k(i,j)$$

Các mặt nạ lọc trung bình:

$$\frac{1}{9}\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad \frac{1}{10}\begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad \frac{1}{16}\begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix} \quad \frac{1}{8}\begin{vmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

Lọc trung bình có hướng là phương pháp lọc bằng cách tính giá trị trung bình các điểm ảnh theo một hướng θ nào đó:

$$\begin{array}{cccccl} * & * & * & * & * & \\ * & * & * & * & * & \theta \\ * & * & * & * & * & \longrightarrow 1 \\ * & * & * & * & * & \\ * & * & * & * & * & \end{array}$$

$$v(m,n,8) = \frac{1}{N_\theta} \sum_{(k,l)\in W} u(m-k, n-l)$$

với hướng θ được chọn sao cho $|u(m,n) - v(m,n,8)|$ nhỏ nhất, và kết quả:

$$v(m,n) = v(m,n,8).$$

Phương pháp này hạn chế nhòe sau khi lọc.

H1: Lọc trung bình có hướng

Mệnh đề 1: *Gọi N_w là số phần tử trong cửa sổ, lọc trung bình làm giảm mức nhiễu N_w lần.*

Chứng minh

$$u(m,n) = y(m,n) + \eta(m,n)$$

trong đó

$y(m,n)$ - ảnh không nhiễu

$\eta(m,n)$ - nhiễu trắng có trung bình bằng không, variance ảnh ra lọc trung bình:

$$v(m,n) = \frac{1}{N_w}\{\sum_{(k,l)\in W}[y(m-k,n-l)+\eta(m,n)]\}$$

$$= \frac{1}{N_w}\sum_{(k,l)\in W} y(m-k,n-l) + \frac{1}{N_w}\eta(m,n)$$

Thành phần nhiễu trong $v(m,n) = (1/N_w)\eta(m,n)$ có:

- trung bình $\overline{\eta} = 0$
- $\overline{\sigma}^2 = (1/N_w)\sigma^2$ Q.E.D.

Nếu $y(m,n)$ là hằng số trong cửa sổ W thì lọc trung bình không làm thay đổi tín hiệu, chỉ làm tăng tỉ số tín hiệu/nhiễu N_w lần. Trong thực tế rất khó chọn W để $y(m,n)$ là hằng số, do đó lọc trung bình dẫn đến méo có dạng nhòe ảnh.

Mệnh đề 2: *Đối với ảnh kích thước $M \times N$, có thể sử dụng $2N(3M+2)$ phép cộng và $2n(2M+1)$ phép nhân để chập mặt nạ 3×3 đối xứng hoặc sử dụng $(M-2)(N-2)$ phép nhân và $(M-2)(N_2)$ phép so sánh để lọc trung bình.*

Kết quả này được suy trực tiếp qua các giải thuật:

* Chập mặt nạ:

1. Tính $T = 1/\sum_{i,j} k(i,j)$
2. Chập cột thứ ba của mặt nạ vào cửa sổ.
3. Cộng kết quả này với kết quả chập cột thứ ba của lần tính trước, cộng chập từng điểm của lần thứ hai.
4. Nhân kết quả với T.
5. Trượt cửa sổ qua 1 pixel, thực hiện lại từ 2.

* Lọc trung bình:

1. Cộng cột thứ 3 của sổ
2. Cộng kết quả này với cột thứ 3 cửa sổ trước, hai phần tử trên dưới cột thứ 2.
3. Chia lấy trung bình
4. So sánh với ε cho trước.

Trong cả 2 giải thuật, nếu ảnh có $M <> N$, chọn $M > N$, số phép tính ít hơn.

2) *Lọc median* [6, 8, 11]

Định nghĩa:

1) Median một chuỗi $\{x_n\}$ có n phần tử x_a sao cho:

- nếu n lẻ: có $(n-1)/2$ phần tử nhỏ hơn hay bằng x_a và $(n-1)/2$ phần tử lớn hơn hay bằng x_a.
- nếu n chẵn: x_a là trung bình cộng của 2 phần tử x_i, $x_j \in \{x_n\}$, có $(n-2)/2$ phần tử $\leq x_i$ và $(n-2)/2 \geq x_j$.

Thuật toán median được dùng lọc nhiễu ảnh bằng cách tạo những cửa sổ trượt trên mặt phẳng ảnh, mỗi lần trượt di chuyển 1 cột pixel, những phần tử trong cửa sổ xem như chuỗi $\{x_n\}$.

2) Chuỗi $\{x_n\}$ được gọi là LOMO(m) [8] nếu có:

$$(x_n, x_{n+1}, \ldots, x_{n+m-1}) \text{ đơn điệu } \forall n.$$

Chuỗi LOMO(m) có tính chất [8]: Chuỗi LOMO(m) muốn đổi hướng phải có ít nhất $m-1$ phần tử liên tiếp bằng nhau.

Mệnh đề 3: *Lọc median bằng cửa sổ có chiều dài K, $K \leq 2m-4$, không làm thay đổi chuỗi LOMO(m).*

Chứng minh: Xét đoạn

$$\{x_k\} = (x_{n-K/2}, \ldots, x_n, \ldots x_{n+K/2}) \subset \{x_n\},$$

gồm có $K+1$ phần tử, có 2 trường hợp xảy ra:

- nếu $\{x_k\}$ đơn điệu, dĩ nhiên median $\{x_n\} = x_n$.
- nếu có sự đổi hướng trong $\{x_k\}$. Theo tính chất chuỗi LOMO(m), có $m-1$ phần tử liên tiếp bằng nhau trong $\{x_k\}$, mà $K \leq 2m-4 \Rightarrow m-1 \geq (K/2)+1$. Trong $\{x_k\}$ có ít nhất $(K/2)+1$ phần tử liên tiếp bằng nhau và bằng x_n vì bất kỳ đoạn nào của $\{x_k\}$ có chiều dài $K/2+1$ đều chứa x_n, suy ra median $\{x_n\} = x_n$. Q.E.D.

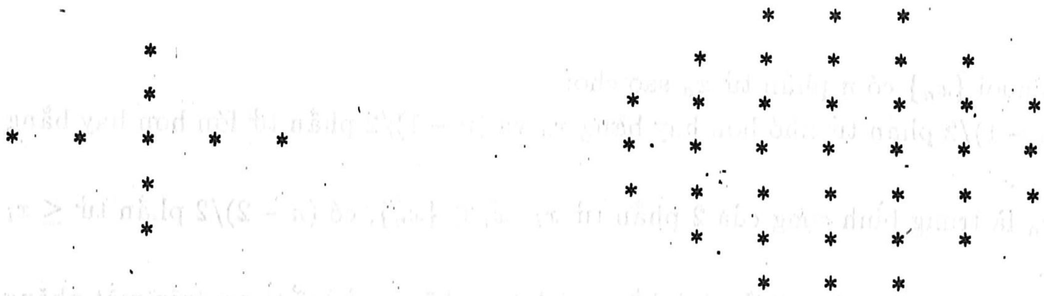
Mệnh đề 4: *Median lọc được nhiễu khi chiều dài cửa sổ lớn hơn 2 lần chiều dài chuỗi nhiễu.*

Chứng minh: Thật vậy, gọi m là chiều dài lớn nhất của chuỗi nhiễu, L là chiều dài cửa sổ. Nếu $L \leq 2m-4$ thì theo mệnh đề 3, nhiễu được giữ nguyên. Ngược lại, $L > 2m > 2m-4$, các phần tử giá trị cao như nhiễu được gán giá trị mới là median của cửa sổ, nhỏ hơn giá trị cũ. Median lọc được nhiễu. Q.E.D.

Áp dụng kết quả này để lọc nhiễu ảnh, kích thước cửa sổ phải được chọn theo đặc tính LOMO của chuỗi ảnh và chuỗi nhiễu, nghĩa là phải chọn cửa sổ median lớn hơn 2 lần chiều dài chuỗi nhiễu lớn nhất, đồng thời phải nhỏ hơn 2 lần chiều dài LOMO của chuỗi pixel ảnh.

Do đó, những ảnh có giá trị pixel thay đổi chậm, median lọc được nhiễu, những ảnh có giá trị pixel thay đổi nhanh, median không lọc được nhiễu.

Mặt khác, về phương diện hình học, nhiễu thường thể hiện lên ảnh thành những điểm tròn, ảnh ít bị suy hao hơn là dùng cửa sổ vuông.



Hình 2: cửa sổ hình chữ thập 5×5 cửa sổ giả tròn 7×7

Giải thuật của T.S.Huang, tính median cửa sổ vuông cột n hàng bằng cách xem những phần tử trong cửa sổ như những phận tử của chuỗi một chiều. Sau mỗi lần tính, bỏ ra khỏi chuỗi một phần tử để nhận vào một phần tử mới, giải thuật bắt đầu tính bằng cách tính histogram của từng cửa sổ và so sánh các bậc của histogram để tìm median. Mỗi cửa sổ cần thực hiện một số phép so sánh:

$$\bar{c} = 2n + |\bar{d}| + 1 + \text{Prob}(d \geq 0)$$

với:

$\bar{c}$ - số phép so sánh trung bình cho một cửa sổ

$d =$ (median cửa sổ hiện hành) - (median cửa sổ trước)

3. Làm rõ đường biên

Thực nghiệm đã chứng minh hình ảnh hoặc tín hiệu hình có đường biên rõ làm cho người xem cảm thấy dễ chịu hơn là hình ảnh nguyên gốc. Do đó, các kỹ thuật làm rõ đường biên được sử dụng nhiều trong công nghệ hình ảnh. Làm rõ đường biên còn được sử dụng để khắc phục nhòe sau khi áp dụng các thuật toán lọc trung bình.

Thuật toán Laplace:

Xuất phát từ yêu cầu tăng tốc độ phần giao tiếp giữa 2 mức xám thấp và cao để làm rõ nét đường biên, có thể lấy đạo hàm bậc hai trong không gian 2 chiều - toán tử Laplace:

$$v(m,n) = \nabla^2 u(m,n) = \frac{\partial^2 u(m,n)}{\partial m^2} + \frac{\partial^2 u(m,n)}{\partial n^2}$$

Trong không gian rời rạc, đạo hàm bậc hai được định nghĩa cho cửa sổ ảnh vào 3×3 là:

$$\frac{\partial^2 u(m,n)}{\partial m^2} = (e - d) - (f - e).$$

$$\frac{\partial^2 u(m,n)}{\partial n^2} = (e - b) - (h - e).$$

và

$$\nabla^2 u(m,n) = 4e - (b + h + d + f).$$

Tổng quát:

$$v(m,n) = u(m,n) - (1/4)[u(m-1,n) + u(m,n-1) + u(m+1,n) + u(m,n+1)].$$

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

H3: đường biên một chiều — Hình 4: cửa sổ Laplace 3×3

Suy rộng có thể làm rõ đường biên bằng lọc chập với:

$$H = \begin{vmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{vmatrix} \quad \text{hoặc } H = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix} \quad \text{hoặc } H = \begin{vmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{vmatrix}$$

2. Lọc thông cao:

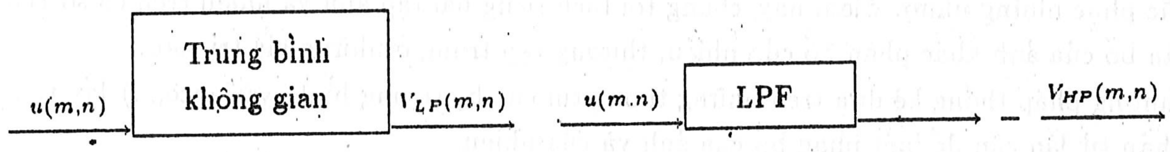
Sau khi áp dụng các thuật toán san bằng nhiễu, hoặc các thuật toán nội suy, ảnh bị nhòe vì các thành phần tần số không gian cao bị nén bớt - lọc thông thấp. Do đó để làm rõ đường biên, có thể cho ảnh nhòe qua lọc thông cao, được định nghĩa:

$$h_{HP}(m,n) = \delta(m,n) - H_{LP}(m,n)$$

trong đó:

- $h_{HP}(m,n)$ là lọc thông cao FIR
- $H_{LP}(m,n)$ là lọc thông thấp FIR

theo sơ đồ sau:



3) Kỹ thuật mặt nạ không nét (unsharp masking):

Xuất phát từ mục đích xóa nhòe do lọc nhiễu gây ra, thuật toán này khảo sát mức xám của hai điểm lân cận

$$\begin{aligned} v(m,n) &= u(m,n) + \lambda[u(m,n) - u_L(m,n)] \\ &= u(m,n) + \lambda g(m,n) \\ &= (1+\lambda)u(m,n) - \lambda u_L(m,n) \end{aligned}$$

Đặt

$$c = 1 + \lambda > 1 \Rightarrow c - 1 > 0$$

$$v(m,n) = cu(m,n) - (c-1)u_L(m,n)$$

4) Ánh xạ nghịch đảo tương phản:

Khả năng mắt nhìn rõ một vật trên một nền đồng nhất phụ thuộc độ phân giải, và tỷ số tương phản γ được định nghĩa:

$$\gamma = \sigma/\mu$$

μ - độ chói trung bình của vật.

σ - độ lệch tiêu chuẩn giữa độ chói của vật với độ chói của nền.

Biến đổi nghịch đảo tương phản được áp dụng để tăng độ rõ nét của đường biên:

$$v(m,n) = \frac{\mu(m,n)}{\sigma(m,n)}$$

với $\mu(m,n)$ và $\sigma(m,n)$ là trung bình cục bộ và độ lệch tiêu chuẩn cục bộ của $u(m,n)$ trên cửa sổ W:

$$\mu(m,n) = \frac{1}{N_w} \sum_{(k,l)\in W} u(m-k, n-l)u(m-k, n-l)$$

$$\sigma(m,n) = \{\frac{1}{N_w} \sum_{(k,l)\in W} [u(m-k, n-l) - \mu(m,n)]^2\}(1/2)$$

4. Ứng dụng phương pháp k-cực trị của phân bố mẫu

Thuật toán median sử dụng bình đẳng đối với những pixels ảnh và nhiễu, những pixels ảnh có giá trị cao và thấp bị loại bỏ để thay vào đó những giá trị median. Do đó median cũng làm giảm chất lượng ảnh và có những trường hợp ít tác dụng lọc.

Để khắc phục những nhược điểm này, chúng tôi tách riêng hai tập ảnh và nhiễu trên cơ sở cho rằng phân bố của ảnh khác phân bố của nhiễu, thường tập trung ở những giá trị cao.

Các phương pháp thống kê dựa trên những thông tin cục bộ (trung bình, variance,...) lấy trên những phần tử lân cận để biết phân bố của ảnh và của nhiễu.

Xem những phần tử trong cửa sổ $W(m,n)$ là lân cận của $u(m,n)$. Tập $W(m,n)$ được chia làm hai tập $W^*(m,n)$ chứa những phần tử giá trị cao, $W(m,n) \setminus W^*(m,n)$ chứa những phần tử giá trị thấp, nghĩa là nếu những pixels của $W(m,n)$ được xếp theo thứ tự tăng dần thì

$$X(1),\ X(2), \ldots X(k), \qquad W(m,n) \setminus W^*(m,n)$$

$$X(N-1),\ X(N-2), \ldots X(N-k), \qquad W^*(m,n)$$

và

$$M[W(m,n)] > M[W(m,n) \setminus W^*(m,n)]$$

với

- $X(k)$ - k-trị của chuỗi tự $W(m,n)$.
- $M(A)$ - trung bình những pixels của A.

Có thể phân biệt hai tập $W^*(m,n)$ và $W(m;n) \setminus W^*(m,n)$ khi biết trung bình và variance của mỗi tập.

Ở đây, chúng tôi đề nghị dùng thông tin k-cực trị để tìm ranh giới tập ảnh $W(m,n)\backslash W^*(m,n)$ từ một vài giá trị thấp và tập nhiễu $W^*(m,n)$ từ một vài giá trị cao trong cửa sổ được chọn 5×5 hoặc 7×7.

Định lý lọc nhiễu theo k-cực trị: [4]

Cho $E[X(k)]$ là trung bình k-cực trị của mẫu n phần tử có phân bố bình thường ? . Nếu n, u chưa biết, giá trị n có thể được tính bằng trung bình của k-1, k+1, - trị của mẫu đó.

Chứng minh Từ [4, 5] ta có

$$E_n[X(k)] = \mu - \sigma[\sqrt{\frac{2\ln(n) - \ln(\ln(n)) + \ln(4\pi) + 2[S_1(k) - C]}{2\sqrt{2\ln(n)}}} + O(\frac{1}{\ln n})]$$

với

$$S_1(k) = \sum_{i=1}^{n-k+1} \frac{1}{i}$$

và C là hằng số Euler.

Áp dụng kết quả này tính $E_n[X(k-1)], E_n[X(k+1)]$:

$$E_n[X(k-1)] = \mu - \sigma[\sqrt{\frac{2\ln(n) - \ln(\ln(n)) + \ln(4\pi) + 2[S_1(k-1) - C]}{2\sqrt{2\ln(n)}}} + O(\frac{1}{\ln n})]$$

$$E_n[X(k+1)] = \mu - \sigma[\sqrt{\frac{2\ln(n) - \ln(\ln(n)) + \ln(4\pi) + 2[S_1(k+1) - C]}{2\sqrt{2\ln(n)}}} + O(\frac{1}{\ln n})]$$

Suy ra

$$\frac{E_n[X(k)] - E_n[X(k+1)]}{E_n[X(k+1)] - E_n[X(k)]} \sim \frac{S_1(k) - S_1(k-1)}{S_1(k+1) - S_1(k)} = \frac{n-k+3}{n-k+2}$$

với $k = 2$,

$$\frac{E_n[X(2)] - E_n[X(1)]}{E_n[X(3)] - E_n[X(2)]} \sim \frac{n+1}{n}$$

Kết quả

$$n \sim \frac{1}{\frac{E_n[X(2)] - E_n[X(1)]}{E_n[X(3)] - E_n[X(2)]} - 1} = \frac{E_n[X(3)] - E_n[X(2)]}{2E_n[X(2)] - E_n[X(1)] - E_n[X(3)]}$$

Mệnh đề 5: *Với ảnh có kích thước $M\times N$, cửa sổ vuông cạnh L giải thuật được trên thực hiện bằng $28MN/L^2$ phép tính nhân và $24MN/L^2$ phép tính cộng.*

Chứng minh Kết quả mệnh đề có thể suy trực tiếp trên thuật toán:

1. Sắp xếp những pixels trong cửa sổ $W(m,n)$ theo thứ tự giá trị tăng dần.
2. Tính trung bình 1-trị, 2-trị, 3-trị bằng trung bình có trọng số

$$E_n[X(k)] \sim \frac{4X(k) + 3X(k+1) + 2X(k+2) + X(k+3)}{10}$$

3. Tính n_a của phân bố ảnh, n_n của phân bố nhiễu theo công thức định lý trên.

4. Có 2 trường hợp:

* $n_a + n_n \leq N$: phân bố ảnh giữ nguyên. Nhiễu bị loại hoàn toàn, không mất thông tin ảnh.

* $n_a + n_n > N$: có một phần phân bố ảnh và nhiễu chồng lên nhau. Trong trường hợp này, hoặc phải chịu mất một phần tín hiệu ảnh hoặc phải chấp nhận một phần nhiễu, chúng tôi đề nghị chọn biển theo tỷ lệ phân bố:

$$n'_a = n_a - n_n + Nn_n(n_a + n_n).$$

5. Những pixels nhiễu được gán giá trị mới trong chuỗi có vị trí thấp hơn vị trí cũ một đoạn $(N - n')$.

5. Kết luận

Mỗi thuật toán lọc trung bình, median, chập mặt nạ có khả năng lọc nhiễu, đồng thời cũng sinh ra một số suy hao nhất định, và cũng có độ tính toán phức tạp khác nhau. Thuật toán k-cực trị xử lý thích nghi theo phân bố nhiễu cho từng vùng, ít ảnh hưởng đến phân bố ảnh và do thuật toán giải quyết trên những cửa sổ kề cận nên số phép tính ít hơn.

Tài liệu tham khảo

1. Lev A. & all.., *Iterative enhancement of noisy images,* IEEE. Sys. Man. Cybernetics No 6 (1977), 435-442.
2. Benjamin M. Dawson, *Image idting lets you iteractively improve, rearrange, modify and adjust images any way you want* Byte No 12 (1989), 293-303.
3. Andrews H.C. & Hunt B.R., Digital Image Restoration, Prentice Hall, Inc. 1977.
4. Hoàng Kiếm, *The fast smoothing algorithms based on local operators,* Reasearch Report AI Lab. Bratislava No 3 (1983).
5. Hoàng Kiếm, Some models and algorithms for pattern resognition, Ph.D.Dissertation, Institute of Informatics and Cybernetics, Vietnam 1981.
6. Jain, Fundamentals of Digital Image Processing, Thomas Kailath Editor.
7. Naryana K.A. et all. , *Image smoothing by local use of Global information,* IEEE. Sys. Man Cybernetics, No 12 (1981), 826-831.
8. Thomas S. Huang, 2-D Digital Signal Processing. Tome II, Springer-Verlage Berlin Heidelberg 1981.
9. Lee T.S., *Digital image enhancement and noise filtering by use of local statics,* IEEE Trans. PAMI-2 No 2 (1980),165-168.
10. Lee T.S., *Rifined Filtering of image noise using local statics,* Computer Graphic and image processing No 15 (1981), 380-389.
11. William K. Pratt, Digital Image Processing, John Wiley & Sons 1978.

Abstract

Some algorithmic aspects of filtering and sharpening techniques in image processing

This paper presents an analysis of the complexity of traditional filtering and sharpening algorithms in image processing. After analysing their efficiency and computation complexity, we propose an approach based on the use k-values information of ordered set for restoring image with high quality and reducing the computation.

Hoàng Kiếm: Đại học TH TP. HCM, 227 Nguyễn Văn Cừ, Quận 5, Tel 350098.
Trần Vĩnh Phước: Đại học Bách khoa TP.HCM, 268 Lý Thường Kiệt, Q. 10, Tel 550157.