High Computer science school - Ourense(ESEI), University of Vigo.

Parallel Architectures

-Follow-up of Image 2.0-
-Raul Villar Ramos-

Information and Resources:
Knowledge acquired from programming in previous courses and the documentation provided in Faitic

OpenCv Manual: http://docs.opencv.org/opencv2refman.pdf

Environment used: CodeBlocks / Windows 7 - 64 bits

Design used and how I came to my solution:
After analyzing the requirements of the practice, I realized that to look for a certain image in a image, the images are not always 100% identical so I decided to redo my whole program and do the following:

1-Create a mask of the objective image and compare it with the whole background image, and we are left with the lowest difference of all.

2-For a more comfortable reading and code organization, I tried to do almost everything with functions and a procedure.

3-One of them will be an inline function of type uchar named getXY that I will use to create pointers to the images.

Step 1:

Once started the program and enter the images does the following:
Create the ladybug mask and save it.
I do it through the function Fmascara() I get the mask and store directly without returning anything.

Step 2:

We searched for the first time the objective, in my case I got it started in
x - 295
y- 216
Now I call getDifference() which is a function that works as follows:
First I create a copy of the background to cover it with the mask.
Then with a loop I dedicate myself to assemble the mask of the origin image with the background.
Once I finish merging them, my function calculates Difference that compares my montage with that frame and tells me the current difference.

Step 3:

Start main loop,
Assuming that we started from the previous position of the ladybug, I look for an area of from -10 to +10 pixels both horizontally and vertically and if at any moment the difference is less than 100,000 we assume that we find it.

We break the loop and invoke my drawing function PintaCuadrado()
where I use my own function of open cv cvRectangle

The loop repeats while there are still frames.
Example:
While (Frame = cvQueryFrame (VideoFondo))
While the video contains frames we will recrend it using:
IplImage * Frame = cvQueryFrame (VideoFondo)
What this call does is to save in Frame the next frame if it exists, if it does not exist, the loop ends.
Once the loop is finished we wait for a key and we release memory and we eliminate the window.

Notes:
I have used parallelism in the CalculateDifference function, since it is the heaviest and most repetitive work.
A brief explanation of it:
I create 2 blocks of 128 bits for comparisons.
I tell to program if the ladybug is longer than the bottom it comes out and returns -1 because it would be an error.
Start the loop with parallelism:
Upload the images in your block,
I make the absolute sum (_mm_sad_epu8 (blo1, bloqu2);
I keep the result and once the total sum is returned it is returned;

I tried to order the code well and comment as much as possible in addition to using easily recognizable names for a more comfortable compression.

I also tried to segment the code into functions to make it more versatile.

Raúl Villar Ramos