Color image Demosaicing

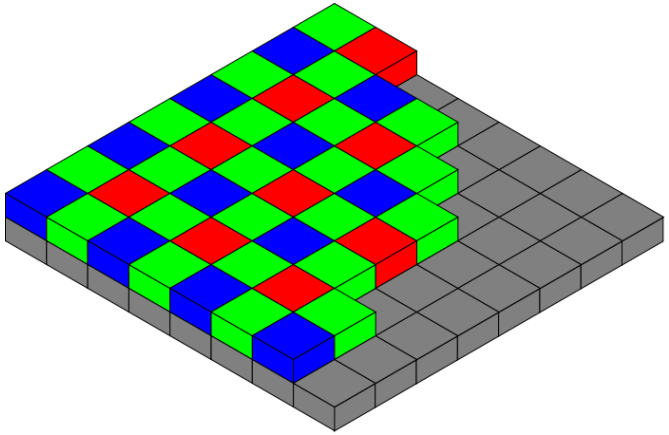
CS 663, Ajit Rajwade

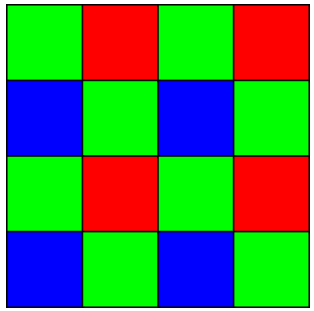
Color Filter Arrays

- It is an array of tiny color filters placed before the image sensor array of a camera.
- The resolution of this array is the same as that of the image sensor array.
- Each color filter may allow a different wavelength of light to pass – this is pre-determined during the camera design.

Color Filter Arrays

- The most common type of CFA is the Bayer pattern which is shown below:





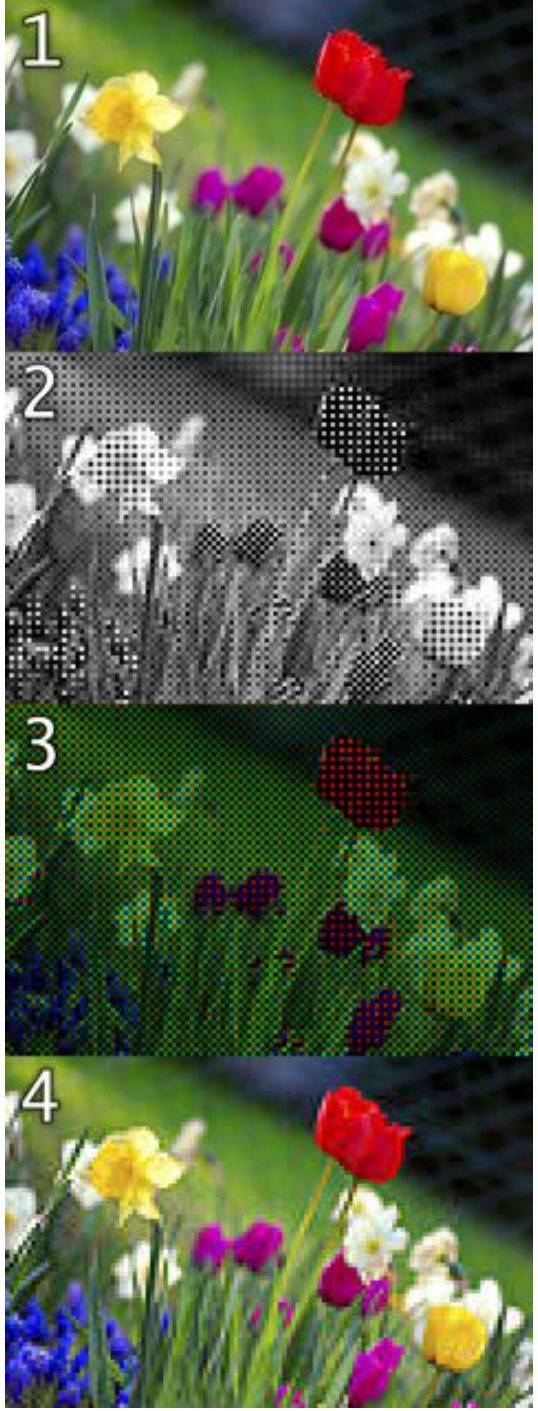
https://en.wikipedia.org/wiki/Color_filter_array

- The Bayer pattern collects information at red, green, blue wavelengths only as shown above.

*The word “mosaic” or “mosaiced” is not to be confused with image panorama generation which is also called image mosaicing.

Color Filter Arrays

- The Bayer pattern uses twice the number of green elements as compared to red or blue elements.
- This is because both the M and L cone cells of the retina are sensitive to green light.
- The **raw** (uncompressed) output of the Bayer pattern is called as the **Bayer pattern image** or the **mosaiced** (*) **image**.
- The mosaiced image needs to be converted to a normal RGB image by a process called color image **demosaicing**.



“original scene”

https://en.wikipedia.org/wiki/Bayer_filter

Mosaiced image

Mosaiced image – just coded with the Bayer filter colors

“Demosaiced” image – obtained by interpolating the missing color values at all the pixels

A Demosaicing Algorithm

- There exist a plethora of demosaicing algorithms.
- We will study one that is implemented in the **“demosaic”** function of MATLAB.
- The algorithm implemented by this function was published in 2004.

Malvar, H.S., L. He, and R. Cutler, *High quality linear interpolation for demosaicing of Bayer-patterned color images*. ICASPP, Volume 34, Issue 11, pp. 2274-2282, May 2004.

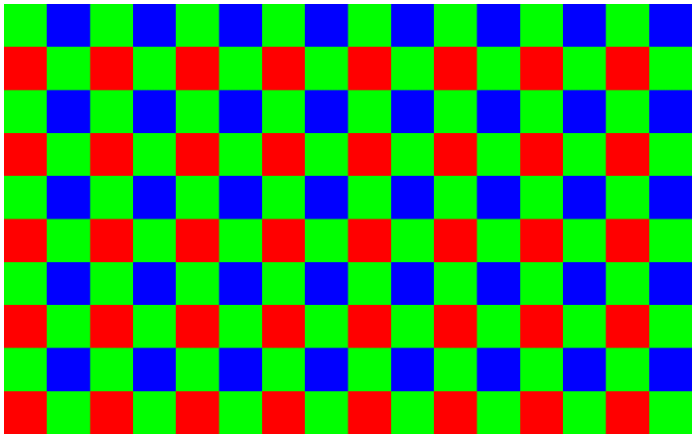
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Demosaicing_ICASSP04.pdf

Demosaicing Algorithm

- Demosaicing involves interpolation of missing color values from nearby pixels.
- The easiest way is to perform linear interpolation – given the structure of the Bayer pattern.

$$\hat{g}(i,j) = \frac{1}{4} \sum g(i+m, j+n)$$

$$(m,n) = \{(0,-1), (0,1), (-1,0), (1,0)\}$$



Demosaicing Algorithm

- But such an algorithm gives highly sub-optimal results at **edges** – as seen in the simulation below.

Original image (top left), o/p of bilinear interpolation for demosaicing (top right), o/p of MATLAB's demosaic algorithm (bottom left)

Demosaicing algorithm

- Make use of the correlation between R,G,B color values for a more edge-aware interpolation!
- Edges in natural images have stronger luminance changes than chrominance changes.
- Consider the case of finding G at an R or a B pixel.
- The R-gradient can be useful information for determining the G value.

Demosaicing algorithm

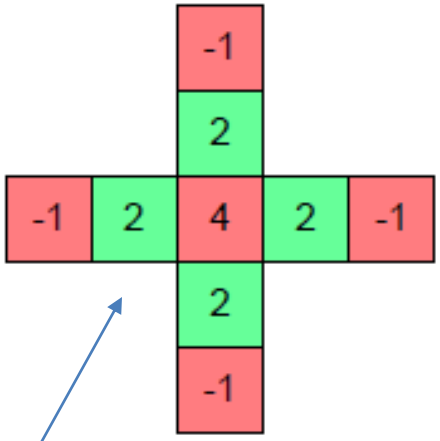
- Consider the case of finding G at an R or a B pixel (x,y).
- Obtain an estimate of the R value at pixel (x,y) by bilinear interpolation.
- If the actual R value at (x,y) differs considerably from the bilinearly interpolated R value at (x,y), it means that there is a sharp luminance change at that pixel.
- The corrected value of G is given as follows:

$$\hat{g}(i,j) = \hat{g}_B(i,j) + \alpha \Delta_R(i,j) \qquad \Delta_R(i,j) \triangleq r(i,j) - \frac{1}{4} \sum r(i+m, j+n)$$

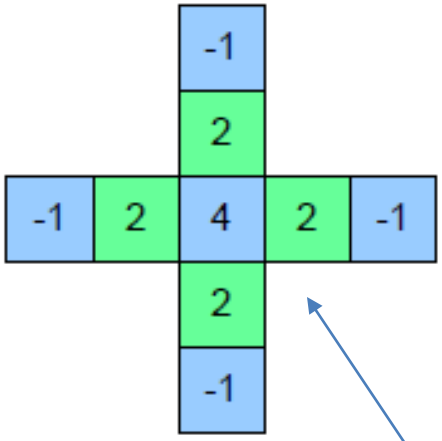
$$(m,n) = \{(0,-2), (0,2), (-2,0), (2,0)\}$$

Bilinearly interp. value

Gain factor



G at R locations



G at B locations

$$\hat{g}(i,j) = \hat{g}_B(i,j) + \alpha\,\Delta_R(i,j)$$

$$\Delta_R(i,j) \triangleq r(i,j) - \frac{1}{4}\sum r(i+m, j+n)$$

$(m,n)=\{(0,-2),\ (0,2),\ (-2,0),\ (2,0)\}$

$$\hat{g}(i,j) = \hat{g}_B(i,j) + \gamma\Delta_B(i,j)$$

$$\Delta_B(i,j) = b(i,j) - \frac{1}{4}\sum_{(m,n)=\{(0,-2),(-2,0),(2,0),(0,2)\}} b(i+m, j+n)$$

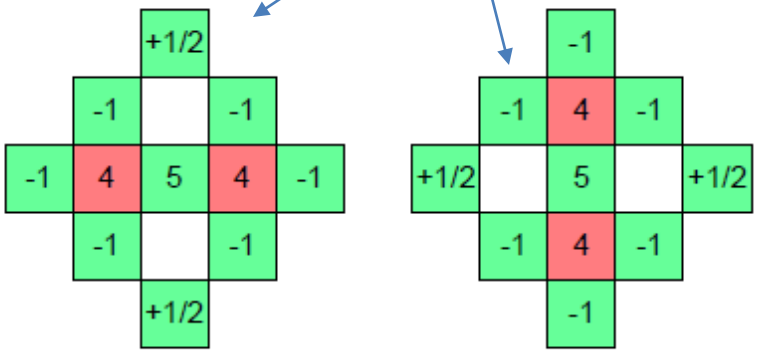
Demosaicing algorithm

- We have seen how to obtain G at an R or a B pixel.

- To obtain the R value at a G pixel, the corresponding formula is

Bilinear interp. value

$$\hat{r}(i,j) = \hat{r}_B(i,j) + \beta\,\Delta_G(i,j)$$

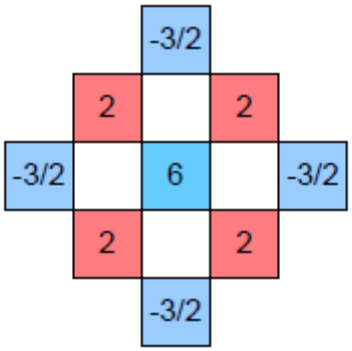


Demosaicing algorithm

- To obtain a R value at a B pixel, the corresponding formula is

Bilinear interp. value

$$\hat{r}(i,j) = \hat{r}_B(i,j) + \gamma\,\Delta_B(i,j)$$



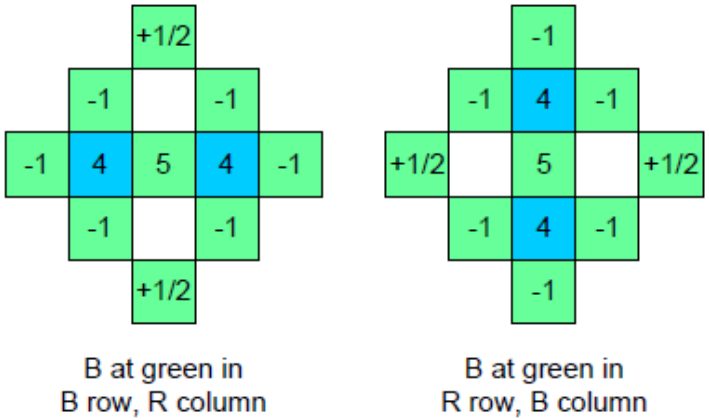
R at blue in
B row, B column

Demosaicing algorithm

- To obtain a B value at a G pixel, the corresponding formula is

Bilinear interp. value

$$\hat{b}(i,j) = \hat{b}_B(i,j) + \beta\Delta_G(i,j)$$

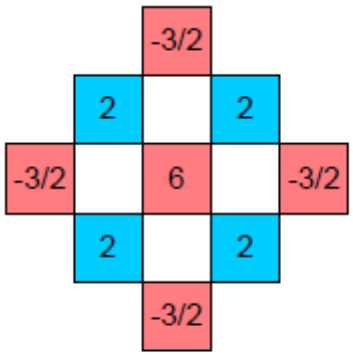


Demosaicing algorithm

- To obtain a B value at a R pixel, the corresponding formula is

Bilinear interp. value

$$\hat{b}(i,j) = \hat{b}_B(i,j) + \gamma\Delta_R(i,j)$$



B at red in
R row, R column

Gain factors

- The values α, β, γ are gain factors for the correction due to gradients in the R,G,B channels respectively.

- How are they estimated? In a training phase of the algorithm – performed offline.

- The gain factors were designed to optimize a mean square error criterion.

Demosaicing: when does it happen?

- Your camera acquires images in a raw format, with 12 bits per pixel.
- Remember: at each pixel, only one of the R,G,B values is measured.
- That is, the camera measures just the CFA image.
- The camera then runs a demosaicing algorithm internally to generate the full RGB image.
- This image then goes through various intensity transformations after which it is JPEG-compressed and stored in the camera memory card.
- The demosaicing algorithm described earlier does not perform any noise removal – which can lead to noisy artifacts in the final image!