



## مقدمه:

در این پروژه می‌خواهیم به پردازش سیگنال‌های دو بعدی همانند تصویر بپردازیم. همانطور که می‌دانید یک تصویر ماتریسی  $M \times N$  می‌باشد که هر درایه آن یک مقدار می‌گیرد. این مقدار را شدت روشنایی آن درایه (پیکسل) می‌نامند. در بیشتر دستگاه‌ها و نرم‌افزارها برای هر پیکسل یک بایت (8 bit) در نظر می‌گیرند. بنابراین هر درایه ماتریس تصویر مقداری از ۰ تا ۲۵۵ می‌گیرد.

همچنین تصاویر ثبت شده توسط دستگاه‌ها معمولاً با مشکلاتی نظیر انواع نویز و آرتیفک‌های حرکتی همراه هستند. که هدف ما برطرف کردن یا کاهش تاثیرات این عوامل مزاحم بروی تصویر مطلوب می‌باشد. تصویر زیر نمونه‌ای از یک تصویر مطلوب و خراب شده توسط نویز سینوسی و آرتیفک حرکتی است.



## قسمت اول:

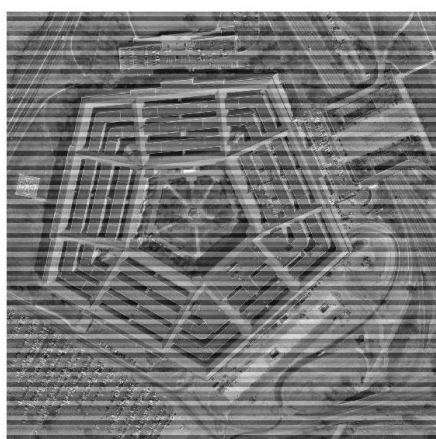
همانطور که مشخص گردید، هدف ما برطرف کردن یا کاهش نویز از تصویر ورودی است.

- در این قسمت شما بایستی تصویر ورودی “car.jpg” را بارگذاری کرده و در گام‌هایی نویز آن را برطرف کنید.

## جهت آشنایی:

- نویزهای سینوسی، نویزهایی هستند که همچون یک الگوی خاص بروی تصویر ظاهر می-گردند و ویژگی آنها تک فرکانسی بودن است. (یعنی یک  $u_0$  و  $v_0$  در فرکانسهای  $u$  و  $v$  دارند). پس می توان با مشاهده طیف فرکانسی تصویر، نقاط درخشانی را که محدوده فرکانسی خاصی دارند، حذف کرد.

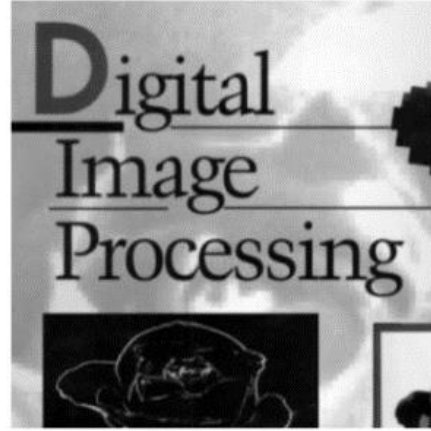
$$n(x, y) = A \sin(u_0 x + v_0 y) \quad (1)$$



- نویزهای حرکتی از حرکت دوربین یا شیء عارض می شود. این نویز را در حوزه فرکانس می توان بصورت تابع زیر مدل کرد. که با ضرب شدن در حوزه فرکانس و یا کانوالو شدن در حوزه مکان (زمان) با تصویر، اقدام به محو و خراب کردن آن می کنند.

$$H(u, v) = \frac{T}{\pi(ua+vb)} \sin[\pi(ua + vb)] e^{-j\pi(ua+vb)} \quad (2)$$

بنابراین می توان با دکانولوشن تصویر خراب شده، همراه با انتخاب مناسب پارامترهای زاویه و میزان حرکت، اثر آن را تخفیف بخشید.



- اکنون بمنظور ارزیابی کیفیت نتایجتان، از معیارهای<sup>۱</sup> SNR و<sup>۲</sup> MSE استفاده کنید. مقادیر را گزارش و بحث نمایید.

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f^{\wedge}(x,y)}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f^{\wedge}(x,y) - f(x,y)]^2} \quad (3)$$

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f_{original}(x,y) - f_{reconstructed}(x,y)]^2 \quad (4)$$

- پلاک خودرو موجود در تصویر چیست؟ و چه تعداد سرنشین در آن قرار دارند؟!

برخی دستورات مهم:

rgb2gray, fft2, fftshift, ifft2,

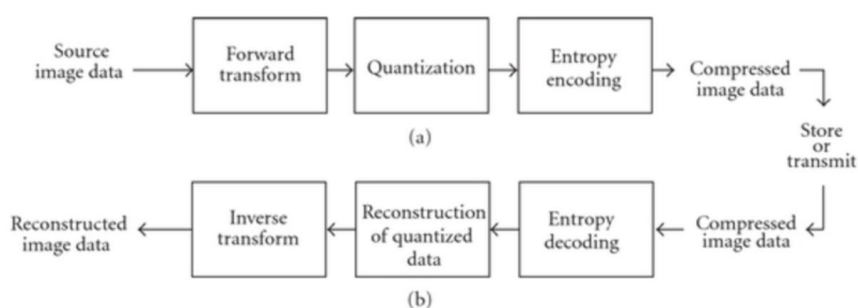
---

<sup>۱</sup>Signal to Noise Ratio

<sup>۲</sup>Mean Square Error

### مقدمه

در این بخش می‌خواهیم با مبحث فشرده‌سازی تصاویر<sup>۳</sup> آشنا شویم. فشرده‌سازی به فرآیندی گفته می‌شود که در آن حجم داده مورد نظر برای نمایش را کاهش می‌دهیم. به صورت ساده یک سیستم فشرده ساز به دو بخش کدکننده<sup>۴</sup> و کدگشا<sup>۵</sup> تقسیم می‌شود که بخش اول نقش فشرده‌ساز و بخش دوم نقش فشرده زدایی را بر عهده دارد. در شکل زیر بلوک دیاگرام یک سیستم فشرده ساز را مشاهده می‌کنیم:



بلوک دیاگرام یک سیستم فشرده ساز

در حالت کلی اگر خروجی کدگشا مشابه ورودی کدکننده باشد، این سیستم را بی‌اتلاف<sup>۶</sup> گوییم و در غیر اینصورت این سیستم دارای اتلاف<sup>۷</sup> خواهد بود، به معنی آن که بعد از کدگشایی داده، خروجی به دست آمده دارای تفاوت‌هایی با ورودی سیستم است. بخش‌های مختلف این بلوک دیاگرام به شرح زیر است:

- ۱- ابتدا داده ورودی از تبدیلی عبور می‌کند تا برای بخش بعدی فشرده سازی و حذف داده های زاید داده آماده شود. این بخش ممکن است اطلاعات زاید تصویر را نیز کاهش بدهد.
- ۲- در بخش دوم، داده از یک کوانتایزر<sup>۸</sup> عبور کرده و تعداد سطوح روشنایی آن را کاهش می‌دهد. در شکل زیر یک کوانتایزر یکنواخت را مشاهده می‌کنیم که داده ورودی را به سطوحی خاص در خروجی تبدیل می‌کند.

<sup>۳</sup>Image Compression

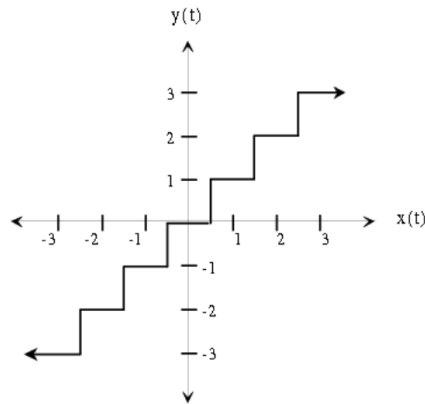
<sup>۴</sup>Encoder

<sup>۵</sup>Decoder

<sup>۶</sup>Lossless

<sup>۷</sup>Lossy

<sup>۸</sup>Quantizer



کوانتایزر یکنواخت هفت سطحی

همان طور که مشاهده می‌کنیم این کوانتایزر ورودی  $x(t)$  را به هفت سطح متفاوت در خروجی  $y(t)$ ، به صورت یکنواخت تبدیل می‌کند.

- ۳- در مرحله سوم داده از یک کدکننده عبور می‌کند تا مقادیر کوانتایز شده مرحله قبل به توالی از داده های دودویی تبدیل شود. روش های مختلفی برای کدینگ وجود دارد که از جمله آن می‌توان به الگوریتم کدکننده هافمن<sup>۹</sup>، الگوریتم کدکننده رنج<sup>۱۰</sup> و الگوریتم لمپل-زیو<sup>۱۱</sup> اشاره کرد. روش های موجود مقادیر خروجی کوانتایزر را به داده هایی با طول ثابت و یا متغیر تقسیم می‌کنند.
- ۴- بخش کدگشای تصویر نیز عکس عمل کدکننده را برای تصویر در هر بخش انجام می‌دهد و به ترتیب آن را دیکد کرده و دکوانتایز نموده و در نهایت از تبدیلی عکس تبدیل بخش اول عبور داده تا داده خروجی به دست آید.

در این بخش می‌خواهیم با الگوریتم فشرده سازی JPEG و شیوه پیاده سازی آن آشنا شویم. این الگوریتم ابتدا تصویر را به زیر تصویرهایی تقسیم کرده و بعد از انجام عملیاتی به نام تبدیل گسسته کسینوسی<sup>۱۲</sup>، این زیرتصویرها آن را به تعداد سطوح خاصی کوانتایز می‌کند. سپس با عبور از یک کدکننده آن را به صورت توالی از داده های باینری برای ذخیره سازی و یا انتقال آماده می‌کند. در مرحله کدگشایی نیز عکس این عمل را انجام می‌دهد و تصویر نهایی را نمایش می‌دهد.

تبدیل کسینوسی گسسته یک بعدی و عکس آن را در رابطه زیر مشاهده می‌کنیم:

<sup>۹</sup>Huffman Coding

<sup>۱۰</sup>Range Coding

<sup>۱۱</sup>Lempel-Ziv-Welch algorithm

<sup>۱۲</sup>Discrete Cosine Transform

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right); \quad k = 0.1 \dots N-1 \quad (5)$$

$$x_k = \frac{1}{N} X_0 + \sum_{n=1}^{N-1} \frac{2}{N} X_n \cos\left(\frac{\pi}{N}n\left(k + \frac{1}{2}\right)\right); \quad k = 0.1 \dots N-1 \quad (6)$$

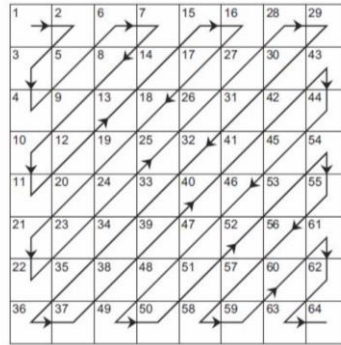
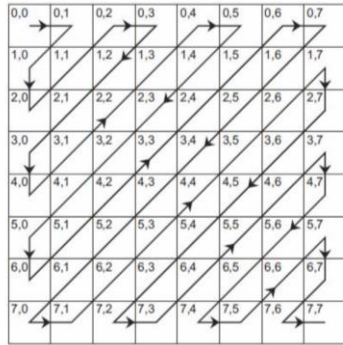
تبدیل گسسته کسینوسی دوبعدی که در الگوریتم JPEG مورد استفاده قرار می گیرد نیز به صورت زیر است:

$$X_{k_1, k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos\left(\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right) \cos\left(\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right) \quad (7)$$

## پیاده سازی

برای پیاده سازی این الگوریتم مراحل زیر را انجام داده و نتایج آن را در گزارش خود نشان دهید.

- ۱- تصویر به دست آمده از قسمت اول را به زیر تصویرهای  $8 \times 8$  تقسیم کنید. (در صورتی که ابعاد تصویر قابل تقسیم شدن به ۸ نیست، می توانید از دو طرف تصویر به گوشه های آن مقادیر صفر اضافه کنید تا قابل تقسیم به ۸ شود. به این عمل zero padding گویند.)
- ۲- در مرحله بعد برای هر بخش  $8 \times 8$  از تصویر شدت روشنایی درایه های آن را از ۱۲۸ کم کنید. (دقت کنید که بازه شدت روشنایی تصویر ۰-۲۵۵ باشد.)
- ۳- حال تبدیل گسسته کسینوسی دوبعدی را برای هر کدام از این تصاویر  $8 \times 8$  محاسبه کنید.
- ۴- مقادیر بدست آمده برای هر بخش از تصویر را به Q سطح کوانتایز کنید.
- ۵- داده های کوانتایز شده در بخش قبلی را به صورت یک توالی از داده ها تبدیل کنید. برای این کار می توانید از الگوی زیگ-زاگ<sup>۳</sup> که در شکل زیر آمده است، استفاده کنید.



الگوی زیگ-زاگ

۶- در این مرحله این توالی از داده ها را باید به صورت توالی از داده های باینری تبدیل کنید. برای این کار می توانید از الگوریتم کدکننده هافمن که الگوریتمی با طول متغیر بوده و داده ها را به داده های باینری با طول های متفاوت تقسیم می کند، استفاده کنید. با اتمام این بخش، قسمت کدینگ تصویر انجام شده است. (برای اطلاع از عملکرد این الگوریتم می توانید به مراجع اشاره شده رجوع کنید).

۷- برای بخش دیکدینگ تصویر باید عکس عملیات قبلی را تکرار کرده و داده اصلی را با تبدیل کد باینری به تصویر ۸\*۸، دکوانتایز کردن تصویر (ضرب در مقدار Q) و عکس تبدیل گسسته کسینوسی و اضافه کردن مقدار ۱۲۸ به شدت روشنایی تصویر، تصویر نهایی را بدست آورید.

۸- اختلاف تصویر اولیه و تصویر بدست آمده از بخش قبل را نمایش دهید و علت این اختلاف را توضیح دهید.

۹- میزان خطا، خطای کل، میانگین مربعات ریشه و میزان سیگنال به نویز تصویر بین تصویر اولیه و تصویر بدست آمده از بخش هفتم را به کمک یک تابع محاسبه کرده و در گزارش خود نتایج را نمایش دهید.

$$e(x,y) = f_{approx}(x,y) - f_{original}(x,y) \quad (8)$$

$$e_{tot} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f_{approx} - f_{original}) \quad (9)$$

$$e_{rms} = \frac{1}{MN} \sqrt{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f_{approx} - f_{original})^2} \quad (10)$$

برای آشنایی بیشتر با هر بخش از این الگوریتم می توانید از مراجع زیر استفاده کنید:

- 1- Digital Image Processing, Rafael Gonzalez, Richard Woods, chapter 8, image compression
- 2- JPEG Image Compression YouTube: <https://www.youtube.com/watch?v=aFbGqXFT0Nw>
- 3- Slide-Share JPEG Image Compression: <https://www.slideshare.net/AishwaryaKM1/jpeg-image-compression-56894348>

## قسمت سوم (امتیازی):

در قسمت قبل با مفاهیم فشرده سازی آشنا شدید. همچنین با یکی از معروفترین آن‌ها (JPEG) دست و پنجه نرم کردید! حال می‌خواهیم با تابع تبدیلی جدید آشنا شویم که به کمک آن می‌توان بسیاری از پردازش‌های سیگنال یا تصویر را براحتی انجام داد. برای اینکار از یکی از جعبه ابزارهای نرم افزار MATLAB بنام wavelet toolbox استفاده خواهیم کرد.

### موجک چیست؟

موجک یا Wavelet گروهی از توابع ریاضی هستند که جهت مولفه‌های فرکانسی و تجزیه سیگنالی بکار می‌رود که رزولوشن هر مؤلفه برابر با مقیاس آن است. تبدیل موجک به این معنا است که تجزیه یک تابع بر مبنای توابع موجک انجام شود. موجک‌ها نمونه‌های مقیاس شده و انتقال یافته یک تابع با طول متناهی و نوسانی شدیداً میرا هستند.

تولباکس Wavelet محیط محاسباتی MATLAB را با تعدادی ابزار گرافیکی و توابع خط فرمان برای توسعه الگوریتم‌های مبتنی بر موجک برای تحلیل، ایجاد، حذف نویز و فشرده سازی سیگنال‌ها و تصاویر ارائه می‌دهد. تحلیل موجک، اطلاعات دقیق‌تری را در مورد داده‌ی سیگنال نسبت به سایر سیگنال‌ها نظیر فوریه فراهم می‌کند. جعبه ابزار موجک برای پردازش صوت و گفتار، پردازش تصویر، پردازش ویدئو، تصویر برداری پزشکی و کاربرد های یک بعدی در مخابرات و ژئوفیزیک مناسب است.<sup>۱۴</sup>

جهت آشنایی بیشتر و کامل‌تر می‌توانید به آدرس‌های زیر مراجعه کنید:

<https://www.youtube.com/watch?v=ZnmvUCtUAEE>

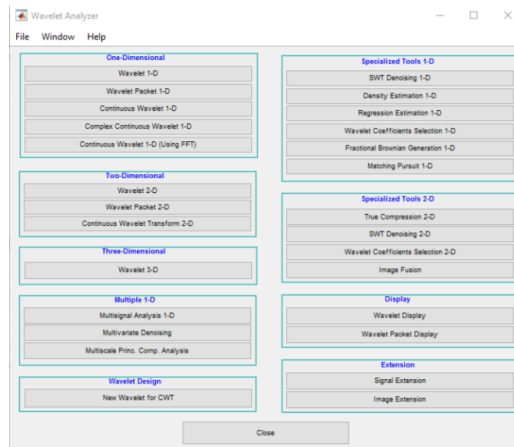
<https://www.mathworks.com/products/wavelet.html>

<sup>۱۴</sup><https://matlabkar.com/wavelet-toolbox-in-matlab/>

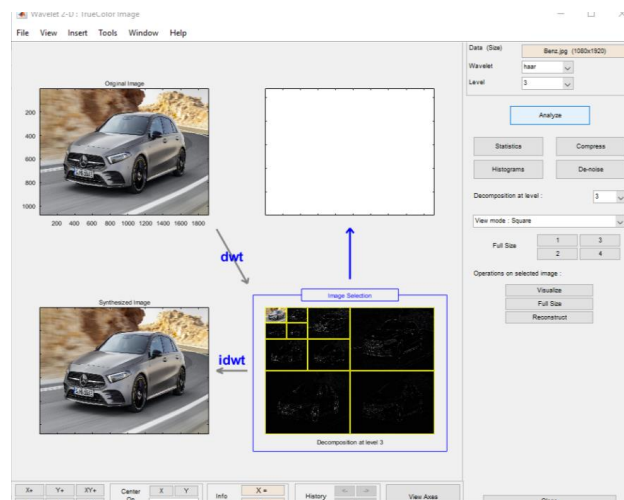


<https://www.mathworks.com/help/wavelet/ug/wavelet-data-compression.html>

- با تایپ کردن عبارت wavemenu در محیط command نرم افزار MATLAB وارد تولباکس ویولت می شوید.



- جهت پردازش تصویر (سیگنال دو بعدی) گزینه Wavelet 2-D را انتخاب کنید. سپس تصویر مورد نظر را بارگذاری کرده و با انتخاب نوع ویولت و سطح آن اقدام به آنالیز تصویر نمایید.



- حال با انتخاب گزینه Compress و تعیین سطح آستانه ها (بصورت کلی یا جزئی) تصویر ورودی را فشرده کنید. این کار را با ویولت های مختلف در سطح های متفاوت و همچنین آستانه های متنوع تکرار کنید. نتایج را مشاهده و مقایسه نمایید.

\* جهت ذخیره تصویر خروجی می‌توانید از مسیر File > Save > Compressed Image اقدام کنید.

برای کسب اطلاعات بیشتر پیرامون تولباکس معرفی شده، می‌توانید به فایل wavelet Ug.pdf در پوشه پروژه مراجعه نمایید.

## نکات مهم:

۱. فایل گزارش خود را با فرمت pdf، به انضمام کدهای MATLAB خود در قالب یک فایل zip تا زمان تحویل که ۱۴ تیرماه می‌باشد، در سایت درس با فرمت زیر بارگذاری کنید:

[Name]\_[student number]\_Project[Project number].zip

۲. علاوه بر گزارش، تحویل حضوری نیز برای پروژه‌ها وجود دارد که تاریخ تحویل حضوری پروژه‌ها ۱۵ تیر ماه می‌باشد. زمان ارایه هر گروه نیز متعاقباً اعلام خواهد شد.

توجه داشته باشید که توضیحات کافی به همراه نتایج شبیه سازی خود را در گزارش کار خود بیاورید. گزارش کار لازم است فرمت یک گزارش علمی داشته باشد. از گرفتن عکس از نوشته های خود و الصاق آن در گزارش خود خودداری کنید. یک تمپلیت برای گزارش در سایت درس آپلود شده است.

۳. می‌توانید پرسش‌های خود را از طریق ایمیل‌های زیر مطرح کنید.

[Behrang.fazli1990@gmail.com](mailto:Behrang.fazli1990@gmail.com)

[Mehdi.bashiri.b@gmail.com](mailto:Mehdi.bashiri.b@gmail.com)

۴. کپی کردن کار دیگران تخلف محسوب می‌شود و در صورت مشاهده کمترین تخلف، نمره کسب شده میان طرفین تقسیم خواهد شد.

شاد باشید ☺