#include <stdio.h>

#include <stdlib.h>

#include <opencv/cv.h>

#include <opencv/highgui.h>

#define ALTOBLOQUE 32

#define ANCHOBLOQUE 32

void copiarBloque(int x0, int y0, IplImage\* imagenOrigen, int x1, int y1, IplImage\* imagenDestino) {

for (int fila = 0; fila < ALTOBLOQUE; fila++) {

uchar \*pImag1 = imagenOrigen->imageData + ((y0 + fila) \* imagenOrigen->widthStep)+(x0 \* 3);

uchar \*pDestino = imagenDestino->imageData + ((fila + y1) \* imagenDestino->widthStep)+(x1 \* 3);

for (int columna = 0; columna < ANCHOBLOQUE; columna++) {

\*pDestino++ = \*pImag1++;

\*pDestino++ = \*pImag1++;

\*pDestino++ = \*pImag1++;

}

}

cvShowImage("Destino", imagenDestino);

}

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

if (argc != 3) {

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\* Img2 = cvLoadImage(argv[1], CV\_LOAD\_IMAGE\_COLOR);

IplImage\* Img1 = cvLoadImage(argv[2], CV\_LOAD\_IMAGE\_COLOR);

if (!Img1) {

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

return EXIT\_FAILURE;

}

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

srand((unsigned) time(0));

int x0, y0;

int b\_y = (Img1->height/ALTOBLOQUE);

int b\_x = (Img1->width/ANCHOBLOQUE);

int array[b\_x][b\_y];

memset(array, 0,sizeof(array));

cvNamedWindow("Img1", CV\_WINDOW\_AUTOSIZE);

cvNamedWindow("Destino", CV\_WINDOW\_AUTOSIZE);

int cuenta = b\_x\*b\_y;

while (cuenta > 0) {

x0 = (random() % b\_x);

y0 = (random() % b\_y);

if (!array[x0][y0]) {

copiarBloque(x0\*ANCHOBLOQUE, y0\*ALTOBLOQUE, Img1, x0\*ANCHOBLOQUE, y0\*ALTOBLOQUE, Img2);

array[x0][y0] = 1;

cuenta--;

cvWaitKey(1);

cvShowImage("Img1", Img1);

}

}

cvWaitKey(0);

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

// memory release for img before exiting the application

cvReleaseImage(&Img1);

cvReleaseImage(&Img2);

// cvReleaseImage(&Destino);

// Self-explanatory

cvDestroyWindow(argv[1]);

cvDestroyWindow(argv[2]);

return EXIT\_SUCCESS;

}