



# AVITCR LLC

## Módulo de acople Bayer a RGB

### 1. Problema

Nuestra empresa *AVITCR LLC* se dedica a producir vehículos autónomos para misiones de rescate y exploración en terrenos riesgosos para los seres vivientes. Para ello es indispensable que los mismos estén equipados con sistemas de captura de audio y video en tiempo real.

Para nuestro último proyecto estamos evaluando el sensor de imagen CMOS de 4MP **OV4689** de OmniVision™ en conjunto con la plataforma empotrada **Jetson TK1** de NVidia™. Nuestro inconveniente principal es que el sensor únicamente produce una imagen cruda Bayer de 10 bits (de los cuáles únicamente utilizaremos 8 bits) y para su procesamiento posterior necesitamos que la misma esté en formato RGB. A pesar de que la Jetson TK1 consta con una unidad de procesamiento de imagen (ISP) que es capaz de realizar esta transformación, su acceso está limitado a un rango de sensores que no incluye el OV4689.

Requerimos de sus servicios para desarrollar un prototipo del módulo de acople entre el sensor y el procesador. El módulo se encargará de la conversión Bayer a RGB eficientemente, de manera que permita la operación del sistema en tiempo real. Para ello nuestro departamento de Investigación y Desarrollo le facilitará un simulador.

### 2. Requerimientos del Sistema

En su configuración más exigente, el sistema operará en *Full HD* a 60 cuadros por segundo. Esto impone la limitante de que el prototipo deberá realizar su procesamiento en menos de 16.67ms.

- **Entrada:** Imágen Bayer de 8 bits en formato *RGGB*.
- **Salida:** Imágen RGB de 8 bits en formato *RGB*.
- **Frecuencia de operación:** 60 fps.
- **Resolución de imagen:** *Full HD* (1920x1080).

- **Mapeo en memoria de la imagen:** Imagen en formato ráster.
- **Métodos de interpolación:**
  - Vecino más cercano - *Nearest Neighbor*
  - Bilineal

## 3. Entorno de Desarrollo

### 3.1. Software de Simulación

Para desarrollar esta tarea se facilita un simulador basado en la arquitectura GNU/Linux, el cual será la base de esta tarea. El simulador consta de tres módulos diferentes:

- *simulator.cpp*: Núcleo del simulador. Este archivo no debe ser modificado.
- *statistics.cpp*: Módulo encargado de tomar las medidas de tiempo de la ejecución de la interpolación.
- *prototype.cpp*: Código base para los interpoladores. Aquí es donde debe ir su código.

Al ejecutar el simulador sin ninguna opción en la línea de comando, este emitirá un patrón de prueba y lo desplegará en una ventana. Puede modificar este comportamiento mediante las siguientes opciones:

- `-h, --help` Despliega la ayuda
- `-n, --nearest` Interpolación del vecino más próximo (Opción por defecto).
- `-b, --bilinear` Interpolación bilineal
- `-i <imagen>, --input <imagen>` Utiliza una imagen como entrada en lugar del patrón artificial. Puede utilizar variedad de formatos (png, jpeg, etc...)
- `-o <destino.png>, --output <destino.png>` Guarda la imagen resultante en un PNG, en lugar de desplegarlo en una ventana. Puede usarse en combinación con `-i`.

Para ejecutar el simulador con un patrón de prueba y desplegarlo en una ventana adyacente se utiliza `./simulator`, pero si se quiere almacenar el patrón en una imagen se debe ejecutar como `./simulator -o output.png`. Asimismo, si se quiere utilizar una imagen desde memoria y desplegar el patrón en una ventana, se indica mediante `./simulator -i input.png`. Finalmente, si se desea utilizar una imagen desde memoria y almacenar el resultado también en memoria, se ejecuta mediante `./simulator -i input.png -o output.png`

Al final de la ejecución, el simulador le imprimirá información acerca del tiempo de ejecución de su algoritmo.

## 4. Entregables

Se debe entregar el simulador con el código de interpolación bilineal y del vecino más cercano, además recomendaciones acerca de la implementación basado en calidad visual de las imágenes, tiempo del procesamiento y complejidad de los algoritmos empleados.

El formato de entrega para esta tarea es `<carne>_iiq_pid_2019_tarea01.tar.gz`. Es importante recalcar que esta tarea será revisada contra rúbrica, la cual se encuentra disponible en la descripción de dicha asignatura en el TECDigital.

## 5. Consideraciones adicionales

Para poder instalar las dependencias adicionales que esta tarea requiere, se brinda un script con el cual se instalan dichas dependencias automáticamente, esto mediante:

```
chmod +x dependencies.sh  
sudo ./dependencies.sh
```

Con la finalidad de analizar la eficacia de las soluciones entregadas, se otorgará un 1 % extra por algoritmo a la persona (o personas, esto si una misma persona no tiene el menor tiempo en ambos algoritmos) que logre tener el menor tiempo de ejecución en una computadora con un procesador i7-6700HQ y 16GB DDR3L.