

# Capítulo 4. Detección del Objeto Conocido, Medición de su Tamaño Virtual y Cálculo de la Distancia

## 4.1 Características del objeto a medir

El objeto a ser procesado es de dimensiones conocidas, y tiene un color uniforme en su superficie, para fácil detección y extracción de la imagen. El diámetro del objeto es medido previamente, y este diámetro corresponde al ancho en píxeles que ocupará el objeto en la imagen, y es el único dato que se conoce a priori para todo el experimento. Hasta este momento, se conocen dos de las variables de la ecuación 1.2 para encontrar la información de la distancia de la cámara al objeto; y estas variables son: el tamaño del objeto y la distancia focal, cuyo cálculo se describe en el Capítulo 3. En caso de que el objeto no sea uniforme, puede utilizarse su centro de masa<sup>1</sup> como referencia para tomar el ancho del objeto en la línea que pasa por éste.

Para que la distancia del objeto hacia el punto de observación sea aproximada lo mejor posible, hay algunos puntos que se deben tomar en cuenta al aplicar el algoritmo. Uno es que el objeto esté fijo o tenga un movimiento muy lento en caso de tratarse de secuencias de video, en orden de captar bien el objeto. Esto es aplicable al caso de un incendio que no incluye movimientos rápidos en la imagen.

---

<sup>1</sup> Centro de Masa: Es la posición geométrica de un cuerpo rígido, en la cual se puede considerar concentrada toda su masa.

Otro detalle importante que se tiene que cumplir es que el objeto no se encuentre extremadamente cerca o lejos de la cámara cuando se realiza el experimento para calcular el tamaño virtual del objeto en la imagen.

De acuerdo con la ecuación 1.1,  $\frac{L'}{f} = \frac{L}{d}$ , si la distancia  $L$  es muy grande, entonces el tamaño del objeto en la imagen será demasiado pequeño (menor al 10% del total del ancho de la imagen), lo cual traerá problemas a la hora de procesar la imagen. En cambio, si el objeto se encuentra muy cerca de la lente, entonces el tamaño del objeto será tan grande que sobrepasará el área del plano de imagen haciendo imposible su detección. Por lo tanto, se dice que el objeto a detectar y localizar debe estar en un rango válido (entre 10% y 90% del ancho total de la imagen).

Se escogió un cilindro como objeto para esta etapa del proyecto ya que tiene una forma constante desde la perspectiva de la cámara y se facilita trabajar con este; sin embargo, pudo haberse elegido otro objeto de forma regular. De igual manera un cilindro asemeja una columna de humo. En la práctica la columna de humo no es un cilindro, pero se hará uso de esta consideración.

## **4.2 Procesamiento de la imagen para detección y medición del objeto**

El algoritmo de medición de la imagen virtual consiste de varios pasos que se mencionan adelante. Dicho algoritmo se implementó en MATLAB, haciendo uso de varias funciones que tiene el programa. El Capítulo 2 menciona cómo se conforma una imagen digitalmente y cómo es representada en MATLAB. El primer paso para procesar una imagen es cargarla en el ambiente MATLAB.

Teniendo una imagen almacenada en memoria y con el nombre “*prueba1.jpg*” entonces se hace uso del comando *imread('prueba1.jpg')* para cargarla en una matriz. Cada vez que se trabaja con una imagen diferente se le pone el nombre “*prueba1.jpg*” a ésta para que funcione correctamente. La Figura 4.1 muestra un ejemplo de cómo se visualiza la imagen almacenada en una matriz.



*Figura 4.1 Imagen original cargada en Matlab para ser procesada*

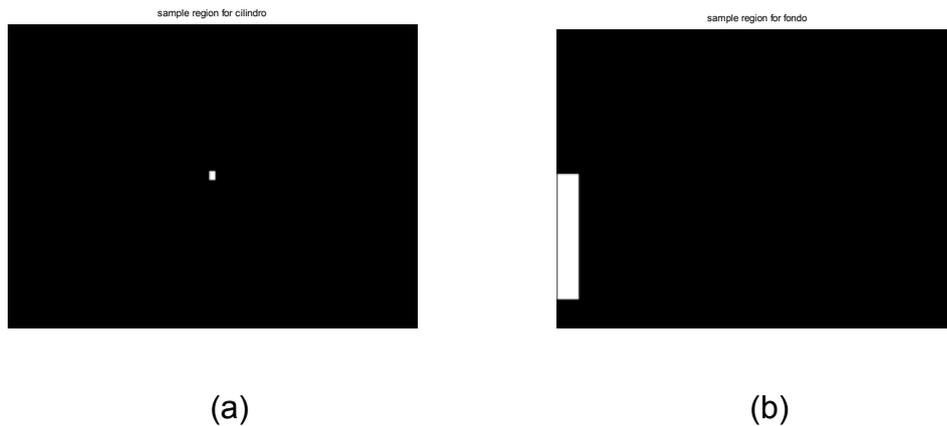
Teniendo la imagen representada por una matriz se procede a detectar el objeto. Detectar el objeto significa que este es aislado de otras regiones en la imagen, para que así sea posible trabajar únicamente sobre la región correspondiente al objeto y medir su tamaño en píxeles.

### **4.3 Detección del objeto por medio de clasificación de colores**

El procedimiento que se realizó en este paso fue discriminar un área de la imagen seleccionada usando un algoritmo de clasificación de color. Dicho algoritmo convierte la imagen del espacio *RGB* al espacio *L\*a\*b*

No existe una función que haga esta conversión en un solo paso en MATLAB, sino que se usan una serie de funciones. En el Apéndice A se puede ver el código completo del programa que se usa para realizar todo el algoritmo para medición del objeto virtual. Una función crucial en este paso es la función *roipoly*, la cual permite tomar una pequeña muestra de la imagen en donde se encuentra el color de interés. En este caso, son dos regiones de interés, la que contiene al cilindro y la que contiene el resto de la imagen, es decir el fondo de la imagen. Se requiere de dos muestras de la imagen, una dentro del cilindro y otra en cualquier parte del fondo de la imagen. El programa toma muestras automáticamente en ambas regiones de la imagen. Las máscaras, para las dos muestras que se toman, son las que se muestran en la Figura 4.2.

Habiendo sido tomadas estas muestras se procede a convertir la imagen al espacio  $L^*a^*b$ , haciendo uso de las funciones *makecform* y *applycform*, con la imagen original. La primera función crea una estructura de transformación del color '*srgb2lab*' (`cform = makecform('srgb2lab');`), que es útil para convertir de *RGB* a  $L^*a^*b$ , y la segunda aplica la transformación definida en la estructura previamente realizada. Después de la conversión de la imagen, se procede a clasificar sus colores, mediante la obtención del promedio del color en cada región, que servirá para identificar los colores mediante una etiqueta en el momento de la clasificación (marcadores de color). Para esto, se usa la función *mean2*, que calcula el promedio de los valores en un vector.



*Figura 4.2 Muestras para cada región. Muestra para el cilindro (a) y para el fondo de la imagen (b)*

Para la clasificación se hace uso del método del vecino más cercano. A cada marcador de color se le asigna un valor 'a' y uno 'b' y entonces es posible clasificar cada *píxel* en la imagen en el espacio  $L^*a^*b$  calculando la distancia euclidiana entre ese *píxel* y los marcadores o etiquetas de color. La distancia más corta dirá que ese *píxel* concuerda con determinado marcador de color. Por ejemplo, si la distancia entre un *píxel* y el color correspondiente al marcador de color del cilindro es la más corta entonces ese *píxel* es clasificado como un *píxel* del color del cilindro. Si el cilindro es verde, el *píxel* es clasificado como verde [18]. Se procede a crear un vector que contenga las etiquetas para cada color, que en este caso son dos, para luego inicializar la matriz que se usa para la clasificación y después realizar la clasificación del color. Al final, es posible visualizar los resultados obtenidos como en el ejemplo que se muestra a continuación.

objetos con color de cilindro



*Figura 4.3 Píxeles correspondientes al color del cilindro*

objetos con color de fondo

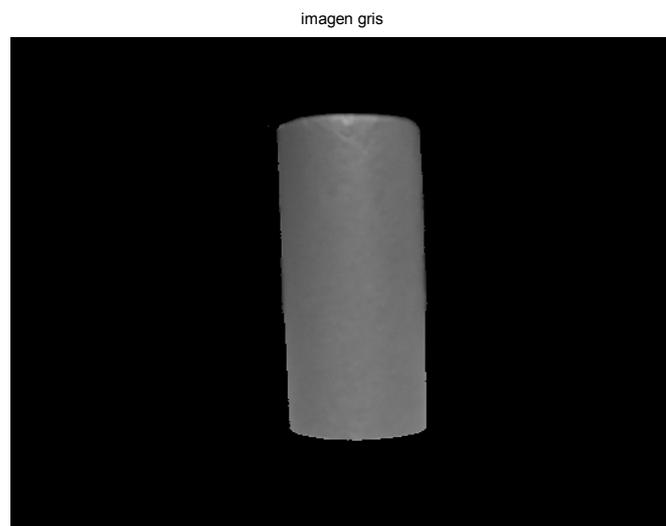


*Figura 4.4 Píxeles correspondientes al color del fondo*

En la Figura 4.3 se observan los píxeles correspondientes al cilindro, después de haber efectuado la clasificación de colores. Del mismo modo, en la Figura 4.4 se muestran los píxeles correspondientes al fondo de la imagen. A partir de este punto, el área de la imagen con la que interesa trabajar es con la

correspondiente al cilindro, el cual es aislado parcialmente en la imagen. Junto con el cilindro pueden presentarse algunas regiones o píxeles aislados en la imagen que coincidan con el color, introduciendo ruido. Antes de eliminar ruido es necesario realizar un par de pasos para facilitar la tarea.

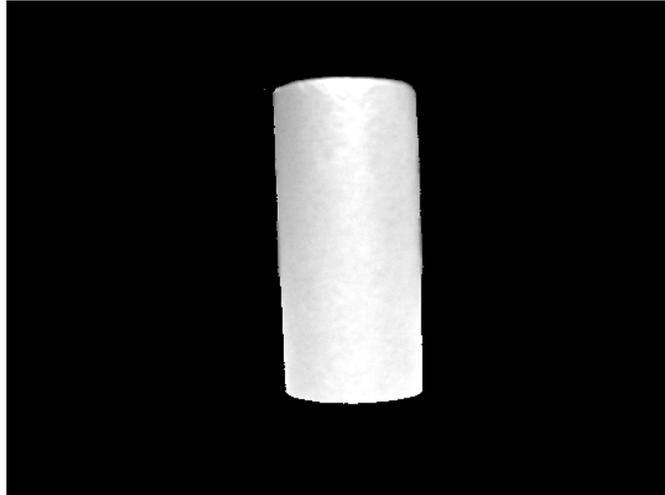
Teniendo aislados todos los puntos pertenecientes al marcador del color del cilindro se procede a cambiar la imagen resultante a una imagen más fácil de procesar. Se transforma a la imagen con color original en el espacio  $L^*a^*b$  a una imagen en blanco y negro, también llamada imagen binaria, donde un píxel negro toma un valor mínimo y un píxel blanco toma un valor máximo, y son los únicos dos valores posibles de existir. Un ejemplo de una imagen en escala de grises se muestra en la Figura 4.5.



*Figura 4.5 Cilindro convertido a escala de grises*

Posteriormente, se hace un ajuste de intensidad a la imagen, aumentando el contraste entre las regiones del objeto y fondo, como en el ejemplo que se muestra en la Figura 4.6.

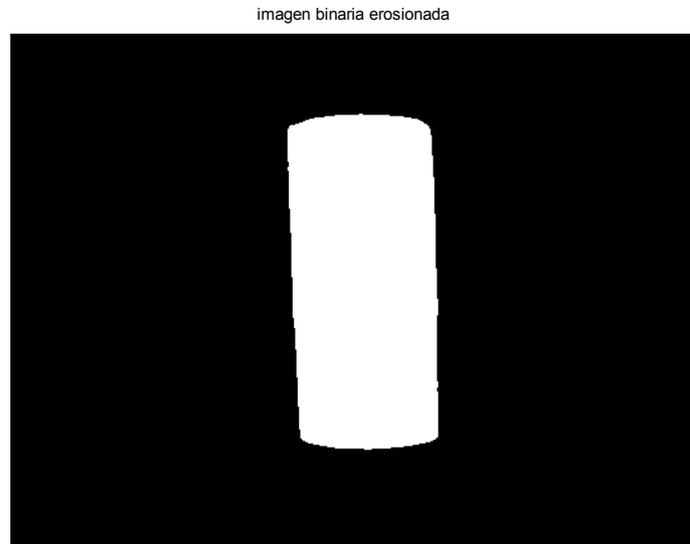
imagen gris mejorada



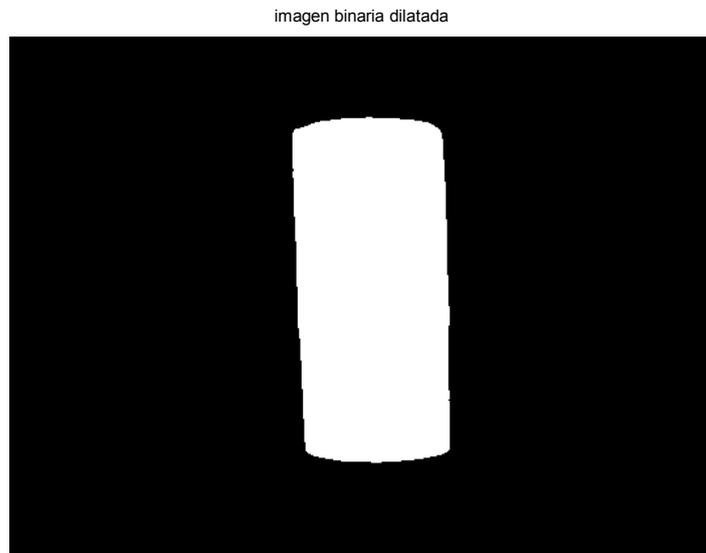
*Figura 4.6 Cilindro en escala de grises reajustada*

Los dos pasos anteriores sirven, en primera, porque no es posible la transformación directa de una imagen en color a una imagen binaria, pero más importante es el hecho de que, al aumentar el contraste, es más fácil seleccionar un umbral durante la transformación de la imagen en escala de grises a imagen binaria. Al final de la transformación, existe todavía la posibilidad de que haya píxeles de sobra en la imagen binaria, por lo que se ejecuta un procesamiento denominado procesamiento morfológica de erosión y dilatación (closing), que a grandes rasgos erosiona o elimina píxeles aislados en la imagen pero a costa de eliminar las orillas de los objetos grandes; es decir, trabaja con las fronteras de los objetos detectados. Sin embargo, después de dilatar la imagen, se regeneran las orillas de los objetos, pero es imposible que esto pase con los píxeles aislados debido a su previa eliminación en la etapa de erosión, eliminándose así el ruido [19].

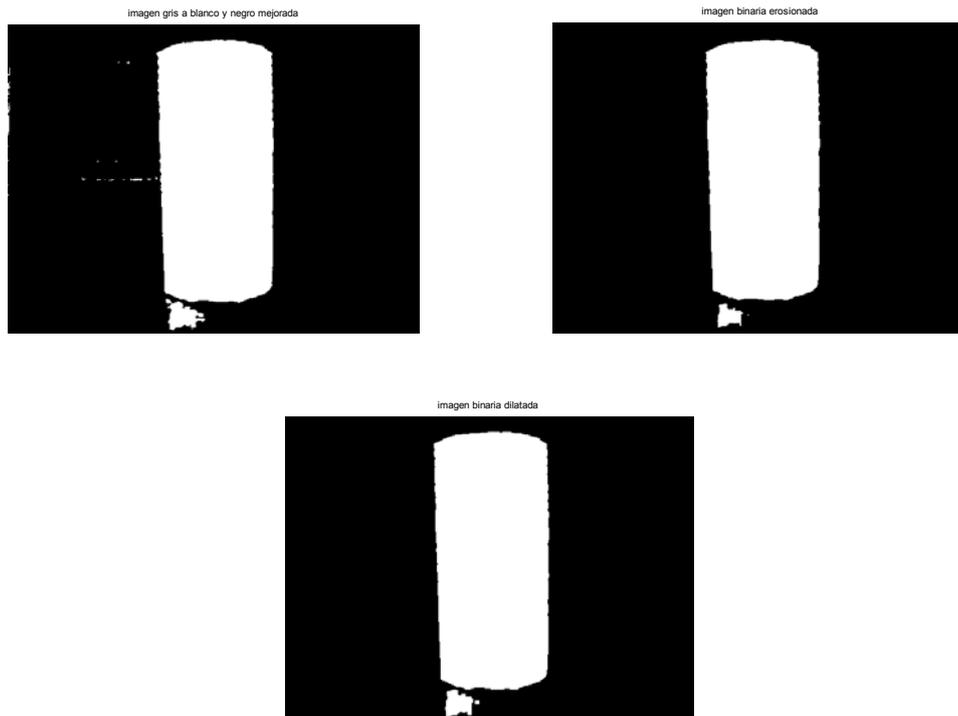
En las Figuras 4.7 y 4.8, se muestra el ejemplo del cilindro después de ser erosionado y dilatado respectivamente; sin embargo, en la Figura 4.9 se ilustra el problema para el caso de que existiera más ruido.



*Figura 4.7 Imagen erosionada: la frontera del cilindro ha sido eliminada pero igual lo han sido los píxeles aislados de la imagen*



*Figura 4.8 Imagen dilatada: Los objetos grandes como el cilindro recuperan su frontera y los píxeles eliminados ya no aparecen*



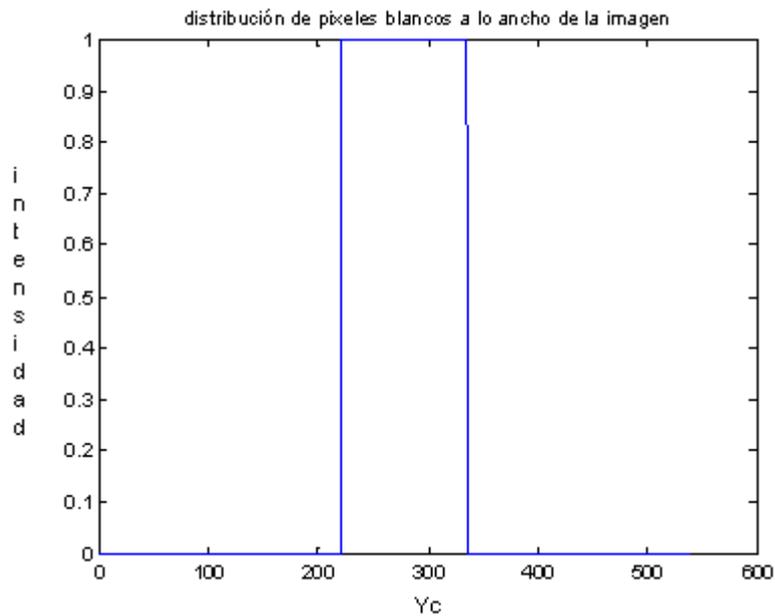
*Figura 4.9 Arriba izquierda: imagen binaria. Arriba derecha: imagen binaria erosionada. Abajo: imagen dilatada*

#### **4.4 Medición en píxeles del tamaño del objeto**

Ahora que se tiene el objeto aislado se analiza la distribución de los píxeles blancos a lo ancho de la imagen mediante el archivo *"pixelweight.m"*, incluido también en el apéndice A con el desarrollo del programa completo. Al terminar esta etapa, se conoce el número de píxeles que el cilindro abarca a lo ancho de la imagen. Al término de dicho análisis puede observarse un histograma como el de la Figura 4.10.

Este histograma denota el número de píxeles que ocupa el cilindro a lo ancho dentro de la imagen. La imagen presenta un ancho original total de 539 píxeles. De 0 a 220 píxeles el histograma está en cero, lo que significa que en la imagen hay píxeles negros, al igual que de 335 a 539. En la parte restante, el

histograma está en uno, lo que significa que en esta región se encuentran los píxeles blancos, que es lo correspondiente al cilindro y que para este ejemplo son un total de 115 píxeles, contados en la gráfica a partir de 220 hasta 335.



*Figura 4.10 Histograma de distribución de píxeles blancos a lo ancho de la imagen*

Para cada distancia a la que se coloque el cilindro, se mide un diferente número de píxeles a lo ancho de la imagen; es decir, se miden  $n$  píxeles cada vez que se corre el programa. Entonces, mientras más lejos esté el objeto menos píxeles ocupará de ancho y viceversa.

#### **4.5 Tamaño virtual del objeto y medición de la distancia con respecto a la cámara**

Hasta este punto se ha logrado calcular el tamaño en píxeles del ancho del cilindro, obteniendo el tamaño virtual del objeto. De este modo se aplica la ecuación 1.2 y se calcula la distancia entre la lente y el cilindro.

De la ecuación 1.2 conocemos, hasta este punto, dos de las variables de interés que son el tamaño real del objeto  $L$  y la distancia focal de la cámara  $f$ . El único dato faltante es la medida virtual del objeto  $L'$ ; sin embargo, se conoce la manera de medir el número de píxeles en la imagen virtual y saber a cuánto equivale en una unidad del sistema métrico decimal, multiplicando el número de píxeles por un factor de conversión. Como es visto en la sección 4.4, cada vez que se haga el procesamiento de una imagen se obtendrán  $n$  píxeles para el objeto, dependiendo de la distancia a la que éste se encuentre.

El factor de conversión se halla evaluando la fórmula en diversas distancias conocidas (previamente medidas). La variable  $L'$ , entonces, es un producto del número de píxeles  $n$  por el factor de conversión  $\alpha$ .

$$L' = \alpha \times n \quad \text{Ecuación 4.1}$$

Se sustituyen las variables obtenidas en la fórmula de la ecuación 1.2

$$d = \frac{L}{L'} f :$$

$$d = \frac{47mm}{L'} \times \text{de } 40.7602mm \quad \text{Ecuación 4.2}$$

donde:

$$L' = \alpha \times n \quad \text{Ecuación 4.3}$$

entonces:

$$d = \frac{47mm}{\alpha \times n} \times \text{de } 40.7602mm \quad \text{Ecuación 4.4}$$

Y como también se conoce  $n$  para cada caso, entonces se despeja  $\alpha$  quedando la expresión de la siguiente manera:

$$\alpha = \frac{47mm}{d \times n} \times 40.7602mm \quad \text{Ecuación 4.5}$$

Como se mencionó anteriormente, para el cálculo de este factor se coloca el objeto a una distancia específica y para cada distancia diferente se registra el número de píxeles. El parámetro  $\alpha$  constante y se mantiene para el cálculo de la distancia, siendo  $n$  el único factor que varía durante el procesamiento final.

Para ilustración, se muestra un ejemplo para el cálculo de  $\alpha$ . Se coloca el objeto a una distancia conocida  $d = 20\text{cm}$  de la cámara y su tamaño en píxeles es  $n = 155$ . Se repite este proceso varias veces para tener diferentes resultados de  $\alpha$ , los cuales se promedian, para obtener un valor de  $\alpha$  para cada resolución de la cámara, que se usa para calcular la distancia de cualquier objeto hacia la cámara.

*Tabla 4.1 Parámetro  $\alpha$  para diferentes distancias del objeto respecto a la cámara en imágenes de 539 x 404 píxeles.*

Distancia en centímetros	Ancho en píxeles	Alfa correspondiente
42	69	0.642690131
52.5	54	0.656972134
28.5	104	0.628379217
21	139	0.638066461
87	32	0.669007184
66	43	0.65627766
135.3	21	0.655515433
110.2	26	0.650047466
79.1	36	0.654065178

En la tabla 4.1 se muestran los valores de  $\alpha$  correspondientes a diferentes distancias conocidas, así como el número de píxeles ocupados en la imagen

dependiendo de la distancia. Al promediar todos estos valores de  $\alpha$ , se obtiene un valor definitivo a utilizar, que para este ejemplo es de  $\alpha = 0.6442$ . Al sustituir este factor en la expresión de la ecuación 5.4, se cuenta con todos los valores necesarios para la fórmula de la ecuación 1.2. Al colocar el cilindro a alguna distancia de la cámara y procesar la imagen correspondiente, se obtiene como resultado la información de la profundidad del objeto respecto a la lente, o sea, la distancia entre la cámara y el objeto.

Cabe mencionar que el resultado anterior fue aplicado para una imagen de resolución 539 x 404 píxeles de ancho y alto respectivamente. Es posible utilizar cualquier fotografía de cualquier resolución ya que el programa la redimensiona automáticamente. Con esta resolución se hizo el desarrollo del algoritmo, sin embargo en las etapas finales se decidió utilizar una resolución mayor para que el funcionamiento fuera más óptimo. Dicha resolución es de 2048 x 1536 píxeles. En el Capítulo 6 se dan a conocer los experimentos realizados con ambas resoluciones y se dice lo que conviene de cada una.

Tuvo que hacerse también una tabla de caracterización de  $\alpha$  en el caso de variar la resolución de la cámara. La tabla 4.2 muestra las mediciones obtenidas. En ambos casos se mide la distancia de la lente al objeto, pero con  $\alpha$  diferente.

Tabla 4.2  $\alpha$  para diferentes distancias del objeto respecto a la cámara en imágenes de 1536 x 2048 píxeles

Distancia en centímetros	Ancho en píxeles	alfa correspondiente
43.15	252	0.171284739
63.35	176	0.167047786
27.2	390	0.175576546
85.3	128	0.170585067
116.49	92	0.173789502

Y el  $\alpha$  promedio para la resolución 2048 x 1536 píxeles es de  $\alpha = 0.171657$ .  
 A manera de resumen, la Figura 4.12 muestra un diagrama de flujo con el funcionamiento del algoritmo.

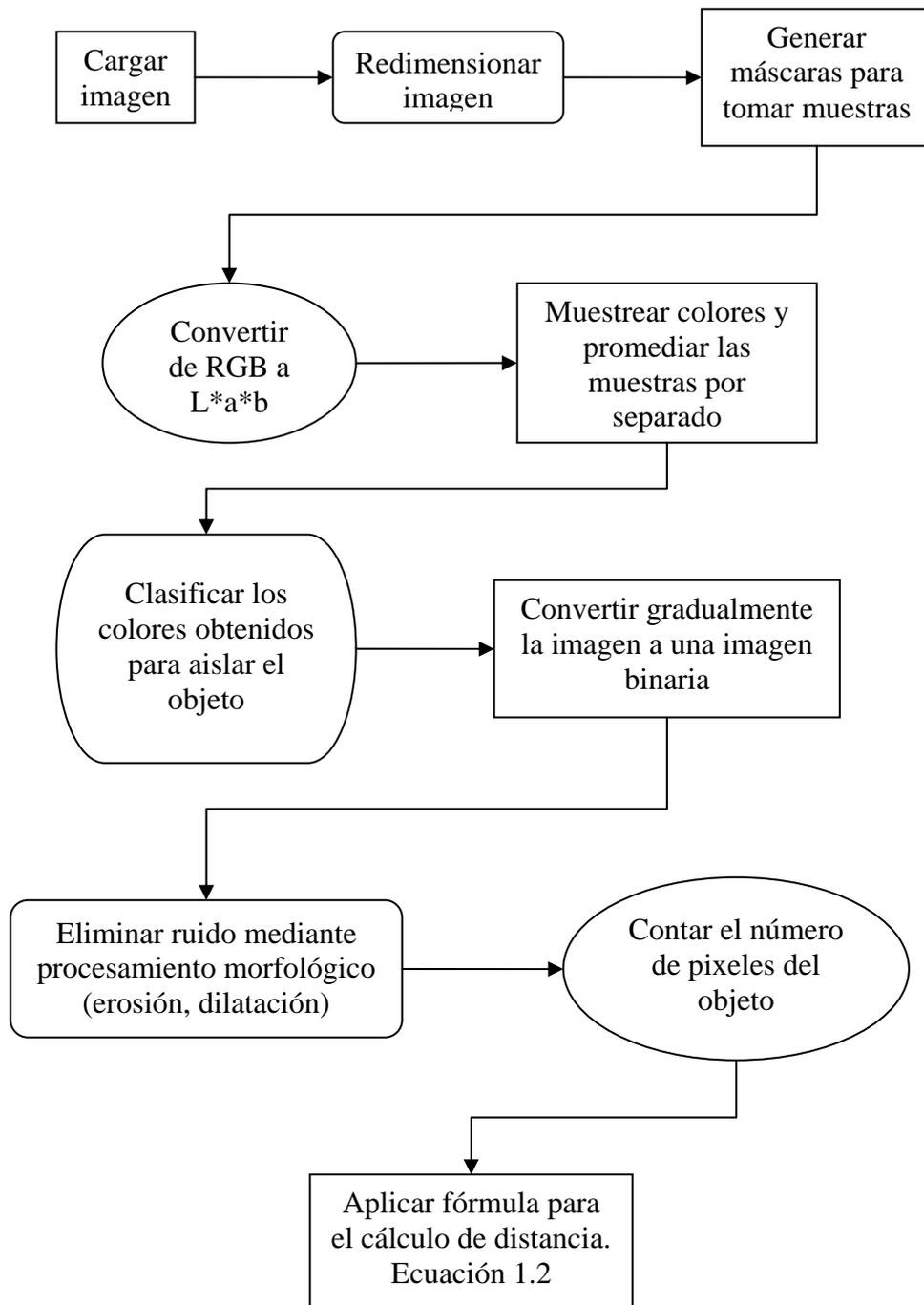


Figura 4.11 Diagrama de flujo del algoritmo de medición de distancias