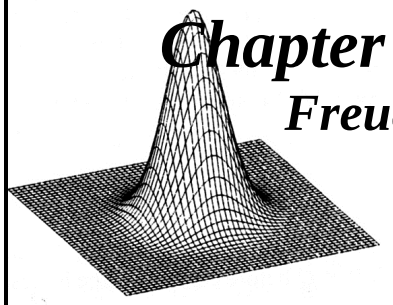


Chapter 4
Freuency Domain Processing

پردازش دامنه فرکانس

پیش نمایش(Preview)

این فصل مطابق با مباحث فیلترسازی است که در فصل ۳ بحث کردیم. ولی در اینجا تمام فیلترسازیها از طریق مبدل فوریر در دامنه فرکانس انجام می‌شود. تبدیل فوریر هم شالوده فیلترسازی خطی است و هم در زمینه‌هایی همچون تقویت و بهبود تصویر، بازگرداندن تصویر به حالت اصلی، فشرده‌سازی داده‌های تصویری و سایر انواع برنامه‌های کاربردی در زمینه طراحی و ایجاد راه حلهای فیلترسازی انعطاف پذیری زیاد ایجاد می‌کند. نکته اصلی در این فصل نحوه فیلترسازی دامنه فرکانس در نرم افزار MATLAB است. همان طور که در فصل ۳ گفتیم برای تشریح فیلترسازی دامنه فرکانس از مثالهای تقویت و بهبود تصویر از جمله فیلترسازی پایین‌گذر، بالاگذر و فرکانس بالا استفاده می‌کنیم. همچنین نشان دادیم که چگونه پردازش فضا و دامنه فرکانس با هم ادغام می‌شود تا نتایجی به دست آید که بهتر از استفاده انفرادی از هر یک از آنها باشد. فرضیه‌ها و شگردهای ابداع شده در بخش بعد کلی هستند و مثالهای زیادی از کاربرد این عناصر در فصلهای ۵، 6 آورده شده است.

۴.۱. تبدیل ۲ بعدی و متمایز فوریر(The 2-D Discrete Fourier Transform)

اگر فرض کنیم در حالتهای x=0,1,2…,M-1, y=0,1,2,…, N-1 دلالت بر تصویر M*N دارند تبدیل ۲ بعدی و تبدیل متمایز فوریر((discrete fourier transform(DFT)) از f که در f(u,v) ذکر شده است در معادله زیر دیده می‌شود:

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+uy/N)}$$

در حالتهای u=0,1,2,…,M-1, v=0,1,2,…,N-1 می‌توان برای گسترش سینوس و کسینوس نسبتهای نمایی با متغیرهای u, v فرکانس آنها را تعیین کرد (x, y جمع بسته می‌شوند) دامنه فرکانس سیستم مختصاتی است که بوسیله f(u,v) پدید آورده می‌شود و u,v متغیرهای فرکانس هستند.

این شبیه به دامنه فضایی است که در فصل قبل مطالعه کردیم. که سیستم مختصاتی است که توسط f(x,y) ایجاد شده است و x,y متغیرهای فضایی آن هستند. مستطیل M*N که با u=0,1,2,…,M-1, V=0,1,2,…,N-1 تعریف می‌شود اغلب مستطیل فرکانس نامیده می‌شود. اندازه مستطیل فرکانس همانند تصویر ورودی است.

معکوس تبدیل متمایز فوریر(DFT) در رابطه زیر نشان داده شده است:

$$F(x,v)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} f(u,v)e^{-j2\pi(ux/M+uy/N)}$$

بنا بر این با توجه به f(u,v) می‌توانیم f(x,y) را با استفاده از تبدیل متمایز فوریر(DFT) به دست آوریم. مقادیر f(u,v) در این معادله گاهی ضرایب گسترده فوریر(fourier coefficient) نامیده می‌شوند.

در برخی از فرمولهای تبدیل متمایز فوریر(DFT) عبارت 1/M/N جلوی تبدیل گذاشته می‌شود و در برخی نیز جلوی معکوس آن گذاشته می‌شود. برای این که تبدیل فوریر سازگار با نرم افزار MATLAB انجام شود در تمام این تحقیق فرض می‌کنیم عبارت فوق جلوی این معکوس گذاشته می‌شود که این مسئله در معادله قبل نیز دیده می‌شود زیرا شاخص‌های آرایه در MATLAB از یک شروع می‌شود نه صفر(f(1,1, در نرم افزار MATLAB مطابق با مقادیر ریاضی f(0,0), F(0,0) در حالتهای تبدیل و معکوس است.

مقدار تبدیل در مبداء دامنه فرکانس (یعنی F(0,0)) جزء dc تبدیل فوریر نامیده می‌شود. اصطلاح dc در مهندسی برق به معنی جریان مستقیم (با فرکانس صفر) است. مشکل است که نشان دهیم که F(0,0) معادل MN ضربدر میانگین f(x,y) است.

حتی اگر f(x,y) واقعی باشد تبدیل آن به طور کلی پیچیده است. اصل کلی تحلیلی یک تبدیل محاسبه طیف(spectrum) آن است (یعنی بزرگی F(u,v) و نمایش آن به صورت یک تصویر) اگر فرض کنیم R(u,v), I(u,v) نمایانگر مولفه‌های واقعی و تخیلی f(u,v) هستند طیف فوریر به شکل زیر تعریف می‌شود:

$$|F(u,v)|=\left[R^2(u,v)+I^2(u,v)\right]^{1/2}$$

زاویه مرحله‌ای(phase angle) این تبدیل به شکل زیر تعریف می‌شود:

$$\phi(u,v)=\tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

دو تابع قبلی می‌توانند نمایانگر F(u,v) در نماد قطبی مقدار پیچیده زیر باشند:

$$F(u,v)=|F(u,v)|e^{j\phi(u,v)}$$

طیف توان(power spectrum) به صورت مربع بزرگی تعریف می‌شود.

$$P(u,v)=|F(u,v)|^2$$
$$=R^2(u,v)+I^2(u,v)$$

هر یک از دو عبارت زیر|F(u,v)| or p(u,v) را که در نظر بگیریم برای تصور کردن آن باید بگوییم که شکل مادی ندارد:

اگر f(x,y) واقعی باشد تبدیل فوریر آن در مبداء متقارن و مزدوج است.

$$F(u,v) = F^*(-u,-v)$$

که تلویحاً نشان می‌دهد که طیف فوریر نیز در مبداء متقارن است.

$$|F(u,v)| = |F(-u,v)|$$

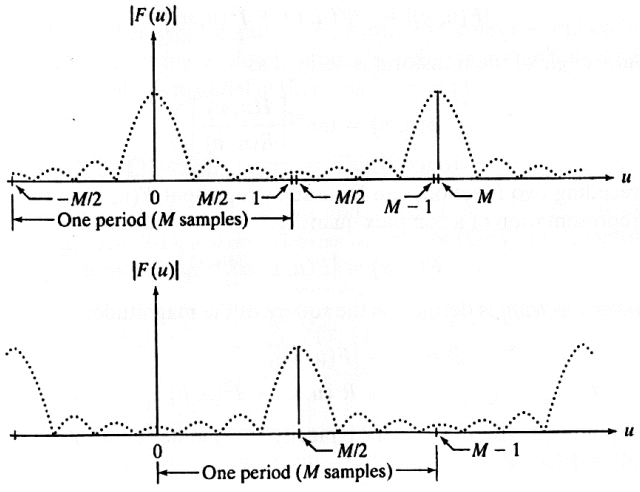
با جایگزینی مستقیم آن در معادله f(u,v) نتیجه زیر حاصل می‌شود:

$$F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,\upsilon+N)$$

به عبارت دیگر تبدیل متمایز فوریر(DFT) در هر دو جهت u, v به طور نامحدود متناوب است و تناوب آن با m, N مشخص می‌شود.. این تناوب همچنین یکی از خصوصیات تبدیل متمایز فوریر(DFT) به صورت معکوس است.

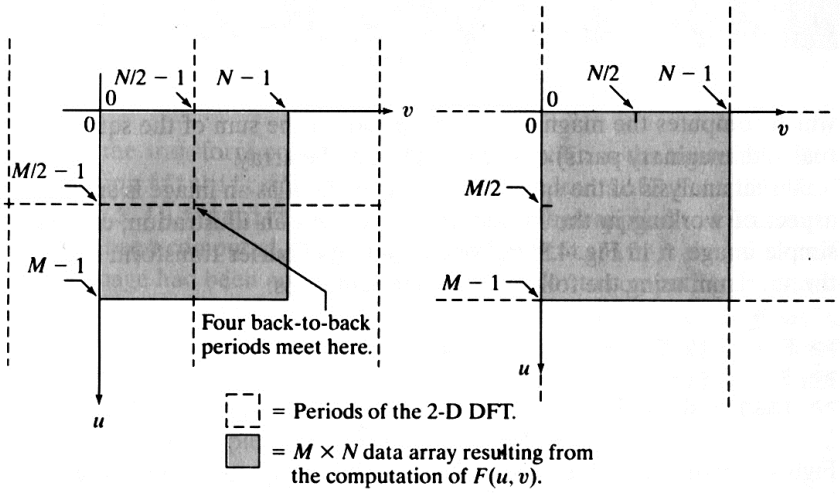
$$f(x,y) = f(x+M,y) = f(x,y+N) = f(x+M,y+N)$$

یعنی تصویری که با تبدیل معکوس فوریر به دست آمده باشد به صورت تناوبی نامحدود است. این موضوع معمولاً باعث سردرگمی می‌شود زیرا تصویرهای حاصل از تبدیل معکوس فوریر لزوماً متناوب نیستند. این صرفاً یکی از خصوصیات ریاضی تبدیل متمایز فوریر(DFT) و حالت معکوس آن است. به خاطر داشته باشید که با اجرای تبدیل متمایز فوریر(DFT) فقط یک تناوب محاسبه می‌شود. بنا بر این با آرایه‌هایی به اندازه M*N کار می‌کنیم.مسئله تناوب وقتی مهم می‌شود که ارتباط داده‌های تبدیل متمایز فوریر(DFT) را با تناوبهای تبدیل بررسی کنیم. مثلاً طیف یک تبدیل یک بعدی F(u) در تصویر ۴.۱ دیده می‌شود. در این صورت عبارت تناوب به شکل |f(u)| = |F(u+m)| می‌شود. در نتیجه (f(u)) = F(u+m) و علت آن تقارن |f(u)|=|F(-u)| است. این خصوصیت تناوب نشان می‌دهد که f(u) تناوبی به اندازه m دارد و خصوصیت متقارن نشان می‌دهد که بزرگی تبدیل در مبداء همچون تصویر ۴.۱.(a) دیده می‌شود. این تصویر و توضیحات پیش از آن نشان می‌دهد که بزرگی مقادیر تبدیل از M/2 به M-1 تکرار مقادیر نیم تناوب سمت چپ مبداء است.



از آنجائی که تبدیل متمایز فوریر(DFT) 1-Dیک بعدی فقط برای M نقطه اجرا می‌شود (یعنی مقادیر u در حد فاصل [0,M-1] محاسبه تبدیلهای 1-D دو نیم تناوب پشت به پشت در این حد فاصل ایجاد می‌کند. تمایل داریم که یک تناوب با ترتیب صحیح حد فاصل [0,M-1]به دست آوریم. Gonzalez & Woods[2002]در سال ۲۰۰۲ می‌گفتند به راحتی می‌توان نشان داد که تناوب مطلوب با ضرب f(x) در $(-1)^x$ قبل از محاسبه تبدیل به دست می‌آید. در این روند مبداء تبدیل به نقطه u=M/2 منتقل می‌شود که در تصویر ۴.۱.(b)نیز دیده می‌شود. حالا مقدار طیف u=0 در تصویر ۴.۱.(b)مطابق با |F(-M/2)| در تصویر ۴.۱.(a) است. به همین ترتیب مقادیر |F(M/2)| و |F(M-1)| در تصویر ۴.۱.ب مطابق با f(0) و |F(M/2-1)| در تصویر ۴.۱.(a) است.

وضعیت مشابهی در تابعهای ۲ بعدی وجود دارد. با محاسبه تبدیل متمایز فوریر(DFT) ۲ بعدی نقاط تبدیل در حد فاصل مستطیل تصویر ۴.۲.(a) دیده می‌شود. قسمت سایه‌دار مقادیر f(u,v) نشان می‌دهد که با اجرای معادله ۲ بعدی تبدیل فوریر که در آغاز این بخش تعریف شد به دست می‌آیند. همان طور که در تصویر ۴.۱.(a) دیده می‌شود مستطیلهای خط‌چین‌دار تکرارهای متناوب هستند. ناحیه سایه‌دار نشان می‌دهد که مقادیر f(u,v) شامل ۴ تناوب پشت به پشت یک چهارم هستند که در نقطه‌ای در تصویر ۴.۲.(a) دیده می‌شوند. با انتقال مقادیر مبداء تبدیل به کانون مستطیل فرکانس تحلیل تصویری طیف ساده می‌شود. سپس همان طور که در تصویر ۴.۲.(b)نشان داده شده است تناوبها همراستا می‌شوند. همان طور که در مبحث قبل در مورد تابعهای تک بعدی دیدیم، مقدار این طیف در (m/2, N/2) در تصویر ۴.۲.(b)همانند مقدار آن در (0, 0) در تصویر ۴.۲.(a) است و مقدار (0, 0) در تصویر ۴.۲.(b)مانند مقدار آن در (M/2, -N/2) در تصویر ۴.۲.(a) است. مقدار (M-1, N-1) در تصویر ۴.۲.(b)مانند مقدار (M/2-1, N/2-1) در تصویر ۴.۲.(a) است.



تصویر ۴.۲.(a): ((a)) طیف فوریر M*N (ناحیه سایه‌دار) ۴ تناوب پشت به پشت را در داده‌های طیف نشان می‌دهد. ((b) طیف محاسبه شده با ضرب کردن f(x, y) در $(-1)^{x+y}$ قبل از محاسبه تبدیل فوریر. فقط یک تناوب سایه‌دار دیده می‌شود زیرا این داده‌ای است که با اجرای معادله f(u, v) به دست می‌آید.

در بحث قبل در مورد تمرکز تبدیلها با ضرب $f(x, y)$ در $(-1)^{x+y}$ فرضیه مهمی است که برای تکمیل شدن در اینجا درج شده است. روش کار در نرم افزار MATLAB آن است که تبدیل را بدون ضرب کردن در $(-1)^{x+y}$ محاسبه می‌کنند و سپس داده‌ها را با استفاده از تابع fftshift دوباره مرتب می‌کنند. این تابع و کاربردهای آن در بخش بعد تشریح شده است.

۴.۲. محاسبه و تجسم تبدیل متمایز ۲ بعدی فوریر در نرم افزار MATLAB

(Computing & Visualizing the 2-D DFT in MATLAB)

تبدیل متمایز فوریر(DFT) و حالت معکوس آن با استفاده از الگوریتم تبدیل سریع فوریر حاصل می‌شوند. FFT یک ارایه تصویر M*N با تابع fft2 در جعبه ابزار که ترکیب ساده دارد به دست می‌آید.

$$F = fft2(f)$$

این تابع یک تبدیل فوریر به اندازه M*N تولید می‌کند و ترتیب داده‌ها به شکل ۴.۲.(a) است یعنی منشاء داده‌ها سمت بالا و چپ است و چهار ربع آن در کانون مستطیل فرکانس یکدیگر را قطع می‌کنند.

همان طور که در بخش ۴.۳.۱ گفتیم وقتی تبدیل فوریر برای فیلترسازی به کار می‌رود باید تصویر ورودی را با صفر لایه‌بندی کرد.

$$F = fft2(f, p, Q)$$

با این ترکیب fft2 داده‌های ورودی را با تعداد مورد نیاز صفر لایه‌بندی می‌کند طوری که تابع حاصل از آن به اندازه P*Q است.

طیف فوریر با استفاده از تابع abs به دست می‌آید:

$$S = abs(F)$$

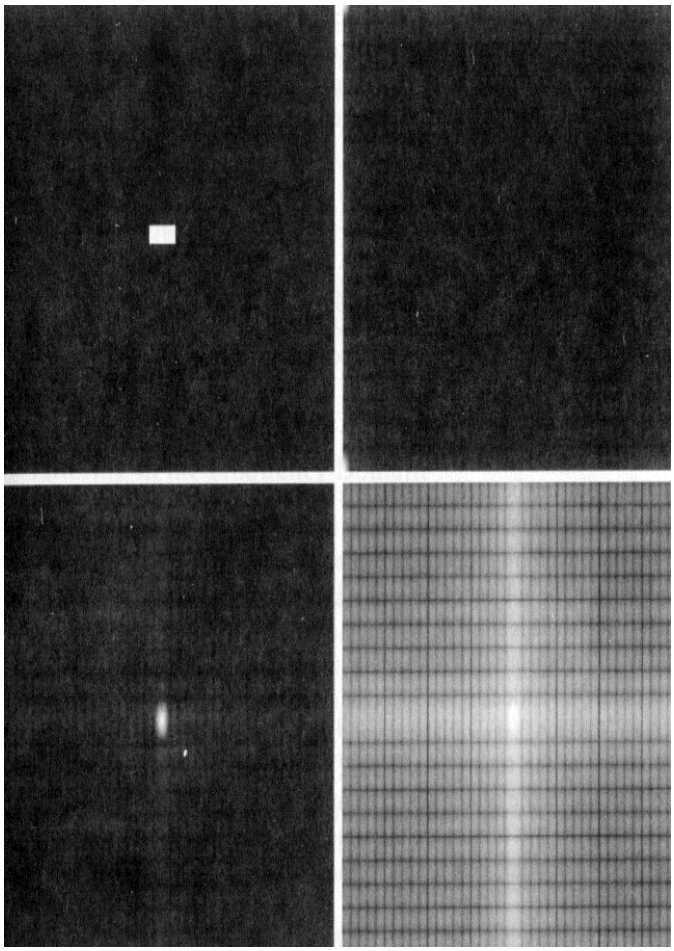
که بزرگی (جذر مجموع جذرهای قسمتهای واقعی و تخیلی) هر عنصر ارایه را محاسبه می‌کند.

تحلیل تصویری طیف با نمایش آن به شکل تصویر جنبه مهمی از کار در دامنه فرکانس است. مثلاً تصویر f در عکس ۴.۳.(a) را در نظر بگیرید. تبدیل فوریر آن را محاسبه می‌کنیم و طیف آن را با مراحل زیر نشان می‌دهیم.

```
>> F = fft2(f);
>> s = abs(F)
>> imshow(s,[ ])
```

نتیجه در تصویر ۴.۳.(b)نشان داده شده است. ۴ نقطه شفاف در گوشه تصویر به علت خاصیت تناوبی هستند که در فصل قبل شرح دادیم. برای انتقال منشاء تبدیل به کانون مستطیل فرکانس از تابع IPT استفاده می‌کنیم. ترکیب آن به شرح زیر است:

$$Fc = fftshift(F)$$



تصویر ۴.۳ (a) یک تصویر ساده (b) طیف فوریر (c) طیف متمرکز (d) طیف تقویت شده با تبدیل

در اینجا f تبدیلی است که با استفاده از fft2 محاسبه شده است و fc تبدیل کانونی است. تابع fftshift با تبادل ربع f کار می‌کند. مثلاً اگر a=[1,2,3,4], fftshift(a)= [4,3,2,1]باشد وقتی پس از محاسبه شدن روی تبدیل اجرا می‌شود. نتیجه خالص استفاده از fftshift همانند وقتی است که تصویر ورودی قبل از محاسبه تبدیل ضربدر $(-1)^{x+y}$ شود. توجه داشته باشید که این دو نوع فرایند را نمی‌توان به جای یکدیگر به کار برد. یعنی اگر 3(0) تبدیل فوریر شناسه باشد رابطه زیر برقرار است:

$3[(-1)^{x+y}f(x,y)]$

که مساوی fftshift(fft1f) است ولی این مقدار معادل fft2(fftshift) نیست.

```
>> FC = fftshift(F);
>> imshow(abc(Fc),[ ])
```

با توجه به تصویر حاصل در عکس ۴.۳ (c)، نتیجه تمرکز در این تصویر مشهود است.گرچه انتقال همان طور که انتظار می‌رفت تکمیل شد، ولی دامنه پویای مقادیر طیف آنقدر زیاد (۰ تا ۲۰۴۰۰۰) است که در قیاس با ۸ بیت نمایشی مقادیر کانونی مسلط بر نتیجه هستند. همان طور که در بخش ۳.۲.۲ گفتیم این مشکل با تبدیل لگاریتم قابل حل است. بنا بر این فرمانهای زیر

$>> S2 = \log\left(1 + abs\left(FC\right)\right):$
$>> imshow(S2\ ,\ [\ \ ]\ )$

منجر به تشکیل تصویر ۴.۳(d)شد.

افزایش جزئیات این تصویر کاملاً عیان است.

تابع ifftshift تمرکز را معکوس‌سازی می‌کند و ترکیب آن به شرح زیر است:

$$F = ifftshift\ \left(FC\right)$$

این تابع را می‌توان برای تبدیل تابعی که در ابتدا روی مستطیل متمرکز است به تابعی که کانون آن در گوشه بالا و سمت چپ است به کار برد. از این خصوصیت در بخش ۴.۴ استفاده کرده‌ایم.

هنگام صحبت در مورد موضوع تمرکز به خاطر داشته باشید که اگر مقدار متغیرهای u, v به ترتیب از ۰ به M-1, N-1 تغییر کند، کانون مستطیل فرکانس در (M/2, N/2) می‌ماند. مثلاً کانون مربع فرکانس ۸ * ۸ در نقطه)۴و۴(است که پنجمین نقطه در امتداد این محور است و از)۰و ۰(شروع می‌شود. اگر این متغیرها مانند نرم افزار MATLAB به ترتیب از 1 به M و ۱ به N تغییر کنند در این صورت، کانون مربع در [(M/2) +1,(N/2) +1] است. در مثال ۸ * ۸ کانون در نقطه (5,5) است. که از (1,1) رو به بالا شمرده می‌شود. هر دو کانون در یک نقطه هستند. ولی وقتی می‌خواهیم موقعیت کانون تبدیل متمایز فوریر(DFT) را در نرم افزار MATLAB مشخص کنیم این موضوع باعث سردرگمی در محاسبات می‌شود.

اگر M و N فرد باشند، کانون محاسبات نرم افزار MATLAB با گرد کردن M/2, N/2 به نزدیکترین عدد صحیح به دست می‌آید. مابقی تحلیل مانند پاراگراف قبل است. مثلاً کانون ناحیه ۷ * ۷ در)۳و ۳(است مشروط بر آن که شمارش از (0,0) شروع شود و اگر از (1,1) شروع می‌کنیم در نقطه (4,4) باشد. در هر یک از این دو مورد کانون چهارمین نقطه از منشاء است. اگر یکی از ابعاد فرد باشد، کانون آن بعد با گرد کردن به روش قبل به دست می‌آید. با استفاده از تابع MATLAB floor, ذکر این نکته که منشاء در (1,1) است کانون مستطیل فرکانس برای محاسبات MATLAB در رابطه زیر مشخص است:

$$\left[Floor\ \left(M/2\right)\ +\ 1\ ,\ floor\left(N/2+1\right)\right]$$

کانون ارائه شده در این اصطلاح برای مقادیر زوج و فرد M و n مصداق دارد.

تبدیل معکوس فوریر با استفاده از تابع ifft2 که ترکیب بنیادی زیر را دارد محاسبه می‌شود.

$$f = ifft2\left(F\right)$$

در اینجا f تبدیل فوریر است و f تصویر حاصل از آن است. اگر داده‌های ورودی مورد نیاز برای محاسبه f واقعی باشند f واقعی می‌شود و معکوس نیز واقعی است. ولی در عمل داده‌های خروجی ifft2 مولفه‌های تخیلی کوچکی دارند که از خطاهای گرد شده که

خصوصیت محاسبات نقطه شناور است نشات می‌گیرند. بنا بر این بهتر است قسمتهای واقعی نتیجه را پس از محاسبه معکوس برای به دست آوردن تصویری که فقط مقادیر واقعی دارد استخراج کرد. این ۲ عملیات را می‌توان ادغام کرد.

```
>> f=real (ifft2(F));
```

همان طور که در مورد پیشرو گفتیم این تابع قالب دیگری به صورت ifft2 (F, P, Q) نیز دارد که f را با صفر لایه‌بندی می‌کند طوری که اندازه آن قبل از محاسبه معکوس P*Q می‌شود. این گزینه در این تحقیق به کار برده نشده است.

۴.۳.فیلتر سازی در دامنه فرکانس (Filtering in the Frequency Domain)

فرضیه فیلترسازی در این دامنه ساده است. در این بخش فرضیه‌های فیلترسازی دامنه فرکانس و اجرای آنها در نرم افزار MATLAB را به طور خلاصه شرح می‌دهیم:

۴.۳.۱. فرضیه‌های بنیادین(Fundamental Concept)

اساس فیلترسازی خطی در دامنه فرکانس و فضا قضیه تلفیق است که ممکن است به شکل زیر نوشته شود:

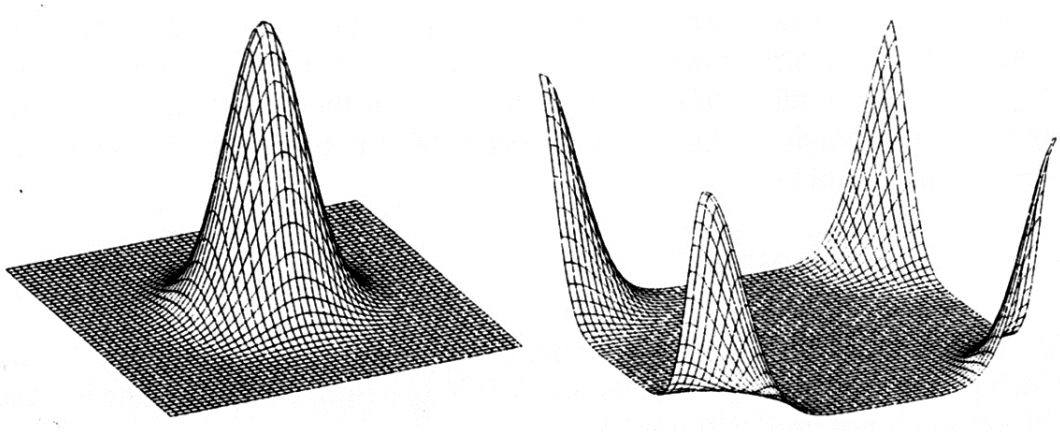
$$f(x,y) * h(h,y) \Leftrightarrow H(u,v)F(u,v)$$
$$f(x,y)h(h,y) \Leftrightarrow H(u,v) * G(u,v)$$

در اینجا * نمایانگر تلفیق دو تابع است و عبارتهای کنار فلشهای دوگانه یک جفت تبدیل فوریر را ایجاد می‌کند. مثلاً اصطلاح اول نشان می‌دهد که تلفیق دو تابع فضایی با محاسبه تبدیل معکوس فوریر و محصول تبدیل فوریر دردو تابع حاصل می‌شود. تبدیل پیشروی فوریر در تلفیق دو تابع فضایی محصول تبدیل دو تابع را ارائه می‌دهد. توضیحات مشابه در مورد عبارت دوم نیز مصداق دارد.

در فیلترسازی به دو اصطلاح اول توجه می‌کنیم. فیلترسازی در دامنه فضایی شامل تلفیق تصویر f(x, y) با نقاب(Mask) فیلتر h(x, y) می‌شود تلفیق فضایی خطی دقیقاً همان است که در بخش ۳.۴.۱. توضیح داده شد.

طبق قضیه تلفیق می‌توان همین نتایج را با ضرب f(u,v) در h(u,v) که تبدیل فوریر در فیلتر فضایی است در دامنه فرکانس به دست آورد. H(u, v) را تابع انتقال فیلتر(filter transfer fuction) می‌نامیم.

نکته اصلی در فیلترسازی دامنه فرکانس انتخاب تابع انتقال فیلتر است به گونه‌ای که f(u, v) را به شکلی خاص تغییر دهد.



تصویر ۴.۴: تابع‌های انتقال ((a)) فیلتر پایین‌گذر متمرکز ((b) قالب مورد نیاز برای فیلترسازی تبدیل متمایز فوریر(DFT) توجه داشته باشید که اینها فیلترهای دامنه فرکانس هستند.

مثلاً فیلتر تصویر ۴.۴.(a) یک تابع انتقالی دارد که وقتی ضربدر کانون f(u, v) شود مولفه‌های فرکانس بالای f(u,v) را تعدیل می‌کند ولی فرکانسهای پایین را نسبتاً بدون تغییر می‌گذارد. فیلترهایی با این خصوصیت فیلترهای پایین‌گذر نامیده می‌شوند. همان طور که در بخش ۴.۵.۲. گفتیم نتیجه فیلترسازی پایین‌گذر تار شدن تصویر است. تصویر ۴.۴.(b)همان فیلتر را پس از پردازش شدن با fftshift نشان می‌دهد. این قالب فیلتر زمانی استفاده می‌شود که با فیلترهای دامنه فرکانسی سر و کار داشته باشیم که تبدیل فوریر داده‌های آنها متمرکز نباشد.

بر اساس قضیه تلفیق می‌دانیم که برای به دست آوردن تصویرهای فیلتر شده در دامنه فضایی تبدیل معکوس فوریر در محصول H(u, v), F(u, v) را محاسبه می‌کنیم.

نکته مهم آن است که فرایندی که تشریح کردیم همانند چیزی است که با استفاده از تلفیق در دامنه فضایی به دست آورده می‌شود مشروط بر آن که نقاب(Mask) فیلتر H(x, y) تبدیل معکوس فوریر H(u, v) باشد. در عمل برای ساده کردن تلفیق فضایی از نقاب(Mask) های کوچکی برای ضبط خصوصیات برجسته همتاهای دامنه فرکانس استفاده می‌کنیم.

همان طور که در بخش ۴.۱ گفتیم، اگر برای اجرای فیلترسازی از تبدیل متمایز فوریر(DFT) استفاده کنیم، تصویرها و تبدیلهای آنها خود به خود متناوب هستند. به راحتی می‌توان تجسم کرد که تابعهای تناوبی تلفیق شده می‌توانند باعث تداخل تناوبهای مجاور شوند مشروط بر آن که تناوبها با توجه به مدت قسمتهای غیرصفر تابع نزدیک به هم باشند.

این تداخل را خطای مداری(wraparound error) می‌نامیم. اگر تابع به شکل زیر با صفر لایه‌بندی شود جلوی این خطا گرفته می‌شود.

فرض کنید تابع‌های F(x, y), H(x, y) به ترتیب به اندازه A*B,C*D باشند. با الحاق صفر به f و g دو تابع لایه‌دار گسترده(extended(padded)function) به اندازه P*Q درست می‌کنیم. با انتخاب رابطه زیر می‌توان جلوی خطای مداری را گرفت:

$$P \geq A + C - 1$$
$$Q \geq B + D - 1$$

اکثر تحقیقات این فصل مربوط به تابعهایی با اندازه یکسان M*N است که مقادیر لایه‌بندی زیر را استفاده می‌کنیم:

تابع زیر به نام paddedsize حداقل مقدار فرد مقادیر P و Q را برای تحقق معادلات قبل محاسبه می‌کند. گزینه‌ای برای لایه‌بندی داده‌های ورودی و تشکیل تصویرهای چهارگوش به اندازه معادل نزدیکترین عدد صحیح به قوه ۲ دارد. زمان اجرای الگوریتمهای FFt تقریباً بستگی به تعداد عوامل اصلی در P و Q دارد. این الگوریتمها زمانی که P و Q توان ۲ داشته باشد سریع‌تر زمانی هستند که آنها عدد اول باشند. در عمل توصیه می‌شود با فیلترها و تصویرهای چهارگوش کار کنید تا فیلترسازی در هر دو جهت یکسان باشد. با تابع paddedsize می‌توان با انتخاب داده‌های ورودی این کار را با انعطاف پذیری انجام داد.در تابع paddedsize بردارهای AB,CD,PQ به ترتیب عناصر [A B][C D][P Q] را دارند. این مقادیر همان هستند که در بالا تعریف شدند.

```
function PQ = paddedsize (AB, CD, PARAM)
%PADDEDSIZE Computes padded sizes useful for FFT-based filtering.
%     PQ = PADDEDSIZE(AB), where AB is a two-element size vector,
%     Computes the two-element size vector PQ = 2*AB.
%
%     PQ = PADDEDSIZE(AB, 'PWR2') computes the vector PQ such that
%     PQ(1) = PQ(2) = 2^nextpow2(2*m), where m is MAX(AB).
%
%     PQ = PADDEDSIZE(AB, CD), where AB and CD are two-element size
%     vectors, computes the tow-element size vector PQ. The elements
%     of PQ are the smallest even integers greater than or equal to
%     AB + CD - 1.
%
%     PQ = PADDEDSIZE(AB, CD, 'PWR2') computes the vector PQ such that
%     PQ(1) = PQ(2) = 2^nextpow2(2*m), where m is MAX([AB CD]).
if nargin == 1
   PQ = 2*AB;
elseif nargin == 2 & -ischar(CD)
   PQ = AB + CD -1;
   PQ = 2 * ceil(PQ / 2);
elseif nargin == 2
  m = max(AB); % Maximum dimension.

  % Find power-of-2 at least twice m.
  P = 2^nextpow2(2*m);
  PQ = [p, p];
elseif nargin == 3
  m = max([AB CD]); % Maximum dimension.
  P = 2^nextpow2(2*m);
  PQ = [p, p];
else
  error('Wrong number of inputs.')
end
```

وقتی P Q به این شکل با تابع paddedsize محاسبه شود برای محاسبه FFT با استفاده از لایه‌بندی صفر از ترکیب زیر برای fft2 استفاده می‌کنیم.

```
F = fft2(f, PQ(1), PQ(2))
```

این ترکیب صرفاً صفرهای کافی را به f وصل می‌کند. طوری که تصویر حاصل از آن به اندازه PQ(1)*PQ(2)می‌شود و بعد FFT را همان طور که قبلاً گفتیم محاسبه می‌کند. توجه داشته باشید که وقتی از لایه‌بندی استفاده می‌کنید تابع فیلترسازی در دامنه فرکانس باید به اندازه PQ(1)*PQ(2)باشد.

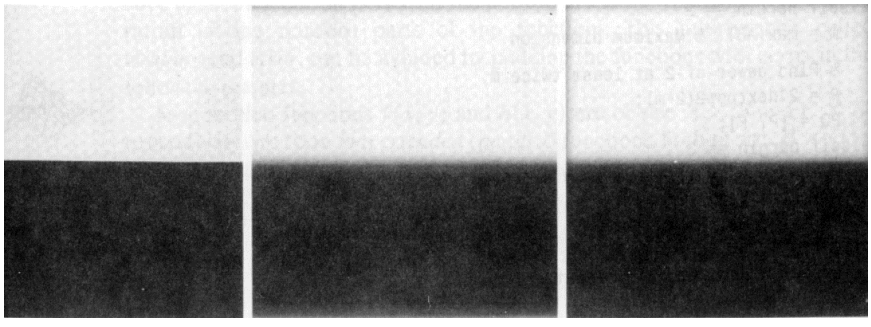
مثال ۴.۱: تاثیرات فیلترسازی با و بدون لایه‌بندی

تصویر f در عکس ۴.۵.(a) در این مثال برای نشان دادن اختلاف بین فیلترسازی با و بدون لایه‌بندی به کار برده شده است. در این مبحث برای ایجاد فیلترهای پایین گذر گوسی از تابع Ipfilter مشابه تصویر ۴.۴.(b)با مقدار مشخص sigma استفاده می‌کنیم. این کاربرد به طور مفصل در بخش ۴.۵.۲ تشریح شده است. ولی ترکیب آن ساده است بنا بر این آن را در اینجا استفاده می‌کنیم و Ipfilter را در آن بخش بیشتر شرح خواهیم داد.

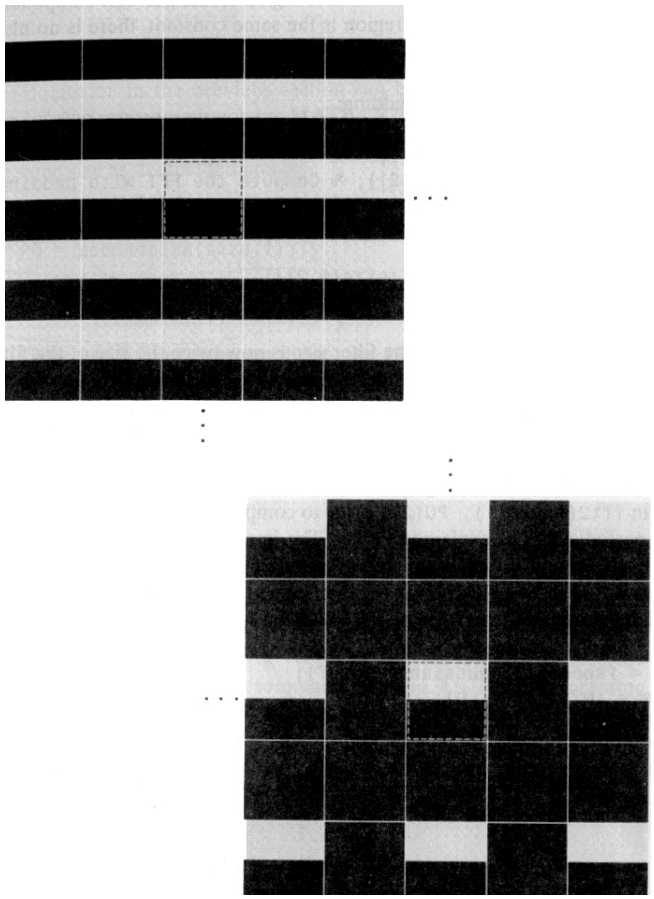
فرمانهای زیر فیلترسازی را بدون لایه‌بندی انجام می‌دهند:

```
>> [M, N] = size(f);
>> F = fft2(f);
>> sig = 10;
>> H = lpfilter('gaussian', M, N, sig);
>> G = H.*F;
>> g = real(ifft2(G));
>> imshow(g, [ ])
```

در عکس ۴.۵.(b)تصویر gرا می‌بینید. همان طور که انتظار می‌رود تصویر تار است. ولی توجه داشته باشید که لبه‌های عمودی آن تار نیستند. علت آن را می‌توان با تصویر ۴.۶.(a) شرح داد که میزان تناوب در محاسبات تبدیل متمایز فوریر(DFT) رایانه‌ای به صورت گرافیکی نشان داده شده است.



تصویر ۴.۵: ((a)) یک تصویر ساده به اندازه ۲۵۶*۲۵۶ ((b) تصویر عبور داده شده از فیلتر پایین‌گذر در ادامه فرکانس بدون لایه‌بندی (c) تصویر عبور داده شده از فیلتر پایین‌گذر در ادامه فرکانس با لایه‌بندی قسمتهای روشن لبه‌های عمودی بی و (c)را مقایسه کنید.



تصویر ۴.۶: (a) ترتیب تناوب نامحدود تصویر در عکس ۴.۵.(a) قسمت خط چین‌دار نمایانگر داده‌های پردازش شده با fft2 است.(H) همان ترتیب تناوب‌دار بعد از لایه‌بندی با صفرها خطوط سفید باریک تصویرها برای راحتی نشان داده شده است. آنها بخشی از داده‌ها نیستند.

خطوط سفید باریک بین تصویرها برای راحتی دید درج شده است. آنها بخشی از داده‌ها نیستند. خط چینها برای مشخص کردن تصویر M*N پردازش شده با fft2 به کار برده شده است. فرض کنید یک فیلتر تار کننده با این ترتیب تناوب نامحدود تلفیق شود. پر واضح است که وقتی فیلتر از بالای تصویر خط‌چین‌دار می‌گذرد بخشی از تصویر و قسمت انتهای مولفه تناوبی بالای آن را نیز در بر می‌گیرد. بنا بر این وقتی یک ناحیه تیره و روشن روی فیلتر قرار می‌گیرد، نتیجه نیمه خاکستری می‌شود. این دقیقاً همان چیزی است که در قسمت فوقانی تصویر ۴.۵.(b)دیده می‌شود.

وقتی فیلتر سمت چپ تصویر خط‌چین‌دار باشد، با نقطه‌ای همانند آن در مولفه تناوبی مواجه می‌شود. از آنجائی که میانگین ناحیه ثابت تغییرناپذیر است این قسمت از نتایج تار نمی‌شود. سایر قسمتهای تصویر در عکس ۴.۵.(b)به شکلی مشابه تشریح شده است.

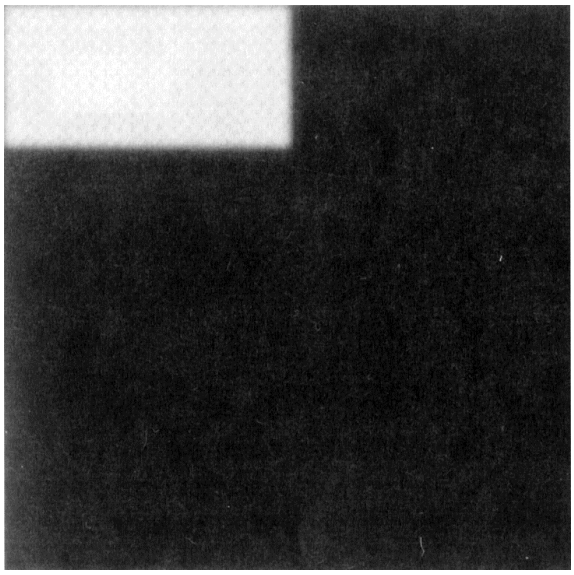
حالا فیلترسازی با لایه‌بندی زیر را در نظر بگیرید:

```
>> PQ = paddedsize(size(f));
>> Fp = fft2(f, PQ(1), PQ(2)); % Compute the FFT with padding.
>> Hp = lpfilter('gaussian', PQ(1), PQ(2), 2*sig);
>> Gp = Hp.*Fp;
>> gp = real(ifft2(Gp));
>> gpc = gp(1:size(f,1), 1:size(f,2));
>> imshow(gp, [ ])
```

در اینجا از 2sig استفاده کردیم زیرا اندازه فیلتر ۲ برابر اندازه فیلتر مورد استفاده بدون لایه‌بندی است.

نتیجه لایه‌بندی کامل P g در تصویر ۴.۷ دیده می‌شود. نتیجه نهایی در تصویر ۴.۵.(c)با برش تصویر ۴.۷ به اندازه اصلی به دست آمد (به فرمان یکی مانده به آخر نگاه کنید). این نتیجه را می‌توان به کمک تصویر ۴.۶.(b)نشان داد در اینجا تصویر خط‌چین‌دار با لایه‌های صفر همان طور که به صورت داخلی با fft2(f,PQ(1),PQ(2)) قبل از محاسبه تبدیل برقرار شده است دیده می‌شود. این تناوب را قبلاً شرح دادیم. حالا تصویر یک مرز مشکی در دورتادور خود دارد بنا بر این تلفیق فیلتر هموارساز با این ترتیب نامحدود باعث نمایان شدن قسمت تار خاکستری در لبه‌های روشن تصویر می‌شود. با اجرای این فیلترسازی فضایی می‌توان نتایج مشابهی به دست آورد.

```
>> h = fspecial('gaussian', 15, 7);
>> gs = imfilter(f, h);
```



تصویر ۴.۷: تصویر لایه‌بندی شده کامل حاصل از ifft2 بعد از فیلترسازی. این تصویر به اندازه ۵۱۲*۵۱۲ پیکسل است.

از بخش ۳.۴.۱ به خاطر داریم که در تابع imfilter مرز تصویر با صفرهای پیش فرض لایه‌بندی می‌شود.

۴.۳.۲. اقدامات اساسی در فیلترسازی تبدیل متمایز فوریر(DFT) (Basic Steps in DFT Filtering)

مباحث قبل را می‌توان به شکل زیر خلاصه کرد که شامل تابع‌های MATLAB باشد. f تصویری است که باید فیلتر شود. gنتیجه است. چنین تصور می‌شود که تابع فیلتر H(u, v) هم اندازه تصویر لایه‌بندی شده است.

۱. پارامترهای لایه‌بندی را با استفاده از تابع paddedsize به دست آورید.

```
PQ = paddedsize(size(f));
```

۲. تبدیل فوریر را با لایه‌بندی به دست آورید.

```
F = fft2(f, PQ(1), PQ(2));
```

۳. با استفاده از هر یک از روشهای بحث شده در مابقی این فصل یک تابع فیلتر H به اندازه pq1*pq2 ایجاد کنید. این فیلتر باید در قالب تصویر نشان داده شده در عکس ۴.۴.(b)باشد. در عوض در کانون عکس ۴.۴.(a) قرار داده شده است. قبل از استفاده از فیلتر فرض کنید h=fftshift است.

۴. تبدیل را ضربدر فیلتر کنید.

```
G = H.*F;
```

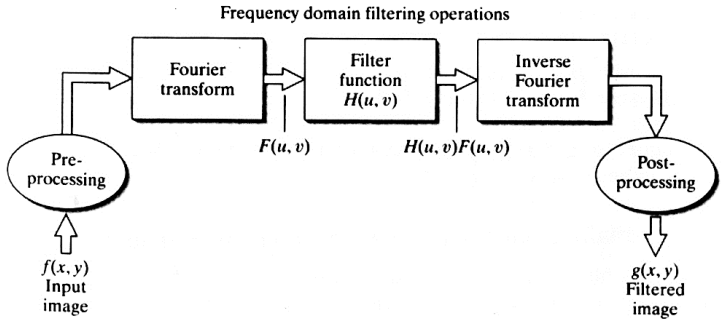
۵. قسمت واقعی معکوس fft gرا به دست آورید.

```
g = real(ifft2(G));
```

۶. گوشه بالا و سمت چپ مستطیل را ببرید تا اندازه واقعی به دست آید:

```
g = g(1:size(f, 1), 1:size(f, 2));
```

فرایند فیلترسازی در تصویر ۴.۸ خلاصه شده است. مرحله قبل از پردازش ممکن است شامل روندی همچون تعیین اندازه تصویر، به دست آوردن پارامترهای لایه‌بندی، و ایجاد فیلتر باشد. اقدامات پس از پردازش شامل محاسبه قسمت واقعی نتیجه، برش تصویر، و تبدیل آن به نوع uint8 یا uint16 برای ذخیره سازی می‌شوند.



تصویر ۴.۳.: مراحل اصلی فیلترسازی در دامنه فرکانس

تابع فیلتر H(u, v) در تصویر ۴.۸ قسمتهای تخیلی و واقعی F(u,v) را ضرب می‌کند. اگر H(u, v) واقعی باشد، در این صورت، مرحله نتیجه تغییر نمی‌کند. این حقیقت در معادله مرحله بخش ۴.۱ دیده می‌شود. اگر مضرب قسمتهای واقعی و تخیلی معادل باشند، همدیگر را خنثی می‌کنند و زاویه مرحله تغییر نمی‌کند. فیلترهایی که به این شکل عمل می‌کنند فیلترهای مرحله صفر(zero-phase shift filter) نامیده می‌شوند. اینها تنها فیلترهای خطی هستند که در این فصل بررسی کردیم.

از فرضیه سیستم‌های خطی مشخص می‌شود که در برخی شرایط عادی داده‌های ورودی یک سیستم خطی می‌تواند خصوصیات سیستم را کاملاً تعیین کند. هنگام کار با داده‌های محدود متمایز واکنش یک سیستم خطی و واکنش آن به ضربه محدود است. اگر سیستم خطی یک فیلتر فضایی باشد، می‌توان با مشاهده واکنش آن به ضربه خصوصیات آن را کاملاً تعیین کرد. فیلتری که به این شکل تعیین شود فیلتری با واکنش محدود به ضربه(finite-impulse-response(FIR)filter) نامیده می‌شود. کلیه فیلترهای فضایی خطی در این تحقیق فیلترهایی با واکنش محدود به ضربه هستند.

۴.۳.۳. تابع M برای فیلترسازی در دامنه فرکانس(An M-function for filtering in the frequency Domain)

ترتیب مراحل فیلترسازی که در بخش قبل تشریح شد در تمام این فصل و فصل بعد استفاده می‌شود. بنا بر این یک تابع M داریم که تصویر و تابع فیلتر را به صورت داده‌های ورودی قبول می‌کند، از عهده کلیه جزئیات فیلترسازی برمی‌آید، و تصویر بریده شده و فیلتر شده را در داده‌های خروجی قرار می‌دهد. تابع زیر این کار را می‌کند.

```
function g = diftfilt(f, H)
%DFTFILT Performs frequency domain filtering.
%     G = DFTFILT(F, H) filters F in the frequency domain using the
%     filter transfer function H. The output, G, is the filtered
%     image, which has the same size as F. DFTFILT automatically pads
%     F to be the same size as H. Function PADDEDSIZE can be used
%     to determine an appropriate size for H.
%
%     DFTFILT assumes that F is real and that H is a real, uncentered,
%     circularly-symmetric filter function.

%   Obtain the FFT of the padded input.
F = fft2(f, size(H, 1), size(H, 2));

%   perform filtering.
g = real(ifft2(H.*F));

% Crop to original size.
g = g(1:size(f, 1), 1:size(f, 2));
```

شگردهای ایجاد فیلترهای دامنه فرکانس در ۳ بخش بعد تشریح شده است.

۴.۳. به دست آوردن فیلترهای دامنه فرکانس از فیلترهای فضایی

(Obtaining Frequency Domain Filter from Spatial filter)

به طور کلی وقتی فیلترها کوچک باشند، فیلترسازی در دامنه فضایی از نظر محاسباتی کارآمدتر از فیلترسازی دامنه فرکانس است. تعریف کلمه کوچک(small) در اینجا مشکل است که پاسخ آن بستگی به عواملی همچون ماشین و الگوریتمهای مورد استفاده و مسائلی مانند اندازه حافظه میانجی، نحوه مدیریت داده‌های پیچیده، و برخی عوامل دیگر دارد. بریگم با مقایسه آنها با استفاده از تابع ۱ بعدی ثابت کرد که فیلترسازی با استفاده از الگوریتمFFT می‌تواند در زمانی که تابعها در ردیف ۳۲ نقطه باشند، سریعتر از اجرای فضایی باشد. بنا بر این ارقام مورد نظر زیاد نیستند. دانستن نحوه تبدیل فیلتر فضایی به فیلتر دامنه فرکانس مفید است تا بتوانیم این دو روش را به شکلی معنی‌دار مقایسه کنیم.

یک روش برای ایجاد فیلتر دامنه فرکانس H که مطابق با فیلتر فضایی H است آن است که فرض کنیمH= fft2(f,PQ(1),PQ(2)) که مقادیر بردار P Q بستگی به اندازه تصویری که می‌خواهیم فیلتر کنیم دارد که در بخش آخر آن را بحث کردیم. در این بخش به دو موضوع اصلی می‌پردازیم:

۱) نحوه تبدیل فیلترهای فضایی به فیلترهای معادل دامنه فرکانس .

۲) نحوه مقایسه نتایج فیلترسازی دامنه فرکانس با استفاده از تابع imfilter و فیلترسازی دامنه فرکانس با استفاده از شگردهای فصل قبل.

همان طور که مفصلاً در بخش ۳.۴.۱. گفتیم imfilter از همبستگی استفاده می‌کند و منشاء این فیلتر در کانون است. برای معادل کردن این ۲ روش باید داده‌ها پردازش شوند. جعبه ابزار تابع freqz2 را ایجاد می‌کند که دقیقاً این کار را می‌کند و فیلتر مطابق با آن را در دامنه فرکانس قرار می‌دهد.

تابع freqz2 واکنش فرکانس فیلترهای دارای واکنش محدود به ضربه(FIR) را محاسبه می‌کند که همان طور که در انتهای بخش ۴.۳.۲. گفتیم تنها فیلترهای خطی بررسی شده در این تحقیق هستند. نتیجه ایجاد فیلتر مطلوب در دامنه فرکانس است. ترکیب جالب در این بحث به شرح زیر است:

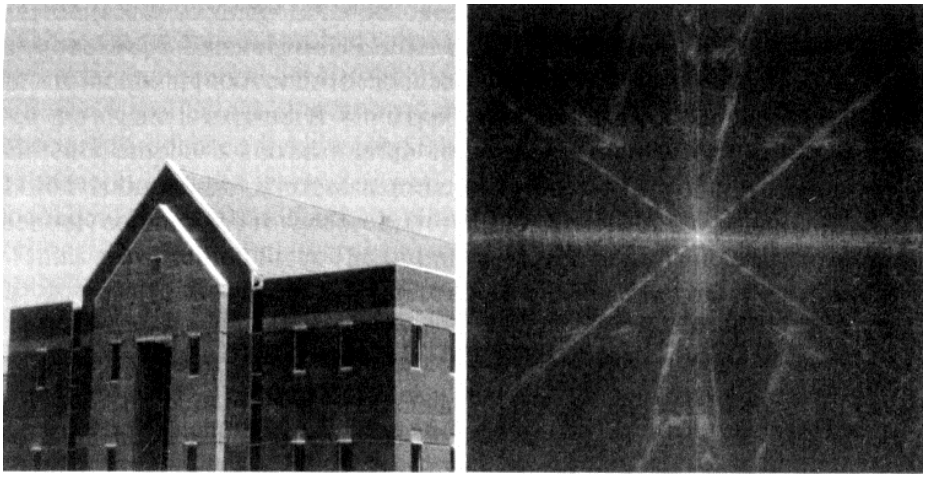
```
H = freqz2(h, R, C)
```

در اینجا H یک فیلتر فضایی ۲ بعدی است و H فیلتر دامنه فرکانس ۲ بعدی مطابق با آن است. آر تعداد ردیف است. C تعداد ستونی است که می‌خواهیم فیلتر H داشته باشد. معمولاً فرض می‌کنیم r=pq1, c=pq2 است که در بخش ۴.۳.۱ شرح دادیم. اگر freqz2

بدون شناسه خروجی نوشته شود قدر مطلق H در نرم افزار MATLAB به صورت یک طرح ۳ بعدی ترسیم می‌شود. اصول مکانیکی مورد استفاده در تابع freqz2 با یک مثال تشریح شده است.

تصویر f به اندازه ۶۰۰ * ۶۰۰ عنصر در عکس ۴.۹.(a) در نظر بگیرید. فیلتر دامنه فرکانس H را مطابق با فیلتر فضایی sobel که لبه‌های عمودی را بهبود می‌بخشد ایجاد می‌کنیم. (جدول ۳.۴). سپس نتیجه فیلترسازی f را در دامنه فضایی با نقاب(Mask) sobel (با استفاده از imfilter) با نتایج به دست آمده از اجرای فرایند معادل در دامنه فرکانس مقایسه می‌کنیم. فیلترسازی با فیلتر کوچکی مانند نقاب(Mask) sobel عملاً در دامنه فضایی اجرا می‌شود. ولی این فیلتر را برای نمایش انتخاب کردیم چون که ضرایب آن ساده هستند و مقایسه نتایج فیلترسازی آسان است. فیلترهای فضایی بزرگ نیز به همین شکل بررسی می‌شوند.

a b



تصویر ۴.۹: ((a)) یک تصویر مقیاس خاکستری ((b) طیف فوریر آن

تصویر ۴.۹(b)تصویری از طیف فوریر f به شرح زیر است:

```
>> F = fft2(f);
>> S = fftshift(log(1 + abs(F)));
>> S = gscale(S);
>> imshow(S)
```

سپس فیلتر فضایی را با استفاده از تابع fspecial ایجاد می‌کنیم:

```
h = fspecial('sobel')'
h =
      1     0     -1
      2     0     -2
      1     0     -1
```

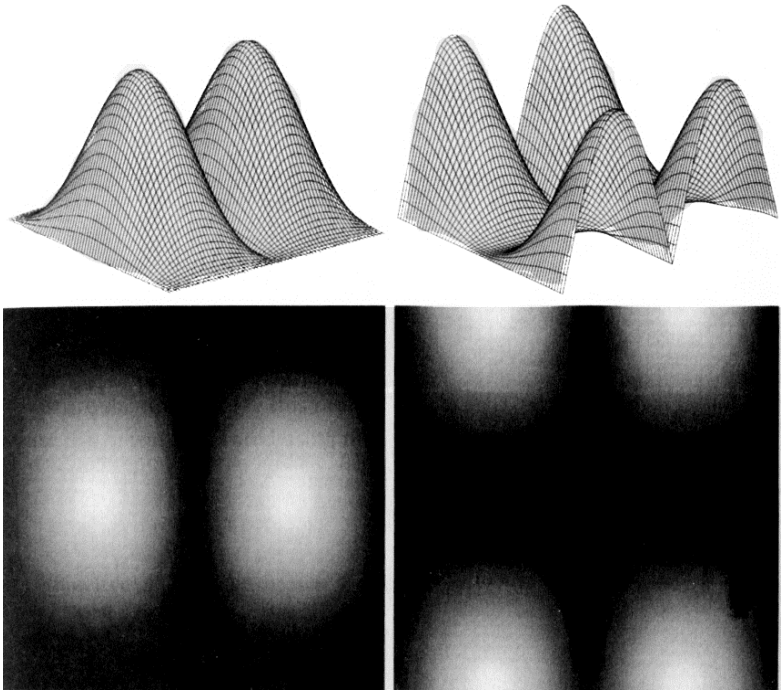
برای مرور طرح فیلتر دامنه فرکانس حروف زیر را تایپ می‌کنیم:

```
>>  freqz2(h)
```

نتیجه با محورهای حذف شده در تصویر ۴.۱۰.(a) دیده می‌شود. (شگردهای به دست آوردن ترسیمهای چشم انداز در بخش ۴.۵.۳ تشریح شده است) این فیلتر با استفاده از فرمانهای زیر به دست آمد:

```
>> PQ = paddedsize(size(f));
>> H = freqz2(h, PQ(1), PQ(2));
>> H1 = ifftshift(H);
```

در اینجا همان طور که قبلاً گفتیم برای مرتب کردن داده‌ها به ifftshift نیاز داریم. تا منشاء آن در بالا و سمت چپ مستطیل فرکانس باشد. ترسیم abs(H1) در تصویر ۴.۱۰.(b)دیده می‌شود. مقادیر مطلق H و H 1 در تصویری که با فرمانها نمایش داده شده در تصویر ۴.۱۰.س و د دیده می‌شود.



a b
c d

تصویر ۴.۱۰. ((a)) مقدار مطلق فیلتر دامنه فرکانس مطابق با نقاب(Mask) sobel عمودی ((b) همان فیلتر بعد از پردازش با fttshift ارقام (c)و (d)فیلترهایی در (e)و بی هستند که به صورت تصویر نشان داده شده است.

```
>> imshow(abs(H), [ ])
>> figure, imshow(abs(H1), [ ])
```

سپس تصویرهای فیلتر شده را ایجاد می‌کنیم. در دامنه فضایی از عبارت زیر استفاده می‌کنیم:

```
>> gs = imfilter(double(f), h);
```

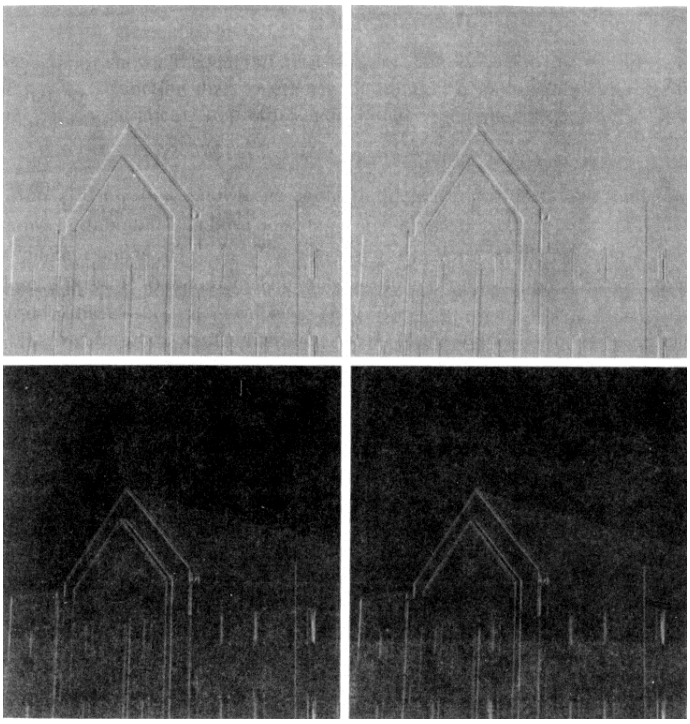
که مرز تصویر را به طور پیش فرض با صفرها لایه‌بندی می‌کند. تصویر فیلتر شده‌ای که با پردازش دامنه فرکانس به دست آورده می‌شود به شرح زیر است:

```
gf = dftfilt(f, H1);
```

نتیجه فرمانها در تصویرهای ۴.۱۱(a) و (b)دیده می‌شود.

```
>> imshow(gs, [ ])
>> figure, imshow(gf, [ ])
```

رنگ خاکستری تصاویر به آن دلیل است که gs و gf مقادیر منفی دارند که باعث می‌شود میانگین مقدار تصاویر توسط فرمان imshow زیاد شود .



a b

c d

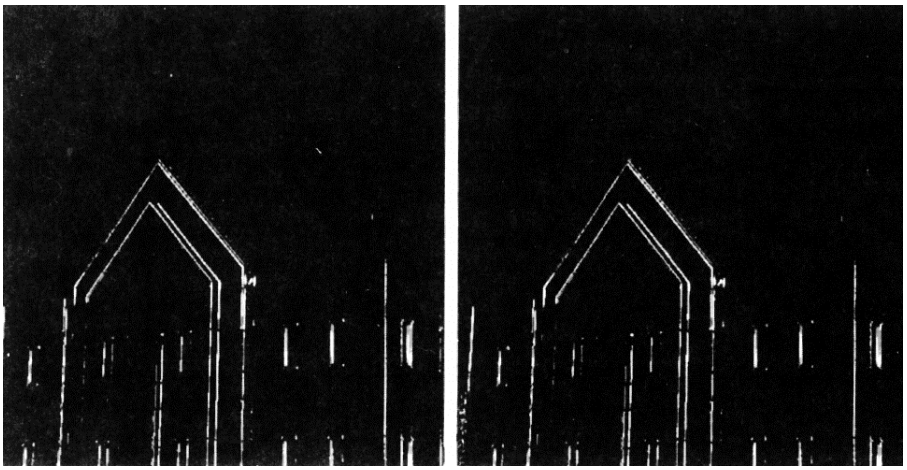
نقاب(Mask) H , sobel که در بالا ایجاد شده است برای کشف لبه‌های عمودی تصویر با استفاده از قدر مطلق واکنش به کار برده می‌شود. بنا بر این بهتر است قدر مطلق تصویرهای محاسبه شده نشان داده شود. تصویرهای به دست آمده با فرمانها در عکسهای ۴.۱۱.(c)و (d)دیده می‌شوند.

```
>> figure, imshow(abs(gs), [ ])
>> figure, imshow(abs(gf), [ ])
```

با ایجاد یک تصویر دودویی آستانه می‌توان تصویرها را بهتر دید.

```
>> figure, imshow(abs(gs) > 0.2*abs(max(gs(:))))
>> figure, imshow(abs(gf) > 0.2*abs(max(gf(:))))
```

مضرب ۰.۲ اختیاری انتخاب شد تا لبه‌هایی که استحکام آنها بیشتر از ۲۰٪ حداکثر مقدار gS و g f است نشان داده شوند. نتایج در تصاویر ۴.۱۲.(a) و (b)دیده می‌شود.



a b

تصویر ۴.۱۲. حد آستانه تصاویر ۴.۱۱.(c)و (d)لبه‌های اصلی را واضح‌تر نشان می‌دهد.

تصویرهای به دست آمده با استفاده از فیلتر دامنه فرکانس و فضایی برای تمام موارد کاربردی یکسان هستند. با محاسبه تفاوت آنها این حقیقت تایید می‌شود:

```
>> d = abs(gs – gf);
>> max(d(:))
asn =
      5.4015e-012
>> min(d(:))
ans =
      0
```

روش تشریح شده برای اجرای روش فیلترسازی فضایی در دامنه فرکانس در بخشهای ۳.۴.۱ و ۳.۵.۱ و فیلتر فضایی واکنش محدود به ضربه با اندازه اختیاری به کار برده می‌شود.

۴.۵.□. ایجاد مستقیم فیلتر در دامنه فرکانس(Generating filter Directly in frequency Domain)

در این بخش نحوه اجرای مستقیم تابعهای فیلتر در دامنه فرکانس را نشان می‌دهیم. روی فیلترهای متقارنی متمرکز می‌شویم که تابعهای گوناگون فاصله از منشاء تبدیل هستند. تابع M برای اجرای این فیلترها مبنایی است که برای سایر تابعها نیز به کار برده می‌شود. کار را در ابتدا با چند فیلتر پایین‌گذر هموار شروع می‌کنیم، سپس نشان می‌دهیم چگونه می‌توان از قابلیت‌های ترسیم سطحی و نمایش لبه‌ها به صورت خط برای تجسم فیلترها استفاده کرد.در آنتها نیز فیلترهای بالاگذر شفاف کنده را بررسی می‌کنیم.

۴.۵.۱. ایجاد آرایه‌های شبکه‌دار برای اجرای فیلترهایی در دامنه فرکانس

(Dreating Meshing Arrays for use in Implementing filter in the frequency Domain)

نکته اصلی در تابع M محاسبه تابعهای مسافت از هر نقطه تا یک نقطه خاص در مستطیل فرکانس است. طبق محاسبات FFT در نرم افزار MATLAB فرض می‌شود که منشاء تبدیل در قسمت بالا و سمت چپ مستطیل فرکانس است. محاسبات مسافت ما با توجه به این نقطه است. این داده‌ها را می‌توان برای تجسم مرتب کرد. (تا مقدار منبع به کانون مستطیل فرکانس با استفاده از تابع fftshift تبدیل شود.

تابع M زیر که آن را dftuv می‌نامیم آرایه شبکه‌دار لازم را برای محاسبه مسافت و سایر کاربردهای مشابه ایجاد می‌کند.

آرایه‌های شبکه‌دار ایجاد شده با dftuv برای پردازش با fft2, ifft2 مرتب هستند و نیازی به مرتب شدن ندارند.

```
function [U, V] = dfutv(M, N)
%DFTUV Computes meshgrid frequency matrices.
%     [U, V] = DFTUV(M, N) computes meshgrid frequency matrices U and
%     V. U and V are useful for computing frequency-domain filter
%     function that can be used with DFTFILT. U and V are both
%     M-by-N.

% Set up range of variables.
u = 0:(M – 1);
v = 0:(N – 1);

% Compute the indices for use in mershgrid.
idx = find(u > M/2);
u(idx) = u(idx) – M;
idx = find(v > N/2);
v(idx) = v(idx) – N;

% Compute the arrays.
[V, U] = mershgrid(v, u);
```

فرمانهای زیر جذر مسافت را از هر نقطه در مستطیلی با اندازه ۸*۵ تا مبداء آن نشان می‌دهد.

```
>> [V, U] = dftuv(8, 5);
>> D = U.^2 + V.^2
D =
      0     1     4     4     1
      1     2     5     5     2
      4     5     8     8     5
      9     10    13    13    10
      16    17    20    20    17
      9     10    13    13    10
      4     5     8     8     5
      1     2     5     5     2
```

توجه داشته باشید که مسافت در قسمت بالا و چپ صفر است و مسافتهای زیاد در کانون مستطیل فرکانس هستند. طبق قالب تشریح شده در عکس ۴.۲.(a) می‌توان از تابع fftshift برای به دست آوردن مسافت با توجه به کانون مستطیل فرکانس استفاده کرد.

```
>> fftshift(D)
ans =
      20    17    16    17    20
      13    10     9    10    13
       8     5     4     5     8
       5     2     1     2     5
       4     1     0     1     4
       5     2     1     2     5
       8     5     4     5     8
      13    10     9    10    13
```

فاصله در مختصات (5,3) صفر است. و آرایه در این نقطه متقارن است.

۴.۵.۲. فیلترهای دامنه فرکانس پایین‌گذر (Lowpass Frequency Domain Filter)

یک فیلتر پایین‌گذر مطلوب تابع انتقال زیر را دارد:

$$H(u,v)=\begin{cases}1 & if \quad D(u,v)\le D_0\\ 0 & if \quad D(u,v)>D_0\end{cases}$$

در اینجا D_0 یک مقدار غیرمنفی است. D(u,v) فاصله نقطه (u,v) تا کانون فیلتر است. مکان هندسی نقاطی که $D(u,v)=D_0$ است دایره است. فیلتر H تبدیل فوریر تصویر را ضرب می‌کند. یک فیلتر مطلوب کلیه مولفه‌های f را که خارج از دایره قرار دارند صفر می‌کند و مولفه‌هایی را که رو یا داخل دایره هستند تغییر نمی‌دهد. گرچه این فیلتر را نمی‌توان با مولفه‌های الکترونیکی به شکل آنالوگ تحقق بخشید ولی در رایانه می‌توان با تابع انتقال قبل آن را شبیه‌سازی کرد. خصوصیات فیلترهای مطلوب برای تشریح پدیده‌هایی همچون خطای مداری مفید هستند.

یک فیلتر پایین‌گذر Butterworth از نوع n با فرکانس کاهش به فاصله D_0 از مبداء تابع انتقال زیر را دارد:

$$H(u,v)=\frac{1}{1+[D(u,v)/D_0]^{2n}}$$

تابع انتقال blpf بر خلاف ilpf در D0 ناپیوستگی ندارد. در فیلترهایی که تابع انتقال هموار دارند می‌توان فرکانس کاهش نقاط هندسی را برای نقاطی که H(u,v) آنها کسری از حداکثر مقدار آنها است تعریف کرد. در معادله قبل H(u,v)=0.5 (که ۵۰٪ از حداکثر مقدار ۱ کمتر است) تابع انتقال فیلتر پایین‌گذر گوسی در رابطه زیر ارائه شده است.

$$H(u,v)=e^{-D^2(u,v)/2\sigma^2}$$

در اینجا δ انحراف معیار است. اگر فرض کنیم $D_0=0$ عبارتهای زیر را در پارامتر کاهش (d)صفر داریم:

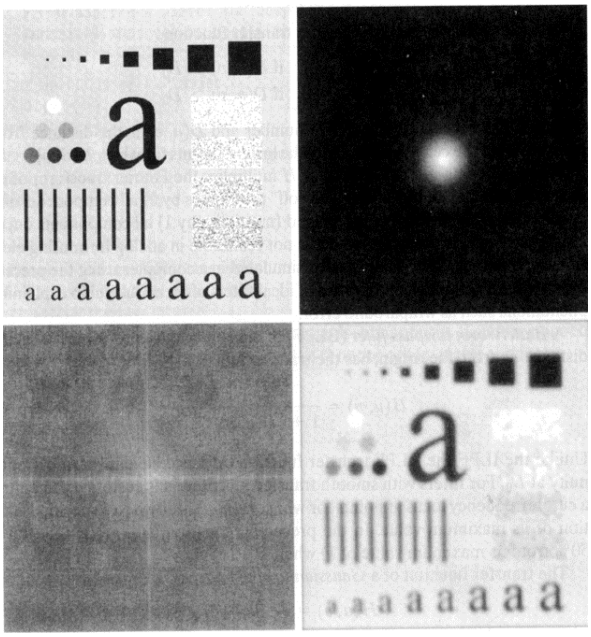
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

در اینجا وقتی $D_0=D(u,v)$ باشد مقدار فیلتر از حداکثر مقدار خود که ۱ است به ۰.۰۶۰۷ تنزل می یابد.

مثال ۴.۴:

یک تابع پایین‌گذر گوسی را به تصویر ۵۰۰*۵۰۰ پیکسل اجرا می‌کنیم مقدار D0 معادل ۵٪ پهنای تصویر لایه‌بندی شده است. با توجه به مراحل فیلترگذاری بخش ۴.۳.۲. روابط زیر برقرار است:

```
>> PQ = paddedsize(size(f));
>> [U, V] = dftuv(PQ(1), PQ(2));
>> D0 = 0.05*PQ(2);
>> F = fft2(f, PQ(1), PQ(2));
>> H = exp(-(U.^2 + V.^2)/(2*(D0^2)));
>> g = dftfilt(f, H);
```



a b
c d

تصویر ۴.۱۳. فیلترسازی پایین‌گذر ((a)) تصویر اصلی ((b) فیلتر پایین‌گذر گوسی که به صورت تصویر نشان داده شده است (سی) طیف تصویرهای پردازش شده

با تایپی حروف زیر تصویر را به صورت شکل ۴.۱۳.(b)می‌بینیم.

```
>> figure, imshow(fftshift(H), [ ])
```

با تایپ حروف زیر این طیف به صورت تصویر ۴.۱۳.(c)دیده می‌شود:

```
>> figure, imshow(log(1 + abs(fftshift(F))), [ ])
```

تصویر خروجی در ۴.۱۳.(d)دیده می‌شود که با فرمان زیر نمایش داده شده است.

```
>> figure, imshow(g, [ ])
```

همان طور که انتظار می‌رود این تصویر نوع تار شده تصویر اصلی است.

تابع زیر تابع انتقال همه فیلترهای پایین‌گذر این بخش را ایجاد می‌کند.

```
function H = lpfilter(type, M, N, D0, n)
%LPFILTER Computes frequency domain lowpass filters.
%     H = LPFILTER(TYPE, M, N, D0, n) creates the transfer function of
%     a lowpass filter, H, of the specified TYPE and size (M-by-N). To
%     view the filter as an image or mesh plot, it should be centered
%     using H = fftshift(H).
%
%     Valid values for TYPE, D0, and n are:
%
%     'ideal'           Ideal lowpass filter with cutoff frequency D0. n need
%                 not be supplied. D0 must be positive.
%
%     'btw'       Butterworth lowpass filter of order n, and cutoff
%                 D0. The default value for n is 1.0. D0 must be
%                 positive.
%
%     'gaussian'  Gaussian lowpass filter with cutoff (standard
%                 deviation) D0. n need not be supplied. D0 must be
%                 positive.

% Use function dftuv to set up the meshgrid arrays needed for
% computing the required distances.
[U, V] = dftuv(M, N);

% Compute the distances D(U, V).
D = sqrt(U.^2 + V.^2);

% Begin filter comutations.
Switch type
case 'ideal'
      H = double(D <= D0);
case 'btw'
      if nargin == 4
         n = 1;
      end
      H = 1./(1 + (D./D0).^(2*n));
Case 'gaussian'
      H = exp(-(D.^2)./(2*(D0^2));
otherwise
      error('Unknown filter type.')
end
```

تابع ipfilter در بخش ۴.۶ مبنای ایجاد فیلترهای بالاگذر است.

۱.۵.۳. ترسیم سطحی و نمایش لبه‌ها به صورت خط(Wireframe & Surface Plotting)

ترسیم تابع یک متغیر در بخش ۳.۳.۱ معرفی شد. در این بخش نمایش ۳ بعدی لبه‌ها را به صورت خط معرفی می‌کنیم. همچنین ترسیم سطحی که برای تجسم تابعهای انتقال فیلترهای ۲ بعدی مفید هستند. راحت‌ترین روش برای ترسیم چهارچوب خطی تابع ۲ بعدی H استفاده از تابع مش است که ترکیب زیر را دارد:

```
mesh(H)
```

این تابع یک چهارچوب خطی برای x=1:m, y=1:n ایجاد می‌کند. این چهارچوبهای خطی بیش از حد ضخیم هستند. اگر M و n بزرگ باشند هر کدام از کی‌امین نقطه‌ها را با ترکیب زیر ترسیم می‌کنیم.

```
mesh(H(1:k:end, 1:k:end))
```

معمولاً ۴۰ الی ۶۰ تقسیم بندی فرعی در امتداد هر محور بین میزان تفکیک پذیری و ظاهر تعادل برقرار می‌کند.

در نرم افزار MATLAB شبکه‌ها به طور پیش فرض رنگی ترسیم می‌شوند. فرمان زیر چهارچوب خطی را مشکی می‌کند.

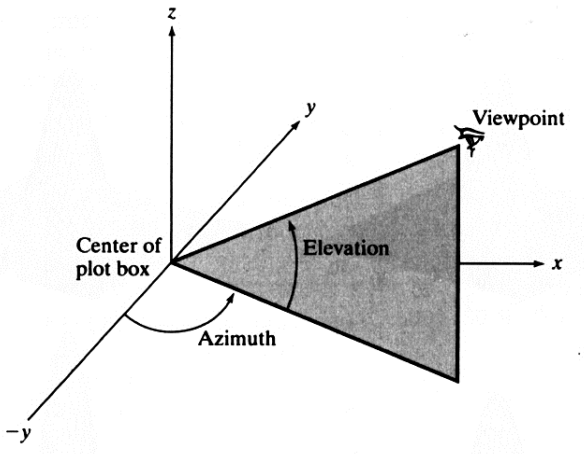
```
colormap([0 0 0])
```

نرم افزار MATLAB یک شبکه و محور آن را روی تور ترسیم شده قرار می‌دهد. این موارد را می‌توان با فرمانهای زیر غیرفعال کرد:

```
grid off
axis off
```

با جایگزین کردن کلمات خاموش با روشن در این عبارتها می‌توان دوباره آنها را روشن کرد. موقعیت بیننده با تابع view کنترل می‌شود که ترکیب زیر را دارد:

```
view(az, el)
```



Az, e1 در تصویر ۴.۱۴ به ترتیب نمایانگر زاویه azimuth و ارتفاع بر حسب درجه هستند. فلشها مقادیر مثبت را نشان می‌دهند. مقادیر پیش فرض az=-37, el=30 است که بیننده را در ربع تعریف شده توسط x, -y– قرار می‌دهد. می‌بینیم که این ربع با محورهای x,y مثبت در تصویر ۴.۱۴ تعریف شده است.

برای تعیین خصوصیات هندسی نما حروف زیر را تایپ می‌کنیم:

```
>> [az, el] = view;
```

برای قرار دادن مقادیر روی حالت پیش فرض حروف زیر را تایپ می‌کنیم:

```
>> view(3)
```

برای تغییر دادن نقطه دید می‌توان روی دکمه چرخش ۳ بعدی در میله ابزار کادر تصویر کلیک کرد و سپس کادر تصویر را جابجا کرد.

همان طور که در فصل ۶ گفتیم می‌توان موقعیت بیننده را طبق مختصات دکارتی (x, y,z) مشخص کرد که هنگام کار با داده‌های rgb مطلوب است. ولی معمولاً این روش برای نمایش ترسیم فقط ۲ پارامتر دارد و خودانگیخته است.

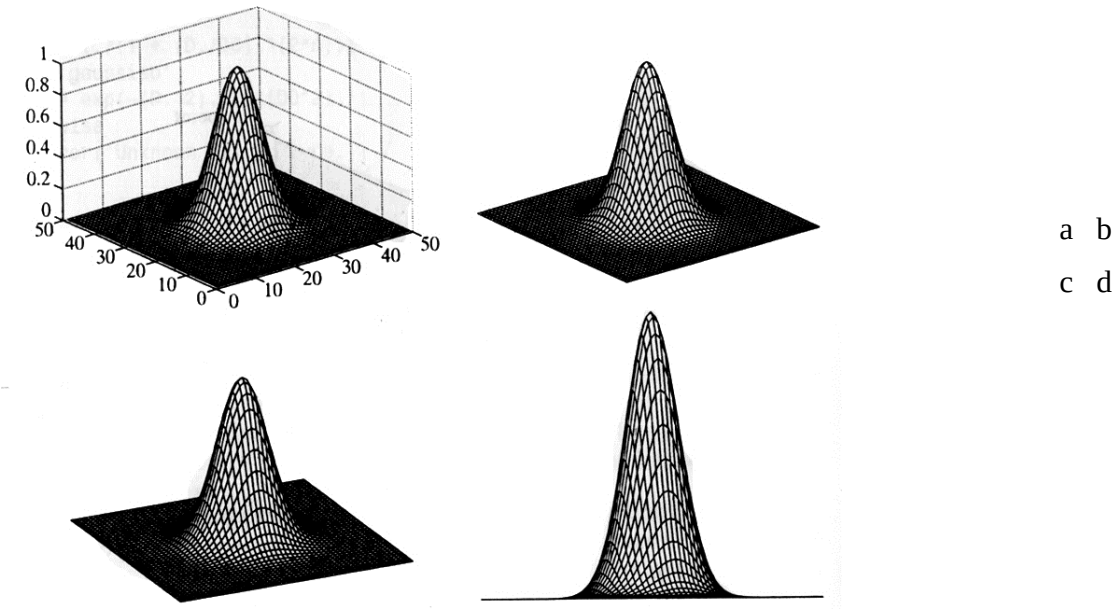
مثال۴.۵ :

یک فیلتر پایین‌گذر گوسی شبیه به فیلتر استفاده شده در مثال ۴.۴ در نظر بگیرید:

```
>> H = fftshift(lpfilter('gaussian', 500, 500, 50));
```

چهارچوب خطی ایجاد شده توسط فرمانهای زیر در تصویر ۴.۱۵ نشان داده شده است

```
>> mesh(H(1:10:500, 1:10:500))
>> axis([0 50 0 50 0 1])
```



تصویر ۴.۱۵:((a)) ترسیمی با استفاده از تابع شبکه ((b) محور و شبکه حذف شده است (c) با تابع view چشم انداز متفاوتی به دست آمده است. (d) نمای دیگری که با همان تابع به دست آمده است.

.همان طور که قبلاً در این بخش گفتیم چهارچوب خطی به طور پیش فرض رنگی است پایه آن آبی و راس آن قرمز است. خطوط ترسیم را مشکی می‌کنیم و محور و شبکه را با تایپ حروف زیر حذف می‌کنیم:

```
>> colormap([0 0 0])
>> axis off
>> grid off
```

نتیجه در تصویر ۴.۱۵.(b)دیده می‌شود. نتیجه فرمان زیر در تصویر ۴.۱۵.(c)دیده می‌شود.

```
>> view(-25, 30)
```

در اینجا ناظر اندکی به راست حرکت کرده است ولی ارتفاع ثابت است. نتیجه ترک azimuth در ۲۵- تنظیم ارتفاع روی صفر در تصویر ۴.۱۵.(d)دیده می‌شود.

```
>> view(-25, 0)
```

در این مثال قدرت ترسیم تابع ساده شبکه دیده می‌شود.

گاهی وقتها بهتر است به جای استفاده از چهارچوب خطدار تابع روی سطح ترسیم شود. تابع surf این کار را انجام می‌دهد. ترکیب آن به شرح زیر است:

```
suft(H)
```

این تابع ترسیمی همانند mesh ایجاد می‌کند. ولی چهارضلعی‌های آن رنگی هستند. (این حالت را سایه‌گذاری وجهی می‌نامیم. از فرمان زیر برای تبدیل رنگها به خاکستری استفاده می‌کنیم:

```
colormap(gray)
```

تابعهای axis, grid, view همان طور که قبلاً برای تابع mesh شرح دادیم کار می‌کنند. مثلاً تصویر ۴.۱۶.(a) نتیجه اجرای فرمانهای زیر است:

```
>> H = fftshift(lpfilter('gaussian', 500, 500, 50));
>> suft(H(1:10:500, 1:10:500))
>> axis([0 50 0 50 0 1])
>> colormap([gray])
>> grid off ; axis off
```

سایه‌گذاری وجهکی هموار می‌شود و خطوط شبکه با استفاده از فرمان زیر درون‌یابی می‌شوند.

```
shading interp
```

با تایپ این فرمان در خط فرمان تصویر ۴.۱۶.(b)ایجاد می‌شود.

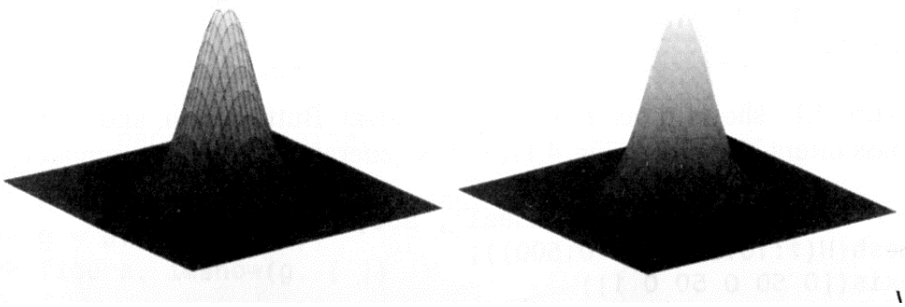
وقتی هدف ترسیم تابع تحلیلی ۲ متغیر باشد، برای ایجاد مقادیر مختصات از meshgrid استفاده می‌کنیم و از این مقادیر نمونه ماتریس متمایز را در تابعهای mesh, surf استفاده می‌کنیم. مثلاً برای ترسیم تابع زیر

$$f(x,y) = xe^{(-x^2-y^2)}$$

از منفی ۱ به ۲ به صورت واحدهای افزایش ۰.۱ برای x,y عبارت زیر را می‌نویسیم:

```
>> [Y, X] = meshgrid(-2:0.1:2, -2:0.1:2);
>> Z = X.*exp(-X.^2 – Y.^2);
```

سپس meshz, surfz را همچون قبل استفاده می‌کنیم. از مبحث ۲.۱۰.۴ به خاطر داریم که در تابع meshgrid ستونهای y اول و ردیفهای x دوم فهرست بندی شده‌اند.



a b

تصویر ۴.۱۶: ((a)) ترسیم به دست آمده با استفاده از تابع surf ((b)) نتیجه استفاده از فرمان shading interp

۴.۶. واضح کردن فیلترهای دامنه فرکانس(Sharpening frequency Domain Filter)

همان طور که فیلتر پایین گذر(lowpass) تصویر را تار می‌کند، فیلتر بالاگذر تصویر را واضح می‌کند و فرکانسهای پایین را تعدیل می‌کند و فرکانسهای بالای تبدیل فوریر را نسبتاً ثابت نگه می‌دارد. در این بخش به بررسی چند روش برای فیلترسازی بالاگذر می‌پردازیم.

۴.۶.۱. مبنای فیلترسازی بالاگذر(Basic Heighpass Filter)

با توجه به تابع انتقال hp(u,v) در این فیلتر پایین گذر(lowpass) با استفاده از این رابطه ساده تابع انتقال فیلتر بالاگذر مطابق با آن را به دست می‌آوریم.

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

تابع ipfilter که در بخش قبل آن را ابداع کردیم مبنای فیلتر بالاگذر به شرح زیر است:

```
function H = hpfilter(type, M, N, D0, n)
%HPFILTER Computes frequency domain highpass filters.
%     H = HPFILTER(TYPE, M, N, D0, n) creates the transfer function of
%     a highpass filter, H, of the specified TYPE and size (M-by-N).
%     Valid values for TYPE, D0, and n are:
%
```

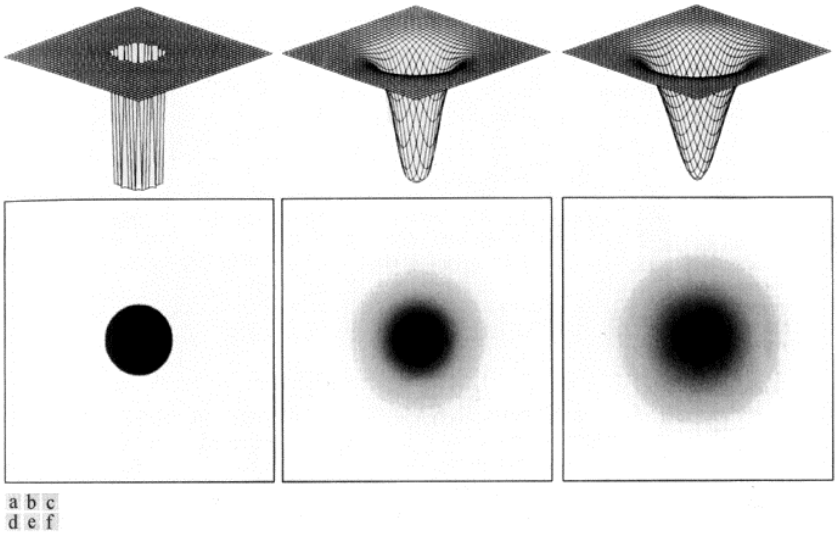
```
%     'ideal'           Ideal highpass filter with cutoff frequency D0. n
%                 need not be supplied. D0 must be positive.
%
%     'btw'       Butterworth highpass filter of order n, and cutoff
%                 D0. The default value for n is 1.0. D0 must be
%                 positive.
%
%     'gaussian'  Gaussian highpass filter with cutoff (standard
%                 deviation) D0. n need not be supplied. D0 must be
%                 positive.

% The transfer function Hhp of a highpass  filter is 1-Hlp,
% where Hlp is the transfer function of the corresponding lowpass
% filter. Thus, we can use function lpfilter to generate highpass
% filters.

if nargin == 4
   n = 1; % Default value of n.
end
% Generate highpass filter.
Hlp = lpfilter(type, M, N, D0, n);
H = 1 - Hlp;
```

ترسیمها و تصاویر مطلوب فیلترهای بالاگذر گوسی و Butterworth در تصویر ۴.۱۷ دیده می‌شوند. این ترسیم با استفاده از فرمانهای زیر ایجاد شد.

```
>> H = fftshift(hpfilter('ideal', 500, 500, 50));
>> mesh(H(1:10:500, 1:10:500));
>> axis([0 50 0 50 0 1])
```



a b c
d e f

تصویر ۴.۱۷: ردیف بالا: چشم اندازی از فیلترهای بالاگذر مطلوب گوسی و Butterworth ردیف پایین: تصاویر مطابق با آنها

```
>> colormap([0 0 0])
>> axis off
>> grid off
```

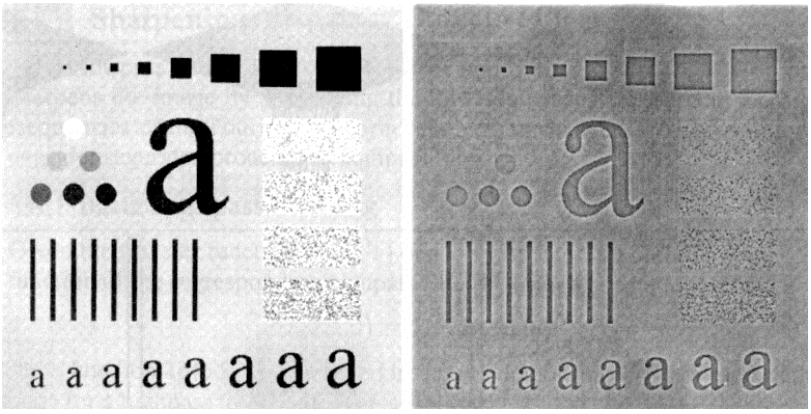
تصویر مطابق با عکس ۴.۱۷.(d)با استفاده از فرمان زیر ایجاد شد:

```
>> figure, imshow(H, [ ])
```

در اینجا مرز مشکی نازک روی تصویر قرار داده شده است تا مرز آن را مشخص کند. فرمانهای مشابه باعث ایجاد تصویر ۴.۱۷ شد. (فیلتر Butterworth در ترتیب ۲ است.

همین الگو در تصویر ۴.۱۸.(a) دیده می‌شود. که در تصویر ۴.۱۳.(a) و ۴.۱۸.(b)با استفاده از فرمانهای زیر به دست آمده است و نتیجه اجرای فیلتر بالاگذر گوسی است.

```
>> PQ = paddedsize(size(f));
>> D0 = 0.05*PQ(1);
>> H = hpfilter('gaussian', PQ(1), PQ(2), D0);
>> g = dftfilt(f, H)
>> figure, imshow(g, [ ])
```



a b

تصویر ۴.۱۸. ((a)) تصویر اصلی ((b) نتیجه اجرای فیلتر بالاگذر گوسی

همان طور که در تصویر ۴.۱۸.(b)می‌بینید لبه و سایر گذرهای تشدید شده تصویر تقویت و بهبود داده شده‌اند. ولی از آنجائی که میانگین تصویر با f(0,0) مشخص می‌شود و فیلترهای بالاگذر در مبداء تبدیل فوریر صفر هستند این تصویر بیشتر رنگ پس زمینه موجود در حالت اصلی خود را از دست داده است. این مسئله در بخش بعد بررسی شده است.

۴.۶.۲. فیلترسازی تشدید کننده فرکانس بالا(Heigh-frequency Emphasis Filtering)

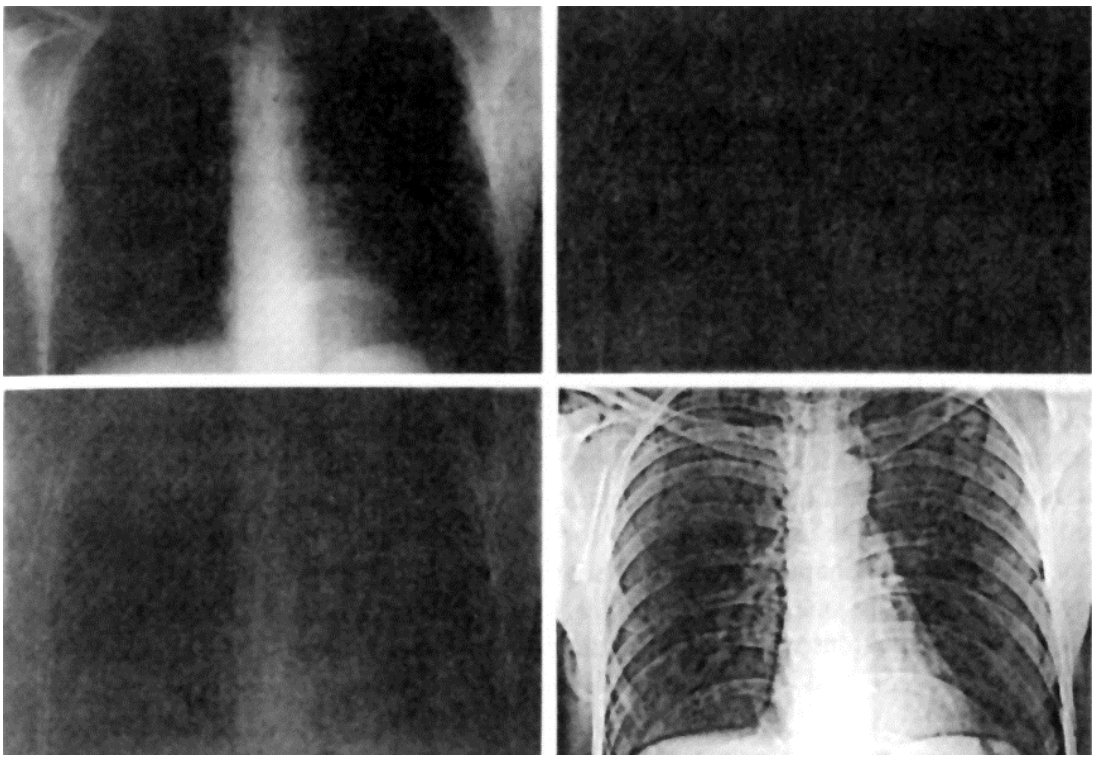
همان طور که در مثال ۴.۷ گفتیم فیلترهای بالاگذر عبارت dc را صفر می‌کنند بنا بر این میانگین تصویر به صفر می‌رسد. یک روش جبران این موضوع اضافه کردن مقدار جبران کننده به فیلتر بالاگذر است. وقتی این مقدار با ضرب کردن فیلتر در یک مقدار ثابت بیشتر از ۱ شود این روش را فیلترسازی تشدید شده فرکانس بالا می‌نامیم چون که مضرب ثابت فرکانسهای بالا را تشدید می‌کند. این مضرب دامنه فرکانسهای پایین را زیاد می‌کند. ولی مادامی که مقدار جبران کننده نسبت به مضرب کم باشد، تاثیر فرکانسهای پایین تقویت شده کمتر از فرکانسهای بالا است. تشدید فرکانس بالا تابع انتقال زیر را دارد:

$$H_{hfe}(u,v) = a + bH_{hp}(u,v)$$

در اینجا a مقدار جبران کننده است. B مضروب است. H(u,v) تابع انتقال فیلتر بالاگذر است.

مثال۴.۸ :

تصویر اشعه ایکس سینه در عکس ۴.۱۹ نشان داده شده است. تصویربردارهای اشعه ایکس را نمی‌توان مانند عدسیهای نوری متمرکز کرد. بنا بر این تصاویر حاصل از آن اندکی تار هستند. هدف از این مثال واضح کردن تصویر ۴.۱۹.(a) است چون که سطوح خاکستری این تصویر خاص متمایل به انتهای تیره مقیاس خاکستری هستند. با این مثال شرح می‌دهیم که چگونه می‌توان دامنه پردازش فضا را برای تکمیل فیلترسازی دامنه فرکانس به کار برد.



a b
c d

تصویر ۴.۱۶. فیلترسازی تشدید فرکانس بالا: ((a)) تصویر اصلی ((b) نتیجه فیلترسازی بالاگذر (c) نتیجه تشدید فرکانس بالا (d) تصویر (e)بعد از برابرسازی پیشینه نما

نتیجه فیلترسازی تصویر ۴.۱۹.(a) در ۴.۱۹.(b)نشان داده شده است و فیلتر بالاگذر Butterworth از ترتیب دوم دارد و مقدار D_0 معادل ۵٪ بعد عمودی تصویر لایه‌بندی شده است. مادامی که شعاع فیلتر آنقدر کوچک نباشد که فرکانسهای نزدیک مبداء تبدیل عبور کنند، فیلترسازی بالاگذر بیش از حد به مقدار D_0 حساس نیست. همان طور که انتظار می‌رفت نتیجه فیلتر شده خاصیتی ندارد. ولی لبه‌های اصلی در تصویر دیده می‌شوند. امتیازهای فیلترسازی تشدید بالا در تصویر ۴.۱۹.(c)نشان داده شده است مقیاس خاکستری با توجه به فرکانس پایین مولفه‌ها حفظ شد. فرمانهای زیر برای ایجاد تصاویر پردازش شده در تصویر ۴.۱۹ به کار برده شد. f نمایانگر تصویر ورودی است.

```
>> PQ = paddedsize(size(f));
>> D0 = 0.05*PQ(1);
>> HBW = hpfilter('btw', PQ(1), PQ(2), D0, 2);
>> H = 0.5 + 2*HBW;
>> gbw = dftfilt(f, HBW);
>> gbw = gscale(gbw);
>> ghf = dftfilt(f, H);
>> ghf = gscale(ghf);
>> ghe = histep(ghf, 256);
```

همان طور که در بخش ۳.۳.۲ گفتیم تصویرهای مقیاس خاکستری نمونه‌های خوبی برای برابرسازی پیشینه نما هستند. همان طور که در تصویر ۴.۱۹.(d)دیده می‌شود این روشی مناسب برای تقویت و بهبود تصویر در این مثال بود.

به وضوح ساختار استخوان و سایر جزئیات که در سایر تصاویر قابل رویت نیستند توجه کنید. تصویر نهایی تقویت شده کمی پارازیت دارد، ولی وقتی مقیاس خاکستری تصاویر اشعه ایکس را گسترش می‌دهیم چنین چیزی روی می‌دهد. نتایج به دست آمده با ادغام تشدید فرکانس بالا و برابرسازی پیشینه نما نسبت به نتیجه استفاده انفرادی از هر یک از این دو روش ارجحیت دارد.

خلاصه مطالب(summary)

به غیر از روشهای تقویت تصویر که در این فصل و فصل قبل شرح دادیم، این فرضیه‌ها و شگردها مبنای سایر روشهای پردازش تصویر است که در مابقی بحثهای تحقیق آورده شده است. تبدیل شدت در مقیاس تبدیل زیاد به کار برده می‌شود. فیلترسازی فضایی به شکلی گسترده برای بازگرداندن تصویر به حالت اصلی در فصل بعد تشریح شده است.