Silvia Arellano García, Alexandre Olivé Pellicer, Alejandro Amat Paya

Session 07: Recording & Sensors: capture (CFA)

Goals

In this session, we will introduce the problems related to 1-CCD cameras and CFA signals.

1 Professional 3CCD cams versus Color Filter Arrays (CFA)

A color image is typically represented by combining three separate monochromatic images R, G and B. Ideally, each pixel reflects three data measurements; one for each of the color bands. This is the strategy used in professional (and expensive) 3-CCD color cameras. In practice, to reduce production costs, non-professional cameras have only one CCD, less expensive in terms of circuitry and arrangement of the optics. This means that a single color measurement (red OR green OR blue) is taken per pixel. The technology of 1-CCD cameras has evolved fast, and now a single CCD sensor may contain far more pixels than what would be needed for HD, or even more than 3 times HD (in 3-CCD HD cameras). These cheaper 1-CCD cameras with larger sensor arrays are nowadays somehow making their way into the production studios…

The detector array of 1-CCD cameras is a grid of CCDs, each made sensitive to one color by placing a Color-Filter Array (CFA) in front of the CCD. The Bayer pattern shown below is a very common example of such a CFA. The values of the missing color bands at every pixel are often synthesized using some form of interpolation from neighboring pixel values. This process is known as color ***demosaicing***.

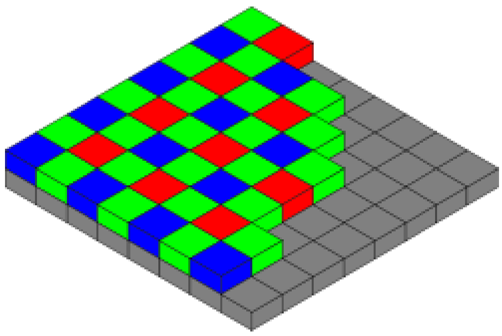


Figure 1. Bayer arrangement of color pixels (from Wikipedia)

The Elphel NC353 is a network camera (IP) with a 5MPix 1-CCD Bayer CFA sensor. In this session, we will analyze the actual definition we can obtain from the camera and the different modes to access the data on the sensor; namely, with onboard processing on the camera or with processing on the host computer (your PC).

According to the specifications the camera is equipped with a color Image Sensor 1/2.5", Bayer-pattern 'grbg', and an effective number of pixels 2592x1936 (5,018,112 pixels). The nominal sensor output is 12bit ADC (maximum depth of acquired sensor pixels is 12 bpp), but unless an HDR (High Dynamic Range) mode is used, the quantization depth is set to 8bpp.

2 Grabbing your first Elphel camera frame

2.1 Capture

Now we will capture our first image from the IP camera. Note that we are capturing from an IP in the local network –private to the labs. As usual, you first need to indicate to Matlab the current working directory.

```
% first change to the session directory
cd('');


% set camera IP and mode
camIP   ='10.4.91.242';
camPORT ='80';
camMODE = 1;% normal color acquisition mode

% set camera mode
setElphelCam(camIP,camPORT,camMODE,1);

% prepare the URL string
camURL = sprintf('http://%s:8081/bimg',camIP)
```

Now you can copy the resulting URL in a browser to capture a camera image.

Then, you can save the image displayed in the browser on your hard drive and then read the file in matlab.

```
% read and display an Elphel image in MATLAB
camRGB=imread('NameOfTheImageSavedOnDisk.png');
figure('Name','RGB'); imshow(camRGB);
```

If for some reasons you cannot access the camera (sometime it may not be working), we have included in the “images” folder, the image TestElphel.png. Assuming the working Matlab directory in the folder containing this word file, you can access to the image with (depending of your operating system, you may have to change the “\” for “/” in the filename).

```
% read and display an Elphel image in MATLAB
camRGB=imread('.\images\TestElphel.png');
figure('Name','RGB'); imshow(camRGB);
```

Note: Don't insert the image in this MS Word document (it is too large, we risk crashing MS Word!).

The image size is shown by the following command.

```
size(camRGB)
```

What is the actual memory needed to store the image? Compute it below:

```
ht=size(camRGB,1)
wt=size(camRGB,2)
ht * wt * 3 / (1024 * 1024)
```

Q1. Justify the computation above: why do we multiply by 3? what are the units of the resulting size? Bits, Kbits, Mbits, Gbits, Bytes, Kbytes, Mbytes, GBytes?

- Multiplicamos por 3 porque la señal tiene 3 canales (RGB).
- Las unidades resultantes son Mbytes, puesto que está considerando el pixel en sí. Cada píxel con sus componentes RGB se codifica con 8 bits, 1 Byte. Al dividir entre 2^20 tenemos Mbytes.

2.2 Bit-rate needed for live video capture

Now imagine that we capture video from the camera at 25fps. Modify the last expression to obtain the raw transmission rate in Mbit/s through the IP channel

```
ht * wt * 3 * 8 * 25 / (1024 * 1024)% modify this for 25fps video

ans = 2.8714e+03 Mbits/s
```

Q2. The actual camera IP connection is 100Mbit (Mbps or Mbit/s) ethernet. Do you think it will be possible to transmit images this size at 25fps? What would be the minimum nominal bitrate of the ethernet connection needed?

- No, no es posible. Para poder enviar el vídeo necesitaríamos una red con un bitrate mínimo de 2.8714e+03 Mbits/s.

Q3. We can't change the ethernet connection rate due to the hardware at the camera end and, therefore, for the actual 100Mbit connection rate, compute:
a) the maximum frame rate we could get for the same image size, and
b) the maximum image size we could get at 25fps (assume that we keep the aspect ratio, i.e. the proportion of wt to ht).

```
100 * 1024 * 1024 / (ht * wt * 3* 8)
ans = 0.8707
```

- El máximo frame rate que podemos enviar es 0.8707
- La imagen resultante tiene un total de 1.7476e+05 píxeles. Como la imagen anterior tenía una relación de 1.33 (wt/ht), las dimensiones obtenidas son wt = 483 y ht = 363

Q4. If we want to stick to the maximum image size of 2592 x 1936 and 25fps, what else could we do to capture full definition video through the camera 100Mbit ethernet connection? Propose a potential solution and discuss its drawbacks… (You may think about a course you already have done or are currently taking at ETSETB)

- Podríamos comprimir la imagen. Así evitamos mandar los bits que no son perceptualmente importantes. Esto lo podríamos realizar aplicando la DCT o la transformada Wavelet. No obstante, esto requeriría más poder computacional tanto en el codificador como en el decodificador.”

3 *Bayer demosaicing and image definition*

The color filter array (CFA) for this camera sensor has a Bayer pattern as shown in Figure 2 (left). Counting together all the G, R and B pixels of the CFA, there is a total of 2592 x 1936 pixels in the sensor. The camera transforms these pixels into a complete 3 component RGB image before downloading to your PC (Figure 2, right).

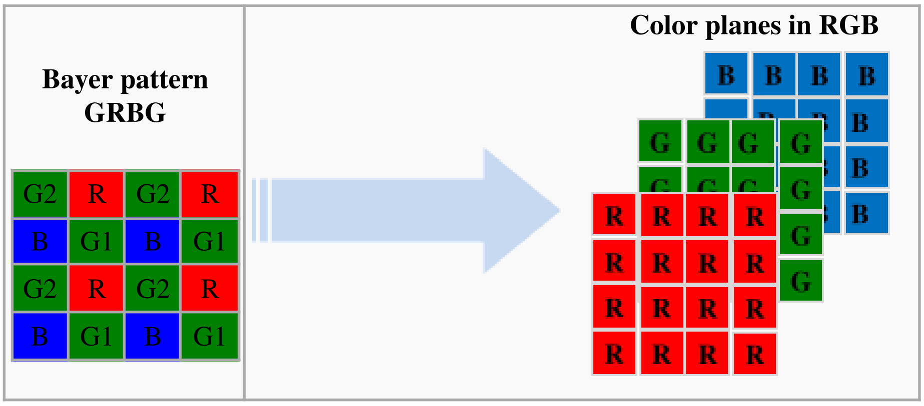


Figure 2. Bayer pattern ***'grbg'*** *in the Elphel camera sensor (single CCD) transformed to an RGB image by camera onboard processing*

Recall the size of the image downloaded to your computer

```
size(camRGB)
```

Q5. What are the maximum horizontal and vertical definitions that the downloaded camRGB image can display in cycles per picture width (cpw) and cycles per picture height (cph)? (Remember that cpw (cph) is a spatial resolution unit corresponding to the maximum number of cycles of frequency 1/2, that is a black pixel followed by a white pixel, that may be fitted across the width (height) of the picture).

- Como el periodo más rápido es 2 (una señal que alterna un pixel blanco y un pixel negro), el número de ciclos necesario lo obtendremos dividiendo entre dos las es 1296 * 968.

The Bayer patterns of the R, B and G acquired by the camera sensor are show in Figure 3.

Figure 3. Separate red, blue and green pixels in the sensor image

Q6. What is the name of the process that the camera has to perform to obtain the R image from the R pixels in the Bayer pattern (or the G image from the G pixels, or the B image from B pixels)? What is the ratio of original pixels in each image?

- Se ha de interpolar. Las imágenes del plano R y del plano B tienen ¼ de los píxeles originales de la imagen del sensor cada una, mientras que en la imagen del plano G, ½ de los píxeles provienen de la imagen del sensor original.

Q7. What are the horizontal and vertical definition considering only the R (or B) pixels in the sensor image in cpw and cph? And for the G pixels?

- R y B tienen una definición de 648cpw * 484 cph.
- La definición de G no ha cambiado: 1296cpw * 968 cph

The processing performed by the camera to create the R, B and G images is illustrated in Figure 4.

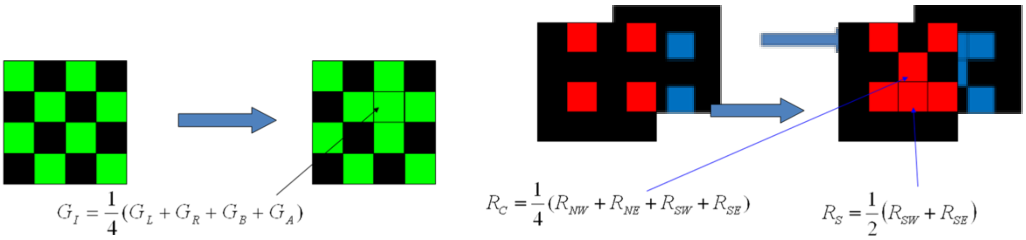


Figure 4. Obtaining the missing pixels in the green, red and blue channels

Q8. After this analysis, review your answer to question **Q4** and propose what needs to be transmitted through the ethernet connection if we want to stick to the raw image definition captured in the sensor pixels? What would be the savings in terms of bitrate?

- Para mantener la definición original y aún así comprimir, habríamos de enviar las componentes sin interpolar, y que fuese el receptor quien interpolase. Así se mandarían muchos menos pixeles y será más fácil ajustarse a un ancho de banda más limitado.
- Inicialmente se querían transmitir 1936*2592*3 píxeles. Ahora se transmiten todos los píxeles del canal G, pero solo ¼ de los canales R y B. Es decir: 1936*2592 + 968*1296 + 986*1296.
- Si comparamos ambas transmisiones, en la segunda se están enviando la mitad de píxeles por frame que en la primera.

There is a way to download raw Bayer images from the Elphel camera sensor, without onboard processing. We download an example:

```
% read and display an Elphel image in MATLAB
 url=grabElphelRaw(camIP,camPORT,1)
```

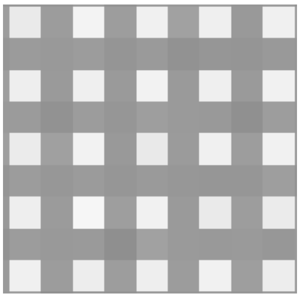
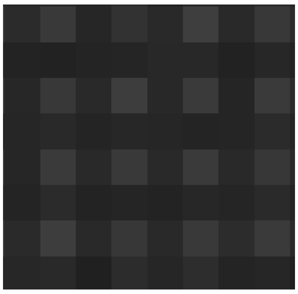
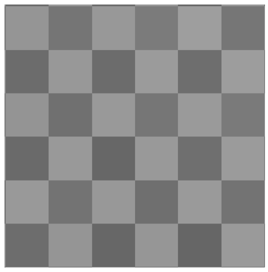
Now, you should visit this URL and download the raw image in the computer hard drive. Then, we need to convert the raw image to a MATLAB image. This can be done with the imReadRaw function. If for some reason, you cannot access the camera, we have done this for you and the resulting image is: TestElphel.raw that can be found in the "images" directory.

First, define the filename of the raw image that you saved in the computer (here again, depending on your operating system, you may have to change the "\" for "/" in the filename).

```
Filename = 'C:\Users\tpa\Downloads\image.raw' % filename used to save the raw image
cfa=imReadRaw(Filename,wt,ht);
figure('Name','CFA'); imshow(cfa,[]);
```

Observe (with zoom) the pixel structure of this image.

Q9. Analyze areas of specific color and the corresponding patterns observed in the Raw image (in order to identify areas with specific colors, note that the raw image corresponds to the same scene as the one seen in the TestElphel.png file that you analyzed at the beginning of this work.) Focus in particular in objects that have either red, green or blue colors. Paste portions of the image in this document. Analyze and comment the observed Bayer patterns. For each pattern, try in particular to identify the locations of pixels capturing a specific color.

		
Papel rojo	Armario azul	Croma verde

- Podemos observar que en la imagen original capturada por la cámara (.raw) se respeta el patrón de Bayer. En los tres casos, los píxeles de interés se representan con un color más claro (valor cercano a 255). Si nos fijamos en regiones de 4x4 píxeles, puede apreciarse que encaja con la teoría explicada anteriormente:

Figure 3. Separate red, blue and green pixels in the sensor image

- Vemos que en el caso de los colores rojo y azul conservamos uno de cada cuatro píxeles, mientras que en el verde conservamos uno de cada dos, tal como se ha explicado anteriormente.

We will now study the demosaicing of the CFA images. A good test image for this task is the following lighthouse image. Let us first read the image acquired by the CFA sensor:

```
% read original CFA image
cfa = uint8(imread('.\images\lighthouseCFA.png'));
figure('Name','CFA'); imshow(cfa);
```

The first demosaicing technique we will use is simply a down-sampling!

```
rgb1=demosaic_downsample(cfa);
% Bayer pattern "grbg". CFA demosaicing in mode 0 (G=G1)
figure('Name','camCFA -demosaicing downsample'); imshow(rgb1)
```

Q10. What are the main issues with this demosaicing technique?

- Aparece aliasing en las frecuencias altas.
- Hay zonas donde se ha generado un color falso.
- Las dimensiones de la imagen se han dividido entre 2.

We can employ better demosaicing method as the one used in the camera that relies on interpolation (see Figure 4).

```
rgb2=demosaic_interpolate(cfa);
figure('Name','camCFA -demosaicing interpolation'); imshow(rgb2)
```

Q11. What are the main differences with the previous demosaicing approach?

- No hay aliasing, pero tenemos errores de interpolación que crean colores falsos (que no están en la imagen original). Además, ahora no se diezma, solo se interpola, por lo que la imagen generada es del tamaño original.

As you can see, false colors are appearing in these images. In order to precisely understand what has occurred, let us analyze a simple test image involving a transition between a white area, RGB=(255,255,255), on the left to a pure green area, RGB=(0,255,0), on the right. The size of the test image is 8x8 pixels:

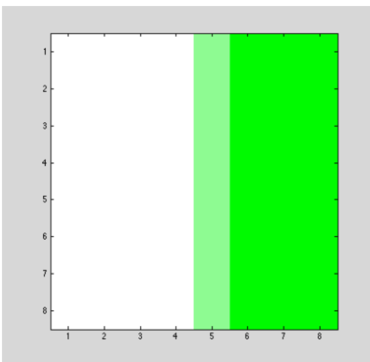


Figure 4. Simple 8x8 test image

Note that the image involves a transition of one pixel between the pure white to the pure green. The RGB value of the transition pixels are: RGB=(128,255,128).

Q12. Assume a 1-CCD camera is observing the scene defined in Fig. 4. Specify manually the CFA image acquired by the 1-CCD sensor by substituting the "." By a value between 0 and 1 (as we have already multiplied the matrix by 255).

```
cfa_test = uint8(255 * [
    1 1 1 1 1 0.5 1 0 ;
    1 1 1 1 0.5 1 0 1 ;
    1 1 1 1 1 0.5 1 0 ;
    1 1 1 1 0.5 1 0 1 ;
    1 1 1 1 1 0.5 1 0 ;
    1 1 1 1 0.5 1 0 1 ;
    1 1 1 1 1 0.5 1 0 ;
    1 1 1 1 0.5 1 0 1 ;
]);
```

Apply first the demosaicing technique by down-sampling.

```
rgb1_test = demosaic_downsample(cfa_test)
figure('Name','Test downsample'); imagesc(rgb1_test)
```

Q13. Why is a new color appearing?

- Al aplicar demosaicing se obtienen las siguientes matrices RGB. El nuevo color aparece debido al patrón de Bayer y los píxeles que se escogen para cada canal. Para que la transición entre colores fuese "limpia", los valores de 0,5 deberían ser 0. Sin embargo, tal como está estructurada nuestra matriz CFA, los valores de 0.5 afectan a los canales R y B, alterando el color del píxel intermedio.

```
Bayer pattern "grbg". CFA demosaicing in mode 0 (G=G1)
rgb1_test = 4×4×3 uint8 array
rgb1_test(:,:,1) =

   255   255   128     0
   255   255   128     0
   255   255   128     0
   255   255   128     0


rgb1_test(:,:,2) =

   255   255   255   255
   255   255   255   255
   255   255   255   255
   255   255   255   255


rgb1_test(:,:,3) =

   255   255   128     0
   255   255   128     0
   255   255   128     0
   255   255   128     0
```

Apply now the demosaicing technique by interpolation.

```
rgb2_test = demosaic_interpolate(cfa_test)
figure('Name','Test interpolation'); imagesc(rgb2_test)
```

Q14. Analyze and interpret the main differences with the previous demosaicing approach on the test image.

- Al aplicar esta técnica aparecen 4 colores intermedios, cada uno de un ancho de un píxel, entre el blanco inicial y el verde final.
- Esto se debe a que al interpolar se crean nuevos valores de píxeles intercalados a los originales, que resultan en una transición más suave entre los dos valores que realmente conocemos. En nuestro caso, el valor que toman los píxeles de los canales R y B decrecen de 64 en 64, resultado de calcular una "media" entre los

```
rgb2_test = 8×8×3 uint8 array
rgb2_test(:,:,1) =

   255   255   255   255   192   128    64     0
   255   255   255   255   192   128    64     0
   255   255   255   255   192   128    64     0
   255   255   255   255   192   128    64     0
   255   255   255   255   192   128    64     0
   255   255   255   255   192   128    64     0
   255   255   255   255   192   128    64     0
   255   255   255   255   192   128    64     0


rgb2_test(:,:,2) =

   255   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255
   255   255   255   255   255   255   255   255


rgb2_test(:,:,3) =

   255   255   255   192   128    64     0     0
   255   255   255   192   128    64     0     0
   255   255   255   192   128    64     0     0
   255   255   255   192   128    64     0     0
   255   255   255   192   128    64     0     0
   255   255   255   192   128    64     0     0
   255   255   255   192   128    64     0     0
   255   255   255   192   128    64     0     0
```

píxeles reales que lo rodean. Esto da lugar a franjas de nuevos colores.

Q15. Interpret now more precisely the issues observed in the lighthouse image. What can be said about the distortions introduced in each case?

- La imagen donde se ha realizado la interpolación se percibe "con menos error", sobre todo en aquellas zonas de altas frecuencias. Si nos fijamos en la valla, en ambos casos se pueden apreciar colores falsos. Sin embargo, en el segundo caso, como la transición es mucho más suave, no son tan perceptibles.
- En la imagen anterior (franjas verdes) los cambios eran mucho más perceptibles, pues la imagen era mucho más simple. Sin embargo en esta, al verlos "de lejos" y juntos con muchos otros píxeles, el efecto de la interpolación no es tan evidente al ojo.

Now that we understand better the issues related to the downsampling and the interpolation approaches, we can apply a more advanced demosaicing technique on the lighthouse (it takes a while)

```
tic;
rgb3 = DDFW(double(cfa));
figure('Name','demosaiced'); imshow(uint8(rgb3));
toc
```

Q16. How does this technique compare to the previous one? Could you speculate about possible ways to obtain this kind of results?

- La principal mejora respecto utilizar un simple diezmado es que se conserva la dimensión de la imagen original y que no se produce aliasing. La principal mejora respecto a una simple interpolación es que no se producen falsos colores o, por lo menos, se parecen más a los reales. Al fin y al cabo se aprovechan las principales ventajas del diezmado y de la interpolación.
- Estos resultados podrían haber sido obtenidos tras considerar los tres canales de la imagen conjuntamente, y después interpolar. En los casos anteriores se interpola cada canal por separado, pero hay que tener en cuenta que el ojo humano los ve juntos.

The light house image was also acquired with a 3-CCD color camera. This original image can be considered as the "ideal" image we want to get back to from the CFA image.

```
% read original image
orig = uint8(imread('.\images\lighthouse.png'));
figure('Name','original'); imshow(orig);
```

We can also objectively measure the introduced distortion by the demosaicing techniques (we do not compare with the downsampling approach as the number of pixels in "orig" and "rgb1" are different):

```
cpsnr_calc2(orig,rgb2,6);
cpsnr_calc2(orig,rgb3,6);
```

In order to deliver this document, please log in Atenea and upload the document in the "Lab7" task (In the "Lab sessions" area).