

Los Histogramas pueden ser considerados como medidas estadísticas de la imagen y normalmente son usados como medio de ayuda para evaluar propiedades importantes de una imagen. Especialmente son los errores producidos en la toma de la imagen los que son fácilmente reconocidos mediante la utilización del histograma. Aparte de poder tratar con los problemas anteriormente mencionados es posible también pre-procesar la imagen tomando como base el histograma para mejorarla o bien para resaltar características de ella que serán o bien extraídas o analizadas en fases de procesamiento posterior (considerando posteriormente un sistema de reconocimiento de patrones en la imagen, por ejemplo).

## 3.1 ¿QUÉ ES UN HISTOGRAMA?

Los histogramas son distribuciones que describen la frecuencia con la que se presentan los valores de intensidad (píxeles) de la imagen. En el caso más sencillo los histogramas son mejor entendidos por medio de imágenes a escala de grises, un ejemplo es mostrado en la figura 3.1. Para una imagen a escala de grises  $I$  con intensidades en el intervalo  $I(u, v) \in [0, K - 1]$  contendrá el histograma  $H$  exactamente  $K$  diferentes valores, el cual considerando una típica imagen a escala de grises de 8 bits será de  $H = 2^8 = 256$ . Cada valor del histograma es definido como

$$h(i) = \text{el número de píxeles de } I \text{ con el valor de intensidad } i$$

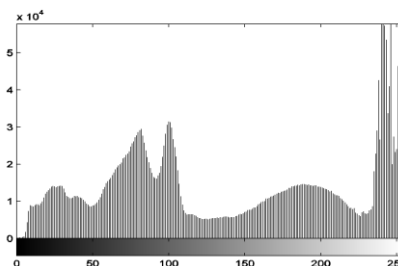
para todos los valores  $0 \leq i < K$ . Esto expresado formalmente:

$$h(i) = \text{card} \{ (u, v) | I(u, v) = i \}^1$$

$h(0)$  es entonces el número de píxeles con el valor 0,  $h(1)$  el número de píxeles que tienen el valor de 1, sucesivamente mientras que finalmente  $h(255)$  representa el número de píxeles blancos (con el máximo valor de intensidad) de la imagen. Como resultado del cálculo del histograma se obtiene un vector unidimensional  $h$  con una longitud  $K$ , tal y como es representado en la figura 3.2 donde  $K=256$ .

Evidentemente el histograma no proporciona información acerca del origen de los píxeles que lo conforman. Esto es, es pérdida la información de la relación espacial que tenían los píxeles en la imagen, por lo que resulta imposible reconstruir una imagen a partir de su histograma. Para evidenciar este hecho la figura 3.3 muestra 3 diferentes imágenes que producen el mismo histograma.

El Histograma muestra importantes características de una imagen, como por ejemplo el contraste y el rango dinámico, los cuales son problemas atribuidos a la captación de la imagen que es necesario



<sup>1</sup>  $\text{card}\{\dots\}$  representa cardinalidad, el número de elementos.

resolver para que puedan ser analizadas con mejor claridad propiedades de la imagen en bloques de procesamiento posterior.

Figura 3.1 Imagen de intensidad (escala de grises) de 8-bits con su respectivo histograma.

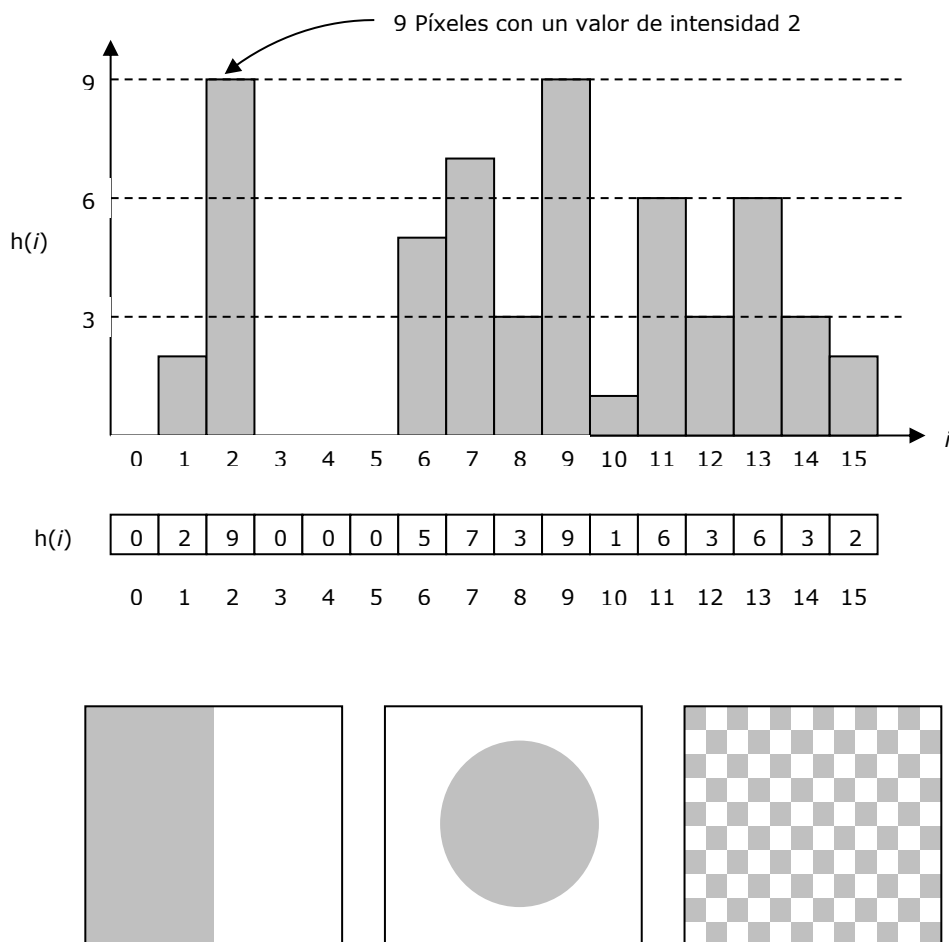


Figura 3.3 Tres diferentes imágenes que producen el mismo histograma

## 3.2 CARACTERÍSTICAS DE TOMA DE UNA IMAGEN

Los histogramas muestran características importantes de una imagen, como por ejemplo el contraste y la dinámica, problemas que son producidos durante la toma de la imagen y generan consecuencias para las siguientes etapas de procesamiento.

### 3.2.1 Iluminación

Los errores de la iluminación son reconocidos en el histograma porque la región final o inicial de la escala de intensidad no es utilizada, mientras que en las otras regiones del histograma son ocupadas por valores de intensidad de píxeles. La figura 3.4 muestra un ejemplo de imágenes con diferentes tipos de iluminación.

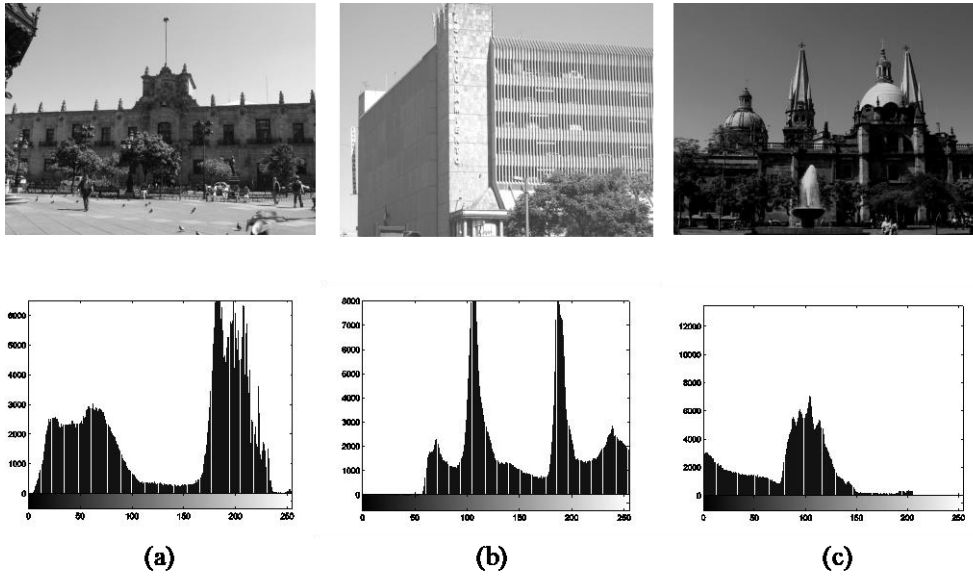


Figura 3.4 Las figuras muestran como los errores de iluminación son fácilmente detectables por el histograma. (a) Imagen con correcta iluminación, (b) con alta iluminación y (c) con baja iluminación.

### 3.2.2 Contraste

Como contraste se entiende el campo de valores de intensidad que en una imagen concreta son utilizados, en pocas palabras la diferencia entre el máximo y mínimo valor de intensidad de los píxeles presentes en la imagen. Una imagen con un contraste completo utiliza el intervalo completo de niveles de intensidad definidos para la imagen  $a = a_{\min} \dots a_{\max} = 0 \dots k - 1$  (de negro a blanco). Por ello es sencillo observar el contraste de una imagen utilizando para ello un histograma. La figura 3.5 muestra diferentes configuraciones de contraste en imágenes y el histograma producido.

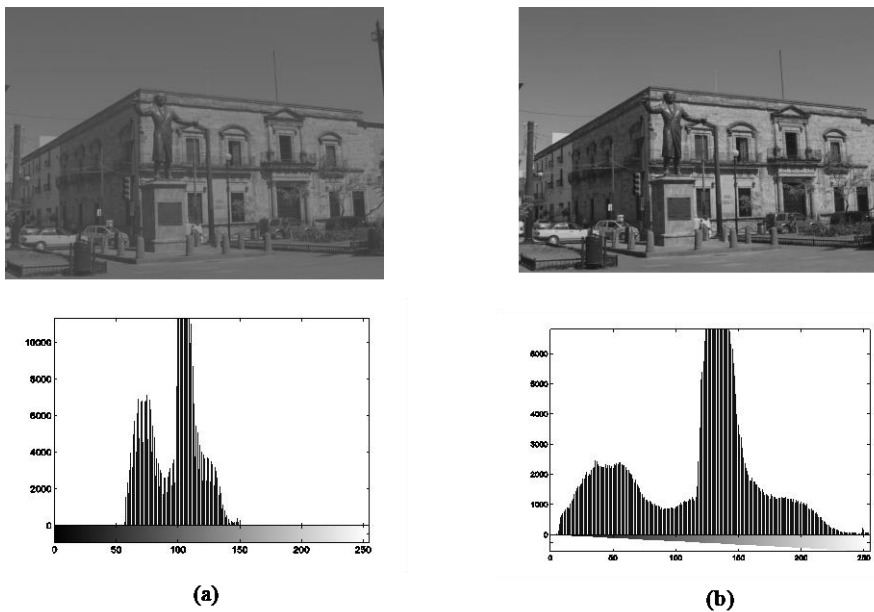


Figura 3.5 Las figuras muestran diferentes contrastes en imágenes y sus efectos en el histograma, (a) bajo contraste y (b) contraste normal.

### 3.2.3 Dinámica

Bajo el término de dinámica se entiende el número de píxeles diferentes que son utilizados en la imagen. El caso ideal para una imagen resulta cuando es utilizado el rango completo de valores de intensidad disponibles  $K$  para la imagen en cuestión, en este caso la región de valores es cubierta completamente. Una imagen que cubra una región de valores de intensidad  $a = a_{\min} \dots a_{\max}$  más pequeña que la completa con

$$a_{\min} > 0 \text{ y } a_{\max} < 255$$

la cual alcanza su máximo rango dinámico cuando todos los valores de intensidad en ese intervalo están presentes en la imagen (figura 3.6).

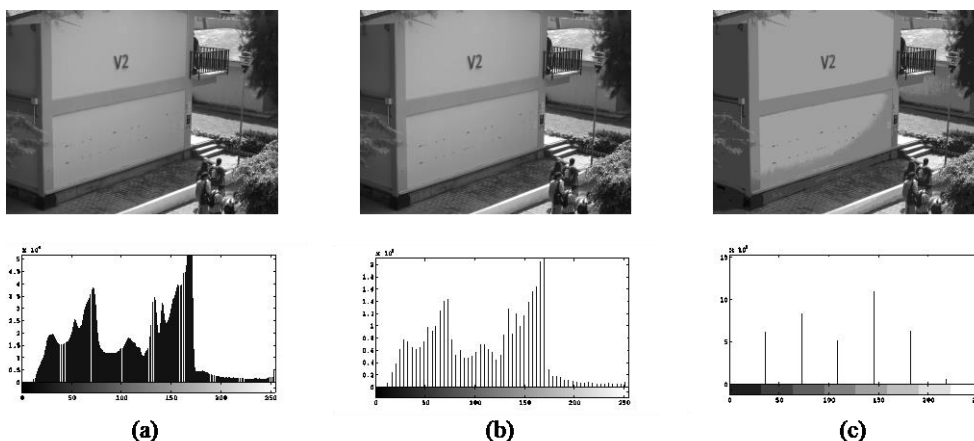


Figura 3.6 Diferentes dinámicas en una imagen y sus efectos en el histograma. Alta dinámica (a), baja dinámica con 64 niveles de intensidad (b) y dinámica demasiado baja con solo 8 niveles de intensidad.

Mientras el contraste de una imagen puede ser elevado en tanto el valor máximo del intervalo de la intensidad de los píxeles no sea sobrepasado, la dinámica de una imagen no puede ser elevada (excepto por interpolación de los valores de intensidad de los píxeles). Una dinámica alta representa para una imagen una ventaja por que se reduce el riesgo de perder calidad en la imagen a través de las siguientes etapas de procesamiento. Por esa razón las cámaras digitales y scanners profesionales tienen una resolución mayor a los 8-bits, normalmente de 12 a 14 bits, aunque los elementos de despliegue de la imagen tengan las resoluciones normales de 256.

## 3.3 CÁLCULO DEL HISTOGRAMA DE UNA IMAGEN CON MATLAB

En esta sección se explican las funciones en MatLAB que pueden calcular y desplegar el histograma de una imagen, esto desde dos perspectivas diferentes que van desde encontrar el histograma de una imagen fija por medio de una función o hacerlo sobre video proveniente de una Webcam utilizando la librería de bloques de procesamiento de imagen y video de Simulink.

### 3.3.1 Función de MatLAB para línea de comandos

La función del toolbox de procesamiento de imagen para el cálculo del histograma de una imagen tiene el formato:

```
[counts,x]=imhist(I, n)
```

Esta función calcula y despliega el histograma de la imagen  $I$ . El número de valores del histograma si no se especifica  $n$  depende del tipo de imagen que se trate. Si  $I$  es una imagen a escala de grises la función utilizara para el cálculo y despliegue 256 valores. Si  $I$  es una imagen binaria la función calculará el histograma con solo 2 valores.

Si  $n$  es especificada el histograma es calculado y desplegado utilizando  $n$  valores en lugar de los especificados por el tipo de imagen. La variable `counts` es un vector que recibe el número de píxeles pertenecientes al valor de intensidad especificado por  $x$ . Relacionando estas dos variables mediante `stem(x, counts)` podríamos encontrar de nueva cuenta el histograma.

### 3.3.2 Librerías de bloques para el procesamiento de imagen y video de Simulink

Otra opción que correspondería ya a la visión artificial, sería el hecho de calcular el histograma de frames de video en tiempo real. Para ello se utilizan las librerías de bloques para el procesamiento de imagen y video de Simulink.

La librería para el procesamiento de imagen y video de Simulink mediante el bloque “2-D Histogram” permite calcular el histograma de una imagen. La figura 3.7 muestra gráficamente la representación de este bloque en Simulink.

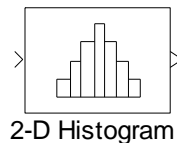


Figura 3.7 Bloque “2-D Histogram” de la librería de procesamiento de imagen y video de Simulink que permite calcular el histograma de una imagen

Este bloque perteneciente a la categoría de estadística “Statistics” de la librería de procesamiento de imagen y video de Simulink, genera un histograma considerando una matriz de entrada. El bloque calcula la distribución de frecuencia de los elementos contenidos en la imagen o matriz, es posible mediante configuración del bloque determinar la distribución de frecuencia de estos elementos durante un instante de tiempo.

El bloque tiene dos configuraciones posibles, 1) la básica y 2) la operación de “Running”. En la **operación básica** el bloque calcula la distribución en frecuencias de cada imagen acoplada a la entrada del bloque dando como salida un vector de  $1 \times N$  donde  $N$  es el número de valores del histograma el cual depende del tipo de la imagen (para el caso de una imagen a escala de grises se tiene  $N=256$ ). En la **operación “Running”** el bloque calcula la distribución de frecuencias en una serie de imágenes.

Normalmente la mayoría de las aplicaciones utilizan a éste bloque en su modo de operación básica, ya que normalmente se requiere tomar decisiones sobre características instantáneas del histograma de una imagen, por lo que en ésta sección será ejemplificado su uso pero solamente en su operación básica.

La figura 3.8 muestra la ventana de configuración de éste bloque. Al seleccionar (1) la casilla “Running histogram” estamos configurando al bloque para que trabaje en su modo de operación “Running”. Si dicha casilla está no seleccionada indica que el bloque trabaja en su operación básica.

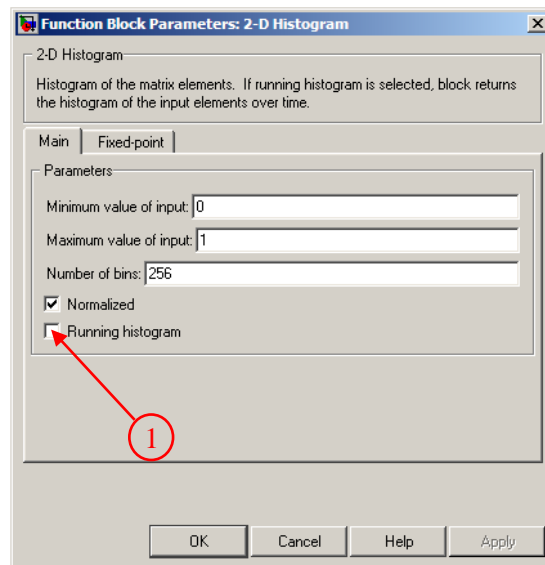


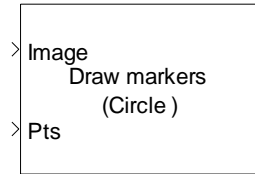
Figura 3.8 Muestra la ventanilla de configuración del bloque “2-D Histogram” de la librería de procesamiento de imágenes y video de Simulink.

Al seleccionar la casilla “Normalized” el bloque escala cada elemento calculado del histograma de tal manera que la suma de todos los elementos del histograma resulte uno.

Un problema es que no se tiene un bloque o función de Simulink que permita el despliegue de los resultados del bloque “2-D Histogram” por lo que en esta sección se presenta un esquema alternativo para el despliegue del histograma en tiempo real de imágenes provenientes de una Webcam. Para poder desplegar el histograma en tiempo real de imágenes provenientes de la Webcam, una vez calculado el histograma mediante el bloque “2-D Histogram” utilizaremos una imagen como fondo para desplegar los resultados. Para ello se crea una matriz de que contenga unos, en línea de comandos que tendrá las dimensiones adecuadas para el despliegue. Será sobre esta imagen sobre la cual se escribirán los valores del histograma calculado. Para realizar esto tendremos que utilizar algunos bloques funcionales que permiten dibujar marcas tales como puntos o círculos sobre de una imagen considerada como fondo o tapete.

### El bloque para dibujar marcas

La librería de procesamiento de imagen y video de MatLAB tiene el bloque “Draw Markers” que pertenece a la categoría de bloques de texto y gráficos, “Text & Graphics” permite dibujar múltiples círculos, marcas x, signos de +, estrellas y cuadrados sobre imágenes mediante la sobre-escritura de los píxeles a través del algoritmo Bresenham. La figura 3.9 muestra gráficamente la representación de este bloque en Simulink.



Draw Markers

Figura 3.9 Bloque “Draw Markers” de la librería de procesamiento de imagen y video de Simulink que permite dibujar sobre imágenes diferentes símbolos en posiciones predeterminadas. Por defecto el bloque está configurado para añadir a la imagen círculos definidos en las posiciones Pts.

La imagen de salida de este bloque es del mismo tipo que la de entrada definida en Image, la imagen tendrá la misma información que Image además de las marcas definidas en los puntos Pts. Los puntos entregados a este bloque es una matriz de  $2 \times N_p$ , donde  $N_p$  es el número de puntos a graficar. La figura 3.10 muestra la ventana de configuración de este bloque.

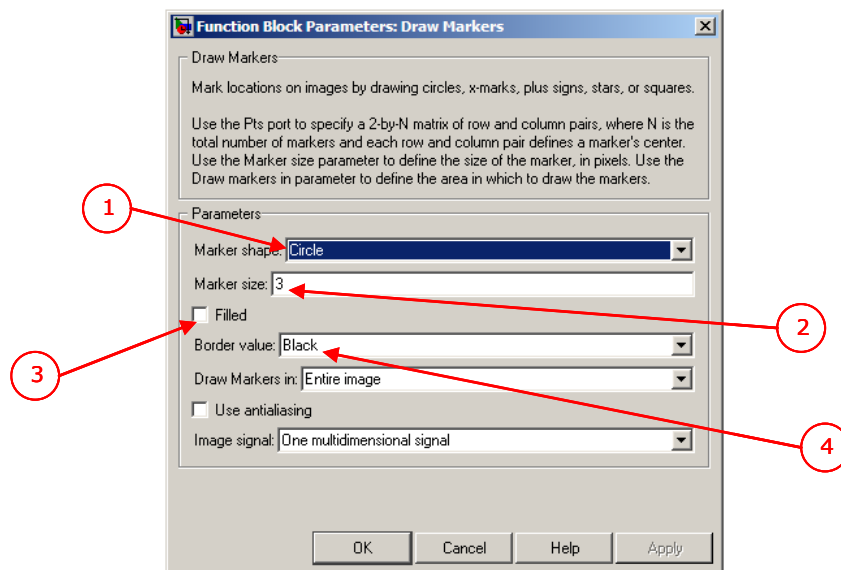


Figura 3.10 Muestra la ventanilla de configuración del bloque “Draw Markers” de la librería de procesamiento de imágenes y video de Simulink.

En la ventanilla se pueden configurar diferentes tipos de marcas que se escriben sobre la imagen a través de la opción “Marker shape” (1), además es posible definir el tamaño de la marca en la imagen mediante la definición del número de píxeles que lo forman (Marker size (2)). Puede configurarse que la marca tenga sólo el borde o sea rellena si se selecciona la casilla “Filled” (3). Una opción interesante es que podemos definir el color de la marca (4), la cual puede ser blanco, negro o un color definido por el usuario a través de la definición de un vector RGB.

### Image from workspace

Este bloque permite añadir al algoritmo de procesamiento una matriz o imagen constante definida previamente desde la consola de comandos de MatLAB. La figura 3.11 muestra el bloque como es representado en Simulink. La configuración de éste bloque permite definir el nombre de la variable tal y como fue declarada en línea de comandos.

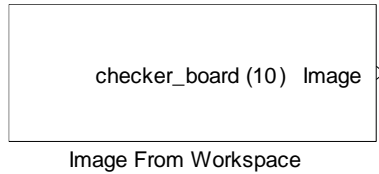


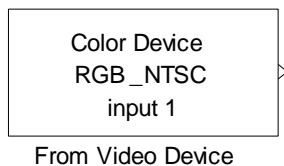
Figura 3.11 Bloque de Simulink para añadir una matriz o imagen al algoritmo de procesamiento de imagen. El bloque por defecto inicializa con una imagen predefinida, tal y como se muestra.

### Bloque from video device

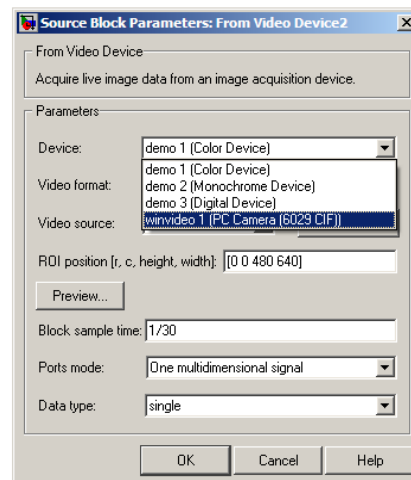
Para poder captar imágenes a partir de una Webcam MatLAB provee un bloque parte de las librerías contenidas en el **Toolbox de Adquisición de Imágenes**. El bloque es accesible desde Simulink y permite capturar frames de una Webcam, con el sólo hecho de estar instalado su driver.

El bloque se encuentra representado en la figura 3.12(a), en ella puede apreciarse como se trata de un bloque cuya funcionalidad es completamente destinada a fungir como una fuente de imágenes provenientes en este caso de la Webcam.

Para poder utilizar este bloque es necesario configurarlo de tal forma que pueda ser capaz de conectarse con el dispositivo de video (tal como la Webcam). Para ello en su mascarilla de configuración es necesario seleccionar el concepto “Device” la opción winvideo, la cual hará posible que el bloque pueda ser capaz de comunicarse con el driver del dispositivo de video, en este caso con la Webcam. La Figura 3.12(b) muestra la ventana de configuración del bloque “From Video Device” y el valor seleccionado para poder utilizar a la Webcam como fuente de video.



(a)



(b)

Figura 3.12 (a) Bloque “From Video Device” parte de la librería del Toolbox de Adquisición de Imágenes. Permite capturar imágenes desde un dispositivo de video compatible con Windows®, tal como una Webcam. (b) Muestra la ventana de configuración del bloque “From Video Device” y el valor seleccionado para poder utilizar a la Webcam como fuente de video. Como puede verse en la figura entre paréntesis se señala el nombre de la Webcam que en ese momento se encuentra conectada y disponible para ser utilizada.

El programa realizado en Simulink para el cálculo y despliegue en tiempo real del histograma sobre imágenes provenientes de una Webcam es esquematizado en la figura 3.13.



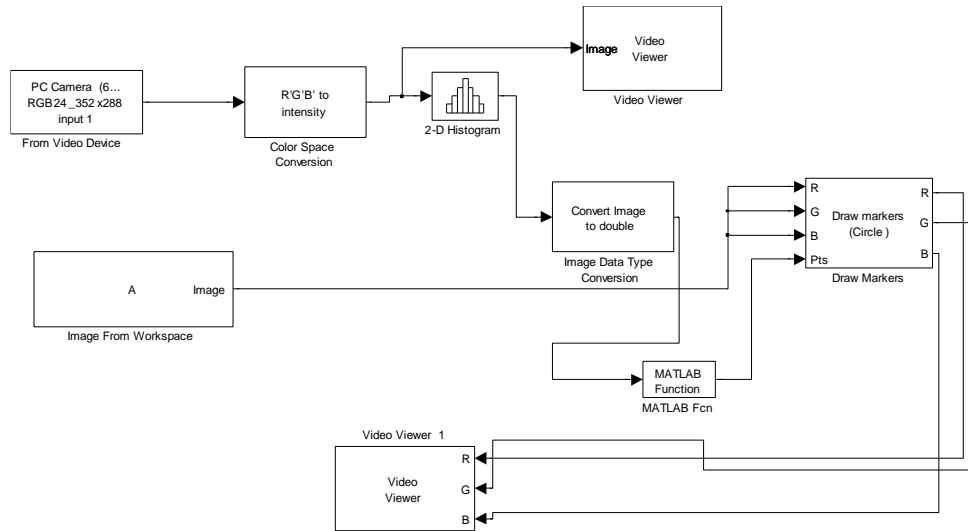


Figura 3.13 Programa en Simulink que calcula y despliega en tiempo real el histograma de imágenes que provienen de una Webcam.

El programa puede entenderse de la siguiente manera, la imagen capturada por la Webcam a través del bloque “From Video Device” es convertida a imagen a escala de grises y posteriormente se calcula su histograma utilizando el bloque “2-D Histogram”. Este bloque entrega como resultado de procesamiento un vector de  $1 \times N$ , donde  $N$  depende del tipo de imagen de que se trate (para una imagen a escala de grises este será de 256). Como la idea es desplegar la información del histograma en forma bidimensional, tendremos que a partir del vector  $1 \times N$  y generar los puntos que serán utilizados por el bloque “Draw Markers” para dibujarlos sobre una imagen y generar de esta manera la impresión de la representación de las distribuciones de frecuencias de los píxeles calculados en el histograma.

Para generar los puntos a partir de la vector  $1 \times N$  entregado por el bloque “2.D Histogram”, se genera una función en MatLAB, la cual tendrá por objetivo empaquetar el vector  $1 \times N$  que contiene la distribución de frecuencia de los píxeles, junto con otro vector que a cada elemento del vector  $1 \times N$  le asigne un valor que describa a que píxel corresponde esa distribución en frecuencia. Como en este ejemplo se trata de imágenes a escala de grises este vector es un vector de 256 elementos donde el primero es 1 y el ultimo 256, de ésta manera la salida de la función son dos vectores empaquetados en una variable donde al elemento 1 corresponda la distribución en frecuencia (puede ser normalizada) de los píxeles de valor 0, mientras que para el valor de 256 el número de píxeles que corresponden al valor 255. En el programa 3.1 se muestra la función utilizada.

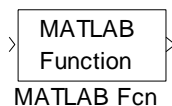
```
%Función que genera los puntos a partir %del vector 1xN
que se obtiene del %bloque "Histogram"
function data=colecta (u)
%La función recibe en u el vector 1xN. %Se genera el
vector num que especifica %el valor de la abscisa.
num=1:256;

%El valor del vector 1xN (u) se escala y %se le resta
un valor para generar el %efecto grafico. Esto se hace
debido a %que el valor entregado por el bloque
%"Histogram" esta normalizado (0-1), en %este caso se
le dio una ganancia de %5000.
val=288-(u*5000);
```

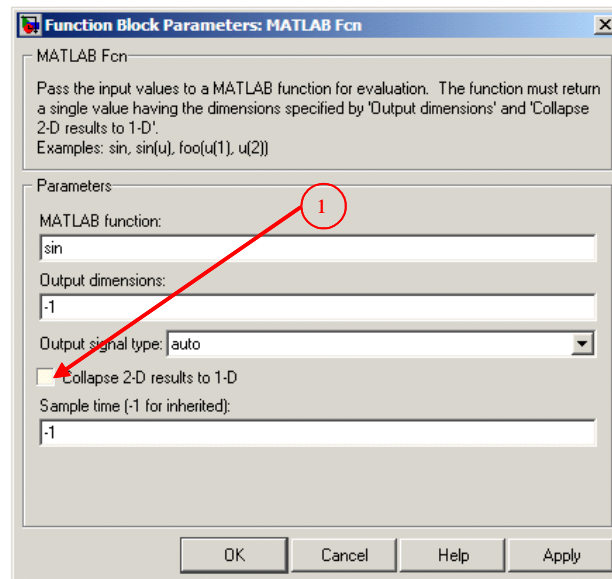
```
%Se empaquetan los datos en la variable %data, los cuales
serán puntos %entregados al bloque "Draw markers" %para
graficarlos sobre una imagen %sintética como fondo
data(:,1)=val;
data(:,2)=num;
```

Programa 3.1 Ejemplo de función que determina el valor del píxel máximo y mínimo

Para utilizar una función de MatLAB en Simulink es necesario utilizar el bloque “MATLAB Fcn”, que es parte de la categoría funciones definidas por el usuario (User-Defined functions). Sin embargo en este caso como la salida de la función no es un vector unidimensional sino pares de puntos de dimensión 2x256, es necesario configurar la mascarilla del bloque “MATLAB Fcn”, esto se logra deseleccionando la casilla (1) de colapso a 2 dimensiones resultados a una dimensión (collapse 2-D results to 1-D), con ello podemos tener como salida los pares de puntos. La figura 3.14(a) muestra el bloque de Simulink “MATLAB Fcn” y la 3.14(b) la ventana de configuración.



(a)



(b)

Figura 3.14 (a) El bloque de función de MatLAB definida por el usuario para el trabajo en Simulink y (b) su ventanilla de configuración.

Una vez calculados los puntos con la función concatena definida en el programa 3.1, se pasan al bloque “Draw Markers” para su graficación. Para ello se utiliza como fondo o tapete una imagen definida previamente en línea de comandos, en este caso la matriz o imagen será definida como A, y utilizada en el programa a través del bloque “Image From Workspace”. La matriz será definida en línea de comandos por:

```
>>A=ones(352,256);
```

Las dimensiones de esta matriz deben de ser en número de filas lo suficientemente grandes de tal forma que plasme adecuadamente las variaciones del histograma, mientras que el número de columnas es siempre de 256. Una vez definida esta matriz y ejecutando el programa en Simulink descrito en la figura 3.13 obtenemos como resultado el descrito en la figura 3.15.

## 3.4 HISTOGRAMAS DE IMÁGENES A COLOR

Por histogramas de imágenes a color se refiere a histogramas de luminosidad o bien a los que se obtienen sobre cada uno de los planos que conforman la imagen a color, considerando a cada plano como si se tratase de una imagen en escala de gris independiente.

### 3.4.1 Histogramas de luminosidad

El histograma de luminosidad de una imagen no es otra cosa que el histograma de la correspondiente versión en escala de grises de la imagen a color, ya que la imagen a escala de grises extraída de la imagen a color, representaría la luminosidad de los diferentes planos que la componen. La figura 3.16 muestra el histograma de luminosidad de una imagen a color.

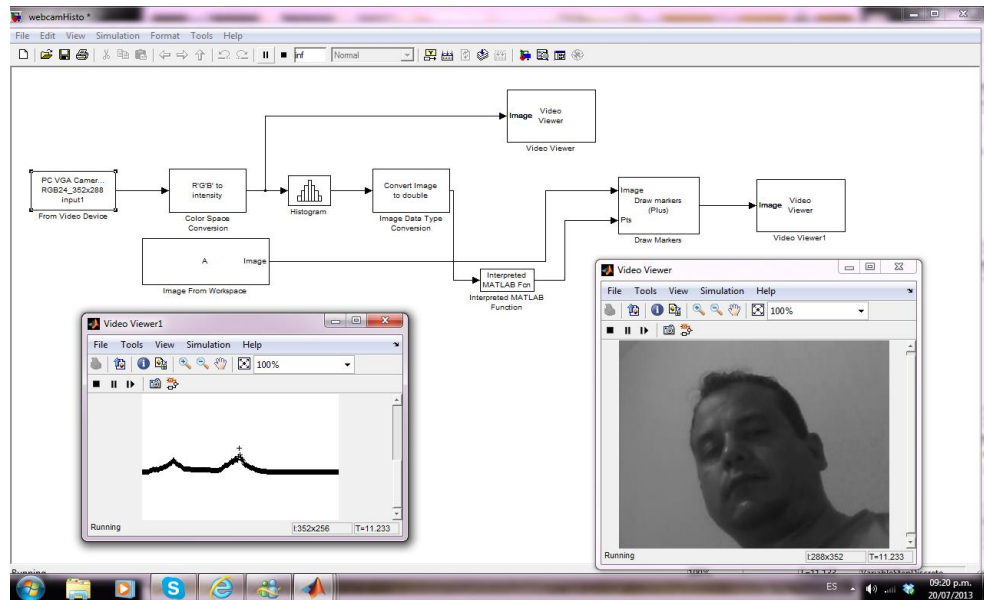


Figura 3.15 Muestra el resultado de ejecutar el programa en Simulink descrito en la figura 3.13.

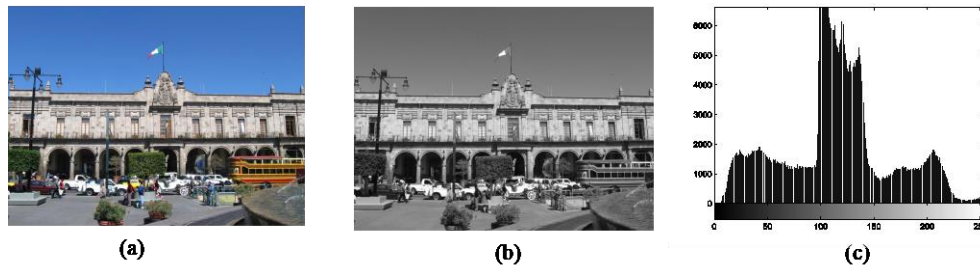


Figura 3.16 Imágenes que muestran el histograma de luminosidad de una imagen a color. (a) Imagen a color, (b) su versión en escala de grises (imagen de luminosidad) y (c) el correspondiente histograma de (b).

### 3.4.2 Histogramas de los componentes de color

Aunque el histograma de luminosidad considera todos los componentes de color es posible que no puedan considerarse errores presentes en la imagen. Por ejemplo, es posible que el histograma de luminosidad parezca adecuado a pesar de que alguno de los planos de color tenga algún error. En las imágenes RGB el plano azul normalmente aporta muy poco a la luminosidad total de la imagen a escala de grises calculada a partir de la imagen de color.

Los histogramas de cada plano dan también información adicional de la distribución de color en la imagen. Bajo este enfoque cada plano de color es considerado como una imagen a escala de grises independiente y es desplegada de igual manera. La imagen 3.17 muestra el histograma de luminosidad  $h_{Lum}$  y los histogramas de cada uno de los diferentes planos de color  $h_R$ ,  $h_G$  y  $h_B$  para una imagen RGB típica.

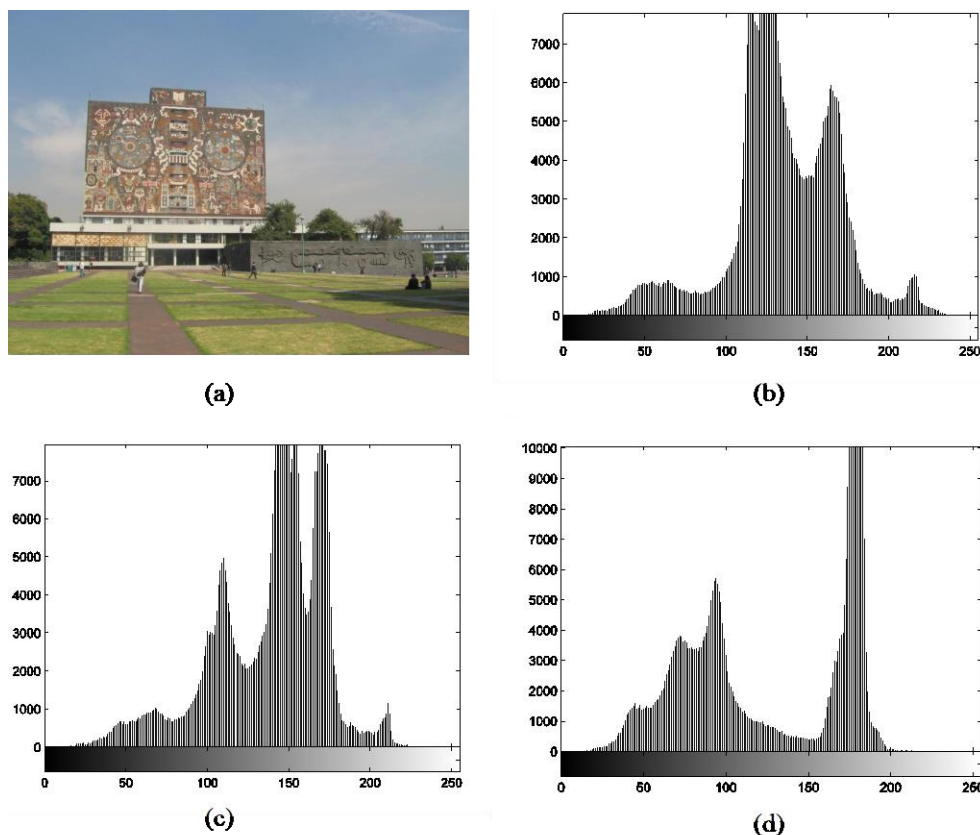


Figura 3.17 Histogramas de los componentes de una imagen a color. (a) Imagen original RGB, (b) histograma del plano R, (c) histograma del plano B y (d) el histograma del plano (d).

### 3.5 HISTOGRAMA ACUMULATIVO

El histograma acumulativo es una variante del histograma normal, el cual refleja información importante para la realización de operaciones de píxel por píxel en imágenes (operaciones de punto), por ejemplo para equilibrar un histograma. El histograma acumulativo  $H(i)$  es definido como:

$$H(i) = \sum_{j=0}^i h(j) \quad \text{para } 0 \leq i < K \quad (3.1)$$

El valor de  $H(i)$  es entonces la suma de todos los valores abajo del valor especificado  $i$  del histograma “normal”  $h(j)$  con los valores  $j = 0 \dots i$ . ó bien, el definido en forma recursiva

$$H(i) = \begin{cases} h(0) & \text{para } i=0 \\ H(i-1) + h(i) & \text{para } 0 \leq i < K \end{cases} \quad (3.2)$$

El histograma acumulativo es de acuerdo a su definición una función monótona creciente, con el valor máximo de

$$H(K-1) = \sum_{j=0}^{K-1} h(j) = M \times N \quad (3.3)$$

La figura 3.18 muestra un ejemplo del histograma acumulativo.

#### 3.5.1 Herramientas de MatLAB para el cálculo del histograma acumulativo

En esta subsección se tratará la forma de calcular el histograma acumulativo utilizando MatLAB. Esto persigue principalmente dos objetivos, primero tener la forma mas directa de calcular el histograma acumulativo y segunda mejorar mediante la realización de ejemplos las habilidades programáticas tanto de MatLAB como de Simulink.

Empezaremos por aplicar directamente en un programa de MatLab la ecuación recursiva mostrada en 3.2. Para ello nos valdremos de la utilización de la función `imhist` anteriormente tratada en la sección 3.3.1, la cual, como fue observado era capaz de regresar el vector  $1 \times 256$  de los valores del histograma calculado. A partir de estos valores es posible recursivamente calcular el histograma acumulativo. El programa 3.2 muestra los pasos necesarios para calcular y desplegar el histograma acumulativo.

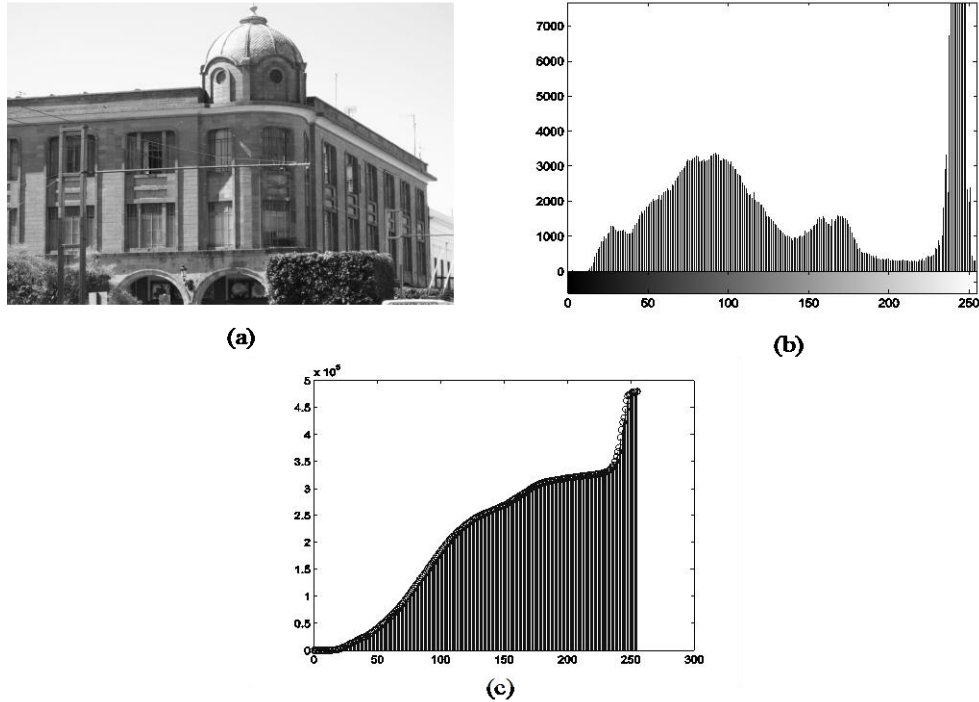


Figura 3.18(a) Imagen original, (b) histograma de (a) y (c) el histograma acumulativo de (a).

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Programa que permite generar el histograma acumulativo
%a partir de la información entregada por la
%funcion imhist

%Se determinan los datos del Histograma
[cont,x]=imhist(Ir);
%Se inicializa la variable utilizada para la recursión a
%cero
va=0;

%Se aplica el modelo definido en la Ecuación 3.2
for v=1:256
    H(v)=va+cont(v);
    va=H(v);
end
%Se grafica el resultado
stem(x,H)

```

Programa 3.2 Muestra el programa en MatLAB para calcular y desplegar el histograma acumulativo.

Es también posible desplegar en tiempo real el histograma acumulativo de una imagen proveniente de una Webcam, tal y como se hizo para el histograma normal en las secciones anteriores. Para ello utilizaremos exactamente el mismo programa en Simulink para la generación del histograma normal descrito en la figura 3.13. Lo único que hay que variar es el contenido de la función de usuario llamada por el bloque “MATLAB Fcn”. En este caso será en ésta función donde a parte de empaquetar los datos para que puedan ser graficados por el bloque “Draw Markers” también se calcule recursivamente según la ecuación 3.2 el histograma acumulativo. El programa 3.3 muestra la función utilizada para el cálculo del histograma acumulativo y el empaquetamiento de datos. La figura 3.18

muestra el resultado de haber ejecutado el programa descrito en 3.13 pero utilizando la nueva función para el despliegue del histograma acumulativo.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Función que permite generar los puntos
%del histograma acumulativo
%a partir del vector 1xN entregado por el
%bloque "2-D Histogram"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Función que permite generar los puntos
%del histograma acumulativo
%a partir del vector 1xN entregado por el
%bloque "2-D Histogram"
function union = concatena(u)
% La funcion recibe en u al vector 1xN

%Se genera el vector num que especifica
%el valor de la abcisa
num=1:256;
%Se inicializa la variable utilizada para la recursión a
%cero
va=0;
%Se aplica el modelo definido en la Ecuación 3.2
for v=1:256
    H(v)=va+u(v);
    va=H(v);
end

%El Valor del vector acumulativo se escala y se le
%resta un valor para generar el efecto gráfico
%Se escala porque normalmente el valor entregado por
%el bloque "2-D Histograma" esta normalizado,
%en este caso 200 es una buena opción para la
%visualización

val=288-(H*200);
%Se empaquetan los datos que serán los
%puntos entregados al bloque "Draw Markers"
%para graficarlos sobre una imagen sintética
%como tapete
union=[val;num];
```

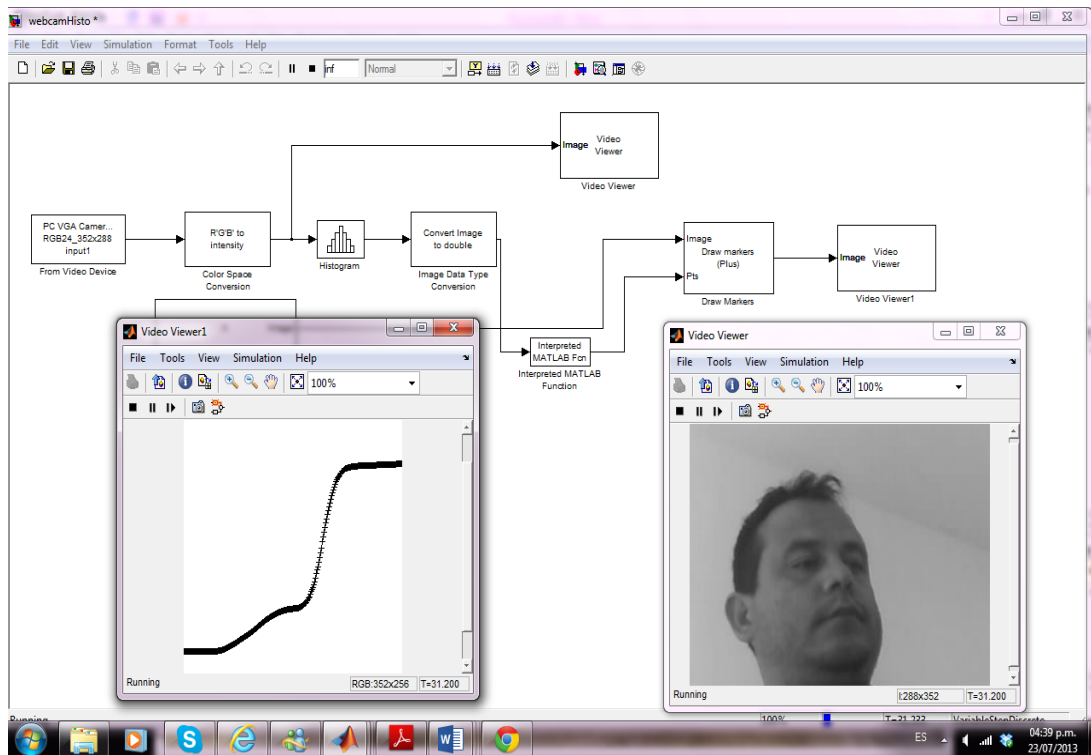


Figura 3.19 Muestra el resultado de ejecutar el programa en Simulink descrito en la figura 3.13, pero utilizando la función definida en el programa 3.3 para el cálculo y despliegue del histograma acumulativo.

**Tarea realizar una función en Matlab con parámetros y sin ellos.**