



University of Tehran

School of Electrical and Computer Engineering



Digital Image Processing

Instructor: Hamid Soltanian-Zadeh

Assignment 1

Sasan Keshavarz

810199253

Spring 2022

فهرست

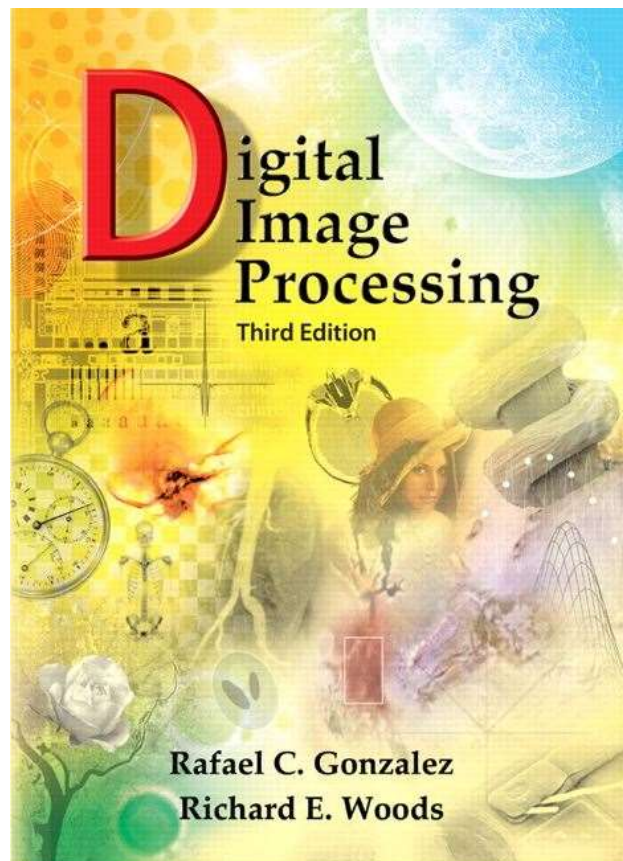
چکیده	۱
سؤال ۱	۲
بخش ۱ و ۲	۲
بخش ۳	۲
بخش ۴	۳
بخش ۵	۴
بخش ۶	۴
سؤال ۲	۶
سؤال ۳	۱۰
سؤال ۴	۱۱
بخش اول	۱۱
بخش ۲	۱۲
سؤال ۵	۱۳
بخش اول	۱۳
بخش دوم	۱۴
سؤال ۶	۱۶
سوالات تحلیلی	۱۷
پیوست ۱: روند اجرای برنامه	۱۹
مراجع	۲۰

در این تمرین با اصول اولیه و دستورات پایه‌ای متلب برای پردازش تصویر آشنا خواهیم شد. این دستورات شامل تغییرات شدت روشنایی تصاویر و تبدیل‌های هندسی و ریاضیاتی بر تصاویر هستند. از عملگرهای منطقی هم برای پردازش استفاده خواهد شد. همچنین دستوراتی برای فیلتر مکانی هم استفاده خواهند شد.

سؤال ۱

بخش ۱ و ۲

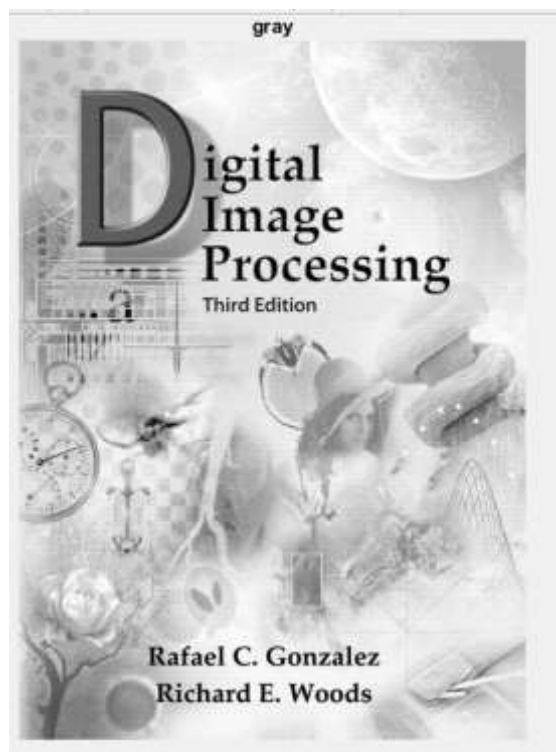
در این قسمت صرفاً به لود کردن و نمایش تصویر رنگی DIP با دستورات `imread` و `imshow` میپردازیم.



شکل ۱ تصویر رنگی DIP

بخش ۳

با تابع `rgb2gray` تصویر رنگی را به سیاه سفید تبدیل میکنیم. نتیجه در شکل ۲ نمایش داده شده است.



شکل ۲ تصویر سیاه-سفید شده DIP

بخش ۴

با دستور `im2double` فرمت تصاویر را از `unit8` به `double` تبدیل میکنیم. در نتیجه شدت روشنایی هر پیکسل عددی بین صفر تا ۱ خواهد شد.

همچنین در بعضی کاربردها مانند تقسیم تصاویر باید از فرمت `double` استفاده کنیم. یک مثال برای این کاربرد هنگامی است که می‌خواهیم نورپردازی محیط را اصلاح کنیم. همچنین پیوسته شدن مقادیر شدت روشنایی میتواند کاربردهای دیگری برای ما داشته باشد. همچنین چون رنج شدت روشنایی مشخص است محاسبات پایداری بیشتری خواهند داشت.

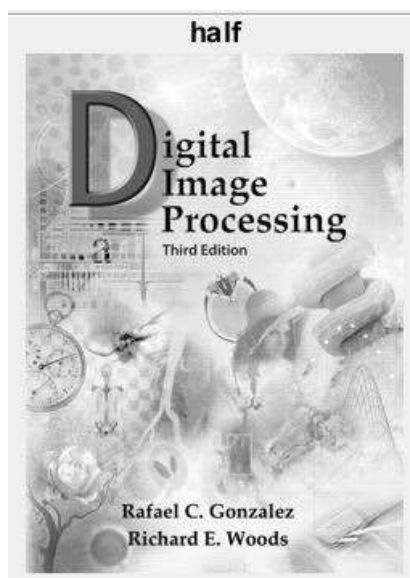
لازم به ذکر است در بعضی کاربردها مانند تفریق استفاده از فرمت `unit8` بهتر است و نتایج بهتری میدهد.

بخش ۵

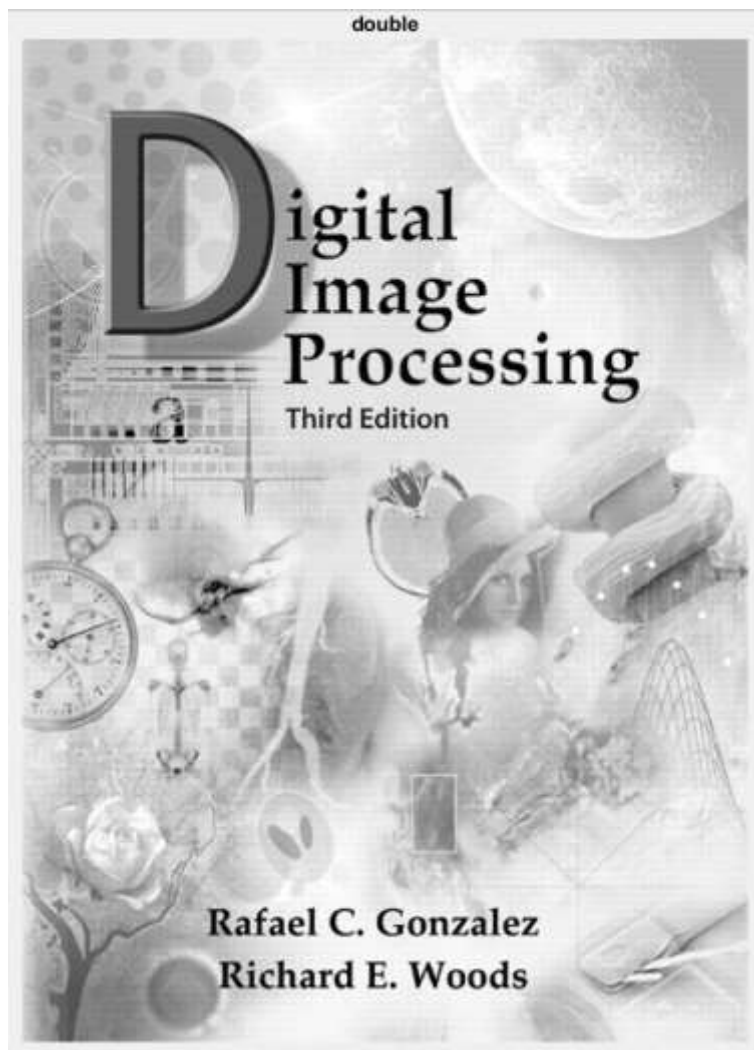
تصویر ایجاد شده توسط تابع `im2double` را با فرمت `tif` ذخیره میکنیم. این کار با دستور `imwrite` انجام میشود.

بخش ۶

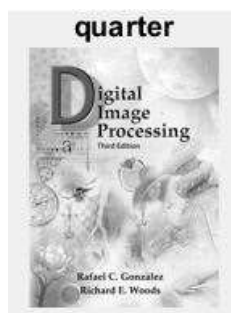
با استفاده از دستور `imresize` اندازه تصویر را دو برابر، نصف و یک‌چهارم میکنیم. نتایج این دستور در اشکال ۳ تا ۵ نمایش داده شده است. با انجام بزرگ‌نمایی در واقع تعداد پیکسل‌ها زیاد میشود و برای تعیین شدت روشنایی پیشگلهای جدید از درون‌یابی استفاده میشود. بزرگ‌نمایی باعث میشود لبه‌های موجود در تصویر نرم شوند و کوچک‌نمایی باعث میشود لبه‌ها تیزتر شوند. با کوچک کردن تصاویر جزییات تصویر کاهش می‌یابد چون بخشی از اطلاعات حذف میشود.



شکل ۳ تصویر با بزرگ‌نمایی ۰.۵ برابر



شکل ۴ تصویر با بزرگ‌نمایی ۲ برابر



شکل ۵ تصویر با بزرگ‌نمایی ۰,۲۵ برابر

سوال ۲

در این سوال قصد داریم تبدیل‌های افاین را بر تصویر T.tif اعمال کنیم. ابتدا مطابق جدول ۱ برای هر تبدیل یک ماتریس تبدیل می‌سازیم. این کار را با دستور maketform انجام می‌دهیم. سپس تبدیل‌های مقیاس، دوران، انتقال، برش افقی و برش عمودی را بر تصویر اعمال می‌کنیم.

جدول ۱ ماتریس‌های مربوط به تبدیلات افاین برگرفته شده از کتاب گونزالس

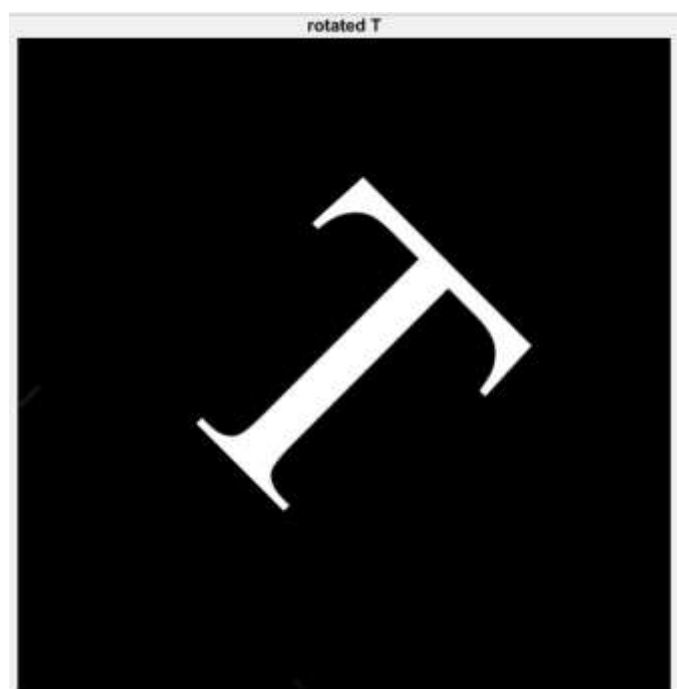
Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \sin \theta + w \cos \theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

در تبدیل مقیاس مقدار بزرگ‌نمایی ۲ در نظر گرفته شده است.



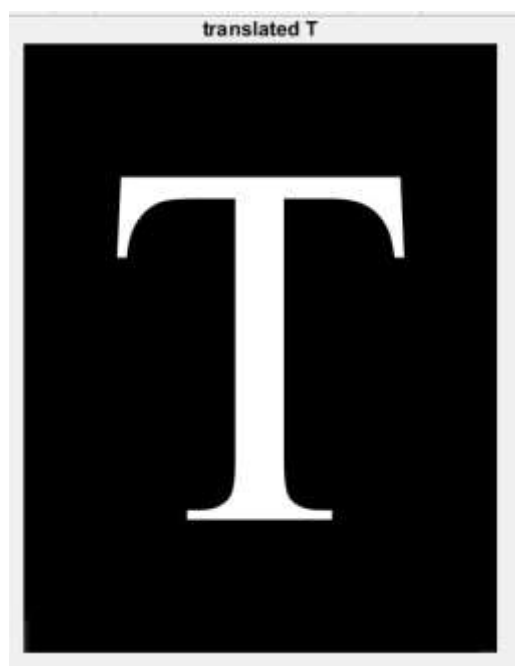
شکل ۶ تصویر حاصل از تبدیل مقیاس

در تبدیل دوران زاویه چرخش ۴۵ درجه در نظر گرفته شده است.



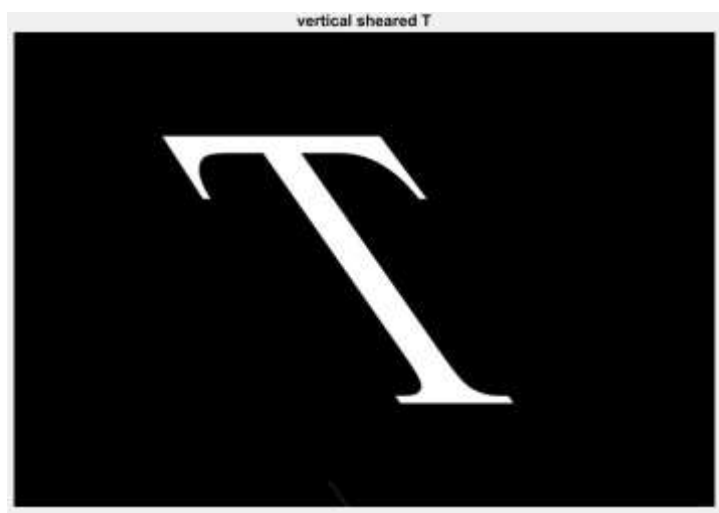
شکل ۷ تصویر حاصل از تبدیل دوران

در تبدیل انتقال مقدار انتقال در هر محور ۱۰۰ در نظر گرفته شده است.



شکل ۸ تصویر حاصل از تبدیل انتقال

در تبدیل برش عمودی مقدار برش در ماتریس ۰,۷ در نظر گرفته شده است. در صورتی که این مقدار کمی بیشتر از ۳ باشد تصویر به شدت کشیده خواهد شد.



شکل ۹ تصویر حاصل از تبدیل برش عمودی

در تبدیل برش افقی مقدار برش در ماتریس ۱,۵ در نظر گرفته شده است.



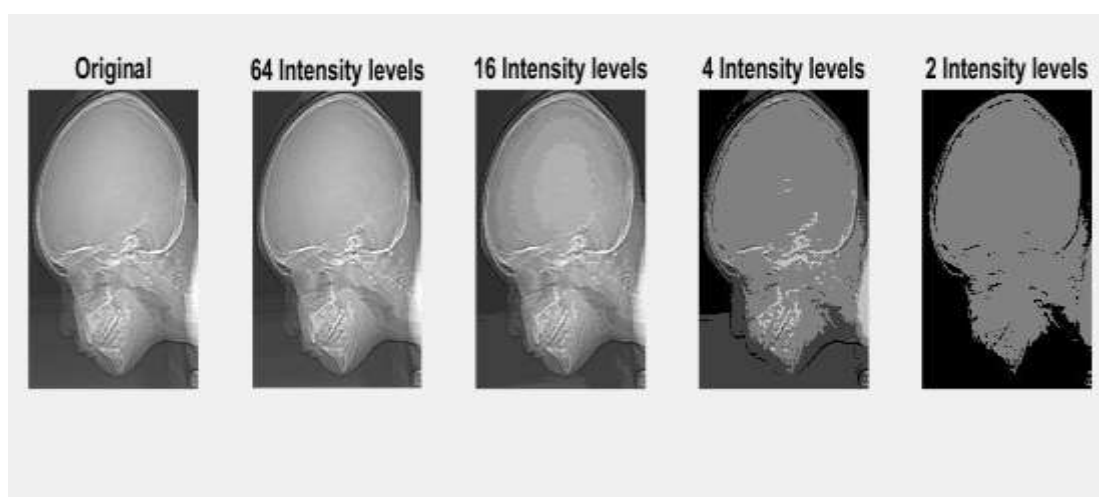
شکل ۱۰ تصویر حاصل از تبدیل برش افقی

سوال ۳

تصویر skull.tif داده شده دارای ۲۵۶ سطح روشنایی است. خواسته سوال این است که سطوح روشنایی را به ۶۴، ۱۶، ۴ و ۲ تبدیل کنیم و سایز تصویر را تغییر ندهیم.

برای مثال می‌خواهیم سطوح روشنایی را به ۱۶ تبدیل کنیم. مقادیر سطوح روشنایی تصویر اولیه را به ۴ تقسیم می‌کنیم و سپس با دستور floor اعداد به وجود آمده را حذف می‌کنیم. پس از آن دوباره مقادیر را در ۴ ضرب می‌کنیم تا به مقادیر اولیه مشابه گردد. با این کار دیگر مقادیر روشنایی که ضریب ۴ نباشند در تصویر وجود ندارند اما بازه تغییرات سطوح روشنایی باز بین ۰ تا ۲۵۵ (۲۵۶) است.

برای تبدیل سطوح روشنایی به ۱۶، ۴ و ۲ هم از همین الگوریتم استفاده می‌کنیم. نتایج در شکل ۱۱ نمایش داده شده‌اند.



شکل ۱۱ نتایج حاصل از تغییر سطوح روشنایی تصاویر

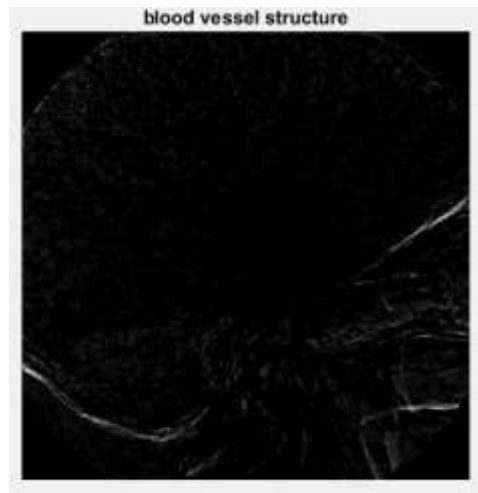
همانطور که از شکل ۱۱ مشخص است، هرچقدر تصویر سطوح روشنایی بیشتر داشته باشد جزئیات و اطلاعاتی بیشتری از تصویر تفسیر خواهد شد. البته تفاوت چندانی بین تصویر ۲۵۶ سطحی و ۶۴ سطحی نیست و میتوان برای کاهش حجم تصویر و انتقال کم‌هزینه تر تصویر از ۶۴ سطح روشنایی استفاده کرد. این انتخاب به کاربرد مورد نظر برمیگردد. در تصاویر ۱۶ سطحی و ۴ سطحی مرزهایی به اشتباه در درون مغز به وجود آمده‌اند که ناشی از الگوریتم کاهش سطوح تصویر هستند و حقیقی نیستند. همچنین تصویر ۲ سطحی اطلاعات چندانی ندارد و صرفاً مرز مجموعه و محیط در آن مشخص است.

سوال ۴

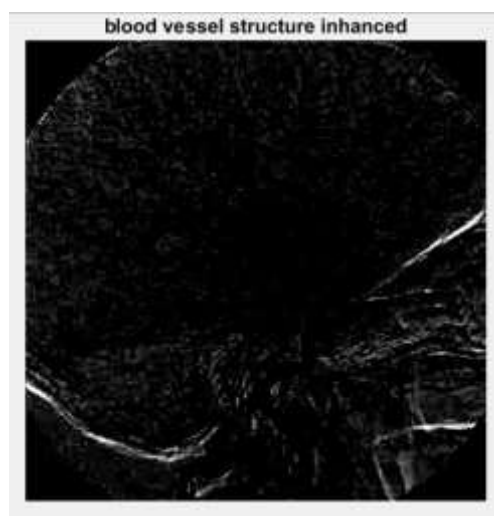
در این بخش قصد داریم از عملگرهای محاسباتی برای پردازش تصاویر استفاده کنیم.

بخش اول

در بخش اول مسئله قصد داریم با تفریق کردن تصویر آنژیوگرافی بیمار بعد از تزریق ماده حاجب از تصویر آنژیوگرافی بیمار بدون ماده حاجب، ساختار رگهای خونی بیمار را نمایش دهیم. بازه سطوح روشنایی در این تصویر [0,52] است. برای اینکه این ساختار بهتر نمایش داده شود خودم یک بار هم تصویر را در یک ضریب ۱۰ ضرب کردم تا جزییات نمایان شود. نتایج در شکلهای ۱۲ و ۱۳ قابل مشاهده هستند.



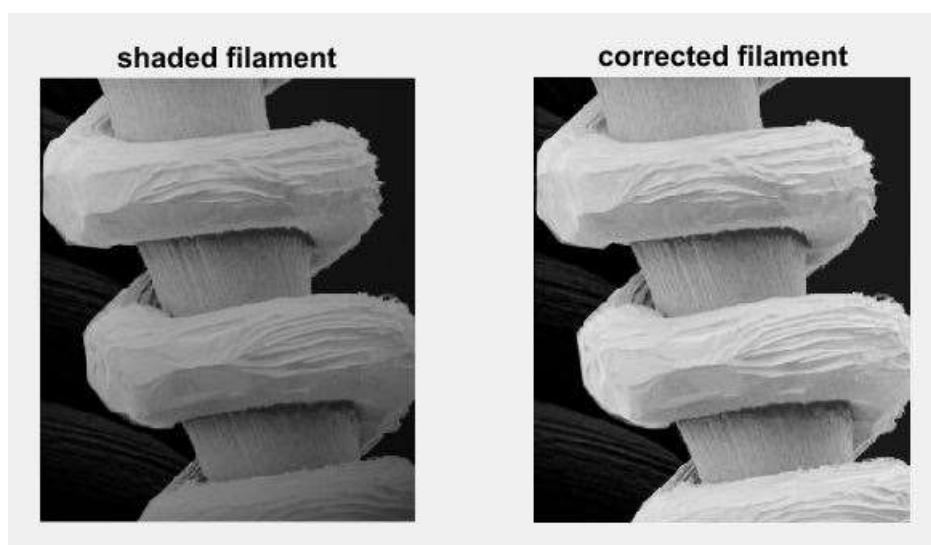
شکل ۱۲ تصویر ساختار رگ های بیمار



شکل ۱۳ تصویر تقویت شده ساختار رگهای بیمار

بخش ۲

در این بخش کاربرد تقسیم تصاویر را برای اصلاح نورپردازی نشان خواهیم داد. باید فرمت تصاویر از unit8 به double تبدیل شود و تا عمل تقسیم انجام شود. تصویر فیلامان سایه‌دار و الگوی نورپردازی سنسور داده شده است. پس با تقسیم تصویر فیلامان بر این الگو میتوان تصویر را اصلاح کرد و اثر نورپردازی و سایه‌ها که شدت روشنایی واقعی فیلامان را تغییر داده‌اند اصلاح کرد. تصویر فیلامان سایه‌دار و تصویر اصلاح شده در شکل ۱۴ نمایش داده شده است.



شکل ۱۴ تصویر فیلامان سایه‌دار و فیلامان اصلاح‌شده

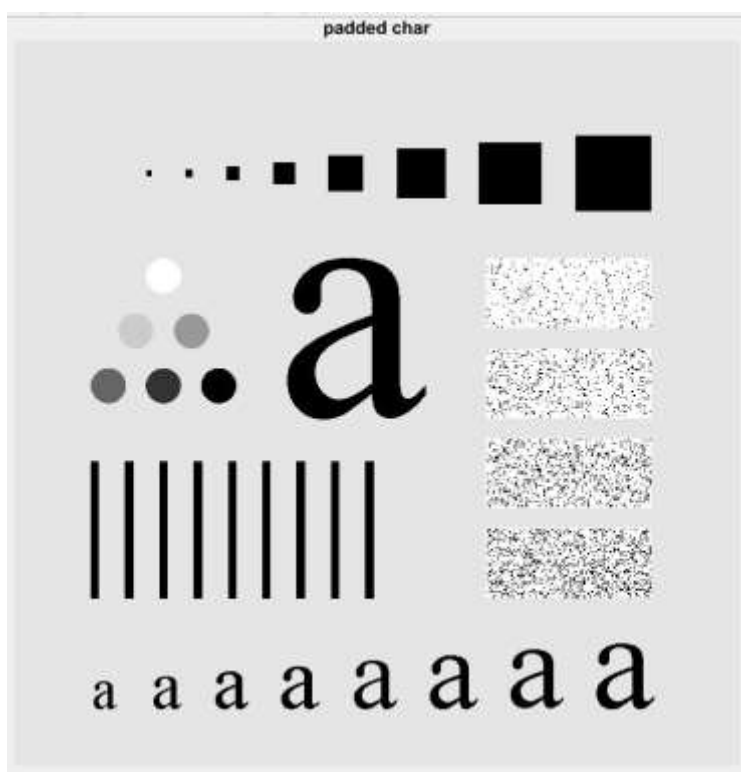
در صورتی که الگوی نورپردازی محیط را ندانیم باید از یک صفحه خاکستری یک با شدت روشنایی یک‌دست استفاده کنیم و از آن تصویر بگیریم، در تصویر حاصل الگوی نورپردازی و اثر آن به سطوح روشنایی مشخص خواهد بود. چون تصویر اولیه یک‌دست بوده است و تغییرات اشی از نورپردازی سنسور است.

سوال ۵

در این سوال قصد داریم اثر میانگیری محلی را بر تصویر char بررسی کنیم. میانگیری محلی یکی از روشهای محبوب برای کاهش حذف نویز تصویر است. همچنین باید این کار را بدون استفاده از توابع آماده متلب مانند conv2 انجام دهیم.

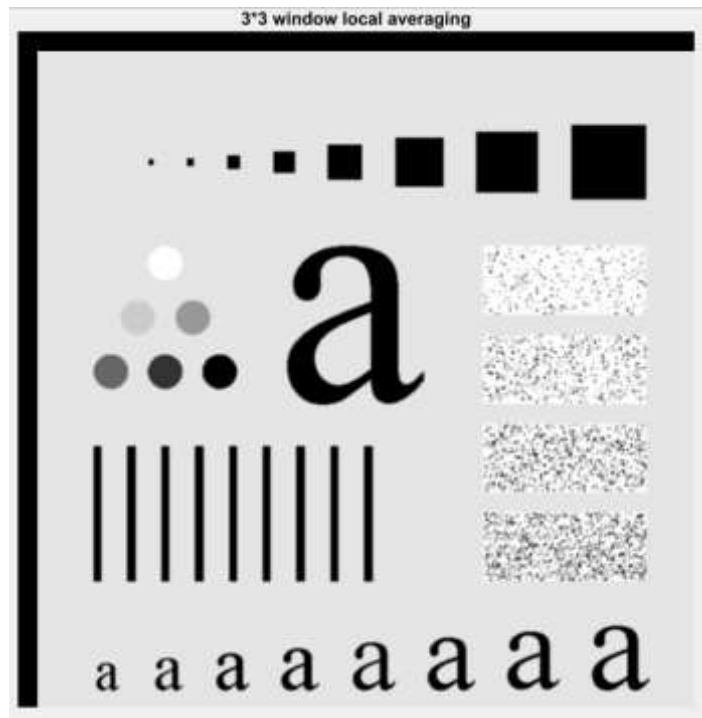
بخش اول

در این قسمت قصد داریم از پنجره‌های 3×3 برای انجام میانگیری محلی استفاده کنیم. پس از لود کردن تصویر chart، آن را padded میکنیم. این عمل را با دستور paddarray انجام میدهیم. علت انجام این عمل این است که میانگیری محلی در گوشه‌ها دچار مشکل میشود زیرا پنجره وقتی در گوشه باشد بخشی از آن پنجره بیرون از تصویر قرار میگیرد. برای آنکه در بدترین شرایط (یعنی پنجره 41×41) هم دچار مشکل نشویم سایز padding را ۲۰ در ۲۰ در نظر گرفتیم که یک قاب ۲۰ تایی به هر یکی از اضلاع تصویر اضافه میکند. مقادیر این قاب را با استفاده 'replicate' تعیین کردیم که مقادیر مرزها را در قاب قرار میدهد. تصویر جدید در شکل ۱۵ نمایش داده شده است.



شکل ۱۵ تصویر char بعد از padding با سایز ۲۰ در ۲۰

سپس برای انجام عملیات میانگیری محلی از دو حلقه تو در تو استفاده کردم که در هر مرحله مقادیر خود پیکسل را با پیکسلهای درون پنجره اطراف آن جمع میکند و بر سائز پنجره تقسیم میکند و مقدار حاصل شده مقدار جدید سطح روشنایی پیکسل مذکور خواهد بود. نتیجه در شکل ۱۶ دیده میشود.



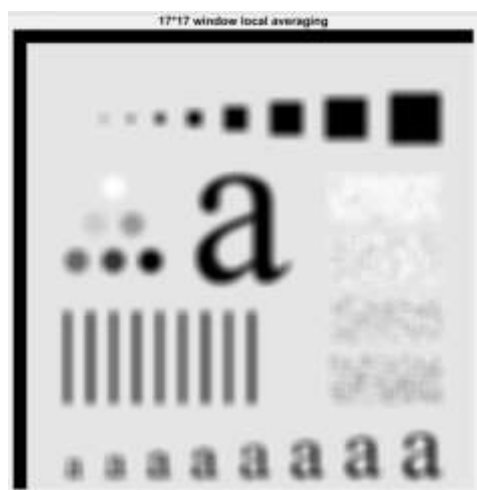
شکل ۱۶ تصویر حاصل از میانگین گیری محلی با پنجره ۳*۳

بخش دوم

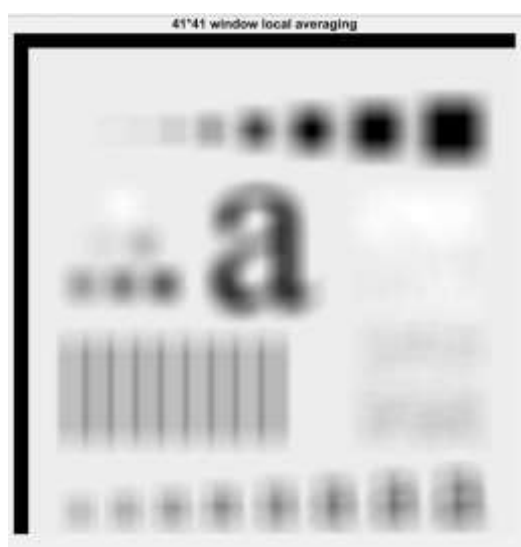
در این بخش همان اعمال بخش قبل را با پنجره‌های با سائز ۹*۹، ۱۷*۱۷ و ۴۱*۴۱ انجام میدهیم. صرفا باید اعداد مربوط به هر پنجره در حلقه‌ها و سائز پنجره که مجموع بر آن تقسیم میشود را تغییر داد. برای مثال برای پنجره ۱۷*۱۷ از ۸ پیکسل سمت چپ تا ۸ پیکسل سمت راست و همچنین ۸ پیکسل بالا تا ۸ پیکسل پایین پیکسل مورد نظر جمع میشوند بر ۲۸۹ تقسیم میشوند. نتایج در شکل‌های ۱۷ تا ۱۹ نمایش داده شده اند.



شکل ۱۷ تصویر حاصل از میانگین گیری محلی با پنجره ۹*۹



شکل ۱۸ تصویر حاصل از میانگین گیری محلی با پنجره ۱۷*۱۷



شکل ۱۹ تصویر حاصل از میانگین گیری محلی با پنجره ۴۱*۴۱

سوال ۶

درون‌یابی به طور کلی روشی است که برای تخمین مقادیر مجهول در تصویر استفاده میشود. یکی از کاربردهای آن در هنگام کوچک‌سازی تصاویر است. در این سوال از سه روش متفاوت nearest neighbor ، bilinear و bicubic برای درون‌یابی استفاده میکنیم. تصویر watch داده شده دارای ابعاد مناسبی نیست پس ابتدا آن را اصلاح میکنیم و فقط لایه اول را برمی‌داریم. سپس در هر مرحله ابتدا با روش درون‌یابی مطلوب تصویر را کوچک میکنیم. برای این کار از دستور imresize استفاده میکنیم. در برگرداندن تصویر به سائز اولیه دیگر از درون‌یابی استفاده نمیکنیم. نتایج این عملیات در شکل ۲۰ نمایش داده شده است.



شکل ۲۰ تصاویر حاصل از درون‌یابی با روشهای مختلف

همانطور که مشخص است قسمتهایی از تصویر nearest به خصوص در مرزها و لبه‌ها دچار اعوجاج شده‌اند. تصویر bilinear کیفیت قابل قبولی دارد، اما بهترین نتیجه با روش bicubic حاصل شده است.

سوالات تحلیلی

تقریب اول پروازش تصویر.

2.2) بازگاری در شبی در چش شرایطی دیده انسان Scotopic است. فقط سلول های استخوانی فعال هستند و سبب میزان شدت روشنایی محیط درک روشنایی سطحی در ذهن شکل گرفته. حساسیت چشم انسان نسبت به روشنایی در شرایط مختلف تغییر می کند. این امر به این دلیل اتفاق افتاده که در شرایط مختلف محیطی چشم بتواند بین اینها تفاوت قائل شود.



2.5) ابعاد لنز: $V \times V$ mm
 شماره: 1024×1024

اندازه هر پیکسل $\frac{V}{1024} \times \frac{V}{1024}$ mm خواهد بود. $\frac{h}{100} = \frac{V}{1024} \rightarrow h = 100 \frac{V}{1024}$

$\frac{1024}{100 \text{ mm}} = 10 \text{ elements/mm} \rightarrow \text{pair} = 4 \text{ elements} \rightarrow 4 \text{ pair/mm}$

2.7) $i(x, y) = k e^{-[(x-x_0)^2 + (y-y_0)^2]}$

$R = 1$ $k = 255 \rightarrow f(x, y) = 255 e^{-[(x-x_0)^2 + (y-y_0)^2]}$
 اعداد حسابی بین 0 تا 255 است که باید گمایتیزه شود.

$\frac{255}{2^k} \geq 1 \rightarrow 32 \geq 2^k \rightarrow 5 \geq k$
 به ازای این مقدار k خطای گمایتیزه اتفاق افتاده.

2.9) $\text{bandrate} = \text{bits/sec}$

transmission: start bit 8 bits stop bit $\rightarrow 10 \text{ bits}$

(a) تصویر 256 سطح دارد پس 8 بیت است. حجم داده $1024 \times 1024 \times (8+2)$

$\text{time} = \frac{\text{total data}}{\text{band rate}} = \frac{1024 \times 1024 \times 10}{54 \times 10^3} = 187/256 \text{ s} = 3/12 \text{ minutes}$

$$t^* = \frac{10^8 \times 10^4 \times 10}{3 \times 10^9} = 3/29 \approx 3/2 \text{ s} \quad (2.9) \quad (b)$$

با این بادریت $3/2$ طولی کند تا تصویر منتقل شود.

$$19/9 : \text{نسبت تصویر} = 1125 \text{ lines} = \text{رزولوشن عمودی}$$

$$\text{رزولوشن افقی} = \frac{19}{9} \times 1125 = 2000 \text{ pixels/line}$$

پس هر تصویر باین 2000×1125 دارد. همچنین هر تصویر ۲۴ بیت برای به رنگ قرمز، سبز و آبی دارد (برای هر رنگ ۸ بیت).

همچنین گفته شده که هر ثانیه $\frac{1}{30}$ ثانیه زمانی بزرگ تصویر تولید می شود و چون دو میدان داریم پس در هر $\frac{1}{30}$ ثانیه یک تصویر تولید می شود. که چون تعداد رنگ های مختار به آن تولید می شود پس این را در محاسبات مقدار دیتای تولید شده در ۲ ساعت باید لحاظ کرد.

$$2H = 2 \times 90 \frac{\text{min}}{\text{hr}} \times 60 \frac{\text{sec}}{\text{min}} = 7200 \text{ sec}$$

$$\text{total data} = 7200 \text{ sec} \times 1125 \times 2000 \times 8 \times 3 \frac{\text{bits}}{\text{sec}} \times 3 \Rightarrow$$

(size) (bits) (sec) (times)

$$\text{total data} = 1/199 \times 10^{12} \text{ bits} = 11/99 \text{ terabits}$$

(2.21) (a) هر دو تصویر سطح روشنایی بین ۲۵۵ تا ۲۵۵ دارند. تصویر حاصل از تفاضل این

دو از هم سطح روشنایی باین ۲۵۵-۲۵۵ را در برخواهد گرفت. با توجه به اینکه برای

حاصل هم ۸ بیت در نظر گرفته شده است باید مقادیر منفی را یا برابر ۲۵۵ یا صفر یا ... قرار

دهیم. چون قصد داریم تفاضلهای نگار انجام دهیم انتخاب صفر برای مقادیر منفی منطقی تر است چون

در صورت تفاضل یک پیکسل باشد در روشنایی صفر از یک پیکسل متناظر دوباره رقم منفی حاصل می شود که

باز هم باید آن را صفر قرار داد. در صورتی که این عمل تفاضلی را بارها انجام دهیم تصویر حاصل در همه

پیکسل ها صفر خواهد شد. پس آن پیکسل های که در مقادیر دوم مقدار صفر داشته اند و هر چه در هم که تفاضل

انجام شده باز مقدار پیکسل در مقادیر اول کم نخواهد شد.

"ط" به- مثلا اگر یک پیکسل در یک تصویر صفر باشد و در تصویر دیگر نه، تغییر کردن تفاوت در نتیجه این عملگر تغییر خواهد کرد.

پیوست ۱: روند اجرای برنامه

پوشه تصاویر در فایل کدها قرار داده شده است و با انجام `set path` کدها اجرا خواهند شد. محل ذخیره شدن دستور `imwrite` تمرین اول در همان محل پوشه فایل `m`. خواهد بود. بخش‌های مختلف هر کد با %% از هم تفکیک شده‌اند. در صورت نیاز توضیحاتی در خود کد نوشته شده است.

- [1] R. C. Gonzalez and R. E. Woods, "Digital image processing." Prentice hall Upper Saddle River, NJ, 2002.
- [2] MATLAB help