Understanding ISP Pipeline - Demosaicking

刘斯宁
Camera技术专家

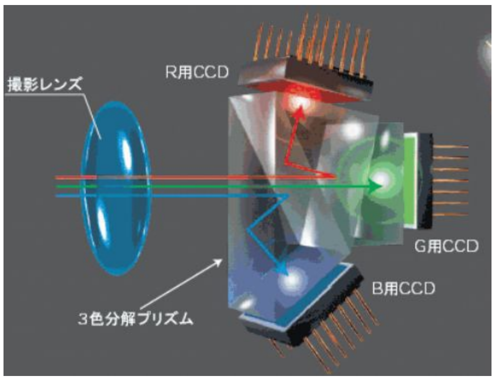
已关注

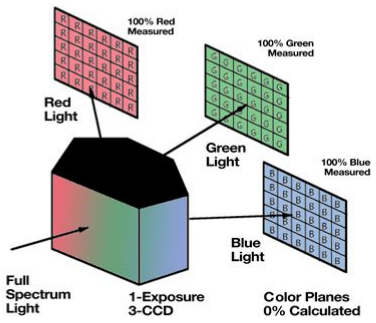
73 人赞同了该文章

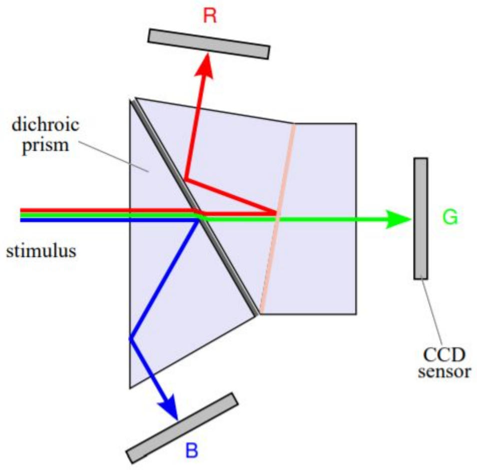
简介

人眼种存在三种感光锥细胞，分别对红、绿、蓝三种波段敏感。因此描述一个人眼可见的颜色需要且仅需要三个分量，这三个分量能够支撑起一个颜色空间，该空间包含了人眼能够知觉的所有颜色。

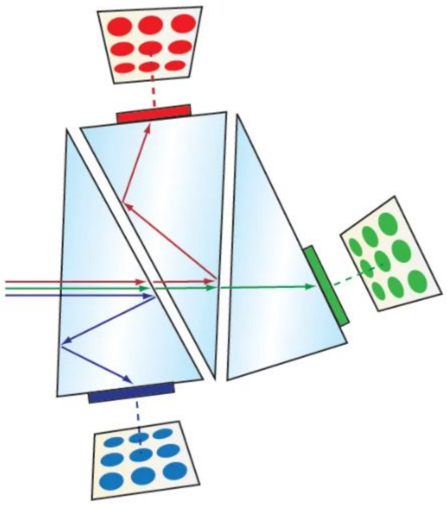
仿照人眼的原理人们设计了3CCD成像方案，如下图所示。这个方案的核心是一个光学分光棱镜，它的使命是将全色谱的入射光线分解成红绿蓝三个波段，分别投射到三个CCD阵列上。





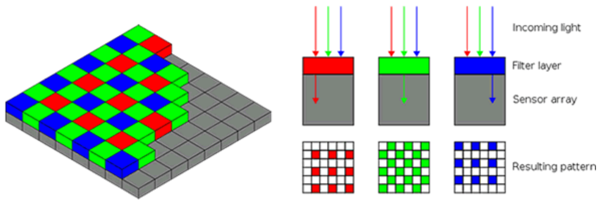


3CCD分光棱镜原理



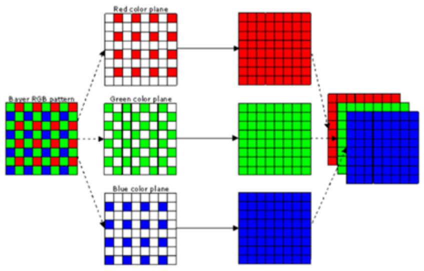
3CCD方案成像效果非常好，但是存在技术复杂、成本高的缺点。光学棱镜本身成本高昂，棱镜与CCD之间必须保证微米级别的位置关系，而且需要在长期使用过程中不能发生变形移位，这么高的要求是很难实现大规模量产的，所以目前普通消费类市场上已经很少看到3CCD产品了。

好在柯达公司的科学家Bryce Bayer（1929-2012）发明了一个突破性的解决方案。这个方案不需要使用昂贵的光学棱镜，也不需要使用3个CCD阵列，只需要在一个CCD阵列上制造三种不同的滤光膜，构成一个滤光膜阵列（Color Filter Array，CFA），就形成一个廉价而高效的解决方案。



这种方案称为一般称为Bayer CFA格式，或者Bayer mosaic格式，支持这种格式的成像器件（包括CCD和CMOS两种）输出的数据格式俗称RAW格式，特点是每个像素只提供RGB三种颜色中的一种，另外两种颜色需要后续处理环节中通过一定的软件算法进行插值补全。RAW格式的主要优点是节省传输带宽，降低硬件成本，对相机、摄像机产品的大规模普及起到了极为重要的推动作用。在相机、摄像机产品中一般会使用专用的ISP硬件负责颜色插值，以支持海量数据的高速实时处理。现在也有越来越多的摄影爱好者和专业影视从业者倾向于前期只录制RAW格式文件，后期用电脑软件进行编辑制作。

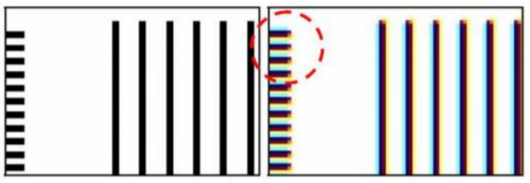
对RAW数据进行插值的过程叫做Bayer demosaicking，其原理如下图所示。



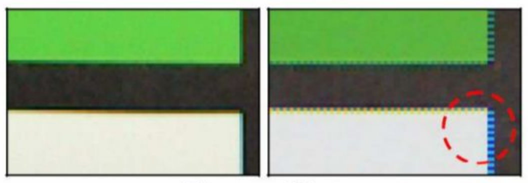
天下没有免费的午餐，RAW格式省带宽省成本的代价是丢弃了2/3的数据量。就像人死不能复生一样，数据一旦被丢弃就再也没人知道它们到底是多少——但是人们还可以猜，图像信号一般都会存在大量的冗余，相邻像素间的差异一般不大，所以很多时候人们都能猜个八九不离十。这个过程在数学上叫做**插值**（interpolation），就是根据已知的数据去推测未知的数据。插值问题在数学上是一种不适定问题（ill-posed problem），理论上有无穷多种方法，因此与其说是一种科学，不如说是一种艺术。

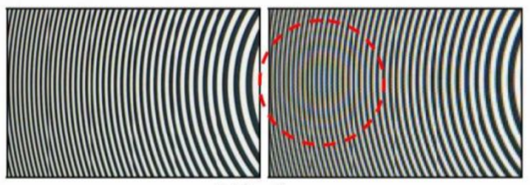
俗话说常在河边走，没有不湿鞋的，虽然猜对的时候有十之八九，但必然也会有猜错的时候，一旦猜错，就会给图像引入各种各样的问题。一般而言，Bayer demosaicking插值算法都会存在以下几个方面的技术挑战：

- 边缘保持，插值算法容易导致图像中的边缘变模糊，而人眼对边缘特别敏感；
- 抗伪彩，插值算法如果预测颜色不准就会在图像中引入伪彩，在灰色区域尤其明显。
- 抗拉链，插值算法如果预测颜色不准容易在图像中引入周期性的模式
- 抗混叠，插值算法容易在图像中引入低频模式


False color

äller ej
don me
Blurring


Zipper effect

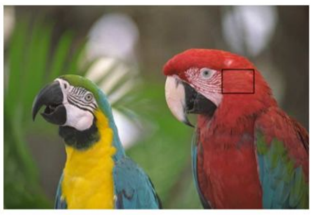

Aliasing

所谓“边缘”就是英文中的edge，指的是图像中亮度急剧变化的地方，而亮度发生变化通常是因为图像所表式的物体材质、颜色等基本属性发生了变化。边缘可以理解为一条几何曲线的片段，在边缘上的点会有切线和法线等两个特殊方向，沿着切线移动像素的亮度一般变化较小，而沿着法线移动则像素亮度会剧烈变化。

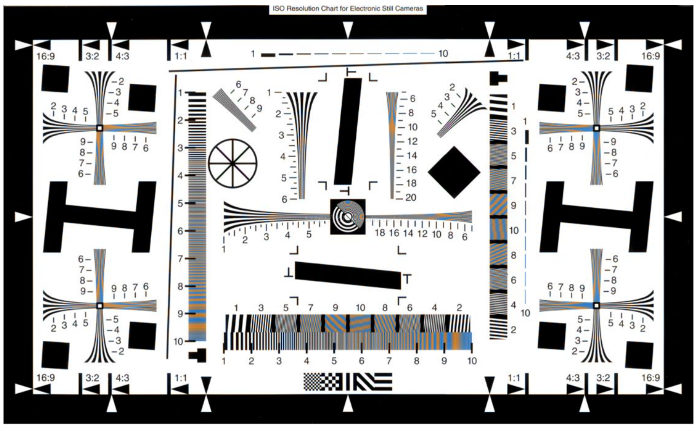
综述

自从Bayer CFA技术发明以来，人们对demosaicking相关算法进行了大量的研究，直到今天仍不时有相关文献出现。大量的文献侧重于基于边缘检测技术实现方向性插值，也有很多文献研究基于频域的算法，包括基于小波理论的算法。

基于边缘检测技术实现方向性插值是被研究的最多的一个技术方向，基本思路是在经典低通滤波器的基础上增加边缘检测机制。常用的经典低通滤波器有中值滤波、高斯滤波、双线性滤波，它们的共同优点是技术简单，对低频信号滤波效果好，共同的缺点是容易破坏图像中的边缘，因为边缘属于高频信号，会被低通滤波器截断。


(a) Original Image.

（a）原图 （b）原图细节 （c）双线性滤波 效果 （d）保边滤波效果



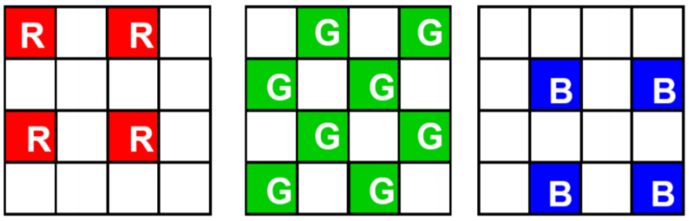
双线性插值，引入伪彩

1987年Cok探索了图像各颜色通道之间的相关性，并提出了色比色差定律，即在图像中的平滑区域内，（R/G）和（B/G）两个参数应该是连续变化的，并且（R/G）和（B/G）的比值应该是相等的。

Adams和Hamilton 提出了利用亮度的梯度和色度的二阶微分来检测边缘的方向，沿着边缘的方向进行插值，避免跨边缘插值。这种方法实践中效果不错，可以有效地保护边缘，已经得到了大量的应用。

> Adams J E, Hamilton J F. Adaptive color plane interpolation in single color electronic camera[J]. Us Patent No, 1997.

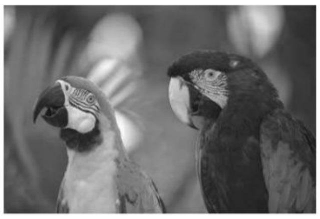
Adams&Hamilton方法先对G分量进行插值，因为CFA中G像素的数量是R/B的两倍，插值错误的概率最小，因此插值后的G分量图像能够更准确地反映图像中的轮廓。



用G分量检测边缘准确率更高

下图的例子能够说明G通道图像往往是原图的一个相当好的近似。

(a) Original image


(b) *G* plane

在现实生活中，G图像上检测到的边缘往往是两种材质的分界线，而两种不同的材质只有G不同但RB都相同这种情况是极为罕见的。因此，在G图像上检出的边缘可以安全地认为在RB分量上也同样存在。接下来的工作就顺理成章了，对于待插值像素R(i,j)，插值时应参考像素G(i,j)处的边缘检测结果，基本规则是，

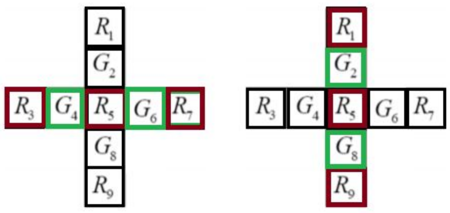
- 如果G(i,j)处检出垂直的边缘，则选用R(i,j)上、下的邻近像素R(i,j-1)和R(j,j+1)进行插值，不使用左右邻近像素；

- 如果G(i,j)处检出水平的边缘，则选用R(i,j)左、右的邻近像素R(i-1,j)和R(i+1,j)进行插值，不使用上下邻近像素；

- 如果G(i,j)处未检出任何边缘，则使用R(i,j)周围的所有像素进行插值。

- 色差恒定，即相邻点的R(i,j)-R(i, j+1) = G(i,j)-G(i, j+1)

有人提出，对G的插值也可以参考其它颜色通道的信息，以提高对G通道边缘的准确率。



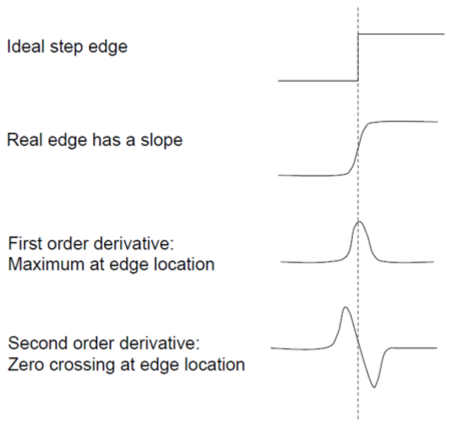
$$\boxed{\begin{aligned}\alpha &= |R_3 - 2\cdot R_5 + R_7| + |G_6 - G_4| \\ \beta &= |R_1 - 2\cdot R_5 + R_9| + |G_8 - G_2|\end{aligned}}$$

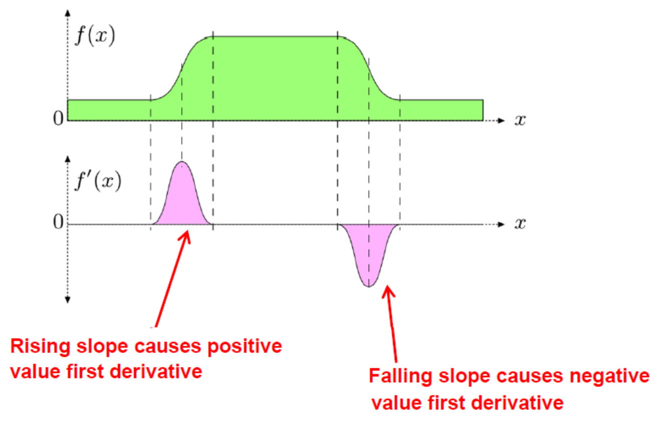
$$\boxed{\begin{aligned}G_5 &= \frac{G_4+G_6}{2} - \frac{R_3 - 2\cdot R_5 + R_7}{2} \quad , if\ \alpha < \beta \\ G_5 &= \frac{G_2+G_8}{2} - \frac{R_1 - 2\cdot R_5 + R_9}{2} \quad , if\ \alpha > \beta \\ G_5 &= \frac{G_2+G_4+G_6+G_8}{4} - \frac{R_1 + R_3 - 4\cdot R_5 + R_7 + R_9}{4} \quad , if\ \alpha = \beta\end{aligned}}$$

上面所述方法的核心思想是使用G分量图像指导RB分量图像的插值过程，因此G分量图像插值的质量就起到了决定性的作用。虽然上述方法在大多数场景下都能收到不错的效果，但仍然存在一些复杂度较高的场景，由于图像中纹理密集，G像素的插值本身就会出现错误，导致误判边缘，RB插值也连带受到影响。因此有人提出可以对G像素的插值过程加以改进，一种方法是在水平和竖直两个方向上分别对G进行插值，得到两个插值结果，然后根据一定的标准挑选出更合理的候选者。一种常用的判断标准是基于同一物体颜色相似的原理：如果插值方向是正确的，那么沿着正确边缘方向上的像素应属于同一物体，颜色的梯度应较小。如果插值方向是错误的，则在错误边缘上的像素将属于不同物体，颜色的梯度应较大。这种先插值再校验的方法称为“后验”方法，已经得到了大量的研究和应用。

边缘检测算子

基本原理是利用一阶微分和二阶微分，求图像的变化量，以及变化量的趋势。




$f(x)$
0
x
$f'(x)$
0
x
Rising slope causes positive value first derivative
Falling slope causes negative value first derivative

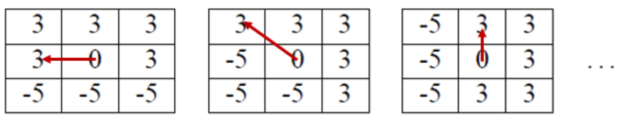
Roberts Kernels

f为图像

特点：使用同一组数据构造2个kernel，分别检测45度和135度上是否存在边缘。

Roberts 算子使用的kernel太小，在图像有噪声的情况下效果不佳。

Kirsch Compass Kernels



3	3	3
3	0	3
-5	-5	-5

3	3	3
-5	0	3
-5	-5	3

-5	3	3
-5	0	3
-5	3	3

. . .

特点：使用同一组数据构造8个kernel，分别检测0，45，90，... 等角度上是否存在边缘。

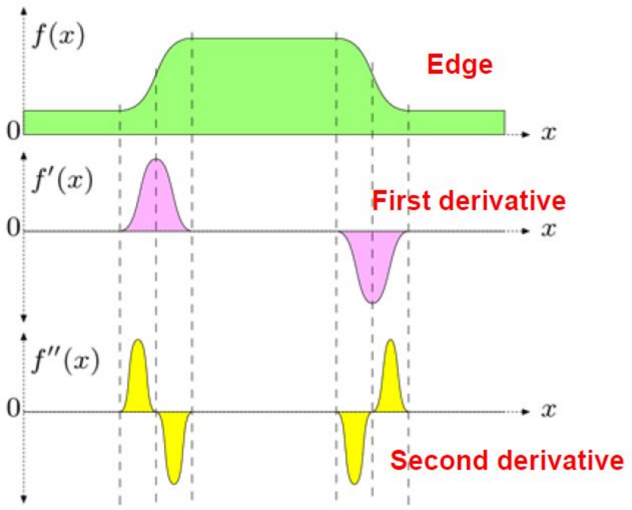
Prewitt Kernels

Sobel Kernels

特点：使用同一组数据构造2个kernel，分别检测0度和90度上是否存在边缘。

Laplacian Operators

$$
\begin{aligned}
\frac{d^2 f}{dx^2} &\approx [f(x+1)-f(x)]-[f(x)-f(x-1)] \\
&= f(x+1)-2f(x)+f(x-1)
\end{aligned}
$$



$$
\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\\hline 1 & -2 & 1 \\\hline 0 & 0 & 0 \\\hline\end{array}
+
\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\\hline 0 & -2 & 0 \\\hline 0 & 1 & 0 \\\hline\end{array}
=
\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\\hline 1 & -4 & 1 \\\hline 0 & 1 & 0 \\\hline\end{array}
$$

基本形式

$$
\begin{array}{|c|c|c|}\hline 0.5 & 0.0 & 0.5 \\\hline 1.0 & -4.0 & 1.0 \\\hline 0.5 & 0.0 & 0.5 \\\hline\end{array}
+
\begin{array}{|c|c|c|}\hline 0.5 & 1.0 & 0.5 \\\hline 0.0 & -4.0 & 0.0 \\\hline 0.5 & 1.0 & 0.5 \\\hline\end{array}
=
\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\\hline 1 & -8 & 1 \\\hline 1 & 1 & 1 \\\hline\end{array}
$$

变种

$$
\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\\hline 1 & -2 & 1 \\\hline 1 & -2 & 1 \\\hline\end{array}
+
\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\\hline -2 & -2 & -2 \\\hline 1 & 1 & 1 \\\hline\end{array}
=
\begin{array}{|c|c|c|}\hline 2 & -1 & 2 \\\hline -1 & -4 & -1 \\\hline 2 & -1 & 2 \\\hline\end{array}
$$

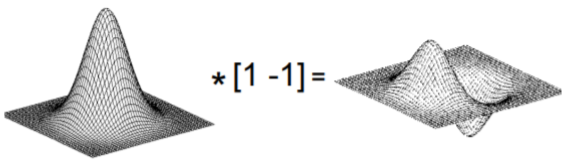
变种

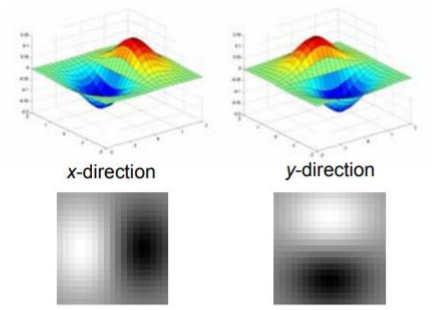
Laplacian of Gaussian Operators （LoG)

$$G(x,y)=e^{-\frac{x^2+y^2}{2\sigma^2}}$$

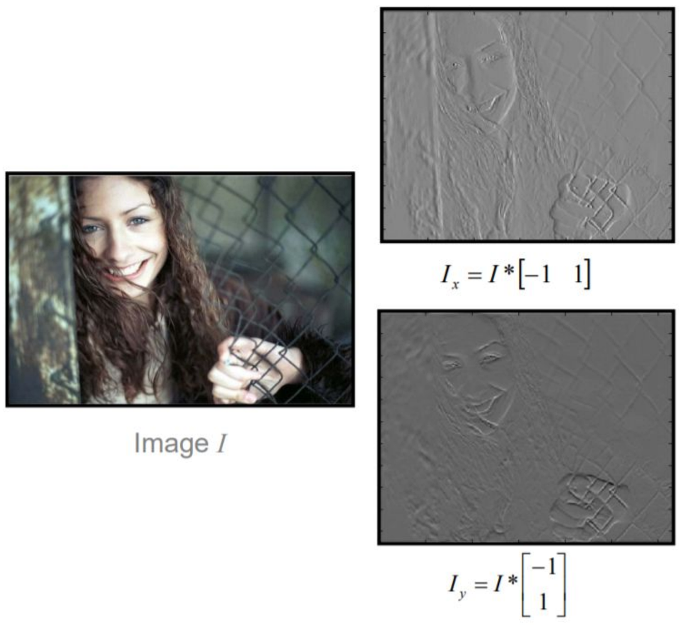
$$\nabla^2 G(x,y)=\frac{\partial^2 G}{\partial^2 x}+\frac{\partial^2 G}{\partial^2 y}=\frac{x^2+y^2-2\sigma^2}{\sigma^4}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

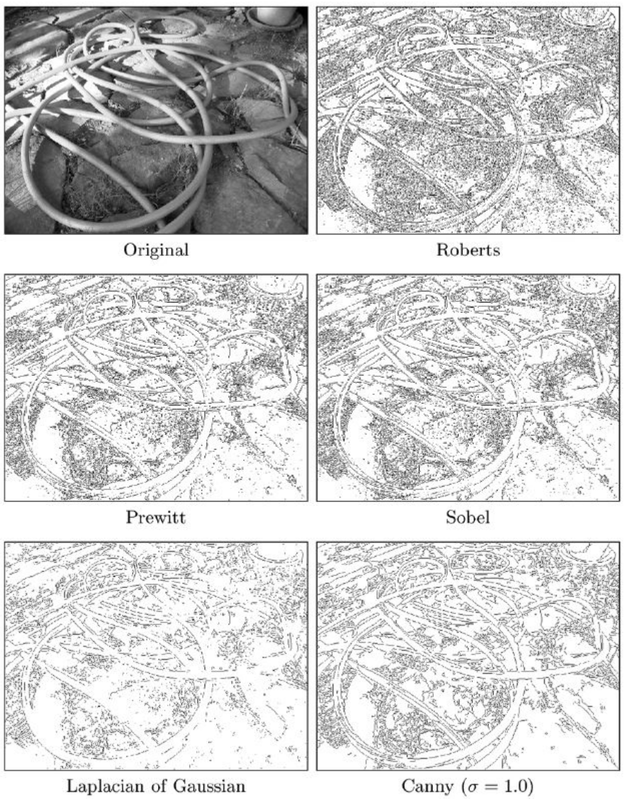
Derivative of Gaussian Operators （DoG）





下图显示的是对一幅图像分别在x 和 y 两个方向上进行微分得到的边缘图像。





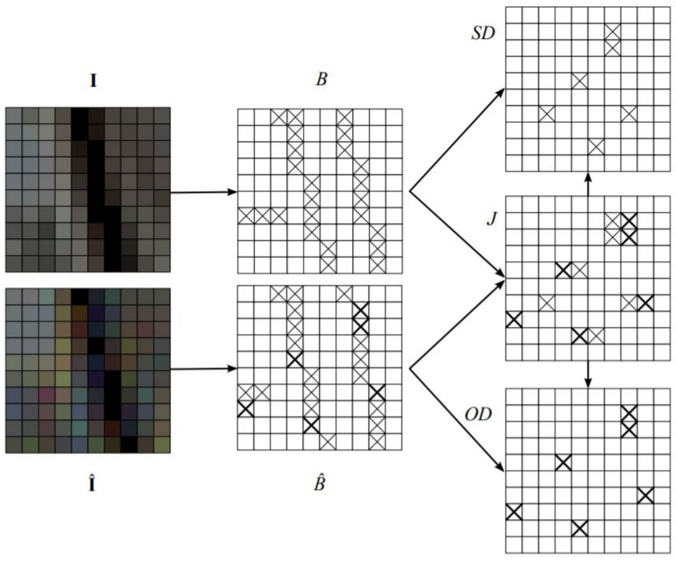
误差评估

定量评估demosaic算法效果一般用MAE（mean absolute error），MSE（mean square error）和PSNR（peak signal-to-noise ratio）两个指标。

$$MAE(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{3XY} \sum_{k=R,G,B} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \left| I_{x,y}^{k} - \hat{I}_{x,y}^{k} \right|,$$

其中R代表三个颜色通道之一。

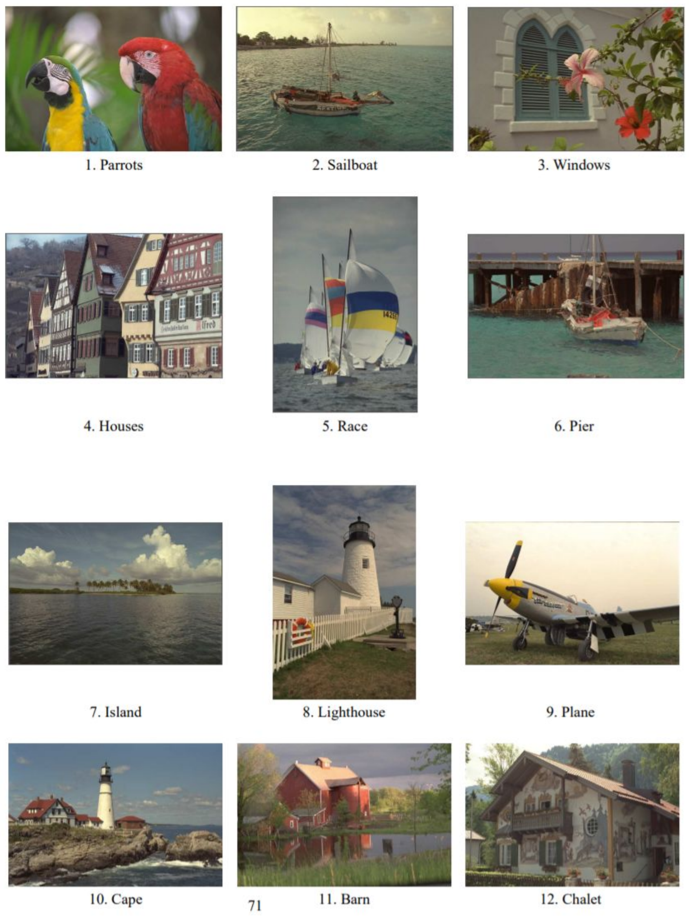
边缘检测的误差可以分为SD和OD两种情况，SD即SubDetect，代表该检的漏检，OD 即OverDetect，代表虚报。下图中I代表输入图像，I^代表插值图像，B图用X指示真实的边缘，B^图用X和粗体X代表实际检出的边缘，J=B^-B可以进一步分解成SD图和OD图。



用SD和OD两个参数可以评估边缘检测算法的准确性等级。

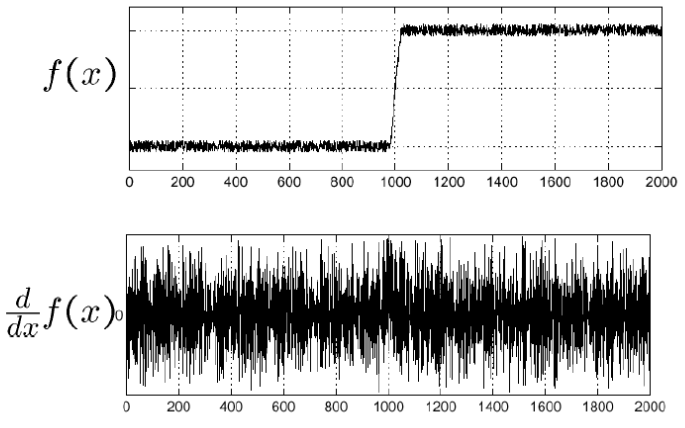
$$\widetilde{SD}_{\%} = \frac{100}{XY}\,\mathrm{Card}\left\{P(x,y) \mid \widetilde{SD}_{x,y} \neq 0\right\},$$
$$\widetilde{OD}_{\%} = \frac{100}{XY}\,\mathrm{Card}\left\{P(x,y) \mid \widetilde{OD}_{x,y} \neq 0\right\}.$$

对于一个demosaic算法，可以在标准数据集测试上述各个指标的分值，得到的分值表格基本上能够反映demosaic算法的整体质量。下图所示的是业界常用的kodak数据集。



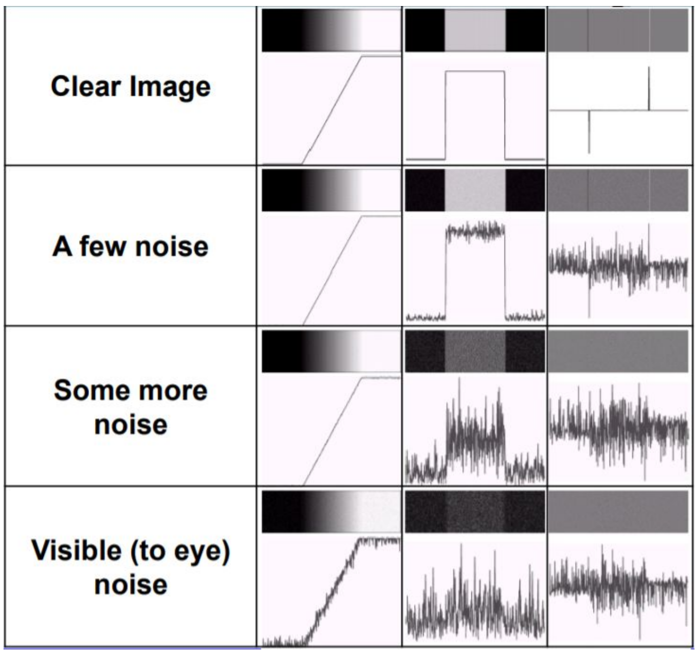
噪声档案 Noise Profile

微分算子对图像的差异敏感，所以不可避免地会放大图像中的噪声。当图像中噪声较大时，就会遇到边缘微分淹没在噪声信号中的情况，如下图所示。



因此一般需要对微分图像进行低通滤波处理，消除一部分噪声。除此之外，插值的时候还需要估计图像的噪声水平，将边缘判断条件设置在噪声平均水平之上，以减少噪声的干扰。图像的噪声水平可以通过IQ标定的方法获取，一种做法是将相机ISO范围分成32个区段，分别标定每个ISO区段噪声的方差（以DN值表示）。

下图显示了图像噪声对边缘检测的影响。



各向异性滤波器 Anisotropic Filter （AF）

传统的低通滤波器容易破坏图像中的边缘，因此人们发明了各向异性滤波器，它需要与边缘检测滤波器配合工作，当图像中检出边缘后，沿边缘切线方向按正常强度滤波，沿法线方向则基本不滤波。如果未检出边缘则正常滤波。这样的设计兼顾了图像去噪和边缘保持两个需求，在实践中得到了广泛应用。

典型算法

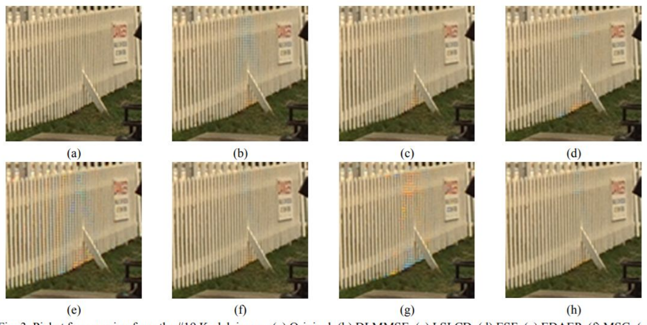
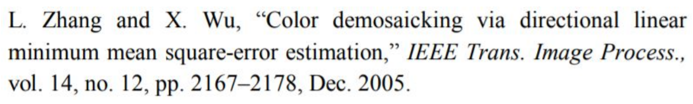
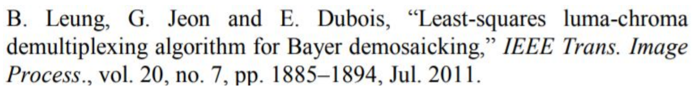


Fig. 3. Picket fence region from the #19 Kodak image. (a) Original, (b) DLMMSE, (c) LSLCD, (d) ESF, (e) EDAEP, (f) MSG, (g) VDI, and (h) HDW (proposed).



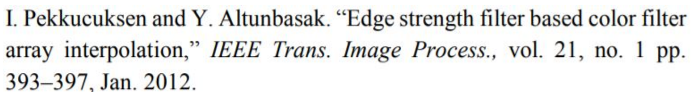
L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2167–2178, Dec. 2005.

DLMMSE



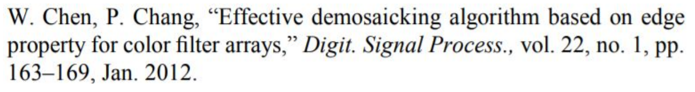
B. Leung, G. Jeon and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1885–1894, Jul. 2011.

LSLCD



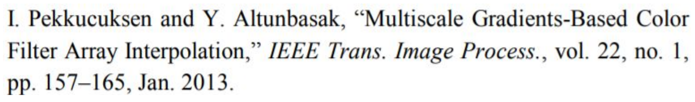
I. Pekkucuksen and Y. Altunbasak. "Edge strength filter based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 21, no. 1 pp. 393–397, Jan. 2012.

ESF



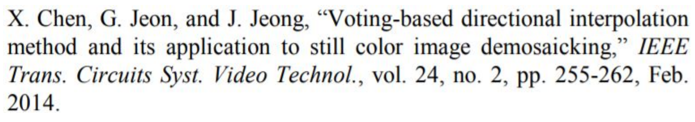
W. Chen, P. Chang, "Effective demosaicking algorithm based on edge property for color filter arrays," *Digit. Signal Process.*, vol. 22, no. 1, pp. 163–169, Jan. 2012.

EDAEP



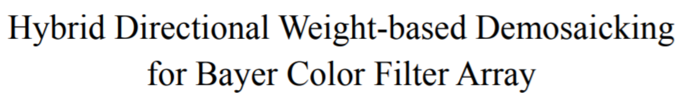
I. Pekkucuksen and Y. Altunbasak, "Multiscale Gradients-Based Color Filter Array Interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 157–165, Jan. 2013.

MSG



X. Chen, G. Jeon, and J. Jeong, "Voting-based directional interpolation method and its application to still color image demosaicking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 255-262, Feb. 2014.

VDI



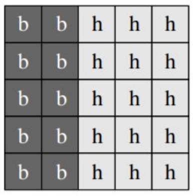
Hybrid Directional Weight-based Demosaicking for Bayer Color Filter Array

HDW

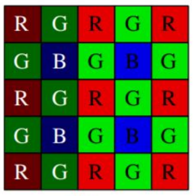
抗伪彩 False Color Suppression

Demosaicking 算法一个主要挑战是预测颜色不准时会在图像中引入伪彩，尤其在灰色区域上伪彩尤其明显，如下图所示。

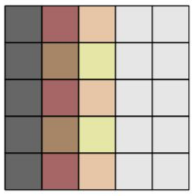
下图显示了伪彩和拉链效应（zipper effect）产生的原理，图(c)中黄色（黄=红+绿）伪彩的产生是因为受第三列高亮像素的影响，第二列低亮像素中的红、绿像素的插值结果比实际值偏大，而蓝像素不需要插值因此保持不变，所以整体出现黄色调。而同样是在第二列，当G像素不需要插值时，只有R像素插值结果偏高，B和G都保持正常值，则整体出现粉色调。于是出现黄、粉、黄、粉... 的拉链模式。



(a) Original image



(b) CFA image


(c) Estimated image

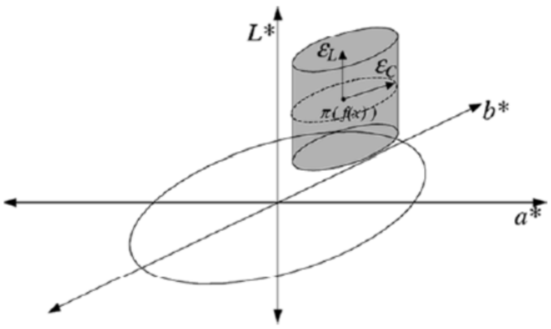
有人提出用以下公式计算FalseColor像素占总像素数的比例，公式引入一个阈值T，像素的RGB值如果与真实值偏离超过T则会被计入FC像素。

$$FC_{\%} = \frac{100}{XY}\,\mathrm{Card}\left\{ P(x,y) \mid \max_{k=R,G,B}\left(\left| I^k_{x,y} - \hat{I}^k_{x,y} \right| \right) > T \right\}.$$

分析伪彩的成因可以发现，伪彩是因为预测错误而因为引入的，它并不具备普通像素所应具有的空间连续性，相反具有跳变特性，因此可以将其视为一种椒盐噪声，在YUV空间用均值滤波的方法处理UV分量，可以将跳变的颜色滤除。这种方法能够在一定程度上减轻伪彩。

有人提出另一种思路，既然伪彩在灰色背景上最敏感，那么不妨专门设计一个灰色检测器，如果发现待插值的像素实际应为灰色，则直接在YUV空间将像素的UV分量钳位至0，这种思路对特定场景具有奇效，已得到实际应用。

Cornell 大学的团队提出了一个算法，能够非常有效地检测到沿某个方向插值是会否引入伪彩。该算法的思路是在中心的一个邻域（集合B）内观察所有像素与中心像素的相似度，因此定义一个相似度函数H。一个像素与中心像素“相似“的标准是，CIELAB空间上的亮度误差小于一个阈值，色度误差也小于一个阈值，符合这个条件的待考察像素定义为相似，否则不相似。H函数的值定义为相似的数量与全部像素数量之比，该值越接近1，伪彩越轻微。



Sobel filter

```
lenna = imread('E:\ImageTest\512\g512_006\lena.pgm');
subplot(241)
```

```
imshow(lenna,[]);title('原图')

%画三维曲面图
% x = 1:512;
% y = 1:512;
% [X,Y] = meshgrid(x,y);
% Z=double(lenna);
% mesh(X,Y,Z);

lenna_1 = edge(lenna,'sobel',0.06);
subplot(242)
imshow(lenna_1,[]);title('Sobel 0.06')

lenna_2 = edge(lenna,'sobel',0.09);
subplot(243)
imshow(lenna_2,[]);title('Sobel 0.09')

lenna_3 = edge(lenna,'sobel',0.12);
subplot(244)
imshow(lenna_3,[]);title('Sobel 0.12')

sigma = 0.6;
N = 5;            %滤波模板大小是（2N+1）×（2N+1）
N_row = 2*N+1;
gausFilter = fspecial('gaussian',[N_row N_row],sigma);  %高斯滤波模板
lenna_0=imfilter(lenna,gausFilter,'conv');
subplot(245)
imshow(lenna_0,[]);title('滤波后')

lenna_4 = edge(lenna_0,'sobel',0.06);
subplot(246)
imshow(lenna_4,[]);title('Sobel 0.06')

lenna_5 = edge(lenna_0,'sobel',0.09);
subplot(247)
imshow(lenna_5,[]);title('Sobel 0.09')

lenna_6 = edge(lenna_0,'sobel',0.12);
subplot(248)
imshow(lenna_6,[]);title('Sobel 0.12')
```

Kirsch filter

```
clear
clc
close all
bw1=imread('E:\ImageTest\512\g512_006\lena.pgm');

%----------------------------------------------------------------
%对图象进行预处理

figure(1)
imshow(bw1,[])
title('原始图象')

%对图象进行均值滤波
bw2=filter2(fspecial('average',3),bw1);

%对图象进行高斯滤波
bw3=filter2(fspecial('gaussian'),bw2);

%利用小波变换对图象进行降噪处理
[thr,sorh,keepapp]=ddencmp('den','wv',bw3);     %获得除噪的缺省参数
bw4=wdencmp('gbl',bw3,'sym4',2,thr,sorh,keepapp);%图象进行降噪处理

%----------------------------------------------------------------------
%提取图象边缘
t=[0.8 1.0 1.5 2.0 2.5].*10^5 ;     %设定阈值
bw5=double(bw4);
[m,n]=size(bw5);
g=zeros(m,n);
d=zeros(1,8);
%利用Kirsch算子进行边缘提取
for i=2:m-1
   for j=2:n-1
       d(1) =(5*bw5(i-1,j-1)+5*bw5(i-1,j)+5*bw5(i-1,j+1)-3*bw5(i,j-1)-3*bw5(i,j+1)-3*b
       d(2) =((-3)*bw5(i-1,j-1)+5*bw5(i-1,j)+5*bw5(i-1,j+1)-3*bw5(i,j-1)+5*bw5(i,j+1)-
       d(3) =((-3)*bw5(i-1,j-1)-3*bw5(i-1,j)+5*bw5(i-1,j+1)-3*bw5(i,j-1)+5*bw5(i,j+1)-
       d(4) =((-3)*bw5(i-1,j-1)-3*bw5(i-1,j)-3*bw5(i-1,j+1)-3*bw5(i,j-1)+5*bw5(i,j+1)-
       d(5) =((-3)*bw5(i-1,j-1)-3*bw5(i-1,j)-3*bw5(i-1,j+1)-3*bw5(i,j-1)-3*bw5(i,j+1)+
       d(6) =((-3)*bw5(i-1,j-1)-3*bw5(i-1,j)-3*bw5(i-1,j+1)+5*bw5(i,j-1)-3*bw5(i,j+1)+
       d(7) =(5*bw5(i-1,j-1)-3*bw5(i-1,j)-3*bw5(i-1,j+1)+5*bw5(i,j-1)-3*bw5(i,j+1)+5*b
       d(8) =(5*bw5(i-1,j-1)+5*bw5(i-1,j)-3*bw5(i-1,j+1)+5*bw5(i,j-1)-3*bw5(i,j+1)-3*b
       g(i,j) = max(d);
    end
end

%显示边缘提取后的图象
figure(5)
```

```
for k=1:5
    for i=1:m
        for j=1:n
            if g(i,j)>t(k)
                bw5(i,j)=255;
            else
                bw5(i,j)=0;
            end
        end
    end
    subplot(1,5,k)
    imshow(bw5,[])
    title(['Kirsch' '  ' num2str(t(k))])
end
```

编辑于 2022-07-17 20:17

Camera　图像信号处理器ISP (Image Signal Processor)