No va

#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\_UNCHANGED);

IplImage\* ImgBig = cvCreateImage(cvSize(Img1->width\*2, Img1->height\*3), Img1->depth, Img1->nChannels);

// Always check if the program can find a file

if (!Img1) {

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

return EXIT\_FAILURE;

}

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

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

// img is shown in 'image' window

cvShowImage("examen", ImgBig);

cvWaitKey(0);

\_\_m128i mask\_rojo = \_mm\_set1\_epi32 (0x00FF0000);

\_\_m128i mask\_verde = \_mm\_set1\_epi32 (0x0000FF00);

\_\_m128i mask\_azul = \_mm\_set1\_epi32 (0x000000FF);

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

\_\_m128i \*pOriginal = (\_\_m128i \*) (Img1->imageData + fila \* Img1->widthStep);

\_\_m128i \*pNormal = (\_\_m128i \*) (ImgBig->imageData + fila \* ImgBig->widthStep);

\_\_m128i \*pRojo = (\_\_m128i \*) (ImgBig->imageData + fila \* ImgBig->widthStep + (ImgBig->width/2));

\_\_m128i \*pVerde = (\_\_m128i \*) (ImgBig->imageData + fila \* ImgBig->widthStep + (ImgBig->width/2) + Img1->height);

\_\_m128i \*pAzul = (\_\_m128i \*) (ImgBig->imageData + fila \* ImgBig->widthStep + (ImgBig->width/2) + Img1->height\*2);

for(int col = 0; col < Img1->widthStep; col = col + 16){

\*pNormal = \*pOriginal;

\*pRojo = \_mm\_and\_si128 (\*pNormal, mask\_rojo);

\*pVerde = \_mm\_and\_si128 (\*pNormal, mask\_verde);

\*pAzul = \_mm\_and\_si128 (\*pNormal, mask\_azul);

pOriginal++;

pNormal++;

pVerde++;

pAzul++;

}

cvShowImage("examen", ImgBig);

}

cvWaitKey(0);

// memory release for img before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow(argv[1]);

return EXIT\_SUCCESS;

}