



دانشکده مهندسی برق و کامپیوتر

سیستم‌های چندرسانه‌ای پیشرفته

پاسخ تمرین سری ۱ و ۲

نام و نام خانوادگی دانشجو: قوام سلیمانی

پاییز ۱۴۰۲

فهرست مطالب

۱	پاسخ سوال اول
۲	۱.۱ کاربرد هر Format
۲	۲.۱ ساختار Header و نحوه ذخیره‌سازی
۴	۳.۱ نحوه ذخیره‌سازی داده‌ها
۴	۲ خواندن PPM در Matlab بدون imread
۷	۳ سوال هشتم

۱ پاسخ سوال اول

[۳] [۲] [۱]

۱.۱ کاربرد هر Format

همه این فایل‌ها به طور گسترده برای ذخیره و نمایش تصاویر در برنامه‌های گوناگون براساس بیت مپ‌های ساده بلافاصله پس از Header مورد استفاده قرار می‌گیرند که به علت این ساختار ساده، قابل حمل نامیده می‌شوند و به راحتی در دستگاه‌های مختلف مورد استفاده قرار می‌گیرند. در ادامه به صورت جداگانه بررسی می‌شوند.

۱. PPM

فایل‌های PPM تصاویر تمام رنگی یا RGB (قرمز، سبز، آبی) را نمایش می‌دهند. آنها اطلاعات رنگ هر پیکسل را به صورت جداگانه ذخیره می‌کنند. فرمت PPM اغلب در زمینه‌هایی مانند ویرایش تصویر و گرافیک کامپیوتری استفاده می‌شود. این فرمت از طیف گسترده‌ای از رنگ‌ها پشتیبانی می‌کند و فقط به تصاویر سیاه و سفید یا باینری محدود نمی‌شود.

۲. PGM

فرمت PGM برای تصاویر خاکستری استفاده می‌شود. هر پیکسل را با یک مقدار عددی جهت نمایش رنگ خاکستری، (از سیاه تا سفید) نشان داده می‌شود. فایل‌های PGM معمولاً در الگوریتم‌های پردازش تصویر و OCR استفاده می‌شوند که در آن اطلاعات رنگ مورد نیاز ما نخواهد بود یا تمرکز بر شدت روشنایی است.

۳. PBM

فرمت PBM برای تصاویر سیاه و سفید یا باینری طراحی شده است. هر پیکسل، سیاه یا سفید را نشان می‌دهد. فایل‌های PBM اغلب در برنامه‌هایی استفاده می‌شوند که به نمایش‌های تک رنگ ساده نیاز دارند، مانند دستگاه‌های فکس و چاپگرهای سیستم‌های OCR

۲.۱ ساختار Header و نحوه ذخیره‌سازی

۱. PPM

هدر یک فایل PPM با مقدار "P6" برای تصاویر باینری و "P3" برای تصاویر اسکی درنظر گرفته می‌شود و سپس عرض و ارتفاع تصویر بر حسب پیکسل و حداکثر مقدار رنگ (از ۰ تا ۲۵۵ برای هر کانال) شروع می‌شود. پس از هدر، داده‌های تصویر به صورت سه گانه RGB ذخیره می‌شوند (۱۶۷۷۷۲۱۶ رنگ). مثال :

P3

```
# Comment (Pixel storage: r1 g1 b1 r2 g2 b2 r3 g3 b3 ...)
4 4
15
0 0 0 0 0 0 0 0 0 15 0 15
0 0 0 0 15 7 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 15 7 0 0 0  
15 0 15 0 0 0 0 0 0 0 0 0
```

PGM .۲

هدر یک فایل PGM با مقدار "P5" برای تصاویر باینری و "P2" برای تصاویر اسکنی درنظر گرفته می شود و سپس عرض و ارتفاع تصویر بر حسب پیکسل و حداکثر مقدار رنگ خاکستری (از ۰ تا ۲۵۵) شروع می شود. پس از هدر، داده های تصویر به صورت مقادیر پیکسل های خاکستری ذخیره می شوند. مثال :

```
P2  
# Comment  
24 7  
15  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 15 0  
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0  
0 3 3 3 3 0 0 0 7 7 7 7 0 0 0 11 11 11 11 0 0 15 15 15 15 15 0  
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0  
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0  
0 0
```

PBM .۳

هدر یک فایل PBM با مقدار "P1" برای تصاویر سیاه و سفید باینری درنظر گرفته می شود که تنها دو رنگ سیاه و سفید را نشان می دهد و سپس عرض و ارتفاع تصویر بر حسب پیکسل درج می شود. پس از هدر، داده های تصویر به صورت پیکسل های سیاه و سفید ذخیره و نمایش داده می شوند. مثال :

```
P1  
# Comment  
6 10  
0 0 0 0 1 0  
0 0 0 0 1 0  
0 0 0 0 1 0  
0 0 0 0 1 0  
0 0 0 0 1 0  
0 0 0 0 1 0  
1 0 0 0 1 0  
0 1 1 1 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0
```

۳.۱ نحوه ذخیره سازی داده ها

۲ خواندن PPM در Matlab بدون imread

```

MyFile = fopen('Test_3.ppm', 'r');

% Type checking
Magic_num = fscanf(MyFile, '%s', 1);
if ~strcmp(Magic_num, 'P6')
    error('Your file is incorrect!');
end

% Get file header
W = fscanf(MyFile, '%d', 1);
H = fscanf(MyFile, '%d', 1);
MaxVal = fscanf(MyFile, '%d', 1);

% Read and Reshaping
Pixels = uint8(fread(MyFile, H * W * 3, 'uint8'));
Reshaped = permute(reshape(Pixels, [3, W, H]), [3, 2, 1]);

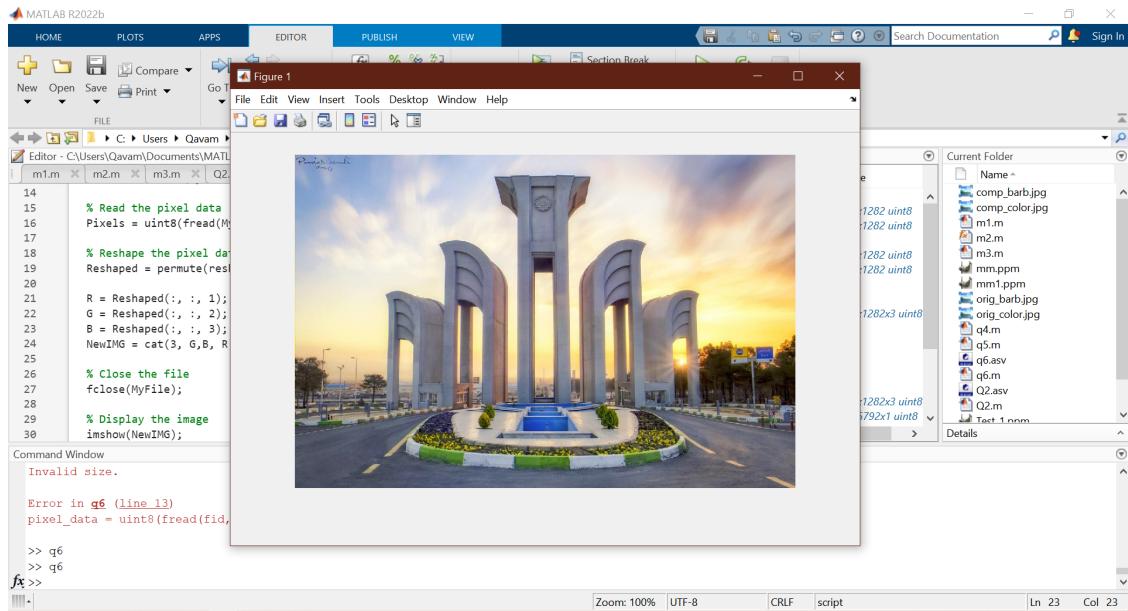
%Change channels:
R = Reshaped(:, :, 1);
G = Reshaped(:, :, 2);
B = Reshaped(:, :, 3);
NewIMG = cat(3, G, B, R);

imshow(NewIMG);

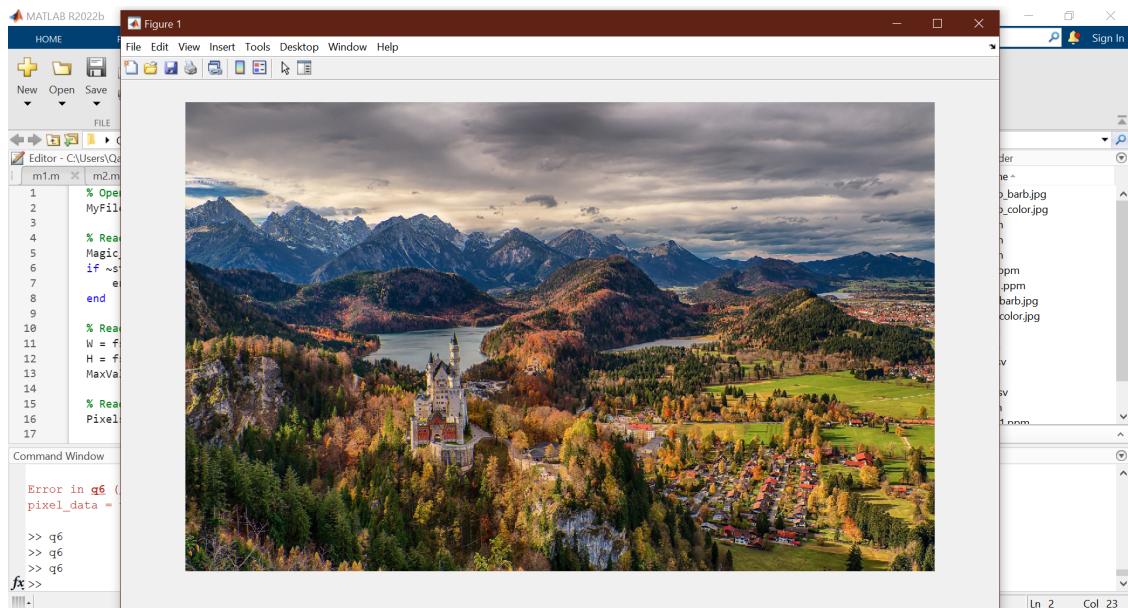
```

پس از بازگشایی فایل PPM و چک کردن نوع، باید Header های فایل را بخوانیم. بعد از آن با توجه به اینکه به تعداد (عرض * ارتفاع * کانال های رنگی) پیکسل داریم، داده ها را خوانده و ضمنا برای آن عدهها از نوع بدون علامت ۸ بیتی صحیح استفاده می کنیم که برای این کار از fread بهره بردیم. سپس مقادیر خوانده شده برای نمایش در حالت استاندارد، بعد اولشان نشان دهنده ای کانال های رنگی بوده سپس ارتفاع و بعد عرض. پس آن را ReShape کردیم. رنگ ها به هم ریخت، پس کانال های سبز و آبی را نیز جا به جا کردیم.

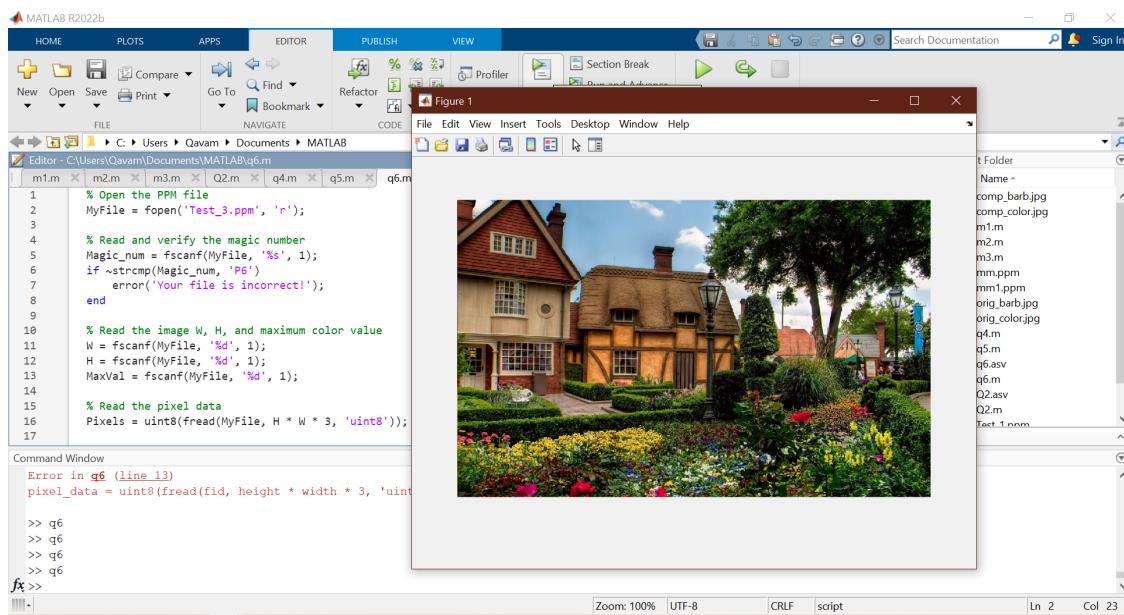
نتیجه:



شکل ۱:۱



شکل ۱:۲



شکل ۱:۳

۳ سوال هشتم

File Name		Text.txt	Scenery.bmp	Angio.bmp
Entropy				
Coding Redundancy				
Unary Code	Compressed Data Size (BPS)			
	Dictionary Size (BPS)			
	Total Size (BPS)			
	Compression Ratio			
	Encoding Time			
	Decoding Time			
	MSE			
	PSNR			
Proposed Method	Compressed Data Size (BPS)			
	Dictionary Size (BPS)			
	Total Size (BPS)			
	Compression Ratio			
	Encoding Time			
	Decoding Time			
	MSE		56	
	PSNR			

منابع

- [1] <https://en.wikipedia.org/wiki/Netpbm>
- [2] <https://paulbourke.net/dataformats/ppm/>
- [3] <https://prasannk.tripod.com/imgformats.htm>