Tarea 9

Angel Manrique Pozos Flores; N.C M07211505
Instituto Tecnológico Nacional de México, Blvd. Industrial, Mesa de Otay, 22430 Tijuana, B.C., México.
(Dated: 22 de Febrero del 2016)

En el presente trabajo se implementa el operador Prewitt y se pretende compararlo con el desarrollado por OpenCV, pero este operador no existe como función en OpenCV por lo tanto nos enfocaremos en la realización del operador y su aplicación.

I. INTRODUCCIÓN

El procesamiento de imágenes es de gran utilidad en la mayoría de las áreas de investigación ya que todo lo que captamos en el mundo la gran mayoría de los datos provienen de nuestros sentidos en especial la vista.

Por ello el estudio de la visión computacional es de gran importancia, el presente trabajo se hace un acercamiento a esta área, utilizando las librerías de visión open source OpenCV y el software libre CodeBlocks el cual se configuro para que pudiera ser capaz de reconocer las librerías de OpenCV.

II. OPERADORES PREWITT PARA LA DETECCION DE BORDES

El operador Prewitt calcula el gradiente de la intensidad de la imagen en cada punto, dando la dirección del mayor incremento de pixeles con mayor intensidad luminosa *light* a los pixeles con menor intensidad luminosa (*dark*) así como la razón de cambio de esa dirección.

Donde el gradiente lo podemos definir como la diferencia entre el valor de la posición anterior y el valor de la posición siguiente teniendo las siguientes aproximaciones.

$$\begin{aligned} G_x(x,y) &\approx f(x+1,y) - f(x-1,y) \\ G_y(x,y) &\approx f(x,y+1) - f(x,y-1) \end{aligned} \tag{1}$$

Este resultado nos muestra como cambia abruptamente la imagen en esos puntos de un pixel de mayor o menor intensidad a su contraparte uno sobre el eje x y otro sobre el eje y, estos kernels de Prewitt se definen como se observa en la Figura 1.

Para el caso del kernel vertical este sera sensitivo a los cambios en el eje y y el kernel horizontal sera sensitivo a los cambios en el eje x, donde la magnitud y la direccion estaran dadas por:

$$\begin{aligned} G(x,y) &= \sqrt{(G_x)^2 + (G_y)^2} \\ \Theta &= \arctan\left(\frac{G_y}{G_x}\right) \end{aligned} \tag{2}$$

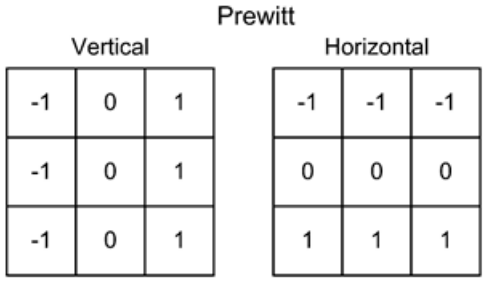


Figura 1: Kernels de Prewitt los cuales presentan cierta similitud a los utilizados por Sobel.

Para reducir la complejidad del calculo de la raiz cuadrada, la magnitud del gradiente es aproximada por la suma absoluta de sus operadores:

$$G = \mid G_x \mid + \mid G_y \mid \tag{3}$$

Teniendo en cuenta estos operadores,se utilizo la imagen de la Figura 2 para realizar las operaciones y detectar los bordes de la imagen.

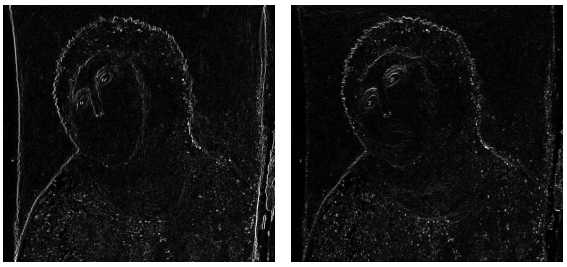
Figura 2: Imagen original para aplicar los operadores de Prewitt.

Para ello se desarrollo el siguiente código en C++ utilizando las librerías de visión de OpenCV.

```
#include <stdlib.h>
#include <cv.hpp>
#include <cxcore.hpp>
```

```
#include <highgui.h>
#include <iostream>
#include <opencv2/imgproc/imgproc.hpp>

using namespace cv;
using namespace std;


int main()
{
    Mat img, gray, Kernelx, Kernely,
        grad, grad_x, grad_y, abs_grad_x,
         abs_grad_y;

    img = imread("ecce.jpg");
    cvtColor(img, gray, CV_RGB2GRAY);


     Kernely = (Mat_<double>(3, 3) <<
         -1, 0, 1,
         -1, 0, 1,
         -1, 0, 1);

    Kernelx = (Mat_<double>(3, 3) <<
         -1, -1, -1,
          0,  0,  0,
          1,  1,  1);

    filter2D(gray, grad_x, CV_16S,
        Kernelx, Point(-1, -1));
    filter2D(gray, grad_y, CV_16S,
        Kernely, Point(-1, -1));

    convertScaleAbs(grad_x, abs_grad_x);
    convertScaleAbs(grad_y, abs_grad_y);

    addWeighted(abs_grad_x, 0.5,
        abs_grad_y, 0.5, 0, grad);

    imshow("Original Image", img);
    imshow("Horizontal Gradient",
        abs_grad_x);
    imshow("Vertical Gradient",
        abs_grad_y);
    imshow("Edge Prewitt", grad);

    cout << "Kernel Prewitt eje Y: " <<
        endl << Kernely << endl;
    cout << "Kernel Prewitt eje X: " <<
        endl << Kernelx << endl;

    waitKey();
    return 0;
}
```

Obteniendo como resultado las imagenes mostradas en la Figura 3.



(a) Imagen procesada con el kernel vertical.

(b) Imagen procesada con el kernel horizontal.

Figura 3: Imágenes procesadas con el operador Prewitt para la deteccion de bordes.

En la Figura 4 se observa el resultado obtenido de la combinacion de ambas operaciones del kernel x y el kernel y de Prewitt.

Figura 4: Imagen obtenida del procesamiento con el operador Prewitt.

En la Figura 5 se muestra los kernels utilizados en la operacion.

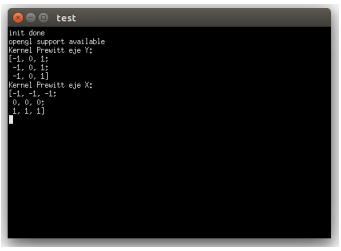


Figura 5: Kernels utilizados.

III. BIBLIOGRAFÍA

- OpenCV, Sobel Derivatives.
  ($http://tinyurl.com/jdzpxzo$)
- Programming Techniques, Sobel & Prewitt edge detector.
  ($http://tinyurl.com/j4jby4r$)
- Neighborhood Processing, Gradients, Image Edges.
  ($http://tinyurl.com/zny79fc$)