باسمه تعالی

سیستم های چندرسانه‌ای (۳۴۲-۴۰)
دانشکده مهندسی کامپیوتر
ترم پائیز ۱۳۸۵
دکتر حمیدرضا ربیعی

تکلیف شماره ۳: ابزار اصلی پردازش تصویر

۱- مقدمه

تصویری که با یک اسکنر یا دوربین دیجیتال گرفته شده، ممکن است کانتراست مطلوب یا فوکوس مناسب را نداشته باشد. یا ممکن است تصویر با نویز مخلوط شده باشد و نیاز به تمیز شدن داشته باشد. در بعضی موارد دیگر، ممکن است نیاز به بزرگ کردن تصویر برای دید راحت‌تر وجود داشته باشد. انواع ابزار پردازش تصویر برای افزایش کیفیت یا ایجاد بعضی جلوه ها ایجاد شده است. در این آزمایش، شما بعضی عملکردهای پایه ای پردازش تصویر را آزمایش کرده و یاد می گیرید.

۲- بهبود تصویر

۲-۱- بهبود کانتراست

بهبود کانتراست به تغییر سطوح تصویر یا متعادل کردن رنگ آن گفته می شود. به عنوان مثال، اگر یک تصویر زیر نور بسیار کم قرار بگیرد، ممکن است تصویر بسیار تیره به نظر برسد. با ابزار بهبود کانتراست مناسب شما می توانید این تصویر را دارای محدوده دینامیک بزرگتری بکنید و محدوده کلی را از سفید تا سیاه توسعه بدهید.

کانتراست تصویر می تواند به صورت ریاضی بوسیله هیستوگرام تصویر اندازه گیری شود. هیستوگرام تعداد پیکسل های دارای هر سطح رنگ به خصوص را می شمارد. اگر هیستوگرام را با تعداد کل پیکسل ها نرمالیزه کنیم، آنگاه حاصل، تقریبی از تابع جرم احتمال سطوح خواهد بود، یک تصویر سیاه سفید دارای یک هیستوگرام است در حالی که یک تصویر رنگی می تواند هیستوگرام های مجزا برای مولفه های مختلف رنگ داشته باشد، یک تصویر با کانتراست خوب یک هیستوگرام صاف خواهد داشت که نشان می دهد تمام رنگهای مختلف یا سطوح خاکستری تقریباً با تعداد مساوی پیکسل ها در تصویر حضور دارند. از سوی دیگر، یک تصویر با کانتراست پایین، پخش هیستوگرام بسیار باریکی خواهد داشت زیرا سطوح خاکستری/ رنگی آن در یک محدوده کوچک متمرکز است. به عنوان مثال، یک تصویر بسیار سیاه، هیستوگرام بسیار باریک حول سطوح پایین خواهد داشت.

هدف بهبود کانتراست، پخش دوباره سطوح تصویر به گونه ای است که هیستوگرام تصویر بهبود یافته به خوبی گسترده شده باشد. به طور کلی، بهبود کانتراست، با یک تابع تناظر پیکسل به پیکسل که مقادیر پیکسل ها را اصلاح می کند، صورت می پذیرد. تناظر فقط به سطح خاکستری پیکسل وابسته است مستقل از اینکه دیگر پیکسل ها چه باشند. برای حفظ پیوستگی و ترتیب سطوح اصلی، تناظر باید پیوسته و غیر نزولی باشد.

۲-۱-۱- مساوی سازی هیستوگرام

مساوی سازی هیستوگرام به عملی گفته می شود که تصویر دارای هیستوگرام دلخواه را به تصویری با هیستوگرام صاف تبدیل می‌کند. روش هیستوگرام بر مبنای نتایج شناخته شده تئوری احتمالات استوار است.

قضیه: تابع توزیع جمعی به صورت زیر از روی تابع توزیع احتمال به دست می آید:

$$c(f) = \int_{f_{min}}^{f} p(f)df$$

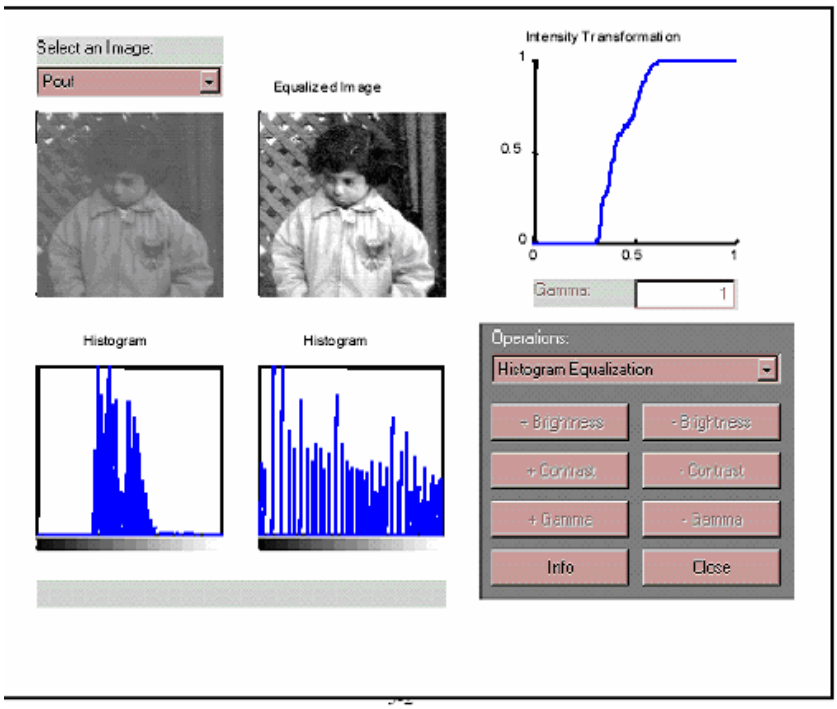
حال اگر ما تبدیل تابع توزیع جمعی را انجام بدهیم، تابع حاصل دارای توزیع یکنواخت بین ۰ و ۱ خواهد بود.

اگرچه نتیجه بالا برای متغیرهای تصادفی پیوسته مطلقاً درست است، می تواند برای متغیرهای گسسته نیز به کار برود. در این صورت، pdf با pmf جایگزین می شود (یا هیستوگرام نرمالیزه شده) و cdf با جمع مقادیر pmf حاصل می شود.

برای انجام مساوی سازی هیستوگرام برای یک تصویر داده شده، ابتدا هیستوگرام آن را محاسبه می کنیم. سپس برای هر مقدار موجود در تصویر اصلی، مقادیر نرمالیزه شده هیستوگرام را از سطح صفر تا آن مقدار جمع می کنیم. جواب به دست آمده، همان مقدار جدید پیکسل است. وقتی قاعده تبدیل برای تمام سطوح ممکن محاسبه شد، نتیجه می تواند به در یک جدول ذخیره شود.

شکل ۱ مثالی از مساوی سازی هیستوگرام نشان می دهد که با برنامه دموی matlab، imadjdemo.m به دست آمده است. می توانید ببنید که تصویر اصلی هیستوگرام باریکی دارد و تصویر مساوی سازی شده تقریباً هیستوگرام صافی بر روی کل مقادیر دارد.

یک هیستوگرام صاف ممکن است به یک تصویر خوب منجر نشود. همان برنامه قبلی به شما اجازه می دهد که تابع تناظر را به صورت دستی تنظیم کنید (به جای اینکه هیستوگرام خروجی همیشه صاف باشد) تا وقتی که شما نتیجه دلخواه را به دست آورید. ابزار مناسبی هم به همین منظور در فوتوشاپ نیز وجود دارد.



شکل ۱: مثالی از مساوی سازی هیستوگرام

۲-۲- حذف نویز

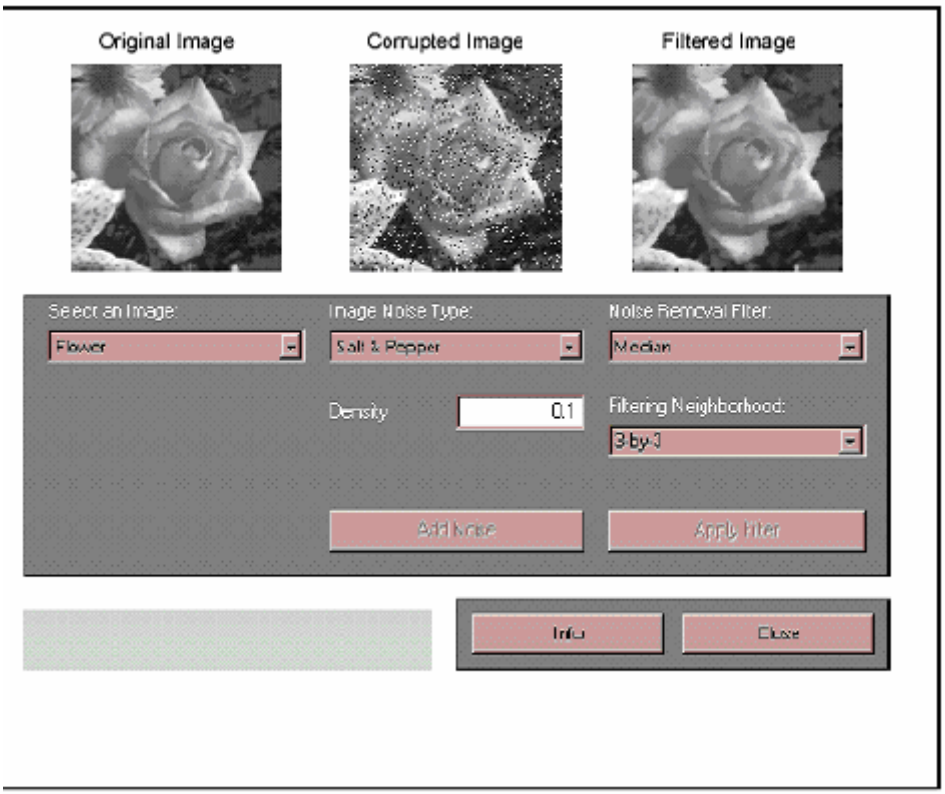
حذف نویز به حذف ضربه های ناخواسته در تصویر گفته می شود. این کار معمولاً با فیلتر پایین گذر قابل انجام است که معمولاً مقدار هر پیکسل را با مقادیر میانگین یا مد پیکسل های اطراف آن جایگزین می کند. فیلتر استفاده کننده از مد برای نویزهای ضربه ای مناسب تر است زیرا مقدار به دست آمده به پیکسل مخلوط شده با نویز بستگی ندارد. فیلتر متوسط گیر برای نویزهای متغیر پیوسته مناسب است. به عنوان مثال، یک فیلتر میانگین ۳×۳ در همسایگی متناظر با فیلتر زیر است:

$$H = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

برای کارایی بهتر ممکن است فیلتر زیر استفاده شود:

$$H = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

شکل ۲، نتیجه حذف نویز نوع نمک و فلفل با فیلتر میانه ۳×۳ را نشان می دهد. نتیجه با برنامه دموی MATLAB (nrfiltdemo.m) به دست آمده است. با همان برنامه شما می توانید فیلترهای مختلف با سایزهای گوناگون و برای انواع مختلف نویز را امتحان کنید. همچنین می توانید از تابع conv۲ برای به کار بردن هر فیلتر خاصی استفاده کنید.



شکل۲: حذف نویز نوع نمک و فلفل با فیلتر میانه ۳×۳

مساله فیلترهای میانین گیر یا مدگیر این است که ممکن است تصویر را در لبه ها تار کند. زمینه گسترش تکنیک های حذف نویز، شامل حذف موثر نویز بدون تار کردن لبه ها می باشد. به این گونه فیلترها معمولاً فیلترهای هموار کننده حفظ کننده لبه گفته می شود.

۳-۲- تمیز کردن لبه ها

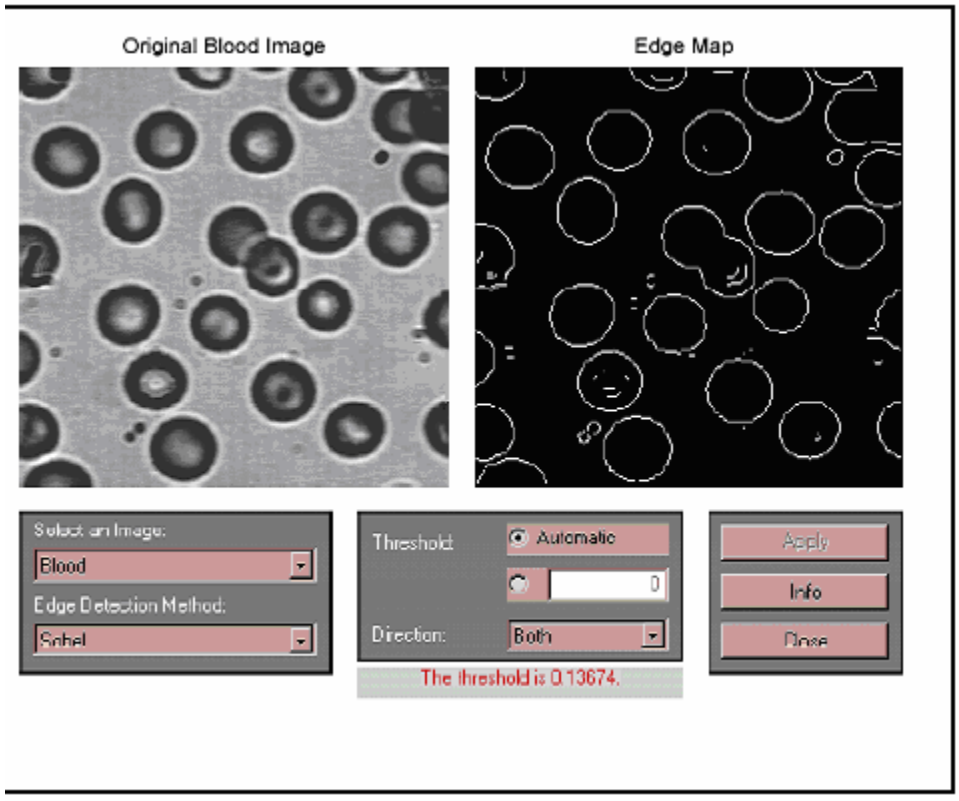
لبه ها فرکانس های بالای تصویر هستند در حالی که نواحی صاف نواحی فرکانس پایین تصویر هستند. برای بهبود لبه ها، یک راهبرد، استفاده از فیلترهای تأکید کننده به فرکانس های بالا است که مولفه های فرکانس پایین را نگه دارد ولی مولفه های فرکانس بالا را تقویت کند.

٤-٢- آشکارسازی لبه ها

لبه های یک تصویر ساختار اشیاء آن را نشان می دهد. آشکارسازی لبه یک مرحله پیش پردازش مهم برای آشکارسازی شیء و شناسایی می باشد. به خاطر اینکه لبه ها ضرورتاً مولفه های فرکانس بالای یک تصویر هستند، تشخیص لبه معمولاً با به کارگیری فیلترهای فرکانس بالا انجام می شود که مولفه های فرکانس پایین را جذب می کند و سپس یک آستانه به کار می برد. یک آشکارساز لبه پرطرفدار، عملگر sobel است که از دو فیلتر زیر تشکیل شده است:

$$H_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & 1 \end{bmatrix}, \quad H_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

فیلترهای بالا دو فیلتر بالاگذر در جهت های افقی و عمودی هستند. برای آشکارسازی یک لبه این دو فیلتر را بر روی یک تصویر به کار می بریم که عددی نمایانگر لبه افقی و عمودی برای هر پیکسل می دهد. اگر تنها علاقمند به آشکارسازی لبه های افقی هستید عدد نمایانگر لبه افقی را با سطح آستانه مقایسه میکنید و آن پیکسل را به عنوان لبه در نظر می گیرید. اگر به لبه ها در تمام جهات علاقمندید اندازه گرادیان را با $g_m = \sqrt{g_x^2 + g_x^2}$ حساب کرده و یک پیکسل را لبه محسوب می کنید اگر $g_m > T$. شکل ۳ یک مثال از آشکارسازی لبه به دست آمده با عملگر sobel را نشان می دهد. با استفاده از برنامهٔ دموی edgedemo.m می توانید تأثیر آشکارسازهای لبه مختلف را بررسی کنید.



شکل۳: تشخیص لبه با استفاده از عملگر Sobel

۳- تغییر اندازه تصویر

در پردازش تصویر موارد زیادی پیش می آید که لازم است نرخ نمونه برداری تصویر تغییر کند. به عنوان مثال ممکن است تصویر برای بهبود نمایش، کوچک شود. یا ممکن است تصویر نیاز به چرخیدن در جهاتی به غیر مضارب ۹۰ درجه داشته باشد. کاربردهای حسگر از راه دور نیاز به تغییر نرخ تصویر برای حذف اعوجاج های جوی یا آرتیفکت‌های ناشی از چرخش زمین دارند.

تغییر نرخ، همچنین، وسیله ای برای به دست آوردن ثبت زیر پیکسلی را از تصاویر حسگرهای مختلف با دنباله تصاویر زمانی، فراهم می کند.

۱-۳- اصول تغییر نرخ تصویر

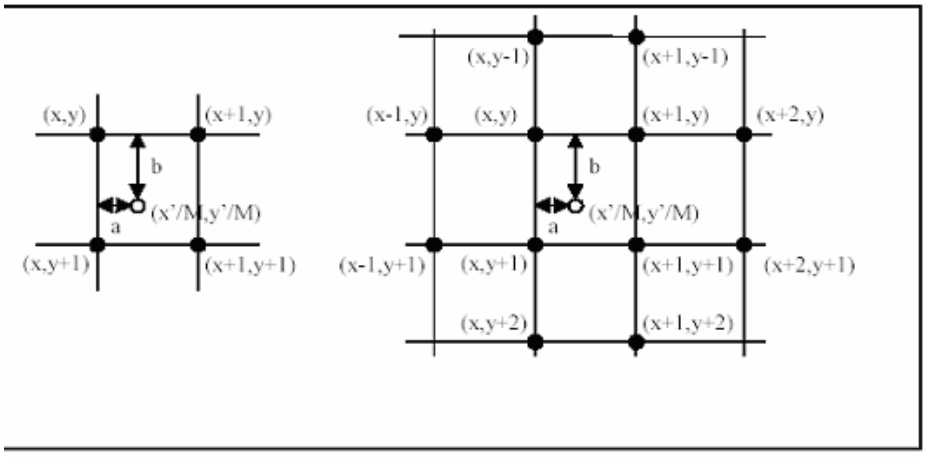
در آزمایش ۲، اصول تغییر نرخ نمونه برداری یک سیگنال صحبت یا صدا که یک بعدی بود را یاد گرفتید. تغییر نرخ تصویر، یک مسأله تبدیل نرخ نمونه برداری در دو بعد است. بزرگ کردن تصویر متناظر با up-sampling و کوچک کردن تصویر متناظر با down-sampling است. به یاد داشته باشید که در یک بعد، up-sampling با قرار دادن صفر بین نمونه های معلوم و به کارگیری فیلتر درونیابی برای تخمین نقاط نامعلوم انجام می شد. برای down-sampling به منظور اجتناب از الیاسینگ باید پیش فیلتر بر روی تصویر اصلی به کار برده شود تا فرکانس های بالاتر از نصف نرخ نمونه برداری جدید را حذف کند. این اصول برای حالت دوبعدی هم کاربرد دارد. می توان مستقیماً یک فیلتر ۲ بعدی به کار برد یا می توان ابتدا یک فیلتر یک بعدی بر روی سطرها و سپس بر روی ستون ها به کار برد. مورد اخیر، پردازش مجزا نامیده می شود که از نظر محاسباتی از پردازش ۲ بعدی مستقیم کاراتر است.
همان طور که در آزمایش ۲ یاد گرفتید، پیش فیلتر و فیلترهای درونیابی باید فیلترهای پایین گذر ایده آل باشند. متأسفانه این فیلترها قابل تحقق نیستند زیرا نمونه های نامحدودی را در بر می‌گیرند. در عمل فیلترهای مورد استفاده برای تغییر نرخ تصویر بسیار ساده تر از راه حل های تئوری هستند. در ادامه تعدادی روش درونیابی محبوب را توضیح می دهیم.
به طور کلی، یک روش درونیابی مقادیر جدیدی را بر اساس مجموعه ای از پیکسل های ورودی تولید می کند. تعداد پیکسل های ورودی و سهم آنها در مقدار پیکسل جدید فیلتر درونیابی را معین می کند. یک عامل در تعیین فیلتر درونیابی مناسب، مقدار زمان موردنیاز برای تولید خروجی است. هر چه تابع درونیابی ساده تر، مقدار زمان موردنیاز برای تولید خروجی کم تر است. هر چه تابع درونیابی به تابع درونیابی ایده آل پیچیده تر می شود به زمان پردازش بیشتری نیاز دارد. برای بیشتر کاربردها درونیابی با اسپلاین مکعبی کفایت می کند.

۲-۳- درونیابی نزدیکترین همسایه (یا تکرار پیکسل)

در این روش، پیکسل درونیابی شده جدید مقدار نزدیکترین پیکسل از تصویر ورودی را می‌گیردو به صورت زیر محاسبه می شود که M فاکتور زومینگ یا درونیابی است (شکل ٤):

$$O[x', y'] = I[(\text{int})(x + 0.5), (\text{int})(y + 0.5)], \qquad x = \frac{x'}{M}, y = \frac{y'}{M}$$

این روش خیلی ساده است. اگر M بزرگ باشد تصویر زوم شده دارای اثر بلوکینگ شدید است.



شکل ٤: عمل درونیابی

۳-۳- درونیابی خطی

با درونیابی خطی، مقدار درونیابی شده جدید، جمع وزن دار چهار همسایگی کناری پیکسل است. وزن ها به طور معکوس با فاصله از مکان هر پیکسل متناسب هستند. درونیابی خطی، تصویر بسیار هموارتری از درونیابی نزدیکترین همسایگی تولید می کند. اگر چه نیاز به زمان پردازش بیشتری نسبت به روش قبلی دارد، کیفیت نهایی تصویر خروجی به شدت بهتر می شود. از نظر ریاضی، الگوریتم درونیابی خطی با معادله زیر نشان داده می شود (شکل b-٤ را ببینید).

در معادله بالا ٤ ضرب برای هر نمونه نیاز است. برای کاهش محاسبات، می توان همان محاسبات را با عملگرهای جداپذیر انجام داد. در ابتدا، برای هر ردیف در تصویر اصلی به صورت افقی درونیابی با فاکتور M انجام می‌شود:

$$F[x', y'] = (1-a)I[x, y] + aI[x+1, y].$$

سپس برای هر ردیف در تصویرحاصل به صورت عمودی درونیابی با فاکتور M انجام می‌شود:

$$o[x', y'] = (1-b)F[x', y] + bF[x', y+1].$$

شما می توانید مقایسه کنید که چند عمل برای درونیابی یک تصویر با روش مستقیم یا جداپذیر نیاز است.

٤-۳- درونیابی اسپلاین مکعبی

با این روش، مقدار درونیابی شده از یک پنجره ٤×٤ در تصویر ورودی به بدست می آید. نتیجتاً، تصویر خروجی پیوسته تر و هموارتر است. خوشبختانه درونیابی مکعبی یک تابع جداپذیر است بنابراین یک تابع یک بعدی می تواند پشت سر هم در دو جهت عمودی و افقی به کار برود. فرمول زیر فیلتر درونیابی مکعبی دوبعدی را شرح می دهد. توجه کنید که این کار در دو مرحله انجام می شود، در ابتدا جهت x و سپس جهت y .

$$F[x', m] = -a(1-a)^2 I[x-1, m] + (1-2a^2+a^3)I[x, m] + a(1+a-a^2)I[x+1, m]$$
$$-a^2(1-a)I[x+2, m], \qquad x = (\text{int})\frac{x'}{M}, a = \frac{x'}{M} - x$$

$$o[x', y'] = -b(1-b)^2 F[x', y-1] + (1-2b^2+b^3)F[x', y] + b(1+b-b^2)F[x', y+1]$$
$$-b^2(1-b)F[x', y+2], \qquad y = (\text{int})\frac{y'}{M}, b = \frac{y'}{M} - y$$

٥-۳- مقایسه روش های مختلف درونیابی با استفاده از MATLAB

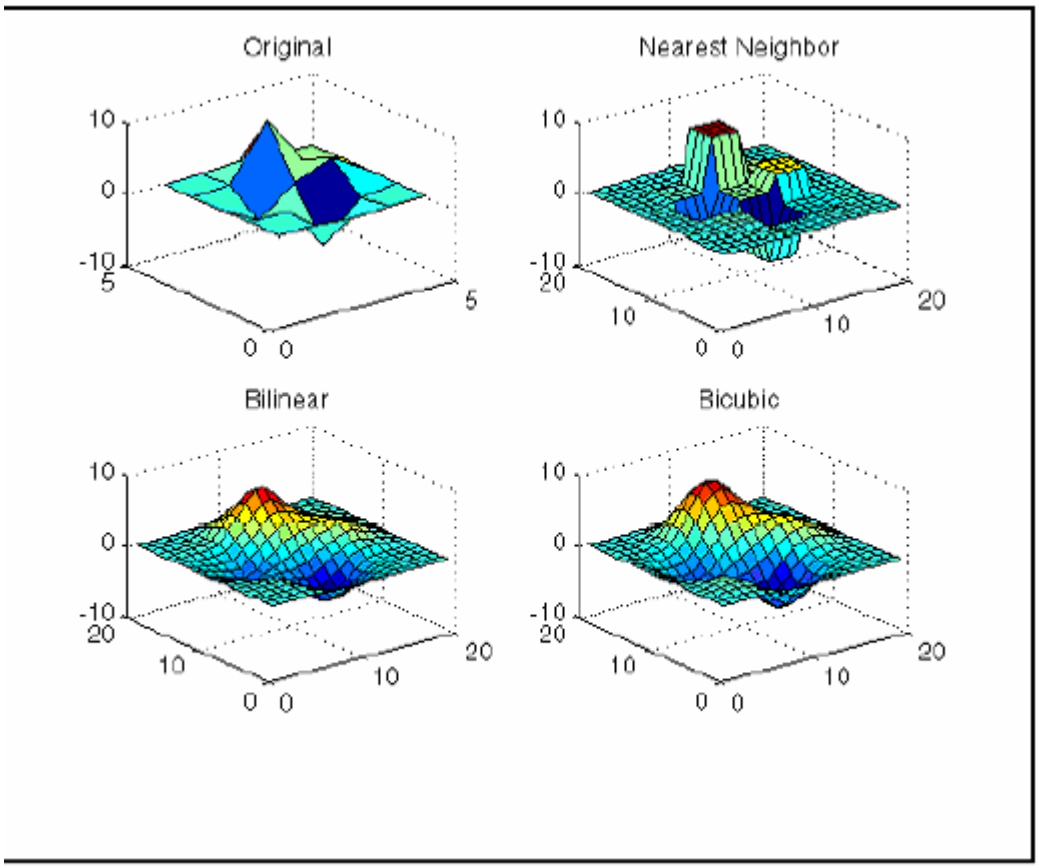
جعبه ابزار پردازش تصویر MATLAB سه روش درونیابی را مهیا می کند:

- نزدیکترین همسایه
- درونیابی خطی
- درونیابی مکعبی

همان طور که قبلاً بحث کردیم، درونیابی نزدیکترین همسایگی یک سطح ثابت قطعه ای را بین مقادیر پیکسل ها جای می دهد. مقدار پیکسل درونیابی شده، همان مقدار نزدیکترین پیکسل است. درونیابی خطی یک سطح خطی را بین مقادیر پیکسل موجود جای می‌دهد. این درونیابی، قطعه ای خطی است و سریعتر از درونیابی مکعبی است. درونیابی مکعبی، یک سطح مکعبی را بین مقادیر موجود جای می دهد. مقدار پیکسل جدید از روی ١٦ همسایگی آن به دست می آید. این روش قطعه ای مکعبی است و سطحی بسیار هموارتر از درونیابی خطی ایجاد می کند. شکل ٥ این سه روش درونیابی را مقایسه می کند این شکل با دستورات MATLAB زیر تولید می شود:

```
%resize_peak.m
[x,y,z] = peaks(5);
subplot(2,2,1), surf(z),title('Original');
zrep=imresize(z,4,'nearest');
subplot(2,2,2), surf(zrep);title('Nearest Neighbor');
zlin=imresize(z,4,'bilinear');
subplot(2,2,3), surf(zlin);title('Bilinear');
zcubic=imresize(z,4,'bicubic');
subplot(2,2,4), surf(zcubic),title('Bicubic');
```

تغییر نرخ یک روش با ارزش در پردازش تصویر است. کارایی آن با تابع درونیابی مورد استفاده تعیین می شود. درونیابی نزدیکترین همسایگی خصوصاً برای تصاویر دودویی به خاطر عدم وجود سطوح میانی مناسب است. درونیابی خطی سریع و با نتایج خوب است. درونیابی اسپلاین مکعبی بهترین است ولی زمان موردنیاز آن در صورت بزرگ بودن تصویر، زیاد است. انتخاب مناسب به کاربرد موردنظر وابسته است.



شکل ۵: مقایسه متد های مختلف درونیابی

٤- آزمایش ها

٤-١- بهبود تصویر با استفاده از برنامه های دموی MATLAB

١- با برنامه image demo بازی کنید. تأثیر ابزار مختلف بهبود کانتراست را مشاهده کنید (برای دو تصویر مختلف). برای هر تصویر روش بهبود را معین کنید. (برای تنظیم مقادیر، نیاز به پیدا کردن بهترین تناظر به وسیله سعی و خطا دارید). در گزارشتان، نتایج

بهترین روش برای هر تصویر را ارائه دهید. مشاهدات خود را از نقاط مثبت و منفی روش‌های مختلف ارائه دهید. توجه ویژه به تغییرات هیستوگرام بکنید.

۲- با برنامه nfiltdemo بازی کنید. تأثیر ابزارهای مختلف حذف نویز برای هر یک از سه نوع نویز (که با مقادیر پیش فرض یا مقادیر انتخابی شما تولید شده‌اند) را بررسی کنید. برای هر نوع نویز، روشی که بهترین نتیجه را می دهد یادداشت کنید. هر مشاهده ای از مزایا و معایب روش ها را بنویسید.

۳- با برنامه edgedemo بازی کنید. تأثیر ابزار مختلف آشکارسازی لبه را بررسی کنید. برای هر تصویر روشی که بهترین نتیجه را می دهد یادداشت کنید.

مشاهدات خود از مزایا و معایب روش ها را بنویسید.

در هر کاری، اگر نمی دانید یک پردازش خاص چگونه انجام میشود از کمک MATLAB استفاده کنید و شرح جزییات تابع صدا شده را فرا بگیرید.

۲-٤- درباره ابزار گرافیکی MATLAB برای پردازش تصویر یاد بگیرید

۱-۲-٤- پیدا کردن توابع MATLAB برای پردازش تصویر

برای اینکه مطمئن شوید جعبه ابزار پردازش Image Processing قبلاً بر روی سیستم شما نصب شده است و تمام توابع توسط جعبه ابزار فراهم شده است، تایپ کنید:

```
help images
```

اگر جعبه ابزار نصب شده باشد MATLAB لیست توابع پردازش تصویر را نشان می دهد.

۲-۲-٤- ورود و خروج تصاویر

قبل از جلو رفتن در هر عملیات پردازش تصویر نیاز به بحث در مورد فرایند ورود و خروج تصاویر به محیط MATLAB داریم.

MATLAB فرمت های متعددی را پشتیبانی می کند شامل JPEG (JPG)، TIFF، BMP و غیره. می توانید از imread برای خواندن هر فایل تصویری با فرمت پشتیبانی شده استفاده کنید. از help imread برای پیدا کردن جزییات بیشتر کمک بگیرید.

برای خواندن تصاویر سیاه سفید یا رنگی دارای یکی از فرمت های پشتیبانی شده از

```
[A] = IMREAD(FILENAME,FMT)
```

استفاده کنید که داده های تصویر در FILENAME را داخل ماتریس A می خواند. اگر فایل تصویر حاوی تصویر سیاه سفید باشد، A یک ماتریس ۲ بعدی می شود. اگر فایل حاوی تصویر رنگی RGB باشد، ماتریس A سه بعدی خواهد بود.

به عنوان مثال اگر شما یک تصویر رنگی با فرمت JPEG با نام image.jpg با سایز M×N دارید با استفاده از

```
[A]=IMREAD('image.jpg','JPG')
```

یک ماتریس M×N×۳ با نام A که(A (۱:M , ۱:N , ۱ مولفه قرمز (A (۱:M , ۱:N , ۲ مولفه سبز (A (۱:M , ۱:N , ۳ مولفه آبی را ذخیره می کنند. اگر تصویر در فرمت indexed است از دستور زیر استفاده کنید:

```
[A,MAP] = IMREAD(FILENAME,FMT)
```

A مقادیر تصویر و MAP نقشه رنگ را ذخیره می کند.

تابع 'imwrite' اجازه ذخیره یک ماتریس دار، در فایلی با فرمت مشخص را می دهد. مثلاً:

```
IMWRITE(A,'outimg.jpg','JPG')
```

داده تصویر A را در فایل 'outimg.jpg' با استفاده از فرمت فشرده سازی JPEG ذخیره می‌کند. از 'help imwrite' برای جزییات بیشتر استفاده کنید.

۳-۲-٤- نمایش تصویر

تابع imshow هر تصویر پشتیبانی شده را نمایش می دهد. برای تصاویر indexed، imshow تصویر را در محورهای حاضر با نقشه رنگ مشخص شده نصب می کند. بدون نام colormap، imshow از colormap حاضر استفاده می کند. برای کاربرد انتشار web معمولاً با تصاویر رنگی indexed کار می کنیم.

تصاویر مختلفی با قالبهای TIFF، BMP و JPG را انتخاب کنید. این تصاویر را در محیط MATLAB بخوانید، نمایش دهید و بنویسید، تا با فرآیند آشنا شوید.

۳-٤- تغییر نرخ تصویر با MATLAB

صدا کردن تابع

```
B = IMRESIZE(A,M,'method')
```

تصویر M برابر شده (بزرگتر یا کوچکتر) تصویر A را در B می دهد. تصویر B با روش های مشخص شده در رشته \method\ به دست می آید:

```
'nearest','bilinear', or 'bicubic'.
```

صدا کردن تابع

```
B = IMRESIZE(A,[MROWS NCOLS],'method')
```

ماتریس با اندازه MROWS×NCLOS می دهد.

اگر A یک تصویر سیاه سفید باشد، تغییر نرخ بر روی ماتریس دوبعدی موجود در A انجام می‌شود. اگر A رنگی باشد، همان عملیات بر روی تمام مولفه های رنگ A انجام می شود.

حال نوبت شماست که آزمایش تغییر نرخ را انجام دهید. می توانید هر تصویر پشتیبانی شده توسط Matlab تا یک تصویر سیاه سفید در فرمت raw را استفاده کنید.

۱- با فاکتور ۲ بدون پیش فیلتر نرخ را کم کنید و سپس با فاکتور ۲ آنرا با استفاده از سه روش درونیابی بزرگ کنید، به روش زیر جلو بروید.

- تصویر را بخوانید.
- تصویر را با فاکتور ۲ در هر دو جهت افقی و عمودی بدون فیلتر کوچک کنید.
- تصویر حاصل را با فاکتور ۲ بزرگ کنید، از روش درونیابی نزدیکترین همسایه نیز استفاده کنید.
- تصویر حاصل را در یک فایل ذخیره کنید.
- اختلاف بین تصویر اصلی و تصویر پس از کوچک و بزرگ شدن را پیدا کنید و تصویر اختلاف را از روی تفاضل پیکسل به پیکسل دو تصویر پیدا کنید.
- فرآیند بالا را با روش درونیابی خطی تکرار کنید.
- فرایند بالا را با روش درونیابی مکعبی تکرار کنید.

نتایج به دست آمده از سه روش را مقایسه کنید. این مقایسه بر حسب تصاویر کوچک / بزرگ شده و اختلاف تصاویر باشد.

تصویر اصلی و پردازش شده (بعد از کاهش و افزایش نرخ) و تصویر اختلاف را چاپ کنید و در گزارشتان بیاورید. همچنین در دیسکت تحویلی هر دو تصویر اصلی و پردازش شده را ذخیره کنید.

به اختلاف زمان اجرای هر روش در گزارشتان اشاره کنید.

آیا هیچ پیغامی هنگام اجرای هر کدام از فرمان ها دریافت کردید؟ آنها چگونه حل شدند؟

۲- مراحل قبلی را با فیلتر پیش فرض down-sampling تکرار کنید. نتایج روش های مختلف را مقایسه کنید. همچنین با نتایج به دست آمده در ۱ مقایسه کنید (اختیاری).

نکته: وقتی M از یک کمتر است،

```
IMRESIZE(A,M,'method')
```

دستور بالا پیش فیلتر را به کار نمی برد. (اگر 'method' و 'nearest' باشد) و اگر روش خطی یا مکعبی باشد یک فیلتر ۱۱ تایی استفاده می کند. به جای استفاده از Imresize برای down-sampling بدون پیش فیلتر می توانید از دستور زیر نیز استفاده کنید:

```
B=A(1:M:W,1:M:N) or B=A(1:M:W,1:M:N, 1:3)
```

۵- گزارش

در گزارشتان باید اسکریپت های matlab استفاده شده در آزمایشها به همراه نتایج مشخص شده در لیست آزمایش ها و مشاهدات شما آورده شود. باید برای پاسخ به هر پرسش درباره دستورات استفاده شده در این بخش آماده باشید.