



#include <stdio.h>

#include <stdlib.h>

#include <opencv/cv.h>

#include <opencv/highgui.h>

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

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

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

IplImage\* ImgTrans = cvCreateImage(cvSize(Img1->width, Img1->height), 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("color", CV\_WINDOW\_NORMAL);

// img is shown in 'image' window

cvShowImage("color", ImgTrans);

cvWaitKey(0);

\_\_m128i unos = \_mm\_set1\_epi8(1);

for (int pixel = 0; pixel < 256; pixel++) {

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

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

\_\_m128i \*pImg1 = (\_\_m128i \*) (Img1->imageData + fila \* Img1->widthStep);

\_\_m128i \*pImg2 = (\_\_m128i \*) (Img2->imageData + fila \* Img2->widthStep);

\_\_m128i \*pImgBlack = (\_\_m128i \*) (ImgBlack->imageData + fila \* ImgBlack->widthStep);

\_\_m128i \*pImgTrans = (\_\_m128i \*) (ImgTrans->imageData + fila \* ImgTrans->widthStep);

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

\*pImg1 = \_mm\_subs\_epu8(\*pImg1, unos);

\*pImgBlack = \_mm\_adds\_epu8(\*pImgBlack, unos);

\*pImgBlack = \_mm\_min\_epu8(\*pImgBlack, \*pImg2);

\*pImgTrans = \_mm\_max\_epu8(\*pImg1, \*pImgBlack);

pImg1++;

pImg2++;

pImgBlack++;

pImgTrans++;

}

}

cvShowImage("color", ImgTrans);

cvWaitKey(1);

}

cvWaitKey(0);

// memory release for img before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow(argv[1]);

return EXIT\_SUCCESS;

}