Copiar imagen bloque a bloque (32x32)

#include <stdio.h>

#include <stdlib.h>

#include <emmintrin.h>

#include <opencv/cv.h>

#include <opencv/highgui.h>

#define ALTURABLOQUE 32

#define ANCHOBLOQUE 32

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

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

char \*pImg1 = imagenOrigen->imageData + ((y0 + fila) \* imagenOrigen->widthStep) + x0 \* (imagenOrigen->nChannels);

char \*pImg2 = imagenDestino->imageData + ((y1 + fila) \* imagenDestino->widthStep) + x1 \* (imagenDestino->nChannels);

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

\*pImg2++ = \*pImg1++;

\*pImg2++ = \*pImg1++;

\*pImg2++ = \*pImg1++;

}

}

}

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

if (argc != 3) {

printf("Error: 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);

IplImage\* Img2 = cvLoadImage(argv[2], CV\_LOAD\_IMAGE\_COLOR);

// Always check if the program can find the image file

if (!Img1) {

printf("Error: file %s not found\n", argv[1]);

return EXIT\_FAILURE;

}

// a visualization window is created with title: image file name

cvNamedWindow("img1", CV\_WINDOW\_AUTOSIZE);

cvNamedWindow("img2", CV\_WINDOW\_AUTOSIZE);

cvShowImage("img1", Img1);

cvShowImage("img2", Img2);

cvWaitKey(0);

for(int fila = 0; fila < Img2->height; fila = fila + ALTURABLOQUE){

for(int col = 0; col < Img2->width; col = col + ANCHOBLOQUE){

copiarBloque(col, fila, Img1, col, fila, Img2);

cvShowImage("img2", Img2);

cvWaitKey(1);

}

}

cvShowImage("img1", Img1);

cvShowImage("img2", Img2);

cvWaitKey(0);

// memory release for images before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow("img1");

return EXIT\_SUCCESS;

}