Tarea 11

Angel Manrique Pozos Flores; N.C M07211505
Instituto Tecnológico Nacional de México, Blvd. Industrial, Mesa de Otay, 22430 Tijuana, B.C., México.
(Dated: 8 de Marzo del 2016)

En el presente trabajo se implementa el detector de esquinas de Harris y un poco de teoría sobre las propiedades que presenta la matriz Hessiana.

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. MATRIZ HESSIANA

Se define la notacion a utilizar para representar las derivadas parciales.

$$\begin{aligned} f_{xx} &= \frac{\partial^2 f}{\partial x^2} \\ f_{xy} &= \frac{\partial^2 f}{\partial x \partial y} \\ f_{yy} &= \frac{\partial^2 f}{\partial y^2} \end{aligned} \tag{1}$$

Donde podemos definir una matriz Hessiana de dos variables de la siguiente forma:

$$\begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial y \partial x} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \tag{2}$$

También podemos definir la matriz Hessiana de "n"variables como:

$$\begin{bmatrix} f_{xx} & f_{xy} & \cdots & f_{xn} \\ f_{xy} & f_{yy} & \cdots & f_{yn} \\ \vdots & \vdots & & \vdots \\ f_{nx} & f_{ny} & \cdots & f_{nn} \end{bmatrix} \tag{3}$$

Una manera de decidir si los puntos críticos son máximos, mínimos o puntos de silla para una función basada en el estudio de las derivadas segundas y en particular de la matriz Hessiana.

Sea A $\subseteq \mathbb{R}^2$ un conjunto abierto y f: A $\to \mathbb{R}$ una funcion con derivadas segundas continuas en A.

$$H(x_1, \ldots, x_n) = \begin{bmatrix} \frac{\partial^2 f(x_1,\ldots,x_n)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x,\ldots,x_n)}{\partial x_n \partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial^2 f(x_1,\ldots,x_n)}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f(x_1,\ldots,x_n)}{\partial x_n^2} \end{bmatrix} \tag{4}$$

En el caso de funciones de dos variables, el signo de los valores propios de la matriz Hessiana (la cual es simétrica) permite decidir si un punto es un máximo relativo, un mínimo relativo o un punto de silla, se tiene que:

$$H(x, y) = \begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial y \partial x} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix} \tag{5}$$

Sea f una funcion de dos variables definida en un conjunto abierto A $\subseteq \mathbb{R}^2$ y continua con sus derivadas hasta el tercer orden. Sea un punto $P_0 = (x_0, y_0) \in$ I : $\nabla f(P_o) = 0$ (es decir que P_0 es un punto critico para f), entonces se puede definir que:

- Si los valores propios de H_{P_0} son positivos, P_0 es un punto de minimo relativo;
- Si los valores propios de H_{P_0} son negativos, P_0 es un punto de maximo relativo;
- Si uno de los valores propios de H_{P_0} es positivo y el otro negativo, P_0 es un punto de silla.

Ejemplo:
Si se tiene la funcion:

$$f(x, y) = x^3 + y^3 - 3xy. \tag{6}$$

Con las correspondientes derivadas.

$$\begin{aligned} f'_x &= 3x^2 - 3y \\ f'_y &= 3y^2 - 3x \end{aligned} \tag{7}$$

Y los puntos criticos estan en (1,1) y (0,0).
Calculando las segundas derivadas.

$$\begin{aligned} f''_{xx} &= 6x \\ f''_{xy} &= -3 \\ f''_{yy} &= 6y \end{aligned} \tag{8}$$

La matriz Hessiana en los puntos criticos es:

$$f(0,0)'' = \begin{bmatrix} 0 & -3 \\ -3 & 0 \end{bmatrix} \leqslant 0,$$

$$f(1,1)'' = \begin{bmatrix} 6 & -3 \\ -3 & 6 \end{bmatrix} > 0 \tag{9}$$

Siendo (1,1) un punto minimo y (0,0) un punto de silla.

III. HARRIS CORNER DETECTOR

En visión computacional, usualmente es necesario encontrar puntos específicos entre diferentes imágenes de un mismo sistema, estos comúnmente les llamamos puntos de interés.

Y las principales características de una imagen, por mencionar solo algunas son:

- Bordes
- Esquinas (Puntos de interés)
- Contornos (Regiones de interés)

Aquí veremos exclusivamente el detector de esquinas.

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \tag{10}$$

Donde:

- w(x,y) = es la ventana en la posición (x,y)
- I(x,y) = es la intensidad en (x,y)
- I(x+u,y+v) es la intensidad en la ventada en movimiento (x+u,y+v)

Donde se busca encontrar esquinas en las ventanas, donde existirá una variación muy grande en su intensidad, donde hay que maximizar la ecuación (1), específicamente el termino:

$$\sum_{x,y}[I(x+u, y+v) - I(x,y)]^2 \tag{11}$$

Usando la expansión de Taylor:

$$E(u,v) \approx \sum_{x,y}[I(x+u, y+v) - I(x,y)]^2 \tag{12}$$

Expandiendo la ecuación y cancelando:

$$E(u,v) \approx \sum_{x,y} u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2 \tag{13}$$

Donde puede ser representada en la forma matricial:

$$E(u,v) \approx [u,v]\left(\sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \tag{14}$$

Definiendo:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{15}$$

Ahora nuestra ecuación es:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{16}$$

Donde un valor es calculado para cada ventana, para determinar la posibilidad de que exista una esquina lo podemos definir entonces como:

$$R = det(M) - k(trace(M))^2 \tag{17}$$

Donde:

- det(M) = $\lambda_1\lambda_2$
- trace(M) = $\lambda_1 + \lambda_2$

Donde un ventana con un valor R grande puede considerarse como una esquina.

Se utilizo la imagen de la Figura 1 para realizar las operaciones y detectar las esquinas de la imagen.

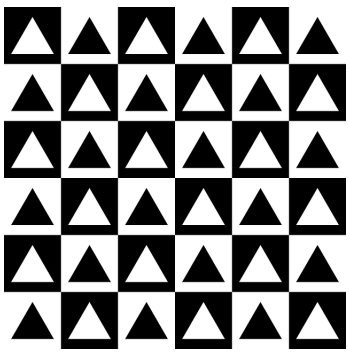
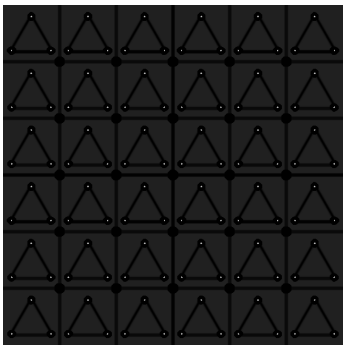


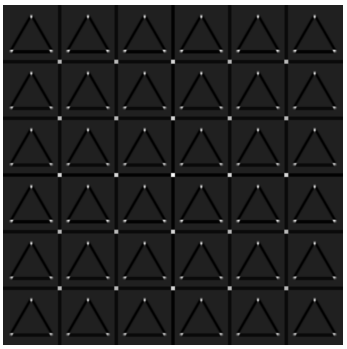
Figura 1: Imagen original.

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>


using namespace cv;
using namespace std;

// Global Variables
Mat img, gray;
int thresh = 200;
int max_thresh = 255;

char* source_window = "Original Image";
char* corners_window = "Harris Detector"
    ;

// Function header
void cornerHarris( int, void* );


int main( int argc, char** argv )
{
  // Original Image
  img = imread( "chessb.jpg", 1 );
  cvtColor( img, gray, CV_BGR2GRAY );


  // Create a window and a trackbar
  namedWindow( source_window,
     CV_WINDOW_AUTOSIZE );
  createTrackbar( "Threshold: ",
     source_window, &thresh, max_thresh,
      cornerHarris );
  imshow( source_window, img );

  cornerHarris( 0, 0 );

  waitKey(0);
  return(0);
}



void cornerHarris( int, void* )
{
    Mat dst, dst_norm, dst_norm_scaled;
    dst = Mat::zeros( img.size(),
        CV_32FC1 );

    // Detector parameters
    int blockSize = 2;
    int apertureSize = 3;
    double k = 0.04;

    // Detecting corners
    cornerHarris( gray, dst, 7, 5, 0.05,
        BORDER_DEFAULT );

    // Normalizing
    normalize( dst, dst_norm, 0, 255,
       NORM_MINMAX, CV_32FC1, Mat() );
    convertScaleAbs( dst_norm,
        dst_norm_scaled );

    // Drawing a circle around corners
    for( int j = 0; j < dst_norm.rows ;
        j++ )
    { for( int i = 0; i < dst_norm.cols;
         i++ )
    {
        if( (int) dst_norm.at<float>(j,i
            ) > thresh )
        {
            circle( dst_norm_scaled,
               Point( i, j ), 5, Scalar
               (0), 2, 8, 0 );
        }
    }
    }

    // Showing the result
    namedWindow(source_window,
        CV_WINDOW_AUTOSIZE );
    imshow( source_window, img );

    namedWindow( corners_window,
        CV_WINDOW_AUTOSIZE );
    imshow( corners_window,
        dst_norm_scaled );

}
```



(a) Imagen con Detector Harris aplicado con un thresholding de 130.



(b) Imagen con Detector Harris aplicado con un thresholding de 255.

Figura 2: Imágenes obtenidas de la aplicación del detector de Harris, a las cuales a sido variado su valor de threshold.

IV. BIBLIOGRAFÍA

- Matriz Hessiana, aplicaciones de formas cuadráticas en calculo de varias variables. ($http://tinyurl.com/jyuczvu$)
- Máximos y mínimos de funciones de varias variables. ($http://tinyurl.com/ztka9pu$)
- OpenCV, Harris corner detector. ($http://tinyurl.com/gv42eaw$)