

# Tarea 8

Angel Manrique Pozos Flores; N.C M07211505

Instituto Tecnológico Nacional de México, Blvd. Industrial, Mesa de Otay, 22430 Tijuana, B.C., México.

(Dated: 18 de Febrero del 2016)

En el presente trabajo se implementa las operaciones de mediana, mínimo y máximo sobre una imagen como se vio en clase.

## 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. FILTROS

El suavizado (*smoothing*) tambien llamado desenfoque (*blurring*) es una operacion simple y que comunmente es utilizada en el procesamiento de imagenes.

Existen diversas aplicaciones para el suavizado, la mas comunmente utilizada es para la reduccion de ruido, donde para poder llevar a cabo el suavizado aplicaremos un filtro a nuestra imagen los tipos mas comunmente utilizados son los *filtros lineales* en donde el valor del pixel de su salida (p.ej.  $g(i,j)$ ) es determinado por la suma ponderada de los valores de los pixeles de entrada (p.ej.  $f(i+k,j+l)$ )

$$g(i,j) = \sum_{k,l} f(i+k,j+l)h(k,l) \quad (1)$$

Donde  $h(k,l)$  es llamado como el *kernel*, este arreglo contiene los coeficientes del filtro, existen una gran variedad de filtros pero los usados en este trabajo seran:

- Filtro mediana.
- Filtro valor máximo.
- Filtro valor mínimo.

### II.1. Filtro Mediana

Este filtro corre sobre cada elemento de la señal, en nuestro caso una imagen, reemplazando cada pixel con el valor de la **mediana** de los pixeles de su vecindad.

### II.2. Filtro Mínimo

Esta utiliza el valor mínimo de los pixeles de su vecindad y lo sustituye en el valor central de su pixel, esta función erosiona la imagen de origen utilizando un elemento estructural específico el cual determinara la zona sobre la cual se tomara el mínimo.

### II.3. Filtro Máximo

Utiliza el valor máximo de los pixeles de su vecindad para hacer la sustitución del valor central de su pixel, esta función dilata la imagen de origen utilizando un elemento estructural específico el cual determinara la zona sobre la cual se tomara el máximo.

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 std;
using namespace cv;
```

```
int main()
{
```

```
    Mat img = imread("lena.jpg",
        CV_LOAD_IMAGE_COLOR);
    Mat dst1, dst2, dst3;
```

```
    namedWindow("Original Image",
        CV_WINDOW_AUTOSIZE);
    imshow("Original Image", img);
```

```
    for (;;)
    {
```

```
        // Median
        medianBlur(img, dst1, 9);
```

```

imshow("Median", dst1);

// Max
dilate(img, dst2, Mat(), Point
      (-1, -1), 3, 1, 1); //Mat by
      default create a 3x3 array
      kernel.
imshow("Max", dst2);

// Min
erode(img, dst3, Mat(), Point
      (-1, -1), 3, 1, 1);
imshow("Min", dst3);

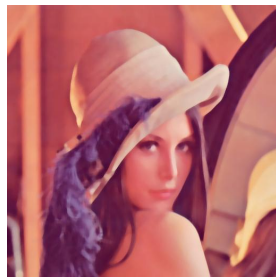
waitKey(0);
return 0;
}

```

Obteniendo como resultado las siguientes imágenes.



(a) Imagen original de lena.



(b) Imagen con *mediana* aplicada.

Figura 1: Imágenes procesadas, donde se observa que el ruido (*puntos blancos*) a sido eliminado al aplicarle una apertura a la imagen.



(a) Imagen con filtro *mínimo* aplicado.

(b) Imagen con filtro *máximo* aplicado.

Figura 2: Imágenes procesadas, donde se observa que el ruido dentro de nuestro objeto a sido eliminado aplicando los presentes filtros.

### III. BIBLIOGRAFÍA

- OpenCV, Smoothing Images.  
(<http://tinyurl.com/hxvk7es>)
- OpenCV 3.1.0 -Dev, Smoothing Images.  
(<http://tinyurl.com/z4uzqnu>)
- Codezone4, How to use order statistic filters with OpenCV.  
(<http://tinyurl.com/jafl2nr>)