

CAPÍTULO 2

PROCESAMIENTO DE IMÁGENES

2.1 Imágenes Digitales

Desde el punto de vista más simple, hoy una imagen digital consiste de lo que se puede ver como una matriz de miles o millones de píxeles cada uno con un valor representado por cierta cantidad de bits que corresponde al código de un color. Un píxel es el elemento mínimo de una imagen y se puede representar cada uno con las coordenadas (x,y) dentro de la imagen considerada como una matriz. Para un píxel en específico usualmente se emplea $P(x, y)$ o simplemente (x, y) [9]. Así es como una imagen digital a color, RGB (Red-Green-Blue, *true color*), está definida por un arreglo tridimensional de m renglones por n columnas por 3 espacios donde se especifica el valor de la intensidad de los colores rojo, verde y azul que definen a cada píxel; una imagen digital de intensidad (*grayscale*) está definida por un arreglo bidimensional de m renglones por n columnas donde cada celda contiene el valor que define la intensidad en la escala de grises de cada píxel; una imagen binaria está definida por un arreglo bidimensional de m renglones por n columnas donde cada celda sólo puede tomar dos valores, 1 (blanco = on) para los píxeles que forman parte de la imagen y 0 (negro = off) para los píxeles que conforman el fondo de la imagen [12].

Las imágenes digitales pueden obtenerse de varias formas: fotografías digitales, imágenes creadas por computadora, imágenes escaneadas, imágenes obtenidas interactivamente donde un sistema captura lo que un usuario plasma en una pantalla

especial, etc.

2.2 Procesado de Imágenes

El procesado de imágenes ha resultado ser una herramienta con un alto potencial en un sin fin de aplicaciones en donde se manipula una imagen digital por lo regular para obtener las características más importantes que permitan trabajar con ella eficientemente.

De una imagen digital se pueden seguir varios procesamientos con distintos fines como convertir una imagen RGB a *grayscale*, de una imagen *grayscale* obtener una imagen binaria que es en la que más se detectan características importantes principalmente bidimensionales. A la imagen binaria sólo llegan las áreas que más resaltan en una imagen RGB o *grayscale* y aquí se pueden hacer una gran variedad de procesamientos para eliminar el ruido, rellenar los bordes, detectar contornos, etc. Además también se puede trabajar en otra forma con la imagen como lo sería el manipular los píxeles con algún fin como obtener la transformada de distancia [9, 12].

2.2.1 Detección de Contorno

La detección de contorno se aplica principalmente a imágenes *grayscale* y binarias. Sirve para quitar el “relleno” a todas las áreas que componen la imagen. Los métodos de detección de contorno buscan en una imagen aquellas regiones donde la intensidad de los píxeles cambia rápidamente o extremadamente ya que es el borde entre un objeto y el fondo [12].

El método más poderoso de detección de contornos es el de *canny* porque es el menos “engañado” por el ruido que puede tener una imagen *grayscale* o binaria. Este método busca las regiones donde la intensidad cambia rápido y no tan rápido y luego checa si las regiones donde la intensidad no cambia tan rápido están junto a las que si lo hacen para ver si pertenecen al borde o se trata simplemente de ruido. El método *canny* obtiene una imagen binaria con los contornos detectados [9, 12]. Este método y algunos otros se encuentran dentro de las librerías de Matlab, lo que hace más fácil la realización de este proyecto al no elaborar un programa cuya función sea detectar contornos que es un tanto complicado.

2.2.2 Transformada de Distancia

La transformada de distancia es un algoritmo que trabaja con los valores de los píxeles de una imagen binaria. El fin es calcular las distancias representativas que todos los píxeles tienen con respecto a un objeto dentro de una imagen. La transformada de distancia de una imagen binaria es una matriz de $m \times n$, donde [12]:

 $m \times n$ es el tamaño en píxeles de la imagen o un segmento de ella,

 cada elemento de la matriz representa el píxel que está en la misma posición dentro de la imagen ((x,y) equivale a $P(x,y)$) y

 cada elemento de la matriz tiene un valor que es la distancia del píxel representado a un objeto/contorno dentro de la imagen.

Existen varias métricas para calcular el valor que un píxel debe tomar de acuerdo a la distancia que este presenta con respecto a un objeto. La más exacta es la métrica euclidiana que calcula la distancia D entre los píxeles (x_1, y_1) y (x_2, y_2) y en sí entre cualquier par de puntos con la fórmula siguiente [6]:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.1)$$

Para obtener la transformada de distancia de la imagen o un segmento de la imagen, primero se debe identificar el objeto/contorno del cual se quiere obtener su transformada de distancia. Una vez identificado el objeto/contorno, se tiene que definir una matriz m por n del mismo tamaño que la imagen o segmento. A los elementos que representan los píxeles pertenecientes al objeto/contorno se les asignará el valor 0 decimal (distancia igual a 0 al objeto) y a los demás que representan los píxeles que conforman el “fondo” de la imagen se les asignará el valor 255 decimal (distancia grande al objeto).

Como ejemplo supóngase la siguiente imagen binaria de 10x10 píxeles:

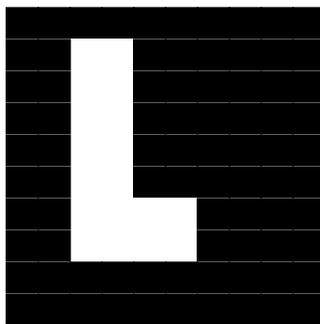


Figura 2.1. Imagen binaria de una “L”.

La imagen de la Figura 2.1 puede ser algún segmento de otra imagen que bien pudo contener más colores así como más objetos/contornos, y pudo ser procesada para extraer de ella sólo el segmento que contiene a la “L”. De esta imagen se necesita dar el primer paso hacia su transformada de distancia que es la generación de una matriz de 10x10 donde cada elemento que representa un píxel perteneciente a la “L” (píxeles blancos) tendrá el valor 0 y cada elemento que representa un píxel perteneciente al “fondo” (píxeles negros) tendrá el valor 255. Esto se logra con un programa que tenga funciones de procesado de imágenes como Matlab: se genera una matriz que se va “llenando” de acuerdo a una rutina que cheque los valores de los píxeles de la imagen: cuando se encuentra un píxel con valor 0 binario, se asigna al elemento correspondiente en la matriz el valor 255 decimal, y cuando se encuentra un píxel con valor 1 binario, se asigna al elemento correspondiente en la matriz el valor 0 decimal. Si se aplica este proceso a la imagen de la figura 2.1, nos da como resultado la matriz de la Figura 2.2.

255	255	255	255	255	255	255	255	255	255
255	255	0	0	255	255	255	255	255	255
255	255	0	0	255	255	255	255	255	255
255	255	0	0	255	255	255	255	255	255
255	255	0	0	255	255	255	255	255	255
255	255	0	0	255	255	255	255	255	255
255	255	0	0	0	0	255	255	255	255
255	255	0	0	0	0	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255

Figura 2.2. Representación de la imagen de una “L” para proceder al cálculo de su transformada de distancia.

Ya que se tiene una representación de la imagen (Figura 2.2) donde los elementos con valor 0 (distancia cero al objeto/contorno mismo) son parte del objeto/contorno se procede a calcular la distancia de los elementos que tienen el valor 255 al objeto/contorno. Para esto se ejecuta el siguiente procedimiento donde (x, y) representa el elemento actual de la matriz [9]:

- 1.- Se hace un barrido de izquierda a derecha y de arriba hacia abajo en la matriz hasta encontrar el primer elemento con valor 0.
- 2.- A partir del primer elemento con valor 0, se sigue la misma dirección del barrido aplicando los pasos 3 y 4 a cada elemento siguiente con cada uno de los elementos vecinos que se encuentran arriba y/o a la izquierda, siempre que existan.
- 3.- A cada elemento vecino de los que se indican en el operador de la Figura 2.3 se le sumará a su valor actual la distancia que tienen con el elemento de referencia como sigue:

 Si se trata de un elemento adyacente horizontal o vertical, se le sumará a su valor actual un 1.

 Si se trata de un elemento adyacente diagonal, se le sumará a su valor actual la cantidad 1.4142 que corresponde a una aproximación de la métrica euclidiana.

$x-1, y-1$	$x-1, y$	$x-1, y+1$
$x, y-1$	x, y	
$x+1, y-1$		

Figura 2.3. Primer operador de la transformada de distancia.

- 4.- Ya que se tienen los valores del paso 3 junto con el valor actual del elemento de referencia, se calcula el menor y éste será el nuevo valor que se asignará al elemento de referencia.

El proceso no acaba aquí, ya que se tienen las distancias de los elementos que se encuentran a la derecha y/o por debajo del objeto/contorno a este último. Entonces se ejecuta el proceso inverso:

- 5.- Se hace un barrido de derecha a izquierda y de abajo hacia arriba en la matriz ahora empezando por el último elemento de la matriz.
- 6.- A partir de este elemento se sigue la misma dirección del barrido aplicando los pasos 3 y 4 a cada elemento siguiente con cada uno de sus cinco elementos vecinos que se encuentran abajo y/o a la derecha, siempre y cuando existan, como indica el operador de la Figura 2.4.

		$x-1, y+1$
	x, y	$x, y+1$
$x+1, y-1$	$x+1, y$	$x+1, y+1$

Figura 2.4. Segundo operador de la transformada de distancia.

La matriz resultante será la transformada de distancia del objeto/contorno dentro de la imagen. La Figura 2.5 muestra los resultados del procedimiento anterior aplicado a la matriz de la Figura 2.2. La Figura 2.5 es la transformada de distancia de la imagen de la Figura 2.1. Cabe mencionar que cuando se tienen objetos/contornos parecidos en dos imágenes, sus transformadas serán similares, es decir, entre píxeles equivalentes habrá una diferencia pequeña en sus valores.

2.4142	1.4142	1	1	1.4142	2.4142	3.4142	4.4142	5.4142	6.4142
2	1	0	0	1	2	3	4	5	6
2	1	0	0	1	2	3	4	5	5.6568
2	1	0	0	1	2	3	3.8284	4.2426	5.2426
2	1	0	0	1	2	2.4142	2.8284	3.8284	4.8284
2	1	0	0	1	1	1.4142	2.4142	3.4142	4.4142
2	1	0	0	0	0	1	2	3	4
2	1	0	0	0	0	1	2	3	4
2.4142	1.4142	1	1	1	1	1.4142	2.4142	3.4142	4.4142
2.8284	2.4142	2	2	2	2	2.4142	2.8284	3.8284	4.8284

Figura 2.5. Transformada de distancia de la imagen de una "L".