

# 解馬賽克方法的實現

## Implementation of a demosaicing method

指導教授：陳培殷

專題成員：許家豪

開發工具：Python、CV2

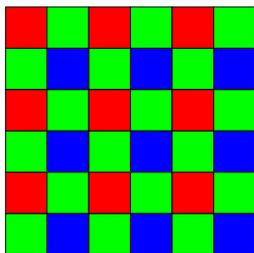
測試環境：Python3.7

### 一、簡介：

大部分的數位相機為了控制成本，通常只用單層的 CCD 感光元件來記錄照片資訊，但是這樣只能記錄單一維度的資訊（明暗），因此會需要透過色彩濾波陣列來分割色塊，以此來記錄各種顏色的明度，這樣一來每個像素只需儲存單個通道的資訊，可以大幅降低儲存成本。拜爾圖形（Bayer Pattern）則是現今主要採用的濾色鏡格式之一，將2\*2的像素分割成1R2G1B 的格式很大程度模仿人類的肉眼，這次研究的會以 RGGGB 格式的拜爾圖形做基底。（圖一）

然而這樣儲存的圖片要還原到彩色影像需要經過解馬賽克的演算法，不同的演算法所需要的成本差異甚大，雖然現在已經存在許多效果極好的演算法，但是效果越好的演算法通常也代表成本越高，對軟硬體配置不夠高的裝置來說可能是難以負荷的，因此想要找到一個效果在可接受範圍且成本相對較低的演算法。

峰值訊噪比（PSNR）是透過均方誤差（MSE）來衡量圖像或視訊品質的常用指標之一（圖二），在30dB 以上通常肉眼就很難察覺處理前後圖像的差距了，接近50dB 則代表還原圖像僅有非常小的誤差了。這次研究將會以此數值為判斷演算法好壞的依據。



圖一：拜爾圖形濾色陣列排列格式（RGGGB）

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$$

圖二：PSNR 計算公式

## 二、測試結果：

參考雙線性插值法（Bilinear）和有效色彩插值法（Effective Color Interpolation, ECI）等較低成本的演算法，將其中的步驟分離後分段進行插值，處理 R、B 像素時參考一開始補出的 G 平面，後面處理 G 像素時再參考先前補好的 R、B 平面，在僅增加幾個加法的成本下（對比 ECI），即可對 R、B 的數值進行更加準確的估計。

測試集的部分選用柯達的24張全彩無失真照片集，結果顯示出小部分樣本的紅色通道會表現比原本方法（ECI）差，但是對於單張圖片的三個通道的平均以及整體平均來說，表現確實是有所進步的。（圖三）

		ECI	新方法	前後差距		R	39.07892	39.45059	0.371676		R	39.57571	40.15251	0.576799
	R	33.33648	33.76403	0.427552	9	G	41.57719	41.57719	0	17	G	40.6378	40.6378	0
1	G	35.54461	35.54461	0		B	38.87098	39.13319	0.26221		B	38.63065	39.04688	0.416233
	B	33.1456	33.65818	0.512586		R	39.98096	40.28622	0.305258	18	R	35.27386	35.77122	0.497353
	R	38.68537	38.29937	-0.386	10	G	42.23122	42.23122	0		G	36.61028	36.61028	0
2	G	40.62461	40.62461	0		B	39.34242	39.64827	0.305855		B	34.38003	34.80538	0.425348
	B	38.68526	39.03762	0.352362		R	35.95378	36.27139	0.317609	19	R	37.25658	37.25658	0
	R	40.79647	40.83816	0.041684	11	G	37.84188	37.84188	0		G	34.73034	35.13827	0.407938
3	G	42.78314	42.78314	0		B	35.76317	36.24283	0.47966		B	34.56656	34.97099	0.404435
	B	39.28229	39.3717	0.089413		R	39.81849	40.0088	0.190311	20	R	39.01167	39.48353	0.471857
	R	38.18889	37.99094	-0.19795	12	G	42.29892	42.29892	0		G	40.44475	40.44475	0
4	G	42.03361	42.03361	0		B	39.17145	39.45823	0.286781		B	37.01842	37.32402	0.305597
	B	40.91716	41.45621	0.539048		R	31.37157	32.07412	0.702557	21	R	35.66246	36.14346	0.481
	R	35.55055	36.01829	0.467733	13	G	32.5996	32.5996	0		G	37.49293	37.49293	0
5	G	36.72205	36.72205	0		B	30.50839	31.07803	0.569644		B	34.78695	35.20641	0.419462
	B	34.18244	34.53968	0.35724		R	35.91171	35.74592	-0.1658	22	R	36.84102	36.98688	0.145856
	R	34.6661	35.10767	0.441575	14	G	37.48252	37.48252	0		G	38.47865	38.47865	0
6	G	36.91618	36.91618	0		B	34.09659	34.24778	0.151191		B	35.23415	35.44828	0.214127
	B	34.11352	34.53042	0.4169		R	38.66389	38.43575	-0.22814	23	R	42.1087	41.92508	-0.18361
	R	40.97157	41.13396	0.162386	15	G	40.56231	40.56231	0		G	43.38085	43.38085	0
7	G	41.91431	41.91431	0		B	38.40437	38.75419	0.349823		B	40.04948	40.16018	0.110704
	B	38.9378	39.09335	0.155551		R	37.51098	37.90767	0.396688	24	R	34.22078	34.60211	0.381338
	R	30.26981	30.63228	0.362474	16	G	40.02589	40.02589	0		G	35.42867	35.42867	0
8	G	33.07614	33.07614	0		B	37.08966	37.46641	0.376747		B	31.84144	32.18136	0.33992
	B	29.78254	30.15136	0.368827						整體平均	37.3743	37.5714	0.197192	

圖三：實驗結果（左：1~8 中：9~16 右：17 - 24 & 整體平均）



圖四：演算法處理前後比較圖（左：原圖 中：ECI 右：新方法）