

## Apéndices.

### Apéndice A.

Pseudocódigo de la función "user\_setgray".

```

Herror IPBSetGray(Hproc_handle  proc_id,
                 char           *in_file, // archivo de entrada con valores de gris
                 INT            width,  // ancho de la imagen de salida
                 INT            height, // altura de la imagen de salida
                 HBYTE          *im_out) // imagen resultante
{
    FILE *filePoint;
    int GrayVal;

    /* abrir el archivo in_file*/
    filePoint = fopen( in_file, "r" );

    for (int i=0; i<=height-1; i+=1)
    {
        for (int j=0; j<=width-1; j+=1)
        {
            //leer desde archivo los valores y ponerlos a la matriz de imagen
            fscanf(filePoint, "%d", &GrayVal);
            im_out[i*width+j] = GrayVal;
        }
    }
    fclose(filePoint);
    return(H_MSG_OK); //regresar la imagen en formato
}

```

## Pseudocódigo del método basado en gradiente.

Pseudocódigo del método consiste en cuatro funciones.

1) Aplicar el operador Sobel y convertir la imagen a un conjunto de píxeles.

```
/*lee la imagen desde el archivo */
read_image(&Ip070,"E:\\Archivos de
programa\\MVTec\\Halcon\\images\\bvlab_2p\\2p_070.tif");
/*saca el apuntador con el tipo de imagen y su tamaño*/
get_image_pointer1(Ip070,&Pointer,&Type,&Width,&Height);
/*abre la ventana para la imagen */
open_window(0,0,Width,Height,0,"visible","",&WindowID);
/*despliega la imagen */
disp_obj(Ip070,WindowID);
/*espera el clic en la ventana */
get_mbutton(WindowID,&Row,&Column,&Button);
/*gradiente Sobel para la amplitud y dirección */
sobel_dir(Ip070,&EdgeAmplitude,&EdgeDirection,"sum_abs",3);
disp_obj(EdgeAmplitude,WindowID);
get_mbutton(WindowID,&Row,&Column,&Button);
/*saca la inversa de la imagen */
invert_image(EdgeAmplitude,&EdgeInvert);
disp_obj(EdgeInvert,WindowID);
get_mbutton(WindowID,&Row,&Column,&Button);
/*escribe la imagen al archivo */
write_image(EdgeInvert,"tiff",0,"E:\\Tania\\halcon\\EdgeMoveImage\\Edge070");
/*abre el archivo de texto para los píxeles */
open_file("E:\\Tania\\tesis\\points\\point070.txt","output",&FileHandle);
for (i=0; i<=Height-1; i+=1)
{
    for (j=0; j<=Width-1; j+=1)
    {
        /*convierte el punto al píxel */
        get_grayval(EdgeAmplitude,i,j,&GrayVal);
        fwrite_string(FileHandle,GrayVal);
        fwrite_string(FileHandle," ");
        if (j==(Width-1))
        {
            fnew_line(FileHandle);
        }
    }
}
/*cierra el archivo de texto */
close_file(FileHandle);
/*cierra la ventana */
```

```
close_window(WindowID);
```

2) Obtener el movimiento de dos imágenes de secuencia [Jain, 1995].

```
for(i=0; i<yPoint; i++)
  for(int j=0; j<xPoint; j++)
  {
    differ = abs(point1[i][j] - point2[i][j]); // la diferencia absoluta entre dos imágenes
    mult = point1[i][j] * differ; // multiplicación de primer imagen a la diferencia
    if (maxMult<mult) //búsqueda del valor máximo en multiplicación
      maxMult = mult;
    if(minMult>mult) //búsqueda del valor mínimo en la multiplicación
      minMult = mult;
    matrixMult[i][j] = mult; // creación de la matriz de valores obtenidos
  }
differ = maxMult - minMult; /* diferencia entre el máximo y el mínimo para la
                             normalización*/
for(i=0; i<yPoint; i++)
  for(int j=0; j<xPoint; j++)
  {
    mult = (255.0/differ)*matrixMult[i][j]; // normalización de valores
    fprintf(finalFile,"%d ", mult); //escritura al archivo el resultado
    if(j==xPoint-1)
      fprintf(finalFile, "\n");
  }
}
```

3) Convertir el conjunto de valores de gris obtenido en el paso dos a la imagen.

```
read_image(&Ip070,"E:\\Archivos de
programa\\MVTec\\Halcon\\images\\bvlab_2p\\2p_070.tif");
get_image_pointer1(Ip070,&Pointer,&Type,&Width,&Height);
open_window(0,0,Width,Height,0,"visible","",&WindowID);
disp_obj(Ip070,WindowID);
get_mbutton(WindowID,&Row,&Column,&Button);
/*convierte los pixeles del archivo a la imagen */
user_setgray(Ip070,&ImageOut,"E:\\Tania\\tesis\\points\\pointMove1.txt");
/*destaca los bordes*/
threshold(ImageOut,&Region,20,255);
/*convierte la región a la imagen */
region_to_bin(Region,&BinImage,5,255,Width,Height);
write_image(BinImage,"tiff",0,"E:\\tania\\halcon\\EdgeMoveImage\\BinImage1");
disp_obj(BinImage,WindowID);
get_mbutton(WindowID,&Row,&Column,&Button);
close_window(WindowID);
```

4) Despliega dos imágenes obtenidos para ver más claro el movimiento.

```
read_image(&BinImage1,"E:\\Tania\\halcon\\EdgeMoveImage\\BinImage1.tif");
read_image(&BinImage2,"E:\\Tania\\halcon\\EdgeMoveImage\\BinImage2.tif");
get_image_pointer1(BinImage1,&Pointer,&Type,&Width,&Height);
open_window(0,0,Width,Height,0,"visible","",&WindowID);
read_image(&Ip070,"E:\\Archivos de
programa\\MVTec\\Halcon\\images\\bvlab_2p\\2p_070.tif");
disp_obj(Ip070,WindowID);
get_mbutton(WindowID,&Row,&Column,&Button);
threshold(BinImage1,&Region1,5,5);
/*pone el color de la ventana */
set_color(WindowID,"red");
/*despliega la región */
disp_region(Region1,WindowID);
threshold(BinImage2,&Region2,5,5);
set_color(WindowID,"blue");
disp_region(Region2,WindowID);
get_mbutton(WindowID,&Row,&Column,&Button);
close_window(WindowID);
```