ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN – TIN HỌC

LỌC TRONG MIỀN TẦN SỐ VÀ LỌC CONTOURLET

Đề tài môn: Phân tích và Xử lý ảnh

GIẢNG VIÊN HƯỚNG DẪN: PGS. TS. Phạm Thế Bảo

SINH VIÊN THỰC HIỆN: Võ Hoàng Trọng

2015

NỘI DUNG

Lời mở đầu

Ngành xử lý ảnh xuất phát vào khoảng thập niên 60 của thế kỷ trước, có nhiều kỹ thuật được ứng dụng trong ảnh vệ tinh, chụp ảnh xa, ảnh y học, …

Một trong những vấn đề quan trọng của xử lý ảnh đó là lọc nhằm giúp cho ảnh rõ ràng, sắc nét hơn hay loại bỏ các tín hiệu "nhiễu" trong ảnh (dễ thấy nhất là các "chấm nhỏ" trên ảnh). Các kỹ thuật lọc này thường áp dụng ngay trên các điểm ảnh và lân cận (gọi là lọc trong miền không gian) hay xử lý thông qua các hàm sóng tần số sin và sóng cos của ảnh (gọi là lọc trong miền tần số). Bài báo cáo này sẽ trình bày về các phép lọc trong miền tần số cũng như một số kết quả thu được từ các phép lọc này.

Ngoài ra, vào khoảng đầu những năm 2000, Đỗ Ngọc Minh và Martin Vetterli đã đề xuất phép chuyển đổi Contourlet ứng dụng trong khử nhiễu, rút trích đặc trưng của anh. Bài báo cáo này sẽ trình bày những khái niệm cơ bản của chuyển đổi Contourlet cũng như một số ứng dụng.

Bài báo cáo này gồm 2 phần: Lọc ảnh trong miền tần số và lọc Contourlet

1. Lọc ảnh trong miền tần số: Tôi trình bày về cơ sở toán học, công thức thực hiện phép chuyển đổi Fourier, các bộ lọc mờ ảnh, sắc ảnh, lọc chặn và hình ảnh ví dụ cũng như ứng dụng thực tế của các bộ lọc này.

2. Lọc Contourlet: Tôi trình bày những ưu, nhược điểm của phép lọc Wavelet, từ đó nêu ra ý tưởng hình thành phép lọc Contourlet. Sau đó tôi sẽ trình bày những công cụ cần thiết của phép lọc này, thuật toán Contourlet và những ví dụ nhằm so sánh hiệu quả của lọc Contourlet so với lọc Wavelet, cuối cùng là một số ứng dụng của phép lọc Contourlet

Phần 1

LỌC TRONG MIỀN TẦN SỐ

1.1 Giới thiệu

Chương này trình bày về khái niệm miền tần số, công cụ toán học sử dụng cũng như các phép lọc trong miền tần số và ứng dụng. Ảnh vào là ảnh xám, ảnh ra là ảnh đã được lọc tùy theo bộ lọc sử dụng. Tôi sử dụng ngôn ngữ Matlab để trình bày thuật toán.

1.2 Miền tần số

Ta sẽ tìm hiểu khái niệm về tần số trong không gian 1 chiều

- Chu kỳ của $\cos t$ là 2π giây, tức tín hiệu sẽ lặp lại sau mỗi 2π giây.

- Chu kỳ của $\cos(2\pi t)$ là 1 giây.

- Ta ký hiệu chu kỳ là T.

- Đơn vị đo cho tần số là Hz (Hertz), ký hiệu là f, tương ứng với số chu kỳ xảy ra trong 1 giây

$$f = \frac{1}{T}$$

- Hàm $\cos(2\pi \cdot ft)$ có chu kỳ $1/T$ và tần số f.

- Đơn vị đo cho tần số góc là rad/s, ký hiệu là ω

$$\omega = \frac{2\pi}{T}$$

Ví dụ: Giả sử một bức ảnh có các điểm ảnh thỏa hàm số

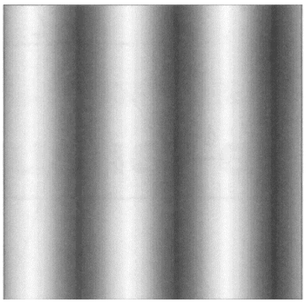
$$f(x, y) = 128 + A \sin\left(\frac{2\pi ux}{N-1} + \phi\right)$$

Khi đó, ta biết được ảnh có mức xám trung bình là 128, biên độ $A \in [1,127]$, độ rộng của ảnh là N, ϕ là pha và u là tần số không gian (số vòng hàm sin "vừa" với độ rộng của hình, chia cho N → tần số không gian trong 1 vòng đơn vị trên điểm ảnh).

Cho hàm

$$f(x, y) = 128 + 127 \sin\left(\frac{2\pi \cdot 3x}{100-1} + 0\right)$$

Ta được ảnh tương ứng là



Hình 1.2 - 1. Ảnh $f(x, y)$

1.3 Sự khác biệt giữa miền không gian và miền tần số

Trong miền không gian, ta xử lý trực tiếp trên từng điểm ảnh, còn trong miền tần số, ta xử lý dựa trên tốc độ thay đổi giá trị ảnh trên miền không gian.

- Miền không gian: Ma trận ảnh đầu vào → Xử lý → Ma trận ảnh đầu ra.

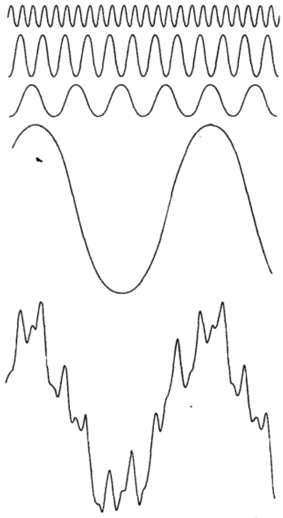
- Miền tần số: Ảnh vào → Phân bố tần số → Xử lý → Chuyển đổi ngược → Ảnh ra.

Miền tần số không gian có thể tạo ra mối quan hệ chu kỳ rõ ràng trong miền không gian, và trong miền tần số, một số toán tử xử lý ảnh sẽ trở nên hiệu quả hơn.

Trong nhiều trường hợp, người ta dùng chuyển đổi Fourier để chuyển ảnh từ miền không gian sang miền tần số.

1.4 Khái niệm về chuỗi Fourier và chuyển đổi Fourier

Chuỗi Fourier (Fourier series) được nhà Toán học người Pháp tên Jean Baptiste Joseph Fourier đưa ra vào thế kỷ 19. Ông khẳng định rằng với bất kỳ hàm số $f(t)$ tuần hoàn với chu kỳ T đều có thể biểu diễn được dưới dạng tổng của các hàm số sine và cosine với những tần số khác nhau, mỗi hàm số nhân với một hệ số tương ứng. Khi đó, ta gọi tổng các chuỗi hàm số sine và cosine này là chuỗi Fourier.



Hình 1.4 - 1. Minh họa về chuỗi Fourier, hàm sóng ở dòng cuối cùng là kết quả tổ hợp tuyến tính của 4 hàm sóng ở trên

Hàm số ở dưới cùng chính là tổng của bốn hàm số ở phía trên

1.4.1 Chuyển đổi Fourier cho hàm liên tục

1.4.1.1 Hàm 1 biến

Chuỗi Fourier có dạng

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{T}t} \quad (1.4-1)$$

Với

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-j\frac{2\pi n}{T}t}\, dt, \quad n = 0, \pm 1, \pm 2, \pm 3, \dots \quad (1.4-2)$$

Phương trình (1.4 – 1) là cách khai triển hàm số sine và cosine theo công thức Euler trong trường số phức:

$$e^{j\theta} = \cos\theta + i\sin\theta$$

Đối với những hàm số không có tính tuần hoàn, nhưng diện tích dưới đường cong của hàm số đó là hữu hạn, ta có thể biểu diễn hàm số đó dưới dạng tích phân của hàm sine và cosine nhân với hàm trọng số. Biểu thức thu được gọi là chuyển đổi Fourier.

Ta xác định phương trình chuyển đổi Fourier của một hàm số liên tục $f(t)$ có biến t liên tục như sau:

$$\Im\{f(t)\} = \int_{-\infty}^{+\infty} f(t)e^{-j2\pi\mu t}\, dt$$

với μ là biến liên tục. Vì khi lấy xong tích phân sẽ mất t nên ta chỉ còn lại μ, vậy ta sẽ ký hiệu lại phương trình chuyển đổi Fourier cho rõ ràng hơn như sau:

$$F(\mu) = \int_{-\infty}^{+\infty} f(t)e^{-j2\pi\mu t}\, dt \quad (1.4-4)$$

Ngược lại, cho trước $F(\mu)$, ta có thể tìm lại $f(t)$ bằng cách sử dụng chuyển đổi ngược Fourier (inverse Fourier transform), $f(t) = \Im^{-1}\{f(t)\}$, viết là:

$$f(t) = \int_{-\infty}^{+\infty} F(\mu)e^{j2\pi\mu t}\, d\mu$$

Sử dụng công thức Euler, ta viết lại phương trình (2) như sau:

$$F(\mu) = \int_{-\infty}^{+\infty} f(t)[\cos(2\pi\mu t) - j\sin(2\pi\mu t)]\ dt \quad (1.4-5)$$

Biến còn lại khi lấy tích phân là μ, chính là tần số của hàm lượng giác nên miền của chuyển đổi Fourier là miền tần số.

1.4.1.2 Hàm 2 biến

Với trường hợp hàm 2 biến, ta có chuyển đổi Fourier như sau:

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(ux+vy)}\, dx\, dy \quad (1.4-6)$$

Chuyển đổi ngược:

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{-j2\pi(ux+vy)} \quad (1.4-7)$$

1.4.2 Chuyển đổi Fourier cho hàm rời rạc

Vì các điểm ảnh là các dữ liệu rời rạc nên để áp dụng chuyển đổi Fourier, ta cần xây dựng một công thức sử dụng cho các biến rời rạc.

1.4.2.1 Hàm 1 biến

Giả sử ta có bộ dữ liệu dãy $x(n)$ (với $n = 1,2,3, \dots N$), ta xác định DFT cho x_n như sau:

$$X(n) = \frac{1}{N}\sum_{k=1}^{N} x_k e^{-jk2\pi n/N}$$

với $n = 1,2,3, \dots N$

Chuyển đổi ngược (IDFT) là

$$x(n) = \sum_{k=1}^{N} X(n) e^{jk2\pi n/N}$$

với $n = 1,2,3, \dots N$

Về bản chất, chuyển đổi Fourier sử dụng trong số phức, nhưng trong xử lý ảnh, ta xem phần ảo là 0.

1.4.2.2 Hàm 2 biến

Giả sử $f(x, y)$ là ảnh đầu vào, DFT cho hàm 2 biến này là

$$F(u, v) = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N} f(x, y) e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

Hàm ngược IDFT là

$$f(x, y) = \sum_{u=1}^{M}\sum_{v=1}^{N} F(u, v) e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

1.4.3 Chuyển đổi Fourier nhanh (FFT)

1.4.3.1 Hàm 1 biến

Để chuyển ảnh từ miền không gian sang miền tần số bằng cách sử dụng chuyển đổi Fourier thông thường đòi hỏi chi phi lớn khi ảnh lớn ($O(N^2)$ với N là số điểm ảnh), phép FFT cho ra chi phí rẻ hơn nhiều $\left(O(N \log_2 N)\right)$ với N là số điểm ảnh).

Thuật toán FFT thông dụng nhất là thuật toán do J.W. Cooley và John Tukey đề xuất, tính chuyển đổi Fourier cho các giá trị rời rạc bằng cách sử dụng đệ quy tính các giá trị ở vị trí chẵn và lẻ.

$$X_k = \underbrace{\sum_{m=1}^{\frac{N}{2}} x_{2m} e^{-\frac{2\pi i}{N}(2m)k}}_{\text{DFT cho các phần chẵn}} + \underbrace{\sum_{m=1}^{\frac{N}{2}} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}}_{\text{DFT cho các phần lẻ}} \quad (1.4-8)$$

$$X_k = \sum_{m=1}^{\frac{N}{2}} x_{2m} e^{-\frac{2\pi i}{N/2} mk} + e^{-\frac{2\pi i}{N}k} \sum_{m=1}^{\frac{N}{2}} x_{2m+1} e^{-\frac{2\pi i}{N/2} mk} \quad (1.4-9)$$

$$X_k = E_k + e^{-\frac{2\pi i}{N}k} O_k$$

với E_k là tổng các phần chẵn còn O_k là tổng các phần lẻ.

Do tính tuần hoàn có chu kỳ của DFT nên

$$E_{k+\frac{N}{2}} = E_k$$

và

$$O_{k+\frac{N}{2}} = O_k$$

Khi đó, ta viết lại phương trình (1.4 – 8) và (1.4 – 9) thành

$$X_k = \begin{cases} E_k + e^{-\frac{2\pi}{N}k} O_k & ; 0 \le k < \dfrac{N}{2} \\ E_{k-N/2} + e^{-\frac{2\pi}{N}k} O_{k-N/2} & ; \dfrac{N}{2} \le k < N \end{cases} \quad (1.4-10)$$

Mặt khác, $e^{-\frac{2\pi}{N}k}$ hình thành từ:

$$\begin{aligned} e^{-\frac{2\pi i}{N}(k+N/2)} &= e^{-\frac{2\pi ik}{N} - \pi i} \\ &= e^{-\pi i} e^{-\frac{2\pi ik}{N}} \\ &= -e^{-\frac{2\pi ik}{N}} \end{aligned}$$

Khi đó, ta có thể giảm khối lượng tính toán xuống một nửa. Với $0 \le K < N/2$, từ phương trình (1.4 – 10), ta được:

$$X_k = E_k + e^{-\frac{2\pi ik}{N}} O_k$$

$$X_{k+N/2} = E_k - e^{-\frac{2\pi ik}{N}} O_k$$

Thuật toán 1.4 – 1. Chuyển đổi Fourier nhanh (FFT)

```
Input: chuỗi số x, số lượng N, giá trị s
Output: chuỗi X = X0, …, X(N-1)
X0,...,N-1 ← ditfft2(x, N, s): \\DFT của(x_0,x_s,x_2s,…,x_(N-1)s):
    if N = 1 then
        X0 ← x0                     \\Trường hợp tầm thường
    else
        X0,...,N/2-1 ← ditfft2(x, N/2, 2s) \\DFT của(x_0,x_2s, …)
        XN/2,...,N-1 ← ditfft2(x+s, N/2, 2s)\\DFT của (x_s,x_3s,…)
        for k = 0 to N/2-1    \\Kết hợp 2 nửa DFTs thành một DFT:
            t ← Xk
```

```
            Xk ← t + exp(−2πi k/N) Xk+N/2
            Xk+N/2 ← t − exp(−2πi k/N) Xk+N/2
        endfor
    endif
```

1.4.3.2 Hàm 2 biến

Từ ý tưởng của hàm 1 biến, ta tính FFT hàm 2 biến bằng cách tính theo một chiều với mỗi giá trị cũa x (theo cột), sau dó tính ngược lại theo y (theo hàng) với giá trị thu được ở trên.

Trong Matlab, ta sử dụng hàm fft2 để chuyển ảnh sang chuỗi Fourier và dùng hàm ifft2 để trả ngược lại.

1.4.4 Định lý tích chập

Khi lọc ảnh hoặc tăng cường ảnh trong miền không gian, người ta sử dụng khái niệm tích chập, trong đó ảnh sau khi xử lý bằng tích chập của ảnh ban đầu và bộ lọc. Về mặt Toán học, cho 2 hàm số liên tục, ta định nghĩ tích chập $*$ của 2 hàm số này là

$$f(t) * h(t) = \int_{-\infty}^{+\infty} f(\tau)h(t-\tau)\,d\tau$$

Ta áp dụng tích chập cho 2 hàm số sau khi thực hiện chuyển đổi Fourier như sau

$$\Im\{f(t) * h(t)\} = \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} f(\tau)h(t-\tau)\,d\tau\right] e^{-j2\pi\mu t}\,dt$$

$$= \int_{-\infty}^{+\infty} f(\tau)\left[\int_{-\infty}^{+\infty} h(t-\tau)e^{-j2\pi\mu t}\,d\tau\right] dt$$

mà

$$\Im\{h(t-\tau)\} = H(\mu)e^{-j2\pi\mu t}$$

với $H(\mu)$ là chuyển đổi Fourier của $h(t)$, khi đó:

$$\Im\{f(t) * h(t)\} = \int_{-\infty}^{+\infty} f(\tau)\left[H(\mu)e^{-j2\pi\mu t}\right] dt$$

$$= H(\mu)\int_{-\infty}^{+\infty} f(\tau)e^{-j2\pi\mu t}\,dt$$

$$= H(\mu)F(\mu)$$

Vậy, giả sử $\Im\{f(t)\} = F(\mu), \Im\{h(t)\} = H(\mu)$, khi đó

$$\Im\{f(t) * h(t)\} = H(\mu)F(\mu)$$

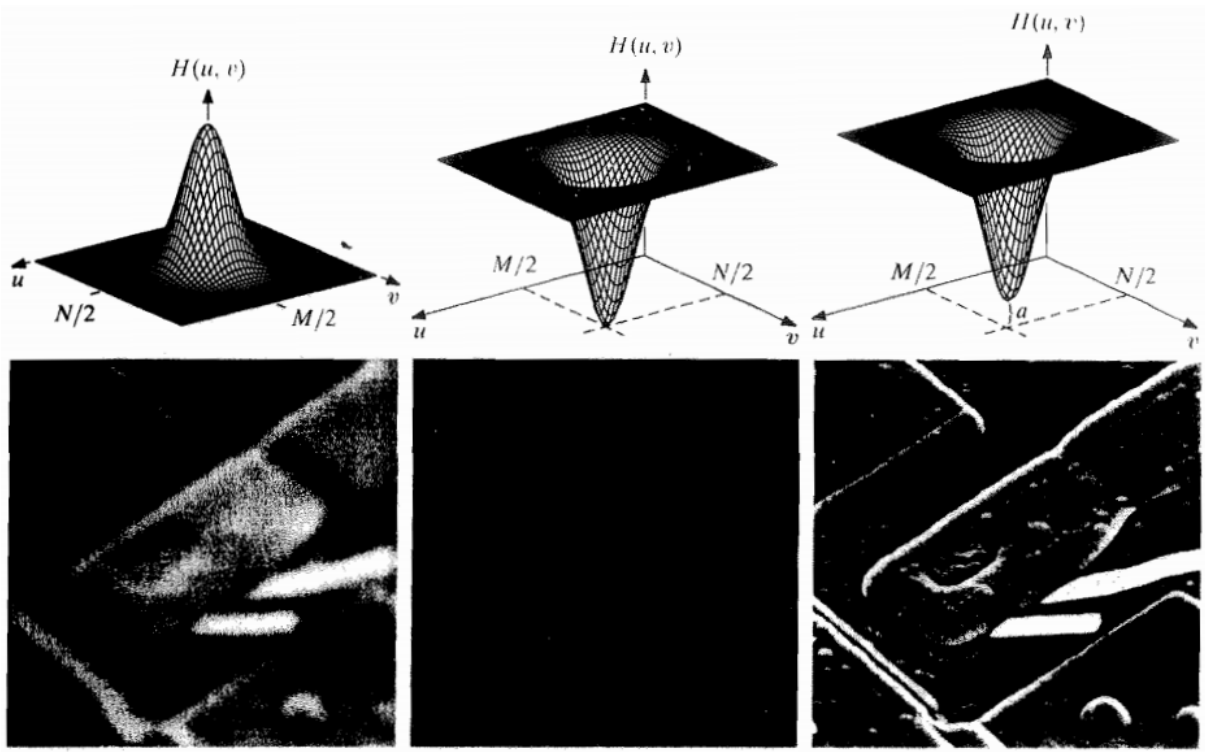
Áp dụng cho trường hợp 2 chiều, giả sử $f(x,y)$ là ảnh đầu vào, $h(x,y)$ là bộ lọc ảnh, $\Im\{f(x,y)\} = F(u,v), \Im\{h(x,y)\} = H(u,v)$, khi đó:

$$\Im\{f(x,y) * h(x,y)\} = H(u,v)F(u,v) \quad (1.4-11)$$

1.5 Lọc ảnh trong miền tần số

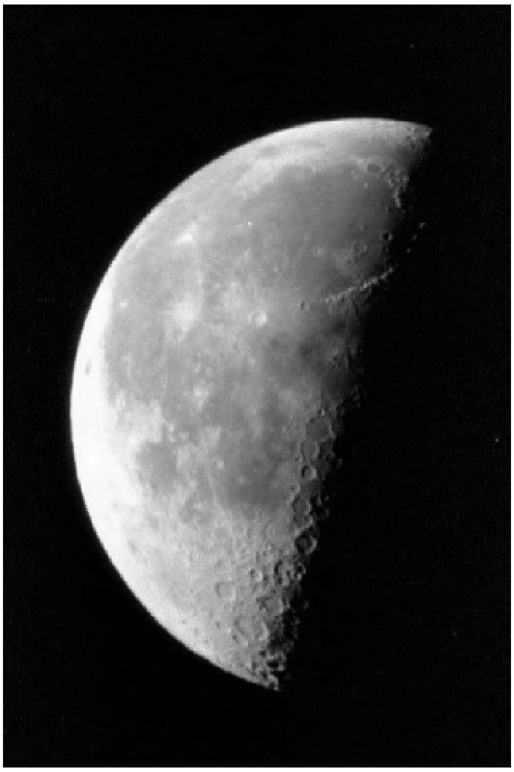
1.5.1 Khái niệm

Khi chụp ảnh, ta có thể thu được các ảnh có tần số thấp (sự thay đổi mức xám của ảnh ít, ví dụ như ảnh một bức tường) hay ảnh có tần số cao (ví dụ như biên của vật thể). Vì vậy, ta cần có bộ lọc $H(u,v)$ có thể làm giảm đi tần số cao trong khi đi qua các tần số thấp (gọi là lọc thông thấp) làm cho ảnh mờ đi. Ngược lại, ta cần có bộ lọc tó tính chất ngược với lọc thông thấp (gọi là lọc thông cao) giúp tăng cường chi tiết hình dạng vật thể, đồng thời làm giảm độ tương phản ảnh. Hình ảnh sau mô tả điều này



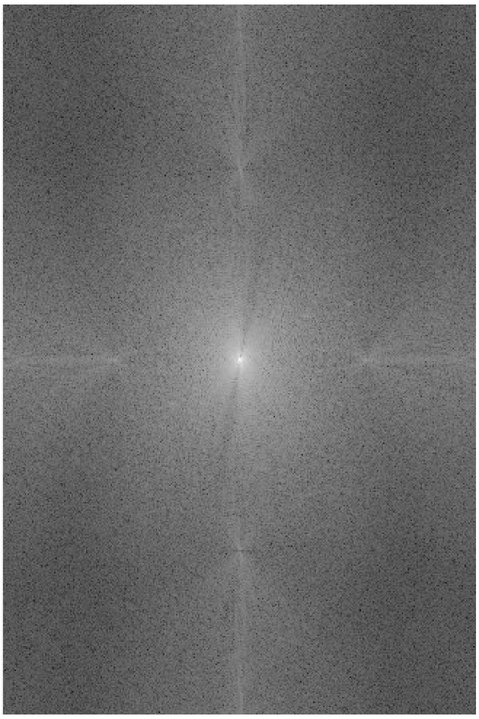
Hình 1.5 – 1. Ảnh trên là hàm đồ thị tần số ứng với ảnh bên dưới

Ta hãy xem một ảnh trong miền tần số trông như thế nào, ta quan sát ảnh sau:



Hình 1.5 – 2. Ảnh Mặt Trăng

Sử dụng hàm `fft2` trong Matlab, ta sẽ biểu diễn ảnh trên trong miền tần số là



Hình 1.5 – 3. Ảnh Mặt Trăng khi chuyển qua miền tần số

Định lý tích chập cho ta mối quan hệ giữa miền không gian và miền tần số, cụ thể, thông qua tích chập, một ảnh trong miền không gian có thể chuyển qua miền tần số và ngược lại.

Quy trình lọc ảnh trong miền tần số như sau:

Ảnh → Chuyển đổi Fourier → Lọc → Chuyển đổi Fourier ngược → Ảnh

- Bước 1: Xử lý ảnh trong miền không gian, tức tăng hoặc giảm độ sáng của ảnh.

- Bước 2: Lấy DFT của ảnh.

- Bước 3: Canh giữa DFT, tức mang DFT từ góc ảnh ra giữa ảnh, trong Matlab ta sử dụng hàm shiftfft để thực hiện.

- Bước 4: Thực hiện tích chập với hàm lọc.

- Bước 5: Trược DFT từ giữa ảnh ra góc.

- Bước 6: Lấy chuyển đổi ngược IDFT, tức chuyển ảnh từ miền tần số sang miền không gian.

1.5.2 Các bộ lọc

Khái niệm bộ lọc trong miền tần số tương tự như khái niệm mặt nạ trong miền không gian.

Sau khi chuyển ảnh sang miền không gian, ta áp dụng một số bộ lọc trong quy trình lọc ảnh nhằm làm mờ ảnh, giảm nhiễu, làm nét ảnh.

Các bộ lọc ảnh thông dụng:

- Lọc thông thấp Ideal (Ideal low pass filter).

- Lọc thông thấp Gauss (Gaussian low pass filter).

- Lọc thông thấp Butterworth (Butterworth lowpass filters).

- Lọc thông cao Ideal (Ideal high pass filter).

- Lọc thông cao Gauss (Gaussian high pass filter).

- Lọc thông cao Butterworth (Butterworth highpass filter).

Ta sẽ đi qua một số ứng dụng sử dụng các bộ lọc này

1.5.3 Làm mờ ảnh trong miền tần số

Ta có thể làm mờ ảnh bằng cách giảm các tần số cao, tức ta sử dụng phép lọc thông thấp.

Phép mờ ảnh có những tính chất sau:

- Tất cả giá trị trong mặt nạ mờ đều dương.

- Tổng các giá trị bằng 1.

- Làm giảm phần cạnh bằng mặt nạ mờ.

- Khi kích thước mặt nạ tăng, ảnh càng mượt.

1.5.3.1 Lọc thông thấp Ideal

Là phép lọc 2 chiều đi qua tất cả tần số mà không làm giảm chúng trong bán kính đường tròn D_0 tính từ tâm phép lọc và "chặt cụt" tất cả tần số bên ngoài hình tròn này.

Bộ lọc này xác định như sau:

$$H(u,v) = \begin{cases} 1 & \text{nếu } D(u,v) \leq D_0 \\ 0 & \text{nếu } D(u,v) > D_0 \end{cases} \quad (1.5-1)$$

với D_0 là hằng số dương và $D(u,v)$ là khoảng cách giữa điểm (u,v) trong miền tần số và tâm của hình chữ nhật tần số, tức:

$$D(u,v) = \left[\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2\right]^{1/2} \quad (1.5-2)$$

với P và Q là kích thước ảnh.

Bây giờ ta sẽ lọc ảnh sau:

Hình 1.5 – 4. Ảnh Mặt Trăng

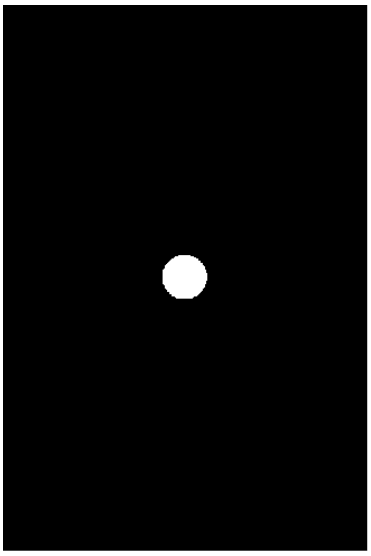
Ta tạo bộ lọc thông thấp Ideal có kích thước như ảnh trên, sử dụng công thức (1.5 – 2) với $D_0 = 1$.

Thuật toán 1.5 – 1. Lọc thông thấp Ideal

```
Input: ảnh cần lọc
Output: ảnh sau khi dùng bộ lọc thông thấp Ideal
%Khai báo ảnh vào, biến kiểu double
f = imread('moon.jpg');
f = double(f);
P = size(f, 1);     %Tạo bộ lọc có kích thước bằng với ảnh
Q = size(f, 2);
```

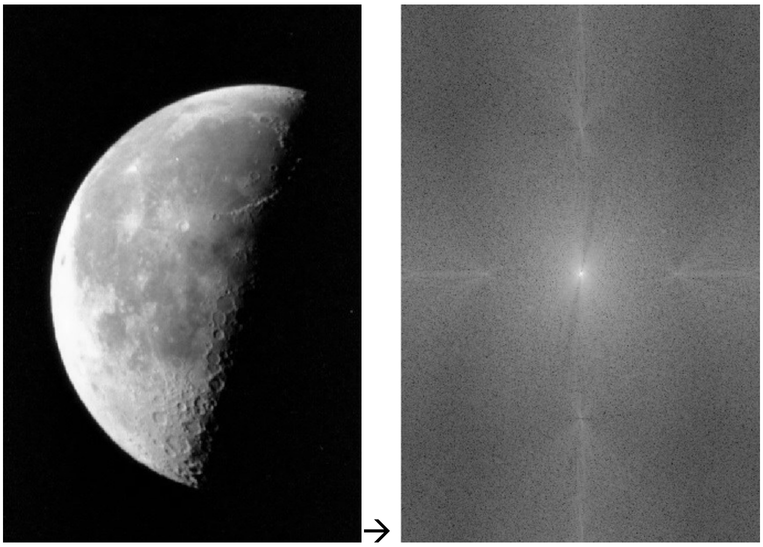
```
D0 = 30;      %Bán kính
h = zeros(P, Q);

for i = 1:P
    for j = 1:Q
        if (i - P/2)^2 + (j - Q/2)^2 <= D0^2
            h(i, j) = 1;
        end
    end
end
%Khai báo ảnh vào, biến kiểu double
f = imread('moon.jpg');
f = double(f);
P = size(f, 1);     %Tạo bộ lọc có kích thước bằng với ảnh
Q = size(f, 2);

D0 = 30;      %Bán kính
h = zeros(P, Q);

for i = 1:P
    for j = 1:Q
        if (i - P/2)^2 + (j - Q/2)^2 <= D0^2
            h(i, j) = 1;
        end
    end
end
```

Ảnh của bộ lọc thông thấp Ideal:

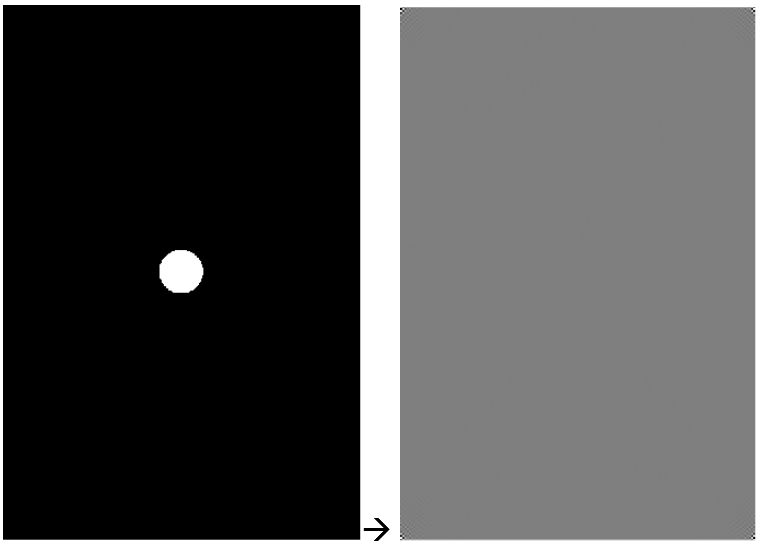


Hình 1.5 – 5. Ảnh của bộ lọc thông thấp Ideal trong miền không gian

Bây giờ ta chuyển ảnh ban đầu và bộ lọc qua miền tần số bằng hàm `fft2`.

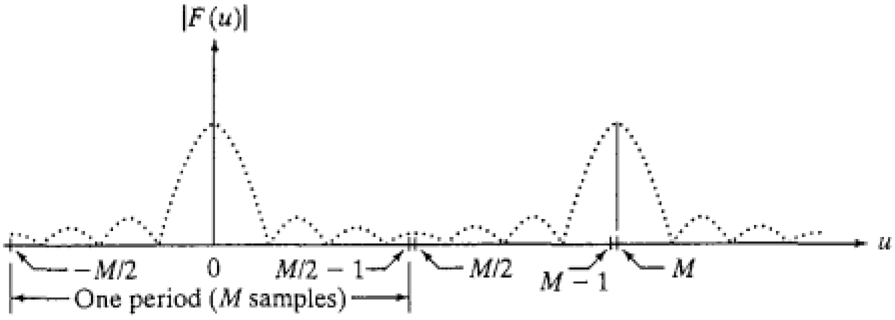


Hình 1.5 – 6. Chuyển ảnh Mặt Trăng từ miền không gian (ảnh trái)
sang miền tần số (ảnh phải)



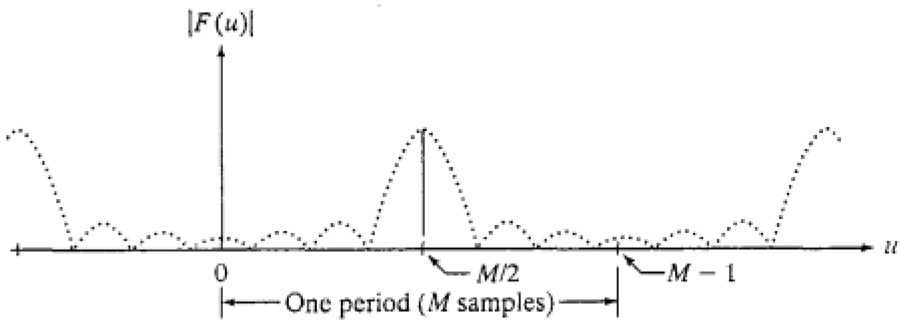
Hình 1.5 – 6. Chuyển ảnh (1.5 – 5) từ miền không gian (ảnh trái)
sang miền tần số (ảnh phải)

Ở hình ảnh chuyển đổi Fourier của bộ lọc, ở 4 góc hình có 4 đốm sáng do tính chất của chu kỳ



Hình 1.5 – 7. Chu kỳ tần số của ảnh, M là chiều rộng của ảnh, $F(u)$ là tần số của ảnh tại vị trí u

Chúng ta muốn sử dụng nguyên vẹn cả 1 chu kỳ, vì vậy ta cần "trượt" tâm chu kỳ vào giữa ảnh.



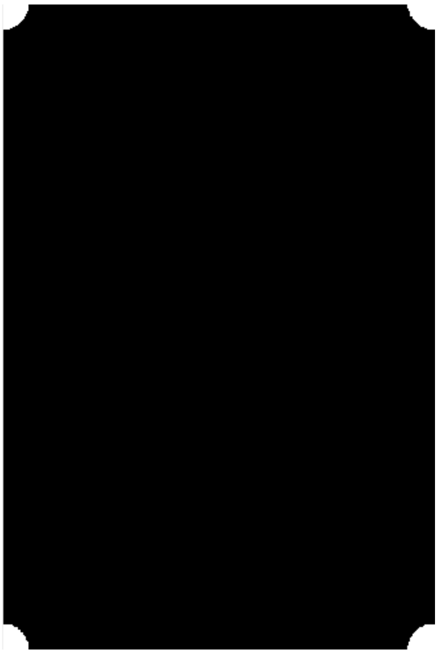
Hình 1.5 – 8. "Trượt" tâm chu kỳ (điểm cao nhất) vào giữa ảnh

Trong Matlab, lệnh trượt tâm chu kỳ về giữa ảnh là lệnh `fftshift`.

Thuật toán 1.5 – 2. Trượt tâm chu kỳ về giữa ảnh

```
Input: ảnh f và ảnh h trong miền tần số
Output: ảnh F và ảnh H trong miền không gian
F = fft2(f);       %Chuyển ảnh f qua miền tần số thông qua FFT
H = fftshift(h);   %Chuyển bộ lọc qua miền tần số, đưa tâm chu kỳ về
                   %giữa hình.
```

Khi đó, ảnh của bộ lọc là:



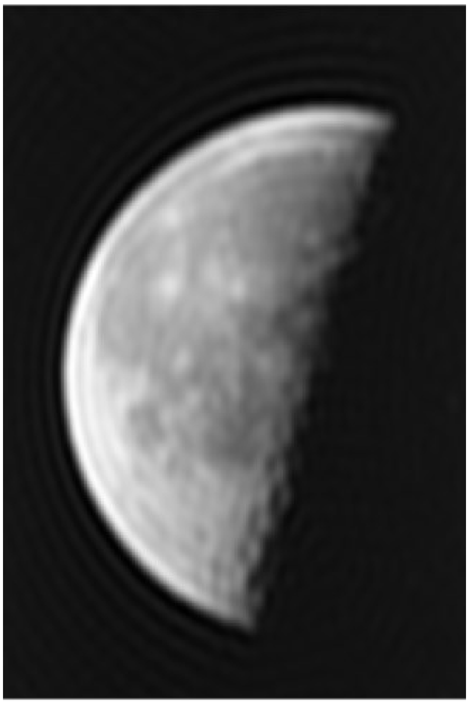
Hình 1.5 – 9. Ảnh bộ lọc trong miền tần số sau khi chuyển tâm về giữa hình

Ta thực hiện phép tính tích chập, sau đó thực hiện phép chuyển đổi ngược Fourier để thu về ảnh sau khi lọc.

Thuật toán 1.5 – 3. Lấy tích chập, dùng chuyển đổi Fourier chuyển ảnh từ miền tần số về miền không gian

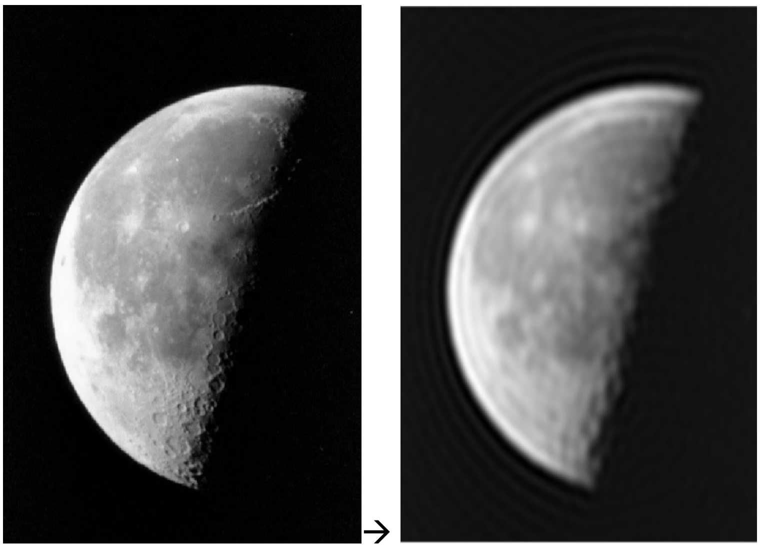
```
Input: ảnh H, ảnh F trong miền không gian
Output: ảnh g trong miền tần số
G = H.*F; %Thực hiện tích chập, H là ảnh trong miền tần số, F là bộ
          %lọc ảnh trong miền tần số, G là ảnh sau khi thực hiện phép
          %tích chập.
g = ifft2(G); %Biến đổi ngược ảnh G từ miền tần số sang ảnh g trong
              %miền không gian.
```

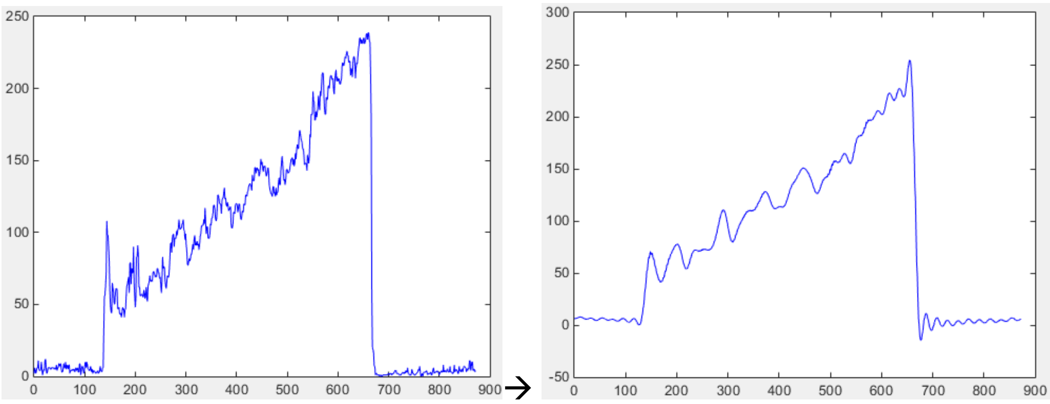
Ảnh thu được là



Hình 1.5 – 10. Ảnh Mặt Trăng được làm mờ khi sử dụng
lọc thông thấp Ideal

Như vậy, với phép lọc thông thấp Ideal, ta đã làm mờ ảnh, làm giảm các tần số cao, ví dụ như độ nét của biên mặt trăng sau khi lọc đã giảm xuống.





Hình 1.5 – 11. So sánh tần số của ảnh trước và sau khi lọc. Ảnh trên – trái: ảnh ban đầu. Ảnh trên – Phải: ảnh sau khi lọc. Ảnh dưới – trái: tần số ảnh ban đầu. Ảnh dưới – phải: tần số ảnh sau khi lọc.

Ảnh phổ (theo đường chéo trên – phải xuống dưới – trái) cho thấy sau khi lọc, tần số của ảnh đã được kéo giãn ra, điều đó có nghĩa nhiều tần số cao đã được kéo xuống.

Hãy để ý quanh biên của mặt trăng, ta thấy có một số đường biên mờ. Đây là hiệu ứng chuông, ta có thể giải thích hiệu ứng này thông qua định lý tích chập

Giả sử $F(u)$ là ảnh đầu vào, ảnh đầu ra là $G(u)$ và hàm lọc là $H(u)$ (xét trong miền tần số), định lý tích chập cho ta

$$G(u,v) = H(u,v)F(u,v) \quad (1.5-1)$$

Tích chập tương ứng trong miền không gian là

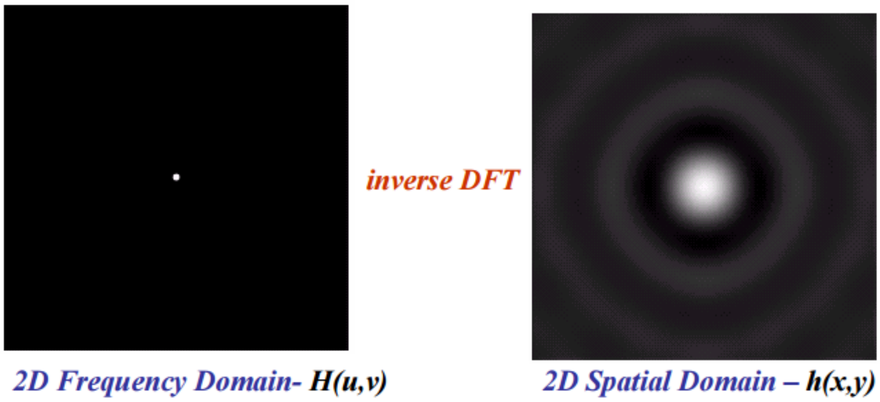
$$g(x,y) = h(x,y) * f(x,y)$$

$h(x,y)$ xét ở miền không gian, là chuyển đổi Fourier ngược của hàm lọc $H(u,v)$.

Hàm lọc trong miền không gian $h(x,y)$ có 2 đặc điểm chính

+ Phần trội nằm ở gốc.

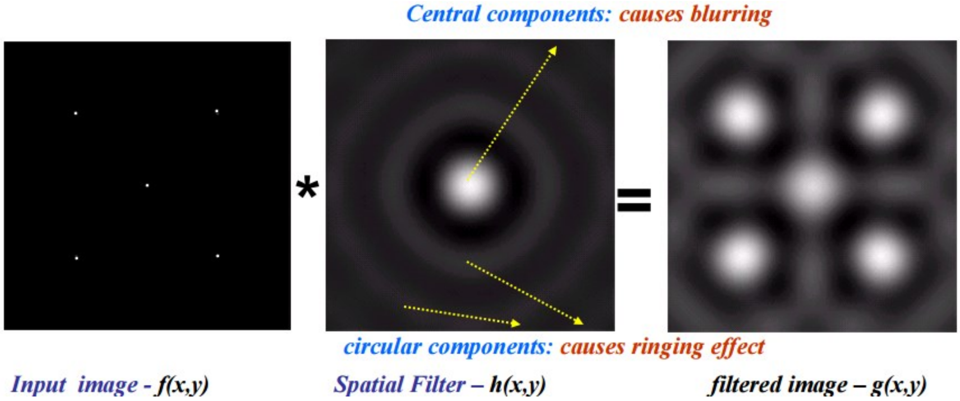
+ Các phần tròn đồng tâm nằm ở phần trung tâm



Hình 1.5 – 11. Ảnh trái: ảnh trong miền tần số. Ảnh phải: ảnh trong miền không gian khi áp dụng IDFT của ảnh bên trái

Phần trung tâm của lọc trong miền không gian dùng để làm mờ.

Phần tròn đồng tâm gây ra hiệu ứng chuông.



Hình 1.5 – 12. Mô tả hiệu ứng chuông khi sử dụng phép tích chập trong ảnh. Ảnh trái: ảnh đầu vào. Ảnh giữa: bộ lọc trong miền không gian có những vòng tròn đồng tâm. Ảnh phải: kết quả ảnh sau khi lọc, xuất hiện hiệu ứng chuông

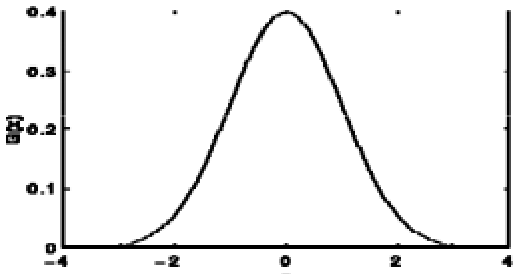
Do xảy ra hiệu ứng chuông nên trong thực tế, người ta ít dùng đến lọc thông thấp Ideal mà thay vào đó người ta sẽ dùng những bộ lọc khác có khả năng loại bỏ hiệu ứng này.

1.5.3.2 Lọc thông thấp Gauss

Đặc trưng cho nhiễu đó là hàm mật độ xác suất thể hiện sự phân bố của nhiễu. Ta sử dụng hàm phân phối Gauss làm bộ lọc nhằm làm mờ ảnh và giảm nhiễu. Trong trường hợp 1 chiều, phân phối Gauss có công thức:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

với σ là độ lệch chuẩn của phân phối, Ta giả sử phân phối này có trung bình là 0.



Hình 1.5 – 13: Đồ thị phân phối Gauss

Khi xử lý ảnh, ta sẽ sử dụng hàm phân phối Gauss cho 2 chiều, hình thành bằng tích của 2 hàm Gauss 1 chiều x và y.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Bộ lọc thông thấp Gauss có dạng:

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{D^2(u,v)}{2\sigma^2}} \quad (1.5-2)$$

với $D(u,v)$ là khoảng cách từ điểm (u,v) đến tâm hình.

Ta sẽ xử lý ảnh sau với phép lọc thông thấp Gauss

Hình 1.5 – 14: Ảnh Mặt Trăng

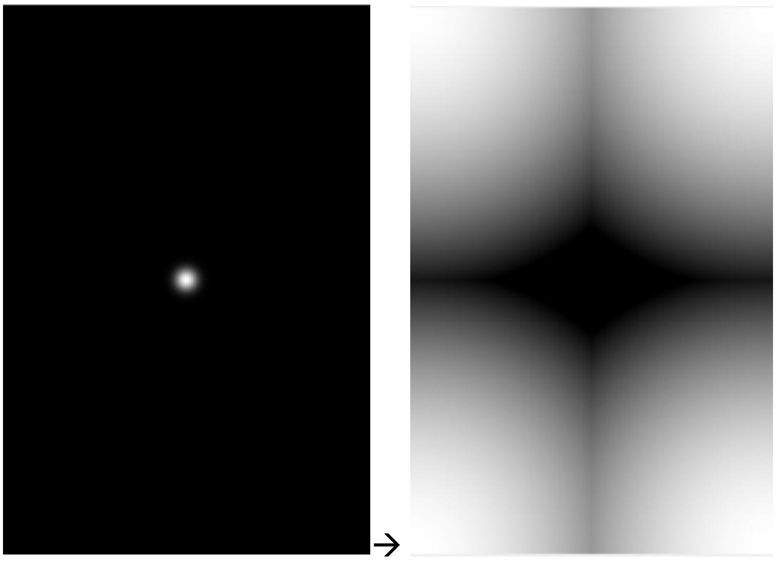
Khởi tạo bộ lọc

Thuật toán 1.5 – 4. Lọc thông thấp Gauss

```
Input: ảnh cần lọc
Output: ảnh sau khi dùng bộ lọc thông thấp Gauss
P = size(f, 1);     %Tạo bộ lọc có kích thước bằng với ảnh.
Q = size(f, 2);
h = zeros(P, Q);

```

```
sig = 10;  %Gán σ = 10
a = 1/(2.*pi.*sig);
b = 2.*sig.*sig;

for i = 1:P
    for j = 1:Q
        D = (i - P./2).^2 + (j - Q./2).^2; %K/c (x,y)->tâm hình, lấy
                                           %mũ 2
        h(i, j) = a*exp(-D./b);
    end
end
```

Ảnh của bộ lọc ở miền không gian chuyển sang miền tần số là



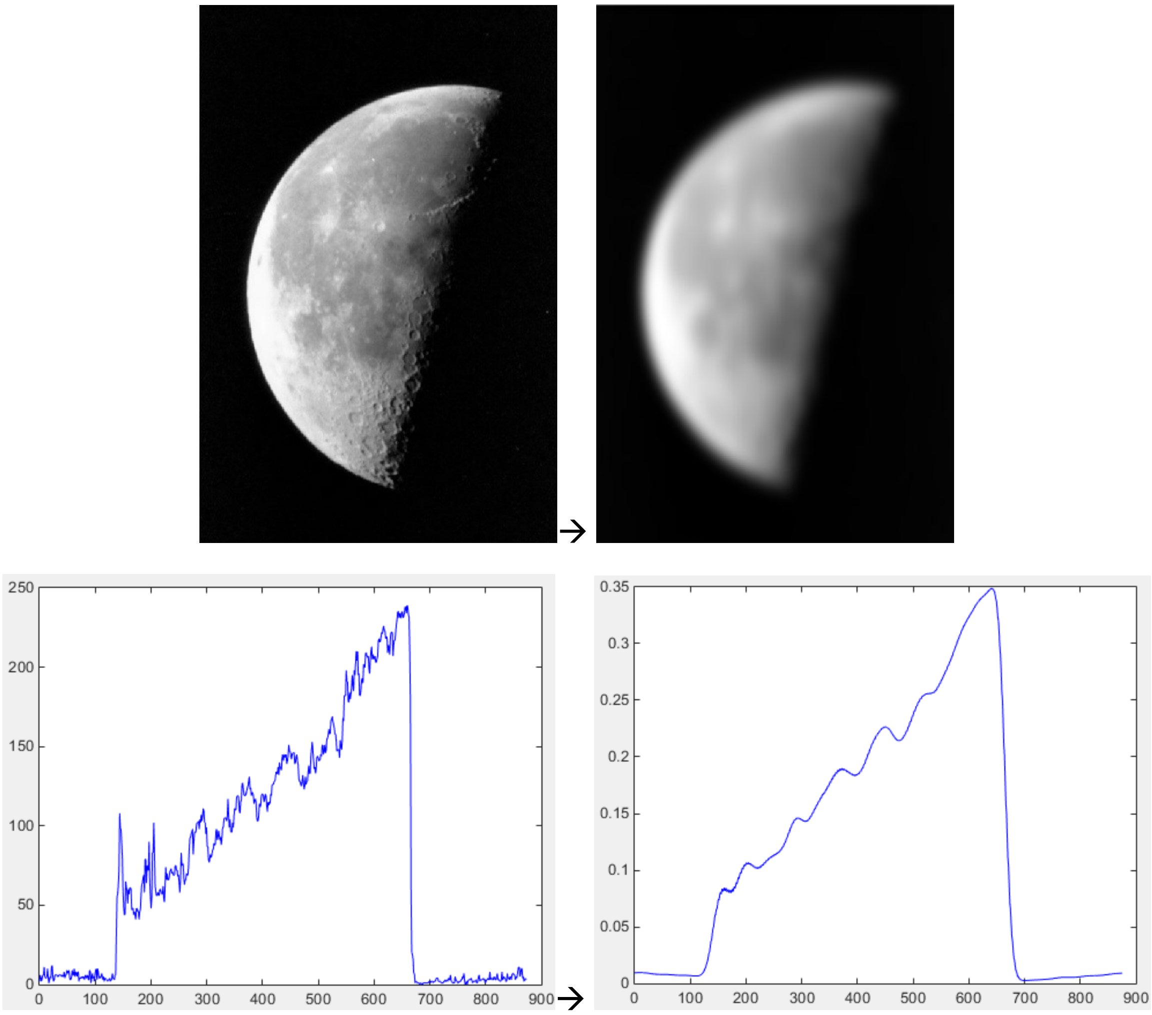
Hình 1.5 – 15. Ảnh trái: ảnh bộ lọc thông thấp Gauss. Ảnh phải: ảnh của bộ lọc sau khi chuyển qua miền tần số

Sử dụng thuật toán 1.5 – 3, ta tính tích chập của ảnh gốc trong miền tần số và bộ lọc trong miền tần số, sau đó lấy chuyển đổi ngược.

Ta được ảnh sau khi lọc là:

Hình 1.5 – 16. Ảnh Mặt Trăng sau khi lọc sử dụng lọc thông thấp Gauss

Ta quan sát phổ của ảnh ban đầu và sau khi lọc (hướng trên – phải xuống dưới – trái)



Hình 1.5 – 17. So sánh tần số của ảnh trước và sau khi lọc. Ảnh trên – trái: ảnh ban đầu. Ảnh trên – Phải: ảnh sau khi lọc. Ảnh dưới – trái: tần số ảnh ban đầu. Ảnh dưới – phải: tần số ảnh sau khi lọc.

Ta thấy rằng sau khi lọc, độ biến thiên tần số của ảnh ít hơn ảnh gốc, đồng thời các giá trị tần số ảnh sau khi lọc cao hơn ảnh gốc.

Ảnh sau khi lọc không xảy ra hiệu ứng chuông như phép lọc thông thấp Ideal. Phép lọc thông thấp Gauss này cho ảnh mượt hơn.

1.5.3.3 Lọc thông thấp Butterworth

Bộ lọc này bao gồm các tính chất của lọc thông thấp Ideal và lọc thông thấp Gauss. Bộ lọc thông thấp Butterworth có dạng sau:

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}} \quad (1.5-3)$$

Trong đó:

- $D(u,v)$ là khoảng cách từ tâm ảnh đến điểm (u,v) (giống như trong lọc thông thấp Ideal).

- D_0 là tần số chặt cụt.

- n là cấp của bộ lọc.

Từ dạng của bộ lọc, ta có những tính chất sau

- Với $D(u,v) \ll D_0, H \approx 1$

- Với $D(u,v) \gg D_0, H \approx 0$

- Với $D(u,v) = D_0, H = 1/2$

Bây giờ, ta sử dụng bộ lọc này để lọc ảnh Mặt Trăng

Hình 1.5 – 18. Ảnh Mặt Trăng

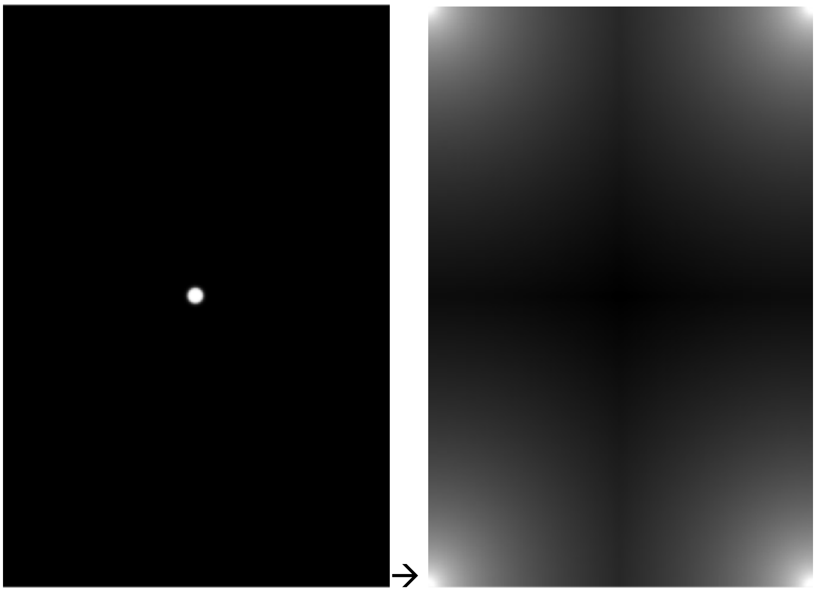
Khởi tạo bộ lọc có kích thước $P \times Q$ (bằng với kích thước hình ban đầu)

Thuật toán 1.5 – 4. Lọc thông thấp Butterworth

```
Input: ảnh cần lọc
Output: ảnh sau khi dùng bộ lọc thông thấp Butterworth
h = zeros(P, Q);
D0 = 10; %Chọn tần số chặt cụt
n = 5; %Chọn cấp

for i = 1:P
    for j = 1:Q
        u = sqrt((i - P/2)^2 + (j - Q/2)^2);
        h(i,j)= 1/(1 + (u/D0).^(2.*n));
    end
end
h = zeros(P, Q);
D0 = 10; %Chọn tần số chặt cụt
n = 5; %Chọn cấp

for i = 1:P
    for j = 1:Q
```

```
        u = sqrt((i - P/2)^2 + (j - Q/2)^2);
        h(i,j)= 1/(1 + (u/D0).^(2.*n));
    end
end
```

Khi đó, ta được ảnh bộ lọc trong miền không gian và miền tần số là



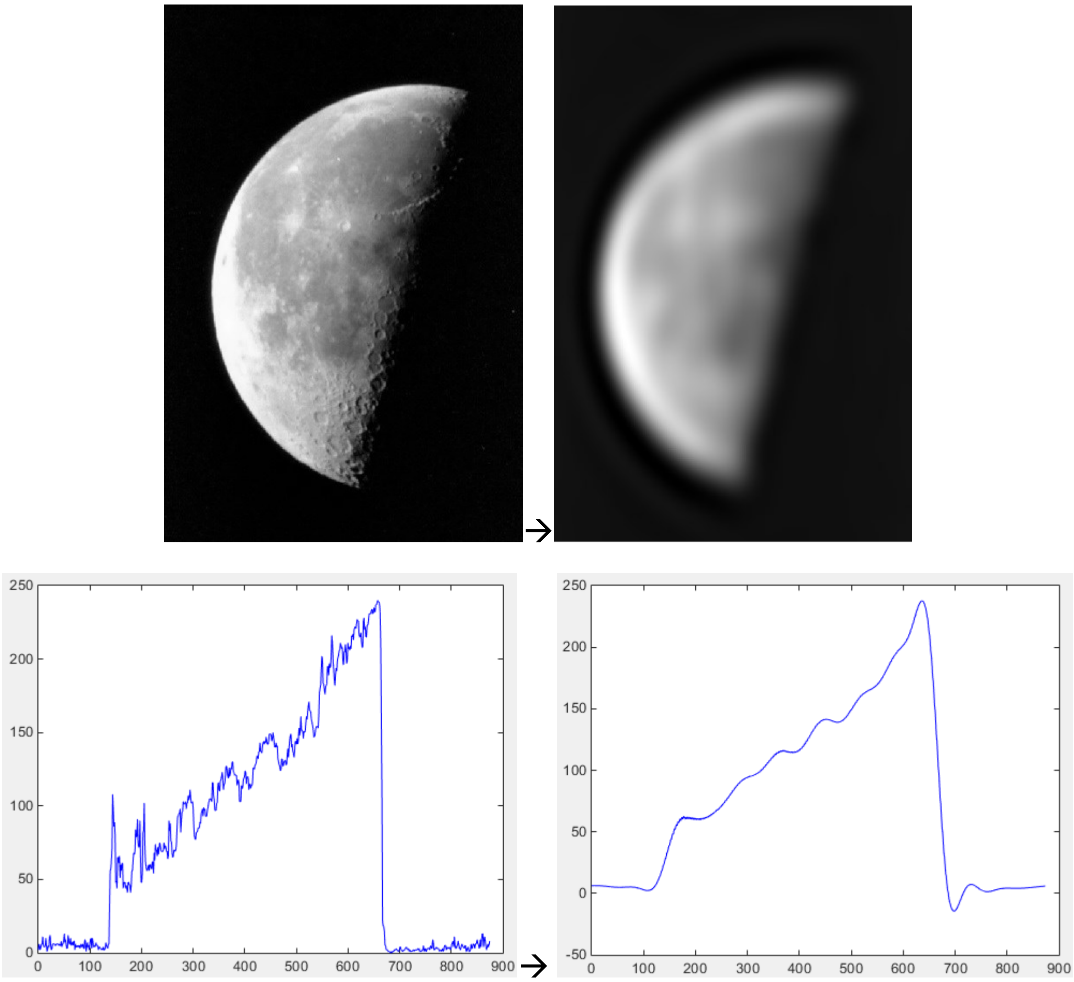
Hình 1.5 – 19. Bộ lọc Butterworth trong miền không gian (bên trái) chuyển sang miền tần số (bên phải)

Sử dụng thuật toán 1.5 – 3, ta tính tích chập của ảnh gốc trong miền tần số và bộ lọc trong miền tần số, sau đó lấy chuyển đổi ngược.

Ta được ảnh sau khi lọc là:

Hình 1.5 – 20. Ảnh Mặt Trăng được làm mờ bằng bộ lọc Butterworth

Ta quan sát phổ của ảnh gốc và sau khi lọc



Hình 1.5 – 21. So sánh tần số của ảnh trước và sau khi lọc. Ảnh trên – trái: ảnh ban đầu. Ảnh trên – Phải: ảnh sau khi lọc. Ảnh dưới – trái: tần số ảnh ban đầu. Ảnh dưới – phải: tần số ảnh sau khi lọc.

Với bộ lọc này, ảnh sau khi lọc có độ biến thiên tần số ít hơn, không làm tăng giá trị tần số, hiệu ứng chuông ảnh hưởng không đáng kể.

1.5.4 Ứng dụng phép lọc làm mờ ảnh

Phép lọc này có nhiều ứng dụng như:

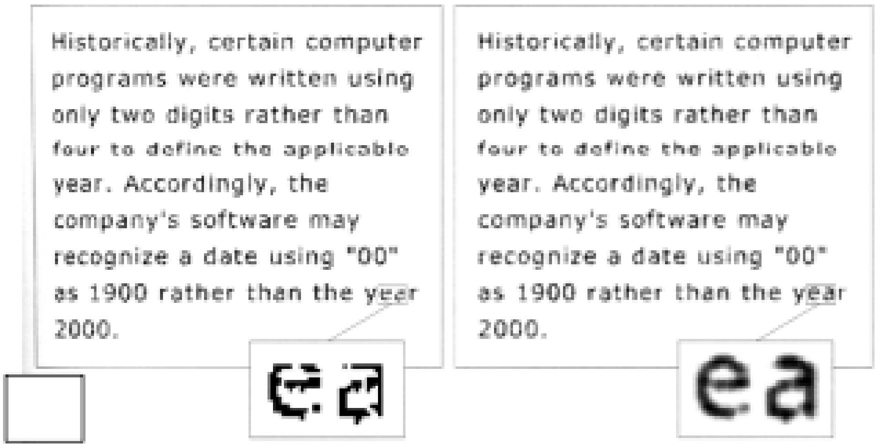
- Nhận dạng ký tự trong tri thức máy.

- Dùng trong công nghiệp in ấn.

- Xử lý ảnh trên vệ tinh hoặc trên không trung.

-…

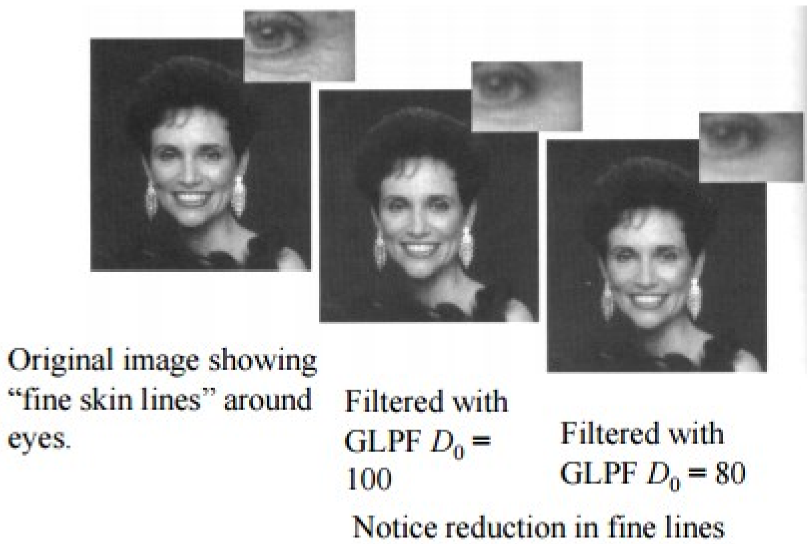
Hình ảnh dưới đây cho thấy một đoạn văn có độ phân giải thấp, một vài ký tự bị đứt gãy. Sử dụng phép lọc thông thấp Gauss, ta được ảnh đoạn văn rõ nét hơn.



Hình 1.5 – 22. Ảnh trái: đoạn văn có ký tự "ea" bị gãy. Ảnh phải: ký tự "ea" liền nét bằng bộ lọc thông thấp Gauss

Mặc dù mắt người dễ dàng nhận diện nét chữ đứt gãy nhưng hệ thống nhận dạng của máy thì không thấy vậy. Một cách giải quyết đó là ta bắc cầu các mảnh nhỏ lại với nhau bằng cách làm mờ chúng. Hình bên phải cho thấy các ký tự rất nét bằng bộ lọc thông thấp Gauss với $D_0 = 80$.

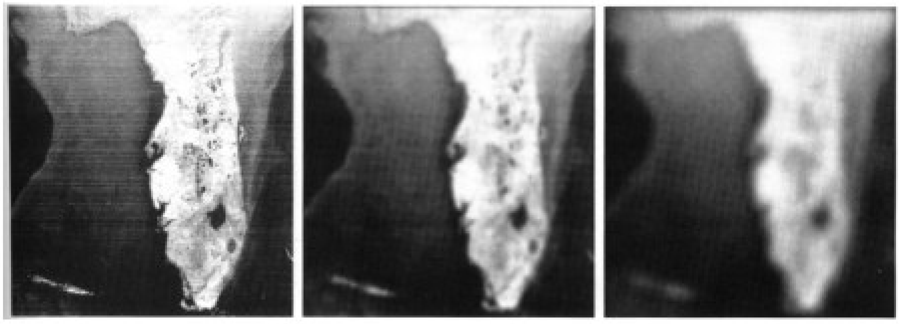
Lọc thông thấp còn được áp dụng trong công nghiệp in ấn với nhiều hàm số dùng để "sơ chế" ảnh, bao gồm ảnh chưa xác định rõ hình dạng. Hình ảnh dưới đây là một ứng dụng của lọc thông thấp, giúp ảnh mượt hơn, dễ nhìn hơn.



Hình 1.5 – 23. Áp dụng lọc thông thấp để làm mắt trông mượt hơn (từ trái sang phải)

Với ảnh người, mục tiêu hướng đến là làm giảm độ nét cửa đường da mảnh và các nhược điểm nhỏ. Trong hình trên, ta đã làm giảm độ nét đường da quanh mắt, giúp cho ảnh nhìn mượt mà hơn.

Ảnh sau cho ta 2 ứng dụng của lọc thông thấp trên cùng một hình nhưng với các đối tượng khác nhau. Ảnh bên trái có độ phân giải bức xạ rất cao, đó là ảnh Vịnh Mexico (tối) và Florida (sáng) do vệ tinh NOAA chụp lại. Biên của các vật thể trong nước được chụp bởi vòng lặp dòng. Ảnh này dùng để minh họa ảnh viễn thám với cảm biến có xu hướng đưa ra các dòng quét rõ ràng theo hướng quét cảnh.



Original image: notice horizontal scan-lines

Smoothed with a GLPF with $D_0 = 50$.

Smoothed with a GLPF with $D_0 = 20$.

Hình 1.5 – 24. Sử dụng lọc thông thấp để giảm độ phân giải bức xạ (từ trái sang phải)

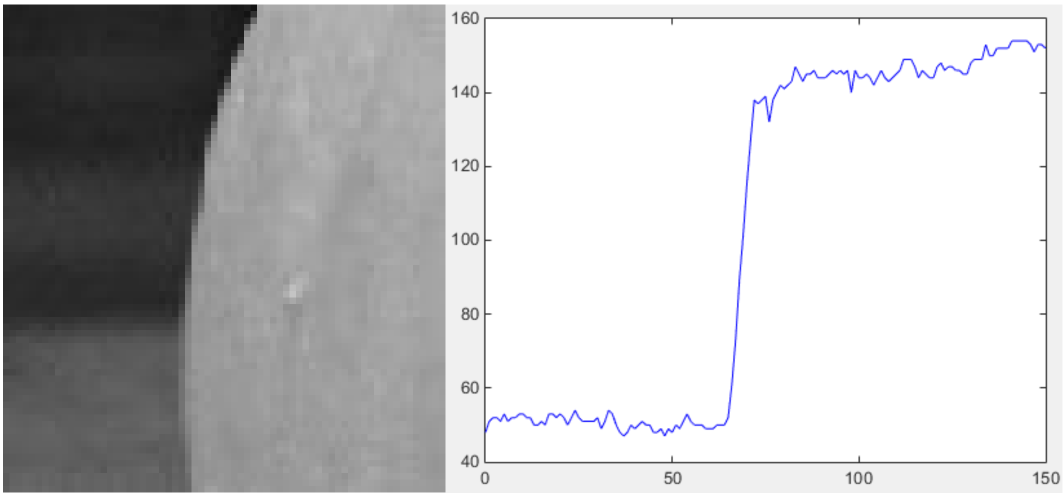
Lọc thông thấp cho kết quả thô nhưng đây là cách đơn giản để làm giảm hiệu ứng đường sọc ngang này. Hình giữa sử dụng lọc thông thấ Gauss với $D_0 = 50$. Làm giảm các hiệu ứng đường quét này giúp đơn giản hóa việc phát hiện các đặc tính như các biên gữa dòng nước biển.

Ảnh bên phải là kết quả của phép lọc thông thấp Gauss với $D_0 = 20$. Ảnh này cho thấy vật thể bị làm mờ nhiều chi tiết nhất có thể, để lại các đặc tính nhận dạng lớn. Cụ thể, cách lọc này là một phần của bước chuẩn bị cho hệ thống phân tích hình ảnh thầm tìm kiếm các đặc tính ở rìa ảnh.

Phép lọc thông thấp giúp đơn giản hóa khi phân tích bằng cách tính trung bình các đặc tính nhỏ hơn đặc tính mong muốn.

1.5.5 Lọc sắc ảnh

Ta có thể làm mờ ảnh bằng cách làm giảm các tần số cao khi sử dụng chuyển đổi Fourier. Bây giờ, để làm sắc các biên ảnh, ta sẽ lọc các tần số cao và làm giảm các phần có tần số thấp bởi vì đường biên vật thể trong ảnh có độ biến thiên tần số lớn, ứng với các tần số cao.



Hình 1.5 – 25. Ảnh phải là đồ thị tần số của ảnh trái khi đi qua đường biên vật thể

Khi đi qua đường biên, độ biến thiên tần số lớn.

Cho bộ lọc thông thấp, ta thu được bộ lọc thông cao bằng công thức:

$$H_{HP}(u,v) = 1 - H_{LP}(u,v) \quad (1.5-4)$$

với $H_{LP}(u,v)$ là hàm chuyển đổi lọc thông thấp. Công thức này có nghĩa rằng khi lọc thông thấp làm giảm tần số thì lọc thông cao bỏ qua điều đó và ngược lại.

Bây giờ, ta sẽ tìm hiểu một số bộ lọc thông cao.

1.5.5.1 Lọc thông cao Ideal

Lọc thông cao Ideal có dạng như sau:

$$H(u,v) = \begin{cases} 0 \text{ nếu } D(u,v) \le D_0 \\ 1 \text{ nếu } D(u,v) > D_0 \end{cases} \quad (1.5-5)$$

Ta thấy rằng dạng của lọc thông cao Ideal ngược với dạng của lọc thông thấp Ideal. Ta dùng phép lọc này để lọc hình sau:

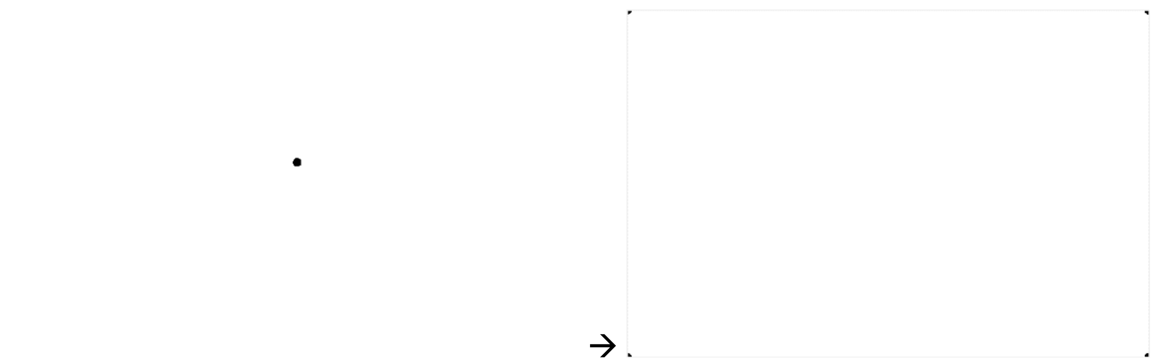
Hình 1.5 – 26. Hình quả bí ngô

Khởi tạo bộ lọc

Thuật toán 1.5 – 5. Lọc thông cao Ideal

```
Input: ảnh cần lọc
Output: ảnh sau khi dùng bộ lọc thông cao Ideal
h = zeros(P, Q); %Tạo ma trận 0 có kích thước P×Q bằng với kích
                 %thước ảnh
D0 = 30; %Bán kính vòng tròn
for i = 1:P
    for j = 1:Q
        if (i - P/2)^2 + (j - Q/2)^2 <= D0^2
            h(i, j) = 0;
        else
            h(i, j) = 1;
        end
    end
end
H = fftshift(h); %Chuyển bộ lọc sang miền tần số
h = zeros(P, Q); %Tạo ma trận 0 có kích thước P×Q bằng với kích
                 %thước ảnh
D0 = 30; %Bán kính vòng tròn
for i = 1:P
    for j = 1:Q
        if (i - P/2)^2 + (j - Q/2)^2 <= D0^2
            h(i, j) = 0;
        else
            h(i, j) = 1;
        end
    end
end
```

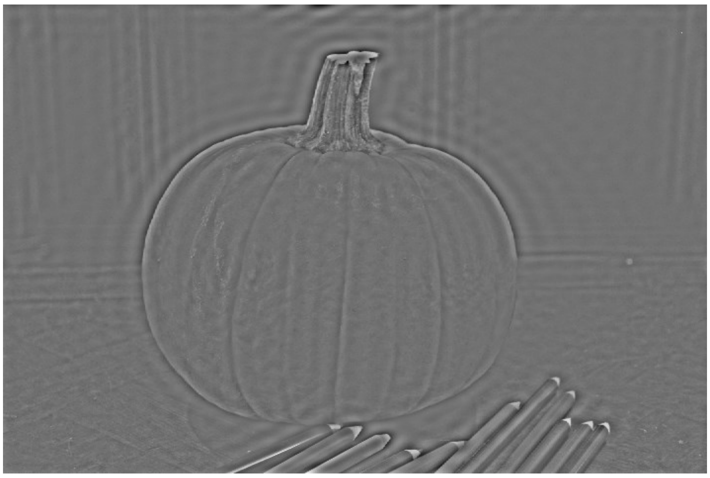
Ta được hình ảnh bộ lọc từ miền không gian sang mình tần số như sau:



Hình 1.5 – 27. Ảnh bộ lọc thông cao Ideal trong miền không gian (ảnh trái) chuyển san miền tần số (ảnh phải)

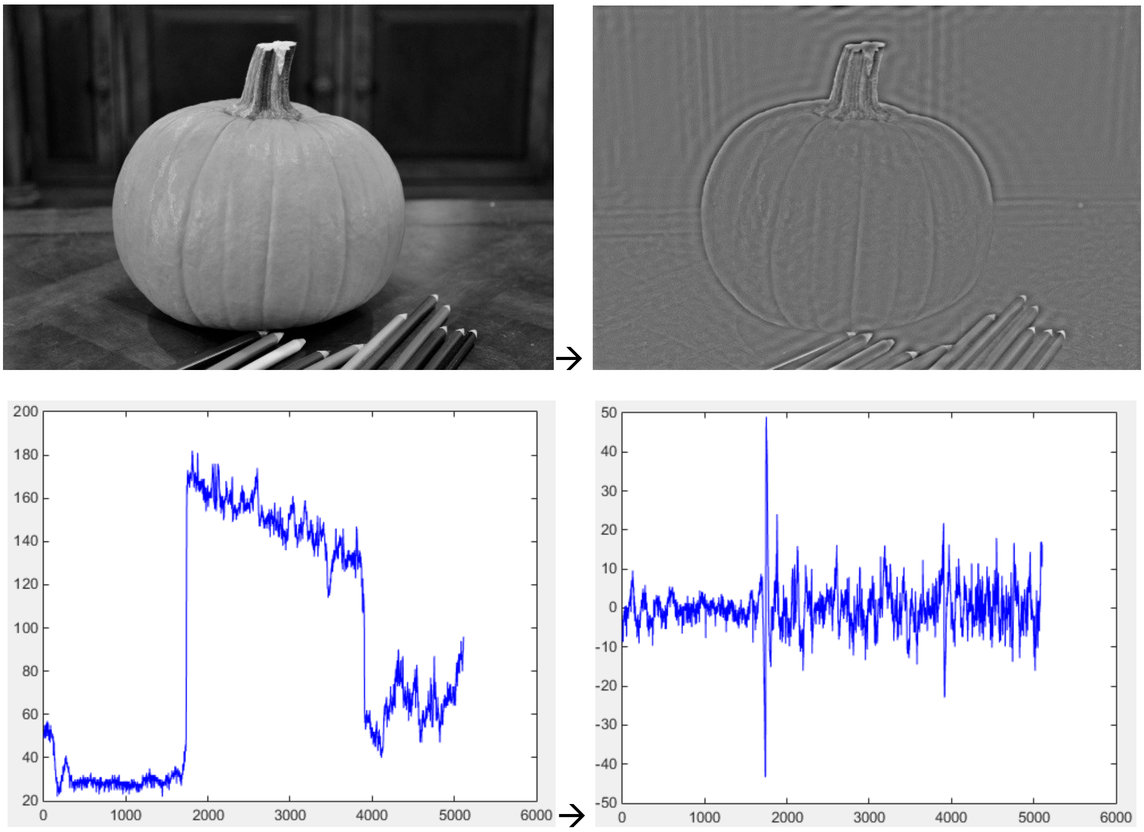
Sử dụng thuật toán 1.5 – 3, ta tính tích chập của ảnh gốc trong miền tần số và bộ lọc trong miền tần số, sau đó lấy chuyển đổi ngược.

Ta được ảnh sau khi lọc là:



Hình 1.5 – 28. Ảnh quả bí ngô sau khi lọc bằng lọc thông cao Ideal

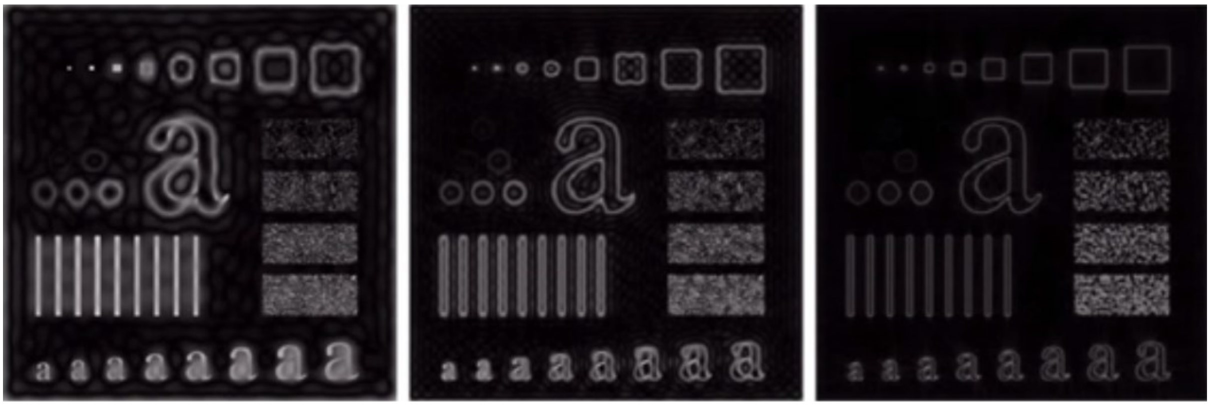
Nhìn vào ảnh, ta thấy rằng phép lọc này gây ra hiệu ứng chuông. Ta quan sát phổ của ảnh gốc và ảnh sau khi lọc (trên – phải xuống dưới – trái)



Hình 1.5 – 29. So sánh tần số của ảnh trước và sau khi lọc. Ảnh trên – trái: ảnh ban đầu. Ảnh trên – Phải: ảnh sau khi lọc. Ảnh dưới – trái: tần số ảnh ban đầu. Ảnh dưới – phải: tần số ảnh sau khi lọc.

Phép lọc này biến các giá trị tần số khác biên trở nên gần như nhau, đồng thời làm nổi bật tần số ở biên quả bí ngô.

Ảnh sau đây minh họa hiệu ứng chuông khi sử dụng phép lọc thông cao Ideal với giá trị D_0 (từ trái sang) lần lượt là 15, 30 và 80



Hình 1.5 – 30. Hiệu ứng chuông trong ảnh sử dụng bộ lọc thông cao Ideal với bán kính từ trái sang là 15, 30, 80

1.5.5.2 Lọc thông cao Gauss

Phép lọc thông cao Gauss 2 chiều có dạng:

$$G(u,v) = 1 - e^{-\frac{D^2(u,v)}{2\sigma^2}} \quad (1.5-6)$$

Ta dùng phép lọc này để lọc ảnh quả bí ngô sau

Hình 1.5 – 31. Ảnh quả bí ngô

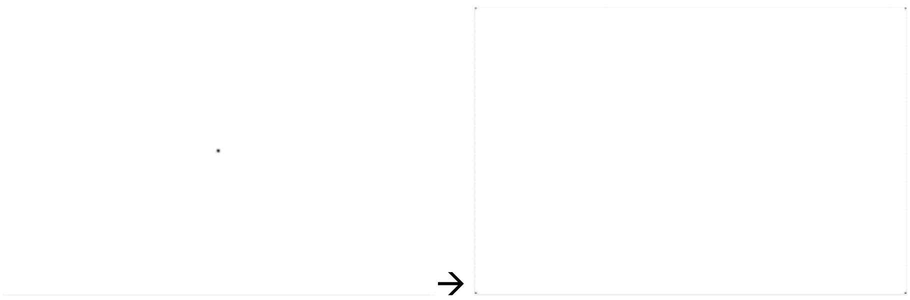
Khởi tạo bộ lọc trong Matlab

Thuật toán 1.5 – 6. Lọc thông cao Gauss

```
Input: ảnh cần lọc
Output: ảnh sau khi dùng bộ lọc thông cao Gauss
P = size(f, 1);     %Tạo bộ lọc có kích thước bằng với ảnh
Q = size(f, 2);
h = zeros(P, Q);

sig = 40;  %Gán σ
b = 2.*sig.*sig;

for i = 1:P
```

```
        for j = 1:Q
            D = (i - P./2).^2 + (j - Q./2).^2;
            h(i, j) = 1 - exp(-D./b);
        end
    end
    H = fftshift(h); %Chuyển bộ lọc sang miền tần số
    P = size(f, 1);    %Tạo bộ lọc có kích thước bằng với ảnh
    Q = size(f, 2);
    h = zeros(P, Q);

    sig = 40;  %Gán σ
    b = 2.*sig.*sig;

    for i = 1:P
        for j = 1:Q
            D = (i - P./2).^2 + (j - Q./2).^2;
            h(i, j) = 1 - exp(-D./b);
        end
    end
    H = fftshift(h); %Chuyển bộ lọc sang miền tần số
```

Ta được ảnh bộ lọc trong miền không gian và miền tần số như sau

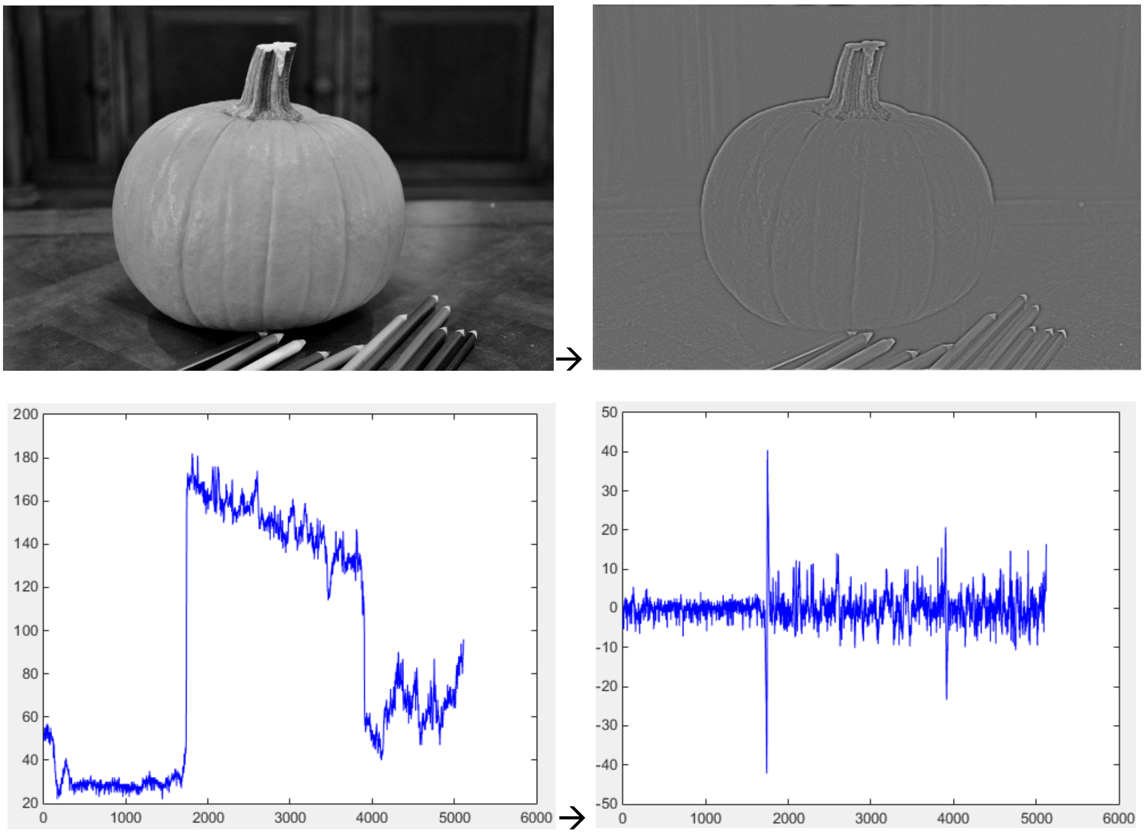


Hình 1.5 – 32. Ảnh bộ lọc thông cao Gauss trong miền không gian (bên trái, có dấu chấm đen nhỏ ở giữa hình) sang miền tần số (bên phải)

Sử dụng thuật toán 1.5 – 3, ta tính tích chập của ảnh gốc trong miền tần số và bộ lọc trong miền tần số, sau đó lấy chuyển đổi ngược.

Hình 1.5 – 33. Ảnh quả bí ngô sau khi lọc bằng bộ lọc thông cao Gauss

Ta quan sát phổ của ảnh gốc và ảnh sau khi lọc (trên – phải xuống dưới – trái)



Hình 1.5 – 34. So sánh tần số của ảnh trước và sau khi lọc. Ảnh trên – trái: ảnh ban đầu. Ảnh trên – Phải: ảnh sau khi lọc. Ảnh dưới – trái: tần số ảnh ban đầu. Ảnh dưới – phải: tần số ảnh sau khi lọc.

Nhìn ảnh, ta thấy rằng ảnh không có hiệu ứng chuông, phổ ảnh cho thấy các giá trị tần số thấp dao động giữa tần số cao (biên quả bí ngô).

1.5.5.3 Lọc thông cao Butterworth

Phép lọc thông cao Butterworth trong không gian 2 chiều có cấp n và tần số chặt cụt D_0 xác đinh bởi công thức

$$H(u,v) = \frac{1}{1 + \left[\frac{D_0}{D(u,v)}\right]^{2n}} \quad (1.5-7)$$

Với $D(u,v)$ là khoảng cách từ tâm hình ảnh đến tọa độ (u,v).

Ta dùng bộ lọc này để lọc hình quả bí ngô:

Hình 1.5 – 35. Quả bí ngô

Khởi tạo bộ lọc

Thuật toán 1.5 – 6. Lọc thông cao Butterworth

```
Input: ảnh cần lọc
Output: ảnh sau khi dùng bộ lọc thông cao Butterworth
P = size(f, 1);     %Tạo bộ lọc có kích thước bằng với ảnh
Q = size(f, 2);
h = zeros(P, Q);
n = 5; %Chọn cấp của bộ lọc
D0 = 30;    %Chọn tần số chặt cụt

for i = 1:P
    for j = 1:Q
        D = sqrt((i - P./2).^2 + (j - Q./2).^2);
        h(i, j) = 1 ./ (1 + (D0 ./ D).^(2.*n));
    end
end
H = fftshift(h); %Chuyển bộ lọc sang miền tần số
P = size(f, 1);     %Tạo bộ lọc có kích thước bằng với ảnh
Q = size(f, 2);
h = zeros(P, Q);
n = 5; %Chọn cấp của bộ lọc
D0 = 30;    %Chọn tần số chặt cụt

for i = 1:P
    for j = 1:Q
        D = sqrt((i - P./2).^2 + (j - Q./2).^2);
        h(i, j) = 1 ./ (1 + (D0 ./ D).^(2.*n));
    end
end
H = fftshift(h); %Chuyển bộ lọc sang miền tần số
```

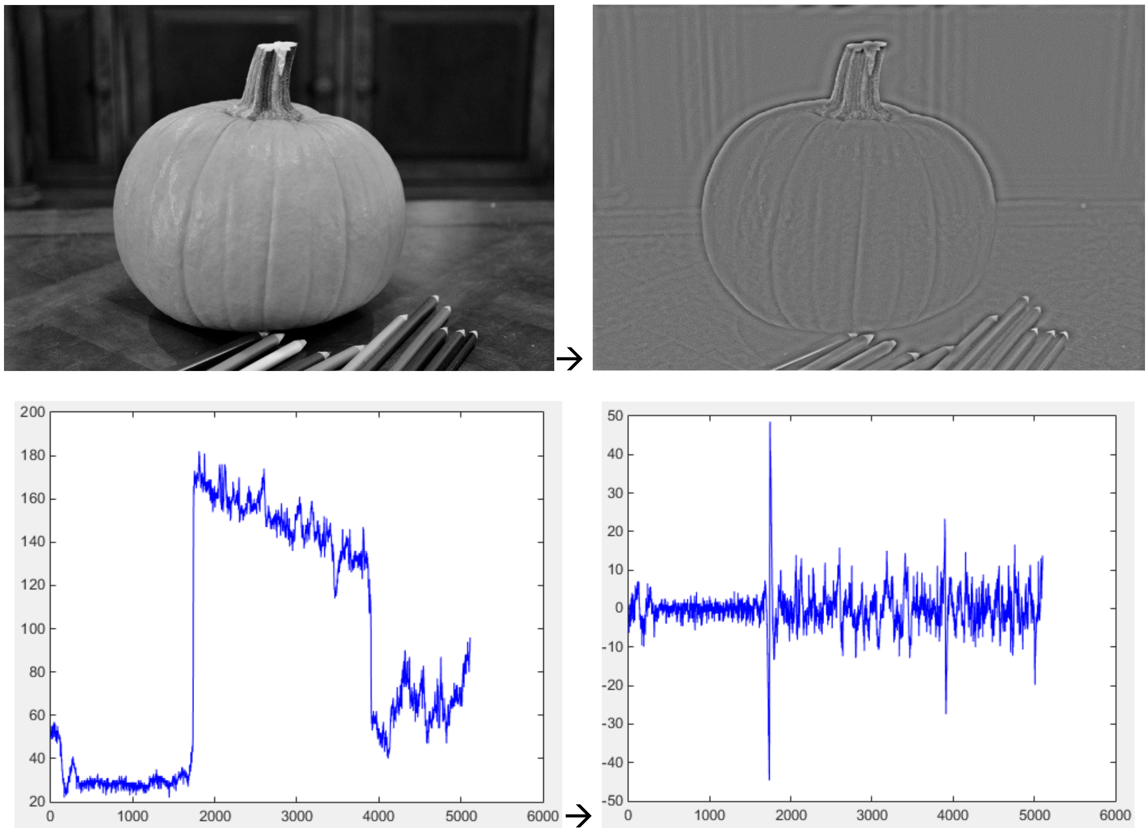
Ta được ảnh của bộ lọc trong miền không gian và miền tần số như sau:

Hình 1.5 – 36. Ảnh bộ lọc thông cao Butterworth trong miền không gian (bên trái) chuyển sang miền tần số (bên phải)

Sử dụng thuật toán 1.5 – 3, ta tính tích chập của ảnh gốc trong miền tần số và bộ lọc trong miền tần số, sau đó lấy chuyển đổi ngược.

Hình 1.5 – 37. Ảnh quả bí ngô sau khi lọc sử dụng bộ lọc thông cao Butterworth

Ta quan sát phổ của ảnh trước và sau khi lọc



Hình 1.5 – 35. So sánh tần số của ảnh trước và sau khi lọc. Ảnh trên – trái: ảnh ban đầu. Ảnh trên – Phải: ảnh sau khi lọc. Ảnh dưới – trái: tần số ảnh ban đầu. Ảnh dưới – phải: tần số ảnh sau khi lọc.

Ảnh sau khi lọc có xảy ra hiệu ứng chuông nhưng nhẹ hơn lọc thông cao Ideal, phổ tần số và ảnh cho thấy phần biên có tần số cao hơn hẳn những phần khác.

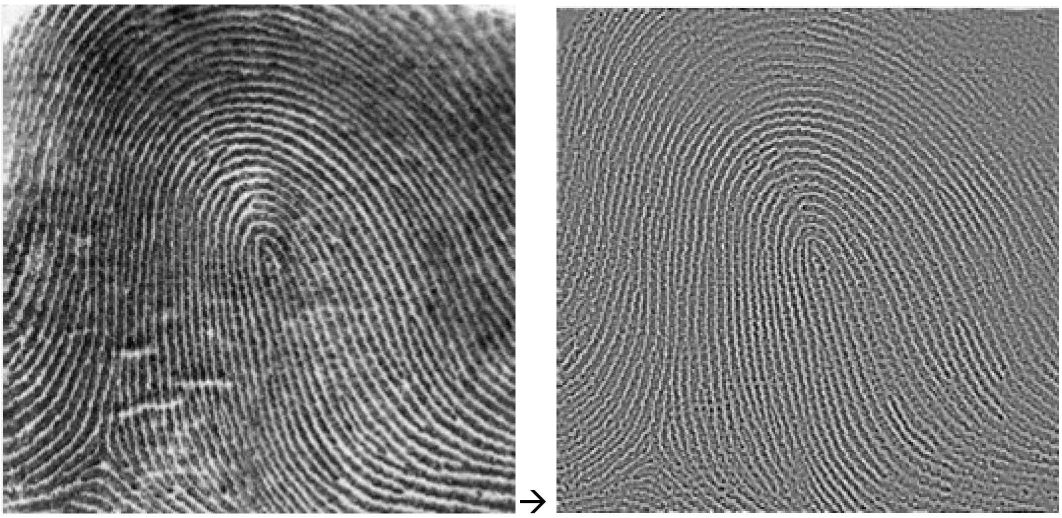
Hình ảnh sau là ví dụ của phép lọc thông cao Butterworth cấp 2 với các mức D_0 lần lượt (trái sang phải) là 15, 30 và 80.

Hình 1.5 – 35. Ảnh sử dụng bộ lọc thông cao Butterworth với D_0 từ trái sang là 15, 30, 80

1.5.6 Ứng dụng phép lọc sắc ảnh

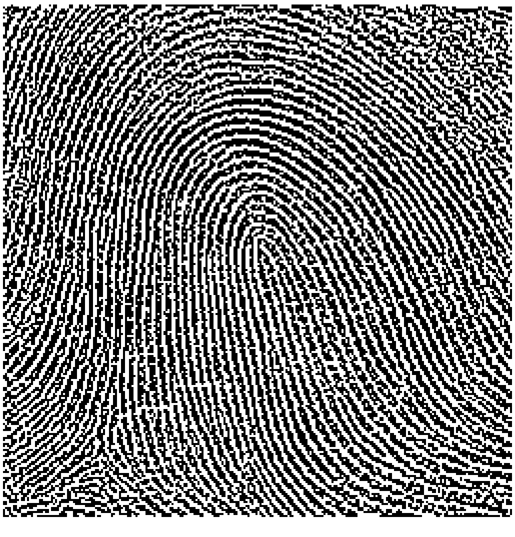
Các phép lọc ảnh tần số cao này còn được dùng để nhận diện dấu vân tay. Yếu tố quan trọng để máy móc nhận diện được vân tay đó là làm sao để tăng cường đường vân và giảm đi các vết bẩn. Để tăng cường đường vân, ta dựa vào yếu tố quan trọng đó là đường vân tay chứa tần số cao, sẽ không đổi khi ta sử dụng lọc thông cao, mặt khác, phép lọc này còn làm giảm các thành phần chứa tần số thấp, ví dụ như phông nền hay các vết bẩn. Vì vậy, ta thu được ảnh tăng cường bằng cách làm giảm tất cả các đặc trưng ngoại trừ đặc trưng có tần số cao.

Hình ảnh vân tay dưới đây sử dụng phép lọc thông cao Butterworth cấp 4 và tần số chặt cụt là 50.



Hình 1.5 – 36. Ảnh trái: ảnh vân tay có nhiều vết mờ, bẩn, đường vân có nơi không rõ nét. Ảnh phải: ảnh vân tay rõ nét sau khi sử dụng phép lộc thông cao Butterworth

Để tiện quan sát, ta sẽ biến đổi ảnh bên phải theo quy tắc điểm ảnh có giá trị âm sẽ có màu đen, còn dương sẽ có màu trắng



Hình 1.5 – 37. Biến đổi ảnh 1.5 – 36 (ảnh phải), điểm ảnh có giá trị âm sẽ có màu đen, còn dương sẽ có màu trắng

Sử dụng phép lọc thông cao, ta đã loại bỏ những vết bẩn trong ảnh và thu về ảnh vân tay rõ nét, thuận lợi hơn trong công tác điều tra.

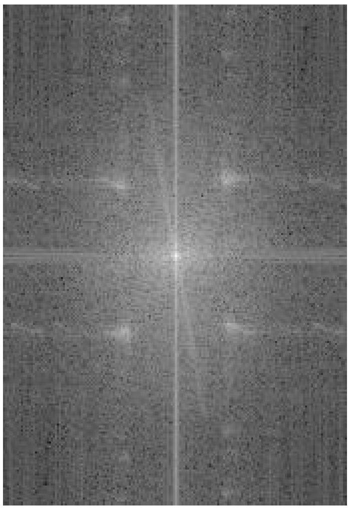
1.5.7 Lọc chặn

Đây là một ứng dụng của lọc ảnh trong miền tần số. Phép lọc chặn loại bỏ (hoặc đi qua) các tần số riêng biệt nào đó, ví dụ như nhiễu chu kỳ ứng với các nhánh hay các đường trong miền tần số, ta sẽ thiết kế một bộ lọc có tần số 0 tại những vị trí đó sẽ loại bỏ các nhiễu.

Ví dụ về nhiễu chu kỳ như khảm ảnh khi kết hợp nhiều ảnh lại để tạo khảm, nhiễu dòng quét khi dùng máy quét, hay nhiễu bán sắc (kiểu gợn sóng) của bức ảnh trong tờ báo dưới đây.

Hình 1.5 – 38. Ảnh xe bị nhiễu bán sắc, có những đường sọc ngang dọc

Chuyển ảnh vào miền tần số, ta được ảnh sau



Hình 1.5 – 39. Ảnh 1.5 – 38 chuyển sang miền tần số

Bạn có thể thấy trên ảnh có những đỉnh nhỏ, đỉnh này tương ứng với dạng nhiễu chu kỳ của ảnh bên miền không gian.

Các bước lọc chặn:

- Nhìn vào phổ $|F(x, y)|$ của ảnh nhiễu $f(x, y)$, tìm vị trí tần số có liên quan đến nhiễu.

- Tạo ảnh mặt nạ $M(u, v)$ với vết khuyết (các số 0) tại vị trí đó, những vị trí còn lại có giá trị 1.

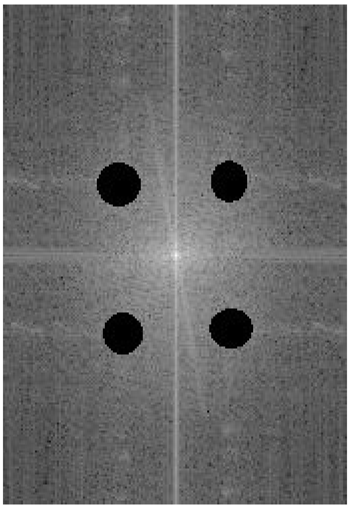
- Lấy tích mặt nạ với ảnh ban đầu đã được chuyển đổi, các giá trị 0 sẽ làm mất các tần số nhiễu

$$G(u, v) = M(u, v)F(u, v)$$

- Lấy chuyển đổi Fourier ngược để thu về ảnh khôi phục.

$$g(x, y) = \Im^{-1}\big(G(u, v)\big)$$

Ta xác định vị trí có vết khuyết



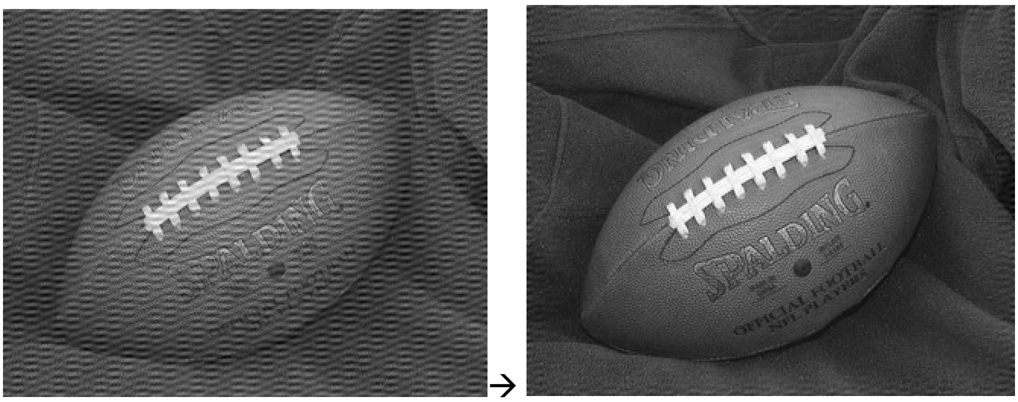
Hình 1.5 – 40. Vòng tròn đen là chỗ có vết khuyết

Lấy tích chập, chuyển đổi ngược, ta được kết quả

Hình 1.5 – 41. Ảnh chiếc xe sau khi lọc chặn

Như vậy ta đã loại bỏ được các đường quét trong ảnh ban đầu.

Một ví dụ sau về phép lọc chặn



Hình 1.5 – 42. Bên trái là ảnh có những đường sọc ngang dọc, dùng phép lọc chặn, ta được ảnh bên phải rõ ràng hơn

1.6 Tổng kết

Sau khi tìm hiểu về lọc trong miền tần số, ta thấy phép lọc này có chức năng lọc những tần số thấp và cao trong ảnh.

- Với thành phần tần số cao liên quan đến biên vật thể trong ảnh, làm rõ biên. Các phép lọc thông cao làm giảm các thành phần tần số thấp và bỏ qua thành phần tần số cao.

- Với thành phần tần số thấp liên quan đến các vùng mượt trong ảnh, làm mờ ảnh. Các phép lọc thông thấp làm giảm các thành phần tần số cao và bỏ qua thành phần tần số thấp.

Ngoài ra, phép lọc chặn giúp loại bỏ các nhiễu lặp lại trong ảnh, làm giảm các tần số chọn trước (và một vài lân cận) và bỏ qua các tần số khác.

Quy trình chung khi lọc trong miền tần số:

- Dùng thuật toán FFT chuyển ảnh $f(x, y)$ sang miền tần số thành ảnh $F(u, v)$.

- Tạo bộ lọc $h(x, y)$ cùng kích thước với ảnh cần lọc, dùng FFT chuyển sang miền tần số thành $H(u, v)$.

- Thực hiện phép tính $G(u, v) = H(u, v)F(u, v)$.

- Lấy chuyển đổi ngược IFFT của $G(u, v)$, ta được hình $g(x, y)$ là hình sau khi lọc.

1.7 Tài liệu tham khảo

1. Rafael C. Gonzalez, Richard E. Woods, 2007. Filtering in the Frequency Domain. In: *Digital Image Processing 3rd edition*, Prentice Hall.

2. William Hoff, 2014, *EENG 510 Lecture 09-1 Frequency Domain Filters*, https://www.youtube.com/watch?v=7Hj6HcI7dSA

3. William Hoff, 2014, *EENG 510 Lecture 09-2 Frequency Domain Filters*, https://www.youtube.com/watch?v=ytW8RnH3Pow

4. William Hoff, 2014, *EENG 510 Lecture 09-3 Frequency Domain Filters*, https://www.youtube.com/watch?v=oKVhNm82drA

5. CS425 Lab: Frequency Domain Processing, http://www.cs.uregina.ca/Links/class-info/425/Lab5/lesson.html

6. Lý Quốc Ngọc, 2015, slide bài giảng tuần 7 và tuần 8. *Xử lý ảnh số và video số*, khoa Công nghệ Thông tin, trường ĐH. Khoa học Tự nhiên – ĐHQG. Tp. Hồ Chí Minh.

7. TutorialsPoint, *Introduction to Frequency domain*, http://www.tutorialspoint.com/dip/introduction_to_frequency_domain.htm

8. TutorialsPoint, *Fourier Series and Transform*, http://www.tutorialspoint.com/dip/fourier_series_and_transform.htm

9. TutorialsPoint, *Convolution Theorem*, http://www.tutorialspoint.com/dip/convolution_theorm.htm

10. TutorialsPoint, *High Pass vs Low Pass Filters*, http://www.tutorialspoint.com/dip/high_pass_vs_low_pass_filters.htm

Phần 2

LỌC CONTOURLET

2.1 Giới thiệu

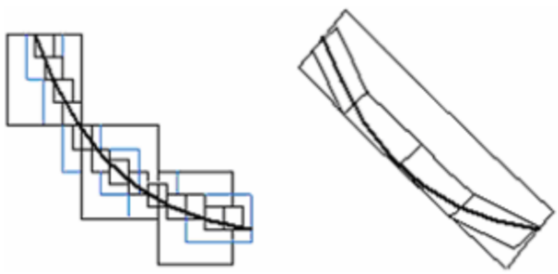
Gần đây, có một lớp các miền biến đổi mới – gọi chung là X-let, được xây dựng dựa trên phân tích đặc trưng thị giác: khoanh vùng (localized), đa mức (multiscale), có định hướng (directional/oriented). Contourlet do tôi tác giả Martin Vetterli và Đỗ Ngọc Minh đề xuất là một phép biến đổi hai chiều trong số đó. Điểm khác biệt giữa Contourlet và các biến đổi tương tự là nó cho phép linh hoạt chọn số hướng ở mỗi mức, với chi phí tính toán thấp ($O(N)$ cho ảnh N-pixels). Biến đổi Contourlet đã đạt kết quả khá tốt với ứng dụng khử nhiễu, rút trích đặc trưng ảnh…

Biến đổi Contourlet là mở rộng của biến đổi Wavelets hai chiều dùng nhiều mức và các băng lọc hướng. Contourlets phát triển các hướng phân tích cơ bản của ảnh (đối với Wavelets là theo các hướng: ngang, dọc, chéo) lên thành nhiều hướng khác nhau trong từng mức với tỉ lệ khung linh hoạt (với Wavelets là những khung vuông còn Contourlet có thể kéo dài khung cho phù hợp độ trơn của đường biên ảnh). Vì thế, biến đổi Contourlet có thể biểu diễn hiệu quả các đường biên trơn (đây cũng là những đặc điểm nổi bật thường thấy trong ảnh tự nhiên).

Trong báo cáo này, tôi sử dụng công cụ Contourlet do Đỗ Ngọc Minh đăng tải lên trang MathWorks vào ngày 27 tháng 11 năm 2005.[1]

2.2 So sánh biến đổi Wavelet và biến đổi Contourlet

Khoảng những năm 1990, người ta thường dùng biến đổi Wavelet để xử lý ảnh, nhưng biến đổi này chỉ tốt để tách ra các tính không liên tục ở các điểm cạnh mà không nhận ra được tính mịn ở các đường biên ảnh.



Hình 2.2 – 1. Biểu diễn bằng Wavelet (trái) và Contourlet (phải)

Ý tưởng của biến đổi Wavelet là sử dụng các nét bút hình vuông dọc theo đường cong để vẽ đường cong, với các kích thướt nét khác nhau tương ứng với cấu trúc đa phân giải của Wavelets. Vì vậy để sự phân giải trở nên tốt hơn, biến đổi Wavelet cần thiết sử dụng nhiều

[1] Tải công cụ Contourlet tại http://www.mathworks.com/matlabcentral/fileexchange/8837-contourlet-toolbox

“dấu chấm” (hình vuông nhỏ) để nắm giữ đường cong. Trái lại, biến đổi Contourlet sử dụng các hình được kéo dài ra ở nhiều hướng theo đường cong để vẽ đường cong với nhiều tính linh động, dễ uốn nén. Biến đổi Contourlet sử dụng các phân đoạn đường cong để thực hiện cục bộ, khai triển ảnh có hướng và đa phân giải. Và như vậy, tính hiệu quả của biến đổi Wavelet có lẽ không cao bằng biến đổi Contourlet nếu đường cong không theo chiều ngang hay dọc.

Dựa vào các ý tưởng trên, kết hợp với hệ thống thị giác con người và thống kê ảnh tự nhiên, phép biến đổi Contourlet khi tạo ảnh mới phải thỏa các tiêu chí:

- Đa phân giải: Cho ra xấp xỉ ảnh tốt, độ phân giải từ thô đến mịn.

- Tính địa phương: Các phần tử cơ sở của ảnh biểu diễn phải nằm trong cả miền không gian và tần số.

- Lấy mẫu giới hạn: Một số ứng dụng (như nén ảnh), ảnh biểu diễn phải tạo thành cơ sở hoặc một khung với số dư nhỏ.

- Có tính định hướng: Phép biểu diễn phải bao hàm các phần tử định hướng cơ sở ở nhiều hướng, nhiều hơn một vài hướng khi sử dụng các Wavelet khác nhau.

- Bất đẳng hướng: Để bắt các đường bao trơn trong ảnh, phép biểu diễn phải chứa các phần tử cơ sở sử dụng đa dạng các hình thon dài với tỉ lệ khác nhau.

3 tiêu chí đầu tiên có thể thực hiện bằng sóng Wavelet, nhưng 2 tiêu chí cuối cùng đòi hỏi một cấu trúc mới. Một thử thách khi bắt tính hình học và hướng của ảnh đến từ tính rời rạc của dữ liệu, đầu vào là ảnh mẫu xác định trên lưới hình chữ nhật. Ví dụ, các hướng không phải hướng ngang và dọc nhìn rất khác trong lưới hình chữ nhật. Do ảnh tạo bởi các điểm ảnh, ta không có khái niệm rõ ràng để xác định đường bao trơn của ảnh.

Nhiều phép chuyển đổi ban đầu xác định trên miền liên tục, sau đó mới phát triển qua rời rạc và sử dụng trên dữ liệu ảnh, còn phép biến đổi Contourlet này phát triển trên cấu trúc miền rời rạc trước, sau đó nghiên cứu tính hội tụ về một khoảng mở trong miền liên tục.

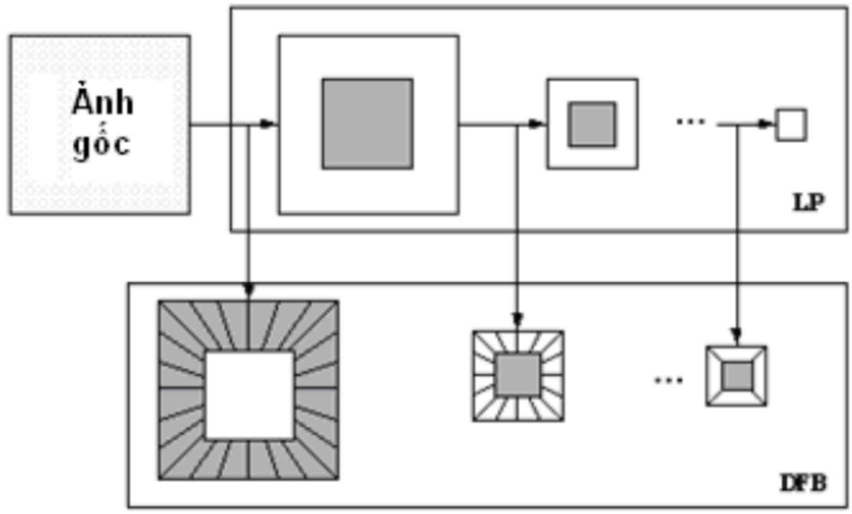
Biến đổi Contourlet dùng một dàn lọc hai chiều, phân tích ảnh thành các subband có hướng tại nhiều mức. Tại mỗi mức là sự kết hợp giữa một tháp Laplace và một dàn lọc có hướng.

Nhờ cấu trúc lợp ngói này mà các bước phân tích độc lập với nhau, mỗi mức sẽ có một số hướng khác nhau (là lũy thừa của 2). Đặc trưng này khiến cho Contourlet là biến đổi có thể đạt được độ linh hoạt cao với một chi phí tính toán chấp nhận được.

2.3 Quá trình biến đổi Contourlet

Quá trình được mô tả cụ thể như sau:

Biến đổi Contourlet gồm hai phân tích: phân tích đa mức (multi-scale) và phân tích có hướng (directional).



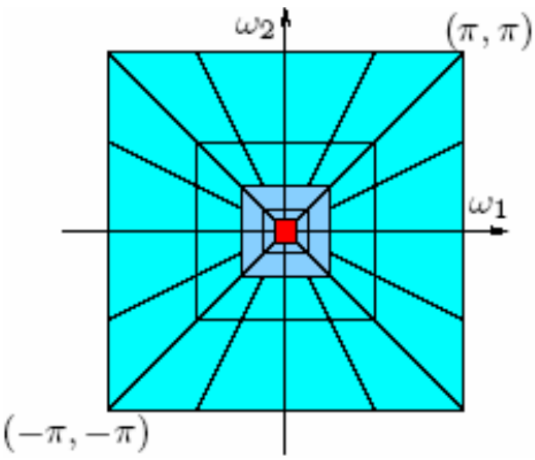
Hình 2.3 – 1. Dàn lọc Contourlet

Ở mức đầu tiên, ảnh đầu vào được phân tích qua hai bước.

- Bước 1: Tháp Laplace (Laplace Pyramid – LP) được dùng để thu giữ những điểm rời rạc. LP phân tích ảnh đầu vào thành 1 ảnh con "thô" và một tập các ảnh band-pass.

- Bước 2: Băng lọc có hướng (Directional Filter Bank - DFB) được dùng để nối các điểm rời rạc thành các cấu trúc dạng tuyến tính theo nhiều hướng. DFB phân tích các ảnh band-pass ở bước 1 thành các ảnh con, "chồng" lên nhau.

Ở các mức tiếp theo, quá trình phân tích như bước 1 và 2 sẽ được lặp lại với đầu vào của mức này là đầu ra của mức trước đó. Kết quả sau cùng là các phân tích LP và DFB.

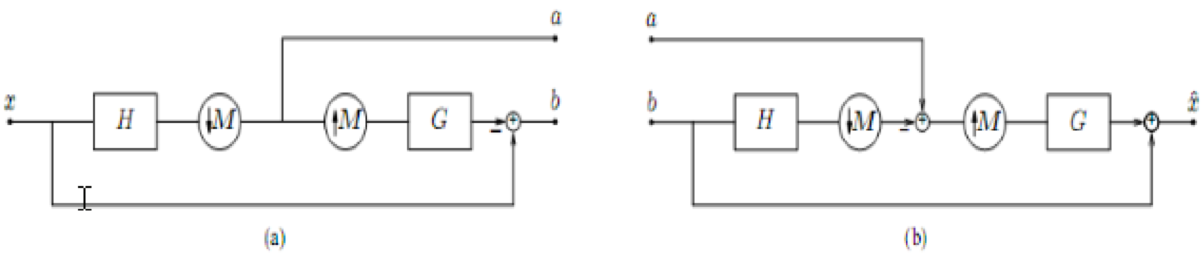


Hình 2.3 – 2. Một phân hoạch thường dùng của biến đổi Contourlet

Biến đổi Contourlet trước hết sử dụng tháp Laplace để bắt các điểm không liên tục và sau đó sử dụng một băng lọc có hướng nối các điểm không liên tục đó thành một cấu trúc tuyến tính. Kết quả chung là một ảnh mở rộng có sử dụng các phần tử cơ sở như các phân đoạn đường viền, gọi là Contourlet. Đặc biệt, Contourlet hỗ trợ kéo dài với nhiều thang đo, hướng và tỉ lệ hình dáng, điều này cho phép Contourlet xấp xỉ hiệu quả đường viền mượt với độ phân giải đa dạng. Trong miền tần số, chuyển đổi Contourlet cho ra cách phân rã đa mức và có hướng.

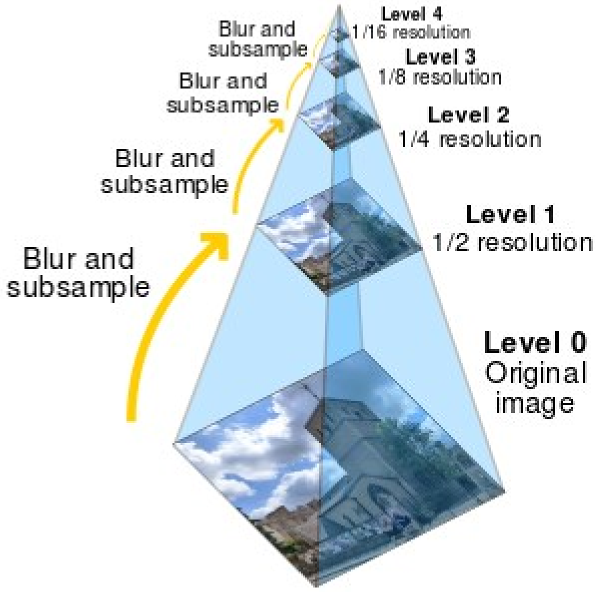
2.3.1 Tháp Laplace

Một cách để đạt được phân rã đa mức là sử dụng LP do Burt và Adelson giới thiệu. Phân rã LP ở mỗi mức sinh ra một thông thấp (lowpass) của tín hiệu gốc và một ảnh băng tần (bandpass) thể hiện sai số giữa tín hiệu gốc và tín hiệu dự đoán, kết qủa của ảnh băng tần được biểu diễn ở hình sau



Hình 2.3 – 3. Lược đồ tháp LP. (a) một mức phân rã, (b) xây dựng lại

H và G là các bộ lọc phân tích và tổng hợp và M là ma trận lấy mẫu. Quá trình có thể được lặp trên tín hiệu thô. Ở hình trên, đầu ra là một xấp xỉ thô $a[n]$ và sai số $b[n]$ giữa tín hiệu gốc và tín hiệu dự đoán. Ta có thể lặp quá trình này bằng cách phân rã tín hiệu thô nhiều lần (thường là theo nhân tử của 2). Ảnh gốc được lấy nhân chập Gauss, ảnh kết quả là một phiên bản được lọc lowpass của ảnh gốc. Sau đó LP tính toán sai số giữa ảnh gốc và ảnh được lọc lowpass. Quá trình này tiếp tục để có được tập các ảnh được lọc bandpass. Như vậy, LP là một tập hợp của các bộ lọc bandpass thu được bằng việc lặp lại các bước này nhiều lần với một chuỗi các ảnh. Sau mỗi lần thực hiện, kích thước ảnh sẽ giảm đi một lượng bằng f_i/f_{i+1} (f_i là ảnh sau khi lọc, f_{i+1} là ảnh trước khi lọc). Theo cách này, ta sẽ có một cấu trúc xếp chồng lên nhau, tương tự như cấu trúc Kim tự tháp mà kích thước giảm dần từ gốc đến đỉnh.

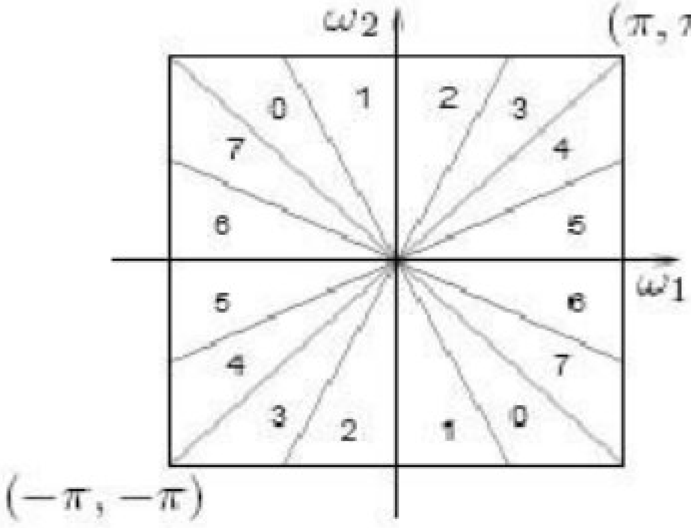


Hình 2.3 – 4. Cấu trúc tháp Laplace

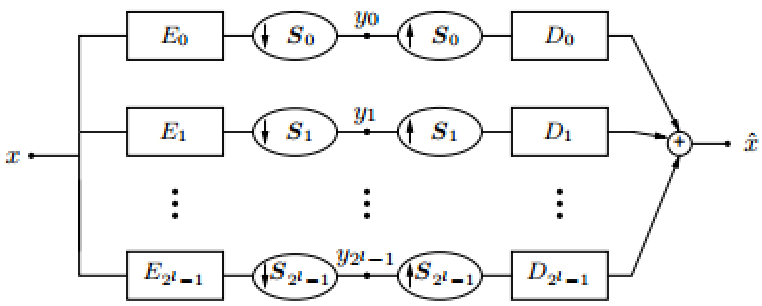
LP có thể được sử dụng để biểu diễn ảnh như một dãy các ảnh được lọc băng tần, mỗi ảnh được lấy mẫu tại các mật độ thưa hơn liên tiếp. LP thường được sử dụng trong xử lý ảnh và nhận dạng do LP giảm sự tính toán. Một hạn chế của LP là lấy mẫu chồng ẩn (implicit oversampling). Tuy nhiên trái ngược với lược đồ Wavelet được lấy mẫu một cách tới hạn thì LP có đặc tính phân biệt mà mỗi mức tháp sinh ra chỉ một ảnh bandpass (thậm chí cho trường hợp đa hướng) và ảnh này không có các tần số bị "đổi tần" ("scrambled"). Sự đổi tần này xảy ra trong dàn lọc Wavelet ở một kênh tần cao, sau khi lấy mẫu xuống, được xếp trở lại băng tần thấp, và như vậy phổ ảnh của nó bị phản chiếu. Trong LP, ta có thể tránh hiệu ứng này bằng việc chỉ lấy mẫu kênh tần thấp.

2.3.2 Băng lọc có hướng được lặp (Iterated Directional Filter Bank – IDFB)

Năm 1992, Bamberger và Smith đã giới thiệu một băng lọc có hướng 2-D (DFB) có thể nén ảnh tối đa mà vẫn tái tạo ảnh tốt. DFB thực hiện hiệu quả thông qua phân giải cấu trúc cây nhị phân $l-$ cấp, đưa ra 2^l băng con với phân vùng tần số hình nêm chữ V (wedge-shaped) như hình dưới đây.

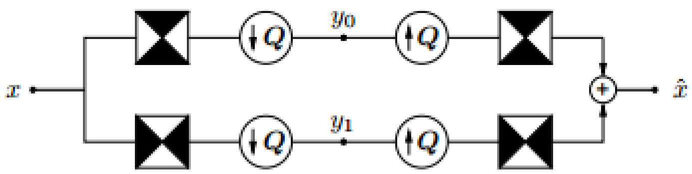


Hình 2.3 – 5. Phân vùng con tần số DFB với $l = 3$



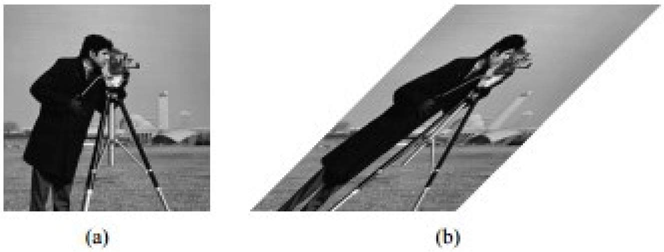
Hình 2.3 – 6. Ảnh đa kênh của băng lọc có hướng cấu trúc cây mức l

Xây dựng chính của DFB liên quan tới việc điều chỉnh ảnh đưa vào và sử dụng băng lọc sắp xếp ngũ điểm (Quincunx Filter Bank - QFB) với các bộ lọc hình thoi. Để đạt được phân vùng con tần số mong muốn,ta phải sử dụng quy tắc mở rộng cây phức tạp để các băng con (subbands) định hướng tốt hơn. Minh và Martin đã giới thiệu một cấu trúc DFB mới, không cần điều chỉnh ảnh đầu vào, có quy tắc đơn giản hơn để mở rộng cây phân giải. DFB này bao gồm 2 khối, khối đầu tiên là QFB 2 kênh với bộ lọc quạt (fan filter) chia phổ 2 chiều thành 2 hướng ngang và dọc như hình sau



Hình 2.3 – 7. Vùng màu đen là tần số lý tưởng hỗ trợ cho mỗi bộ lọc,
Q là ma trận lấy mẫu sắp xếp ngũ điểm

Khối thứ 2 là toán tử kéo (shearing operator) nhằm định lượng các sắp xếp lại cho mẫu ảnh. Hình dưới đây minh họa cho toán tử kéo với cạnh có hướng -45^o trở thành thẳng đứng.



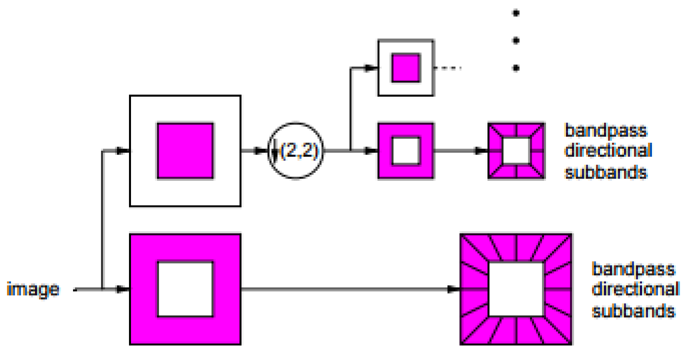
Hình 2.3 – 8. (a): Ảnh ban đầu. (b): Kéo ảnh (a) nghiêng -45^o

Bằng cách thêm cặp toán tử kéo và ngược của toán tử này (không kéo) ở trước và sau băng lọc 2 kênh, ta thu được phân vùng tần số có hướng khác nhau trong khi vẫn tái tạo ảnh tốt. Do đó, mấu chốt của DFB là phải biết cách kết hợp các toán tử kéo thích hợp với phân vùng 2 hướng của QFB ở mỗi node trong cấu trúc băng lọc cây nhị phân để thu về phổ phân loại 2 chiều mong muốn.

2.3.3 Biến đổi Contourlet rời rạc

Kết hợp tháp Laplace và băng lọc có hướng, ta có cấu trúc băng lọc đôi (Double Filter Bank – DoFB). Do DoFB thiết kế để bắt các tần số cao (đại diện cho hướng) của ảnh vào, tần số thấp sẽ giữ ít.

Ảnh dưới đây là sự phân rã đa tỉ lệ và có hướng sử dụng tổ hợp LP và băng lọc có hướng (DFB). Ảnh bandpass từ LP được đưa vào DFB, do đó ta có thể bắt được thông tin có hướng. Giản đồ này có thể lặp cho ảnh thô.



Hình 2.3 – 9. Giản đồ phân rã đa tỉ lệ có hướng sử dụng LP và DFB

Kết quả tổ hợp này là cấu trúc lọc băng đôi lặp, có chức năng phân rã ảnh thành các băng con có hướng ở nhiều thang đo.

2.4 Thuật toán biến đổi Contourlet

Đầu vào:

- Ảnh đưa vào

- Chọn số phân rã và hướng ở mỗi cấp

- Chon bộ lọc để tính toán các phân rã và hướng

Xử lý:

- Tính toán phân rã của một ảnh

- Tính toán phân rã hướng của ảnh bandpass

- Lặp lại hai bước trên đến khi số phân rã tháp và hướng được hoàn thành

Đầu ra:

- Ảnh được xây dựng lại.

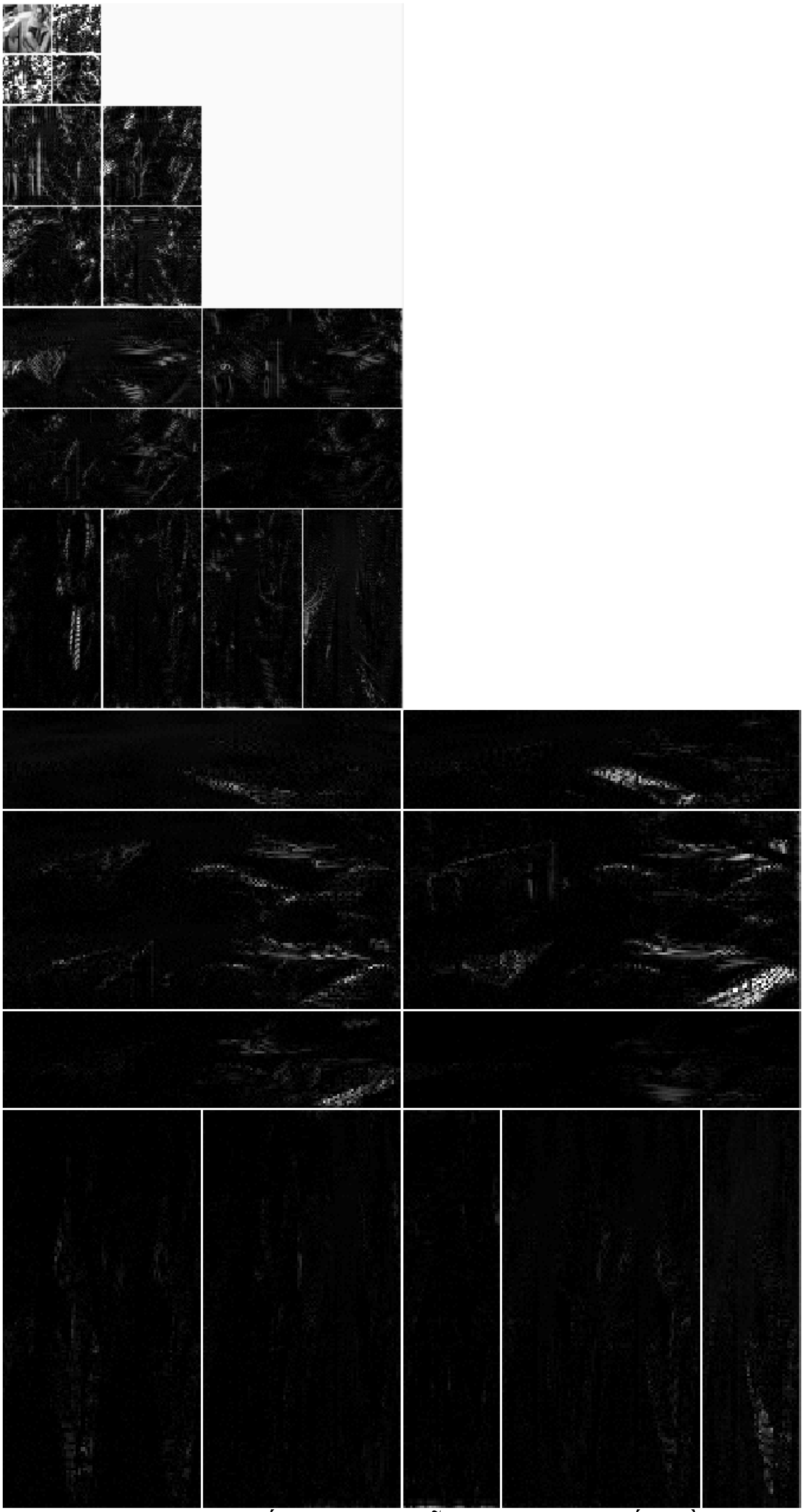
2.5 Ví dụ minh họa

Ví dụ: xét ảnh

Hình 2.5 - 1

Ta chọn số phân rã là 4, mức của mỗi phân rã lần lượt là 0, 2, 3, 4 và bộ lọc pkva.

Sau khi phân rã ảnh, ta được các ảnh con ở mỗi mức như sau



Hình 2.5 – 2. Số ảnh con ở mỗi mức từ trên xuống lần lượt là $2^0, 2^2, 2^3, 2^4$

Sau khi xử lý từng mức, tái tạo ảnh, ta được ảnh sau

Hình 2.5 – 3. Ảnh trái: ảnh ban đầu. Ảnh phải: ảnh sau khi xử lý

Sai số bình phương trung bình là 0.

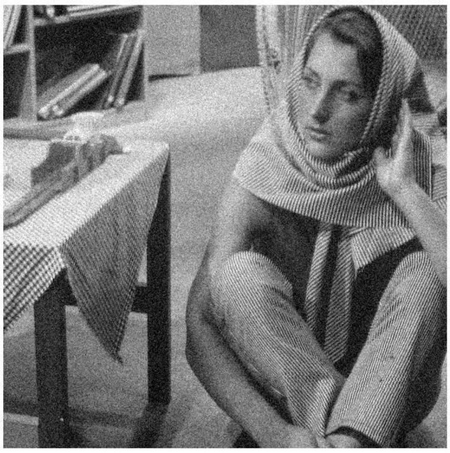
2.5.1 Lọc nhiễu

Ta sẽ dùng biến đổi Wavelet và biến đổi Contourlet để khử nhiễu cùng một ảnh.

Ta chọn số phân rã là 5, mức của mỗi phân rã lần lượt là 0, 0, 4, 4, 5 và bộ lọc pkva để phân rã ảnh, bộ lọc 9-7 dùng trong LP.

- Ảnh gốc

- Làm nhiễu ảnh



Hình 2.5 – 4. Ảnh sau khi làm nhiễu ảnh gốc

Kết quả:

-Wavelet

Hình 2.5 – 5. Lọc nhiễu ảnh 2.5 – 4 bằng Wavelet

- Contourlet

Hình 2.5 – 6. Lọc nhiễu ảnh 2.5 – 4 bằng Contourlet

So sánh 2 ảnh với ảnh gốc, ta thấy phép Contourlet có khả năng phục hồi các đường bao tốt hơn Wavelet.

2.5.2 Xấp xỉ phi tuyến

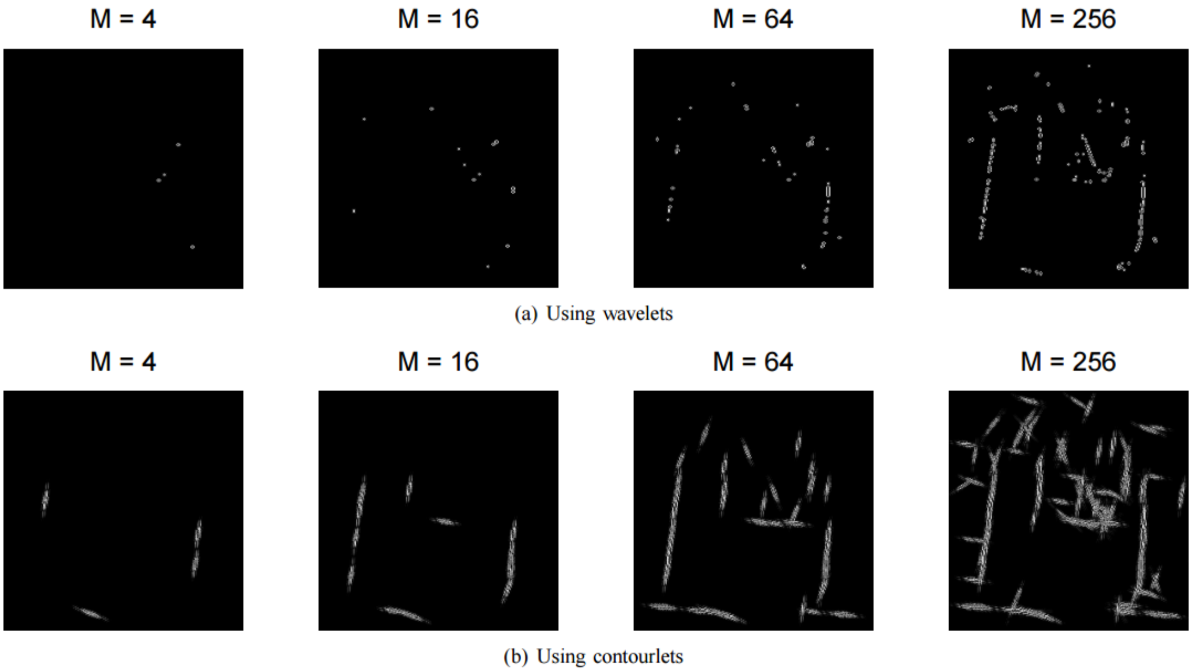
Ta sẽ so sánh phép xấp xỉ phi tuyến của biến đổi Wavelet và biến đổi cotourlet.

Với giá trị M cho trước, ta chọn M hệ số thích hợp nhất ở mỗi miền biến đổi, sau đó so sánh ảnh tái tạo từ tập M hệ số này. Ta hi vọng rằng đa số phép làm mịn xảy ra ở biên của ảnh.

Ta xử lý ảnh sau

Hình 2.5 – 7

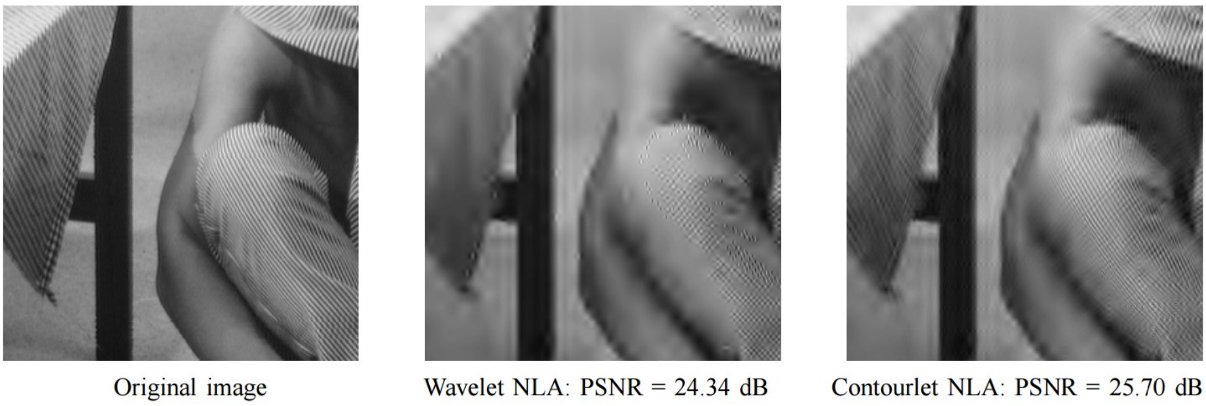
Kết quả



Hình 2.5 – 8. Xấp xỉ ảnh 2.5 – 7 sử dụng Wavelet (hàng trên) và Contourlet (hàng dưới)

Hình ảnh trên là chuỗi các ảnh xấp xỉ phi tuyến, ảnh trên sử dụng Wavelet, ảnh dưới sử dụng Contourlet. Với giản đồ Wavelet cho thấy phép biến đổi này bắt các đường bao chậm bằng cách cô lập các dấu "chấm". Ngược lại, giản đồ Contourlet cho thấy khả năng chọn lọc nhanh bằng cách "phác họa" đường bao.

Ảnh dưới đây so sánh chi tiết 2 ảnh xấp xỉ phi tuyến bằng chuyển đổi Contourlet và Wavelet. Phép Contourlet cho thấy khả năng bắt các đường bao mịn tốt hơn Wavelet (ví dụ như đường chỉ có hướng trên quần).



Hình 2.5 – 9. Xấp xỉ phi tuyến ảnh bên trái bằng Wavelet (ảnh giữa) và Contourlet (ảnh phải)

2.6 Ứng dụng của biến đổi Contourlet

Biến đổi Contourlet là một cấu trúc biến đổi rời rạc có khả năng cung cấp cách mở rộng thưa cho các ảnh có đường bao mượt. Biến đổi này có những ứng dụng như

- Khử nhiễu ảnh, rút trích các đặc trưng ảnh.

- Dùng làm miền biến đổi trong các hệ thống ẩn thông tin nhằm tăng cường độ mạnh cũng như độ bền vững, ứng dụng cho việc chống sao chép, bảo vệ bản quyền, … trên các tín hiệu số.

- Phát hiện cạnh: Thuật toán phát hiện cạnh như sau:

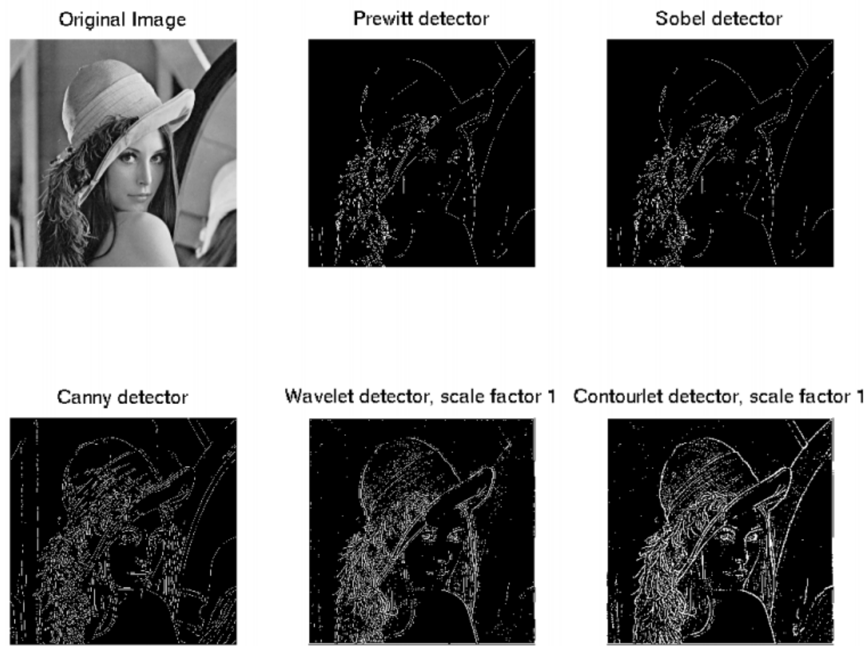
+ Lấy biến đổi Contourlet của ảnh.

+ Chọn hệ số tỉ lệ, chặt cụt các hệ số khác.

+ Thực hiện biến đổi ngược.

+ Lấy ngưỡng dựa trên trung bình các điểm ảnh.

Dưới đây là ảnh so sánh kết quả phát hiện cạnh dựa trên biến đổi Contourlet so với các phép biến đổi khác.



Hình 2.6 – 1. Dùng các phép biến đổi để phát hiện cạnh của ảnh trên – trái.
Ảnh trên – giữa: Biến đổi Prewitt
Ảnh trên – phải: Biến đổi Sobel
Ảnh dưới – trái: Bến đổi Canny
Ảnh dưới – giữa: Biến đổi Wavelet
Ảnh dưới – phải: Biến đổi Contourlet

2.7 Tài liệu tham khảo

1. M. N. Do and M. Vetterli, *The contourlet transform: an efficient directional multiresolution image representation*, IEEE Transactions Image on Processing, vol. 14, no. 12, pp. 2091-2106, Dec. 2005.

2. M. N. Do and M. Vetterli, *Framing pyramids*, IEEE Transactions on Signal Processing, vol. 51, pp. 2329-2342, Sep. 2003.

3. A. L. Cunha, J. Zhou, and M. N. Do, *The nonsubsampled contourlet transform: Theory, design, and applications*, IEEE Transactions on Image Processing, vol. 15, no. 10, pp. 3089-3101, Oct. 2006.

4. Wei-shi Tsai, *Contourlet Transforms for Feature Detection*, May 9, 2008

5. Dương Minh Đức và Dương Anh Đức, *Kỹ thuật ẩn thông tin trên ảnh dựa trên điều biến lượng tử và biến đổi Contourlet*, Tạp chí phát triển KH&CN, tập 12, số 11 – 2009.