Apéndice C. Operadores adicionales.

Operador inverso.

Este operador genera una imagen inversa a la imagen original, es decir, invierte los valores de la intensidad de los niveles de gris. Para lograrlo aplica una función a cada píxel de la imagen original para obtener el valor del nuevo píxel. La función para invertir la imagen fue obtenida de [Pajares, et al., 2004] y es la siguiente:

$$IixFin = 255 - PixOrig$$

```
matOrig=convierte(imagen);
for(int i=0; i<alto; i++){
        for(int j=0; j<ancho; j++){
                m[i][j] = Math.abs(255-matOrig[i][j]);
        }
}
```

Operador umbral.

Este operador también es de tipo individual y su función es crear una imagen binaria, es decir, sólo contiene dos valores, en este caso serán blanco y negro. El nivel de transición está determinado por un valor de entrada (umbral). La función de transformación fue obtenida de [Pajares, et al., 2004] y es la siguiente:

$$PixFin = \begin{cases} 0, PixOrig \leq umbral \\ 255, PixFin > umbral \end{cases}$$

```
matOrig = convierte(imagen);
for(int i=0; i<alto;i++){
        for(int j=0; j<ancho; j++){
```

```
                if(matOrig[i][j]<umbral){
                        m[i][j] = 255;
                }
                if(matOrig[i][j]>=umbral){
                        m[i][j] = 0;
                }
        }
}
```

Operador suma y resta.

Ambos operadores son del tipo individual, debido a que reciben dos imágenes y obtienen el valor del píxel final por medio de la suma o resta de los píxeles de las imágenes originales en la misma posición.

```
try{
        mat1 = ti.convierte(i);
        mat2 = ti.convierte(f);
}catch(Exception e){
        System.out.println("ERROR"+e.toString());
}
for(int k=0; k< alto ; k++){
        for(int l=0; l< ancho ; l++){
                //En el caso de suma
                m[k][l] = (mat1[k][l] + mat2[k][l])/2;
                //en el caso de resta
                m[k][l] = Math.abs( (mat1[k][l] - mat2[k][l])%256);
        }
}
```