

## Tarea 5

Angel Manrique Pozos Flores; N.C M07211505

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

(Dated: 7 de Febrero del 2016)

En el presente trabajo se implementa un algoritmo aplicando los operadores morfológicos de erosión y dilatación, así como la variación del tamaño de estos operadores.

### 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. OPERACIONES MORFOLÓGICAS

Las operaciones morfológicas se utilizan principalmente para procesar imágenes basándose en su forma, estas operaciones se aplican mediante un *elemento estructural* a una imagen base de entrada, generando una imagen de salida.

Las operaciones morfológicas mas básicas son dos principalmente, la dilatación y la erosión, estas tienen una gran cantidad de aplicaciones dentro de la visión computacional por ejemplo:

- Remover ruido de una imagen.
- Aislamiento de elementos individuales dentro de una imagen.
- Unión de elementos disconjuntos dentro de una imagen.
- Encontrar huecos o topes de intensidad en una imagen.

Usaremos la Figura 1 como ejemplo practico para explicar la dilatación y la erosión.



Figura 1: Imagen original.

#### II.1. Dilatación

Este operador consiste en la convolucion de una imagen  $A$  con un kernel  $B$  el cual puede tener cualquier forma y tamaño usualmente un cuadrado o un circulo.

El kernel  $B$  tiene un punto central definido el cual usualmente es su centro, este kernel escanea toda la imagen tomando el valor máximo de cada pixel reemplazando la posición del punto central del kernel con su valor máximo, causando que las regiones mas brillantes de una imagen "crezcan" o se "dilaten".



Figura 2: Imagen dilatada, donde el fondo de color blanco se dilata alrededor de las regiones negras de la letra.

#### II.2. Erosión

Esta operación es la hermana de la dilatación, lo que hace es que computacionalmente genera un mínimo local sobre el área del kernel, donde el kernel  $B$  se escanea sobre la imagen, computando el pixel con el valor mínimo superpuesto por  $B$  y reemplazado por el pixel de la imagen bajo el punto de anclaje con ese valor mínimo.



Figura 3: Imagen erosionada, donde el fondo blanco se reduce y las zonas mas oscuras que hacen la letra se ensanchan o crecen.

### II.3. Desarrollo

En la Figura 4 se muestra la imagen de un gatito al cual se le aplicaran estas operaciones morfológicas, variando los tipos de elementos estructurales y en la segunda parte de la practica vamos a variar el tamaño del elemento estructural para observar que sucede sobre nuestra imagen.



Figura 4: Imagen de prueba.

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

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

using namespace std;
using namespace cv;

int main()
{
    Mat img, dst1, dst2, dst3, dst4,
        dst5, dst6;

    img = imread("gatito.jpg",
        CV_LOAD_IMAGE_COLOR);

    //Structural Elements
    int erosion_s1 = 4;
    Mat Erect = getStructuringElement
        (cv::MORPH_RECT,
         cv::Size(2 * erosion_s1 +
                 1, 2 * erosion_s1 + 1),
         cv::Point(erosion_s1,
                   erosion_s1));

    int erosion_s2 = 4;
    Mat Ecross =
        getStructuringElement(cv::
        MORPH_CROSS,

        cv::Size(2 * erosion_s2 +
                 1, 2 * erosion_s2 + 1),
        cv::Point(erosion_s2,
                  erosion_s2));

    int erosion_s3 = 4;
    Mat Eelli = getStructuringElement
        (cv::MORPH_ELLIPSE,
         cv::Size(2 * erosion_s3 +
                 1, 2 * erosion_s3 + 1),
         cv::Point(erosion_s3,
                   erosion_s3));

    // Apply Erosion on the image
    erode(img, dst1, Erect);
    erode(img, dst2, Ecross);
    erode(img, dst3, Eelli);

    //Apply Dilation on the Image
    dilate(img, dst4, Erect);
    dilate(img, dst5, Ecross);
    dilate(img, dst6, Eelli);

    cout << "Morphological Operators:
        \n 0: Erode \n 1: Dilation"
        << endl;

    /// SHOW IMAGES

    namedWindow("Display Original",
        CV_WINDOW_AUTOSIZE);
    imshow("Display Original", img);

    //Erode
    namedWindow("Erode Rectangle",
        CV_WINDOW_AUTOSIZE);
    imshow("Erode Rectangle", dst1);

    namedWindow("Erode Cross",
        CV_WINDOW_AUTOSIZE);
    imshow("Erode Cross", dst2);

    namedWindow("Erode Ellipse",
        CV_WINDOW_AUTOSIZE);
    imshow("Erode Ellipse", dst3);

    //Dilation
    namedWindow("Dilate Rectangle",
        CV_WINDOW_AUTOSIZE);
    imshow("Dilate Rectangle", dst4);

    namedWindow("Dilate Cross",
        CV_WINDOW_AUTOSIZE);
    imshow("Dilate Cross", dst5);

    namedWindow("Dilate Ellipse",
        CV_WINDOW_AUTOSIZE);
```

```

imshow("Dilate Ellipse", dst6);

waitKey(0);
return 0;
}

```

Obteniendo las imágenes siguientes.



(a) Dilatación con un elemento estructural cuadrado de orden 4.



(b) Dilatación con un elemento estructural circular de orden 4.



(c) Dilatación con un elemento estructural tipo cruz de orden 4.

Figura 5: Imágenes con diversos elementos estructurales con dilatación aplicada.



(a) Erosión con un elemento estructural cuadrado de orden 4.



(b) Erosión con un elemento estructural circular de orden 4.



(c) Erosión con un elemento estructural tipo cruz de orden 4.

Figura 6: Imágenes con diversos elementos estructurales con erosión aplicada.

#### II.4. Cambiando el tamaño de nuestros operadores morfológicos

Se modificó un poco el código para que mostrara el mismo elemento estructural (*cuadrado*), variando su tamaño para poder apreciar la erosión y dilatación más perfectamente.

En la Figura 7 se muestra los cambios en la dilatación cuando se varía el tamaño de su elemento estructural.



(a) Dilatación con un elemento estructural cuadrado de orden 2.



(b) Dilatación con un elemento estructural cuadrado de orden 8.



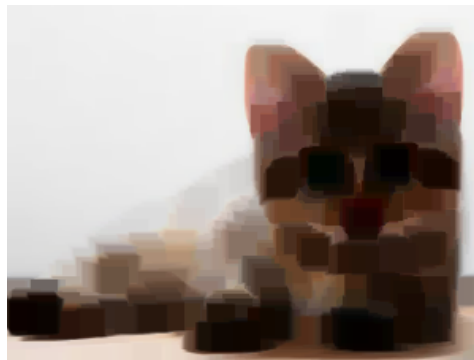
(c) Dilatación con un elemento estructural cuadrado de orden 12.

Figura 7: Imágenes con dilatación aplicada, variando el tamaño de su elemento estructural.

En la Figura 8 de igual manera observamos los cambios en su erosión cuando variamos el tamaño de su elemento estructural.



(a) Erosión con un elemento estructural cuadrado de orden 2.



(b) Erosión con un elemento estructural cuadrado de orden 8.



(c) Erosión con un elemento estructural cuadrado de orden 12.

Figura 8: Imágenes con erosión aplicada variando el tamaño de su elemento estructural.

### III. BIBLIOGRAFÍA

- OpenCV, Eroding and dilating.  
(<http://tinyurl.com/pbe3ksw>)