**1. Se propone crear una nueva imagen compuesta a partir de otra ya existente tal como muestra la figura 4. En esta nueva imagen se podrá ver en una columna, la imagen original y sus tres componentes, roja, verde y azul.**

Él en clase lo hizo con una variable salto antes de recorrer fila y columna y un if(salto==x) para ir copiando los colores uno a uno, no todos a la vez como hice yo.

#include <stdio.h>

#include <stdlib.h>

#include <opencv/cv.h>

#include <opencv/highgui.h>

int main(int argc, char\*\* argv) {

if (argc != 2) {

printf("Usage: %s image\_file\_name\n", argv[0]);

return EXIT\_FAILURE;

}

//CV\_LOAD\_IMAGE\_COLOR = 1 forces the resultant IplImage to be colour.

//CV\_LOAD\_IMAGE\_GRAYSCALE = 0 forces a greyscale IplImage.

//CV\_LOAD\_IMAGE\_UNCHANGED = -1

IplImage\* Img1 = cvLoadImage(argv[1], CV\_LOAD\_IMAGE\_COLOR);

// Always check if the program can find a file

if (!Img1) {

printf("Error: fichero %s no leido\n", argv[1]);

return EXIT\_FAILURE;

}

IplImage\* ImgRGB =cvCreateImage(cvSize(Img1->width,Img1->height\*4),Img1->depth,Img1->nChannels);

// a visualization window is created with title 'image'

cvNamedWindow("rgb", CV\_WINDOW\_NORMAL);

// img is shown in 'image' window

cvShowImage("rgb", ImgRGB);

cvWaitKey(0);

for (int fila=0;fila<Img1->height;fila++){

char \*pImg1=Img1->imageData+(fila\*Img1->widthStep);

char \*pImgNormal=ImgRGB->imageData+(fila\*ImgRGB->widthStep);

char \*pImgAzul=ImgRGB->imageData+(fila\*ImgRGB->widthStep)+Img1->imageSize;

char \*pImgVerde=ImgRGB->imageData+(fila\*ImgRGB->widthStep)+2\*Img1->imageSize;

char \*pImgRojo=ImgRGB->imageData+(fila\*ImgRGB->widthStep)+3\*Img1->imageSize;

for(int columna=0;columna<Img1->width;columna++){

\*pImgNormal=\*pImg1;

\*pImgAzul=\*pImg1; //copio B del origen a B en el destino

\*pImgVerde=0;

\*pImgRojo=0;

pImg1++;

pImgNormal++;

pImgAzul++;

pImgVerde++;

pImgRojo++;

\*pImgNormal=\*pImg1;

\*pImgAzul=0; //pongo G a 0

\*pImgVerde=\*pImg1;

\*pImgRojo=0;

pImg1++;

pImgNormal++;

pImgAzul++;

pImgVerde++;

pImgRojo++;

\*pImgNormal=\*pImg1;

\*pImgAzul=0; //pongo R a 0

\*pImgVerde=0;

\*pImgRojo=\*pImg1;

pImg1++;

pImgNormal++;

pImgAzul++;

pImgVerde++;

pImgRojo++;

}

cvShowImage("rgb",ImgRGB);

cvWaitKey(10);

}

cvWaitKey(0);

// memory release for img before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow(argv[1]);

return EXIT\_SUCCESS;

}