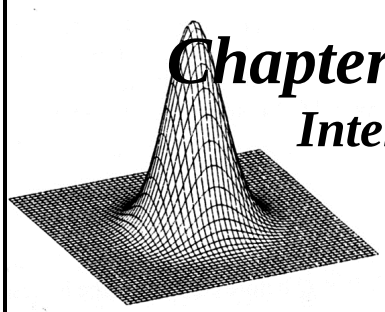


Chapter 3
Intensity Transformation & Spatial filtering

تبدیل نیرو و غربال مواضع فضایی

پیش نمایش(Preview)

عبارت دامنه فضایی(Spatial domain) اشاره به سطح تصویر دارد و روشهای این مقوله مبتنی بر دستکاری مستقیم عنصرهای تصویری است. در این فصل به دو مقوله مهم پردازش دامنه فضا می‌پردازیم: تبدیل شدت (یا سطح خاکستری) و غربال مواضع فضایی

در قسمتهای بعد فرمولهای MATLAB را ابداع و شرح می‌دهیم که نمایانگر شگردهای پردازش در این دو نوع مقوله است. برای این که موضوع هماهنگ باشد اکثر مثالهای این فصل مربوط به بهبود و تقویت تصویر است. این روش خوبی برای معرفی پردازش فضایی است زیرا بهبود و تقویت آن مخصوصاً برای مبتدی‌ها جذاب و خودانگیخته است. همان طور که در این تحقیق می‌بینید این شگردها کلی هستند و کاربردهای زیادی در سایر زمینه‌های پردازش تصاویر دیجیتالی دارند.

۳.۱. تاریخچه(Background)

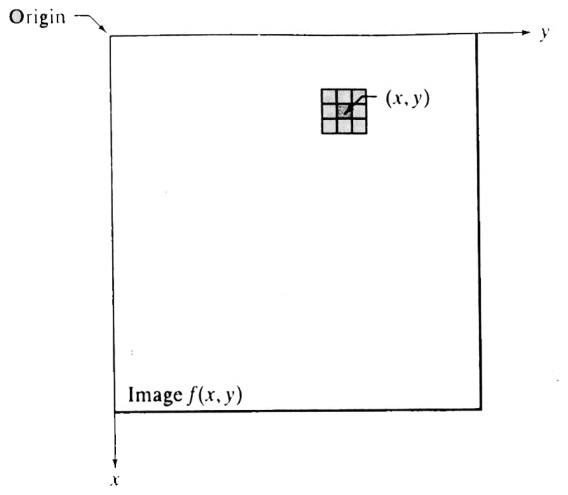
همان طور که در پاراگراف قبل گفتیم شگردهای دامنه‌دار فضایی مستقیماً روی عناصر تصویرها اثر می‌گذارند. فرایندهای دامنه فضا که در این فصل بحث شده‌اند مربوط به رابطه زیر هستند:

$$g(x,y) = T[F(X,y)]$$

در این رابطه $f(x,y)$ تصویر حامل داده‌های ورودی است. $G(x,y)$ تصویر پردازش شده داده‌های خروجی است و T عملگری روی f است که در فضای همجوار حول نقطه (x, y) تعریف می‌شود.

T می‌تواند روی یک مجموعه تصاویر همچون افزایش k عدد تصویر در کاهش اختلال به کار برده شود.

روش اصلی برای تعریف فضای همجوار حول نقطه (x, y) استفاده از شکل مربع یا مستطیلی در کانون (x, y) است که در تصویر ۳.۱ نشان داده شده است. کانون این موضوع در عناصر تصویری جابجایی دارد و مثلاً از نقطه بالا سمت چپ شروع شده



و نقاط همجوار گوناگون را در بر می‌گیرد. عملگر T در هر موضع x,y اعمال می‌شود تا داده‌های خروجی g را در آن موضع تولید کند. برای محاسبه مقادیر g در x,y فقط از عناصر تصویری همجوار استفاده می‌شود.

مابقی این فصل مربوط به اجرای گوناگون معادله قبل است. گرچه فرضیه این معادله ساده است ولی اجرای محاسبات آن در نرم افزار MATLAB مستلزم دقت زیاد به نوع داده‌ها و دامنه مقادیر آنها است.

۳.۲. تابع‌های تبدیل نیرو(Intensity Transformation Function)

ساده‌ترین شکل تبدیل T آن است که نقاط همجوار تصویر ۳.۱ به اندازه ۱ * ۱ (یک عنصر تصویری) باشند. در این صورت، مقدار g در x, y بستگی به شدت f در آن نقطه دارد و t تابع تبدیل سطح خاکستری یا شدت می‌شود. هنگام کار با تصویرهای خاکستری تک رنگ این دو واژه به صورت مترادف به کار برده شده است. واژه شدت هنگام کار با تصاویر رنگی به معنی یکی از اجزای رنگی در برخی مواضع فضایی است که در فصل ۶ تشریح شده است.

از آنجائی که آنها فقط بستگی به مقادیر شدت دارند و در خصوص (x, y) نیز صراحتا بیان نشده‌اند، تابع‌های تبدیل نیرو به شکل ساده زیر نوشته می‌شوند:

$$S=T(r)$$

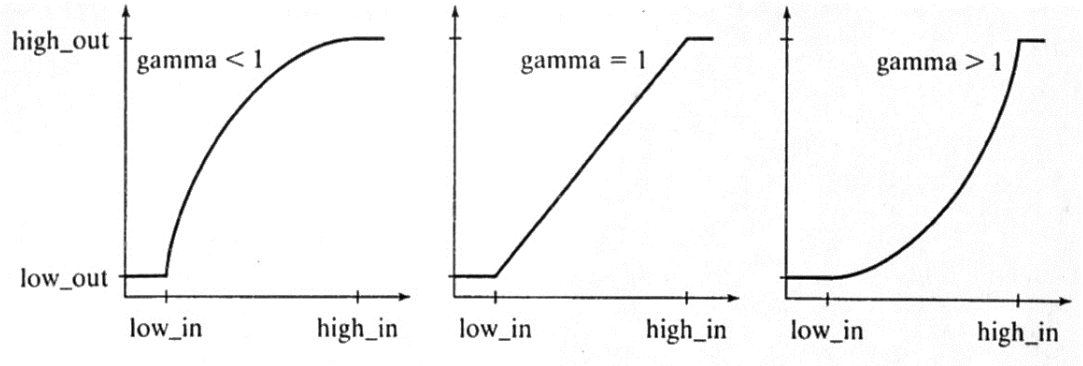
در این رابطه r شدت f است و S شدت g است و هر دو در مختصات نقطه (x, y) تصویر هستند.

۳.۲.۱. تابع imadjust (یعنی تنظیم تصویر)

این تابع ابزار IPT اصلی برای تبدیل شدت تصاویر مقیاس خاکستری است. ترکیب آن به شرح زیر است:

```
g = imadjust(f, [low_in high_in], [low_out high_out], gamma)
```

همان طور که در تصویر ۳.۲ می‌بینید، این تابع مقادیر شدت رنگ را در تصویر f به مقادیر جدید در g تبدیل می‌کند طوری که مقادیر ورودی بین حداقل و حداکثر به مقادیر خروجی بین حداقل و حداکثر تبدیل می‌شوند.



a b c

مقادیر کمتر از مقدار حداقل ورودی و حداکثر ورودی حذف می‌شوند یعنی مقادیر کمتر از حداقل داخلی روی حداقل خروجی ترسیم می‌شوند و مقادیر بیشتر از حداکثر ورودی روی حداکثر خروجی ترسیم می‌شوند. تصویر داده‌های ورودی می‌تواند از نوع uint ۸، uint ۱۶ یا مضاعف باشد. و تصویر خروجی از همان نوع داده‌های ورودی است. کلیه داده‌های ورودی تابع imadjust به غیر از f صرف نظر از کلاس f مقادیر بین ۰ و ۱ هستند. اگر f از کلاس uint ۸ باشد، تابع imadjust مقادیر مقیاس ۲۵۵ را چند برابر می‌کند تا مقادیر واقعی کاربردی مشخص شود. اگر f از کلاس uint ۱۶ باشد، مقادیر ضرب در ۶۵۵۳۵ می‌شود. استفاده از ماتریس خالی در این فرمولها باعث پدید آمدن مقادیر پیش فرض می‌شود که کمتر از حداقل خروجی هستند. شدت داده‌های خروجی معکوس می‌شود.

پارامتر گاما شکل منحنی را نشان می‌دهد که مقادیر شدت را در f ترسیم می‌کند تا g را ایجاد کند. اگر گاما کمتر از ۱ باشد، ترسیم متمایل به مقادیر بالای داده‌های خروجی انجام می‌شود مانند تصویر ۳.۲. اگر گاما بیشتر از ۱ باشد، ترسیم متمایل به مقادیر داده‌های خروجی تیره و پایین است. اگر از شناسه تابع حذف شود مقدار پیش فرض گاما ۱ می‌شود که ترسیم خطی است.

تصویر ۳.۳ یک تصویردیجیتالی اشعه ایکس از پستان است که زخم کوچکی را روی آن نشان می‌دهد. تصویر ۳.۳. (b)نگاتیوی است که با این فرمان به دست آمده است.

```
>> g1 = imadjust(f, [0 1], [1 0]);
```

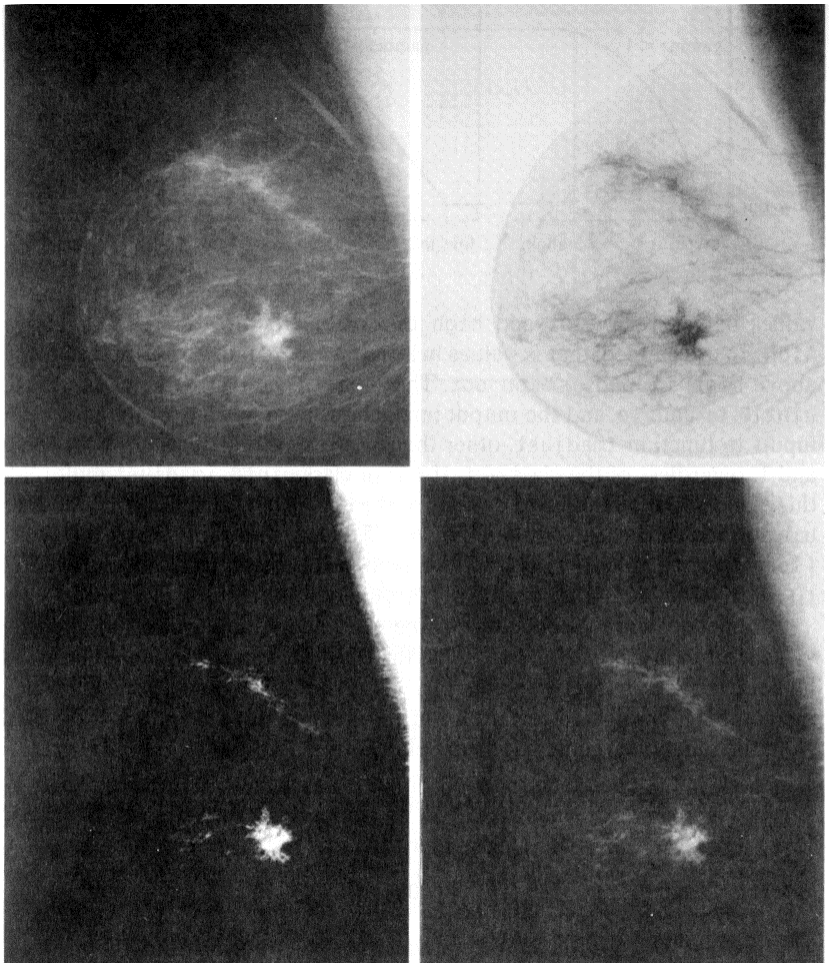
این فرایند که معادل دیجیتالی به دست آوردن نگاتیو است برای بهبود و تقویت جزئیات خاکستری و سفید تعبیه شده در یک محیط تیره بزرگ مفید است. ببینید تحلیل بافت پستان در تصویر ۳.۳ ب چقدر آسانتر است. نگاتیو این تصویر را می‌توان با اجرای تابع IPT نیز به دست آورد.

```
g = imcomplement(f)
```

تصویر ۳.۳ (c). نتیجه استفاده از این فرمان را نشان می‌دهد.

```
>> g2 = imadjust(f, [0.5 0.75], [0 1]);
```
که ناحیه دارای مقیاس خاکستری بین ۰.۵ تا ۰.۷۵ را تا حیطه کامل گسترش می‌دهد. این نوع پردازش برای مشخص کردن ناحیه مورد نظر مفید است.

```
>> g3 = imadjust(f, [ ], [ ], 2);
```



a b
c d

استفاده از این فرمان باعث تولید نتیجه‌ای شبیه به تصویر ۳.۳ (c)می‌شود (که رنگهای خاکستری آن بیشتر است) و انتهای پایین فشرده و انتهای بالای مقیاس خاکستری گسترده می‌شود.

۳.۲.۲ تبدیل تضاد رنگها و لگاریتم(Logrithmic & Contrast-streching Transformation)

ابزار تبدیل وضوح و تضاد رنگها ابزاری اساسی برای دستکاری میزان پویایی تصویر هستند. تبدیل لگاریتمی با عبارت زیر انجام می‌شود.

```
g = c*log(1 + double(f ))
```

در این رابطه c ثابت است. شکل این تبدیل شبیه به نمودار گاما در تصویر ۳.۲ (a) است که در هر دو مقیاس مقادیر حداقل ۰ و مقادیر حداکثر ۱ هستند. ولی ببینید شکل منحنی گاما تغییرپذیر است در صورتی که شکل تابع ثبت گردش کار ثابت است.

یکی از کاربردهای اصلی تبدیل لگاریتم فشرده‌سازی دامنه پویا است. مثلاً استفاده از طیف فوریه (فصل ۴) با مقادیر دامنه ۰ تا ۱۰۶ یا بیشتر غیرعادی نیست. وقتی روی مانیتوری که مقیاس خطی آن ۸ بیت است نمایش داده می‌شودمادیر بالا بر صفحه نمایش مسلط هستند که باعث محو شدن مقادیر کم شدت در این طیف می‌شوند. با محاسبه لگاریتم دامنه پویای مثلاً ۱۰ به توان ۶ به ۱۴ کاهش می‌یابد که مدیریت آن آسان‌تر است.

هنگام انجام یک تبدیل لگاریتمی بهتر است مقادیر فشرده را به حیطه کامل نمایش بازگردانیم. راحتترین روش برای انجام این کار در نرم افزار MATLAB در صفحه‌های ۸ بیت استفاده از فرمول زیر است:

```
gs = im2uint8(mat2gray(g));
```

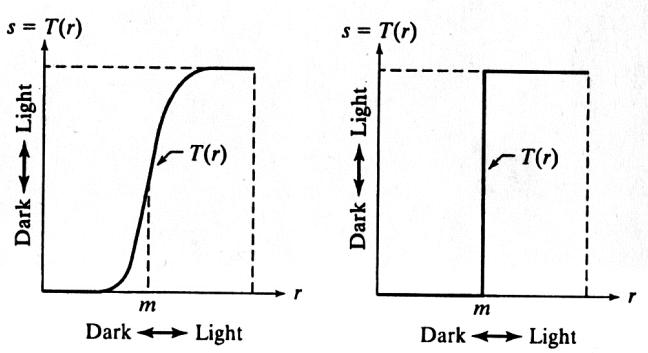
با استفاده از تابع mat2gray می‌توان مقادیر را به حیطه [0,1] و با استفاده از im2wint8 می‌توان آنها به حیطه [0,255] بازگرداند. در بخش ۳.۲.۳ یک تابع مقیاس بندی معرفی می‌کنیم که نوع داده‌های ورودی را به صورت خودکار تشخیص داده و تبدیل مناسب را اجرا می‌کند.

تابع تصویر ۳.۴ (a) یک تابع تبدیل و کشش وضوح تصویر(contrast-stretching transformation function) است زیرا داده‌های ورودی کمتر از m را به سطوح تیره در تصویر خروجی تبدیل می‌کند. مقادیر بیشتر از m را به یک سری سطوح روشن در داده‌های خروجی تبدیل می‌کند. در نتیجه وضوح تصویر بالا می‌رود. در واقع در مثال تصویر ۳.۴ (b)می‌بینید که داده‌های خروجی یک تصویر دودویی هستند. این تابع محدود کننده تابع آستانه‌یابی نامیده می‌شود که همان طور که در فصل 7 می‌بینیم ابزاری ساده برای ترسیم تصویر است. با استفاده از نشانه‌گذاریهای ابتدای این بخش تابع شکل ۳.۴. (a) به صورت زیر تغییر می‌کند:

$$s = T(r) = \frac{1}{1+(m/r)^E}$$

در اینجا r نمایانگر شدت تصویر ورودی است s مقادیر مطابق با آن در تصویر خروجی است. E شیب تابع را کنترل می‌کند. این معادله در نرم افزار MATLAB برای تشریح یک تصویر کامل به شرح زیر است:

```
g = 1./(1 + (m./(double(f) + eps)).^E)
```



به کاربرد eps جدول ۲.۱۰ برای جلوگیری از سرریز در صورتی که f مقدار ۰ داشته باشد توجه کنید. از آنجائی که مقدار محدود کننده T(r) ۱ است هنگام کار با این نوع تبدیل، مقادیر خروجی به مقیاس ۰و ۱ مقیاس بندی می‌شوند. شکل تصویر ۳.۴. (a) با E=20 به دست آمده است.

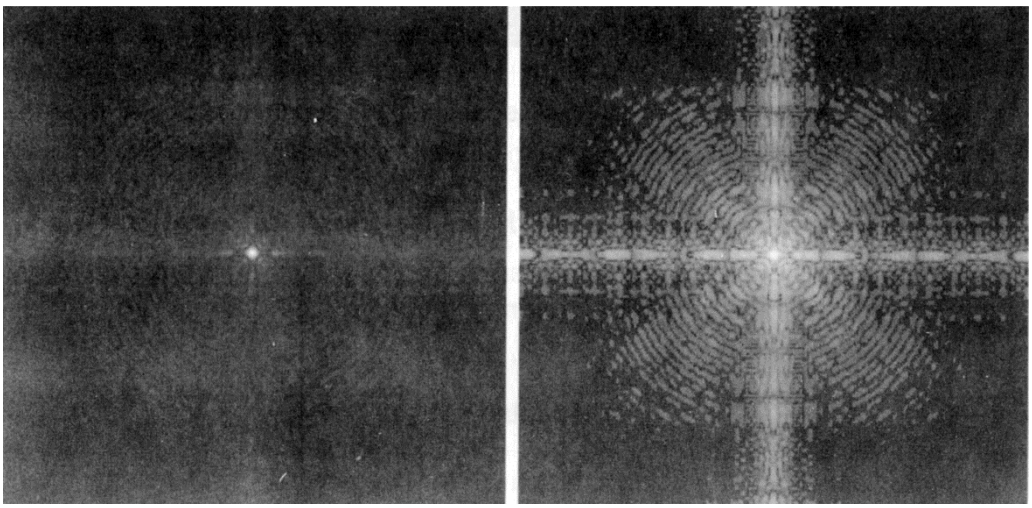
تصویر ۳.۵. (a) یک طیف فوریه است که مقادیر آن در دامنه ۱.۵ * $۱۰^{۶}$ هستند که در سیستم ۸ بیتی با مقیاس خطی نمایش داده شده است. در تصویر ۳.۵ (b)نتایج به دست آمده از اجرای فرمان را می‌بینید.

```
g = im2uint8(mat2gray(log(1 + doubie(f))));
>> imshow(g)
```

بهبود g در تصویر اصلی کاملاً مشهود است.

۳.۲.۳. تابع‌های M مفید برای تبدیل شدت(Some Utility M-Functions for Intensity Transformation)

در این بخش ۲ تابع M ابداع می‌کنیم که جنبه‌های مختلف تبدیل شدت را که در ۲ بخش قبل گفتیم در هم ادغام می‌کنند. برای بررسی خطاها جزئیات یکی از آنها را نشان دادیم تا مشخص شود تابع‌های MATLAB به چه شکل‌هایی فرمول بندی می‌شوند طوری که بتوانند از عهده انواع داده‌های ورودی و خروجی برآیند، و نمونه کدهای استفاده شده در تحقیق را نشان دهند. از این نقطه به بعد شرح کامل کدهای جدید M فقط زمانی در مباحث ما درج می‌شود که هدف تشریح سازه‌های برنامه‌های خاص باشد. برای تشریح کاربرد یک تابع جدید MATLAB یا IPT و یا برای مرور فرضیه‌های قبلی است. در غیر این صورت فقط ترکیب تابع را می‌توان تشریح کرد و کد آن در ضمیمه (c)درج شده است. برای تمرکز روی ساختار اساسی تابع‌های تشریح شده در مابقی تحقیق این آخرین بخشی است که کاربرد گسترده بررسی خطاها را شرح می‌دهیم. این روشها نشان می‌دهد خطاها چگونه در برنامه MATLAB بررسی می‌شوند.



a b

بررسی داده‌های ورودی و خروجی متغیر(Handling a Variable Number of Inputs and/or Outputs)

برای بررسی شناسه‌های وارد شده در تابع M از حاشیه تابع استفاده می‌کنیم.

```
n = nargin
```

که تعداد واقعی شناسه‌های قرار داده شده در تابع M را تولید می‌کند. Nargout تابع همراه با داده‌های خروجی تابع M استفاده می‌شود. ترکیب آن به شرح زیر است:

```
n = nargout
```

فرض کنید تابع M زیر را در خط فرمان اجرا کنیم:

```
>> T = testhv(4, 5);
```

استفاده از nargin در این تابع باعث تولید ۲ می‌شود در حالی که استفاده از nargout باعث تولید ۱ می‌شود.

تابع nargchk در تابع M برای بررسی وارد شدن تعداد صحیح شناسه بررسی می‌شود. ترکیب آن به شرح زیر است:

```
msg = nargchk(low, high, number)
```

اگر عدد کمتر از حداقل باشد این تابع پیام "نبود پارامترهای کافی" را صادر می‌کند و اگر عدد بیشتر از حداکثر باشد، پیام "وجود تعداد بیش از حد پارامتر" را صادر می‌کند. اگر عدد بین حداقل و حداکثر باشد nargchk ماتریس خالی تولید می‌کند. یکی از کاربردهای تابع nargchk توقف اجرای برنامه از طریق تابع error است مشروط بر آن که تعداد صحیح شناسه‌ها وارد شده باشد. تعداد واقعی شناسه‌های ورودی با تابع nargin مشخص می‌شود. مثلاً کد زیر را در نظر بگیرید.

```
function G = testhv2(x, y, z)
.
.
.
error(nargchk(2, 3, nargin));
.
.
.
Typing
» testhv2(6);
```

که فقط یک شناسه ورودی دارد و خطای زیر را تولید می‌کند.

```
Not enough input argument.
```

و بعد اجرا خاتمه می‌یابد.

اغلب نوشتن تابعهایی که تعداد شناسه‌های داده‌های ورودی و خروجی آن متغیر است مفید است. برای این کار متغیرهای varargin, varargout را استفاده می‌کنیم. در این عبارتها متغیرهای فوق باید با حروف کوچک نوشته شوند. مثلاً

```
function [m, n] = testhv3(varargin)
```

تعداد متغیرهای تابع testhv3 را قبول می‌کند و

```
function [varargout] = testhv4(m, n, p)
```

تعداد داده‌های خروجی متغیر تابع testhv4 را تولید می‌کند. اگر تابع testhv3 یک شناسه داده‌های ورودی ثابت به نام ایکس داشته باشد، که بعد از آن متغیر شناسه‌های داده‌های ورودی باشد؛ در این صورت،

```
function [m, n] = testhv3(x, varargin)
```

باعث می‌شود varargin هنگام فراخوانی تابع با دومین شناسه ورودی شروع شود که کاربر تامین می‌کند. همین توضیح در خصوص vargout نیز مصداق دارد. تابعی دارد که تعداد شناسه‌های ورودی و خروجی آن متغیر هستند. وقتی vargin به صورت شناسه ورودی یک تابع استفاده می‌شود، نرم افزار MATLAB آن را در آرایه سلول قرار می‌دهد(بخش ۲.۱۰.۵) که یک سری متغیر ورودی را از کاربر دریافت می‌کند. از آنجائی که varargin یک ارایه سلولی است، یک جنبه مهم این ترتیب آن است که فراخوانی تابع می‌تواند مجموعه‌ای داده ورودی داشته باشد. مثلاً اگر کد تابع فرضی testhv3 برای بررسی آن تجهیز شده باشد، می‌توان مجموعه مختلطی از داده‌های ورودی را به شرح زیر داشت:

```
>> [m, n] = testhv3(f, [0 0.5 1.5], A, 'label');
```

در اینجا f یک تصویر است. شناسه بعدی بردار ردیف با طول ۳ است، a یک ماتریس است و label یک رشته است. این یک خصوصیت قوی است که می‌تواند برای ساده کردن ساختار تابع‌هایی به کار رود که انواع داده‌های ورودی را نیاز دارند. همین توضیحات در خصوص varargout نیز صدق می‌کند.

یک تابع M دیگر برای تبدیل شدت(Another M-Function for Intesity Transformations)

در این بخش تابعی را ابداع می‌کنیم که تابعهای تبدیل زیر را محاسبه می‌کند. منفی، لگاریتمT گاما، و کشش وضوح تصویر. علت انتخاب این مبدلها آن است که آنها را بعداً نیاز خواهیم داشت. همچنین برای تشریح خصوصیات مکانیکی نوشتن تابع M برای تبدیل شدت به کار برده می‌شوند. در نوشتن این تابع از تابع changeclass استفاده می‌کنیم، که ترکیب زیر را دارد:

```
g = change class(newclass,f)
```

این تابع تصویر f را به کلاس مشخص شده در پارامتر newclass تبدیل می‌کند و آن را به صورت g در داده‌های خروجی قرار می‌دهد. مقادیر معتبر برای این تابع uint8,uint16 و مضاعف هستند.

توجه داشته باشید که در تابع M زیر که آن را Intrans می‌نامیم نحوه فرمت شدن عملیات تابع؛ نحوه بررسی تعداد متغیرهای تابع ورودی، نحوه جاگذاری بررسی خطاها، و نحوه انطباق تصویر خروجی با داده‌های ورودی در بخش راهنمای کد تشریح شده است. حین مطالعه کد زیر به خاطر داشته باشید که varargin ارایه سلول است لذا عناصر آن با استفاده از آکولادهای مورب انتخاب می‌شوند.

```
function g=intrans(f, varargin)
%INTRANS Performs intensity (gray-level) transformations.
%  G INTRANS(F, 'neg') computes the negative of input image F
%
%  G = INTRANS(F, 'log', C, CLASS) computes C*log(1 + F) and
%  multiplies the result by (positive) constant C. If the last two
%  parameters are omitted, C defaults to 1. Because the log is used
%  frequently to display Fourier spectra, parameter CLASS offers the
%  option to specify the class of the output as 'uint8' or
%  'uint16'. If parameter CLASS is omitted, the output is of the
%  same class as the input.%
%  G = INTRANS(F, 'gamma', GAM) performs a gamma transformation on
%  the input image using parameter GAM (a required input).
%
%  G = INTRANS(F, 'stretch', M, E) computes a contrast-stretching
%  transformation using the expression 1./(1 + (M./(F +
%  eps)).^E). Parameter M must be in the range [0, 1]. The default
%  value for M is mean2(im2double(F)), and the default value for E
%  is 4.
%  For the 'neg', 'gamma', and 'stretch' transformations, double
%  input images whose maximum value is greater than 1 are scaled
%  first using MAT2GRAY. Other images are converted to double first
%  using IM2DOUBLE. For the 'log' transformation, double images are
%  transformed without being scaled; other images are converted to
%  double first using IM2DOUBLE.
%
%  The output is of the same class as the input, except if a
%  different class is pecified for the 'log' option.
%
%  Verify the correct number of inputs.
error(nargchk(2, 4, nargin))

Store the class of the input for use later.
%  classin = class(f);
% [0, 1], and the specified transformation is not 'log', convert the
% input to the range [0, 1].
if strcmp(class(f), 'double') & max(f(:)) > 1 & . . .
     -strcmp(varargin{1}, 'log')
   f = mat2gray(f);
else % Convert to double, regardless of class(f).
   f = im2double(f);
end
% Determine the type of transformation specified.
 method = varargin{1};
% Perform the intensity transformation specified.
 switch method
case 'neg'
    g = imcomplement(f);
```

```
case 'log'
    if length(varargin) == 1
        c = 1;
    elseif length(varargin) == 2
      c = varargin{2};
  elseif length(varargin) == 3
      c = varargin{2};
     classin = varargin{3};
 else
     error(' Incorrect number of inputs for the log option.')
 end
 g = c*(log(1 + double(f)));
case 'gamma'
   if length(varargin) < 2
       error('Not enough inputs for the gamma option.')
 end
 gam = varargin{2};
 g = imadjust(f, [ ], [ ], gam);
case 'stretch'
     if length(varargin) == 1
        % Use defaults. m = mean2(f);
        E = 4.0;
     elseif length(varargin) == 3
        m = varargin{2};
        E = varargin{3};
     else error('Incorrect number of inputs for the stretch option.')
     end
   g = 1./(1 + (m./(f + eps)).^E);
otherwise
    error('Unknown enhancement method.') end
% Convert to the class of the input image.
g = changeclass(classin, g);
```

مثال ۳.۳ :

برای تشریح تابع Intrans تصویر ۳.۶. (a) در نظر گرفته شود. این نمونه خوبی برای کشش وضوح تصویر و بهبود ساختار اسکلتی است.

نتیجه تصویر ۳.۶. (b)با فراخوانی Intrans به شکل زیر حاصل شد:

```
g = intrans(f, 'stretch', mean2(im2double(f)), 0.9);
 » figure, imshow(g)
```

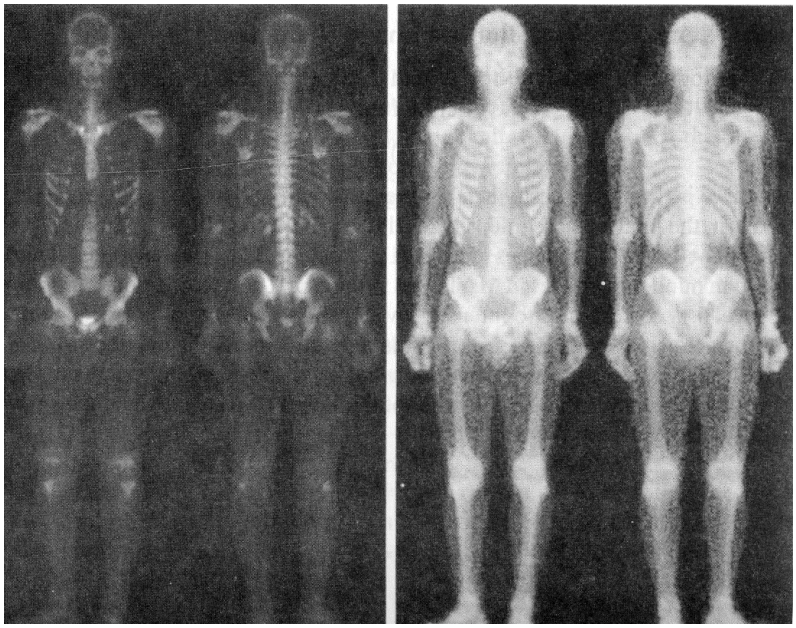
ببینید چگونه تابع mean2 برای محاسبه مستقیم میانگین f در حین فراخوانی تابع استفاده شد. نتیجه آن برای M به کار برده شد. تصویر f با استفاده از im2double به ddouble تبدیل شد تا مقادیر آن به مقیاس [0,1] در آیند و میانگین آن نیز در همین حیطه باشد که در داده‌های ورودی M نیز به آن اشاره شد. مقدار e در این کنش و واکنش مشخص شد.

تابع M برای مقیاس بندی شدت(An M-Function for Intensity Scaling)

وقتی با تصاویر کار می‌کنید معمولاً با نتایجی مواجه می‌شوید که عنصرهای تصویری آنها در یک دامنه مثبت تا منفی هستند. این موضوع در محاسبات حد فاصل مسئله ساز نیست ولی وقتی می‌خواهیم برای ذخیره کردن یا مشاهده تصویر از فرمت ۸ یا ۱۶ بیت استفاده کنیم، مسئله ساز می‌شود. در این صورت، بهتر است مقیاس تصویر را روی حداکثر)۰و ۲۵۵(یا)۰و ۶۵۵۳۵(قرار دهیم. تابع M زیر که آن را

gscale می‌نامیم، برای تحقق این امر است. این تابع می‌تواند سطوح خروجی را در دامنه‌ای خاص ترسیم کند. کد این تابع فرضیه‌های جدیدی ندارد بنا بر این آن را در اینجا درج نمی‌کنیم..

```
g=gscale(fmethod,low,high)
```



a b

در اینجا f تصویر مقیاس بندی شده است. مقادیر معتبر این روش full8 (مقدار پیش فرض) و داده‌های خروجی حیطه کامل)۰و ۶۵۵۳۵(هستند. در صورت درج آن، پارامترهای حداقل و حداکثر در این دو تبدیل نادیده گرفته شده است. سومین مقدار معتبر این روش minmax است که در آن پارامترهای حداقل و حداکثر در دامنه [0,1] باید تهیه شوند. اگر Minmax انتخاب شود سطوح در حیطه (حداقل و حداکثر) ترسیم می‌شوند. گرچه این مقادیر در حیطه [0,1] مشخص شده‌اند، برنامه مقیاس بندی صحیح را بسته به نوع داده‌های ورودی انتخاب می‌کند و بعد داده‌های خروجی را به همان نوع داده‌های ورودی تبدیل می‌کند. مثلاً اگر f از نوع uint8 باشد، و minmax را با حیطه [0,5] مشخص کنیم، داده‌های خروجی نیز از نوع ۸ خواهند بود. و مقادیری در حیطه [0,128] خواهد داشت. اگر f از نوع مضاعف باشد، و مقادیر آن خارج از حیطه [0,1] باشد، برنامه قبل از پیش رفتن آن را به این حیطه تبدیل می‌کند. تابع gscale در قسمتهای مختلف تحقیق به کار برده شده است.

3.3 پردازش پیشینه نما و ترسیم تابع(Histogram Processing & Function Plotting)

تابعهای تبدیل شدت مبتنی بر اطلاعات استخراج شده از سابقه‌نماهای شدت تصویر نقش اساسی در پردازش تصویر دارند مخصوصاً زمینه‌هایی مانند بهبود و تقویت، فشرده سازی، قطعه‌سازی، و توصیف. در این بخش به ساخت، ترسیم و استفاده از پیشینه نما(Histogram)ها برای بهبود و تقویت تصاویر می‌پردازیم. سایر کاربردهای پیشینه نما(Histogram) در فصلهای بعد بررسی می‌شوند.

۳.۳.۱. تولید و ترسیم پیشینه نما(Histogram)های تصویری

سابقه‌نمای یک تصویر دیجیتالی با Lعدد سطح تشدید (Intensity level)در حیطه [0,G] به صورت تابع متمایز زیر تعریفمی‌شود:

$$h(r_k)=n_k$$

در اینجا r_k , Kامین سطح تشدید در فاصله [0,G]و n_k تعداد عناصر تصویری است که سطح تشدید آن r_k است. مقدار g برای تصاویر نوع uint8 ۲۵۵ و برای تصاویر نوع uint16 ۶۵۵۳۵ و برای تصاویر double ۱ است. به خاطر داشته باشید که شاخص‌های نرم افزار MATLAB نمی‌توانند صفر باشند بنا بر این r_1 مطابق با شدت سطح صفر است و r_2 مطابق با شدت ۱ است، و r_l مطابق با سطح g است. توجه داشته باشید که $g = l - 1$ برای تصاویر نوع uint8 و ۱۶.

بهتر است با سابقه‌نماهای عادی کار کرد که از تقسیم کلیه عناصر hrk بر کل تعداد عناصر تصویری که مشخص می‌کنیم حاصل می‌شوند.

$$p(r_k)=\frac{h(r_k)}{n}$$

$$=\frac{nk}{n}$$

از این احتمال متوجه می‌شویم که $p(r_k)$ برآورد احتمال وقوع سطح تشدید r_k است. تابع اصلی حین کار با سابقه‌نمای تصویر imhist است که ترکیب زیر را دارد:

```
h = imhist(f, b)
```

در اینجا f تصویر ورودی است. H پیشینه نما(Histogram)ی آن است. $H(r_k)$, b تعداد مقادیر عددی مورد نیاز برای ساخت پیشینه نما(Histogram) است. (اگر bدر شناسه درج نشود، به صورت پیش فرض ۲۵۶ استفاده می‌شود). این مقدار عددی تقسیم بندی فرعی مقیاس شدت است. مثلاً اگر با تصاویر uint8 کار کنیم و b=2 باشد در این صورت، مقیاس شدت به دو حیطه ۰ تا ۱۲۷ و ۱۲۸ تا ۲۵۵ تقسیم می‌شود. پیشینه نما(Histogram)ی حاصل از آن ۲ مقدار خواهد داشت: h1 معادل تعداد عناصر تصویری در تصویر با مقادیر اصله‌های [0,128] و h2 معادل تعداد عناصر تصویری با مقادیر فواصل [128,255] می‌توان با استفاده از عبارت زیر پیشینه نما(Histogram)ی عادی را به دست آورد.

```
h = imhist(f, b)
```

به یاد داریم که تابع numelf تعداد عناصر آرایه f را مشخص می‌کند (یعنی تعداد عناصر تصویری موجود در تصویر)

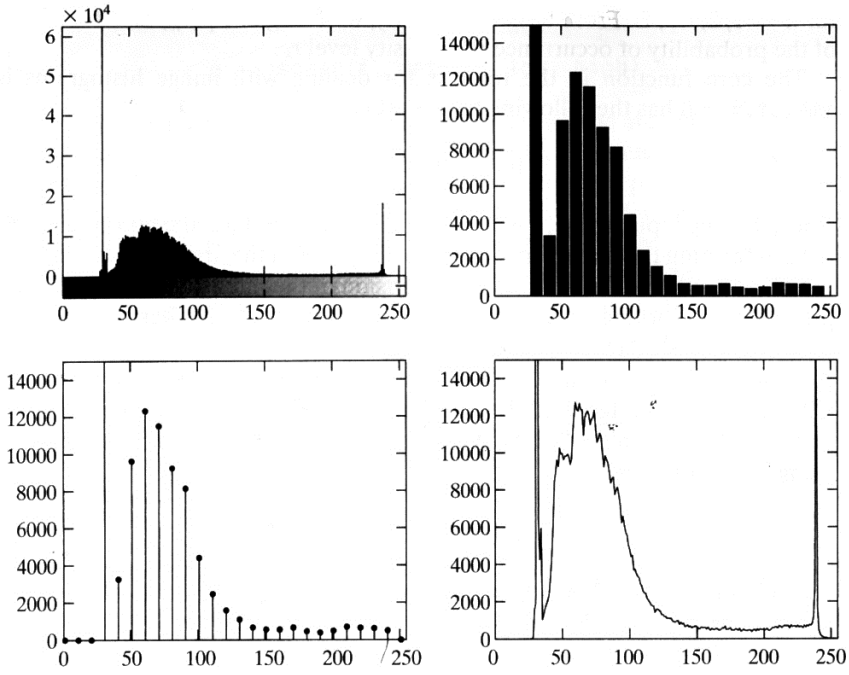
مثال۳.۴ :تصویر f را از عکس ۳.۳. (a) در نظر بگیرید. ساده‌ترین راه برای ترسیم سابقه‌نمای آن استفاده از imhist بدون مشخص کردن داده‌های خروجی است.

```
p = imhist(f, b)/numel(f)
```

نتیجه در تصویر ۳.۷ (a) نشان داده شده است. این مقدار پیش فرض پیشینه نما(Histogram) در جعبه ابزار است. ولی راههای دیگری نیز برای ترسیم پیشینه نما(Histogram) وجود دارد. در اینجا مواردی را شرح دادیم که نمایانگر روشهای مورد استفاده در پردازش تصویر هستند. معمولاً سابقه‌نماها با استفاده از نمودارهای میله‌ای(bar graphs) ترسیم می‌شوند. پس از تابع زیر استفاده کرد:

```
bar(horz, v, width)
```

در اینجا v بردار ردیفی است که نقاط ترسیم را دارد. Horz بردار همان بعد به صورت v است که مقادیر فزاینده مقیاس افقی را دارد و پهنا عددی بین صفر و ۱ است. اگر horz حذف شود، محور افقی به واحدهایی بین صفر تا طول v تقسیم می‌شود. وقتی پهنا ۱ باشد میله‌ها مماس هم هستند و وقتی صفر باشد میله‌ها صرفاً خطوط عمودی هستند که در تصویر ۳.۷ (a) دیده می‌شود. مقدار پیش فرض ۰.۸ است. هنگام ترسیم نمودار میله‌ای، معمولاً برای کاهش تفکیک پذیری محور افقی آن را تقسیم بر باندها می‌کنند. عبارتهای زیر یک نمودار میله‌ای ایجاد می‌کند و محور افقی به گروه‌های ۱۰ سطحی تقسیم می‌شود.



a b
c d

نتایج در تصویر ۳.۷. (b)نشان داده شده است. نقطه اوج مقیاس تشدید تصویر ۳.۷ (a) در نمودار میله‌ای دیده نمی‌شود علت آن مقادیر فزاینده افقی ترسیم شده در این مورد است.

```
» h = imhist(f);
» h1 = h(1:10:256);
horz = 1:10:256;
» bar(horz, h1)
» axis([0 255 0 15000])
>> set(gca, 'xtick', 0:50:255)
```

<<عبارت پنجم کد قبلی برای توسعه حیطه تحتانی محور عمودی در تحلیل تصاویر `set(gca, 'ytick', 0:2000:15000)` به کار رفت. و محور افقی را در همان دامنه تصویر ۳.۷ (a) قرار داد. تابع محور ترکیب زیر را دارد:

```
Axis ([horzmin  horzmax  vermin vermax])
```

که مقادیر حداقل و حداکثر محورهای افقی و عمودی را تعیین می‌کند. Gca در ۲ عبارت آخر یعنی اخذ محور کنونی (یعنی محورهای تصویری که آخرین بار نمایش داده شده است). Xtick, ytick نشانه‌های محور افقی و عمودی را در فواصل نشان داده شده مشخص می‌کند.برچسب محور را می‌توان با استفاده از تابع‌ها به محورهای افقی و عمودی افزود.

```
xlabel('text string', 'fontsize', size)
xlabel('text string', 'fontsize', size)
```

در اینجا منظور از اندازه، اندازه قلم نقطه‌ها است. با استفاده از تابع text به شکل زیر می‌توان متنها را به تصویر افزود.

```
text(xloc, yloc, 'text string', 'fontsize', size)
```

Xloc, yloc موضعی را تعریف می‌کنند که متن در آن شروع می‌شود. استفاده از این ۳ تابع در مثال ۳.۵ تشریح شده است. مهم آن است که تابعهایی که مقادیر محور و برچسبها را تعیین می‌کنند بعد از ترسیم تابع استفاده می‌شوند. می‌توان با تابع title عنوانی را به ترسیم شده افزود که ترکیب اساسی آن به شکل زیر باشد:

```
title('titlestring')
```

در اینجا titlestring رشته نویسه‌هایی است که در عنوان بالای ترسیم تصویر رسم می‌شوند.

نمودار سقه‌دار(stem graph) شبیه به نمودار میله‌ای است.

```
stem(horz, v, 'color_linestyle_marker', 'fill')
```

در اینجا v بردار ردیف حاوی نقاط ترسیم است و horz برای این میله توصیف شده است. شناسه به شرح زیر است:

```
color_linestyle_marker,
```

این مقادیر سه گانه جدول ۳.۱ است مثلاً میله (v,'r- -s') میله‌ای ترسیم می‌کند که خطوط و علائم آن قرمز هستند. خط فاصله دارد و علائم مربع هستند. اگر از fill استفاده شود نشانگر دایره، مربع یا مورب باشد، در این صورت با رنگ مشخص شده در color پر می‌شود.

رنگ پیش فرض مشکی است. حالت پیش فرض خط ممتد است و نشانگر پیش فرض دایره‌ای است. نمودار میله‌ای در تصویر۳.۷. (c)با عبارتهای زیر به دست آمد.

```
» h = imhist(f);
 »h1 = h(1:10:256)
```

Symbol	Color	Symbol	Line Style	Symbol	Marker
k	Black	–	Solid	+	Plus sign
w	White	--	Dashed	o	Circle
r	Red	:	Dotted	*	Asterisk
g	Green	-.	Dash-dot	.	Point
b	Blue	none	No line	x	Cross
c	Cyan			s	Square
y	Yellow			d	Diamond
m	Magenta			none	No marker

```
» horz = 1:10:256;
» stem(horz, h1, 'fill')
» axis([0 255 0 15000])
» set(gca, 'xtick', [0:50:255])
>> set(gca, 'ytick', [0:2000:15000])
```

در آخر تابع plot را در نظر می‌گیریم که یک مجموعه نقطه را با پیوند دادن آنها با خطوط مستقیم ترسیم می‌کند.

```
plot(horz, v, 'color_linestyle_marker')
```

در اینجا شناسه‌ها برای نمودار میله‌ای تعریف شده است. مقادیر تابعهای رنگ، سبک خط و نشانگر در جدول ۳.۱ ارائه شده است. در این میله می‌بینیم که خصوصیات این ترسیم را می‌توان در قسمت سه بخشی مشخص کرد. وقتی از none برای سبک خط یا نشانگر استفاده شود، خصوصیات باید انفرادی مشخص شوند مثلاً فرمان زیر:

```
>> plot(horz, v, 'color', 'g', 'linestyle', 'none', 'marker', 's')
```

مربعهای سبز را بدون ایجاد خطوط راهنما بین آنها ترسیم می‌کند. پیش فرض این ترسیم خطوط مشکی ممتد بدون نشانگر هستند. تصویر ترسیم شده در عکس ۳.۷. د با استفاده از عبارتهای زیر به دست آمد.

```
>> h = imhist(f);
>> plot(h)  % Use the default values.
 >> axis([0 255 0 15000])
>> set(gca, 'xtick', [0:50:255])
>> set(gca, 'ytick', [0:2000:15000])
```

تابع plot برای نمایش تابع‌های تبدیل زیاد به کار برده می‌شود. (مثال ۳.۵)

در مبحث قبل محدوده محور و نشانگرها به صورت دستی تعیین شد. می‌توان با استفاده از تابعهای ylim, xlim محدوه‌ها و تیکها را به صورت خودکار مشخص کرد. در این مثال ترکیب به شکل زیر است:

```
ylim('auto')
xlim('auto')
```

یکی از انواع ترکیبهای این و تابع گزینه دستی به شرح زیر است: (برای جزئیات به تحقیقچه راهنما اینترنتی مراجعه کنید)

```
ylim([ymin ymax])
xlim([xmin xmax])
```

که می‌توان خصوصیات محدوه‌ها را به صورت دستی مشخص کرد. اگر محدوه‌ها فقط برای یک محور مشخص شده باشند، محدوده‌های محور دیگر به صورت پیش فرض روی حالت خودکار قرار داده می‌شوند. این تابع‌ها را در بخش زیر استفاده می‌کنیم:

وقتی عبارت hold on را در خط فرمان تایپ کنیم، تابع ترسیم و برخی خصوصیات محور را نگه می‌دارد طوری که فرمانهای ترسیم به سایر جنبه‌های نمودار می‌افزایند. مثلاً به تصویر ۱۰.۶ مراجعه کنید.

۳.۳.۲. تعادل پیشینه نما(Histogram Equlization)

فرض کنید سطوح تشدید مقادیر پیوسته‌ای هستند که در حیطه ۰و ۱ هنجارمند شده‌اند. و pr تابع چگالی احتمال pdf در سطوح تشدید در یک تصویر را مشخص می‌کند و زیرنویس برای تفاوت بین pdfها تصویرهای ورودی و خروجی به کار برده می‌شود. فرض کنید برای به دست آوردن سطوح تشدید پردازش شده خروجی تبدیل زیر را روی سطوح داده‌های ورودی انجام می‌دهیم.

$$s = T(r) = \int_o^r p_r(w)dw$$

در اینجا w یک متغیر ساختگی ادغام کننده است. می‌توان اثبات کرد که تابع چگالی احتمالی سطوح داده‌های خروجی یکسان است یعنی:

$$ps(s)\begin{cases}1 & for\, 0 \le s \le 1\\ 0 & otherwise\end{cases}$$

به عبارت دیگر تبدیل قبل تصویری ایجاد می‌کند که سطوح تشدید آن همانند هستند و کل حیطه [0,1] را در بر می‌گیرد. نتیجه فرایند برابر کردن(equalization) سطوح تشدید افزایش خصوصیات پویای تصویر است که وضوح بیشتری نیز دارد. توجه داشته باشید که تابع تبدیل قابلیتی بیشتر از تابع توزیع انباشت(CDF) Cummulative distribution function ندارد.

وقتی با مقادیر متمایز سر و کار داریم، با پیشینه نما(Histogram) کار می‌کنیم و این روش را برابرسازی پیشینه نما(Histogram) می‌نامیم. گرچه پیشینه نما(Histogram)ی تصویر پردازش شده منظم نیست، علت آن ماهیت متمایز متغیرها است. با توجه به مباحث

بخش ۳.۳.۱ l پیشینه نما(Histogram)ی مربوط به سطوح تشدید یک تصویر را نشان می‌دهد و مشخص می‌کند که مقادیر پیشینه نما(Histogram)ی هنجارمند نزدیک به هر سطح تشدید در تصویر هستند. در مقادیر متمایز با جمع اعداد سر و کار داریم. و تبدیل برابرها به شکل زیرمی‌شود: $sk = T(rk)$

$$= \sum_{j=1}^{k} pr(rj)$$

$$= \sum_{j=1}^{k} \frac{nj}{n}$$

در اینجا s_k مقدار تشدید در تصویر پردازش شده خروجی مطابق با مقدار r_k در تصویر ورودی است.

برابرسازی سابقه‌نما در جعبه ابزار با تابع histeq که ترکیب زیر را دارد مشخص می‌شود:

```
g=histeq(f,nlev)
```

در اینجا f تصویر ورودی و nlev تعداد سطوح تشدید است که برای تصویر خروجی مشخص شده است. اگر nlev معادل l باشد (یعنی کل سطوح ممکن در تصویر ورودی) در این صورت، histeq تابع تبدیل را اجرا می‌کند. اگر nlev کمتر از l باشد histeq سطوح را طوری توزیع می‌کند که نزدیک به یک پیشینه نما(Histogram)ی صاف باشند. مقدار پیش فرض در histeq بر خلاف imhist معادل =nlev 64 است. در اکثر موارد حداکثر مقدار سطوح (معمولاً ۲۵۶) را برای nlev استفاده می‌کنیم زیرا در این صورت، روش برابرسازی پیشینه نما(Histogram) درست اجرا می‌شود.

مثال ۳.۵:

تصویر ۳.۸. (a) تصویر میکروسکوپی الکترون گرده است که تقریباً ۷۰۰ بار بزرگنمایی شده است. این تصویر تیره است و خصوصیات پویای اندک دارد بنا بر این از این لحاظ باید بهبود و تقویت شود. این موضوع در پیشینه نما(Histogram)ی تصویر ۳.۸. (b)دیده می‌شود. پیشینه نما(Histogram) به سمت قسمت تیره مقیاس خاکستری است. پهنای پیشینه نما(Histogram) نسبت به کل مقیاس خاکستری باریک است و از همین موضوع به خصوصیات پویای اندک آن پی می‌بریم. اگر f تصویر ورودی باشد، تصویر ۳.۸. (a) از طریق دی با مراحل زیر ایجاد می‌شود.

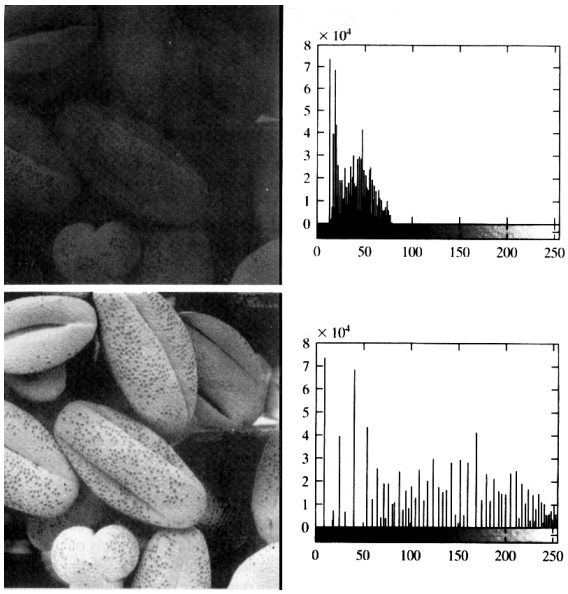
```
>> imshow (f )
>> figure, imhist (f)
>> ylim ('auto')
>> g = histeq (f, 256) ;
>> figure , imshow (g)
>> figure, imhist (g)
>> ylim ('auto')
```

این تصویرها با پسوند tif با وضوح ۳۰۰ نقطه در اینچ و با استفاده از imwrite در دیسک ذخیره شدند. با استفاده از تابع چاپ در بخش ۲.۴ موارد ترسیم شده به دیسک صادر شد.

نتایج برابرسازی پیشینه نما(Histogram) در تصویر ۳.۸. (c)نشان داده شده است. بهبود میانگین شدت و وضوح تصویر مشهود است. این خصوصیات در پیشینه نما(Histogram)ی تصویر ۳.۸. دی نشان داده شده است. علت افزایش وضوح تصویر گسترش پیشینه نما(Histogram) به تمام مقیاسهای تشدید است. علت این افزایش آن است که میانگین سطح تشدید پیشینه نما(Histogram)ی تصویر برابرسازی شده بیشتر یا روشن‌تر از تصویر اصلی است. گرچه در روش برابرسازی پیشینه نما(Histogram) یک پیشینه نما(Histogram)ی هموار ایجاد نمی‌شود ولی با خصوصیات آن می‌توان دامنه پویای سطوح تشدید تصویر را افزایش داد.

همان طور که قبلاً گفتیم، تابع تبدیل $T(r_k)$ مجموع مقادیر هنجارمند شده پیشینه نما(Histogram) است. می‌توان از تابع cumsum برای به دست آوردن تابع تبدیل به شرح زیر استفاده کرد.

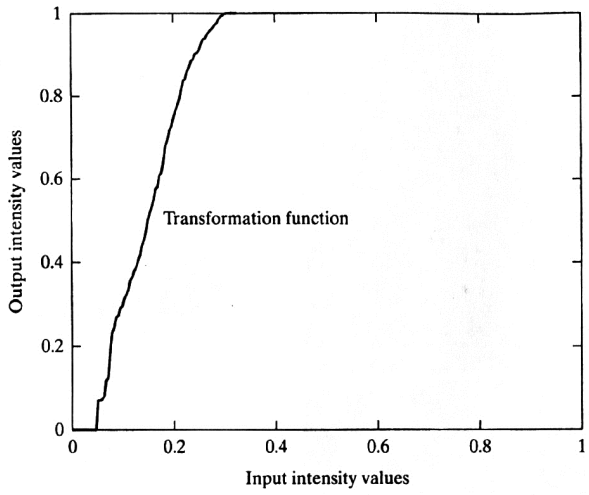
```
>> hnorm =  imhist (f) . / numel (f) ;
>> cdf = cumsum (hnorm) ;
```



a b

c d

```
>> x = linspace (0, 1 , 256) ;        % Intervals for  [ 0, 1 ] horiz scale. Note
                                 %  the use of linspace from sec. 2.8.1.
>> Plot (x , cdf)              % Plot cdf  vs. x.
>> axis [ 0 1 0 1]                    % Scale , settings , and labels:
>> set (gca, 'xtick' , 0 : .2: 1)
>> set (gca, 'ytick' , 0: .2:1)
>> xlabet ('Input intensity values" , 'fontsize' , 9)
>> ylabet ('Output intensity values" , 'fontsize' , 9)
>> % Specify text in the body of the graph:
>> text (0. 18, 0.5, 'Transform ting function' , 'fontsize' , 9)
```



۳.۳.۳. تطابق خصوصیات پیشینه نما(Histogram)

برابرسازی سابقه‌نما تابع تبدیلی ایجاد می‌کند که سازگار است یعنی بر اساس سابقه‌نمای یک تصویر مشخص است. ولی بعد از محاسبه تابع تبدیل یک تصویر فقط در صورتی که پیشینه نما(Histogram)ی تصویر تغییر کند آن هم تغییر می‌کند. همان طور که در بخش قبل گفتیم برای بهبود و تقویت برابرسازی سابقه‌نما سطوح تصویر ورودی در یک مقیاس تشدید کننده گسترده پراکنده می‌شود. در اینجا نشان می‌دهیم که این کار همیشه نتایج موفقیت آمیز در بر ندارد. در برخی برنامه‌ها باید شکل سابقه‌نمایی را که می‌خواهیم تصویر پردازش شده داشته باشد مشخص کرد. روش ایجاد تصویر پردازش شده که سابقه‌نمای خاصی دارد تطابق سابقه‌نما(histogram matching) یا خصوصیات پیشینه نما(Histogram) نامیده می‌شود. اصل این روش ساده است. سطوح پیوسته‌ای را در نظر بگیرید که در فاصله ۰ تا ۱ هنجارمند شده‌اند. R, z سطوح تشدید تصویرهای ورودی و خروجی را مشخص می‌کنند. تابع چگالی سطوح ورودی $p_r(r)$ و تابع چگالی سطوح خروجی $p_z(z)$ است. از مباحث بخش قبل متوجه شدیم که تبدیل زیر

$$s = T(R) = \int_o^r pr(w)dw$$

منجر به تشکیل سطوح تشدید S می‌شود که تابع چگالی احتمالی آنها $p_s(s)$تغییرناپذیر است. فرض کنید متغیری به نام z را با خصوصیات زیر تعریف می‌کنیم:

$$H(z) = \int_o^z pz(w)dw = s$$

به خاطر داشته باشید که به دنبال تصویری با شدت z هستیم، که چگالی خاص آن $p_z(z)$ است. از دو معادله قبل به نتیجه زیر می‌رسیم:

$$z = H^{-1}(s) = H^{-1}[T(r)]$$

$T(r)$ را می‌توان از تصویر ورودی به دست آورد (این همان تبدیل برابرهای سابقه‌نما است که در بخش قبل آن را بحث کردیم). بنا بر این برای یافتن سطوح تبدیل شده z که pdf آن مادامی که بتوانیم H^{-1} را پیدا کنیم $p_z(z)$ است می‌توان از معادله قبل استفاده کرد. حین کار با متغیرهای متمایز می‌توان تضمین کرد که اگر $p_z(z)$ یک سابقه‌نمای معتبر باشد، معکوس H نیز وجود دارد (یعنی واحد مساحت دارد و کلیه مقادیر آن غیرمنفی هستند) و هیچ یک از اعضای آن صفر نیست (یعنی هیچ یک از مقادیر عددی $p_z(z)$ خالی نیست) اجرای متمایز روش قبل همچون برابرسازی سابقه‌نما نتیجه‌ای تولید می‌کند که نزدیک به سابقه‌نمای مورد نظر است.

جعبه ابزار با استفاده از ترکیب زیر در histeq موارد مطابق با سابقه‌نما را ایجاد می‌کند.

$$g = histeq(f, hspec)$$

در اینجا f تصویر ورودی است. hspec سابقه‌نمای مورد نظر است (که یک ردیف بردار از مقادیر مشخص دارد)، و g تصویر خروجی است که سابقه‌نمای آن شبیه به سابقه‌نمای hspec است. این بردار باید شمارشهای عدد صحیح مطابق با مقادیر عددی با فاصله‌های مساوی داشته باشد. یکی از خصوصیات histeq آن است که سابقه‌نمای g وقتی طول hspec بسیار کمتر از تعداد سطوح تشدید f است بهتر با hspec تطبیق دارد.

مثال ۳.۶ :

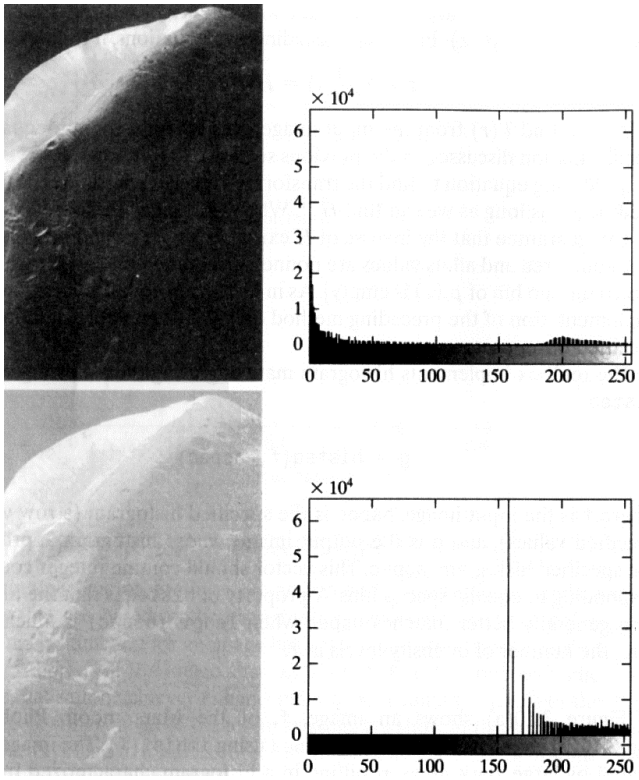
یک تصویر از قمر مریخ به نام فوبوس در عکس ۳.۱۰. (a) دیده می‌شود. در تصویر ۳.۱۰. (b)سابقه‌نمای آن را می‌بینیم که با استفاده از imhistf به دست آمده است. این تصویر نقاط تیره بزرگی دارد که سابقه‌نمایی ایجاد کرده است که تراکم عنصرهای تصویری در انتهای تیره مقیاس خاکستری است. در نگاه اول ممکن است چنین نتیجه بگیریم که برابرسازی سابقه‌نما روش خوبی برای بهبود و تقویت این تصویر است طوری که جزئیات مواضع تیره واضح‌تر می‌شوند. ولی نتیجه تصویر ۳.۱۰.(c)با استفاده از فرمان زیر

$$>> f1 = histeq(f, 256);$$

نشان می‌دهد که برابرسازی سابقه‌نما در این مورد نتیجه مطلوبی ایجاد نکرده است. برای پی بردن به علت این موضوع باید سابقه‌نمای تصویر ۳.۱۰(d) که برابرسازی شده است مطالعه شود. در اینجا می‌بینیم که سطوح تشدید به نیمه فوقانی مقیاس بندی خاکستری منتقل شده است و باعث می‌شود که ظاهر تصویر کمرنگ شود. علت این انتقال تراکم اجزای تیره نزدیک صفر در سابقه‌نمای اصلی است. در عوض تابع تبدیل یکجای حاصل از این سابقه‌نما شیبدار است بنا بر این تراکم عناصر تصویری انتهای پایینی مقیاس خاکستری را تا انتهای فوقانی ترسیم می‌کند.

a b

c d



یک راه حل این مسئله تطبیق سابقه‌نماها است طوری که تراکم اجزا در سابقه‌نما مطلوب در انتهای تحتانی مقیاس خاکستری باشد و شکل سابقه‌نما تصویر اصلی حفظ شود. از تصویر ۳.۱۰. (b)به این نتیجه رسیدیم که این سابقه‌نما اصولاً ۲ حالته است. یک حالت اصلی بزرگ و یک حالت کوچک در انتهای فوقانی مقیاس خاکستری دارد. این نوع سابقه‌نماها را می‌توان با استفاده از تابع‌های گوسی چند حالته الگوبرداری کرد. تابع M زیر یک تابع گوسی چند حالته را در واحد سطح هنجارمند می‌کند تا بتوان آن را به صورت یک سابقه‌نما خاص به کار برد.

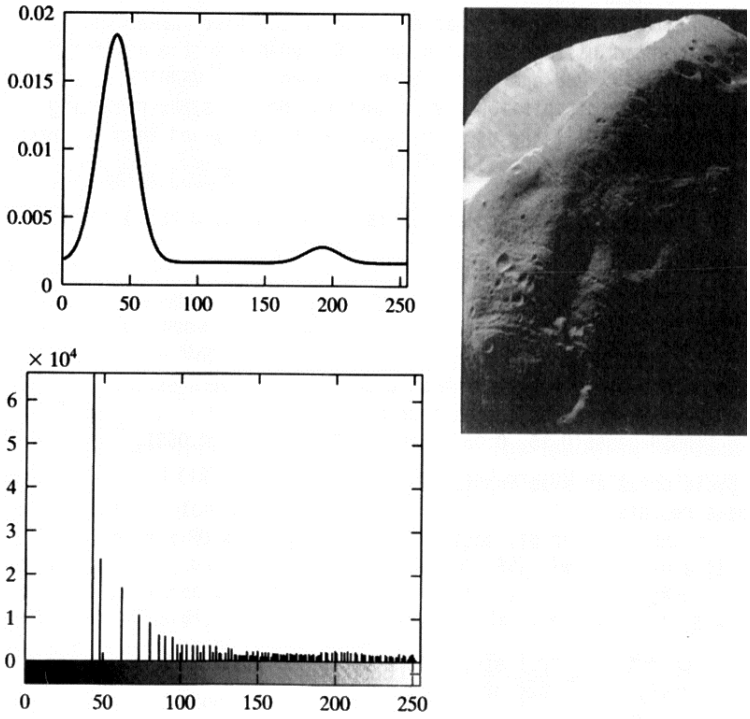
```
function p = twomodegauss(m1, sig1, m2, sig2, Al, A2, k)
%TWOMODEGAUSS Generates a bimodal Gaussian function.
%     P = TWOMODEGAUSS(M1, SIG1, M2, SIG2, Al, A2, K) generates a bimodal,
%     Gaussian-like function in the interval [0, 1]. P is a 256-element
%     vector normalized so that SUM(P) equals 1. The mean and standard
%     deviation of the modes are (M1, SIG1) and (M2, SIG2), respectively.
%     Al and A2 are the amplitude values of the two modes. Since the
%     output is normalized, only the relative values of Al and A2 are
%     important. K is an offset value that raises the "floor" of the
%     function. A good set of values to try is M1 = 0.15, SIG1    0.05,
%     M2 = 0.75, SIG2 = 0.05, Al = 1, A2 = 0.07, and K = 0.002.
c1 = Al * (1      ((2 * pi) ^ 0.5) * sigl); k1 = 2 * (sig1 ^ 2);
c2 = A2 * (1 / ((2 * pi) ^ 0.5) * sig2); k2 = 2 * (sig2 ^ 2);
z = linspace(0, 1, 256);
p = k + c1 * exp(–((z – ml) .^ 2) ./ k1) + … c2 * exp(–((z – m2) . 2) ./ k2);
P = p ./ sum(P(:));
```

تابع محاوره‌ای زیر داده‌های ورودی را از صفحه کلید دریافت می‌کند و تابع گوسی حاصله را ترسیم می‌کند. برای تشریح تابعهای input, str2num به بخش ۲.۱۰.۵ مراجعه کنید. به محدوده موارد ترسیم شده توجه کنید.

```
function p = manualhist
(%MANUALHIST Generates a bimodal histogram interactively.
%     P = MANUALHIST generates a bimodal histogram using
%     TWOMODEGAUSS(m1, sig1, m2, sig2, Al, A2, k). m1 and m2 are the means
%     of the two modes and must be in the range [0, 1]. sig1 and sig2 are
%     the standard deviations of the two modes. Al and A2 are
%     amplitude values, and k is an offset value that raises the
%     "floor" of histogram. The number of elements in the histogram
%     vector P is 256 and sum(P) is normalized to 1. MANUALHIST
%     repeatedly prompts for the parameters and plots the resulting
%     histogram until the user types an 'x' to quit, and then it returns the
%     last histogram computed.
%     A good set of starting values is: (0.15, 0.05, 0.75, 0.05, 1,
%     0.07, 0.002).
Initialize. repeats = true; quitnow = 'x';
% Compute a default histogram in case the user quits before % estimating at least
one histogram.
p = twomodegauss(0.15, 0.05, 0.75, 0.05, 1, 0.07, 0.002);
Cycle until an x is input.
while repeats
s = input('Enter m1, sig1, m2, sig2, Al, A2, k OR x to quit:', 's'); if s ==
quitnow
break end
% Convert the input string to a vector of numerical values and % verify the number
of inputs.
v = str2num(s);
if numel(v) -= 7
disp('Incorrect number of inputs.') continue
end
p = twomodegauss(v(1), v(2), v(3), v(4), v(5), v(6), v(7));
% Start a new figure and scale the axes. Specifying only xlim % leaves ylim on
auto.
figure, plot(p)
xlim([0 255])
end
```

از آنجائی که مسئله برابرسازی سابقه‌نما در این مثال به علت تراکم عناصر تصویری در تصویر اصلی با سطوح نزدیک به صفر است روش منطقی آن است که سابقه‌نما آن تصویر طوری تغییر داده شود که این خصوصیت را نداشته باشد. در تصویر ۳.۱۱. (a) ترسیمی از تابعی را می‌بینیم که با برنامه manualhist به دست آمده است که شکل کلی سابقه‌نما اصلی را حفظ می‌کند ولی انتقال سطوح آن در ناحیه تیره مقیاس تشدید است. داده‌های خروجی برنامه p از ۲۵۶ نقطه تشکیل شده است که فاصله یکسانی تا این تابع دارند و سابقه‌نما مطلوب مورد نظر است. با استفاده از این فرمان تصویری با سابقه‌نما خاص ایجاد شد.

```
g=histeq(f,p)
```



a b
c

نتیجه در تصویر ۳.۱۱. (b)دیده می‌شود. با مقایسه این ۲ تصویر می‌توان بهبود نتیجه‌ای را که سابقه‌نما آن برابرسازی شده است مشاهده کرد. توجه داشته باشید که این سابقه‌نما خاص نسبت به سابقه‌نما اصلی اندکی تغییر کرده است. نکته اصلی برای بهبود و تقویت آن همین تغییر است. در تصویر ۳.۱۱.(c)سابقه‌نما تصویر ۳.۱۱. (b)دیده می‌شود. متمایزترین خصوصیت این سابقه‌نما نحوه نزدیک شدن انتهای تحتانی به ناحیه روشن مقیاس خاکستری است که در نتیجه به شکل مورد نظر نیز نزدیکتر شده است. توجه داشته باشید که تغییر مکان به سمت راست به اندازه تغییر مکانی که در سابقه‌نما ۳.۱۰. (b)دیده می‌شود نیست که مطابق با تصویر ۳.۱۰.(c)است که اندکی بهبود و تقویت شده است.

۳.۴. غربال مواضع فضایی(Spatial Filtering)

همان طور که در بخش ۳.۱ گفتیم و در تصویر ۳.۱ نشان دادیم پردازش نقاط همجوار از موارد زیر تشکیل می‌شود:

(۱) تعریف نقطه کانونی (x, y)

(۲) انجام عملیاتی که فقط شامل عناصر تصویری یک موضع از پیش تعریف شده حوالی آن نقطه کانونی باشد

(۳) نتیجه آن عملیات واکنش این پردازش در آن نقطه باشد

(4) تکرار این پردازش برای هر نقطه در آن تصویر. فرایند انتقال نقاط کانونی مواضع جدیدی برای هر یک از عناصر تصویری موجود در تصویر ورودی ایجاد می‌کند.

دو عبارت پردازش همجوار(neighborhood processing) و غربال مواضع فضایی (spatial filtering)دو عبارتی هستند که برای شناسایی این عملیات به کار برده می‌شوند و عبارت دوم متداولتر است. همان طور که در بخش زیر توضیح دادیم، اگر محاسبات عناصر تصویری همجوار خطی باشند، این عملیات را غربال کردن خطی مواضع فضایی(linear spatial filtering) می‌نامیم (عبارت تلفیق فضایی نیز در این مورد به کار برده می‌شد) در غیر این صورت غربال سازی غیرخطی فضایی نامیده می‌شود.

۳.۴.۱. غربال سازی خطی مواضع فضایی(Linear Spatial Filtering)

ریشه عبارت غربال‌سازی خطی(linear filtering) در تبدیلهای فوریه برای پردازش علائم در دامنه فرکانس است. این موضوعی است که مفصلاً در فصل ۴ بحث شده است. در این فصل به غربال‌سازیهایی می‌پردازیم که مستقیماً روی عناصر تصویری انجام می‌شوند. استفاده از عبارت غربال‌سازی خطی مواضع فضایی این فرایند را از غربال‌سازی دامنه فرکانس متمایز می‌کند.

برخی از عملیات خطی این فصل عبارتند از: ضرب هر عنصر تصویری همجوار در ضریب مطابق با آن و جمع بستن نتایج برای به دست آوردن پاسخ در هر نقطه (x, y) اگر اندازه قسمت همجوار $m * n$ باشد ضرایب mn مورد نیاز هستند. این ضرایب به صورت ماتریس مرتب می‌شوند و آنها را فیلتر(Filter)، نقاب (Mask), نقاب فیلتر(Filter mask)، هسته(Kernel)، الگو(template)، یا کادر(window) می‌نامیم و سه عبارت اول متداولتر هستند. بنا به برخی دلایل واضح از کلمات فیلتر تلفیق، نقاب (Mask)، و هسته نیز استفاده می‌شود.

خصوصیات مکانیکی غربال‌سازی خطی مواضع فضایی در تصویر ۳.۱۲ نشان داده شده است. در این فرایند کانون فیلتر w در تصویر f در مواضع گوناگون جابجا می‌شود. در هر نقطه با مختصات (x, y) واکنش فیلتر در آن نقطه مجموع محصولات ضرایب فیلتر و عناصر تصویری مجاور آن است که از فیلتر عبور می‌کنند. اگر اندازه نقاب (Mask) $m * n$ باشد معمولاً فرض می‌کنیم که $m = 2a + 1, n = 2b + 1$. در اینجا a, b اعداد صحیح غیرمنفی هستند.

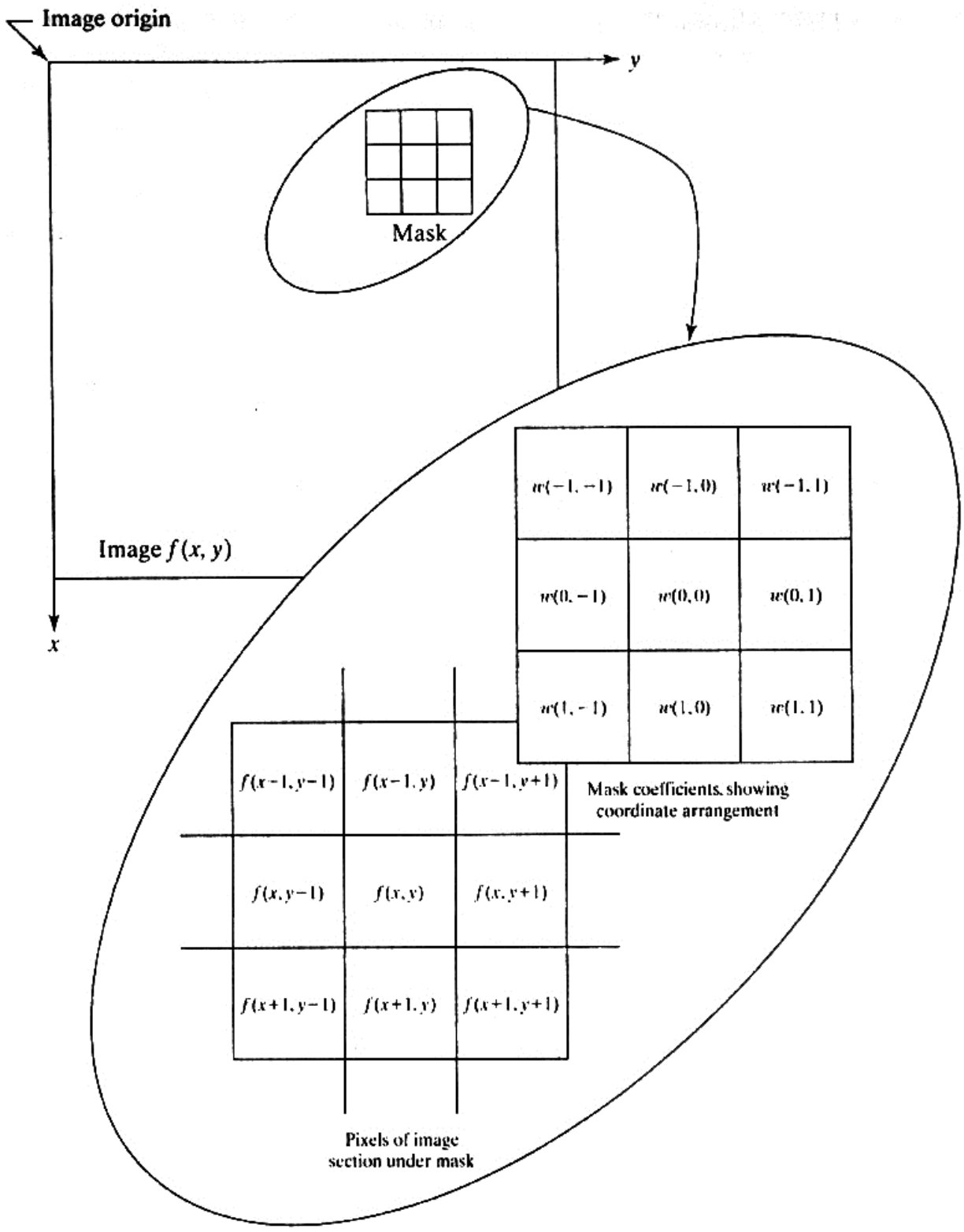


Image origin
y
Mask
Image f(x, y)
x
w(−1, −1)
w(−1, 0)
w(−1, 1)
w(0, −1)
w(0, 0)
w(0, 1)
w(1, −1)
w(1, 0)
w(1, 1)
Mask coefficients, showing coordinate arrangement
f(x−1, y−1)
f(x−1, y)
f(x−1, y+1)
f(x, y−1)
f(x, y)
f(x, y+1)
f(x+1, y−1)
f(x+1, y)
f(x+1, y+1)
Pixels of image section under mask

در اینجا روی نقاب (Mask)هایی با اندازه‌های فرد متمرکز هستیم که اندازه کوچکترین جزء معنی‌دار ۳ * ۳ است (مورد جزئی نقاب (Mask) ۱ * ۱ را از این مثال حذف کرده‌ایم). این یکی از شرایط نیست ولی کار با نقاب (Mask)های اندازه فرد خودانگیخته انجام می‌شود زیرا آنها نقطه کانونی منحصر به فرد دارند.

دو فرضیه مرتبط وجود دارد که باید حین اجرای غربال‌سازی خطی مواضع فضایی به خوبی درک شوند. یکی همبستگی(Correlation) و دیگری تلفیق(Convolution) است.

همبستگی یعنی فرایند عبور دادن نقاب (Mask) w از آرایه تصویر f به روش توصیف شده در تصویر ۳.۱۲. تلفیق از نظر مکانیکی همین فرایند است تنها فرقش آن است که w قبل از عبور از f ۱۸۰ درجه چرخانده می‌شود. این ۲ فرضیه را با چند مثال ساده شرح می‌دهیم.

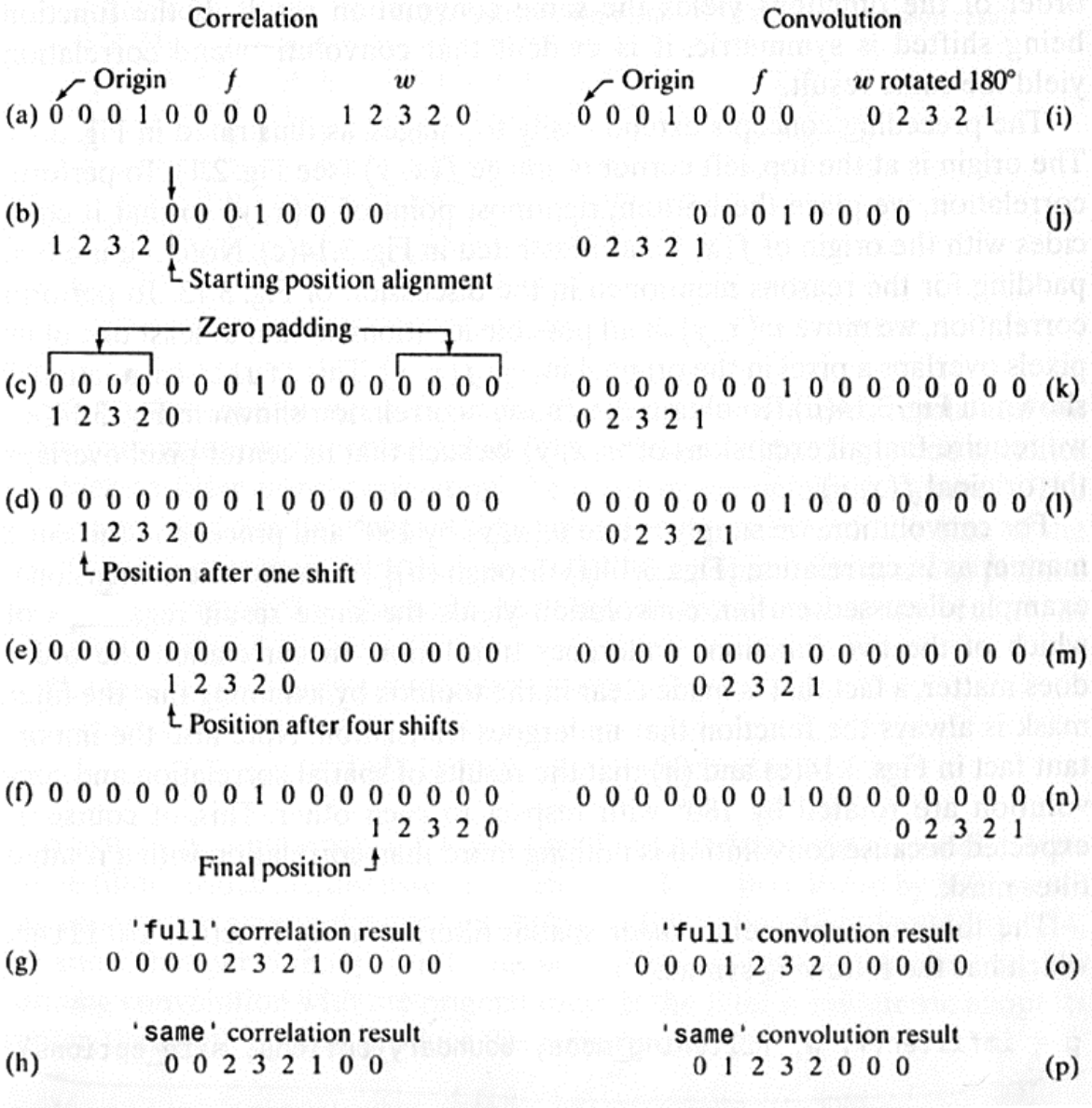
تابع تک بعدی f و نقاب (Mask) w در تصویر ۳.۱۳. (a) دیده می‌شوند. نقطه منتها الیه سمت چپ ریشه f است. برای همبستگی این ۲ تابع w را طوری حرکت می‌دهیم که نقطه منتها الیه سمت راست آن با خاستگاه f همان طور که در تصویر ۳.۱۳. (b)دیده می‌شود هماهنگ باشد.

توجه داشته باشید که نقاطی بین این ۲ تابع وجود دارند که با یکدیگر همپوشی ندارند. متداولترین روش برای حل این مسئله آن است که صفرهای زیادی را به f تخصیص دهیم تا همیشه برای انتقال w از کنار f نقاط مطابق وجود داشته باشد.

حالا آماده انجام همبستگی هستیم. اولین مقدار این همبستگی مجموع محصولات دو تابع در موقعیت نشان داده شده در تصویر ۳.۱۳. (c)است. مجموع محصولات در این مثال صفر است.

حالا w را یک درجه به راست جابجا می‌کنیم و فرایند را تکرار می‌کنیم (تصویر ۳.۱۳.د). مجدّداً مجموع محصولات صفر است. پس از ۴ جابجایی (تصویر ۳.۱۳.ای) با اولین مقدار غیرصفر همبستگی برخورد می‌کنیم که $2=(1)(2)$ است اگر آنقدر ادامه دهیم که w کاملاً از f عبور کند (انتهای شکلهای هندسی در تصویر ۳.۱۳. f نشان داده شده است) به نتایج تصویر ۳.۱۳ g دست می‌یابیم.

این مجموعه مقادیر همبستگی w, f است. توجه داشته باشید که در صورتی که w را در حالت ساکن رها می‌کردیم، و در عوض f را از w عبور می‌دادیم نتایج مختلف می‌شد بنا بر این ترتیب آنها مهم است.



برچسب full در همبستگی تصویر ۳.۱۳. g یک علامت است (که بعداً بحث خواهد شد) که در جعبه ابزار برای نشاندادن تصاویر توسعه یافته به کار برده می‌شود که با روش قبلی محاسبه شده‌اند. این جعبه ابزار گزینه دیگری به نام same دارد (تصویر ۳.۱۳.ایچ) که همبستگی به همان اندازه f ایجاد می‌کند. در این محاسبه نیز از لایی‌گذاری صفر استفاده می‌شود. ولی شروع کار در نقطه کانونی نقاب (Mask) است (نقطه‌ای که برچسب ۳ را در w دارد) که همراستا با خاستگاه f است. آخرین محاسبه در کانون نقاب (Mask) همراستا با آخرین نقطه f است.

برای تلفیق کردن w را ۱۸۰ درجه می‌چرخانیم و همان طور که در تصویر ۳.۱۳. g دیده می‌شود نقطه منتها الیه سمت راست آن را در خاستگاه f قرار می‌دهیم. سپس فرایند حرکت کشویی و محاسبه همبستگی را همان طور که در تصویرهای ۳.۱۳ k تا n دیده می‌شود تکرار می‌کنیم. نتایج تلفیق full, same به ترتیب در تصویرهای ۳.۱۳ o ,p نشان داده شده است.

تابع f در تصویر ۳.۱۳ یک تابع متمایز است که در یک موضع ۱ و در سایر مواضع صفر است. از نتایج تصویرهای ۳.۱۳ o,p مشهود است که در این تلفیق w در موضع محرک آنی کپی شده است. این خصوصیت کپی کردن ساده (که آن را غربال کردن می‌نامیم) نکته اصلی فرضیه‌های سیستم‌های خطی است. به همین دلیل است که یکی از تابع‌ها در این تلفیق همیشه ۱۸۰ درجه می‌چرخد. توجه داشته باشید که معکوس‌سازی ترتیب تابع‌ها بر خلاف همبستگی نتایج تلفیق یکسان تولید می‌کند. اگر تابع منتقل شده متقارن باشد، پر واضح است که تلفیق و همبستگی نتایج یکسان دارند.

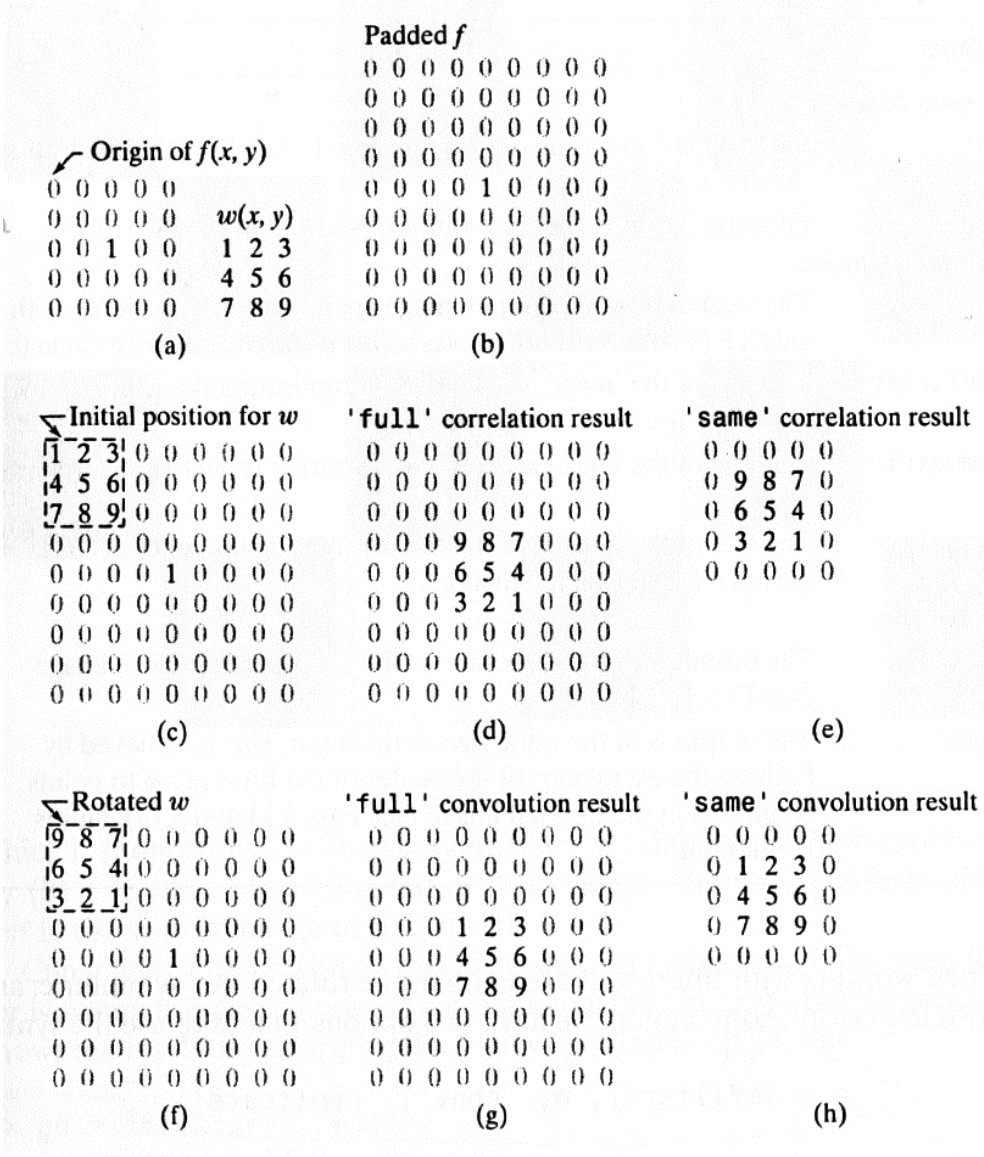
فرضیه‌های قبل را می‌توان به راحتی در تصویرها همچون عکس ۳.۱۴ تعمیم داد. خاستگاه در قسمت بالا و سمت چپ تصویر با مختصات f(x, y) قرار دارد .

برای انجام همبستگی نقطه منتها الیه سمت راست s(x,y) را طوری قرار می‌دهیم که با خاستگاه f(x, y) که در تصویر ۳.۱۴ (c)دیده می‌شود هماهنگ باشد. به لایه گذاری صفر به دلایل ذکر شده در مبحث تصویر ۳.۱۳ توجه کنید. برای ایجاد همبستگی w(x, y) را در تمام جهتها طوری حرکت می‌دهیم که حداقل یکی از عناصر تصویری آن با عناصر تصویر اصلی f(x, y) همپوشی داشته باشد. این همبستگی کامل در تصویر ۳.۱۴ د نشان داده شده است. برای به دست آوردن همبستگی یکسان که در تصویر ۳.۱۴ (c)نشان داده شده است کلیه جابجایی‌های w(x, y) باید به گونه‌ای باشد که عناصر تصویری کانونی آن با عناصر اصلی f(x, y) همپوشی داشته باشند.

برای تلفیق کردن فقط کافی است که w(x, y) ۱۸۰ درجه چرخانده شود و به همان روش همبستگی تصویرهای ۳.۱۴ f تا H عمل شود. در مثال تک بعدی قبل دیدیم که در تلفیق صرف نظر از این که کدام یک از ۲ تابع جابجا می‌شوند نتیجه یکسان تولید می‌شود. ولی در همبستگی ترتیب مهم است. این حقیقت در جعبه ابزار مهم است و فرض می‌کنیم فیلتر همیشه تابعی است که جابجا می‌شود. به حقایق مهم تصویرهای ۳.۱۴ e و H نیز توجه کنید. نتایج تلفیق و همبستگی فضایی با توجه به یکدیگر ۱۸۰ درجه چرخانده می‌شوند. انتظار همین موضوع را داشتیم زیرا تلفیق چیزی فراتر از همبستگی با فیلتر چرخشی نیست.

در جعبه ابزار با استفاده از تابع imfilter که ترکیب زیر را دارد غربال‌سازی خطی مواضع فضایی انجام می‌شود.

```
g = imfilter (f , w, filtering_mode, boundary_ option, size_ options)
```



در اینجا f تصویر ورودی است، w فیلتر است، g نتایج فیلتر شده است، و سایر پارامترها در جدول ۳.۲ خلاصه شده‌اند. در این نوع غربال‌سازی مشخص می‌شود که آیا از روش همبستگی و یا تلفیق استفاده شود. مسائل لایه‌گذاری حد فاصل آنها با اندازه فیلتر مشخص می‌شود. این گزینه‌ها در مثال ۳.۷ تشریح شده است. اندازه‌ها را می‌توان روی حالتهای یکسان یا کامل تنظیم کرد که در تصویرهای ۳.۱۳ و ۳.۱۴ تشریح شده است.

متداولترین ترکیب imfilter به شرح زیر است:

```
g = imfilter (f , w, ' replicte' )
```

این ترکیب حین اجرای غربال‌سازی خطی مواضع فضایی استاندارد IPT به کار برده می‌شود. این فیلترها که در مبحث ۳.۵.۱ شرح داده شده‌اند پیشاپیش ۱۸۰ درجه چرخانده شده‌اند تا بتوانیم از پیش فرض همبستگی در imfilter استفاده کنیم.

با توجه به تصویر ۳.۱۴ می‌دانیم که انجام همبستگی با فیلتر چرخانده شده همانند انجام تلفیق با فیلتر اصلی است. اگر فیلتر حول کانون خود متقارن باشد در این صورت، هر ۲ گزینه نتایج یکسان تولید می‌کنند.

Options	Description
Filtering Mode	
`'corr'`	Filtering is done using correlation (see Figs. 3.13 and 3.14). This is the default.
`'conv'`	Filtering is done using convolution (see Figs. 3.13 and 3.14).
Boundary Options	
`P`	The boundaries of the input image are extended by padding with a value, `P` (written without quotes). This is the default, with value 0.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'circular'`	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
`'full'`	The output is of the same size as the extended (padded) image (see Figs. 3.13 and 3.14).
`'same'`	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 3.13 and 3.14). This is the default.

جدول ۳.۲: گزینه‌های تابع imfilter

جدول ۳.۲: گزینه‌های تابع imfilter :

گزینه‌ها	تشریح گزینه‌ها
حالت فیلتر کردن Filtering mode	
Corr	در این حالت فیلتر کردن با همبستگی انجام می‌شود (تصویرهای ۳.۱۳ و ۳.۱۴) این حالت پیش فرض است.
Conv	در این حالت فیلتر کردن با تلفیق انجام می‌شود (تصویرهای ۳.۱۳ و ۳.۱۴) ا
گزینه‌های سرحدات Boundary option	
p	مرزهای تصویر ورودی با لایه‌گذاری به مقدار p (که بدون علامت نقل قول نوشته می‌شود) انجام می‌شود. حالت پیش فرض آن صفر است.
تکثیرreplicate	با کپی کردن مقادیر خارج از مرز اندازه تصویر توسعه داده می‌شود.
متقارنsymmetric	اندازه تصویر با منعکس کردن سرحدات آن توسعه داده می‌شود.
مدورcircular	با اجرای تابع ۲ بعدی روی تصویر اندازه آن توسعه داده می‌شود.
گزینه‌های اندازه Siza option	
کامل full	تصویر خروجی به همان اندازه و با همان لایه‌های تصویر توسعه داده شده است (تصویرهای ۳.۱۳ و ۳.۱۴)
یکسان same	اندازه تصویر خروجی و ورودی یکسان است. این کار با محدود کردن جابجایی کانون فیلتر به نقاط داخل تصویر اصلی انجام می‌شود (تصویرهای ۳.۱۳ و ۳.۱۴) این حالت پیش فرض است.

حین کار با فیلترهایی که پیشاپیش چرخانده نشده و متقارن نیستند و تمایل به تلفیق آنها داریم ۲ گزینه پیش روی ما است: یکی استفاده از ترکیب زیر است:

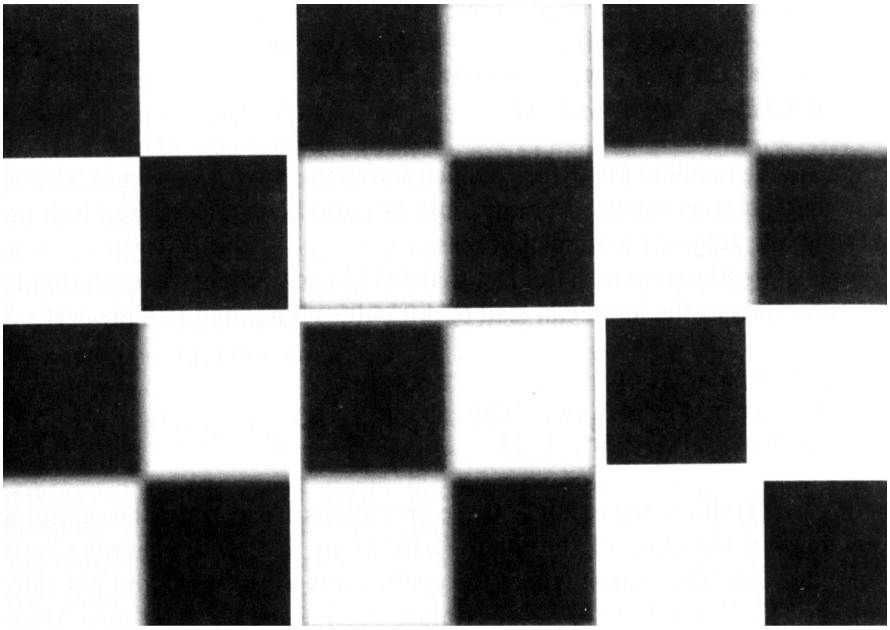
```
g = imfilter (f , w, 'conv' , 'replicate' )
```

روش دیگر پیش پرداخت w با استفاده از تابع rot90(w, 2) است تا آن را ۱۸۰ درجه بچرخانیم و بعد از imfilter (f, w, replicate) استفاده کنیم. البته این دو مرحله را می‌توان در یک مرحله ادغام کرد. در ترکیب قبل تصویر g ایجاد می‌شود که هم اندازه داده‌های ورودی است (یعنی پیش فرض محاسبات همان حالت same است که آن را قبلاً بحث کردیم).

هر عنصر تصویر فیلتر شده با دقت مضاعف و اصول ریاضی نقطه شناور محاسبه می‌شود. Imfilter تصویر خروجی را به همان نوع داده‌های ورودی تبدیل می‌کند. بنا بر این اگر f آرایه‌ای با عدد صحیح باشد در این صورت، عناصر خروجی که فراتر از حیطه این عدد صحیح هستند برش می‌خورند و مقادیر کسری گرد می‌شوند. اگر نتایج دقیقتر مورد نیاز باشد در f e باید با استفاده از im2double و یا double قبل از استفاده از imfilter تبدیل شود.

تصویر ۳.۱۵ تصویر مضاعف f با اندازه ۵۱۲ * ۵۱۲ پیکسل است. فیلتر ساده ۳۱ * ۳۱ را در نظر بگیرید.

```
>> w = ones (31) ;
```



a b c
d e f

که متناسب با فیلتر میانگین گیری است. برای نشان دادن تاثیر مقیاس گیری و استفاده از imfilter با تصویر نوع uint8 ضرایب را در انتهای این مثال تقسیم بر (۳۱) نکردیم.

تلفیق فیلتر w با یک تصویر نتایج مبهم ایجاد می‌کند. از آنجائی که این فیلتر متقارن است می‌توان از پیش فرض همبستگی در imfilter استفاده کرد. تصویر ۳.۱۵ ب نتایج عملیات فیلترسازی زیر را نشان می‌دهد.

```
>> gd = imfilter (f , w) ;
>> imshow (gd ,[  ] )
```

در این جا از گزینه‌های پیش فرض مرزی استفاده کردیم که مرز تصویر را با صفرهای (مشکی) لایه‌گذاری می‌کند. همان طور که انتظار می‌رفت لبه‌های بین قسمتهای سیاه و سفید در تصویر فیلتر شده مبهم هستند همچنین لبه‌های بین قسمتهای روشن تصویر و حاشیه آنها. علت آن است که حاشیه لایه‌گذاری شده مشکی است. می‌توان با استفاده از گزینه تکثیر این مشکل را برطرف کرد.

```
>> gr = imfilter (f , w, 'replicate' )
>> figure, imshow (gr , [ ] )
```

همان طور که در تصویر ۳.۱۵ (c)می‌بینید، حاشیه‌های تصویر فیلتر شده همان طور که انتظار می‌رفت دیده می‌شوند. در این صورت، نتایج مشابه با گزینه تقارن به دست می‌آید.

```
>> gs = imfilter (f , w, 'symmetric') ;
>> figure , imshow (gs , [ ] )
```

نتیجه در تصویر ۳.۱۵ دی دیده می‌شود که از گزینه مدور استفاده شده است.

```
>> gc = imfilter(f, w, 'circular');
>> figure, imshow(gc, [ ])
```

نتیجه تولید شده در تصویر ۳.۱۵(e) همان مسئله را با لایه‌گذاری صفر نشان می‌دهد. انتظار همین موضوع را داشتیم زیرا استفاده از حالتهای تناوبی باعث می‌شود قسمتهای سیاه و روشن تصویر مجاور یکدیگر قرار بگیرند. حالا نشان می‌دهیم که وقتی imfilter نتایجی از همان نوع ایجاد می‌کند اگر درست کنترل نشود مسئله ساز می‌شود.

```
>> f8 = im2uint8(f);
>> g8r = imfilter(f8, w, 'replicate');
figure, imshow(g8r, [ ])
```

نتیجه این عملیات در تصویر ۳.۱۵ f دیده می‌شود. در اینجا وقتی داده‌های خروجی توسط imfilter به نوع داده‌های ورودی (uint8) تبدیل شدند، برش باعث از دست دادن برخی از داده‌ها شد. علت آن است که مجموع ضرایب این نقاب (Mask) به حیطه [0,1] نرسید، و مقادیر فیلتر شده خارج از حیطه [0,255] پدید آمد. برای جلوگیری از این مسئله می‌توان ضرایب را هنجارمند کرد تا مجموع آنها در حیطه (۰و ۱) باشد. (در این مثال ضرایب را تقسیم بر (۳۱) کردیم تا مجموع ۱ شود و یا داده‌ها را به صورت فرمت مضاعف وارد کردیم. توجه داشته باشید که حتی اگر از گزینه دوم استفاده می‌شد باید اطلاعات به فرمت معتبری هنجارمند می‌شد (مثلاً برای ذخیره سازی). هر یک از این دو روش معتبر است. نکته مهم آن است که حیطه داده‌ها را باید به خاطر داشته باشید تا نتایج غیرمترقبه به وجود نیاید.

۳.۴.۲. غربال کردن غیرخطی مواضع فضایی(Nonlinear Spatial Filtering)

غربال کردن غیرخطی مواضع فضایی بر اساس عملیات مواضع همجوار است و تعریف خصوصیات مکانیکی m * n با انتقال کانون در تصویر به روشی که قبلاً بحث کردیم است. ولی در مواردی که غربال کردن غیرخطی مواضع فضایی بر اساس محاسبه مجموع محصولات باشد (که عملیات خطی است) غربال کردن غیرخطی مواضع فضایی بر اساس عملیات غیرخطی است که عناصر تصویری همجوار در آن دخیل

هستند. مثلاً واکنش هر یک از کانونها می‌تواند برابر حداکثر مقدار عناصر تصویری همجوار باشد که یک عملیات فیلترسازی غیرخطی است. تفاوت دیگر آن است که نقاب (Mask)ها در پردازش غیرخطی به همان اندازه متداول نیستند. فیلتر کردن ادامه دارد ولی فیلتر یک تابع غیرخطی است که روی عناصر تصویری همجوار عمل می‌کند و واکنش آن واکنش عملیات در کانون نقاط همجوار است.

این جعبه ابزار ۲ تابع برای انجام فیلترهای غیرخطی کلی دارد. nlfilter, colfilt فیلتر اول عملیات ۲ بعدی مستقیم انجام می‌دهد و فیلتر colfilt داده‌ها را به شکل ستون سازمان‌دهی می‌کند. گرچه colfilt به حافظه بیشتری نیاز دارد ولی عملکرد آن سریع‌تر از nlfilter است.

در اکثر عملیات پردازش تصاویر سرعت عامل مهمی است. بنا بر این برای اجرای غربال کردن غیرخطی مواضع فضایی فیلتر coltfilt به فیلتر nlfilt ارجحیت دارد.

با توجه به تصویر ورودی f به اندازه m*n و قسمتهای همجوار به اندازه m * n تابع coltfilt ماتریسی به نام A ایجاد می‌کند که حداکثر اندازه آن MN * mn است که هر ستون مطابق با عناصر تصویری است که مواضع همجوار در کانون تصویر در بر گرفته‌اند. مثلاً ستون اول مطابق با عناصر تصویری است که مواضع همجوار هنگامی که کانون در منتها الیه بالا و سمت چپ در f قرار می‌گیرد احاطه می‌شوند. فیلتر coltfilt لایه‌گذاریهای شفاف را با استفاده از لایه‌گذاری صفر) انجام می‌دهد.

```
g = colfilt(f, [m n], 'sliding', @fun, parameters)
```

در اینجا m, n ابعاد ناحیه فیلتر هستند. کلمه sliding نشان می‌دهد که این روند همراه با حرکت کشویی که در ناحیه m * n انتقال عناصر تصویری در تصویر ورودی f ایجاد می‌شود. fun@ اشاره به تابعی دارد که ما آن را تابع FUN می‌نامیم و parameter s اشاره به پارامترهایی دارد که با کاما از یکدیگر جدا شده‌اند و ممکن است توسط تابع FUN مورد نیاز باشند. نماد @ دستگیره تابع نامیده می‌شود یکی از داده‌های نرم افزار MATLAB است که حاوی اطلاعات مورد استفاده برای مرجع‌گذاری تابع است. این نکته مهمی است.

با توجه به نحوه سازمان‌دهی ماتریس A تابع FUN باید روی هر یک از ستون‌هایA انفرادی عمل کند و یک بردار ردیف تولید کند. V حاوی نتایج همه ستونها است. Kامین عنصر v نتیجه عملیات انجام شده توسط FUN روی Kامین ستون a است از آنجائی که ممکن است mn ستون در A وجود داشته باشد. حداکثر اندازه v 1 * mn است.

فیلترگذاری خطی که در بحث قبل به آن اشاره شد برای برطرف کردن مسائل حاشیه در غربال کردن غیرخطی مواضع فضایی شرایطی دارد. ولی وقتی از colfilt استفاده می‌شود تصویر ورودی باید قبل از فیلتر شدن لایه‌گذاری شود. برای این کار از تابع padarray استفاده می‌کنیم که در تابعهای ۲ بعدی ترکیب زیر را دارد:

```
fp = padarray(f, [r c], method, direction)
```

F تصویر ورودی است. F_p تصویر لایه‌گذاری شده است. [r c] تعداد ردیفها و ستونهایی است که با f لایه‌گذاری می‌شوند. روش و جهت همان است که در جدول ۳.۳ گفتیم. مثلاً اگر f = [1, 2, 3 , 4] باشد فرمان زیر

```
>> fp = padarray(f, [3 2], 'replicate', 'post')
```

نتایج زیر را تولید می‌کند.

Options	Description
Method	
'symmetric'	The size of the image is extended by mirror-reflecting it across its border.
'replicate'	The size of the image is extended by replicating the values in its outer border.
'circular'	The size of the image is extended by treating the image as one period of a 2-D periodic function.
Direction	
'pre'	Pad before the first element of each dimension.
'post'	Pad after the last element of each dimension.
'both'	Pad before the first element and after the last element of each dimension. This is the default.

جدول ۳.۳:گزینه های تابع padrrary

گزینه‌ها options	تشریح گزینه‌ها
روش method	
متقارن symmetric	اندازه تصویر با بازتابیدن آن از حاشیه کشیده می‌شود
تکثیر replicate	اندازه تصویر با تکثیر مقادیر حاشیه آن کشیده می‌شود.
مدور circular	این تصویر یک مرحله از تناوب تابع تناوبی ۲ بعدی محسوب می‌شود و از e طریق کشیده می‌شود.
جهت Direction	
پیشاپیش pre	لایه‌گذاری قبل از اولین عنصر هر بعد
پس از عبارت post	لایه‌گذاری بعد از اخرین عنصر هر بعد
هر دو both	لایه‌گذاری قبل از اولین عنصر و بعد از آخرین عنصر هر بعد. این حالت پیش فرض است.

```
Fp =
     1     2     2     2
     3     4     4     4
     3     4     4     4
     3     4     4     4
     3     4     4     4
```

اگر جهت در شناسه درج نشود، حالت پیش فرض انتخاب هر دو(both) است. اگر روش درج نشود، لایه‌گذاری پیش فرض با صفرها انجام می‌شود. اگر هیچ یک از پارامترها در شناسه درج نشود، لایه‌گذاری پیش فرض صفر است. و جهت پیش فرض هر دو است. در پایان محاسبات تصویر به اندازه اصلی درمی‌آید.

مثال ۳.۸ :برای نمایش عملکرد تابع colfilt یک فیلتر غیرخطی اجرا می‌کنیم که واکنش آن در هر نقطه میانگین هندسی مقادیر تشدید عناصر تصویری در نقاط همجوار کانون آن موضع است. میانگین هندسی در نقاط همجوار با اندازه m *n همان محصول مقادیر تشدید شده در نقاط همجوار است که به توان 1/mn رسیده است. ابتدا تابع فیلتر غیرخطی را اجرا می‌کنیم و آن را gmean می‌نامیم.

```
Function v = gmean (A)
Mn size (A, 1) ; % the length of the columns of A is always mn.
V = prod (A, 1). ∧ (1/mn) ;
```

برای کاهش تاثیرات حاشیه تصویر ورودی را با استفاده گزینه تکثیر در تابع padarray لایه‌گذاری می‌کنیم.

```
>> f = padarray (f, [m n], 'replicate' ) ;
>> g = colfilt (f, [m n] , 'sliding' , @gmean);
```

در این کنش و واکنش چند نکته مهم وجود دارد. گرچه ماتریس a بخشی از شناسه تابع gmean است، در پارامترهای colfilt درج نشده است. این ماتریس توسط colfilt با استفاده از دستگیره تابع(function handle) به صورت خودکار به gmean منتقل می‌شود. از آنجائی که ماتریس a به صورت خودکار با colfilt مدیریت می‌شود تعداد ستونهای a متغیر است. (ولی همان طور که قبلاً گفتیم تعداد ردیفها و طول ستون همیشه mn است) بنا بر این هر زمان که تابع شناسه توسط colfilt فراخوانی می‌شود a باید محاسبه شود. فرایند فیلترسازی در این حالت شامل محاسبه محصول همه عناصر تصویری همجوار و رساندن نتیجه به توان 1/mn است. نتیجه فیلتر شده در یک موضع برای هر مقدار از (x, y) در ستون مربوطه در v درج می‌شود. تابع شناسایی شده توسط دستگیره @ می‌تواند هر نوع تابعی باشد که از جایی که دستگیره تابع ایجاد شده فراخوانده شده است. نکته اصلی آن است که تابع روی ستون a عمل کند و یک بردار ردیف حاوی نتیجه کلیه ستونهای انفرادی تولید کند. سپس تابع colfilt از این نتایج استفاده کرده و آنها را مرتب می‌کند تا تصویر خروجی g را تولید کند.برخی فیلترهای غیرخطی را می‌توان بر حسب تابع‌های نرم افزار MATLAB و IPT همچون imfilter, ordfilt2 اجرا کرد (به بخش ۳.۵.۲. مراجعه کنید). تابع spfilt در بخش ۵.۳ تابع میانگین هندسی در مثال ۳.۸ بر حسبimfilter و لگاریتم نرم افزار

MATLAB و تابعهای عبارت اجرا می‌کند. وقتی این کار امکان‌پذیر باشد عملکرد معمولاً بسیار سریع‌تر است و مقداری از حافظه که مورد نیاز colfilt است مصرف می‌شود. تابع colfilt بهترین انتخاب برای عملیات فیلترسازی غیرخطی که این اجراهای تناوبی را ندارند می‌بشد.

۳.۵. جعبه ابزار پردازش تصویرها و غربال کردن غیرخطی مواضع فضایی

(Image Processing Toolbox Standard Spatial Filters)

در این بخش فیلترهای فضایی خطی و غیرخطی مورد پشتیبانی IPT را بحث می‌کنیم. سایر فیلترهای غیرخطی در بخش ۵.۳ اجرا شده است.

۳.۵.۱. فیلترهای فضایی خطی(Linear Spatial Filtering)

این جعبه ابزار یک سری فیلترهای فضایی خطی ۲ بعدی از پیش تعریف شده دارد که با استفاده از تابع fspecial که نقاب (Mask) فیلتر w را با ترکیب زیر تولید می‌ند به دست آمده است.

```
W = fspecial ('type' , parameters)
```

کلمه type نوع فیلتر را مشخص می‌کند، و کلمه parameter نوع فیلتر را تعیین می‌کند خلاصه فیلترهای مورد پشتیبانی fspecial همراه با پارامترهای کاربردی هر فیلتر در جدول ۳.۴ است.

Type	Syntax and Parameters
'average'	fspecial('average', [r c]). A rectangular averaging filter of size r × c. The default is 3 × 3. A single number instead of [r c] specifies a square filter.
'disk'	fspecial('disk', r). A circular averaging filter (within a square of size 2r + 1) with radius r. The default radius is 5.
'gaussian'	fspecial('gaussian', [r c], sig). A Gaussian lowpass filter of size r × c and standard deviation sig (positive). The defaults are 3 × 3 and 0.5. A single number instead of [r c] specifies a square filter.
'laplacian'	fspecial('laplacian', alpha). A 3 × 3 Laplacian filter whose shape is specified by alpha, a number in the range [0, 1]. The default value for alpha is 0.5.
'log'	fspecial('log', [r c], sig). Laplacian of a Gaussian (LoG) filter of size r × c and standard deviation sig (positive). The defaults are 5 × 5 and 0.5. A single number instead of [r c] specifies a square filter.
'motion'	fspecial('motion', len, theta). Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of len pixels. The direction of motion is theta, measured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.
'prewitt'	fspecial('prewitt'). Outputs a 3 × 3 Prewitt mask, wv, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: wh = wv'.
'sobel'	fspecial('sobel'). Outputs a 3 × 3 Sobel mask, sv, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: sh = sv'.
'unsharp'	fspecial('unsharp', alpha). Outputs a 3 × 3 unsharp filter. Parameter alpha controls the shape; it must be greater than or equal to 0 and less than or equal to 1.0; the default is 0.2.

جدول ۳.۴ :فیلترهای فضایی تابع fspatial

نوع Tupe	ترکیب و پارامترها Syntax &Parameters
میانگین average	میانگین fspecial یک فیلتر میانگین گیری مستطیلی به اندازه r * c پیش فرض آن ۳ * ۳ است. استفاده از یک رقم به جای rc بیانگر فیلتر مربعات است.
دیسک disk	دیسک fspecial یک فیلتر میانگین گیری مدور با مربعی به اندازه 2r + 1 با شعاع r پیش فرض شعاع ۵ است.
گوسی gussian	گوسی fspecial یک فیلتر گوسی به اندازه r * c و انحراف معیار مثبت. پیش فرضهای آن ۳ * ۳ و نیم هستند. استفاده از یک رقم به جای rc نمایانگر فیلتر مربعی است.
لاپلاسی laplacian	لاپلاسی و (a)ای fspecial فیلتر لاپلاسی ۳ * ۳ که شکل آن با (a)ا مشخص می‌شود یک رقم در حیطه ۰ تا ۱ دارد. پیش فرض (a)ا نیم است.
لگاریتمی log	لگاریتم fspecial لاپلاس گوسی فیلتری به اندازه r * c و انحراف معیار مثبت. پیش فرض آن ۰.۵ * ۰.۵ است. استفاده از یک رقم به جای rc نمایانگر فیلتر مربعی است.
حرکتی motion	Fspecial فیلتری در داده‌های خروجی قرار می‌دهد که وقتی با یک تصویر تلفیق شود حرکت خطی دوربین را با توجه به تصویر تداعی می‌کند. جهت حرکات ثتا است که بر حسب درجه و خلاف جهت عقربه‌های ساعت از افق سنجیده می‌شود. پیش فرض‌های آن ۹ و ۰ هستند که نمایانگر حرکت ۹ عنصر تصویری در جهت افق هستند.
prewitt	Fspecial یک ماسک پریویت را در داده‌های خروجی قرار می‌دهد، که شبیه به شیب عمودی است. با جابجایی این نتیجه ماسک شیب افقی به دست می‌آید.
sobel	Fspecial یک ماسک ۳ * ۳ سوبل را در داده‌های خروجی قرار می‌دهد که یک شیب تقریباً عمودی ایجاد می‌کند. ماسک شیب افقی با جابجایی نتیجه به دست می‌آید.
Unsharp	Fspecial یک فیلتر مبهم ۳ * ۳ در داده‌های خروجی قرار می‌دهد. پارامتر (a)ا شکل را کنترل می‌کند. باید بیشتر یا معادل صفر و کمتر یا معادل ۱ باشد پیش فرض آن ۰.۲ است.

مثال ۳.۹ : با بهبود و تقویت یک تصویر با فیلتر لاپلاس کاربردهای fspecial , imfilter نشان داده شده است. لاپلاس یک تصویر f(x,y) به شرح زیر تعریف می‌شود:

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

مقادیر تقریبی دیجیتالی دومین مشتق به شرح زیر است:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

این عبارت را می‌توان در کلیه نقاط با مختصات (x, y) در یک تصویر با تلفیق تصویر با ماسک فضایی زیر اجرا کرد.

```
0     1     0
1     -4    1
0     1     0
```

در تعریف دیگر مشتقهای دوم دیجیتالی عناصر مورب نیز در نظر گرفته می‌شوند و با ماسک زیر قابل اجرا هستند:

```
1     1     1
1     -8    1
1     1     1
```

این دو مشتق گاهی با علائمی بر خلاف علائمی که در اینجا نشان داده شده است تعریف می‌شوند و این امر منجر به تشکیل نقاب (Mask)هایی می‌شود که نگاتیو دو ماسک قبلی هستند.

بهبود و تقویت بر اساس لاپلاس مبتنی بر معادله زیر است:

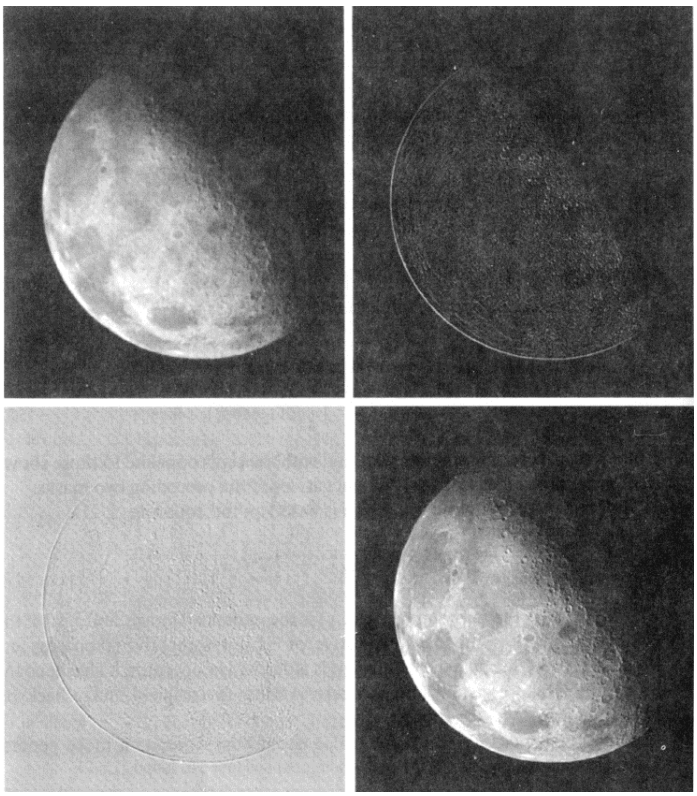
$$g(x,y) = f(x,y) + c[\nabla^2 f(x,y)]$$

در اینجا f(x, y) تصویر ورودی است g(x,y) تصویر بهینه‌سازی شده است. اگر ضرایب کانون ماسک مثبت باشند c=1 است و اگر منفی باشند ۱ – است. از آنجائی که لاپلاس یک عملگر اشتقاقی است تصویر را شفاف می‌کند ولی نقاط ثابت را به سمت صفر میل می‌دهد. افزودن تصویر اصلی باعث می‌شود سطح خاکستری به حالت اصلی خود بازگردد.

تابع fspecial (لاپلاسی و الفایی) یک ماسک لاپلاسی کلی‌تر را اجرا می‌کند.

$$\begin{array}{ccc} \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \\ \frac{1-\alpha}{1+\alpha} & \frac{-4}{1+\alpha} & \frac{1-\alpha}{1+\alpha} \\ \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \end{array}$$

که با آن می‌توان نتایج بهینه‌سازی را دقیقتر تنظیم کرد. ولی استفاده از لاپلاس بر اساس دو ماسکی است که قبلاً بحث کردیم. حالا تصویر ۳.۱۶ (a) را با استفاده از لاپلاس بهینه‌سازی می‌کنیم. این تصویری مبهم از قطب شمال ماه است. برای بهبود و تقویت تصویر آن را واضح می‌کنیم و در عین حال سعی می‌کنیم رنگهای مقیاس خاکستری آن حفظ شود. ابتدا فیلتر لاپلاس را تولید و نشان می‌دهیم.



a b
c d

تصویر ۳.۱۶ تصویر قطب شمال کره ماه (b) تصویر فیلتر شده لاپلاسی با فرمت uint8 (c)تصویر فیلتر شده لاپلاسی با فرمت دوگانه (d) بهبود نتایج با تفریق (چاپ تصاویر با کسب اجازه از ناسا)

```
>> w = fspecial ('laplacian', 0)
W =
      0.0000      1.0000      0.0000
      1.0000      -4.0000     1.0000
      0.0000      1.0000      0.0000
```

توجه داشته باشید که این فیلتر مضاعف است و شکل الفا = ۰ فیلتر لاپلاسی است که قبلاً بحث کردیم. این شکل را می‌توان به صورت دستی به طریق زیر مشخص کرد»

```
>> w = [0 1 0 ; 1  -4 ; 0 1 0] ;
```

سپس w را در تصویر ورودی f قرار می‌دهیم که از نوع uint8 است.

```
>> g1 = imfilter (f, w, ' replicate' ) ;
>> imshow (g1 , [  ] )
```

تصویر حاصله در عکس ۳.۱۶ (b)نشان داده شده است. این نتیجه منطقی به نظر می‌رسد. ولی مسئله‌اش آن است که همه عناصر تصویری مثبت هستند. از آنجائی که ضریب فیلتر کانونی منفی است می‌دانیم که می‌توان انتظار یک تصویر لاپلاسی با مقادیر منفی را داشت. ولی f در این مورد از نوع uint9 است و همان طور که در بخش قبل گفتیم، فیلترسازی آن با imfilter باعث می‌شود داده‌های خروجی از همان نوع تصویر ورودی باشند بنا بر این مقادیر منفی برش داده می‌شوند. برای رفع این مسئله قبل از فیلتر کردن f آن را به کلاس مضاعف تبدیل می‌کنیم.

```
>> f2 = im2double (f) ;
>> g2 = imfilter (f2, w, 'replicate' ) ;
>>. imshow (g2, [ ] )
```

نتیجه نشان داده شده در تصویر ۳.۱۶ (c)نشان می‌دهد تصویری که به طور صحیح با لاپلاس پردازش شده باشد چگونه باید به نظر برسد. سپس برای بازگرداندن هاله‌های خاکستری که در لاپلاس از بین رفته است تصویر لاپلاس را از تصویر اصلی کم می‌کنیم ((چون که ضریب کانونی منفی است)

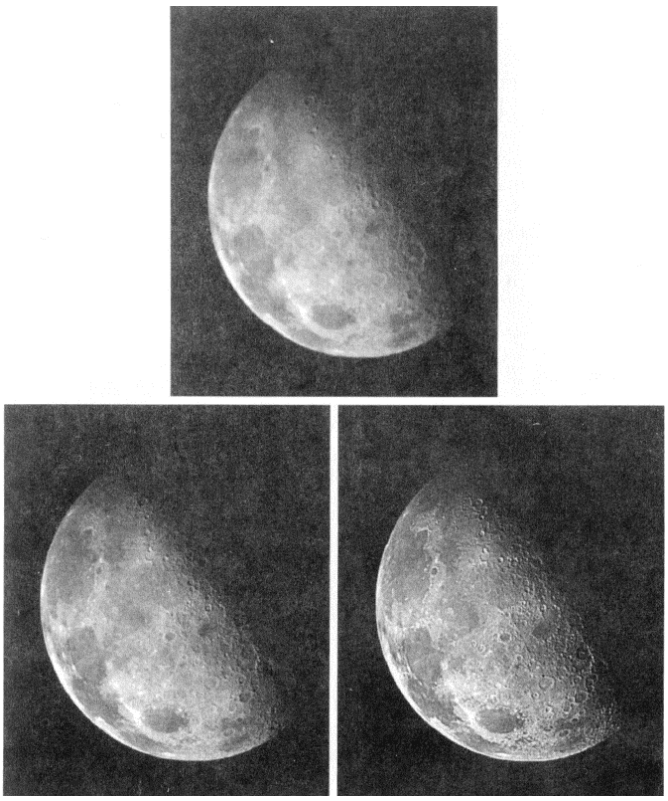
```
>> g = f2 – g2 ;
>> imshow (g)
```

نتیجه در تصویر ۳.۱۶ د نشان داده شده است که واضح‌تر از تصویر اصلی است.

مثال ۳.۱۰ :

برای بهبود و تقویت تصاویر گاهی باید خصوصیات فیلتر فراتر از خصوصیات جعبه ابزار مشخص شوند. لاپلاس مثال خوبی است. این جعبه ابزار یک فیلتر لاپلاسی ۳ * ۳ دارد که کانون آن ۴- است برای بهبود وضوح می‌توان از فیلتر لاپلاسی ۳ * ۳ استفاده کرد که ۸- در کانون دارد و همان طور که قبلاً گفتیم توسط واحدهای ۱ احاطه شده است. هدف از این مثال اجرای دستی فیلتر است. همچنین مقایسه نتایج به دست آمده با استفاده از ۲ فرمول لاپلاس. ترتیب فرمانها به شرح زیر است:

```
>> f = imread ('moon. Tif' ) ;
>> w4 = fspecial ('laplacian' , 0) ; % same as w in Example 3.9.
>> w8 = [ 1 1 1 ; 11 -8 1 ; 1 1 1 ] ;
>> f = im2 double (f) ;
>> g 4= f – imfilter (f, w4, 'replicate' ) ;
>> g 8 = f- imfilter (f, w8, 'replicate' ) ;
>> imshow (f)
>> figure, imshow (g4)
>> figure, imshow (g8)
```



a

b c

تصویر ۳.۱۷: ((a) تصویر اصلی کره ماه را برای مقایسه کردن نشان می‌دهد. ((b) g4 است که همان تصویر ۳.۱۶ (d)و تصویر ۳.۱۷ (c)است((c)g8 همان طور که انتظار می‌رفت این نتیجه بسیار واضحتر از تصویر ۳.۱۷ (b)است.

۳.۵.۲. فیلترهای فضایی غیرخطی(Nonlinear Spatial Filtering)

تابع ordfilt2 ابزاری متداول برای ایجاد فیلترهای فضایی غیرخطی در IPT است.

که فیلترهایی به ترتیب آماری(Order-statistic filters) ایجاد می‌کند (که گاهی وقتها آنها را فیلترهای رده بندی شده نیز می‌نامیم) اینها فیلترهای فضایی غیرخطی هستند که واکنش آنها بر اساس ترتیب یا رده بندی عناصر تصویری موجود در مواضع همجوار تصویر است. سپس مقدار عنصر تصویری کانونی مواضع همجوار با مقداری که در نتیجه رده بندی تعیین می‌شود جایگزین می‌گردد.

در این بخش روی فیلترهای غیرخطی که ordfilt2 تولید می‌کند متمرکز می‌شویم. فیلترهای خطی دیگری نیز ابداع می‌شوند و در بخش ۵.۳ اجرا شده‌اند.

ترکیب تابع oprdfilt2 به شرح زیر است:

```
g = ordfilt 2 (f , order, domain)
```

این تابع با جایگزین کردن هر عنصر f با چندمین عنصر هماهنگ در مواضع همجوار که توسط عناصر غیرصفر دامنه مشخص می‌شود تصویر خروجی g را می‌سازد. در اینجا دامنه ماتریس m * n از یکها و صفرهایی است که مواضع عناصر تصویری همجوار را مشخص می‌کنند و در محاسبات به کار برده می‌شوند. در این حالت دامنه همچون نقاب (Mask) عمل می‌کند. عناصر تصویری همجوار که مطابق با صفر در ماتریس دامنه باشند، در محاسبات به کار برده نمی‌شوند. مثلاً برای اجرای min filter (درجه ۱) با اندازه m * n از ترکیب زیر استفاده می‌کنیم:

```
g = ordfilt 2 (f , 1 , ones (m , n))
```

در این فرمول ۱ نمایانگر اولین نمونه در بین نمونه‌های mn است و ones (m,n) یک ماتریس m * n ایجاد می‌کند که مشخص می‌کند که کلیه نمونه‌های مواضع مجاور در محاسبات به کار برده می‌شوند.

در واژگان فنی رشته آمار فیلتر حداقل (که اولین نمونه یک مجموعه مرتب است) به عنوان صفر درصد مشخص می‌شود. به همین ترتیب صدمین درصد آخرین نمونه این مجموعه است که mnامین نمونه است. این مطابق با max filter است که با استفاده از ترکیب زیر اجرا می‌شود:

```
g = ordfilt 2 (f , m*n, ones (m, n))
```

بهترین فیلتر شناخته شده مرتب آماری در پردازش تصاویر دیجیتالی فیلتر میانگین گیری(Median filter) است که مطابق با پنجاهمین درصد است. برای ایجاد این فیلتر متوسط می‌توان از تابع ordfilt2 در نرم افزار MATLAB استفاده کرد. در اینجا median (1mn) میانه این رشته مرتب را محاسبه می‌کند.

```
g = ordfilt 2 (f, median (1: m*n), ones (m, n))
```

تابع میانه نیز ترکیب زیر را دارد:

```
v = median (A, dim)
```

در اینجا v برداری است که عناصر آن میانه A در بعد dim هستند. مثلاً اگر dim = 1 باشد هر عنصر v میانه عناصر مطابق با ستون A است.

با توجه به اهمیت کاربردی آن، این فیلتر میانه ۲ بعدی در جعبه ابزار به شکل خاصی اجرا می‌شود.

```
g = medfilt 2 ( f , [m n] , padopt)
```

در اینجا tuple [m n] مواضع همجوار را به اندازه m * n طوری تعریف می‌کند که میانه محاسبه شود. و padopt یکی از سه گزینه لایه‌گذاری مرزی است. حالت صفر پیش فرض است و در حالت متقارن با انعکاس یافتن از حاشیه به طور متقارن منعکس می‌شود. شاخص بندی طوری است که اگر f از کلاس مضاعف باشد با یکها لایه‌گذاری می‌شود و اگر نباشد با صفرها لایه‌گذاری می‌شود. شکل پیش فرض این تابع به شرح زیر است:

```
g = medfilt 2 (f)
```

که برای محاسبه میانه از مواضع همجوار ۳ * ۳ استفاده می‌کند و مرز داده‌های ورودی را با صفرها لایه‌بندی می‌کند.

مثال ۳.۱۱ :

فیلترسازی میانه ابزاری مفید برای کاهش ناهماهنگی‌های تصویر است. گرچه در خصوص کاهش ناهماهنگی به طور مفصل در فصل ۵ بحث کردیم، ولی بهتر است اجرای فیلتر میانه را به طور خلاصه شرح دهیم.

تصویر ۳.۱۸ (a) تصویر اشعه ایکس f از یک مدار صنعتی است که حین بازرسی خودکار تجهیزات گرفته شده است. .

تصویر ۳.۱۸. (b)همان تصویر است که با اختلالهای نقطه‌ای تغییر داده شده است که احتمال ایجاد شدن نقاط سیاه و سفید ۰.۲ است. این تصویر با استفاده از تابع imnoise که به طور مفصل در بخش ۵.۲.۱ تشریح شده است ایجاد گردیده است.

```
>> fn = imnoise (f , 'salt & pepper ' , 0.2) ;
```

تصویر ۳.۱۸.(c)نتیجه فیلترسازی میانه این تصویر اختلال‌دار است که از عبارت زیر در آن استفاده شده است.

```
>> gm = medfilt 2 (fn) ;
```

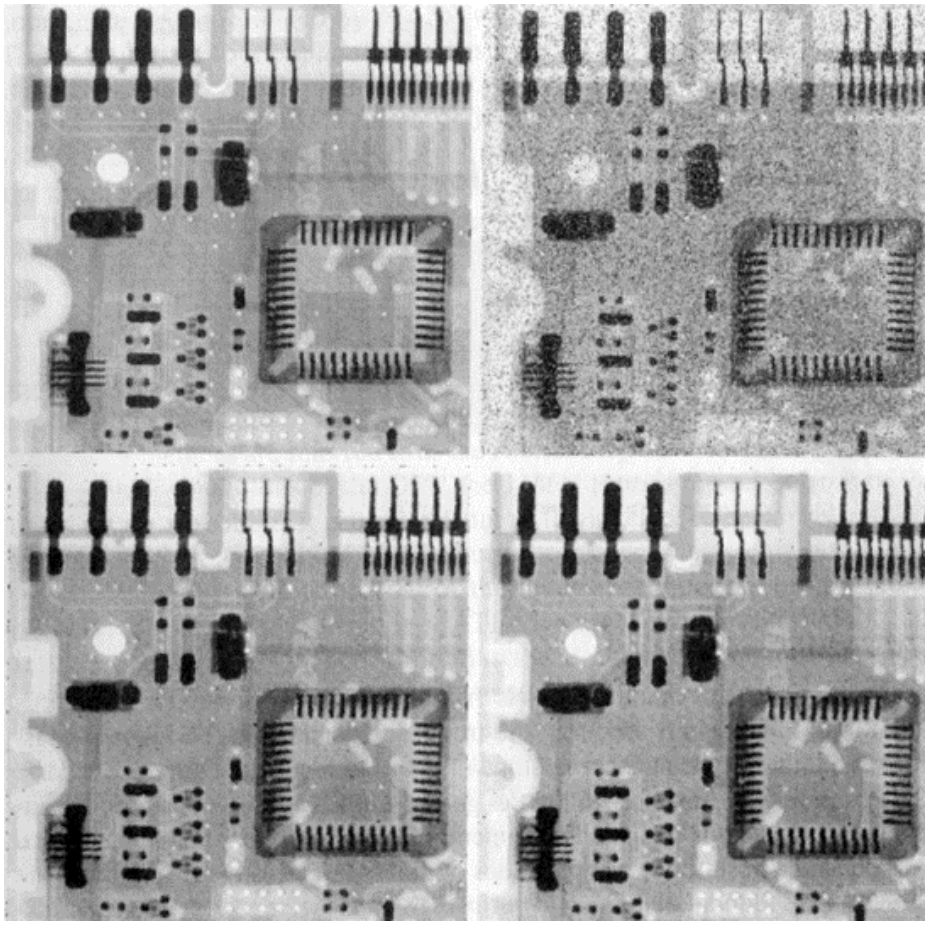
با توجه به سطح اختلال در تصویر ۳.۱۸. (b)فیلترسازی میانه با استفاده از تنظیمات پیش فرض به خوبی باعث کاهش اختلالها شده است. به خالهای مشکی دور حاشیه توجه کنید.

این خالها توسط نقاط مشکی دور تصویر ایجاد شده است (توجه داشته باشید که در حالت پیش فرض حاشیه با صفرها لایه‌گذاریمی‌شود) این تصویر را می‌توان با استفاده از گزینه "متقارن" symmetric کاهش داد.

```
>> gms = medfilt2 (fn , 'symmetric') ;
```

نتیجه نشان داده شده در تصویر ۳.۱۸.(d)نزدیک به نتیجه تصویر ۳.۱۸.(c)اس. تنها فرقش آن است که تاثیر حاشیه مشکی مشخص نیست.

a b
c d



تصویر ۳.۱۸

خلاصه مطالب(Summary)

مطالب این فصل هم مربوط بهبود و تقویت تصاویر است و هم شالوده موضوع‌های مختلف فصلهای آتی است. مثلاً در فصل ۵ مجدداً با مبحث پردازش تصاویر برای برگرداندن تصاویر به حالت اولیه و اصلی خود مواجه می‌شویم و به کاهش اختلال و تابعهای اختلال زا در نرم افزار MATLAB نیز می‌پردازیم. برخی از نقاب (Mask)های فضایی که به طور خلاصه در اینجا ذکر شده است به صورت گسترده در فصل ۱۰ برای کشف لبه‌ها در برنامه‌های قطعه‌سازی به کار برده شده است. فرضیه‌های تلفیق و همبستگی از دیدگاه دامنه فرکانس دوباره در فصل ۴تشریح شده است. پردازش نقاب (Mask)ها و اجرای فیلترهای فضایی در مبحثهای گوناگون این تحقیق مطرح شده است. این بحث گسترده می‌شود و نشان می‌دهیم چگونه می‌توان فیلترهای فضایی کارآمد را در نرم افزار MATLAB اجرا کرد.