

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

**DISEÑO E IMPLEMENTACION DE UN SISTEMA DE
MONITOREO DE TRAFICO VEHICULAR UTILIZANDO
PROCESAMIENTO DIGITAL DE IMAGEN.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TITULO DE INGENIERO
EN ELECTRÓNICA Y CONTROL**

**CARRERA HERRERA DENNIS JHONATAN
MOYA POZO DANIEL LEONARDO**

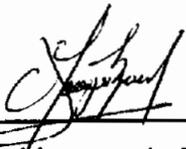
DIRECTORA: ING. YADIRA BRAVO

Quito, Mayo 2005

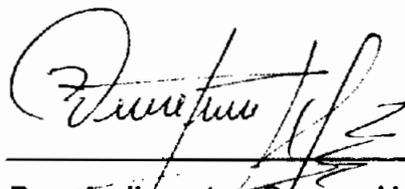
DECLARACIÓN

Nosotros, Daniel Leonardo Moya Pozo y Dennis Jhonatan Carrera Herrera, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Daniel Leonardo Moya Pozo



Dennis Jhonatan Carrera Herrera

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Daniel Leonardo Moya Pozo y Dennis Jhonatan Carrera Herrera, bajo mi supervisión.

A handwritten signature in black ink, appearing to read 'Yadira Bravo', written over a horizontal line.

Ing. Yadira Bravo

CONTENIDO

RESUMEN	vi
PRESENTACION	vii

CAPITULO 1: PROCESAMIENTO DE SECUENCIA DE IMÁGENES

1.1 INTRODUCCION	1
1.2 VIDEO DIGITAL	2
1.2.1 INTRODUCCION AL VIDEO DIGITAL.....	2
1.2.2 CODIFICACIÓN MPEG.....	4
1.2.3 FORMATOS DE VIDEO DIGITAL.....	5
1.3 IMAGENES DIGITALES	6
1.3.1 DIGITALIZACIÓN.....	7
1.3.2 PROPIEDADES DE UNA IMAGEN DIGITAL.....	9
1.3.2.1 Resolución.....	9
1.3.2.2 Definición.....	10
1.3.2.3 Numero de Planos.....	10
1.3.3 TIPOS DE IMÁGENES.....	11
1.3.3.1 Imágenes en escala de grises.....	11
1.3.3.2 Imágenes a Color (True Color).....	11
1.3.3.3 Imágenes Complejas.....	12
1.3.4 VISUALIZACION DE IMÁGENES DIGITALES.....	12
1.3.4.1 Espacios de color.....	12
1.3.4.1.1 Espacio de Color RGB.....	13
1.3.4.1.2 Transformaciones entre Espacios de Color.....	13
1.3.4.2 Paletas de color.....	14

1.3.5 ARCHIVOS DE IMÁGENES	15
1.3.5.1 Tipos de formatos para imágenes digitales	15
1.4 ANÁLISIS Y PROCESAMIENTO DIGITAL DE IMÁGENES	17
1.4.1 ANALISIS DE IMÁGENES	17
1.4.2 PROCESAMIENTO DE IMÁGENES	18
1.4.2.1 Operadores aritméticos y lógicos	20
<i>1.4.2.1.1 Operadores Aritméticos</i>	20
<i>1.4.2.1.2 Operadores Lógicos y de Comparación</i>	21
1.4.2.2 Convolución Kernel	22
1.4.2.3 Filtrado Espacial	24
<i>1.4.2.3.1 Filtros Lineales</i>	25
<i>a) Filtro Gradiente</i>	26
<i>b) Filtro Laplaciano</i>	27
<i>c) Filtro de aplanamiento (smoothing)</i>	28
<i>d) Filtro Gaussiano</i>	29
<i>1.4.2.3.2 Filtros No – Lineales</i>	29
1.4.2.4 Análisis de partículas (blob)	29
<i>1.4.2.4.1 Umbralización (Thresholding)</i>	30
<i>1.4.2.4.2 Morfología Binaria</i>	31
<i>a) Elementos Estructurales</i>	31
<i>b) Conectividad</i>	33
<i>1.4.2.4.3 Operaciones morfológicas primarias</i>	34
<i>a) Funciones Erosión y Dilatación</i>	34
<i>b) Hit – miss (ganancia o pérdida)</i>	36
<i>c) Función Auto – Median</i>	38
<i>1.4.2.4.4 Operaciones morfológicas avanzadas</i>	38
<i>a) Border</i>	39
<i>b) Hole Filling</i>	39
<i>c) Labeling</i>	39
<i>d) Separation</i>	39
<i>e) Circle</i>	40
<i>f) Convex</i>	40

1.4.2.4.5 <i>Medición de Partículas</i>	41
a) <i>Área</i>	41
b) <i>Longitud</i>	42
1.4.2.5 Emparejamiento de modelo (Pattern Matching)	43
1.4.2.5.1 <i>Técnicas de Emparejamiento</i>	45
a) <i>Correlación Cruzada</i>	45
b) <i>Emparejamiento Piramidal</i>	47
c) <i>Emparejamiento de Escala – Invariante</i>	47

CAPITULO 2: DISEÑO DEL CONTROL PARA EL MONITOREO Y CONTEO DE VEHÍCULOS

2.1 INTRODUCCIÓN	49
2.2 CAPTURA DE VIDEO	50
2.3 DESCARGA DE VIDEO	51
2.3.1 INTERFAZ IEEE 1394	52
2.4 SEGMENTACION DE VIDEO A IMAGENES	55
2.5 DISEÑO DEL PROGRAMA PRINCIPAL	59
2.6 DISEÑO DE LOS ALGORITMOS DE CONTROL	61
2.6.1 DISEÑO DEL ALGORITMO DE RECONOCIMIENTO EMPAREJAMIENTO DE MODELO	61
2.6.1.1 <i>Proceso de Aprendizaje</i>	64
2.6.1.2 <i>Proceso de Búsqueda</i>	65
2.6.2 DISEÑO DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE RUEDAS	68
2.6.2.1 <i>Pre-procesamiento</i>	69
2.6.2.2 <i>Umbralización (Thresholding)</i>	71
2.6.2.3 <i>Morfología Binaria</i>	72
2.6.2.4 <i>Medición de partículas</i>	75

2.6.3 DISEÑO DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE FORMA.....	78
---	-----------

CAPITULO 3: IMPLEMENTACION Y DESARROLLO DEL PROGRAMA DE CONTROL

3.1 INTRODUCCIÓN.....	88
3.2 IMPLEMENTACION DEL ALGORITMO DE RECONOCIMIENTO EMPAREJAMIENTO DE MODELO.....	88
3.2.1 PROCESO DE APRENDIZAJE.....	89
3.2.2 PROCESO DE BUSQUEDA.....	94
3.3 IMPLEMENTACION DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE RUEDAS.....	101
3.4 IMPLEMENTACION DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE FORMA.....	110
3.5 IMPLEMENTACION DEL PROGRAMA PRINCIPAL.....	118
3.5.1 CONTROL PRINCIPAL.....	120
3.5.2 VENTANA DE VISUALIZACION NORMAL (MONITOREO VEHICULAR).....	121
3.5.3 CONTROLADORES PARA DETECCION DE PERTURBACION... 	122
3.5.4 VENTANA DE CALIBRACION.....	125
3.5.5 CONTROLADORES DE CONFIGURACION GENERAL.....	126
3.6 DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL.....	129

CAPITULO 4: PRUEBAS Y RESULTADOS

4.1 INTRODUCCIÓN.....	138
------------------------------	------------

4.2 RESULTADOS DE LOS DISTINTOS ALGORITMOS DE CONTROL.....	138
4.2.1 RESULTADOS DEL ALGORITMO DE RECONOCIMIENTO EMPAREJAMIENTO DE MODELO.....	146
4.2.2 RESULTADOS DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE RUEDAS.....	148
4.2.3 RESULTADOS DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE FORMA.....	150
4.3 RESULTADO TOTAL DEL SISTEMA DE CONTROL VEHICULAR.....	153
4.4 EJEMPLO DEL PROCESO DE CALIBRACION.....	154

CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES.....	157
5.2 RECOMENDACIONES.....	159

REFERENCIAS BIBLIOGRAFICAS.....	162
--	------------

ANEXOS.....	163
--------------------	------------

RESUMEN

En el presente proyecto se presenta el diseño e implementación de un sistema de instrumentación virtual para el monitoreo y conteo de vehículos, utilizando el procesamiento digital de imágenes con ayuda del software LabVIEW, con el fin de crear una herramienta fácil de utilizar en aplicaciones relacionadas con el control de tráfico vehicular que sea de bajo costo y sobre todo de tamaño reducido.

En primera instancia, en el Capítulo 1, se analizan tópicos importantes sobre las imágenes digitales estáticas y los distintos tipos de procesamiento digitales que son aplicadas a éstas para poder extraer información necesaria a fin de realizar el análisis en general de una secuencia de imágenes.

En el Capítulo 2, se describe los pasos seguidos para detección de los vehículos empezando por la captura, edición y segmentación del video en secuencias de imágenes, además de diseñar los distintos algoritmos de reconocimiento vehicular utilizando el paquete computacional de LabVIEW que posee extensas librerías de funciones como lo es IMAQ Vision dedicada al procesamiento digital de imágenes.

Luego se presenta en el Capítulo 3, la implementación misma del sistema de control de tráfico vehicular desarrollado completamente en el ambiente de programación gráfica de LabVIEW. Se describe también programa principal de control y sus algoritmos mediante diagramas de flujo.

El Capítulo 4, describe y analiza los resultados de los distintos algoritmos de manera independiente y del sistema de control general implementado. Y un breve ejemplo del sistema en modo de calibración.

Y en el último Capítulo, se concluye que los objetivos y alcances tratados para el desarrollo y ejecución de este proyecto se han cumplido en su totalidad. Se recomienda también los requerimientos del sistema para aplicarse en tiempo real.

PRESENTACION

Los sistemas de control vehicular en las últimas décadas se han basado en sensores de luz infrarroja, magnéticos, las tradicionales células fotoeléctricas, galgas extensométricas entre otros.

El sistema de detección de vehículos con sensores de luz infrarroja no tiene la necesidad de instalar dispositivos en el pavimento, consta de dos columnas de 1.5 metros situadas a ambos lados de cada vía de paso, lo que permite extraer información de cada vehículo que atraviesa el sistema de planos de luz infrarroja.

El sistema con campos magnéticos detecta vehículos a través de las alteraciones producidas por los mismos en el campo magnético terrestre. Está basado en un fiable y sensible sensor magneto-métrico capaz de detectar variaciones de la diezmilésima parte del campo magnético terrestre.

Los sistemas de visión artificial y su aplicación en reconocimiento de entornos es un campo en constante desarrollo. Las primeras investigaciones comenzaron en la década de 1970, y desde entonces cada vez son más las aplicaciones industriales, científicas y de investigación que hacen uso de ella.

Un paso más en la "escala evolutiva" de la visión artificial es aplicada al movimiento de un robot. Este es un problema para el que, pese a los avances logrados, todavía no se ha hallado una solución idónea.

Actualmente se trabaja en el desarrollo de complejos algoritmos que, intentando imitar las funciones del cerebro humano, se centran en los tres principales problemas: la detección y reconocimiento de un objeto, diferenciándolo de los demás, la distancia que le separa de él, y la detección de objetos en movimiento. Para poder precisar la distancia que le separa de un punto, al robot o sistema de visión se le dota de visión estereoscópica; es decir, de dos o más cámaras cuyas imágenes en dos dimensiones, al igual que en el sistema de visión humano, son fusionadas por un procesador para generar una vista tridimensional. La

separación de distintos objetos en una imagen se realiza por métodos estadísticos: la imagen es procesada y pixelizada en matrices de puntos a las que se aplican diversos gradientes: de color, de luminosidad, escala de grises, contraste, etc. Este proceso se conoce con el nombre de "segmentación". El área que contenga la misma información o muy aproximada debe conformar un objeto. Una vez que el objeto está reconocido, se puede almacenar. Un ejemplo son los parches, que son unidades de información que tienen datos como el de su silueta, además de su color y posición en la pantalla. Estos parches pueden ser, al mismo tiempo que se detectan, comparados con los que ya existan en una base de datos. En cuanto a la detección de objetos en movimiento por observadores que también pueden estar moviéndose, existen estudios basados en dos hechos que se dan para estos casos. Primero: cuando el observador se mueve, la velocidad proyectada sobre cualquier punto de la imagen es un vector cuya orientación depende del movimiento del observador. Si conocemos este movimiento, conocemos la orientación. El truco consiste en que un objeto moviéndose en la pantalla tendrá un vector de velocidad cuya orientación no suele coincidir con la del movimiento del observador. El segundo hecho es aún más impreciso, y consiste en que el observador (robot) normalmente se mueve lentamente, mientras que un objeto en la pantalla suele moverse más de prisa; luego si se supera un límite de velocidad de movimiento, es indicativo de que hay algo ahí fuera moviéndose, en vez de moverse uno mismo.

En el presente proyecto de titulación se presenta el diseño e implementación de un sistema moderno de instrumentación virtual para el monitoreo y conteo de vehículos, utilizando el procesamiento digital de imágenes para la detección de vehículos en movimiento a partir de secuencia de imágenes. Con el fin de crear una herramienta fácil de utilizar en aplicaciones relacionadas con el control de tráfico vehicular que sea de bajo costo y tamaño reducido, ya que a la final se requiere de una cámara digital y de un computador personal en el cual se visualice el paso de los vehículos al mismo tiempo que los cuente para tomar acciones como por ejemplo una base de datos del número de vehículos que ingresaron a un parqueadero, y sobretodo sin la presencia de un operario.

CAPITULO 1

PROCESAMIENTO DE SECUENCIA DE IMAGENES

1.1 INTRODUCCION

Hasta hace poco tiempo los sistemas de visión artificial operaban extrayendo solamente información de imágenes estáticas, para lo cual podían operar a velocidades relativamente lentas y con una cantidad de información limitada en cierto sentido. Hoy en día, es posible efectuar procesamiento de imágenes a mayor velocidad y procesar mayores volúmenes de información, por esta razón se comienza a buscar algoritmos para el procesamiento de secuencias de imágenes. Sin embargo, hasta el momento la extracción de información de secuencia de imágenes en tiempo real y en forma satisfactoria no puede considerarse un objetivo totalmente alcanzado.

La extracción de información de secuencias de imágenes es una tarea clave en muchos sistemas de visión artificial. Esta información puede ser útil para la solución de problemas tales como la obtención de información referente al desplazamiento de objetos, la medición de velocidad de objetos, la modelación tridimensional del entorno, la restauración de imágenes, etc.

El seguimiento y caracterización dinámica de objetos en movimiento permite la utilización de la visión artificial en aplicaciones muy útiles y prometedoras. En aplicaciones como la de control de tráfico de automóviles, es posible ajustar automáticamente y a un bajo costo los sistemas de señalización en función de las características del tráfico. Pueden utilizarse también para aplicaciones de seguridad, donde un operario humano puede distraerse con facilidad.

Los siguientes tópicos estudiarán conceptos importantes sobre las imágenes digitales estáticas y los distintos tipos de procesamiento digitales que son aplicadas a éstas para poder extraer información necesaria a fin de realizar el análisis en general como si se tratara de una secuencia de imágenes.

1.2 VIDEO DIGITAL

1.2.1. INTRODUCCION AL VIDEO DIGITAL [1]

La información de video es provista en una serie de imágenes ó "cuadros" y el efecto del movimiento es llevado a cabo a través de cambios pequeños y continuos en los cuadros. Debido a que la velocidad de estas imágenes es de 30 cuadros por segundo, los cambios continuos entre cuadros darán la sensación al ojo humano de movimiento natural. Las imágenes de video están compuestas de información en el dominio del espacio y el tiempo. En los sistemas de video digital, cada cuadro es muestreado en unidades de píxeles ó elementos de imagen (frames).

El video al ser una secuencia de imágenes, se puede caracterizar de la siguiente manera:

- Resolución Horizontal (anchura): en píxeles (160, 320, 352, 640, 720, 1280, 1920)
- Resolución Vertical (altura): en píxeles (120, 240, 480, 485, 720, 1080)
- Definición: en bits (1, 4, 8, 16, 24)
- Velocidad de Refresco: en frames o cuadros por segundo – fps (5, 15, 20, 24, 25, 30)
- Velocidad de transferencia: se refiere al consumo de kbits que hace el video por segundo. Mientras más alto es su valor, el video será de mejor calidad pero disminuye la capacidad de almacenamiento.

El tamaño de un archivo de video sin comprimir se define como:

$$\text{Anchura} * \text{Altura} * \text{Definición}(\text{bytes}) * \text{fps} = \frac{\text{bytes}}{\text{seg}} \quad (\text{Ec. 1.1})$$

Por ejemplo para un video de resolución 800*600 a color y a una frecuencia de 25 fps, 1 segundo de dicho video ocupará un espacio de 36 MB.

La información de video ocupa gran cantidad de espacio en memoria de disco; por lo que, para transmisión o almacenamiento, se requiere de la compresión (o codificación) de la imagen. Al utilizar alguna técnica de compresión, se introduce un determinado factor de compresión, y de esta manera el tamaño de un archivo de video comprimido será:

$$\text{Anchura} * \text{Altura} * \text{Definición}(\text{bytes}) * \text{fps} * \text{factor_compresión} = \frac{\text{bytes}}{\text{seg}} \quad (\text{Ec. 1.2})$$

Para el archivo de video del ejemplo anterior y usando un factor de compresión de 10, se tendrá que 1 segundo de video ocupará un espacio de 3,6 MB.

Para introducir el concepto de compresión de video, se debe primeramente diferenciar los conceptos de formato y codec. Un formato de video digital, es la manera en que se guardan los datos en el fichero, el mismo que puede cumplir diferentes requisitos según el uso para el que esté diseñado, en tanto que el codec es la compresión algorítmica a la que se ha visto sometido el contenido del formato de video digital, y está relacionado con los siguientes aspectos: factor de compresión, velocidad de proceso y calidad de la información comprimida. De esta manera, es muy posible que el mismo codec esté insertado en diferentes formatos de video.

Los tipos de codecs más utilizados son los siguientes:

- No comprimido
- DV (comprime solo video)
- M – JPEG (motion JPEG)
- MPEG (comprime audio y video)
- DivX

1.2.2 CODIFICACIÓN MPEG

Originalmente se diferenciaron cuatro tipos, MPEG-1, 2, 3 y 4. La principal diferencia entre ellos es la calidad de imagen que ofrecen y el ancho de banda que necesitan. Ofrecen tres ventajas fundamentales: un gran nivel de compresión, escasa pérdida de calidad, y permiten la compatibilidad con carácter retroactivo entre diferentes formatos de video.

La cadena de datos MPEG se puede dividir en tres capas o partes bien diferenciadas: una para la compresión de audio, otra para la de vídeo y una tercera relativa al sistema, encargada de la sincronización de datos, calidad de imagen, etc.

Actualmente se usan 3 formatos de compresión MPEG:

- **MPEG-1:** diseñado para introducir video en un CD-ROM, la velocidad de transferencia queda limitada a 1.5 Mbits y la resolución a 352x240. Su calidad es similar al VHS y es el usado en el VCD y CVCD.
- **MPEG-2:** ofrece mayor calidad con mayor ancho de banda (típicamente de 3 a 10 Mbits). En esa banda, proporciona 720x486 píxeles de resolución; es decir, calidad TV. Ofrece compatibilidad con MPEG-1. Es el usado en los DVD's, SVCD's, XVCD's y en las televisiones digitales (satélite y cable).
- **MPEG-4:** Se trata de un formato de muy bajo ancho de banda y resolución de 176x144, pensado para videoconferencias y es muy usado en los vídeos DivX.

Un archivo MPEG está compuesto de unas secuencias cíclicas llamadas GOP (grupo de imágenes: *I-frames*, *P-frames* y *B-frames*), las cuales engloban normalmente 15 imágenes. Aunque no es necesario que estos grupos estén formados siempre por el mismo número de fotogramas, se suele asignar el mismo patrón GOP en todo el archivo MPEG.

Los GOP's están formados por tres grupos distintos de imágenes:

- **I-picture (Intra frames):** son los únicos estrictamente necesarios. Cada cuadro es comprimido con un tipo de compresión llamada "Intra-frame DCT coding" (codificación interna de cuadros DCT por transformación discreta de coseno [2]) dividiendo la imagen en grupos de 8x8 píxeles. Si el GOP tan sólo contara con cuadros-I se tendría una secuencia de JPG's, el llamado Motion JPEG.
- **P-picture y B-picture (Predictive frames y Bidirectionally-predictive frames).** Los cuadros-P analizan los cambios con respecto a cuadros I u otros cuadros P anteriores, mientras que los cuadros-B pueden analizar los cambios de cuadros-P anteriores y posteriores ("B" de bidireccionales) alcanzando los mayores grados de compresión.

Se debe elegir el mejor número de cuadros P y B (dependiendo del video). Secuencias largas de cuadros P y B lograrán una estupenda compresión sin afectar la calidad, pero cuando el vídeo es cambiante, hay muchos cuadros en los que la información entre cuadro y cuadro cambia al 100%, las técnicas de análisis y predicción no resultan muy efectivas. Si el vídeo contiene mucho movimiento quizás convendría reducir o eliminar el número de cuadros P y B en cada GOP. Para el formato DVD la secuencia GOP contiene un máximo de 15 cuadros (también recomendada para formatos de vídeo en CD) y, normalmente, suele ser:]I BB P BB P BB P BB P BB[.

1.2.3 FORMATOS DE VIDEO DIGITAL

Los formatos más comunes para la visualización de imágenes dinámicas en PC's son los siguientes:

- AVI y AVI 2.0 (.avi)
- Microsoft Windows Media Video (.asf .wmv .wma)
- Real Video
- Apple Quicktime (.mov)

Ahora los formatos que utilizan soportes ópticos para el almacenamiento de video son los siguientes:

- Video – CD (VCD): VCD 2.0 (estándar), CVCD, HQ – VCD y XVCD.
- Super – Video CD (SVCD): SVCD 1.0 (estándar) y XSVCD.
- DVD

En la siguiente Tabla se muestra las características más importantes de los formatos VCD 2.0 y SVCD 1.0.

<i>Característica</i>	Video – CD 2.0	Super – Video CD 1.0
Video	MPEG-1	MPEG-2
<i>Bit rate</i>	constante 1,150 Kbps	variable hasta 2.6 Mbps
<i>Resolución NTSC</i>	352x240 , 29.97Hz	480x480 entrelazado, 29.97Hz
<i>Resolución PAL</i>	352x288 , 25Hz	480x576 entrelazado, 25Hz
Imágenes	MPEG-1	MPEG-2 (1 frame)
<i>Resolución NTSC</i>	352x240, 704x480	480x480, 704x480
<i>Resolución PAL</i>	352x288, 704x576	480x576, 704x576
Audio	MPEG-1 layer II	MPEG-1 layer II
<i>Muestreo</i>	44.1 KHz	44.1 KHz
<i>Bit rate</i>	224 Kbps	desde 32 hasta 384 Kbps
<i>Canales de audio</i>	1 estéreo o 2 mono	hasta 2 estéreo ó 4 mono
<i>Sonido Surround</i>	estéreo Dolby Pro-Logic	extensión MPEG-2 (5+1)

Tabla 1.1 Características de video, imágenes y audio para formatos VCD y SVCD [1]

1.3 IMAGENES DIGITALES [3]

Las ventajas de la imagen digital frente a la analógica vienen dadas por las múltiples posibilidades de manipulación que ofrecen. Además se puede procesar dichas imágenes para obtener información. Reconocer, contar y medir tamaño, forma, posición o densidad de determinados objetos; es algo que, con una correcta preparación de la muestra, está al alcance de cualquier ordenador personal complementado con un software adecuado de análisis de imágenes.

Al tomar una fotografía a los objetos se puede obtener imágenes. La obtención de dichas imágenes depende de dos factores: la naturaleza de la iluminación y la forma en que las caras del objeto reflejan la iluminación hacia la cámara.

En la siguiente Figura se puede ver claramente el espacio de los objetos y de las imágenes.

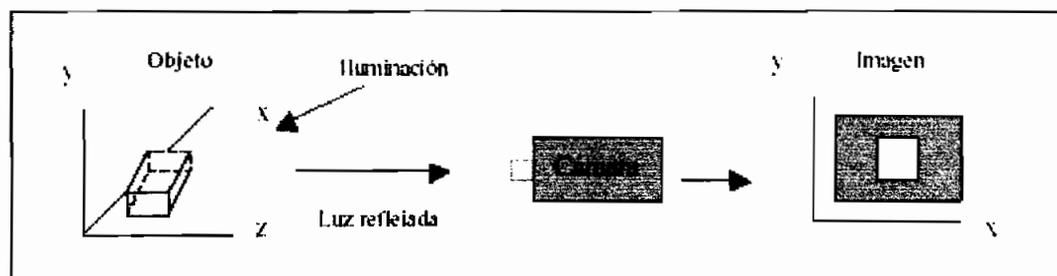


Figura 1.1 Representación del espacio de un objeto y del espacio de una imagen

1.3.1 DIGITALIZACIÓN

Una imagen natural capturada con una cámara, un telescopio, un microscopio o cualquier otro tipo de instrumento óptico presenta una variación de sombras y tonos continuos. Imágenes de este tipo se llaman *imágenes analógicas*.

La transformación de una imagen analógica a otra discreta se llama *digitalización* y es el primer paso en cualquier aplicación de procesamiento de imágenes digitales. Este proceso consta de dos partes: *muestreo* y *cuantificación*

Un muestreo consiste en una subdivisión de la imagen analógica en porciones que envuelven polígonos regulares como triángulos, cuadrados y hexágonos.

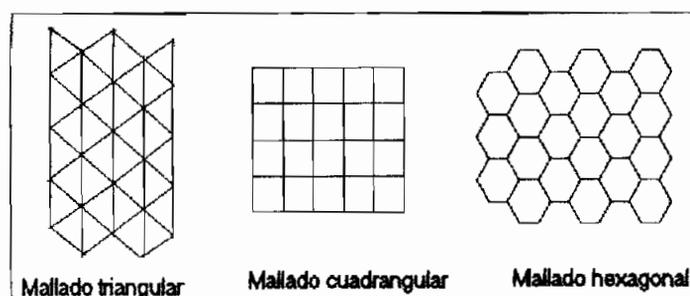


Figura 1.2 Tipos de polígonos en el proceso de muestreo de imágenes

Estos polígonos representan sensores sensibles a la intensidad de luz.

La salida de estos sensores es un valor (amplitud) dentro de una escala (color).

El resultado puede ser, o bien un único valor (escala de grises) o bien un vector con tres valores por polígono (RGB) que se corresponden con la intensidad de color rojo (R), verde (G) y azul (B). Las imágenes en escala de grises con sólo 2 colores: blanco y negro, se llaman *imágenes binarias*. A este proceso de discretización del color se le llama *cuantificación*. Un polígono de color constante se llama *píxel*. El esquema de subdivisión más común, es el de la cuadrícula rectangular que a continuación se muestra:

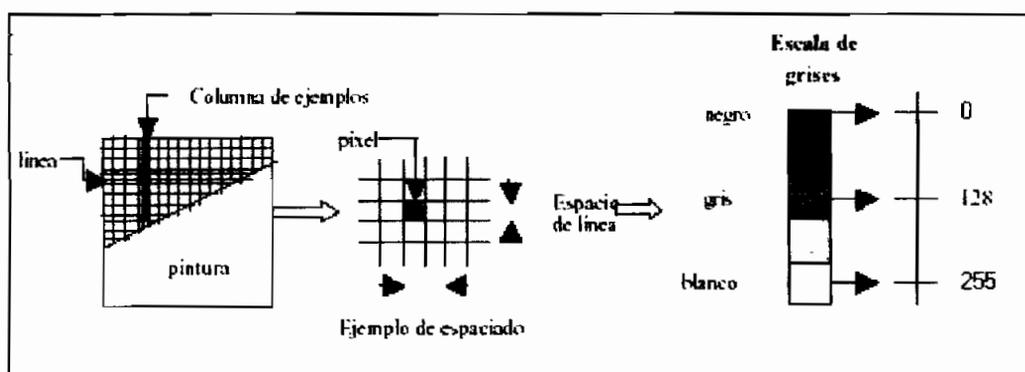


Figura 1.3 Digitalización de una imagen

En la siguiente Figura se observa un ejemplo de digitalización de una imagen. El muestreo se ha hecho usando un mallado cuadrangular de 9 por 9 cuadrados y la cuantificación consiste en una paleta de 256 niveles de gris (donde 0 indica el color negro y 255 el color blanco).



Figura 1.4 Creación de una imagen digital

Por tanto una imagen digital está dada por una matriz bi-dimensional de valores que representan la intensidad de luz. Es una función de la intensidad de luz.

$$f(x, y)$$

Donde f representa el brillo del punto (x, y) , x e y representan las coordenadas espaciales de un píxel (elemento de cuadro).

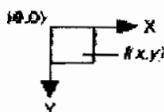


Figura 1.5 Referencia espacial del píxel (0,0)

1.3.2 PROPIEDADES DE UNA IMAGEN DIGITAL

Una imagen digitalizada tiene tres propiedades básicas: resolución, definición y número de planos.

1.3.2.1 Resolución

Representa el número de filas y de columnas de píxeles. Una imagen compuesta de m columnas y n filas tiene una resolución de $m \times n$. Esta imagen tiene m píxeles a lo largo de su eje horizontal y n píxeles a lo largo de su eje vertical.

La siguiente Figura muestra el efecto que tiene la resolución en una imagen del mismo tamaño.

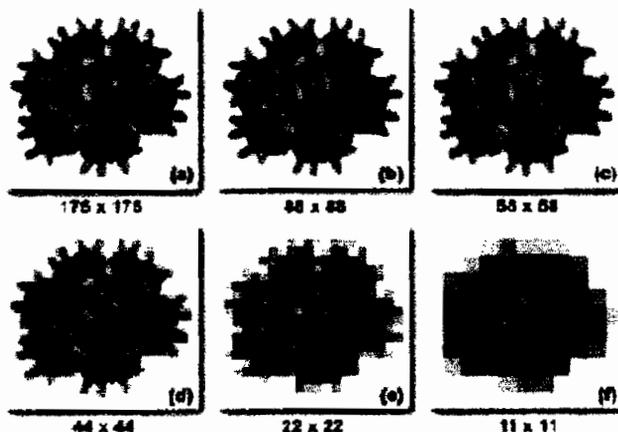


Figura 1.6 Efecto de la resolución espacial

1.3.2.2 Definición

La definición de una imagen indica el grado de nitidez que se puede observar en una imagen. El bit de profundidad de una imagen es definido como el número de bits usado para codificar una imagen. Para un número dado n de bit de profundidad, la imagen tiene una definición de 2^n ; es decir, un píxel puede tomar 2^n valores diferentes. Por ejemplo si $n = 8$ bits, un píxel puede tomar 256 valores diferentes que pueden variar desde 0 hasta 255. En la siguiente Figura se puede observar como varia la definición (nitidez) de una imagen en función de distintos valores de bits de profundidad.

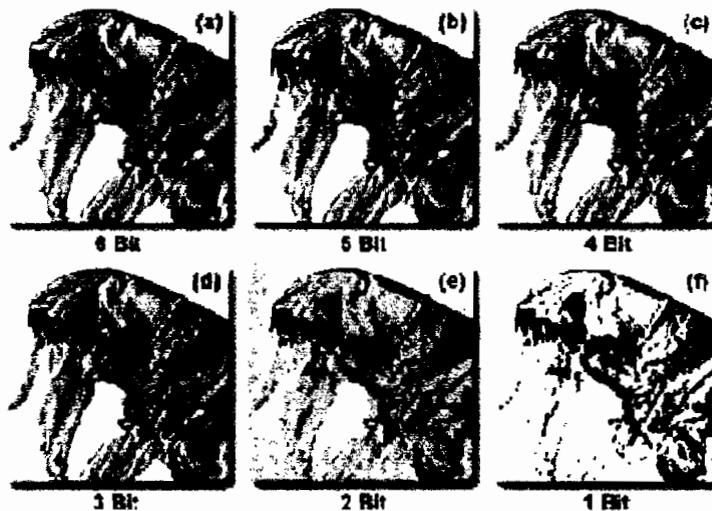


Figura 1.7 Influencia del bit de profundidad en la definición de una imagen

1.3.2.3 Número de Planos

El número de planos en una imagen corresponde al número de matrices de píxeles que componen una imagen. Una imagen en escala de grises o de pseudo-color está compuesta de un solo plano, mientras que una imagen a color está compuesta de tres planos, uno para la componente roja, para la componente verde y uno para la componente azul.

1.3.3 TIPOS DE IMÁGENES

1.3.3.1 Imágenes en escala de grises

Una imagen en escala de grises está compuesta de un simple plano de píxeles. Cada píxel es codificado usando un solo número. Este número puede ser:

- Un entero sin signo de 8 bits representando valores de escala de grises entre 0 y 255.
- Un entero con signo de 16 bits representando valores de escala de grises entre -32768 y +32767.
- Un simple número de precisión flotante (codificado mediante 4 bytes) representando valores de escala de grises desde $-\infty$ a ∞ .

1.3.3.2 Imágenes a Color (True Color)

Una imagen a color es codificada en memoria como una imagen RGB o una imagen HSL. Los píxeles de una imagen a color son una composición de 4 valores. Las imágenes RGB guardan la información del color utilizando 8 bits tanto para el rojo (R), como para el verde (G) y para el azul (B). Las imágenes HSL guardan la información del color utilizando 8 bits tanto para el matiz (H), como para la saturación (S) y para la luminancia (L).

En todos los modelos a color, existe un valor de 8 bits que no es usado (plano alfa), que representa la transparencia de ese punto y a veces es utilizado para combinar distintas fuentes de imagen; teniéndose así una codificación de 32 bits (4 x 8 bits) como se indica en la siguiente Figura.

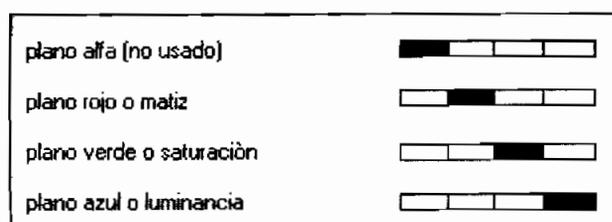


Figura 1.8 Codificación para imágenes RGB y HSL

1.3.3.3 Imágenes Complejas

Una imagen compleja contiene información de frecuencia de una imagen en escala de grises luego de haber aplicado a ésta la transformada rápida de fourier (FFT) [2][3], y así de esta manera se puede permitir operaciones en el dominio de la frecuencia sobre la imagen.

Cada píxel en una imagen compleja es codificada mediante dos números simples de precisión flotante que representan las componentes real e imaginaria del píxel complejo. De una imagen compleja se pueden extraer las siguientes componentes: parte real, parte imaginaria, magnitud y fase.

1.3.4 VISUALIZACION DE IMÁGENES DIGITALES

1.3.4.1 Espacios de color

Los espacios de color permiten representar un determinado color y son un subespacio con un sistema de coordenadas 3D donde cada color es representado por un punto. Se puede usar espacios de color para facilitar la descripción de colores entre personas, máquinas o programas (software).

Los espacios de color más importantes son los siguientes:

- **RGB:** basado en rojo, verde y azul. Usado por computadores para desplegar imágenes.
- **HSL:** basado en el matiz, saturación y luminancia. Usado en aplicaciones de procesamiento de imágenes.
- **CIE:** basado en brillo, matiz y llenura de color. Definido por la Comisión Internacional de Iluminación como las diferentes sensaciones de color que el cerebro humano percibe.
- **CMY:** basado en azul verdoso (cyan), magenta y amarillo. Usado en la industria de la imprenta.
- **YIQ:** separa la información de luminancia (Y) de la información del color (I y Q). Usado para transmisión de televisión.

1.3.4.1.1 Espacio de Color RGB

Este espacio de color se lo puede visualizar como un cubo donde el rojo, verde y azul se encuentran en las esquinas de cada eje como lo muestra la Figura 1.9. El color negro se encuentra en el origen, mientras que el blanco en la esquina opuesta del cubo. Cada lado del cubo tiene valores entre 0 y 1 (normalizados). La línea diagonal del cubo desde el negro (0,0,0) hasta el blanco (1,1,1) representa los valores en escala de grises; es decir, cuando todas las componentes R, G y B son iguales en magnitud.

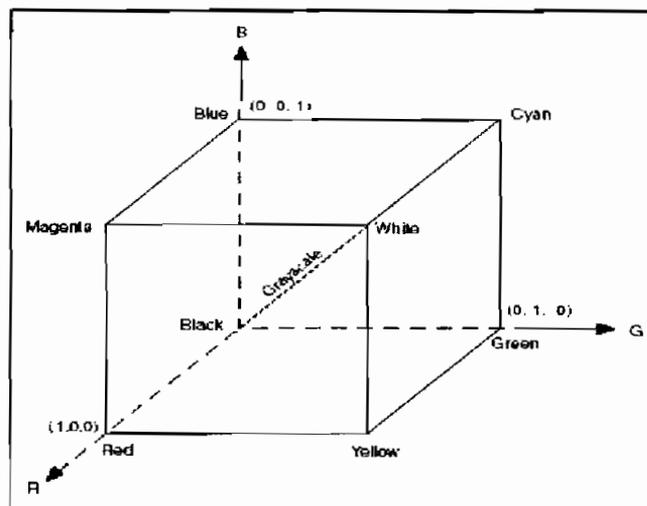


Figura 1.9 Espacio de color RGB

1.3.4.1.2 Transformaciones entre Espacios de Color

En algunos casos, son más apropiados modelos diferentes del RGB para algoritmos y aplicaciones específicas.

Sabiendo que R , G y B son los valores codificados de las tres componentes de un píxel a color y pueden variar de 0 a 255, y ng (niveles de gris) representa el valor de un píxel en escala de grises; por lo tanto, la forma de obtener una imagen de éste tipo a partir de una en RGB, es aplicando al valor RGB de cada píxel la siguiente fórmula:

$$ng = 0.299R + 0.587G + 0.114B \quad (\text{Ec. 1.3})$$

Una conversión alternativa para obtener imágenes en escala de grises es mediante un simple promedio:

$$ng = (R + G + B) / 3 \quad (\text{Ec. 1.4})$$

1.3.4.2 Paletas de color

Con este esquema, se crea una tabla en la que se almacenan los colores que aparecen en la imagen, en forma de componentes R, G y B. Esta tabla recibe el nombre de paleta de color.

La matriz de la imagen almacena el índice de la entrada en la paleta que contiene el color del punto que se esté referenciando. Este esquema resulta interesante siempre que el número de entradas de la paleta, es decir, el tamaño del índice en bytes, sea menor que el tamaño en bytes de la definición de color. Generalmente se tiene paletas de 256 colores con lo cual con un solo byte se puede direccionar todas sus entradas, y cada entrada de la paleta almacena 3 bytes, uno por cada componente de color.

El formato de paleta de color es útil cuando se trata solamente de representar imágenes, puesto que de cara a su procesamiento es necesario convertirla a formato true-color, lo cual se hace de forma inmediata sustituyendo cada índice de la matriz de la imagen, por el contenido de la paleta apropiado. La razón por la que no se puede procesar las imágenes con un formato de paleta de color, es debido a que cuando se aplica algún tipo de procesado a la imagen se generan nuevos colores que no están definidos en la paleta, con lo cual no pueden ser representados.

En la siguiente Figura se muestra las gráficas que representan las tablas de color asignada a cada paleta de color más utilizados, el eje horizontal del gráfico representa el rango de entrada en niveles de gris (0, 255), y el eje vertical representa la intensidad RGB asignada a un valor de nivel de gris dado.

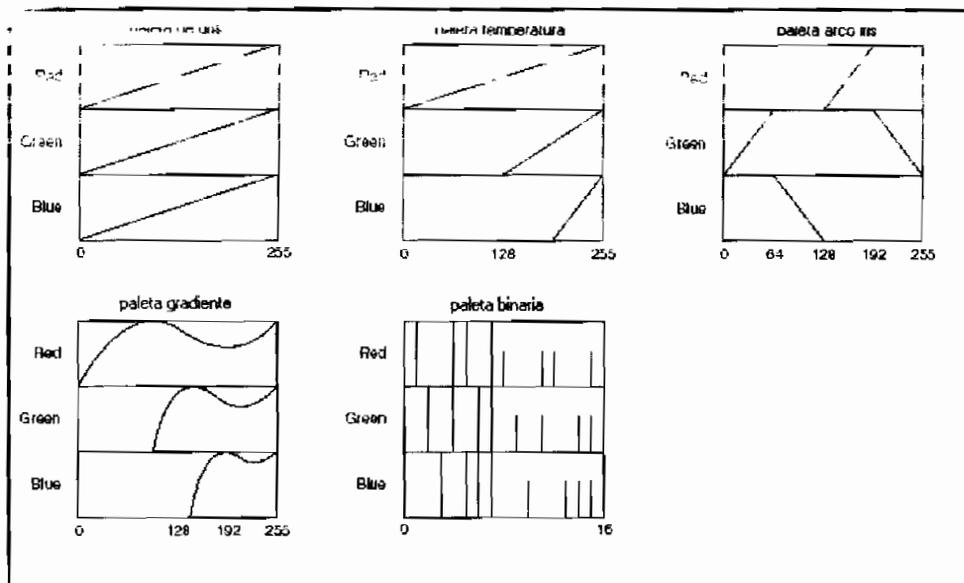


Figura 1.10 Paletas de color

1.3.5 ARCHIVOS DE IMAGENES

Un archivo de imagen está compuesto de una cabecera seguido por los valores de los píxeles. Dependiendo del formato del archivo, la cabecera puede contener información acerca de la resolución horizontal, vertical, definición de píxel y la paleta original. Se requiere por ende algún tipo de compresión ya sea para su almacenamiento o para su transmisión. Una compresión sin pérdidas devuelve la imagen descomprimida exactamente igual a la original. Por el contrario, la compresión con pérdidas acepta alguna degradación en la imagen de cara a una mayor compresión.

1.3.5.1 Tipos de formatos para imágenes digitales

- **TIFF (Tagged Image File Format):** es un formato muy flexible con o sin pérdida. Los detalles de este algoritmo se incluyen como parte del fichero. En la práctica, TIFF se usa casi exclusivamente como formato de almacenamiento de imágenes sin pérdidas y sin ninguna compresión. Consecuentemente, los archivos en este formato suelen ser muy grandes.

- **PNG (Portable Network Graphics):** es un formato de almacenamiento sin pérdida y además tal compresión es totalmente reversible; por lo tanto, la imagen que se recupera es exacta a la original.
- **GIF:** crea una tabla de 256 colores a partir de una de 16 millones. Si la imagen tiene menos de 256 colores, puede almacenar la imagen sin pérdidas. Produce compresión de dos formas. Primero, reduce el número de colores de la imagen a 256 y por tanto, reduce el número de bits necesario por píxel. Después, reemplaza áreas de color uniforme usando código de secuencias.
- **JPG (Joint Photographic Experts Group Format):** permite obtener unos ratios de compresión muy altos manteniendo a su vez una calidad en la imagen muy elevada. JPG analiza las imágenes y elimina la información que no es apreciable. JPG almacena imágenes de 16 millones de colores. Permite también distintos niveles de compresión.
- **RAW:** es un formato no estandarizado y se refiere a la imagen de salida que ofrece algunas cámaras digitales. Aunque es un método sin pérdida, ofrece un factor de tres o cuatro menor que el formato TIFF de la misma imagen.
- **BMP (Bitmap):** BMP es un formato de almacenamiento sin compresión de imágenes propiedad de Microsoft.
- **PSP / PSD:** son formatos usados en distintos programas básicos (Paint Shop Pro, Photoshop).
- **AIPD (National Instruments Internal Image File format) [3][4]:** Este formato es usado para grabar imágenes en punto flotante, complejas y HSL.

Formatos estándar en escala de grises a 8 bits son: BMP, TIFF, PNG, JPEG y AIPD. Formatos estándar en escala de grises a 16 bits son: PNG y AIPD. Formatos estándar en color para imágenes RGB son: BMP, TIFF, PNG, JPEG y AIPD. Formatos estándar de imágenes complejas son: PNG y AIPD.

En la siguiente Tabla se muestra una comparación de tamaños entre distintos tipos de archivos de una misma imagen.

Tipo de archivo	Tamaño
TIFF sin ningún tipo de compresión	901 K
TIFF con compresión LZW	928 K
JPG	105 K
PNG compresión sin perdida	741 K
GIF compresión sin perdida (256 colores)	131 K

Tabla 1.2 Comparación de tamaños en archivos

1.4 ANÁLISIS Y PROCESAMIENTO DIGITAL DE IMÁGENES

1.4.1 ANALISIS DE IMÁGENES [3]

El análisis de imágenes combina técnicas que procesan las estadísticas y las medidas basadas en las intensidades de niveles de gris de los píxeles de la imagen. Las funciones en el análisis de imágenes son usadas para comprender el contenido de la imagen y así poder decidir el tipo de herramientas de inspección a usar para resolver una determinada aplicación. Ciertas funciones en el análisis de imágenes son las siguientes: histograma, perfiles de línea y medición de intensidades.

El histograma es una herramienta fundamental en el análisis de imágenes, ya que representa una descripción general de la apariencia de una imagen y ayuda a identificar los varios componentes como el fondo, los objetos y el ruido. Realiza conteo y graficación del número total de píxeles para cada nivel de gris. El histograma es una función H , definida en el rango de escala de grises $[0, \dots, k, \dots, 255]$, como el número de píxeles igual al valor de nivel de gris k .

$$H(k) = n_k$$

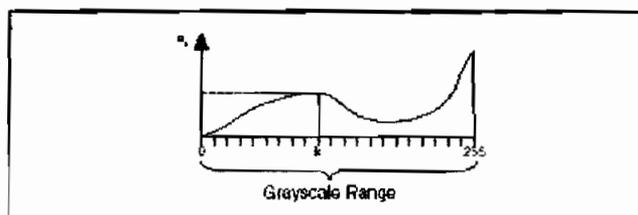


Figura 1.11 Gráfica de un histograma

1.4.2 PROCESAMIENTO DE IMÁGENES [3][5]

Procesamiento de imágenes es el término usado para denominar las operaciones desarrolladas sobre un conjunto de datos de imagen para mejorarlas de alguna forma, para ayudar a su interpretación o para extraer algún tipo de información útil de ella. Algunas de las operaciones principales en el procesamiento de imágenes son el escalamiento, codificación, extracción de características, reconocimiento de patrones, entre otras. Las mismas que se expresan generalmente en forma algorítmica y es implementado en software, aunque algunas ocasiones es necesario utilizar un hardware especializado, debido a la necesidad de velocidad en algunas aplicaciones.

En el tratamiento de imágenes hay que distinguir dos tipos de información a procesar:

- **Imágenes.** Se dice que la información es del tipo *imagen* cuando los datos que componen el paquete de información pueden organizarse en una matriz espacial.
- **Datos.** Se dice que la información es del tipo *datos* cuando por ejemplo se citan las coordenadas de los segmentos que componen una imagen, el nivel promedio de gris en una imagen, o el histograma de una imagen.

El procesamiento de imagen puede clasificarse según el tipo de información de entrada y salida. De esta manera se ilustra los siguientes casos:

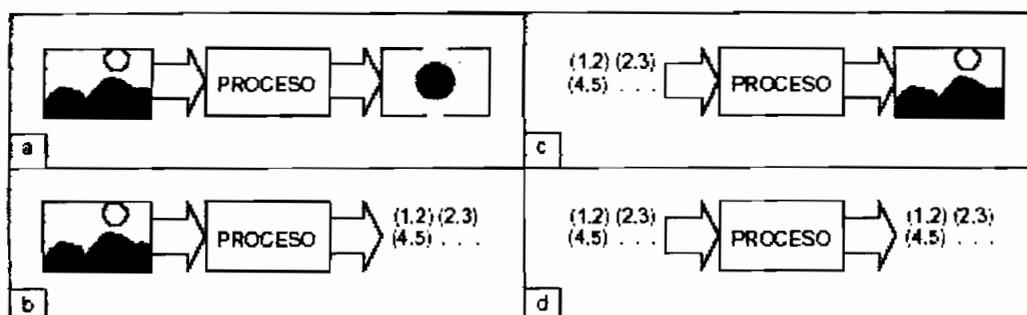


Figura 1.12 Clasificación de los algoritmos de procesamientos de imágenes: a) imagen-imagen, b) imagen-datos, c) datos-imágenes, d) datos-datos

Las operaciones que se realizan sobre imágenes y que tienen como resultado otra imagen se pueden clasificar en tres grupos:

- **Operaciones puntuales:** Son aquellas en que el valor de cada píxel $G[x,y]$ de la imagen resultante se obtiene a partir del valor del píxel $F[x,y]$ de la imagen original, sin involucrar ningún otro píxel.
- **Operaciones locales:** Son aquellas en las que el valor de cada píxel $G[x,y]$ de la imagen resultante se obtiene a partir de los valores del píxel $F[x,y]$ y de sus vecinos en la imagen original.
- **Operaciones globales:** Son aquellas en las que el valor de cada píxel $G[x,y]$ de la imagen resultante se obtiene de los valores del píxel $F[x,y]$ y de todos los demás píxeles en la imagen original.

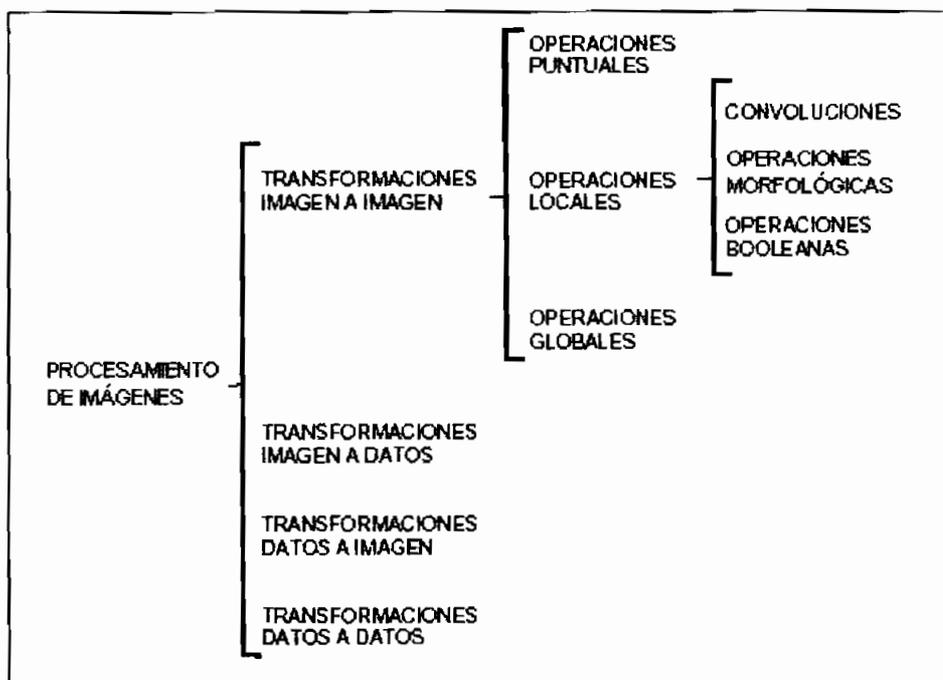


Figura 1.13 Tipos de procesamientos de imágenes

Para este proyecto se realizará un estudio mas profundo en lo que se refiere a los operadores aritméticos / lógicos, convolución, morfología aplicada a imágenes binarias dentro del análisis de partículas y al emparejamiento de modelo (pattern matching), ya que a partir de ellos se realizarán los distintos algoritmos de reconocimiento y conteo de vehículos.

1.4.2.1 Operadores aritméticos y lógicos

Son comúnmente utilizados para la identificación de la unión o intersección entre imágenes, corrección del fondo de la imagen para eliminar tendencias de luz, y comparaciones entre muchas imágenes y un modelo.

Una operación aritmética o lógica entre imágenes es una transformación píxel a píxel. Produce una imagen en que cada píxel deriva sus valores en base a los píxeles de la misma coordenada en las otras imágenes. Si A es una imagen con una resolución XY , B es una imagen con resolución XY y Op es el operador, entonces la imagen N resulta de la combinación de A y B a través del operador Op en la que cada píxel P de la imagen resultante N es asignado el valor $p_n = (p_a)(Op)(p_b)$. Donde p_a y p_b son los valores del píxel P en la imagen A y B , respectivamente. La siguiente Figura ilustra dicha combinación entre píxeles.

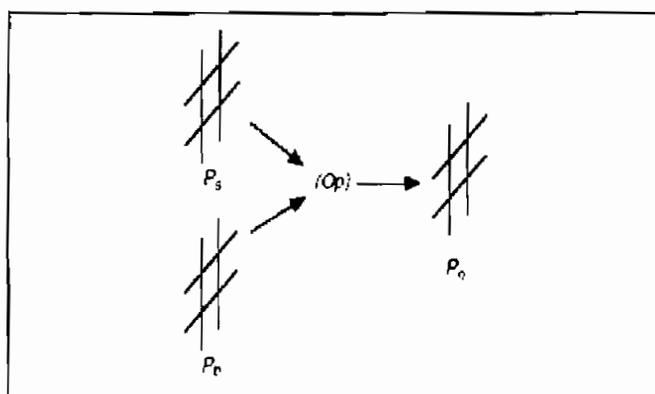


Figura 1.14 Operaciones entre píxeles

1.4.2.1.1 Operadores Aritméticos

Las Ecuaciones en la Tabla 1.3 describen el uso de operadores aritméticos para imágenes a y b de 8 bits.

Operador	Ecuacion
Multiplicacion	$p_n = \min(p_a \times p_b, 255)$
Division	$p_n = \max(p_a / p_b, 0)$
Suma	$p_n = \min(p_a + p_b, 255)$
Resta	$p_n = \max(p_a - p_b, 0)$
Modulo	$p_n = p_a \bmod p_b$
Diferencia absoluta	$p_n = p_a - p_b $

Tabla 1.3 Operadores Aritméticos

Si el valor del píxel resultante p_n es negativo, entonces éste es seteado a 0. Y si es mayor a 255, entonces $p_n = 255$.

1.4.2.1.2 Operadores Lógicos y de Comparación

Las Ecuaciones en la Tabla 1.4 describen el uso de operadores lógicos y de comparación.

Operador	Ecuacion
Operadores Lógicos	
AND	$p_n = p_a \text{ AND } p_b$
AND	$p_n = p_a \text{ NAND } p_b$
OR	$p_n = p_a \text{ OR } p_b$
OR	$p_n = p_a \text{ NOR } p_b$
XOR	$p_n = p_a \text{ XOR } p_b$
Diferencia Lógica	$p_n = p_a \text{ AND (NOT } p_b)$
Operadores Comparación	
Máscara	<i>if</i> $p_b = 0$, <i>then</i> $p_n = 0$, <i>else</i> $p_n = p_a$
Medía	$p_n = \text{mean}[p_a, p_b]$
Max	$p_n = \max[p_a, p_b]$
Min	$p_n = \min[p_a, p_b]$

Tabla 1.4 Operadores Lógicos y de Comparación

En el caso de imágenes a 8 bits, los operadores lógicos son principalmente designados para combinar imágenes a niveles de gris con imágenes de máscara (compuestas por valores de píxeles 0 y 255), éstos sirven también para combinar o comparar imágenes con imágenes binarias o etiquetadas.

La Tabla 1.5 ilustra como los operadores lógicos pueden ser usados para extraer o remover información en una imagen.

<i>Dado P_a</i>	<i>Si $P_b=255$, entonces</i>	<i>Si $P_b=0$, entonces</i>
AND	$P_a \text{ AND } 255 = P_a$	$P_a \text{ AND } 0 = 0$
NAND	$P_a \text{ NAND } 255 = \text{NOT } P_a$	$P_a \text{ NAND } 0 = 255$
OR	$P_a \text{ OR } 255 = 255$	$P_a \text{ OR } 0 = P_a$
NOR	$P_a \text{ NOR } 255 = 0$	$P_a \text{ NOR } 0 = \text{NOT } P_a$
XOR	$P_a \text{ XOR } 255 = \text{NOT } P_a$	$P_a \text{ XOR } 0 = P_a$
Logic Difference	$P_a - \text{NOT } 255 = P_a$	$P_a - \text{NOT } 0 = 0$

Tabla 1.5 Usando Operadores Lógicos con imágenes binarias de máscara

1.4.2.2 Convolución Kernel

Una convolución kernel define un filtro en dos dimensiones que se puede aplicar a una imagen en escala de grises (ng); por tanto, se trata de una estructura 2D cuyos coeficientes definen las características del filtro de la convolución que lo representa.

En una operación típica de filtrado, los coeficientes de la convolución kernel determinan los valores de filtrado de cada píxel en la imagen. IMAQ Vision provee un set de convoluciones kernels que se puede usar para distintos tipos de operaciones de filtrado en una imagen. Filtrando una imagen se puede aumentar la calidad de la misma o para cumplir ciertos requerimientos dependiendo de la aplicación; por tanto, se usa filtros para aplanar imágenes, reducir o remover el ruido de una imagen, incrementar la información del borde en una imagen, etc.

El valor de filtrado de un píxel es una combinación ponderada de sus valores originales y los valores de sus píxeles vecinos. El tamaño de la convolución Kernel determina el número de píxeles vecinos cuyos valores son considerados durante el proceso de filtrado, en el caso de un kernel 3x3, el valor del píxel a procesar es derivado de los valores de sus ocho vecinos que le rodean, un kernel 5x5 especifica 24 vecinos y así sucesivamente.

La operación de filtrado involucra a todos y cada uno de los píxeles que componen la imagen, como se indica en la Figura 1.15.

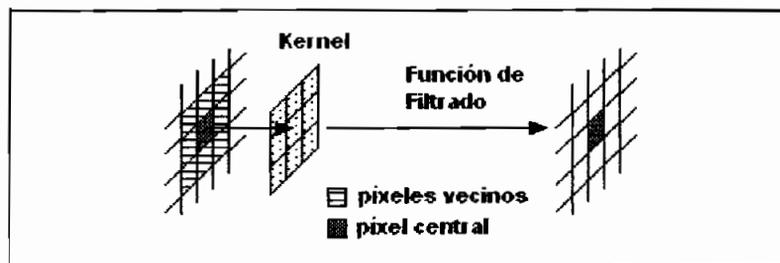


Figura 1.15 Mecanismo de filtrado

Cuando se procesa los píxeles a lo largo del borde de la imagen, parte de la estructura kernel cae fuera de la imagen. Por ejemplo en la Figura 1.16, la primera fila y la primera columna del kernel 3x3 están fuera de la imagen cuando se está procesando el valor del píxel de coordenadas (0,0).

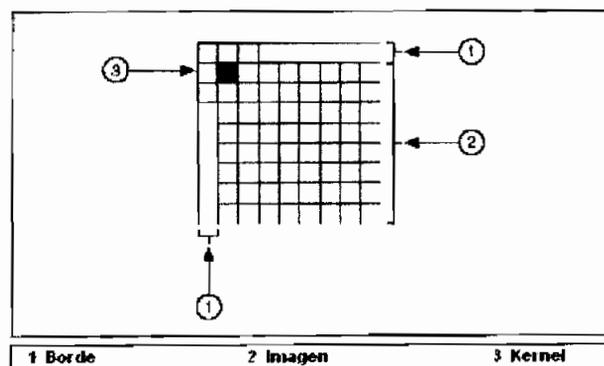


Figura 1.16 Filtrando píxeles del borde

IMAQ Vision automáticamente localiza una región de borde en el momento de crear una imagen, el valor por defecto es de tres píxeles y sus píxeles contienen

el valor de 0, también se puede configurar el tamaño del borde; en cuyo caso, el tamaño de ésta debe ser mayor o igual que la mitad del número de filas o columnas del kernel que se requiere para el proceso de filtrado.

1.4.2.3 Filtrado Espacial

Los filtros espaciales sirven para una variedad de propósitos, como la detección de bordes a lo largo de una dirección específica, contornos de modelos, reducción de ruido, resaltar detalles y aplanar; de esta manera, poder extraer la información que se necesite.

Los filtros son divididos en dos tipos: lineales (llamados también de convolución) y no-lineales.

Convolución es un algoritmo que consiste en la recalculation de los valores de los píxeles, basándose en su propio valor y de los valores ponderados de sus vecinos de acuerdo a los coeficientes de una convolución kernel. La suma de éste cálculo es dividido para la suma de los elementos en el kernel y así obtener un nuevo valor de píxel. El tamaño de una convolución kernel no tiene límite teórico y puede ser cuadrado o rectangular (3x3, 5x5, 5x7, 9x3, etc.)

El propósito de los filtros no-lineales es el de poder extraer contornos (detección de bordes) o el de remover los píxeles aislados.

Por todo lo mencionado anteriormente, se puede decir que los filtros espaciales alteran los valores de los píxeles con respecto a variaciones en intensidad de luz en su "vecindario", el cual es definido por el tamaño de una matriz, o máscara, centrada en el píxel a procesar. Esos filtros pueden ser sensitivos a la presencia o ausencia de variaciones en la intensidad de luz.

Los filtros espaciales pueden ser clasificados en dos categorías:

- Filtros pasa – altos, enfatizan en variaciones significativas de la intensidad de luz usualmente encontrados en los límites de los objetos.

- Filtros pasa – bajos, atenúan variaciones de la intensidad de luz, tiene la tendencia de aplanar las imágenes, eliminando detalles y manchando los bordes.

La Tabla 1.6 describe los diferentes tipos de filtros espaciales.

Tipo	Filtros
Lineal	
Pasa – altos	Gradiente, Laplaciano
Pasa – bajos	Aplanamiento, Gausiano
No – Lineal	
Pasa – altos	Gradiente, Roberts, Sobel, Prewitt, Diferenciación, Sigma
Pasa – bajos	Mediana, Orden N, Pasa bajo

Tabla 1.6 Clasificación de los filtros espaciales

Para la presente aplicación de este proyecto se utilizará uno o varios de los filtros lineales con el objetivo de mejorar las imágenes que serán sometidas al procedimiento de reconocimiento vehicular, por tanto se explicará con más detalle los filtros lineales.

1.4.2.3.1 Filtros Lineales

Un filtro lineal reemplaza cada píxel por la suma ponderada de sus vecinos. La matriz que define el “vecindario” del píxel es llamada convolución kernel y también especifica el valor (peso) asignado a cada vecino.

Si el filtro kernel contiene coeficientes negativos y positivos, la función de transferencia es equivalente a una ponderación diferenciada y produce un afilamiento de la imagen o filtro pasa – alto, como en el caso de los filtros gradiente y laplaciano. Si todos los coeficientes en el kernel son positivos, la función de transferencia es equivalente a una suma ponderada y produce un aplanamiento de la imagen o filtro pasa – bajo, como en el caso de los filtros de aplanamiento y gausiano.

a) Filtro Gradiente

Resalta las variaciones de intensidad de luz a lo largo de una dirección específica y tiene el efecto de perfilar los bordes y realzar la textura. Un filtro de convolución tipo gradiente es una matriz derivativa de primer orden, su kernel usa el siguiente modelo:

$$\begin{matrix} a & -b & c \\ b & x & -d \\ c & d & -a \end{matrix}$$

Donde a , b , c , d son enteros y x puede tomar los valores de 0 ó 1. Dicha matriz tiene un eje de simetría que corre entre los coeficientes positivos y negativos del kernel y a través del elemento central, dicho eje da la orientación de los bordes a perfilar, como se observa en la siguiente Figura.

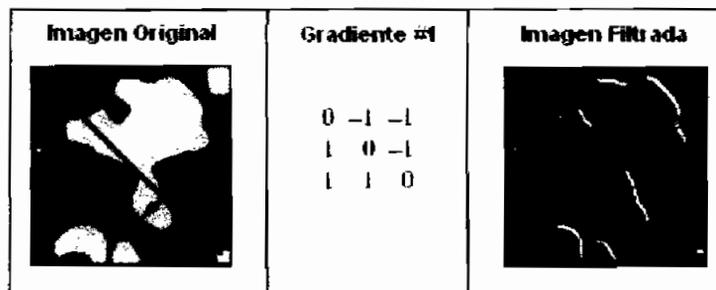


Figura 1.17 Acción del filtro gradiente con el kernel indicado

Si $x=1$, el filtro de convolución usa un segundo kernel al lado derecho de la siguiente ecuación, el cual reproduce la imagen original. El resultado total se muestra en la Figura 1.18.

$$\begin{matrix} -1 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{matrix} = \begin{matrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} + \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} \quad (\text{Ec. 1.5})$$

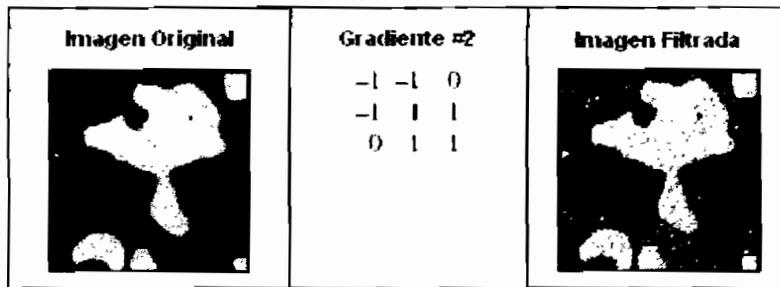


Figura 1.18 Acción del filtro gradiente con el kernel indicado

b) Filtro Laplaciano

Resalta las variaciones de intensidad de luz alrededor de un píxel, por tanto extrae el contorno de los objetos y perfila los detalles, a diferencia del anterior filtro, el filtro laplaciano se aplica en todas las direcciones. Un filtro de convolución tipo laplaciano es una matriz derivativa de segundo orden y su kernel usa el siguiente modelo:

$$\begin{matrix} a & d & c \\ b & x & b \\ c & d & a \end{matrix}$$

Donde a, b, c, d son enteros y el valor de x puede ser igual o mayor que la suma de los valores absolutos de los demás coeficientes; por lo tanto, este filtro tiene dos efectos diferentes, como se indica en la Figura 1.19.

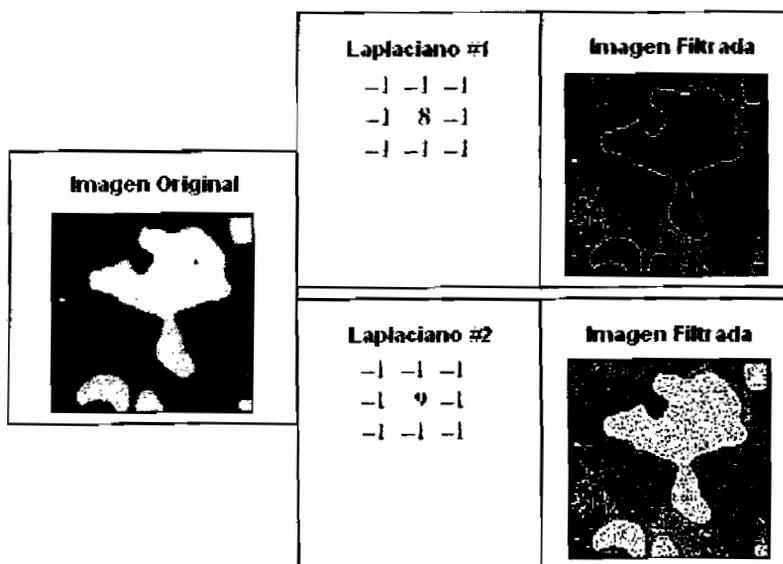


Figura 1.19 Efectos del filtro laplaciano

Mientras más aumenta el tamaño de la matriz de convolución kernel, el contorno de la imagen también aumentará.

c) Filtro de aplanamiento (smoothing)

Atenúa las variaciones de intensidad de luz en el vecindario de un píxel; es decir, aplanar completamente la forma de los objetos, mancha o empaña los bordes y también remueve los detalles.

Un filtro de convolución tipo smoothing es un filtro promediado cuyo kernel usa el siguiente modelo:

$$\begin{array}{ccc} a & d & c \\ b & x & b \\ c & d & a \end{array}$$

Donde a, b, c, d son enteros y el valor de x puede ser 0 ó 1, cuando $x=0$ se tiene un empañamiento más fuerte que cuando $x=1$, tal como se ilustra en la siguiente Figura.

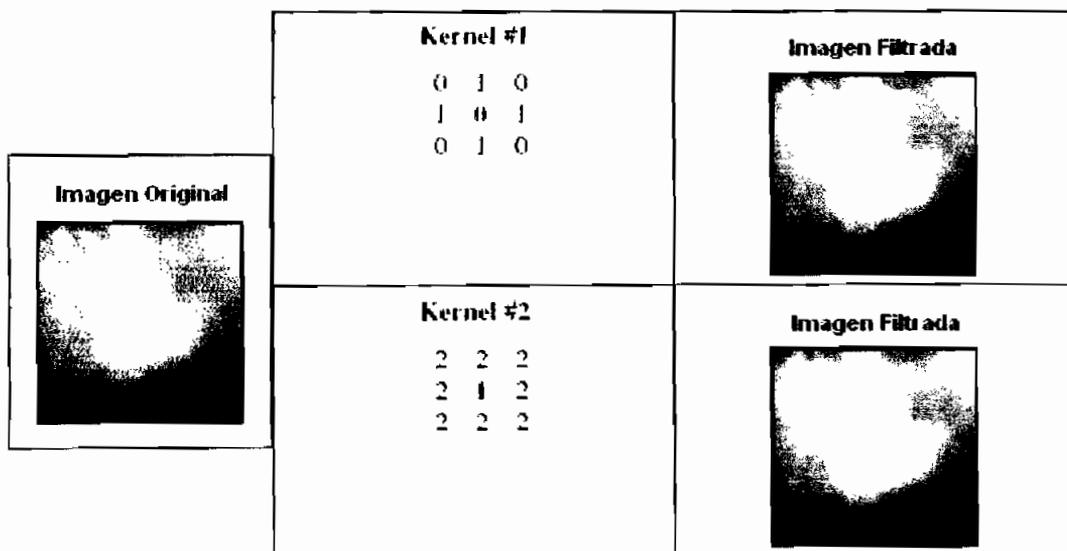


Figura 1.20 Ejemplos de acción para el filtro de aplanamiento (smoothing)

d) Filtro Gaussiano

Atenúa las variaciones de intensidad de luz en el vecindario de un píxel. Aplana completamente la forma de los objetos y atenúa detalles. Tiene un efecto de empañamiento más suave que el filtro anterior. Un filtro de convolución tipo gaussiano es un filtro promediado cuyo kernel usa el siguiente modelo:

$$\begin{array}{ccc} a & d & c \\ b & x & b \\ c & d & a \end{array}$$

Donde a , b , c , d son enteros y el valor de x es mayor que 1. Para un tamaño dado los coeficientes son las mejores aproximaciones posibles usando números enteros de una curva gaussiana. Su efecto se ilustra en la siguiente Figura.

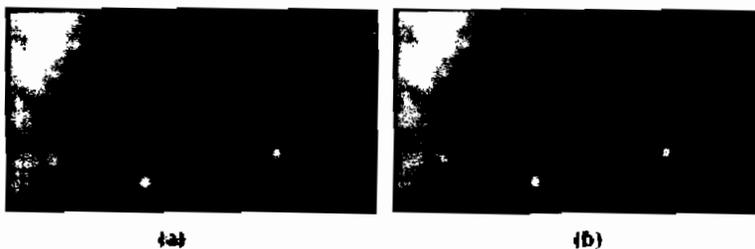


Figura 1.21 Acción del filtro gaussiano (a) imagen original, (b) imagen filtrada

1.4.2.3.2 Filtros No – Lineales

Este tipo de filtros reemplazan cada valor del píxel con una función no – lineal de los píxeles que lo rodean. Al igual que los anteriores filtros, los filtros no – lineales operan sobre el vecindario del píxel en cuestión.

1.4.2.4 Análisis de partículas (blob) [3]

Se trata de un proceso de escaneo en la imagen entera para detectar todas las partículas (blobs – binary large objects) y construir un reporte detallado de cada partícula. Este reporte consiste aproximadamente de 50 piezas de información, tal como: localización de la partícula en la imagen, tamaño, forma, orientación con respecto a otras partículas, segmento más largo, momento de inercia, área, etc.

Sirve por tanto para detectar regiones o grupos de píxeles conectados en una imagen para entonces hacer determinadas operaciones en dichas regiones; de esta manera, se puede detectar y analizar cualquier forma de 2 dimensiones en una imagen.

El proceso a seguir en el análisis de partículas es el siguiente:

- Umbralización (Thresholding)
- Morfología Binaria
- Medición de partículas

1.4.2.4.1 Umbralización (Thresholding)

Este proceso es muy usado para aislar objetos de interés en una imagen desde el fondo, eligiendo un intervalo adecuado entre 0 a 255 para una imagen en niveles de gris, y de ésta manera focalizar el análisis hacia esos objetos.

Consiste en la segmentación de una imagen en dos regiones: una región de partícula y una región de fondo; por lo tanto, crea imágenes con partículas binarias. Poniendo a 1 (objeto blanco) a todos los píxeles que pertenecen a un intervalo de nivel de gris (intervalo de thresholding) y a 0 el resto de píxeles (fondo negro).

Para escoger el intervalo de thresholding se podría ayudar del perfil de línea, así como también del histograma de una imagen. En la Figura 1.22 se observa un ejemplo de aquello, el histograma de la imagen original presenta dos regiones una clara y una oscura; a partir de esto se escoge un intervalo entre 166 y 255 con el fin de realzar las regiones claras de la imagen. Obteniendo de esta manera una imagen binaria.

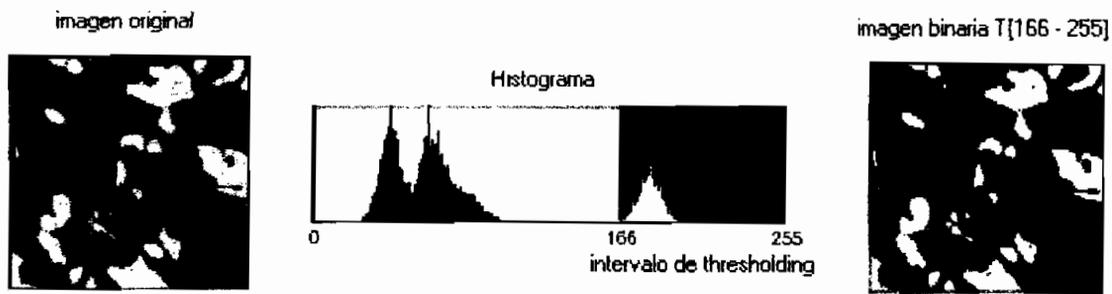


Figura 1.22 Proceso de thresholding con intervalo de [166-255]

A veces se requiere reforzar a la imagen (aplicando transformaciones LUT, filtros, FFT o ecualización [3] [2]) antes de realizar un thresholding para obtener una mejor forma de los objetos.

Para imágenes a color es necesario definir 3 intervalos de thresholding, uno para cada componente de color obteniéndose así también una imagen binaria.

1.4.2.4.2 Morfología Binaria

Las operaciones morfológicas binarias extraen y alteran la estructura de las partículas en imágenes binarias. Se pueden usar estas operaciones durante una aplicación de inspección para mejorar la información en una imagen binaria, antes de empezar con mediciones de partículas.

Antes de revisar las distintas operaciones morfológicas, es necesario aclarar ideas acerca de los elementos estructurales así como también de la conectividad.

a) Elementos Estructurales

Los elementos estructurales (structuring elements) son máscaras binarias 2D usadas por las transformaciones morfológicas para definir el tamaño y efecto del vecindario sobre cada píxel, controlando el efecto de las funciones morfológicas primarias en la forma y el límite de las partículas.

El tamaño de un elemento estructural determina el tamaño del vecindario que rodea al píxel que está siendo procesado, cuyas coordenadas son determinadas

como una función del elemento estructural. Dicho tamaño también influye en la velocidad de las transformaciones morfológicas. Al usar elementos estructurales requiere un borde de imagen, por ejemplo para un elemento de 3x3, la imagen requiere un tamaño mínimo de borde de 1.

Los valores binarios (1/0) de un elemento estructural determinan que píxeles del vecindario se consideran durante una transformación, de la siguiente manera:

- Si el valor de un sector del elemento estructural es 1, el valor del correspondiente píxel de la imagen original afecta al valor del píxel central durante una transformación.
- Si el valor de un sector del elemento estructural es 0, la operación morfológica descarta el valor del correspondiente píxel de la imagen original.

En la siguiente Figura se observa que una transformación morfológica usa un elemento estructural para alterar un píxel P_0 que viene como una función de sus píxeles vecinos.

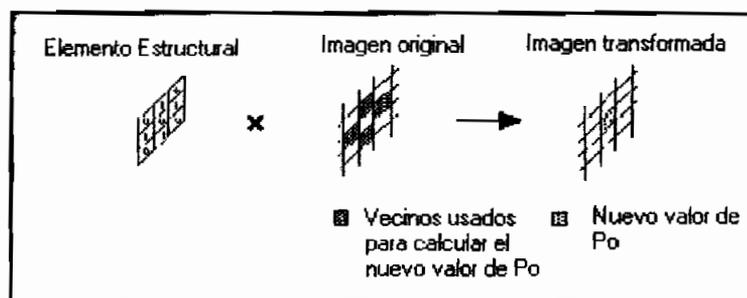


Figura 1.23 Efecto de los valores del elemento estructural

Las transformaciones morfológicas que extraen y alteran la estructura de las partículas permiten procesar píxeles en una configuración cuadrada o hexagonal. Dichas configuraciones introducen el concepto de forma de cuadro del píxel (pixel frame) como se indica en la Figura 1.24, y éste es un parámetro importante para funciones que alteran el valor de los píxeles de acuerdo a los valores de intensidades de sus vecinos.

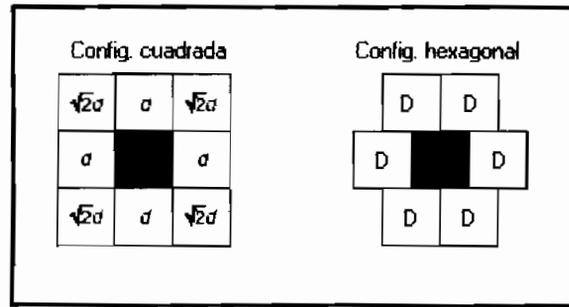


Figura 1.24 Configuraciones de cuadro del píxel

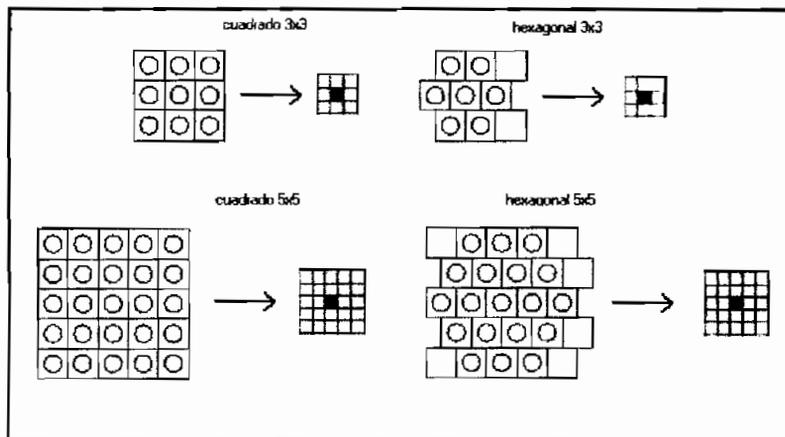


Figura 1.25 Vecindario de píxeles basados en formas de cuadros de píxel

La Figura anterior ilustra la diferencia de configuraciones de cuadros de píxel cuando elementos estructurales de 3x3 y de 5x5 son aplicados, así como también el vecindario que implica en cada una de las configuraciones y tamaños.

b) Conectividad

El proceso de umbralización usado para obtener imágenes binarias, introduce el concepto de píxeles adyacentes o conectividad. En algunas funciones se puede especificar la conectividad de píxel, para que el software determine si dos píxeles son incluidos en la misma partícula. Existen dos tipos de conectividad:

- Conectividad - 4: usado para imágenes binarias que contienen partículas que topan en un punto; según esto, dos píxeles son considerados de la misma partícula si ellos son adyacentes horizontalmente o verticalmente.

- **Conectividad - 8:** usado en imagen que contienen partículas con áreas angostas; según esto, dos píxeles son considerados de la misma partícula si ellos son adyacentes horizontalmente, verticalmente o diagonalmente.

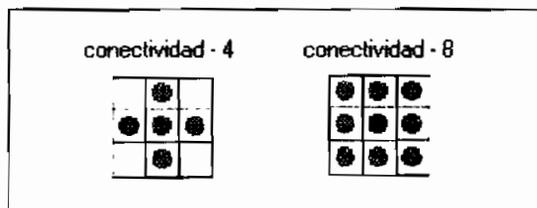


Figura 1.26 Tipos de conectividad

1.4.2.4.3 Operaciones morfológicas primarias

Estas operaciones son utilizadas para expandir o reducir partículas, aplanar los bordes de los objetos, encontrar límites externos e internos de las partículas y localizar configuraciones particulares de píxeles. Las funciones fundamentales de procesamiento binario son: Erosión, Dilatación y Ganancia o Pérdida (Hit miss). Las otras transformaciones son combinaciones de estas funciones y son las siguientes:

- Apertura (Opening)
- Cierre (Closing)
- Gradiente interior (Inner gradient)
- Gradiente exterior (Outer gradient)
- Adelgazamiento (Thinning)
- Engrosamiento (Thickening)
- Apertura apropiada (Proper-opening)
- Cierre apropiado (Proper-closing)
- Mediana automática (Auto-median)

a) Funciones Erosión y Dilatación

La **Erosión** es una función que elimina píxeles aislados en el fondo y erosiona el contorno de partículas de acuerdo a la plantilla definida por el elemento estructural.

Para un píxel dado P_o de la imagen original (igual a 1), el elemento estructural está centrado sobre P_o . Los píxeles enmascarados por un coeficiente del elemento estructural igual a 1, son entonces referidos como P_i .

- Si el valor de un píxel de P_i es igual a 0, entonces P_o es puesto a 0, sino P_o es puesto a 1.
- Función equivalente a una compuerta AND.

La **Dilatación** es una función que elimina agujeros pequeños aislados y expande el contorno de las partículas de acuerdo a la plantilla definido por el elemento estructural. Esta función es equivalente a erosionar el fondo.

- Si el valor de un píxel de P_i es igual a 1, entonces P_o es puesto a 1, sino P_o es puesto a 0.
- Función equivalente a una compuerta OR.

En la siguiente Figura se ilustra el efecto de la erosión y la dilatación. La Figura 1.27a muestra la imagen binaria original. La Figura 1.27b muestra la imagen binaria luego de una erosión. La Figura 1.27c muestra la imagen luego de una dilatación.

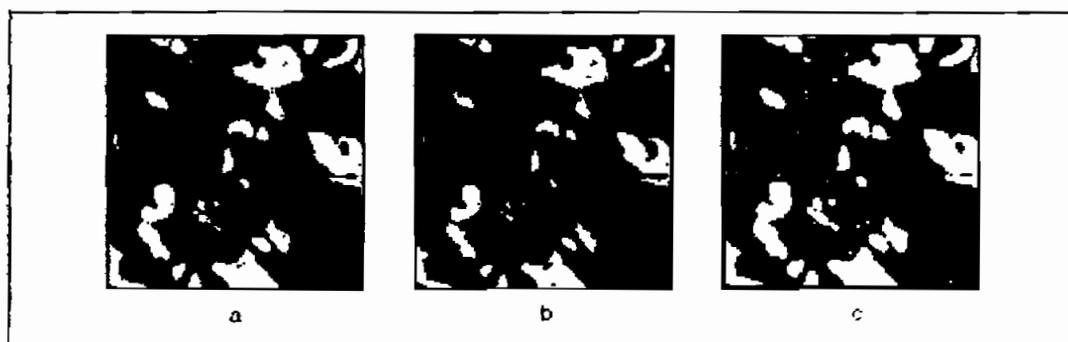


Figura 1.27 Efectos de las funciones de Erosión y Dilatación

A continuación se muestra un ejemplo de aplicación de estas funciones, la Figura 1.28 ilustra la imagen original para las imágenes mostradas en la Tabla 1.7, en donde los cuadros grises indican píxeles igual a 1.

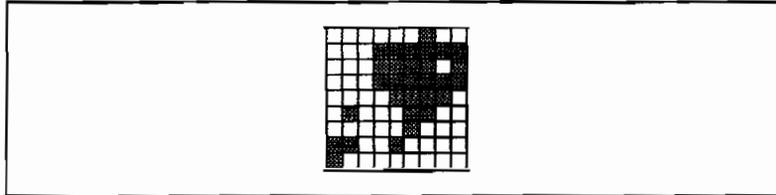


Figura 1.28 Imagen original antes de la erosión y la dilatación

La siguiente Tabla muestra como el elemento estructural puede controlar el efecto de la erosión y dilatación.

	Elem. Estructural		Descripción
EROSIÓN			Un píxel es puesto a 0, si su valor original es 1 y si sus tres vecinos no son igual a 1.
			Un píxel es puesto a 0, si su valor original es 1 y si sus dos vecinos no son igual a 1.
DILATACIÓN			Un píxel es puesto a 1, si su valor original es 1 o si uno de sus tres vecinos valen 1.
			Un píxel es puesto a 1, si su valor original es 1 o si uno de sus dos vecinos valen 1.

Tabla 1.7 Efecto del elemento estructural a la erosión y a la dilatación

b) Función Ganancia o Pérdida (Hit – miss)

Esta función es usada para localizar configuraciones particulares de píxeles (herramienta básica para la detección de formas). Extrae cada píxel localizado en

un vecindario exactamente emparejado a la plantilla definido por el elemento estructural.

Puede localizar píxeles aislados simples, modelos longitudinales, ángulos correctos a lo largo del borde de las partículas y otras formas especificadas por el usuario. En un elemento estructural con un coeficiente central igual a 0, ésta función cambian todos los píxeles puestos a 1 en la imagen original al valor de 0.

Para un píxel dado P_o de la imagen original (igual a 1), el elemento estructural está centrado sobre P_o . Los píxeles enmascarados por el elemento estructural, son entonces referidos como P_i .

- Si el valor del píxel de P_i es igual al coeficiente del elemento estructural localizado encima de él, entonces P_o es puesto a 1, sino P_o es puesto a 0.
- En otras palabras, si los píxeles P_i definen la misma plantilla como el elemento estructural, entonces $P_o = 1$, sino $P_o = 0$.

En las Figuras 1.29b, 1.29c, 1.29d, 1.29e se muestra el resultado de cuatro funciones Ganancia o Pérdida aplicados a la misma imagen original representado en la Figura 1.29a. Cada función usa un diferente elemento estructural, que es especificado arriba de cada imagen transformada.

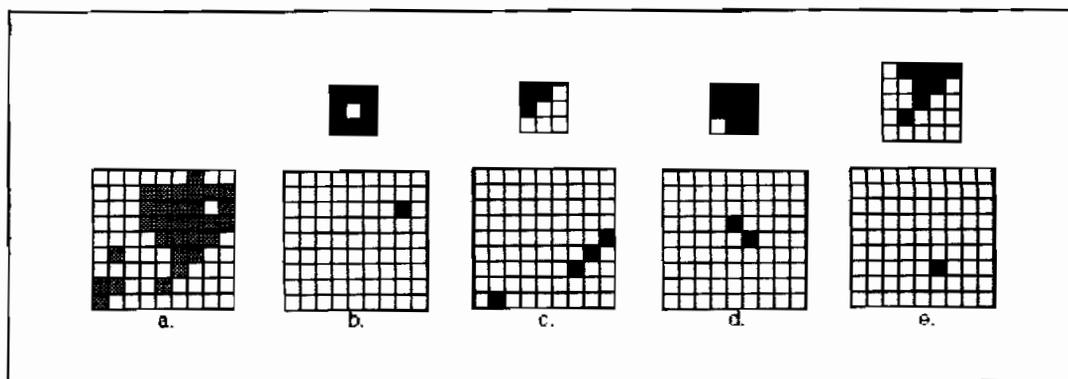


Figura 1.29 Función Ganancia o Pérdida (Hit – miss)

c) *Función Mediana automática (Auto-Median)*

Esta función usa combinaciones duales de aperturas y cierres. Por lo tanto puede generar partículas más simples que contengan muy pocos detalles.

Si I es la imagen original, ésta función extrae la propiedad de entre las funciones Apertura apropiada (Proper-opening) y Cierre apropiada (proper-closing), se tiene que:

$$\begin{aligned} \text{auto-median}(I) &= \text{AND}(\text{DEEDDE}(I), \text{EDDEED}(I)) \text{ ó} \\ \text{auto-median}(I) &= \text{AND}(\text{OCO}(I), \text{COC}(I)) \end{aligned}$$

Donde:

- E es una erosión
- D es una dilatación
- O es una apertura
- C es un cierre

1.4.2.4.4 *Operaciones morfológicas avanzadas*

Estas operaciones son combinaciones condicionales de los operadores morfológicos primarios y trabaja sobre las partículas en lugar de los píxeles en la imagen.

Son usadas para llenar agujeros en las partículas, remover partículas que tocan el borde de la imagen, remover partículas indeseadas ya sean pequeñas o grandes, separar partículas que se tocan, encontrar la forma convexa de las partículas, preparar partículas para análisis cuantitativo, observar la geometría de las regiones, etc. Las funciones morfológicas avanzadas son las siguientes:

- Borde (Border)
- Llenado de agujeros (Hole filling)
- Etiquetado (Labeling)
- Filtros pasa – bajos
- Filtros pasa – altos

- Separación (Separation)
- Esqueleto (Skeleton)
- Segmentación (Segmentation)
- Distancia (Distance)
- Danielsson
- Círculo (Circle)
- Convexo (Convex)

A continuación se explicarán las operaciones más importantes que forman parte del procesamiento mismo para la aplicación de éste proyecto.

a) Borde

Esta función elimina partículas que tocan el borde de la imagen. Esas imágenes pueden haber sido truncadas durante la digitalización de la imagen y eliminándolas y así ayudar a evitar estadísticas y mediciones erróneas de partículas.

b) Llenado de agujeros

Esta función llena los agujeros presentes en las imágenes con partículas.

c) Etiquetado

Asigna un valor diferente de nivel de gris a cada partícula. La imagen producida no es binaria sino etiquetada, la cual usa un número de niveles de gris igual al número de partículas en la imagen más el nivel de 0 usado en el área de fondo. Esta función identifica partículas usando el criterio de conectividad.

d) Separación

Separa partículas que se topan, particularmente pequeños istmos encontrados en ciertas partículas con respecto al tamaño del elemento estructural definido por el

usuario. Esta operación usa las funciones de erosión, etiquetado (labeling) y dilatación.

e) Círculo

Permite separar partículas circulares sobrepuestas. Esta función usa el coeficiente de Danielsson para reconstruir la forma de una partícula, previendo que dichas partículas son esencialmente circulares. Las partículas son tratadas como un juego de discos sobrepuestos pero en su análisis son considerados como discos separados. De esta forma, ésta función permite trazar círculos correspondientes a cada partícula.

La Figura 1.30a ilustra la imagen original para el ejemplo de aplicación de ésta función. La Figura 1.30b muestra la imagen procesada.

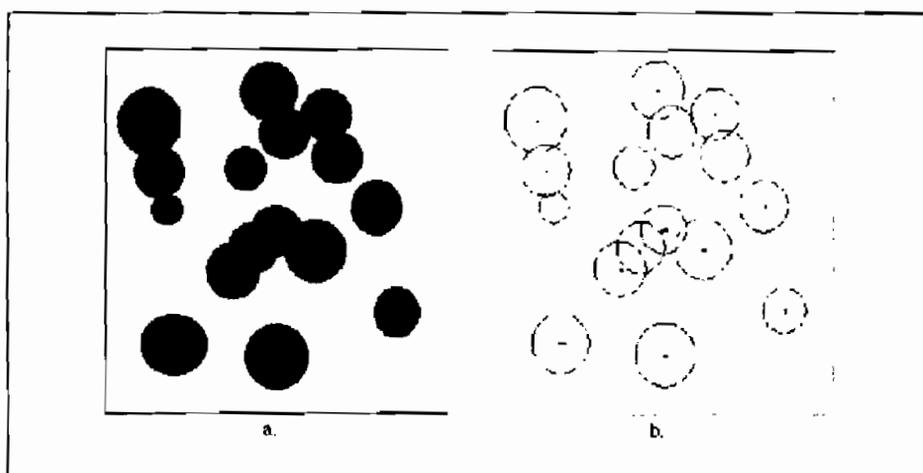


Figura 1.30 Función Círculo

f) Convexo

Es usada para cerrar el contorno de las partículas y así poder realizar mediciones sobre las mismas, siempre y cuando su contorno sea discontinuo. Esta función es usualmente necesaria cuando la partícula de la muestra se encuentra cortada debido al proceso de adquisición. La imagen a ser tratada debe ser binaria o etiquetada. La Figura 1.31a representa la imagen binaria original usada en este

ejemplo. La Figura 1.31b muestra el resultado de la función convexo aplicada a la imagen.

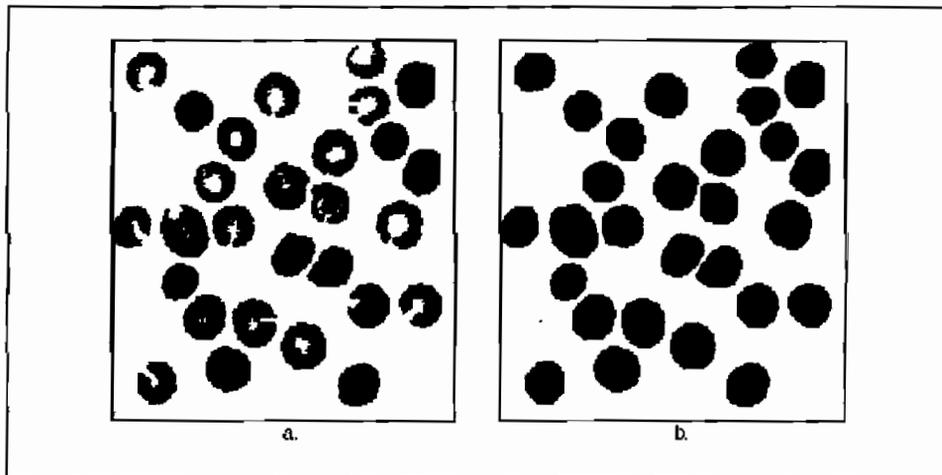


Figura 1.31 Función Convexo

1.4.2.4.5 Medición de Partículas

Permite realizar caracterizaciones en la forma de las partículas digitales en base a un juego de parámetros morfológicos y de intensidad realizadas sobre partículas en imágenes binarias, como: área, longitud, cuerdas y ejes, coordenadas, equivalencias y características de forma.

a) Área

Permite discernir los siguientes parámetros:

- **Numero de píxeles:** es el área de una partícula (sin agujeros) en unidades de píxel.
- **Área de la partícula:** es el área de la partícula pero expresada en unidades reales, este valor es igual al número de píxeles cuando 1 píxel representa 1 unidad cuadrada.
- **Área Explorada:** área de la imagen entera expresada en unidades reales.
- **Razón:** es la relación entre el área de la partícula y el área escaneada, representa el porcentaje de la imagen ocupada por todas las partículas.

- **Área de agujeros:** es el área total de los agujeros dentro de una partícula.
- **Área total:** es la suma del área de la partícula con el área de agujeros.

b) Longitud

A continuación se describe los parámetros relacionados a las longitudes de las partículas:

- **Perímetro de la partícula:** corresponde a la longitud externa del contorno de una partícula.
- **Perímetro de agujeros:** es la suma de los perímetros de los agujeros dentro de una partícula.
- **Ancho:** es la distancia entre los píxeles máximo y mínimo de una partícula en el sentido horizontal. También es igual al segmento horizontal del rectángulo más pequeño que puede contener a la partícula.
- **Alto:** es la distancia entre los píxeles máximo y mínimo de una partícula en el sentido vertical. También es igual al segmento vertical del rectángulo más pequeño que puede contener a la partícula.

La siguiente Figura ilustra algunos de los parámetros de longitud en una partícula.

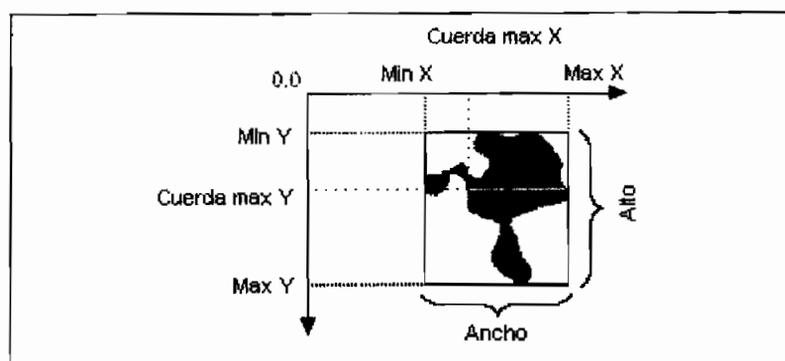


Figura 1.32 Longitudes en una partícula

1.4.2.5 Emparejamiento de modelo (Pattern Matching)

Esta función localiza regiones de una imagen en escala de grises que empata con una plantilla predeterminada. Encuentra plantillas de emparejamiento a pesar de que exista poca iluminación, ruido, emborronamiento, cambio o rotaciones de plantilla.

Permite crear un modelo o plantilla que representa el objeto de búsqueda para cualquier aplicación, calculando además un puntaje (score) para cada empate. Por tanto dicho puntaje se refiere a cuan cercano puede ser un determinado emparejamiento con respecto a la plantilla original.

Los algoritmos de emparejamiento son algunas de las funciones más importantes en procesamientos de imágenes porque son usados en muy variadas aplicaciones, como en las siguientes:

- **Alineación:** determina la posición y la orientación de un objeto conocido con la ayuda de algunos puntos de referencia de dicho objeto.
- **Calibración:** mide longitudes, diámetros, ángulos y otras dimensiones críticas. Por ejemplo si dichas mediciones están fuera de los límites de tolerancia, el componente o la pieza bajo análisis son rechazados.
- **Inspección:** detecta fallas comunes, como la pérdida de una pieza.

En procesos automatizados, la apariencia visual de los materiales o componentes bajo inspección pueden cambiar debido a factores como la orientación de la pieza, cambios de escala y cambios de luz. Las herramientas del emparejamiento de modelo tiene la capacidad de localizar los modelos de referencia a pesar de que existan cambios.

Orientación de Modelo y Múltiples Instancias

Cuando un modelo es rotado o escalado en la imagen, las herramientas del emparejamiento de modelo pueden localizar lo siguiente:

- El modelo en la imagen
- La posición del modelo en la imagen
- La orientación del modelo
- Instancias múltiples del modelo en la imagen.

La Figura 1.33a muestra una imagen de plantilla o modelo y el resto de Figuras muestran imágenes en las cuales el modelo de búsqueda es encontrado a pesar de los cambios mencionados.

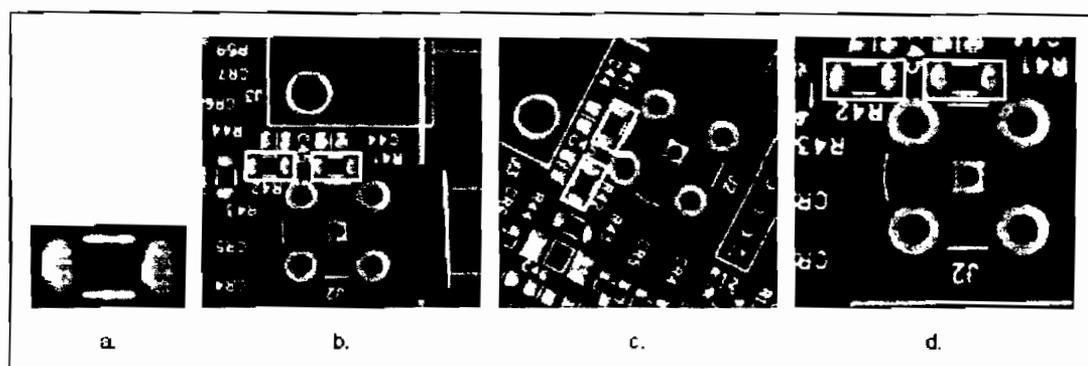


Figura 1.33 Orientación de modelo y múltiples instancias

Condiciones Ambientales de luz

Las herramientas del emparejamiento de modelo pueden encontrar el modelo de referencia en una imagen bajo condiciones de cambios uniformes de iluminación a lo largo de la imagen. La siguiente Figura ilustra las condiciones típicas en las cuales las herramientas del emparejamiento de modelo funcionan correctamente.

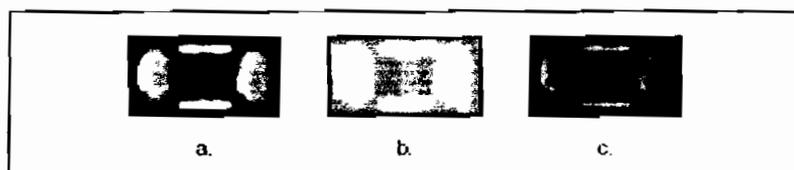


Figura 1.34 Ejemplos de condiciones de iluminación

Condiciones de Emborronamiento y Ruido

El emparejamiento de modelo puede encontrar modelos que han sufrido alguna transformación debido al ruido o al emborronamiento, el cual ocurre debido a un enfoque incorrecto o cuando cambia la profundidad de campo.

La siguiente Figura ilustra las condiciones típicas bajo las cuales las herramientas del emparejamiento de modelo funcionan correctamente. La Figura 1.35a muestra la imagen del modelo original. La Figura 1.35b muestra cambios en la imagen debido al emborronamiento. La Figura 1.35c muestra cambios en la imagen debido al ruido.

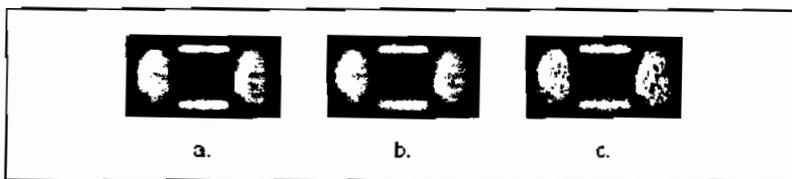


Figura 1.35 Ejemplos de emborronamiento y ruido

1.4.2.5.1 Técnicas de Emparejamiento

a) Correlación Cruzada

La correlación cruzada normalizada es la forma más común de encontrar una plantilla en una imagen. Su mecanismo está basado en una serie de multiplicaciones, este proceso consume algo de tiempo. Nuevas tecnologías como el MMX permiten hacer multiplicaciones paralelas y por ende reducir el tiempo de procesamiento. Para incrementar la velocidad en el proceso de emparejamiento se debe reducir el tamaño de la imagen y restringir la región de búsqueda en la imagen. Sin embargo esta técnica no reúne los requerimientos de velocidad para muchas aplicaciones.

Considerando una sub-imagen $w(x,y)$ de tamaño (resolución) $K \times L$ y una imagen $f(x,y)$ de tamaño $M \times N$, donde $K \leq M$ y $L \leq N$. La correlación $C(i,j)$ entre $w(x,y)$ y $f(x,y)$ en el punto (i,j) está dada por:

$$C(i, j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x, y) f(x+i, y+j) \quad (\text{Ec. 1.6})$$

Donde $i = 0, 1, \dots, M-1$, $j = 0, 1, \dots, N-1$, y la suma es tomada como la región en la imagen donde w y f se traslapan.

La Figura 1.36 ilustra el procedimiento de correlación. El origen de la imagen f es la esquina superior izquierda.

Por lo tanto, la correlación es el proceso de mover la plantilla o sub-imagen w a través del área de la imagen f y computar el valor de C en esa área. Esto envuelve multiplicaciones de cada píxel de la plantilla con el píxel de la imagen que se traslapa y entonces sumando todos resultados. El máximo valor de C indica la posición donde w empata de mejor manera con f . Los valores de correlación no son exactos en los bordes de la imagen.

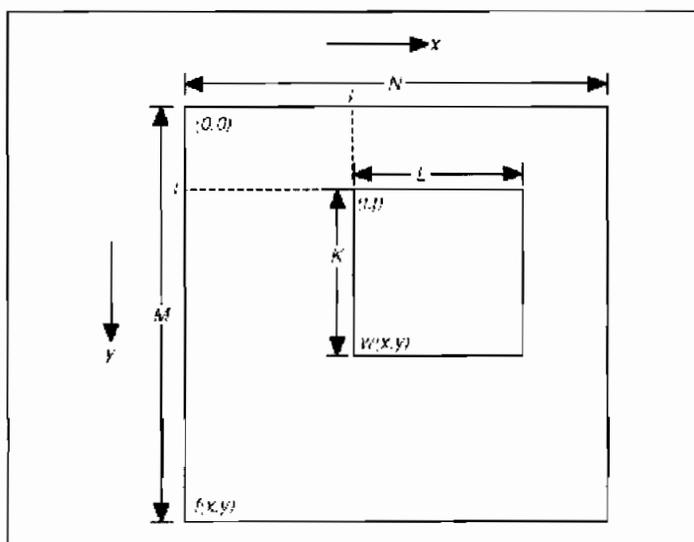


Figura 1.36 Procedimiento de Correlación

La Correlación básica es muy sensitiva a cambios de amplitud (intensidad) en la imagen y en la plantilla. Por ejemplo, si la intensidad de la imagen f es doblada, entonces los valores de C también se verán afectados. Se puede superar esa sensibilidad mediante el computo del coeficiente de correlación normalizado $R(i,j)$, que es definido como:

$$R(i, j) = \frac{\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} \left(w(x, y) - \bar{w} \right) \left(f(x+i, y+j) - \bar{f}(i, j) \right)}{\left[\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} \left(w(x, y) - \bar{w} \right)^2 \right]^{\frac{1}{2}} \left[\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} \left(f(x+i, y+j) - \bar{f}(i, j) \right)^2 \right]^{\frac{1}{2}}} \quad (\text{Ec. 1.7})$$

Donde \bar{w} (calculado solamente una vez) es el promedio de valores de intensidad de píxeles de la imagen modelo w de resolución $K \times L$, en tanto que \bar{f} es el valor promedio de la imagen f en la región coincidente con la actual localización de w . El valor de R varía en el rango -1 a 1 y es independiente de cambios de escala en los valores de intensidad de las imágenes f y w .

b) Emparejamiento Piramidal

Se puede mejorar el tiempo de procesamiento mediante la reducción del tamaño de la imagen y de la plantilla. En este método, la imagen y la plantilla son probadas a una resolución espacial más pequeña, pueden ser reducidas a $\frac{1}{4}$ de su tamaño original. El emparejamiento es realizado primero sobre las imágenes reducidas, por tanto es más rápido. Una vez completado el emparejamiento, solo las áreas con un alto empate necesitan ser consideradas como áreas de emparejamiento en la imagen original.

c) Emparejamiento de Escala – Invariante

La correlación cruzada es una buena técnica para encontrar modelos en una imagen cuando los modelos en la imagen no están escalados o rotados un ángulo superior a los 10° . Para ésta técnica, se debe repetir el proceso de escalamiento del modelo y entonces realizar la operación de correlación. Esto incrementa considerablemente el tiempo de procesamiento.

Normalizar la rotación es siempre más difícil, si existe algún indicio con respecto a la rotación, ésta puede ser extraída de la imagen; por lo tanto, se debe simplemente rotar la plantilla y hacer la correlación. Sin embargo, si la naturaleza

de la rotación es desconocida, se debe buscar técnicas que requieran exhaustivas rotaciones de la plantilla.

Se puede llevar a cabo la correlación en el dominio de la frecuencia usando la transformada rápida de fourier (FFT). Si la imagen y la plantilla son de igual tamaño, el proceso de correlación es más eficiente que en el dominio espacial. En el dominio de la frecuencia, la correlación es obtenida mediante la multiplicación de la FFT de la imagen y la conjugada compleja de la FFT de la plantilla; por lo tanto, es más difícil implementar esta técnica en el dominio de la frecuencia.

Las limitantes que presentan cada una de las técnicas tradicionales aquí revisadas, dan lugar al origen de nuevas técnicas de procesamiento de imágenes que tratan de incorporar el entendimiento de imágenes, las cuales se refieren al poder generar información acerca de las características de una imagen de modelo. Esos métodos incluyen lo siguiente:

- Modelado geométrico de las imágenes
- Eficiente muestreo (no uniforme) de las imágenes
- Extracción de información de la plantilla que es independiente tanto a la rotación como al escalamiento.

Estas técnicas reducen la cantidad de información que se requiere para caracterizar totalmente a una imagen o modelo, lo cual genera mucha velocidad al proceso. También, extraen información importante de una plantilla y remueve las informaciones redundantes y de ruido, con lo cual brinda una búsqueda mas precisa.

Una vez que se ha estudiado ciertas técnicas para el procesamiento de imágenes, se dará énfasis en el siguiente capítulo, al diseño del programa de control con la ayuda de LabVIEW y de su librería IMAQ Vision, para de ésta manera aplicar los conceptos estudiados para la implementación del software para el reconocimiento y conteo de vehículos en varias secuencias de imágenes.

CAPITULO 2

DISEÑO DEL CONTROL PARA EL MONITOREO Y CONTEO DE VEHÍCULOS

2.1 INTRODUCCIÓN

En este capítulo se describe los pasos seguidos en base a poder realizar el monitoreo y conteo vehicular a partir de varios videos tomados bajo diferentes condiciones ambientales en diferentes calles (una vía, un carril) para luego ser descargadas en disco duro de un computador para de ahí empezar con el procesamiento del video recopilado; de esta manera, se aprecia que el control implementado se realizará en tiempo no real.

El software encargado del procesamiento es LabVIEW, ya que constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Es un entorno destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el lenguaje C o BASIC. Sin embargo, LabVIEW se diferencia de dichos programas al usar programación gráfica o lenguaje G para crear programas basados en diagramas de bloques.

LabVIEW posee extensas librerías de funciones y subrutinas. Además de las funciones básicas de todo lenguaje de programación, LabVIEW incluye librerías específicas como para la adquisición de datos, control de instrumentación (VXI), GPIB, comunicación serie, análisis presentación y bases de datos, procesamiento digital de imagen (IMAQ Vision), etc. A partir de ésta última, el presente proyecto hará buen uso de los distintos bloques de ésta librería, con el fin de lograr el objetivo de reconocimiento vehicular a partir de secuencias de imágenes.

El proceso a realizarse es el siguiente:

- Se realiza filmaciones de pocos minutos de duración, utilizando una cámara digital miniDV.
- Una vez capturado el video se procede a descargar a un determinado formato en el computador con la ayuda del puerto firewire, para obtener una buena calidad de descarga así como también un incremento en la velocidad de la misma.
- Utilizando un programa de edición y conversión de video, se realiza la correspondiente segmentación del video en imágenes para de esta forma poder realizar un procesamiento de imágenes aptas para poder hacer uso de LabVIEW así como también de sus librerías, en este caso las de IMAQ Vision.
- Mediante el software de control se ejecuta un programa encargado de procesar cada una de las imágenes obtenidas en el paso anterior, para de esta forma entrar al proceso de reconocimiento y conteo de vehículos en dichos videos.

2.2 CAPTURA DE VIDEO

En base a los requerimientos del proyecto (reconocimiento vehicular en calles de un carril y en un solo sentido) se realizaron filmaciones de corta duración en distintas condiciones como puede ser de: iluminación, sentido de movilización y congestión vehicular. Para de esta forma demostrar la eficacia en el proceso de identificación y conteo de vehículos en distintas calles de la ciudad.

Para la captura del video se escogió la cámara digital DCR – TRV33 (ver anexo A) debido a su capacidad, definición, versatilidad y opciones de descarga en la PC.

En base a esto las condiciones de filmación serán las siguientes:

- Posición de la cámara perpendicular con respecto al sentido de movimiento de los vehículos.
- A una altura de aproximadamente 1.5 m del nivel de la acera.
- A una distancia aproximada de 5 m. (medidos desde la cámara hasta el vehículo)

De ésta manera se tiene un campo de visión considerable en el que se logre visualizar un vehículo de considerables dimensiones, como se indica en la siguiente figura:

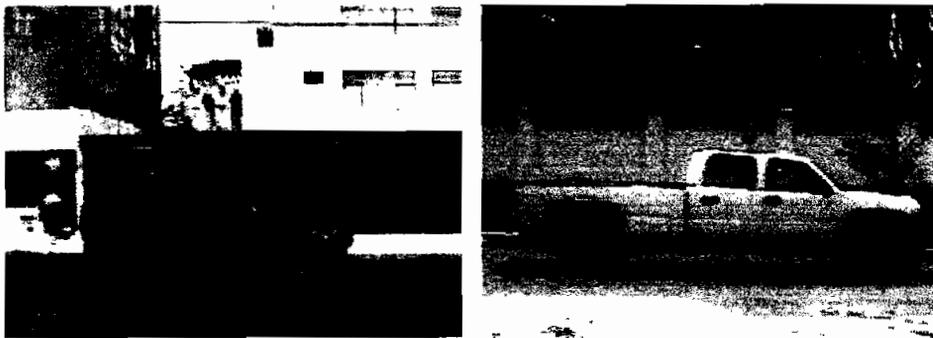


Figura 2.1 Campos de visión

2.3 DESCARGA DE VIDEO

Una vez realizada la filmación y con la grabación de la misma en un cassette de video digital mini DV, se procede con la transferencia de datos desde la cámara hacia el computador usando la interface IEEE 1394 (Firewire) disponible como opción de descarga en la cámara digital elegida, para que así exista tal transferencia de audio y video sin pérdida de calidad y de "frames" o desincronización.



Figura 2.2 Transferencia de audio y video

En el computador se dispone de un software apropiado de edición no lineal (NLE) como lo es Ulead Media Studio Pro 7, que configure parámetros de descarga. Y así elegir un formato de grabación como lo es el MPEG – 1, cuya resolución y calidad es suficiente para el proyecto, además por la calidad de compresión.

2.3.1 INTERFAZ IEEE 1394

El bus IEEE1394 [6] (también llamado Firewire, iLink o terminal DV) es un bus serie de alta velocidad complementario del USB que mejora la conectividad de dispositivos incluyendo videocámaras, dispositivos de almacenamiento y periféricos, coexiste pacíficamente con USB quedando éste para periféricos de menor ancho de banda, los cuales no son compatibles.

Las diferencias entre ambos buses se pueden observar en la siguiente Tabla:

	IEEE 1394 Firewire	USB
Número máximo de dispositivos	62	127
Cambio en caliente (agregar o quitar dispositivos sin tener que reiniciar el ordenador)	Si	Si
Longitud máxima del cable entre dispositivos	4,5 metros	5 metros
Velocidad de transferencia de datos	400 Mbps (50MB/s)	12 Mbps (1,5 Mb/s)
Velocidad en el futuro	800 Mbps (100MB/s) 1 Gbps+ (125MB/s+)	versión 2.0 hasta 460MB
Implementación en Macintosh	Si	No
Conexión de periféricos interna	Si	No
Periféricos típicos	<ul style="list-style-type: none"> - Videocámaras DV - Cámaras digitales de alta resolución - HDTV (TV de alta definición) - Cajas de conexiones - Discos duros - Unidades DVD-ROM - Impresoras - Escáneres 	<ul style="list-style-type: none"> - Teclados - Ratones - Monitores - Joysticks - Cámaras digitales de baja resolución - Unidades CD-ROM de baja velocidad - Módems

Tabla 2.1 Diferencias básicas entre los buses USB y Firewire

Este bus fue desarrollado por Apple para su gama de ordenadores con la idea de sustituir al bus SCSI. A finales de 1995 el IEEE editó el actual estándar 1394.

El protocolo es tanto asíncrono como isócrono. Esto significa que es posible negociar tanto un ancho de banda fijo (para dispositivos como las cámaras DV que necesitan una transferencia constante y en tiempo real) como variable (para impresoras, escáners, etc.) simultáneamente por el mismo bus.

Las tarjetas Firewire han pasado de ser una rareza a ser algo habitual como la que se muestra en la Figura 2.3. Se dieron a conocer debido sobre todo a la lista de tecnologías contenidas en Windows 98, pero como está orientado al vídeo, se crea cierta confusión, al llamarlas "capturadoras de vídeo" más que como puertos de transferencia de datos, siendo tan solo una interfaz de transferencia de datos.

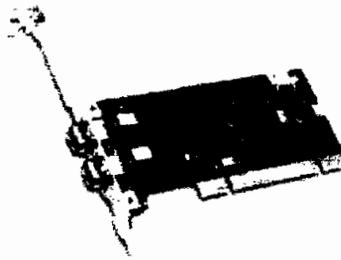


Figura 2.3 Tarjeta Firewire

Normalmente, las tarjetas Firewire llevan un conector para seis cables como se indica en la Figura 2.4, cuatro de señal (en modo diferencial) y dos más para alimentar los dispositivos externos (algunas tarjetas Firewire, como la Digital Origin IntroDV vienen con un conector de 4 pines y un cable de 4 pines por ambos extremos dando opción para interconectar dos cámaras miniDV). Las cámaras de vídeo, sin embargo, tienen un conector de 4 pines, ya que no necesitan ser alimentadas externamente. Por tanto, se necesita un cable de "6 a 4 pines" para conectar una tarjeta Firewire a una cámara miniDV.

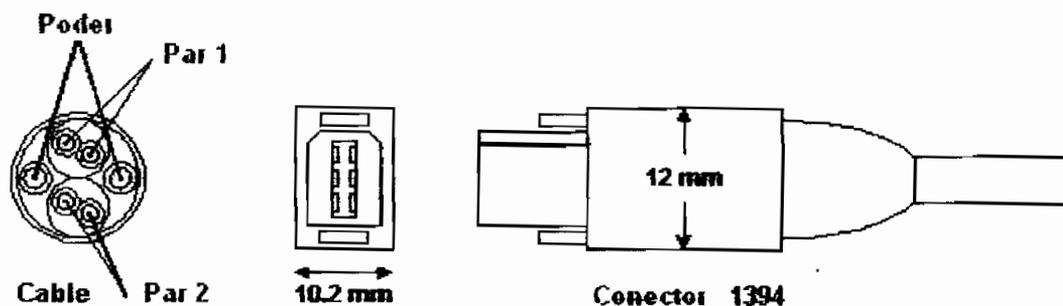


Figura 2.4 Estructura del cable y conector 1394

Requerimientos de instalación.

Por el momento hay muy pocas placas bases con este bus. Por consiguiente, hay que tener una tarjeta PCI IEEE1394.

Las tarjetas son interfaces de transferencia de los datos (audio y video) grabados en cámaras miniDV al ordenador en tiempo real. También permiten controlar las funciones de avance, rebobinado, etc. Todas hacen uso del mismo driver OHCI hecho por Microsoft.

- El sistema operativo debe ser W98SE (sólo 98 no vale) o superior (W98ME, W2000, XP) en tanto que Windows NT no es compatible.
- El driver es el mismo para todas las cámaras, no es específico para una en concreto, y van incluidos en el disco de instalación de Windows.
- Es absolutamente fundamental que el disco duro del ordenador esté funcionando en modo DMA. Si no lo está, es muy probable que se tenga problemas en la transferencia de datos.
- También es importante que no comparta interrupciones con otros dispositivos físicos (tarjetas) del PC.
- Es recomendable instalar la última versión de DirectX.

2.4 SEGMENTACION DE VIDEO A IMAGENES

Para la descomposición del video en frames, se utilizó el software de edición y conversión "Blaze MediaConvert 3.0" ya que ofrece mayores facilidades en formatos de entrada (video) como de salida (imágenes), obteniendo condiciones favorables de nitidez , definición en video e imagen, resolución configurable y poco tamaño de imagen en disco.

A continuación se detalla las características más relevantes de dicho software así como también un ejemplo de edición.

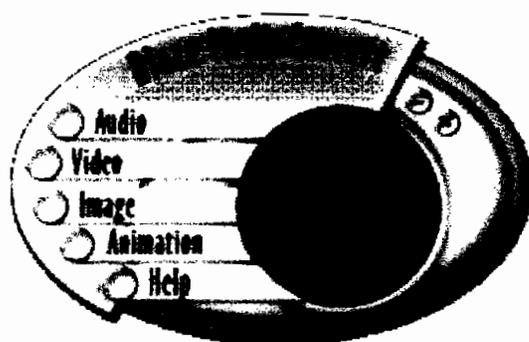


Figura 2.5 Pantalla principal del programa Blaze Media Convert 3.0

Este software es una herramienta que permite realizar conversiones entre archivos en más de 70 formatos de audio, imágenes, animaciones y videos; de una manera rápida y sencilla. A diferencia de otras aplicaciones que solo convierten entre uno o pocos formatos, con este programa se puede realizar casi cualquier tipo de conversión. También permite convertir MP3s o WAVs a CD y viceversa. Entre otras características incluye la creación y decompilación de GIF's animados, y mucha versatilidad en lo que a edición se refiere.

En cuanto a video, este programa puede realizar conversiones bidireccionales a través de MPEG-1, MPEG-2, AVI, WMV, Multi-Page TIFF y FLIC. El constructor de video (compilador) combina imágenes elegidas como frames para poder crear un video en un determinado formato. En tanto que el descompilador graba todos los frames de un archivo de video seleccionado en imágenes simples.

Las imágenes extraídas pueden ser grabadas en formatos como: BAY, BMP, GIF, JPG, PCX, PNG, PNM, RAS, RGB, TGA, TIFF y XPM

Ejemplo de segmentación de video a imágenes

Los formatos de entrada de video del programa deben ser AVI, MPEG, WMV, FILC, TIFF. Para tener mejor calidad y a su vez resolución, el formato elegido fue MPEG – 1. Como se busca hacer la conversión de video a secuencia de imágenes, se introduce en el menú video ofreciéndonos las siguientes posibilidades:

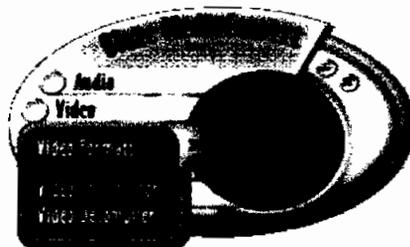


Figura 2.6 Modos para la edición de video

- **Video Formats** (Cambiar formato de video)
- **Video Constructor** (Construir video a partir de secuencias de imagen)
- **Video Decompiler** (Transformar video a secuencias de imágenes)

Una vez escogido la opción (decompilar video) se presenta la pantalla mostrada en la Figura 2.7, en la cual se puede elegir en la barra de herramientas add (traer el video a procesar) , remove (borrar video) , decompilar (inicio de segmentación).

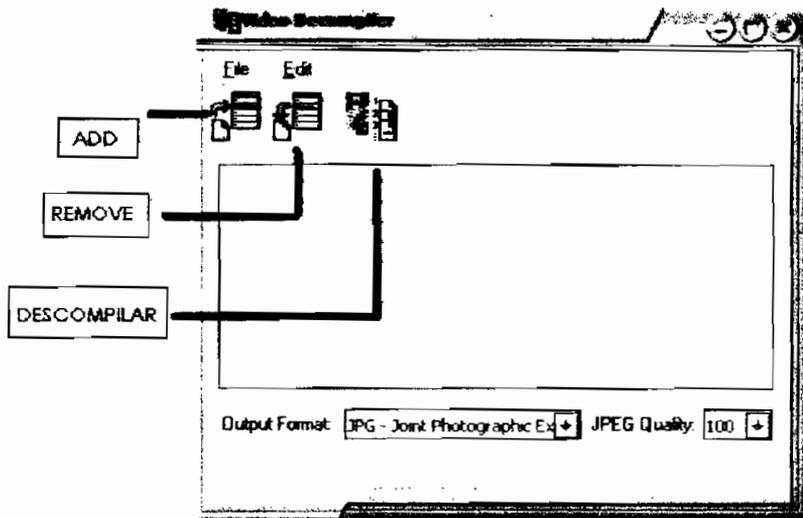


Figura 2.7 Pantalla para segmentación de video a imágenes

Primero se debe elegir la velocidad de refresco o de captura acorde con el tiempo en el proceso de control implementado, para ello en la opción *FILE / ENABLE FRAME INCREMENT* se introduce dicha constante en la siguiente ventana.

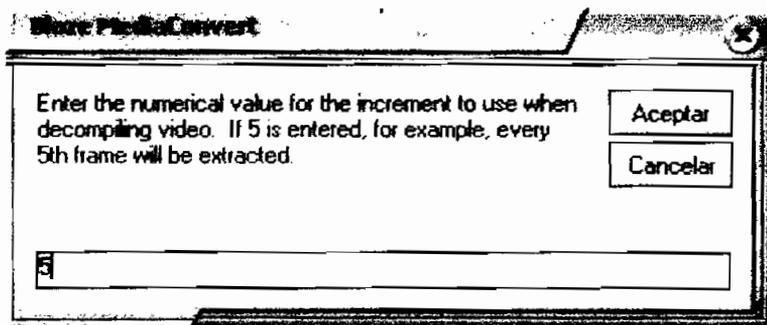


Figura 2.8 Elección de velocidad de refresco

En la Figura 2.8 se ilustra que se elige incrementos en múltiplos de un número entero, con lo cual el software hace un cálculo de número total de frames que componen el video y luego establece los cuadros seleccionados. Como por ejemplo al elegir 5 y la velocidad de refresco original del video era 30 fps; entonces, se tomarán del total de cuadros solo los múltiplos de 5 dándonos una conversión a 6 fps (como secuencia de imágenes).

El valor escogido para este parámetro es de tres para de esta forma tener todavía una percepción continua de movimiento; además, se logrará emparejar el tiempo normal de ejecución de la película con el tiempo destinado a la reproducción de la secuencia de imágenes y el tiempo de procesamiento de las mismas.

En cuanto al formato de salida ya en imágenes se eligió JPEG, con la finalidad de ocupar poco tamaño en el disco duro y por ende acelerar la velocidad del procesamiento. Así también se puede ajustar la calidad de conversión refiriéndose a la nitidez de la imagen.

Antes de iniciar con el proceso de segmentación, aparece una ventana de dialogo como se indica en la Figura 2.9, en el cual se elige una carpeta en donde se van a ubicar la secuencia de imágenes.

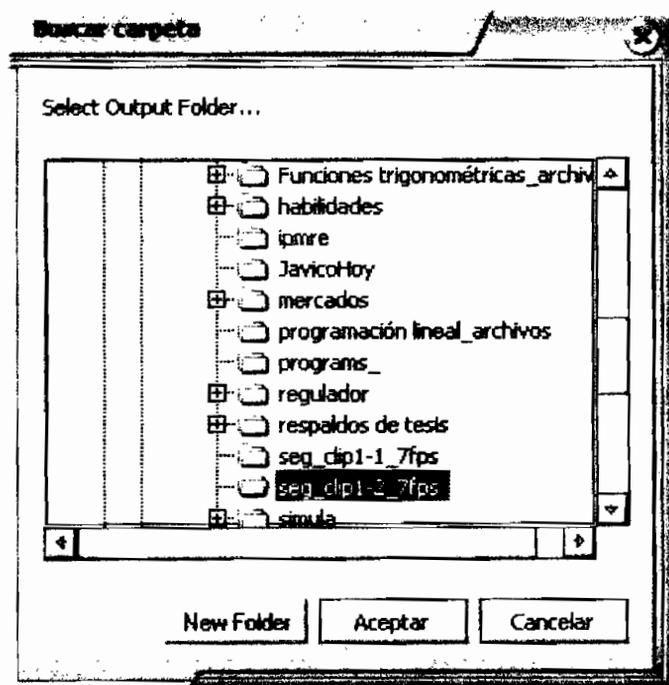


Figura 2.9 Explorador para la ubicación de imágenes

Una vez aceptada la ubicación de la carpeta, el proceso de conversión es ilustrado en la barra de estado presentada en la Figura 2.10.

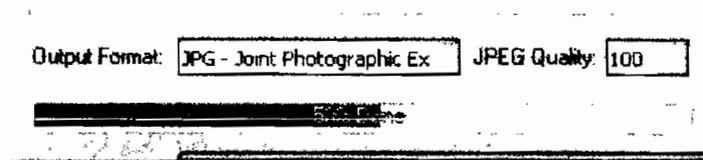


Figura 2.10 Estado del proceso de segmentación

Una vez finalizado este proceso, a cada imagen capturada es asignado un nombre, como se indica en la siguiente Figura, en la cual también se ilustra el tipo, tamaño y dimensiones de la imagen. El tamaño o resolución de la imagen también conservó las medidas originales que tenía en el video cuyo formato era MPEG - 1.

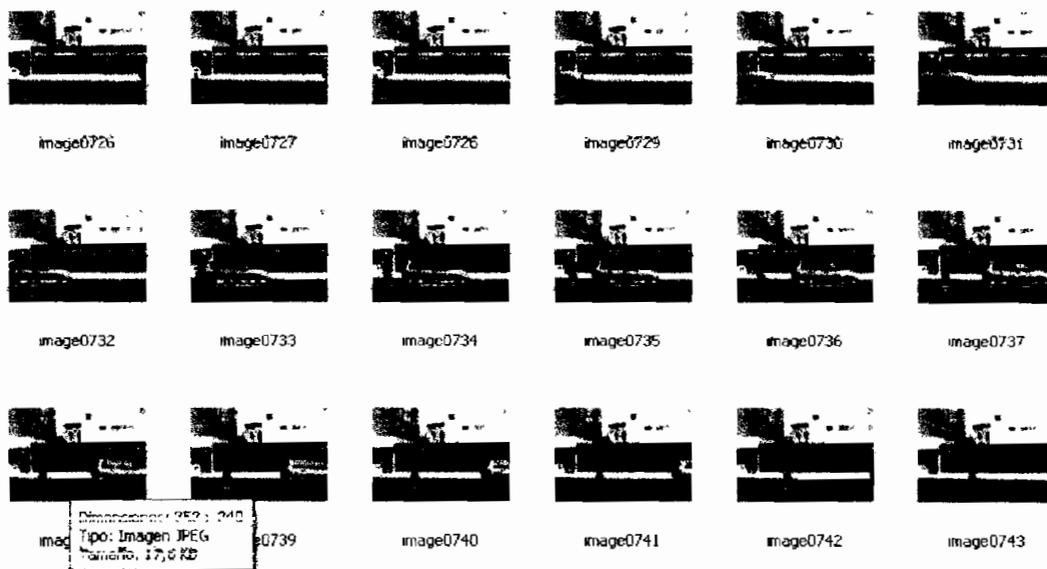


Figura 2.11 Secuencia de imágenes

2.5 DISEÑO DEL PROGRAMA PRINCIPAL

El programa principal se encarga básicamente de dos tareas, el monitoreo de vehículos y el procesamiento de imágenes:

- En lo referente al monitoreo, el programa se encarga de la visualización de las imágenes a secuencia y así tener la percepción de movimiento (video); es decir, volver de imágenes estáticas a video. De esta manera el usuario

podrá ir observando el video en tanto que dichas imágenes son procesadas.

- En cuanto al procesamiento de imágenes, el programa principal analiza todas las imágenes y define el frame en el cual se aplica los distintos algoritmos de reconocimiento vehicular.

Dicho análisis de todas las imágenes se realiza trazando dos líneas verticales en cada una de las imágenes, para de esta manera poder medir el nivel de gris (ng) promedio que existe en cada una de las mismas. Estas líneas por defecto tendrán las siguientes coordenadas:

- Línea 1: (150, 180, 150, 203)
- Línea 2: (270, 180, 270, 203)

La Figura 2.12 ilustra la posición de dichas líneas colocadas a una distancia adecuada y cerca de las ruedas para que sólo objetos de dimensiones parecidas a las de un vehículo produzcan cambios de contraste; entonces, presentándose la perturbación.

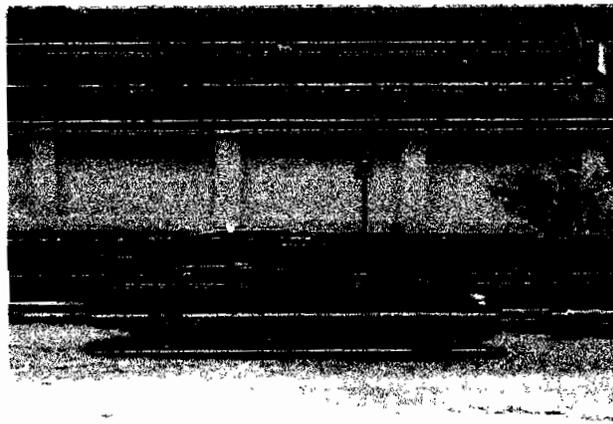


Figura 2.12 Detección de perturbación

Por lo tanto, este método consiste en medir la variación media del nivel de gris de los píxeles a lo largo de las dos líneas; en donde, si no existe cambios de contraste, los valores medidos de ng permanecerán en un rango de 110 a 255

(tono claro), dichas variaciones son producidas solamente por condiciones ambientales.

La perturbación existirá solamente cuando el nivel de gris promedie valores menores a los 110 (tono oscuro); de esta manera, se dará paso a los distintos algoritmos de reconocimiento vehicular.

2.6 DISEÑO DE LOS ALGORITMOS DE CONTROL

Los algoritmos de reconocimiento diseñados en este proyecto son aplicados a las imágenes seleccionadas en serie, con la finalidad de que si uno de ellos falla venga un segundo o sino un tercero, y de esta manera aumentar el porcentaje de efectividad en lo que a reconocimiento se refiere.

El orden de aplicación de los mismos es en base a los números de aciertos obtenidos en las pruebas, el cual es el siguiente:

1. Emparejamiento de modelo
2. Distancia entre ruedas
3. Aislar la forma del vehículo

2.6.1 DISEÑO DEL ALGORITMO DE RECONOCIMIENTO EMPAREJAMIENTO DE MODELO

Para este algoritmo se utiliza la técnica del emparejamiento de modelo (pattern matching), la cual es una de las más poderosas en lo que al reconocimiento se refiere. Básicamente consiste en buscar modelos o plantillas en las imágenes a procesar, con un determinado grado de exactitud. En este caso los modelos son imágenes de corta resolución que tienen las siguientes partes: el aro, la rueda, parte de la carrocería y de pavimento.

Para ser considerados como plantillas deben aprobar algunos criterios como son aros claros u oscuros, llanos o con detalles; de esta forma, se seleccionó cinco

vehículos "tipo" de la variedad de vehículos que se tuvo en la segmentación de algunos videos, con los cuales en la fase experimental se tuvieron un mayor número de aciertos en lo a que emparejamiento se refiere. En la siguiente Figura se ilustra la imagen estática de uno de los vehículos que será de base para crear uno de los modelos en el proceso de emparejamiento.

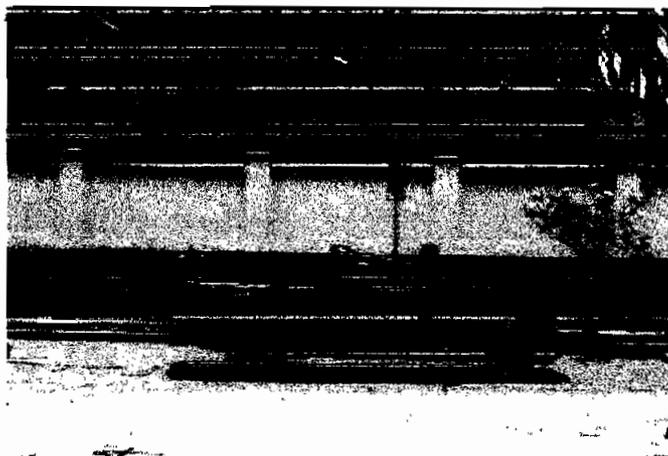


Figura 2.13 Imagen estática de uno de los vehículos "tipo"

Para crear en sí el modelo o plantilla es necesario realizar un enfoque en la parte característica del vehículo como se explicó anteriormente, esto se logra con la ayuda del siguiente bloque de las librerías de IMAQ Vision.

IMAQ Extract

Reduce una imagen o parte de una imagen con el ajuste de su resolución horizontal y vertical.

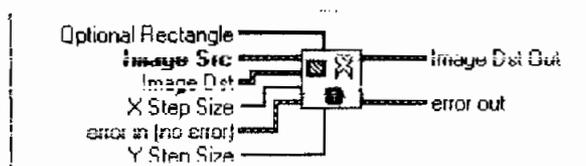


Figura 2.14 IMAQ Extract

Para cada uno de los modelos, en *Optional Rectangle* se ingresa un valor constante, que indicará las coordenadas de los puntos P1 y P2; como se observa

en la Figura 2.15, para así poder extraer cierta parte de los vehículos "tipo", quedando dichas coordenadas de la siguiente forma:

- *Modelo1* (235, 165, 278, 205)
- *Modelo2* (195, 163, 238, 208)
- *Modelo3* (208, 158, 258, 198)
- *Modelo4* (228, 168, 288, 208)
- *Modelo5* (220, 170, 250, 203)

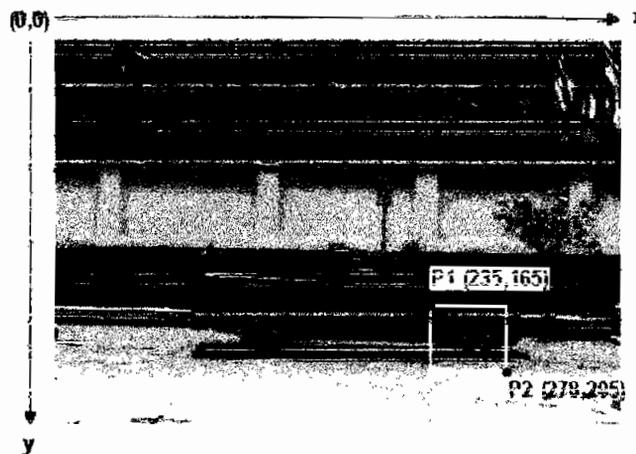


Figura 2.15 Ejemplo de uso del bloque IMAQ Extract

De esta forma ajustando las coordenadas de las imágenes estáticas de los vehículos "tipo" se logró tener los siguientes cinco modelos mostrados conjuntamente con su resolución en la siguiente Figura.

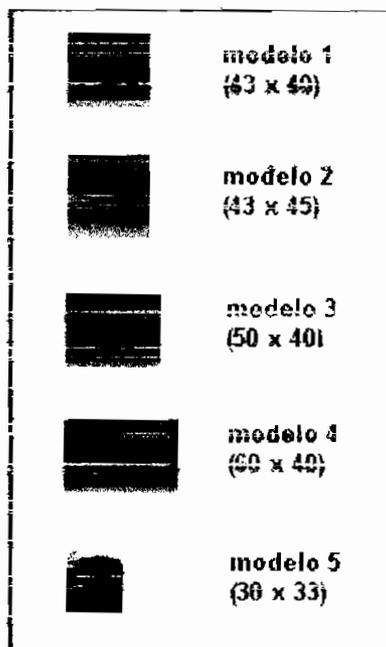


Figura 2.16 Modelos usados para el proceso de emparejamiento

2.6.1.1 Proceso de Aprendizaje

Antes de realizar la tarea de búsqueda, el programa debe realizar el aprendizaje de dichos modelos utilizando los siguientes bloques:

IMAQ Setup Learn Pattern

Configura parámetros que son usados durante éste proceso.

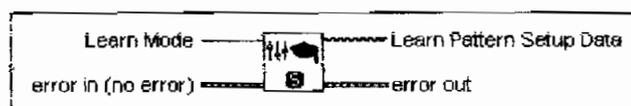


Figura 2.17 IMAQ Setup Learn Pattern

Learn Mode: especifica el modo de invariancia a usar en el proceso de aprendizaje y pueden ser:

- *Shift Information*
- *Rotation Information*
- *All*

En el programa se utilizó un modo de aprendizaje total (*All*); es decir, con información de tamaño y de rotación.

IMAQ Learn Pattern

Crea una descripción de la plantilla a buscar durante el proceso de emparejamiento, en base a los datos entregados por el bloque de configuración explicado anteriormente.

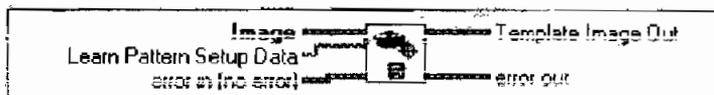


Figura 2.18 IMAQ Learn Pattern

Image: se refiere a una determinada plantilla a aprender con su respectiva resolución como se indicó en la figura 2.16.

2.6.1.2 Proceso de Búsqueda

Una vez aprendidos los modelos, se procede con la configuración de emparejamiento y la búsqueda en si de cualquiera de los modelos o plantillas sobre la imagen original. En la siguiente Figura se ilustra la imagen estática con resolución original de un vehículo al cual se realizará el proceso de emparejamiento de los modelos mencionados anteriormente.

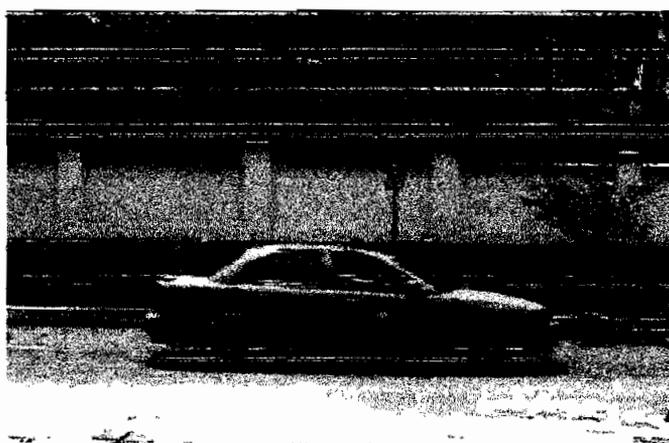


Figura 2.19 Imagen de un vehículo a procesar

Los bloques a utilizarse en éste proceso son los siguientes:

IMAQ Setup Match Pattern

Configura parámetros, tales como el contraste mínimo así como también el modo de emparejamiento como puede ser de tamaño o de rotación invariante.

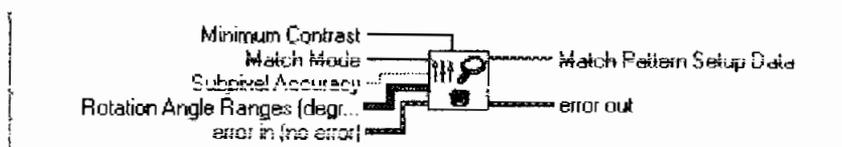


Figura 2.20 IMAQ Setup Match Pattern

Minimum Contrast: especifica el mínimo contraste que se espera en la imagen. Contraste es definido como la diferencia entre los valores máximos y mínimos de píxel en la imagen. El valor escogido es 10.

Match Mode: especifica la técnica de búsqueda y puede ser:

- *Shift Invariant (modo escogido)*, busca la plantilla en la imagen asumiendo que no tiene una rotación de $\pm 4^\circ$.
- *Rotation Invariant*, busca la plantilla en la imagen sin ninguna restricción en la rotación de la plantilla.

Subpixel Accuracy: (puesto a falso), se refiere a que si después de este proceso los píxeles de la imagen procesada son alterados.

IMAQ Match Pattern

Busca una imagen modelo o una imagen de plantilla en la imagen de entrada de acuerdo a un determinado grado de exactitud del emparejamiento, ajustando también el número de objetivos máximo a buscar.

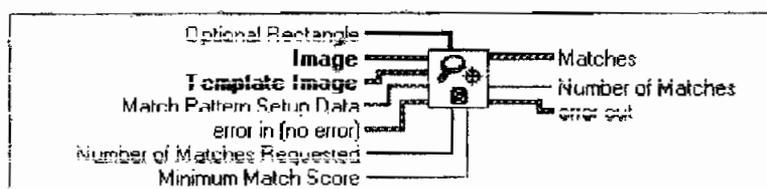


Figura 2.21 IMAQ Match Pattern

Image: corresponde a la imagen o secuencia de imágenes a procesar, la cual utilizando el IMAQ Extract se consigue realizar la búsqueda solamente en la parte inferior de las imágenes de acuerdo a las siguientes coordenadas (0, 152, 352, 220), y de esta forma se consigue aumentar la velocidad en el proceso de emparejamiento, así como también la posibilidad de escoger un objetivo equivocado.

Template Image: es la imagen plantilla aprendida en el proceso anterior.

Number of matches request: se refiere al número de objetivos o aciertos válidos que se esperan en la imagen a procesar. Vale 2 para los modelos uno, dos y tres; sin embargo, para los modelos cuatro y cinco, ésta entrada toma el valor de 1 para evitar tener objetivos erróneos ya que se trata de modelos con poca posibilidad de acierto.

Minimum match store: es el puntaje mas bajo que un objetivo debe tener para ser considerado como válido, y son los siguientes dependiendo de los modelos:

- *Modelo1: 820 / 1000*
- *Modelo2: 750 / 1000*
- *Modelo3: 700 / 1000*
- *Modelo4: 730 / 1000*
- *Modelo5: 850 / 1000*

Number of matches: es un valor numérico de los aciertos del proceso y cuando éste es mayor que 0, la imagen procesada corresponde al paso de un vehículo.

Draw Matches Position.vi

Dibuja en la imagen procesada los objetivos como resultado del emparejamiento de modelo



Figura 2 22 IMAQ Matches Position.vi

Colocando el bloque anterior se consigue sobrescribir en la imagen o región de búsqueda, los objetivos que tuvo el proceso de emparejamiento como se observa en la siguiente figura:



Figura 2.23 Región de búsqueda y objetivos del emparejamiento

2.6.2 DISEÑO DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE RUEDAS

Este método se focaliza en la búsqueda de las ruedas como también en la medición de la distancia entre las mismas; siendo estas, partes básicas constitutivas de un vehículo. La distancia entre las ruedas juega un papel importante ya que se puede diferenciar vehículos livianos como: bicicletas y motocicletas, que no entran al reconocimiento del presente proyecto.

Con el fin de que este algoritmo sea eficaz, se desarrollaron dos tipos de procesamientos en paralelo para cubrir con la diversidad en el color del aro de los vehículos; de esta manera, en las imágenes bajo análisis se podrá reconocer sus ruedas ya sean de un tono claro o de un tono oscuro. Una vez detectadas las dos ruedas (al mismo nivel) por cualquiera de los dos procesos, se procederá a medir la distancia entre círculos; y si es la adecuada, dicha imagen procesada corresponderá a la de un vehículo.

Para lograr el objetivo de aislar las dos ruedas de las imágenes estáticas que requieren de procesamiento, será necesaria la aplicación de las siguientes etapas:

- Pre-procesamiento
- Umbralización
- Morfología Binaria
- Medición de partículas

Se propone para la explicación del proceso a los vehículos, cuyos tonos de los aros son claros, la descripción de cada etapa así como también de los respectivos bloques involucrados para la aplicación de este algoritmo de reconocimiento.

2.6.2.1 Pre-procesamiento

En este caso también con la ayuda del IMAQ Extract, se focaliza el análisis solamente en la parte inferior de las imágenes con las siguientes coordenadas (0, 135, 352, 208), como se muestra en la siguiente Figura:



Figura 2.24: Región de búsqueda para el aislamiento de las ruedas.

Esta etapa consiste en preparar a la imagen para un determinado tipo de procesamiento que se vaya a efectuar, en este caso al de umbralización. Para ello se uso un tipo de filtro pasa bajos para conseguir atenuar las variaciones de la intensidad de luz y enfatizar modelos de variaciones graduales como los objetos y el fondo; es decir, se tiene una tendencia a aplanar las imágenes, eliminando detalles y manchando los bordes. A continuación se describen los bloques de la librería de IMAQ Vision para conseguir dicho objetivo.

IMAQ GetKernel

Lee un kernel predefinido. Su código consiste en tres unidades separadas: Kernel Family, Kernel Size y Kernel Number.

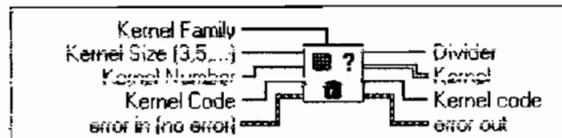


Figura 2.25 IMAQ GetKernel

Kernel Family: determina el tipo de matriz a usar y estas pueden ser:

- *Gradiente*
- *Laplaciano*
- *Aplanamiento (escogido, para aplicar un filtro lineal pasa bajos)*
- *Gausiano*

Kernel Size: determina el orden de la matriz a aplicarse y estos pueden ser 3, 5 y 7 correspondiente a las convoluciones del tipo 3x3, 5x5 y 7x7 respectivamente. El valor escogido para este caso es 3.

Kernel Number: corresponde a un número de dos dígitos entre 0 y n, que pertenece a una familia y a un tamaño. El número escogido es 0.

Kernel Code: es un código que puede utilizarse para acceder directamente a la matriz de convolución, en vez de aplicar valores a las tres entradas anteriores.

Divider: es el factor de normalización asociado con el Kernel retribuido.

Kernel: es la matriz resultante y corresponde a la codificación Kernel especificada por las entradas anteriormente explicadas. Esta salida se la puede conectar directamente a la entrada *Kernel* del bloque IMAQ Convolute.

IMAQ Convolute

Filtra una imagen usando un tipo de filtrado lineal.

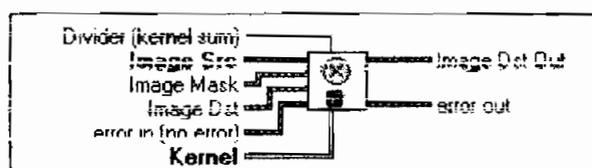


Figura 2.26: IMAQ Convolute

Sus entradas corresponden a las salidas del bloque IMAQ GetKernel, con esto se consigue la siguiente imagen filtrada:



Figura 2.27: Región de búsqueda filtrada

2.6.2.2 Umbralización (Thresholding)

El proceso de Umbralización es usado para aislar objetos de interés en una imagen, eligiendo un intervalo adecuado entre 0 y 255 para una imagen en niveles de gris, y de esta forma seguir realizando técnicas de procesamiento sobre imágenes binarias. Esto se consigue con la aplicación del siguiente bloque:

IMAQ Threshold

Produce partículas binarias con la ayuda de la umbralización de la imagen.

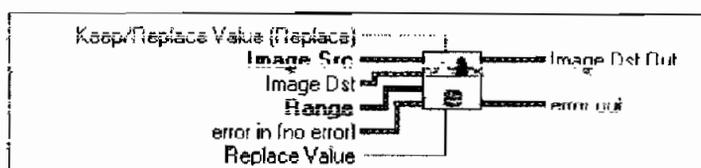


Figura 2.28 IMAQ Threshold

Keep/Replace Value: Replace (default), este parámetro determina si reemplaza o no los valores de píxeles que están en el intervalo de umbralización (*Range*) por el valor de la entrada *Replace Value*. El valor por defecto es verdadero y reemplaza esos valores de píxel; en el caso de que sea falso, esos píxeles mantienen sus valores originales.

Image Src: esta entrada corresponde a la imagen en niveles de gris que se requiere transformar a imagen binaria.

Range: se refiere al intervalo de umbralización, en donde todos los píxeles cuyos valores de niveles de gris están fuera de este rango, son puestos a cero y todos los demás píxeles son reemplazados por el valor de *Replace Value* (si se permite). En el presente ejemplo se eligió el rango (50, 100).

La paleta a usar en la visualización dependerá del valor colocado en el *Replace Value*. Para observar las partículas se recomienda usar una paleta binaria, como lo es en el presente caso.



Figura 2.29 Imagen binaria producida por el bloque IMAQ Threshold

2.6.2.3 Morfología Binaria

Una vez obtenida la imagen binaria, se realizará una serie de operaciones morfológicas primarias y avanzadas de tal forma que se logre aislar las ruedas en las imágenes bajo análisis. Esto se consigue con la aplicación de los bloques que se describen a continuación, así como también de los parámetros que se ajustarán al tipo de aros en cuestión.

IMAQ RejectBorder

Elimina partículas que tocan el borde de una imagen. La imagen fuente (*Image Src*) debe ser una imagen binaria de 8 bits.

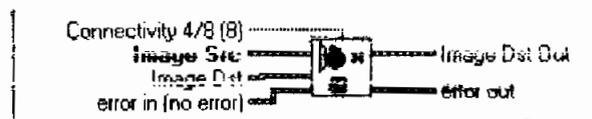


Figura 2.30 IMAQ RejectBorder

Connectivity 4/8: 8 (default), especifica cómo el algoritmo determina si un pixel adyacente pertenece a la misma o a una diferente partícula.



Figura 2.31 Eliminación de partículas que tocan el borde de la imagen

IMAQ FillHole

Rellena los agujeros encontrados dentro de una partícula, los mismos que son llenados con un valor de píxel igual a 1. La imagen fuente debe ser una imagen binaria de 8 bits.

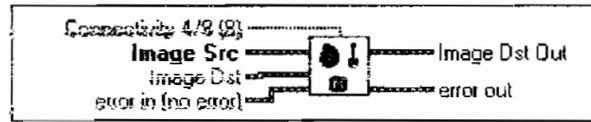


Figura 2.32 IMAQ FillHole

En la siguiente Figura se ilustra el relleno de las partículas que básicamente constituyen los aros de los vehículos.

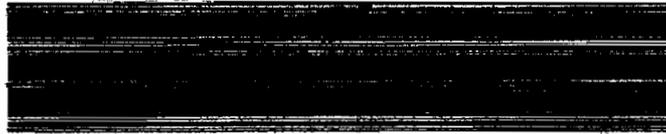


Figura 2.33 Relleno de partículas

IMAQ Morphology

Realiza transformaciones morfológicas primarias. Las imágenes fuente y destino deben ser imágenes binarias de 8 bits. La imagen fuente debe ser creada con un borde capaz de soportar el tamaño de un elemento estructural [3], en tanto que, el tamaño del borde de la imagen destino no es importante.

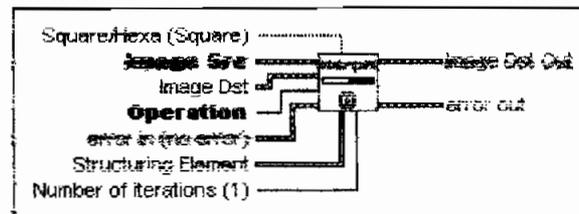


Figura 2.34 IMAQ Morphology

Square/Hexa: *Square (default)*, especifica la forma del pixel-frame durante la transformación.

Operation: especifica el tipo de transformación morfológica primaria a usar, los cuales pueden ser:

- *Mediana Automática (Operación escogida)*: realiza combinaciones duales de aperturas (*open*) y cierres (*close*).
- *Cierre*: dilatación seguida por una erosión.
- *Dilatación*: realiza lo contrario a una erosión.
- *Erosión*: elimina píxeles aislados en el fondo de la imagen.
- *Gradiente*: extracción del contorno externo e interno de una partícula.
- *Gradiente exterior*: extracción del contorno externo de una partícula.
- *Gradiente interior*: extracción del contorno interno de una partícula.
- *Ganancia o pérdida*: eliminación de todos los píxeles que no siguen el mismo modelo como el encontrado en el elemento estructural.
- *Apertura*: erosión seguida por una dilatación.
- *Cierre apropiado (PClose)*: sucesión de siete cierres y aperturas.
- *Apertura apropiada (Popen)*: sucesión de siete aperturas y cierres.
- *Engrosamiento (Thick)*: activación de todos los píxeles que emparejan el modelo en el elemento estructural.
- *Adelgazamiento (Thin)*: activación de todos los píxeles que emparejan el modelo en el elemento estructural.

Estructuring Element: es una matriz 2D que contienen el elemento estructural para aplicar en la imagen, el tamaño de éste determina el tamaño del procesamiento. Si esta entrada no está conectada como lo es en este caso, es usada un elemento estructural del tipo 3 x 3.

Number of iterations: es el número de veces que el VI realiza la operación indicada. Este parámetro de entrada es puesto a 15.

La Figura 2.35 muestra la imagen de salida bajo los parámetros configurados anteriormente.



Figura 2.35 Aplicación de la operación morfológica primaria Auto - Median

IMAQ RemoveParticle

Elimina o conserva partículas que se resisten a un número específico de erosiones del tipo 3x3. La imagen fuente debe ser una imagen binaria de 8 bits.

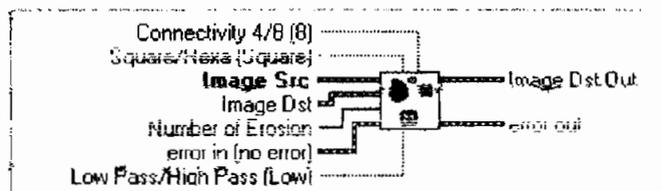


Figura 2.36 IMAQ RemoveParticle

Connectivity: 8 (default).

Square/Hexa: Square (default).

Number of erosion: especifica el número de erosiones del tipo 3 x 3 para aplicar a la imagen, el valor de default y el escogido es 2 para poder eliminar partículas que no corresponden a la de los aros.

Low Pass/High Pass: Low (default). Especifica si los objetos resistentes a n erosiones son descartados o mantenidos.

La siguiente Figura ilustra la eliminación de las partículas que no corresponden a las del aro de un vehículo.



Figura 2.37 Eliminación de partículas

2.6.2.4 Medición de partículas

Una vez aisladas las partículas de los aros, se procede ahora con la búsqueda de los dos círculos, así como también con la medición de la distancia entre las mismas (en píxeles) con la ayuda del siguiente bloque:

IMAQ Find Circles

Separa objetos circulares que se sobrelapan y los clasifica basándose en su radio, área de superficie y perímetro. Además encuentra el radio y el centro de dichos objetos circulares, así como también este VI puede trazar dichos círculos en la imagen de destino como se ilustra en la Figura 2.39.

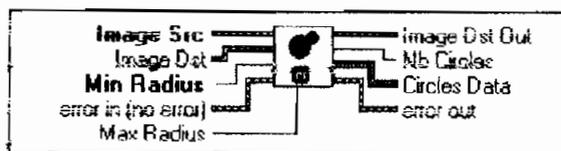


Figura 2.38 IMAQ Find Circles

Min Radius: especifica el radio más corto en píxeles que pueda detectar, el valor de default es uno. El valor ajustado para este tipo de vehiculos es igual a 6.

Max Radius: especifica el radio más largo en píxeles a detectar, el valor de default es 10. El valor ajustado para este tipo de vehiculos es igual a 15.

Circles Data: retoma una matriz de mediciones para todos los círculos detectados, cada elemento (círculo encontrado) consta de los siguientes parámetros:

- **PosX**, es la posición horizontal en píxeles del origen del círculo.
- **PosY**, es la posición vertical en píxeles del origen del círculo.
- **Radius**, es el radio del círculo encontrado en píxeles.
- **Core Area**, es el área en píxeles de la superficie del núcleo del círculo definido por el mapa de distancia de Danielsson [3].



Figura 2.39 Detección de ruedas

Para finalizar y si la salida *Nb Circles* es mayor a 2, se logra aislar los círculos que se encuentren al mismo nivel (semejantes coordenadas en Y), para de allí proceder con restas absolutas en píxeles de las posiciones en X de los radios y si

esta es mayor a la *distancia mínima* (parámetro calibrable -90-), la imagen procesada corresponde a un vehículo.

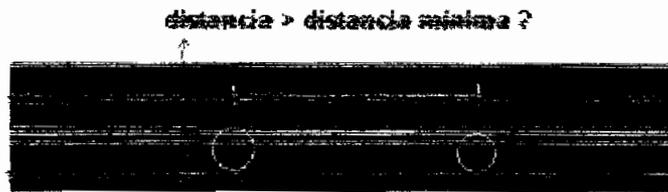


Figura 2.40 Condición para considerar como vehículo

El segundo tipo de procesamiento (vehículos con aros oscuros) se realiza en la misma región de búsqueda que en el caso anterior dada por el IMAQ Extract y con la ayuda de los siguientes bloques y parámetros para cada una de las etapas:

- IMAQ Extract: *Optional Rectangle (0, 135, 352, 208)*
- IMAQ GetKernel: *Kernel Family, Smoothing*
Kernel Size 3
Kernel Numer 0
- IMAQ Convolute
- IMAQ Threshold: *Range (10, 58)*
- IMAQ RemoveParticle: *Conectivity 8 (default)*
Square/Hexa, Square (default)
Number of erosion 2
Low Pass/High Pass, Low (default)
- IMAQ FillHole: *Conectivity 8 (default)*
- IMAQ Morphology: *Square/Hexa, Square (default)*
Operation, Hit miss
Number of Iterations 8
- IMAQ RejectBorder: *Conectivity 8 (default)*
- IMAQ Find Circles: *Min Radius 10 (píxeles)*
Max Radius 15 (píxeles)

En la siguiente Figura se presenta la imagen original para este tipo de vehículos y cada una de las imágenes que se producen con la aplicación de cada uno de los bloques indicados anteriormente.

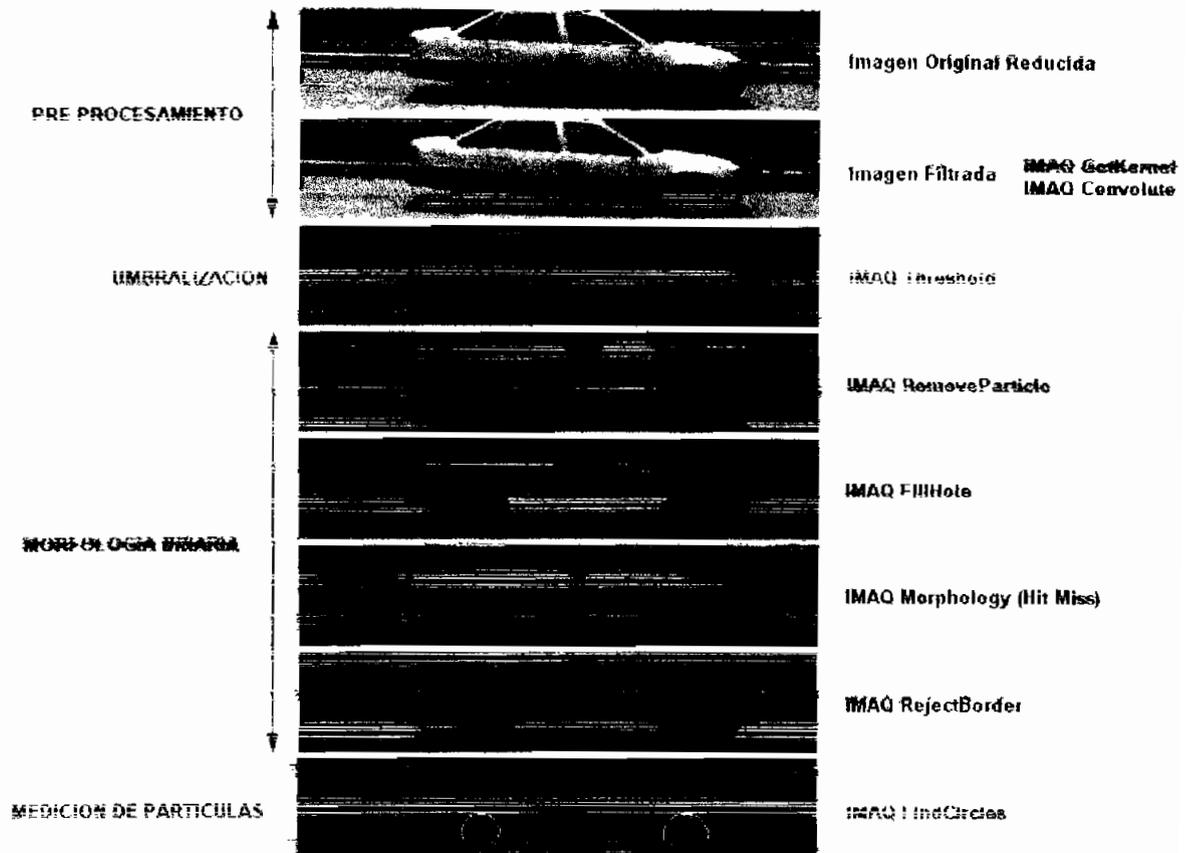


Figura 2.41 Procesamiento para los vehículos con aros oscuros

2.6.3 DISEÑO DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE FORMA

Este tipo de reconocimiento se fundamenta básicamente en poder aislar la forma de la carrocería de los vehículos, con esto se puede eliminar objetos de menores dimensiones como podrían ser personas, motocicletas, etc.

Para lograr aislar la carrocería del vehículo, se realiza restas de imágenes [3] y así eliminar detalles de fondo como son: vereda, pared, postes, árboles, en fin objetos estáticos que podrían distorsionar el análisis pretendido. Dicha resta se

realiza entre la imagen que requiere de procesamiento y una imagen del fondo, la cual es refrescada continuamente.

Como en el anterior algoritmo, primero se realiza un mejoramiento de la imagen antes de entrar al análisis de partículas sobre imágenes binarias.

Al final del procesamiento se obtendrá una sola partícula binaria que dependiendo de su longitud, se justificará si esta partícula corresponde o no a la forma del chasis del vehículo.

Como en todos los métodos de reconocimiento, el color del vehículo influye enormemente en la caracterización de parámetros, por ende y en base a los resultados se tomó en consideración lo señalado en la siguiente Tabla:

	Operación Aritmética
Para vehículos claros (ng mayor a 90)	<i>imagen – imagen de fondo (pared)</i>
Para vehículos oscuros (ng menor a 90)	<i>imagen de fondo (pared) - imagen</i>

Tabla 2.2 Operación Aritmética en base a la tonalidad del vehículo

Con lo antes mencionado, este algoritmo ejecutará las dos operaciones aritméticas indicadas en cascada, debido a que no se conoce dicha tonalidad. A continuación se explica más detalladamente este método aplicado a vehículos de color claro.

También con la ayuda del IMAQ Extract, se focaliza el análisis solamente en la parte media inferior de las imágenes con las siguientes coordenadas (0, 98, 352, 210). Como en el caso de la siguiente figura que ilustra la imagen de un vehículo y una de las imágenes de fondo (pared) usadas para este algoritmo.

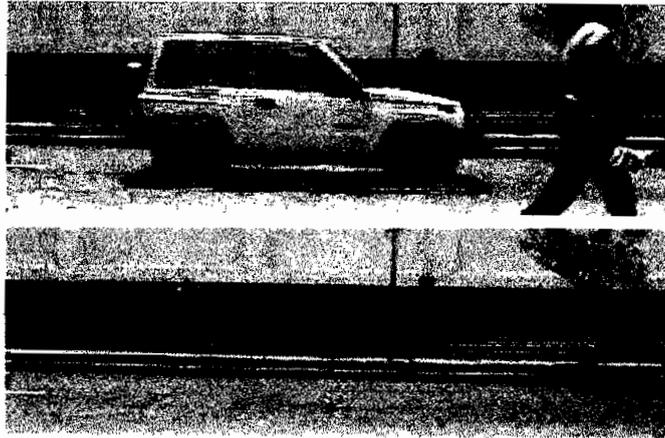


Figura 2.42 Focalización de análisis con el bloque IMAQ Extract

La operación de resta se realiza con la ayuda del siguiente bloque IMAQ:

IMAQ Subtract

Subtrae una imagen de otra o una constante de una imagen.

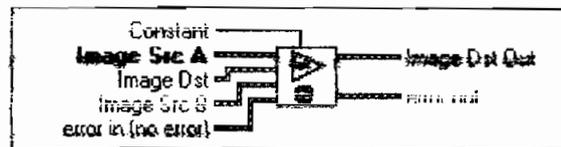


Figura 2.43 IMAQ Subtract

La siguiente Figura ilustra la imagen de salida luego de realizar dicha operación (imagen – pared) con el objetivo de aislar el vehículo del fondo de filmación.



Figura 2.44 Resultado de una resta de imágenes

Continuando con la etapa de pre-procesamiento se aplicará el aplanamiento de la imagen con un filtro (convolución) pasa bajos, para lograr resaltar más el chasis del vehículo y en este caso a la persona que asoma a un extremo del vehículo.

Esto se logra gracias a los bloques IMAQ GetKernel, eligiendo *Smoothing* en la entrada *Kernel Family*, el valor de 5 en *Kernel Size* y con el valor de 0 en *Kernel Number*, de esta manera, sus salidas corresponderán a las entradas del bloque IMAQ Convolute y así poder a la salida de éste, obtener la imagen filtrada que se ilustra en la Figura 2.45.



Figura 2.45 Imagen aplicada un filtro pasa bajos

Ahora en la etapa de umbralización, se utiliza igualmente el bloque IMAQ Threshold, con un *rango* de (20,200), el resultado de ésta se muestra la siguiente Figura:

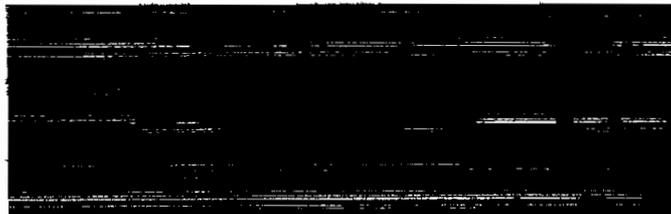


Figura 2.46 Umbralización

Empezando con la etapa de morfología binaria, se aplica el bloque IMAQ FillHole con el fin de rellenar los agujeros presentes dentro de la carrocería del vehículo, cuyo efecto se ilustra en la siguiente Figura:



Figura 2.47 Acción de relleno con IMAQ FillHole

Con el fin de aislar objetos que pertenecen a la imagen de fondo, se realiza la aplicación del bloque IMAQ Morphology, con una operación de *Erosión* que se realizará en un número de *3 iteraciones*.



Figura 2.48 Resultado de aplicar IMAQ Morphology con la operación de Erosión

IMAQ Separation

Separa partículas que se topan, particularmente pequeños istmos encontrados en ciertas partículas.

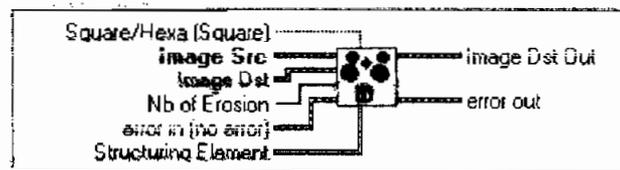


Figura 2.49 IMAQ Separation

Aplicando esta operación con *una iteración*, se logra conseguir también aislar ciertas partículas que pueden todavía pertenecer al fondo de la imagen, como se muestra en la imagen 2.50.



Figura 2.50 Resultado del bloque IMAQ Separation

Como consecuencia de ésta etapa se ejecuta la aplicación de los bloques IMAQ FillHole, IMAQ RejectBorder e IMAQ RemoveParticle (*con 7 erosiones*) para lograr tener en una sola partícula la forma del chasis del vehículo, como se indica en la Figura siguiente:

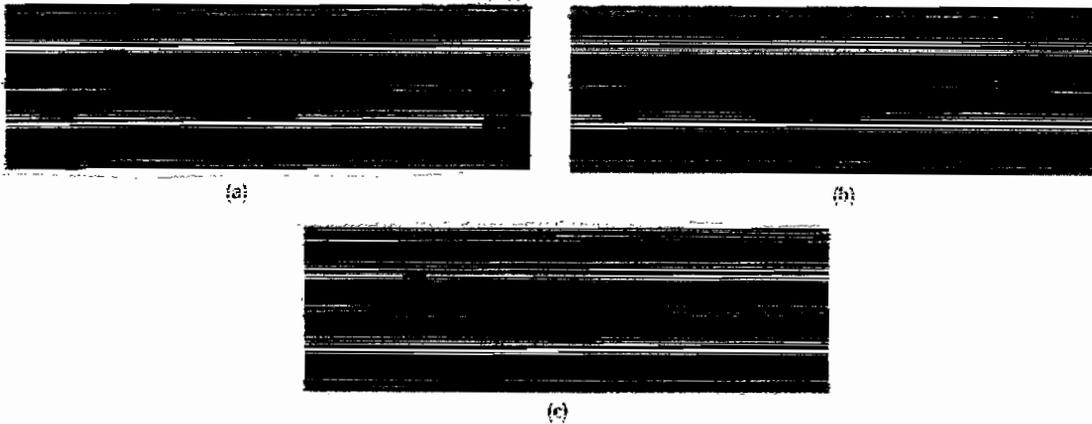


Figura 2.51 Aislamiento final de la forma del chasis del vehículo (a) Acción de IMAQ FillHole, (b) Acción de IMAQ RejectBorder, (c) Acción de IMAQ RemoveParticle.

Para la finalización de éste algoritmo, se ejecuta la medición horizontal de la única partícula resultante del análisis anterior, para lo cual se ayuda en la salida *Basic Reports* del siguiente bloque.

IMAQ BasicParticle

Detecta y mide partículas, así como también retorna el área y la posición de las mismas en una imagen binaria.

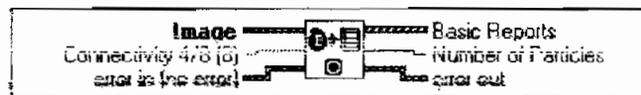


Figura 2.52 IMAQ BasicParticle

Basic Reports: proporciona información en lo relacionado a:

- *Area (pixels)*: indica el área de una partícula.
- *Area (calbrated)*: indica el área de una partícula en unidades definidas por el usuario.
- *Global Rectangle*: es un cluster que contiene las coordenadas de un rectángulo que rodea al objeto detectado en la imagen. Este cluster incluye los siguientes parámetros: *X1 left*, *Y1 top*, *X2 right*, *Y2 bottom*.

Con los parámetros *X2 right* y *X1 left* se identifica la longitud horizontal de la única partícula reconocida, por tanto se realiza una resta (en píxeles) de estos

parámetros, y si este valor es mayor que la *distancia mínima* (*parámetro calibrable -170-*), se concluye que dicha partícula pertenece a un vehículo.



Figura 2.53 Condición para considerar como vehículo

La segunda parte en la ejecución de este algoritmo (aislar la carrocería de vehículos con tonalidad oscura), se realiza en la misma región de búsqueda que en el caso anterior dada por el IMAQ Extract y con la ayuda de los siguientes bloques y parámetros para cada una de las etapas:

- IMAQ Extract: *Optional Rectangle (0, 98, 352, 210)*
- IMAQ Subtract (*pared – imagen*)
- IMAQ GetKernel: *Kernel Family, Smoothing*
Kernel Size 7
Kernel Numer 0
- IMAQ Convolute
- IMAQ Threshold: *Range (20, 200)*
- IMAQ RemoveParticle: *Conectivity 8 (default)*
Square/Hexa, Square (default)
Number of erosion 10
Low Pass/High Pass, Low (default)
- IMAQ FillHole: *Conectivity 8 (default)*
- IMAQ Morphology: *Square/Hexa, Square (default)*
Operation, Erode
Number of Iterations 6
- IMAQ Separation: *Square/Hexa, Square (default)*
Number of erosion 3

- *IMAQ RemoveParticle: Conectivity 8 (default)*
Square/Hexa, Square (default)
Number of erosion 7
Low Pass/High Pass, Low (default)

En la siguiente Figura se presenta la imagen original para este tipo de vehículos y cada una de las imágenes que se producen con la aplicación de cada uno de los bloques indicados anteriormente.

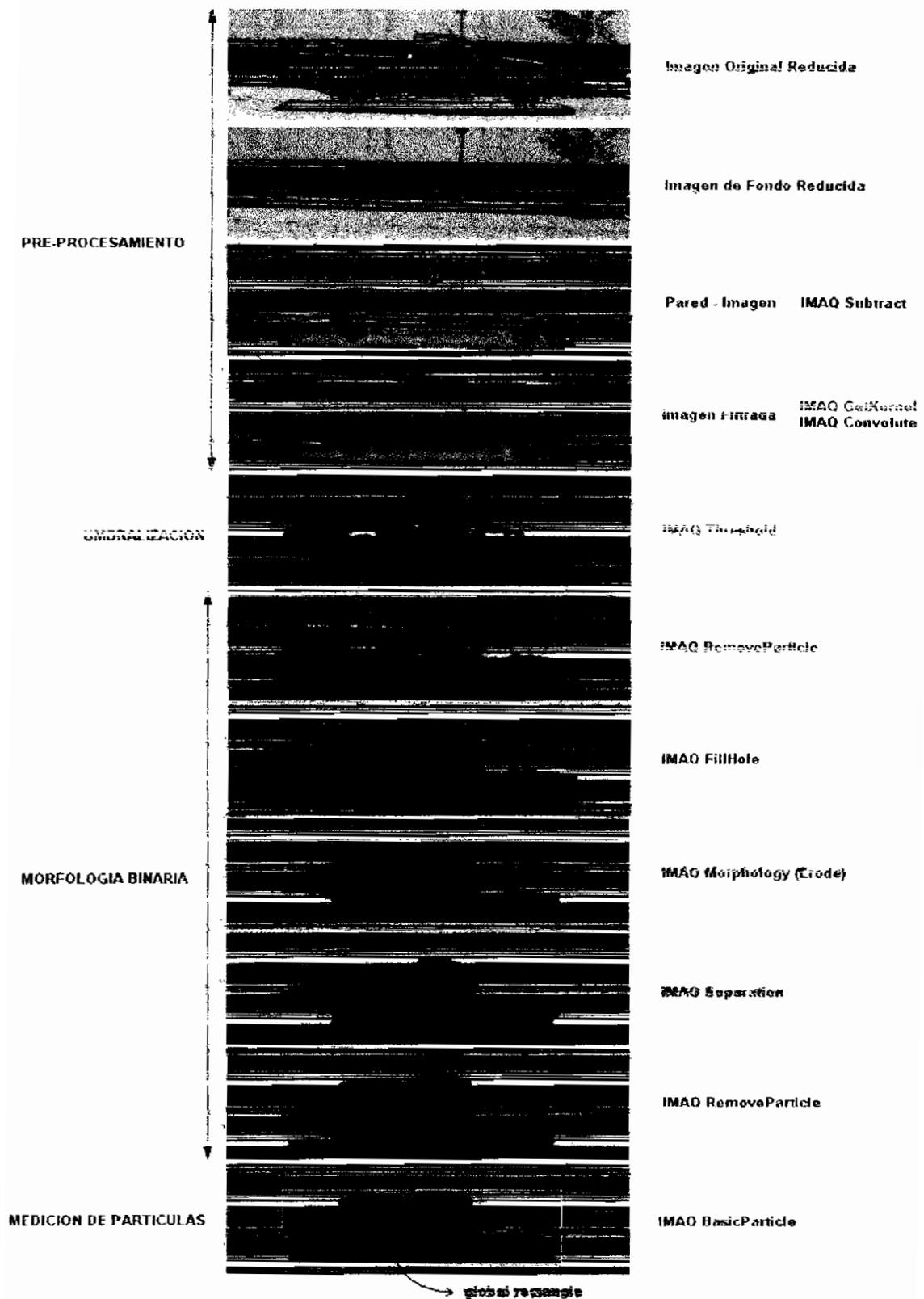


Figura 2.54 Aislamiento de carretera para vehículos de tonalidad oscura

Una vez explicado el diseño del programa principal (monitoreo) así como también de los algoritmos de reconocimiento vehicular, se procede en el siguiente capítulo con la implementación del software en el entorno gráfico de programación LabVIEW.

CAPITULO 3

IMPLEMENTACION Y DESARROLLO DEL PROGRAMA DE CONTROL

3.1 INTRODUCCIÓN

El presente capítulo consiste en la implementación misma del sistema de control de tráfico vehicular desarrollado completamente en el ambiente de programación gráfica de LabVIEW, utilizando los bloques de aplicación general, estructuras, así como también de los bloques de la librería IMAQ Vision, los cuales son exclusivos para aplicaciones que hagan uso del procesamiento digital de imágenes.

Se empezará por tanto a desarrollar los sub - programas para el reconocimiento de vehículos que atraviesan una calle (un sentido y una vía), para luego continuar con el desarrollo del programa principal que será el encargado de poder procesar imágenes estáticas en secuencia, con el fin de lograr reconocer (contar) los vehículos al mismo tiempo que son reproducidas en una ventana de visualización (monitoreo). Además dicho programa tendrá dos modos de operación: uno encargado del monitoreo y conteo de vehículos y otro de calibración, con el fin de ajustar al sistema a los distintos tipos de filmaciones que el mismo necesite procesar.

3.2 IMPLEMENTACION DEL ALGORITMO DE RECONOCIMIENTO EMPAREJAMIENTO DE MODELO

Este algoritmo de reconocimiento vehicular como se mencionó en el capítulo anterior, se trata de la búsqueda de modelos en la secuencia de imágenes.

Para la implementación del mismo y con el objetivo de crear un programa modular, se crearon dos subVIs: uno para el proceso de aprendizaje y otro para el de búsqueda. Los mismos que van a ser descritos más detalladamente a

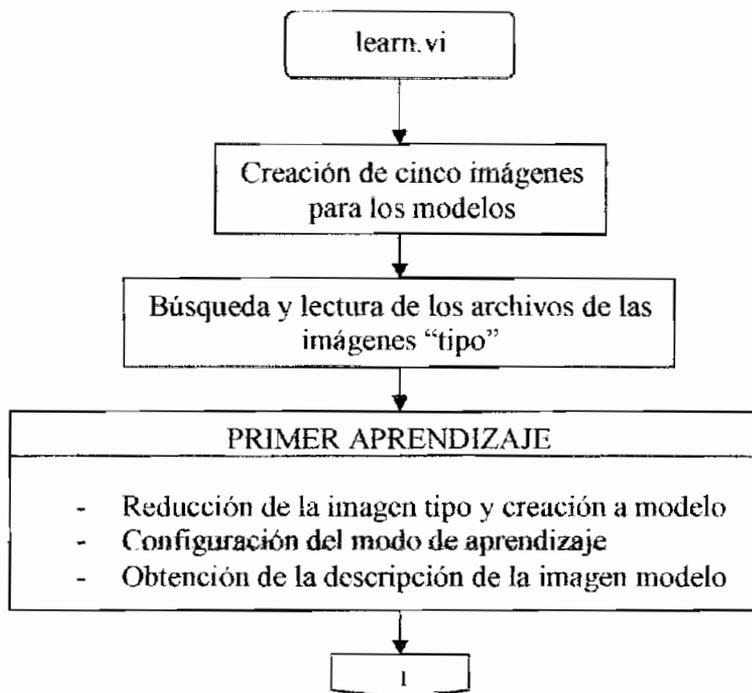
través de sus diagramas de flujo, así como también a partir del entorno de programación gráfica que ofrece el programa LabVIEW (diagrama de bloques), cuyos parámetros quedaron configurados en el capítulo anterior.

Otra de las razones para esta división, consiste en que el proceso de aprendizaje es necesario realizarlo una sola vez, en tanto que la búsqueda de emparejamiento se va realizando cada vez que exista una perturbación durante el procesamiento de la secuencia de imágenes.

3.2.1 PROCESO DE APRENDIZAJE

Este proceso se trata de crear una descripción lo más cercana posible de cada una de las imágenes que fueron seleccionadas como modelos, mismas que serán utilizadas durante el proceso de búsqueda.

El mencionado proceso corresponde al siguiente diagrama de flujo:



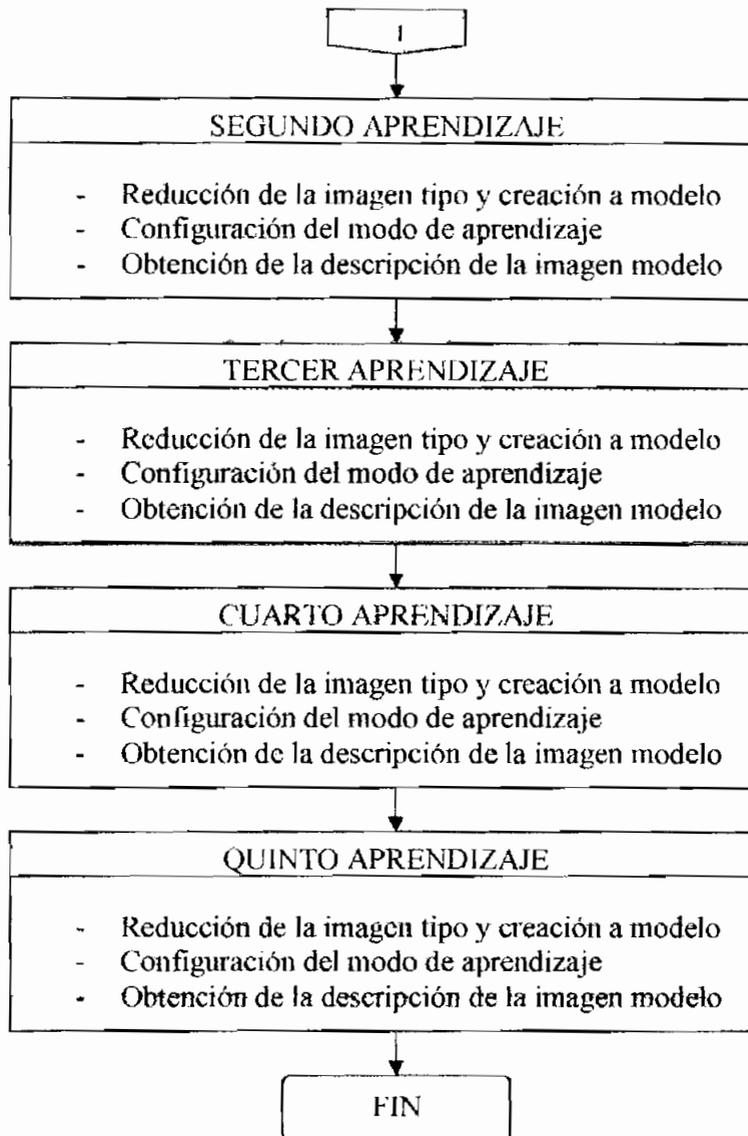


Figura 3.1 Diagrama de flujo del proceso de aprendizaje

Como primer paso en la ejecución del programa, se crean las imágenes con una definición a 8 bits; para que posteriormente sean los modelos a buscar, con la ayuda del siguiente bloque:

IMAQ Create

Crea una imagen, la cual puede ser de los siguientes tipos:

- *8 bits*: 8 bits por píxel (sin signo, monocromático estándar)
- *16 bits*: 16 bits por píxel (con signo)

- *Float*: 32 bits (punto flotante) por píxel
- *RGB*: 32 bits por píxel (color RGB)
- *HSL*: 32 bits por píxel (color HSL)
- *Complex*: 2×32 bits (punto flotante) por píxel (formato único después de aplicar una FFT)

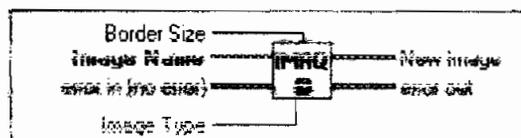


Figura 3.2 IMAQ Create

Por lo tanto el diagrama de bloques del subVI creado (learn.vi) empezará con la secuencia cero, de la siguiente manera:

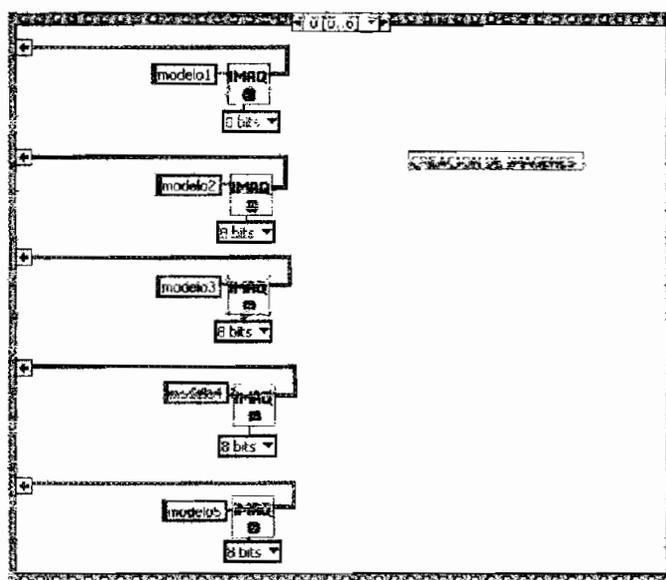


Figura 3.3 Primera pantalla del proceso de aprendizaje

Como siguiente paso se procede con la lectura de las imágenes tipo, las cuales se encuentran en la carpeta *modelos* dentro de la unidad *D* del disco, dicha captura del archivo de extensión *jpg*, se la realiza con el siguiente bloque:

IMAQ ReadFile

Lee un archivo de imagen, y este puede ser estándar (BMP, TIFF, JPEG, PNG, and AIPD) o no estándar.

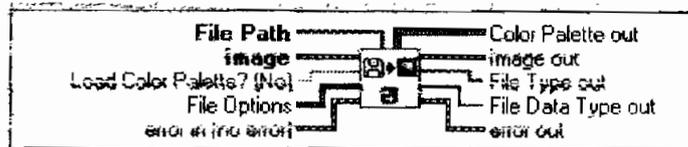


Figura 3.4 IMAQ ReadFile

En la secuencia uno del subVI, el *File Path* direccionará a la imagen "tipo" dentro de la carpeta *modelos* de la unidad *D*, tal como se ilustra en la siguiente Figura:

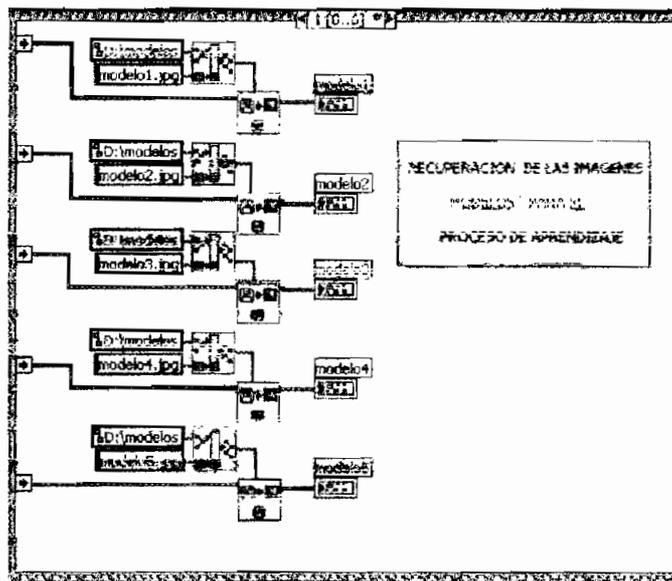


Figura 3.5 Segunda pantalla del proceso de aprendizaje

La Figura 3.6 ilustra el diagrama de bloques de la secuencia, en la cual se realiza el aprendizaje en sí del primer modelo, dicho diagrama consta de los siguientes pasos y bloques involucrados en los mismos:

- Reducción de la imagen y creación de modelo: MAQ Extract.
- Configuración del modo de aprendizaje: IMAQ Setup Learn Pattern
- Obtención de la descripción de la imagen modelo: IMAQ Learn Pattern

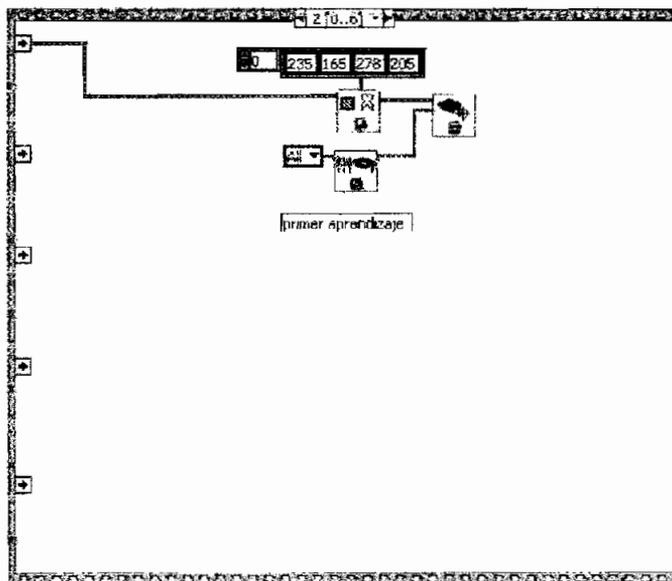


Figura 3.6 Primer aprendizaje

Los siguientes aprendizajes de los modelos 2, 3, 4 y 5 son realizados en cuadros independientes del *sequence structure*, de manera semejante a la Figura anterior, lo único que varía son las coordenadas para la extracción de una parte de la imagen "tipo", para ser considerada como modelo de aprendizaje, como se muestra en la siguiente Tabla.

BLOQUES \ MODELOS	1	2	3	4	5	Config
IMAQ Create						
Image Type	8 bits	No				
IMAQ Read File						
File Path	D:\modelos\modelo*.jpg					
IMAQ Extract						
Optional Rectangle	(235,165) (278,205)	(195,163) (238,208)	(208,158) (258,198)	(228,168) (288,208)	(220,170) (250,203)	No
IMAQ Setup Learn Pattern						
Learn Mode	All	All	All	All	All	No
Resolución de los modelos (píxeles)						
	(43 x 40)	(43 x 45)	(50 x 40)	(60 x 40)	(30 x 33)	

Tabla 3.1 Parámetros para el proceso de aprendizaje

De tal forma los diagramas explicados anteriormente nos permiten crear el subVI, cuyo icono se ilustra en la siguiente Figura.

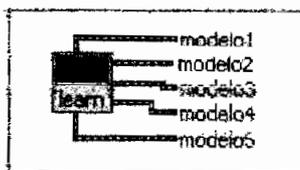


Figura 3.7 learn.vi

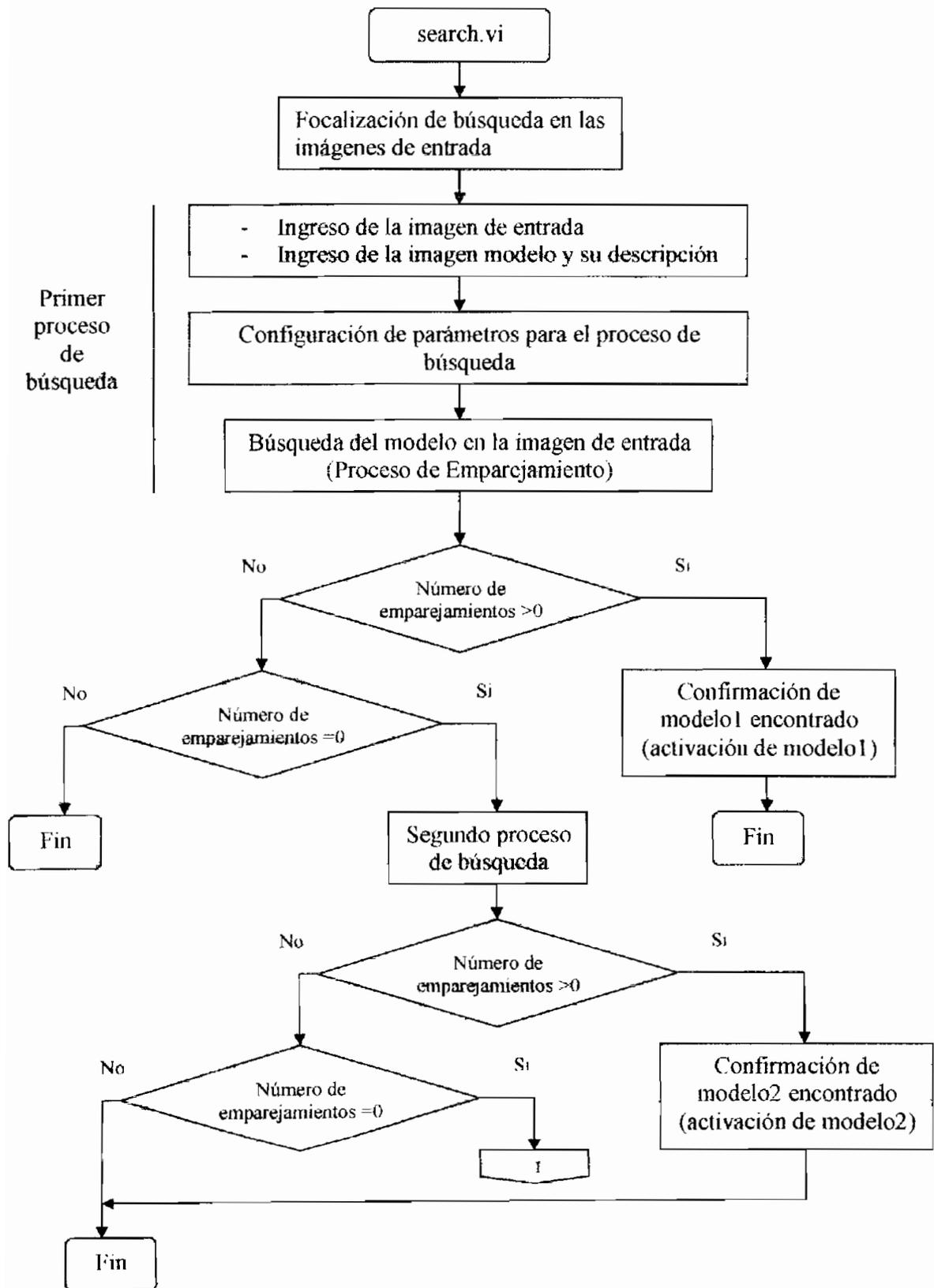
Las cinco salidas que se aprecian, corresponden a los modelos que serán usados en el siguiente proceso. Una vez ejecutando dicho subVI se logra obtener la descripción de cada uno de ellos, como se observa en la siguiente Tabla.

learn.vi			
		Descripción	Tipo
Salidas	<i>modelo1</i>	Imagen y descripción de cada una de las imágenes extraídas a partir de las imágenes "tipo".	Cluster (2 elementos): - String - Unsigned long 32 bit integer
	<i>modelo2</i>		
	<i>modelo3</i>		
	<i>modelo4</i>		
	<i>modelo5</i>		

Tabla 3.2 Descripción de las salidas del subVI learn.vi

3.2.2 PROCESO DE BUSQUEDA

En esta parte del algoritmo se realiza el verdadero objetivo, que consiste en la búsqueda de los cinco modelos aprendidos en el proceso anterior. La implementación de dicho emparejamiento se realiza en base al siguiente diagrama de flujo:



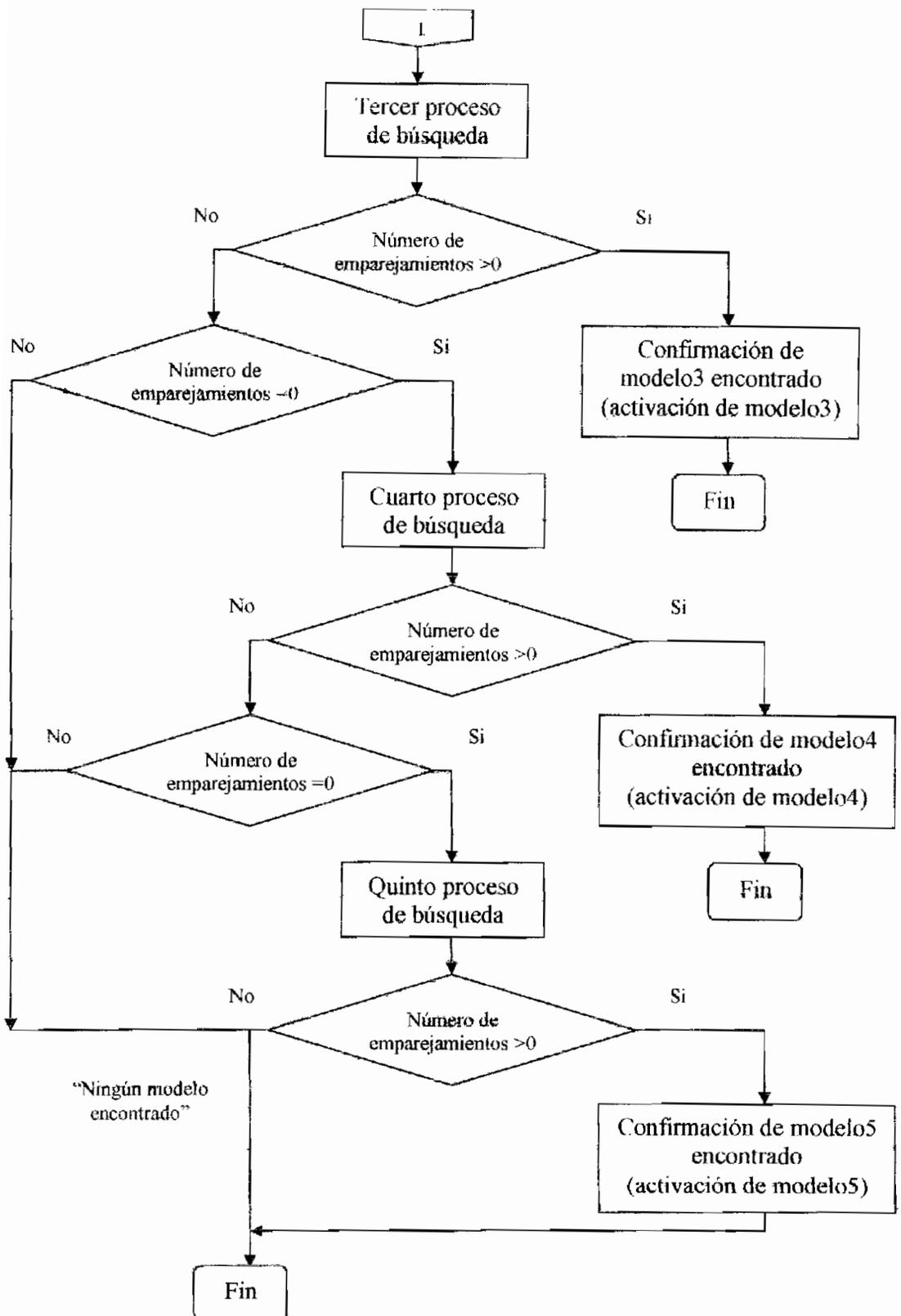


Figura 3.8 Diagrama de flujo del proceso de búsqueda

Con la ayuda de la programación gráfica de LabVIEW, y como se indica en la Figura 3.9, el proceso se inicia con la extracción de la parte inferior de las imágenes de entrada con el objetivo de aumentar la velocidad de ejecución del mismo y disminuir la posibilidad de obtener resultados erróneos.

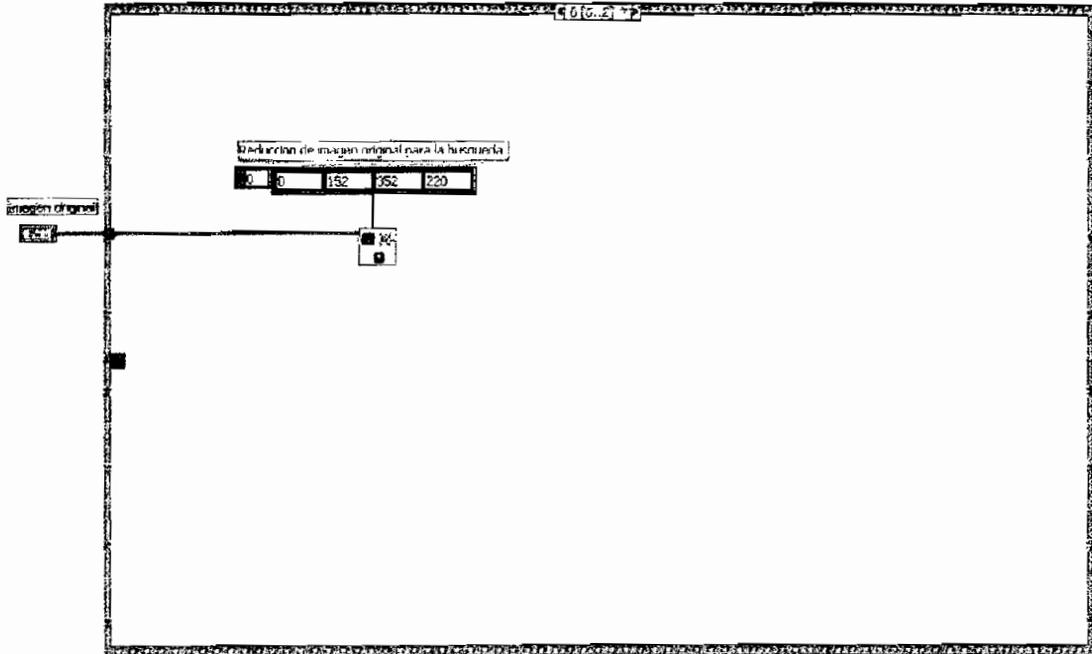


Figura 3.9 Focalización de búsqueda en las imágenes de entrada

La próxima secuencia de ejecución del proceso se observa en la siguiente Figura, en el cual se calibran parámetros (IMAQ Setup Match Pattern) y se empieza con la búsqueda de dicho modelo sobre la imagen de entrada (IMAQ Match Pattern), además se cuenta también con la posibilidad de sobrescribir en la imagen de entrada, los objetivos que se pudieron haber encontrado (Draw Matches Position.vi).

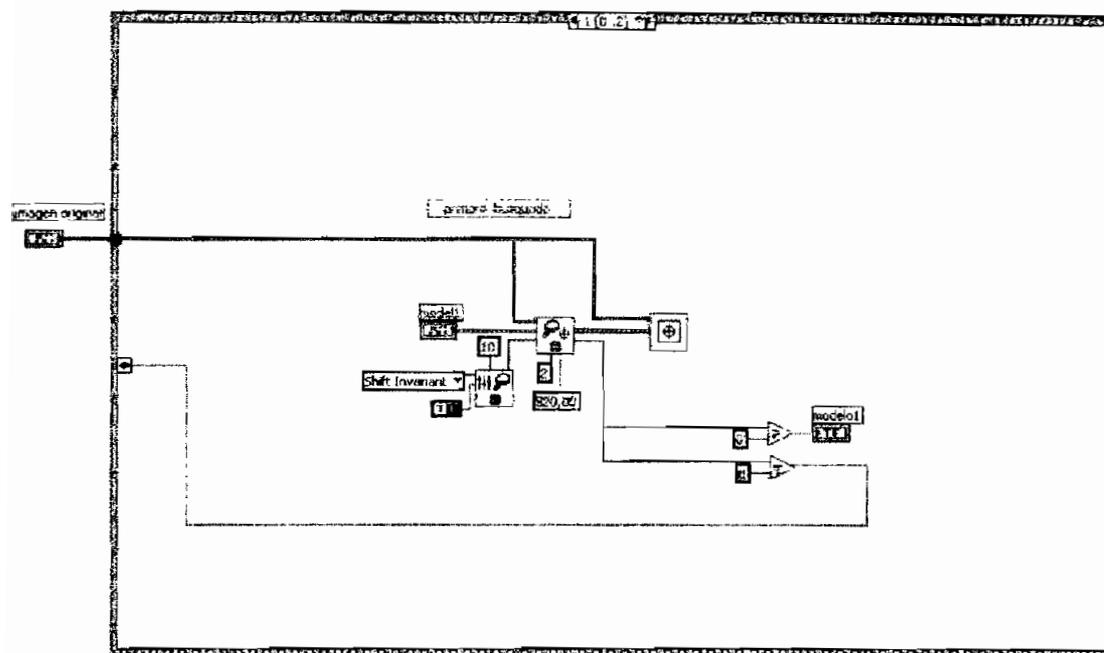


Figura 3.10 Primer proceso de búsqueda

La salida del bloque de búsqueda *Number of Matches* es usada para la confirmación de que la búsqueda del primer modelo ha sido exitosa (determinación de vehículo y salida del subVI), tal como se ilustra en la siguiente Figura.

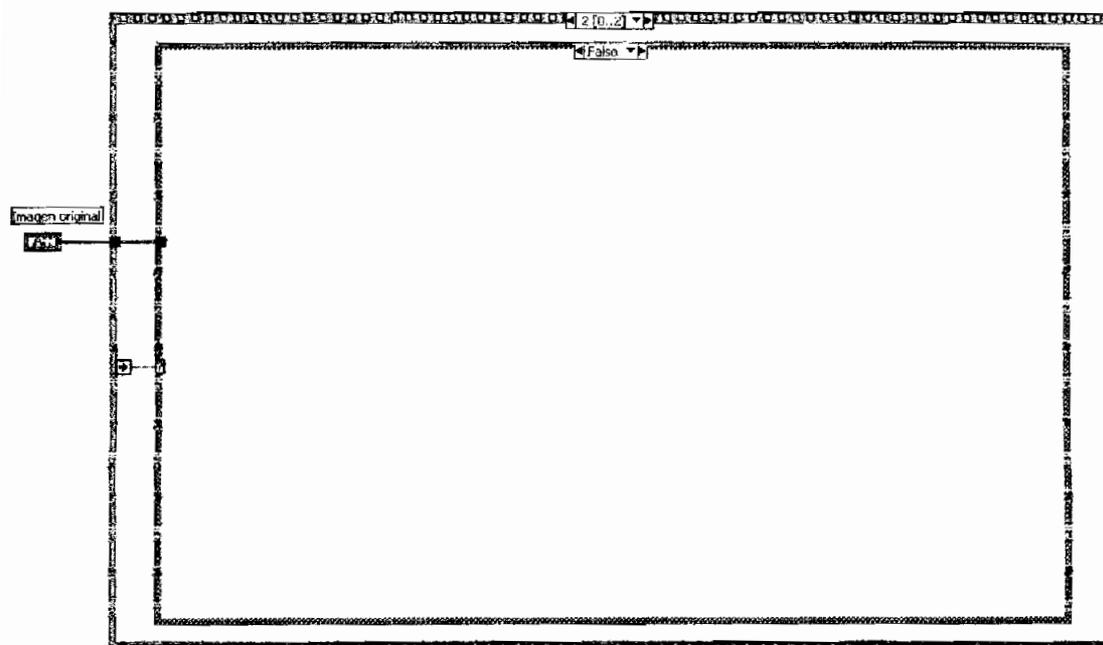


Figura 3.11 Confirmación del primer proceso y salida del subVI.

En el caso de que la salida antes mencionada sea nula (búsqueda del primer modelo fallida), la condición verdadera del *case structure* es activada, en el cual se procede con la configuración y búsqueda del siguiente modelo.

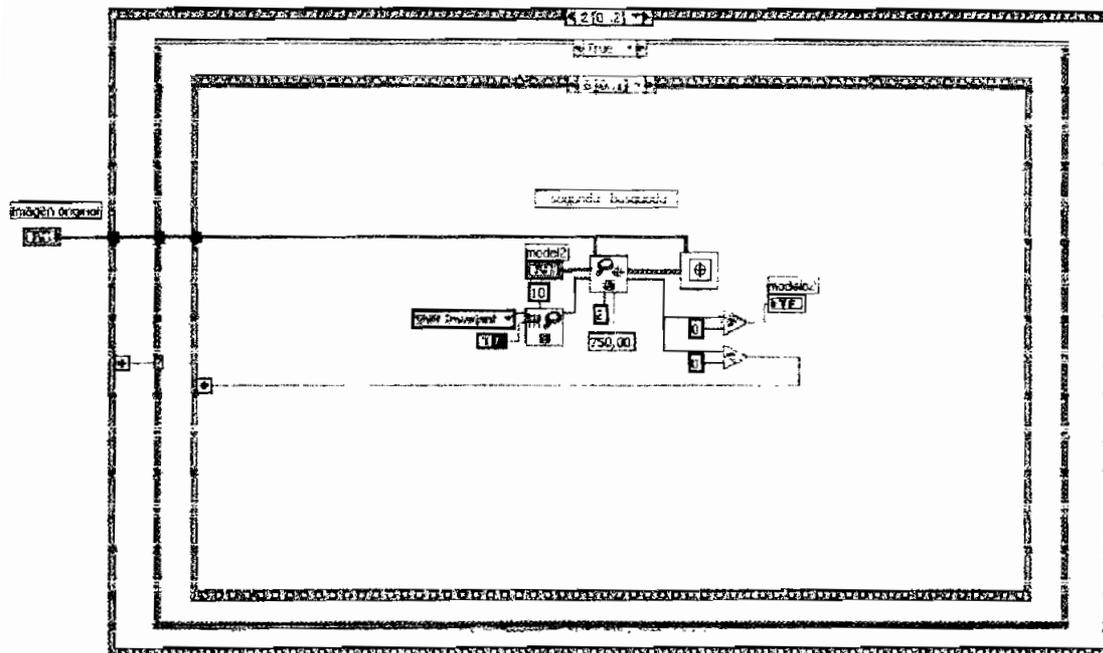


Figura 3.12 Segundo proceso de búsqueda

El método de aplicar la búsqueda de los siguientes modelos se la realiza en forma similar al procedimiento indicado anteriormente; de tal forma, el paso para el último proceso (cuatro anteriores fallidos) se indica en la siguiente Figura 3.13, luego del cual se sale del subVI ya sea si éste último proceso sea satisfactorio o no.

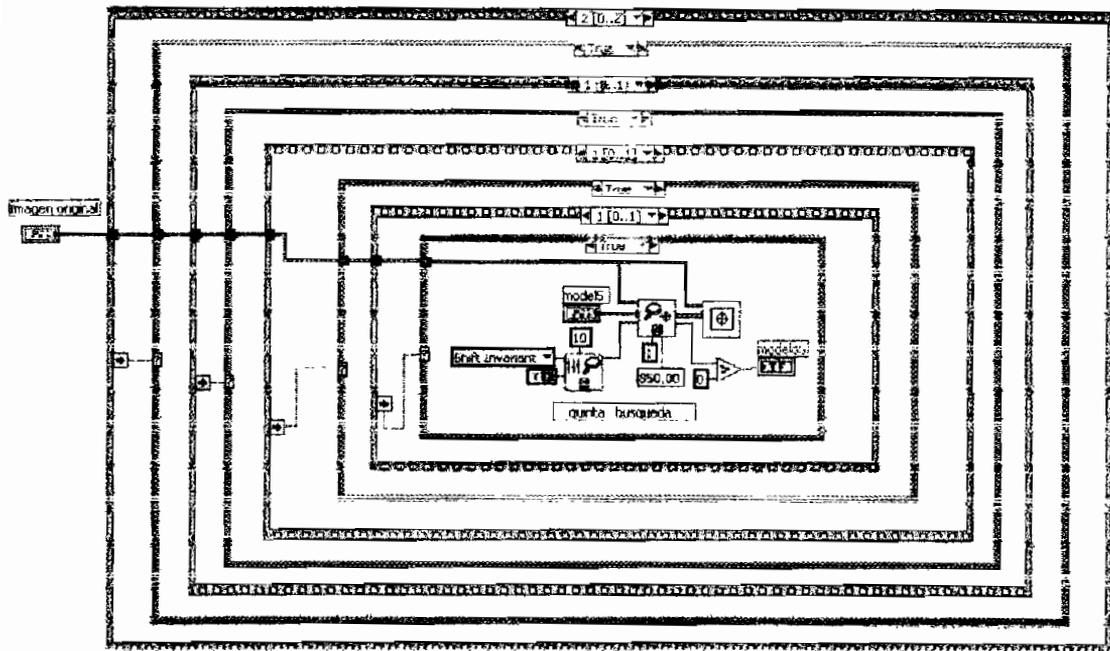


Figura 3.13 Último proceso de búsqueda y salida del subVI

La Tabla 3.3 ilustra todos los parámetros que quedaron configurados dentro de éste subVI, así como también los bloques involucrados en el mismo.

MODELOS	1	2	3	4	5	Config
BLOQUES						
IMAQ Extract	Extracción de la imagen de entrada en la zona de interés con las coordenadas (0, 152, 352, 220)					
IMAQ Setup Match Pattern						
<i>Minimum Contrast</i>	10	10	10	10	10	No
<i>Match Mode</i>	Shift	Shift	Shift	Shift	Shift	No
	Invariant	Invariant	Invariant	Invariant	Invariant	
<i>Subpixel Accuracy</i>	False	False	False	False	False	No
IMAQ Match Pattern						
<i>Template Image</i>	Modelo1	Modelo2	Modelo3	Modelo4	Modelo5	
<i>Number of Matches Requested</i>	2	2	2	1	1	No
<i>Minimum Match Score /1000</i>	800	750	700	730	850	No

Tabla 3.3 Parámetros para el proceso de búsqueda

De lo anterior explicado, se diseñó el siguiente subVi cuyas entradas y salidas son detalladas en la Tabla 3.4. En el programa principal dicho subVi será visualizado como se ilustra en la siguiente Figura:

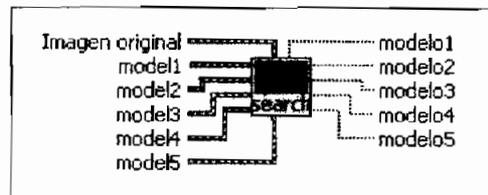


Figura 3.14 search.vi

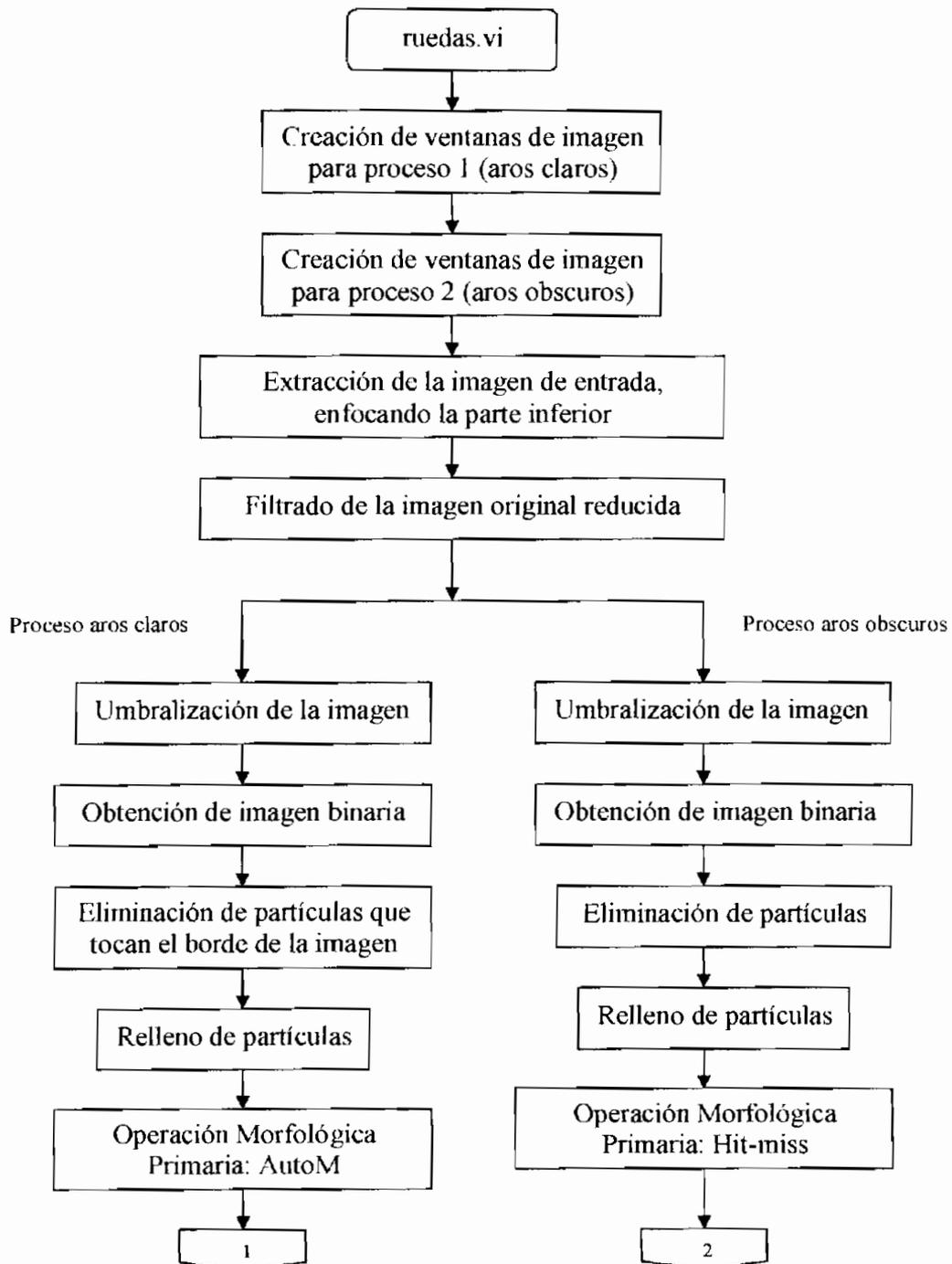
search.vi			
		Descripción	Tipo
Entradas	Imagen Original	Imagen en la cual va a realizarse el proceso de búsqueda	Cluster (2 elementos):
	<i>modelo1</i>	Imagen y descripción de cada una de las imágenes extraídas a partir de las imágenes "tipo". Estas entradas son unidas con las salidas del bloque learn.vi	- String
	<i>modelo2</i>		- Unsigned long
	<i>modelo3</i>		32 bit integer
	<i>modelo4</i>		
<i>modelo5</i>			
Salidas	<i>modelo1</i>	Son confirmaciones del proceso de búsqueda de acuerdo al número de modelo encontrado.	Booleana
	<i>modelo2</i>		Booleana
	<i>modelo3</i>		Booleana
	<i>modelo4</i>		Booleana
	<i>modelo5</i>		Booleana

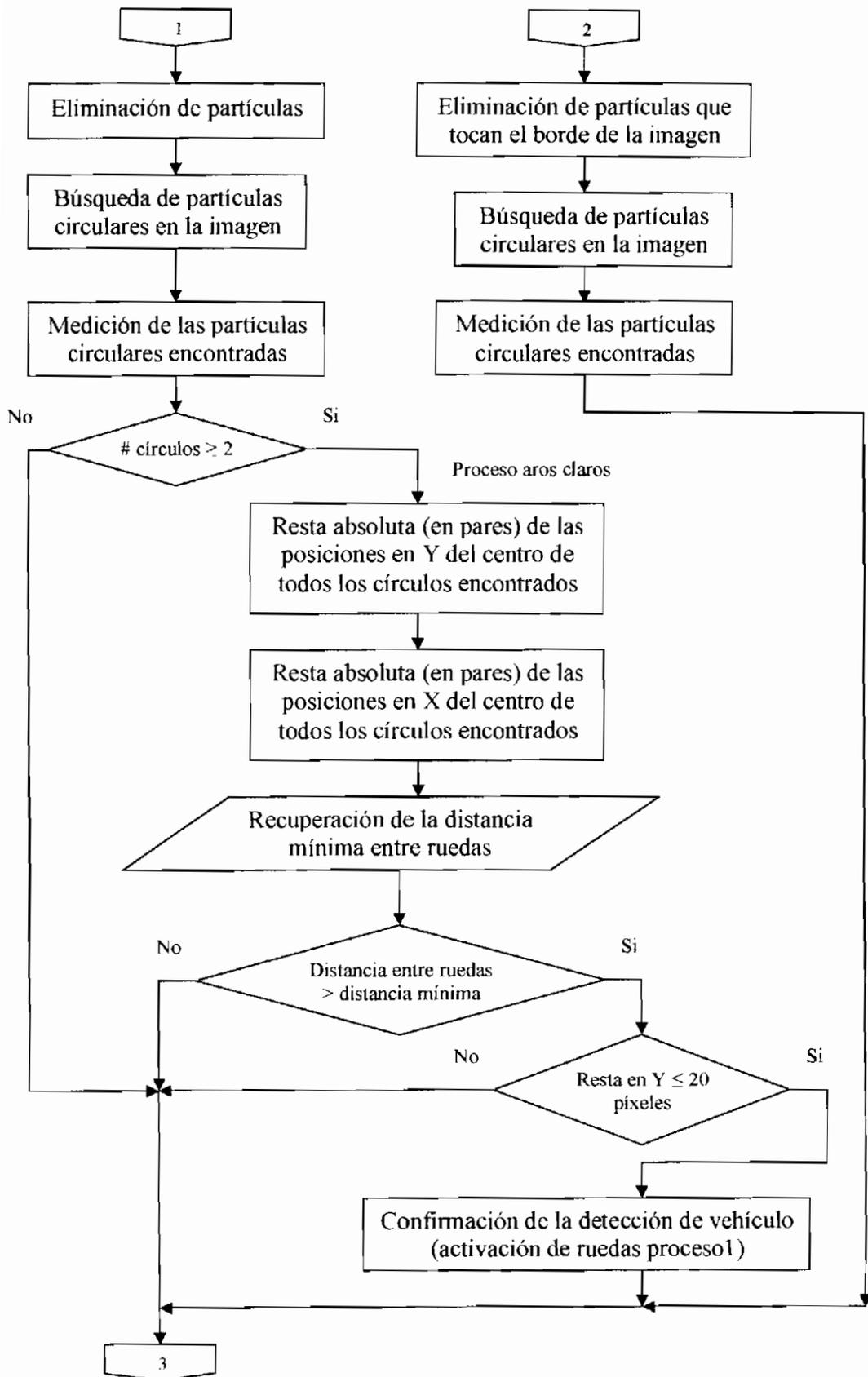
Tabla 3.4 Descripción de entradas y salidas del subVi search.vi

3.3 IMPLEMENTACION DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE RUEDAS

Como se explicó en el capítulo anterior, el presente algoritmo de reconocimiento vehicular consiste en procesar las imágenes utilizando básicamente morfología binaria de partículas, para de esta forma lograr aislar las dos ruedas del vehículo y también poder medir la distancia entre dichos círculos, en el caso que ésta medición sea la adecuada se podrá concluir que la imagen procesada corresponde a la de un vehículo.

Con el fin de que este algoritmo sea eficaz, se desarrollaron dos tipos de procesamientos en paralelo para cubrir con la diversidad en el color del aro de los vehículos. El siguiente diagrama de flujo indica el procedimiento a seguir para lograr con el objetivo de éste algoritmo.





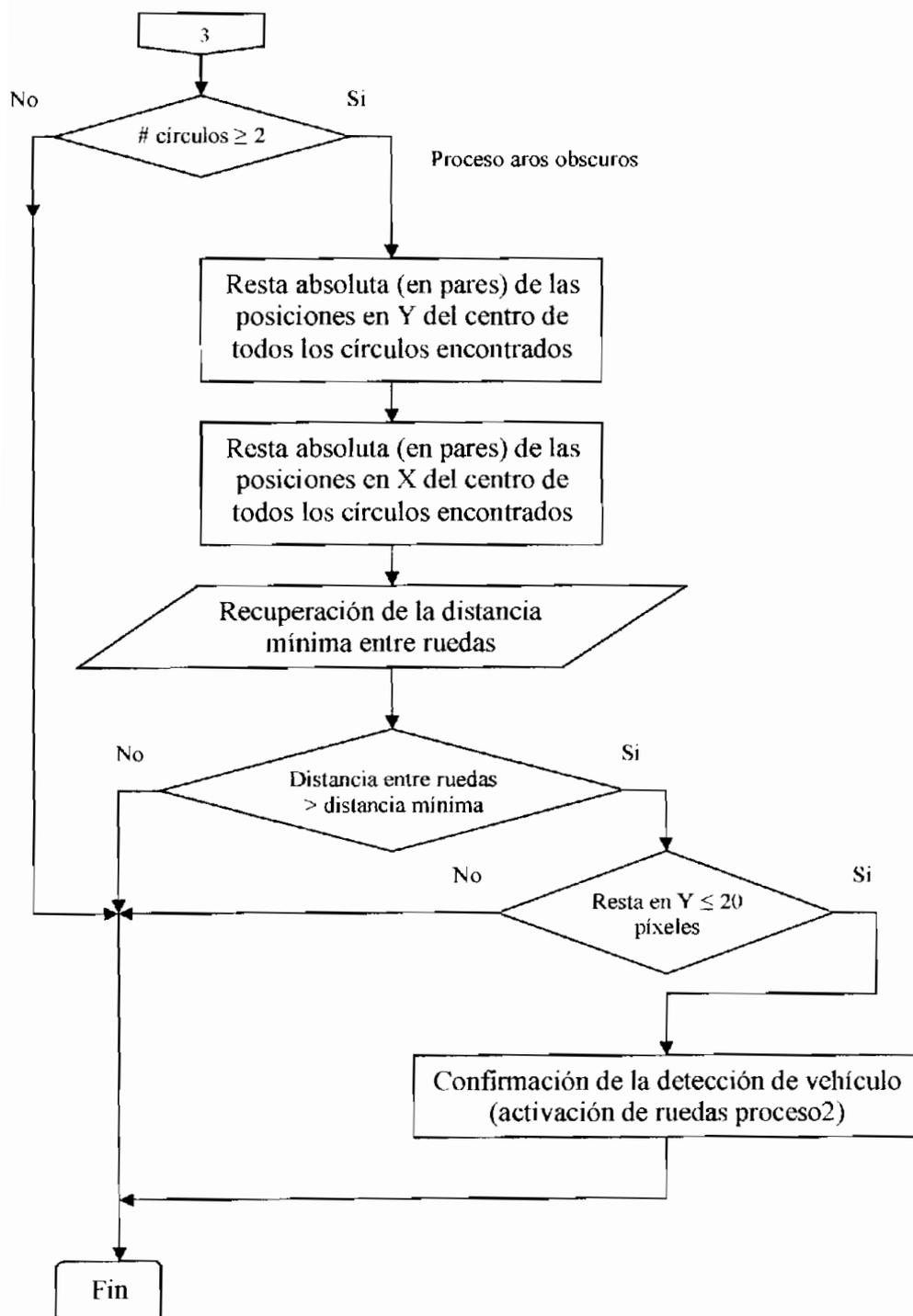


Figura 3.15 Diagrama de flujo para el proceso de detección de ruedas

El diagrama de bloques para éste tipo de reconocimiento, empezará con la secuencia cero, creando las imágenes que servirán en los dos procesos (aros claros y aros oscuros), focalizando también la region inferior de las imágenes para la detección de las dos ruedas, y por último en esta misma secuencia se realiza el filtrado (etapa de pre-procesamiento) de dicha imagen reducida, tal como se ilustra en la siguiente Figura:

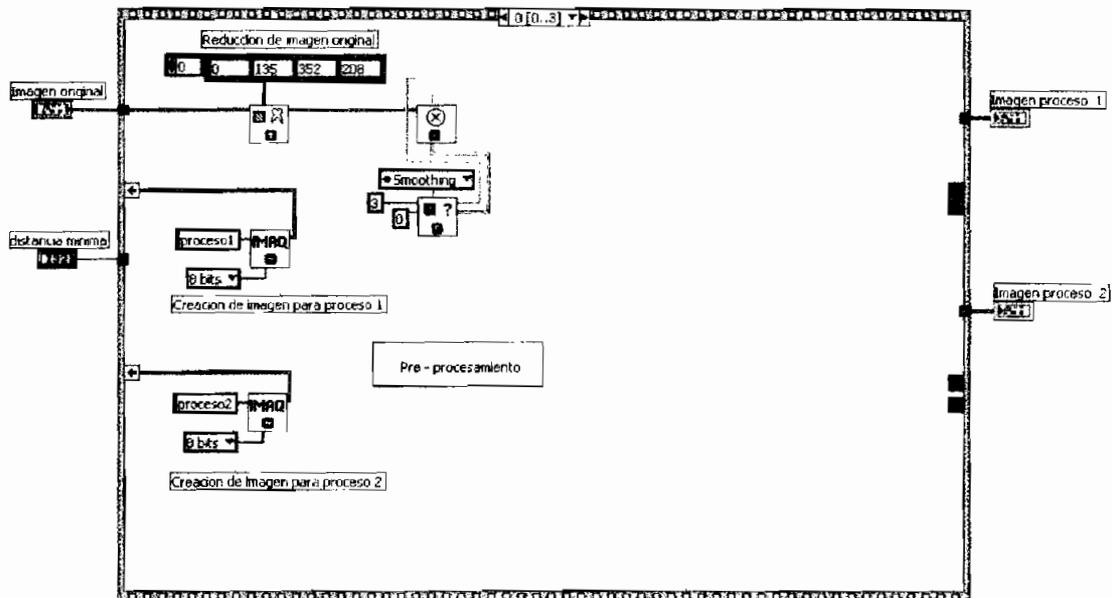


Figura 3.16 Primera pantalla del proceso de identificación de ruedas

A continuación se realiza la ejecución de los dos procesos en forma paralela, tal como se indica en la Figura 3.17, en donde cada uno de ellos es pasado por el siguiente procedimiento (análisis de partículas):

- Umbralización: IMAQ Threshold
- Morfología Binaria: IMAQ RejectBorder, IMAQ FillHole, IMAQ Morphology, IMAQ RemoveParticle (el orden depende del tipo de proceso)
- Medición de Partículas: IMAQ Find Circles

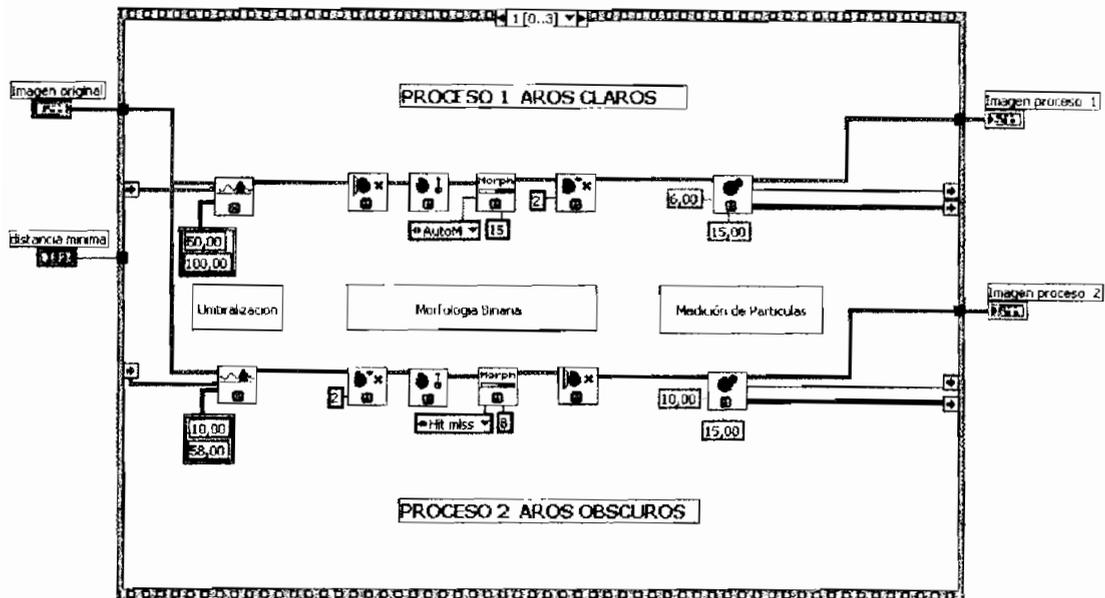


Figura 3.17 Análisis de partículas para el reconocimiento de las ruedas

Las siguiente Figura ilustra el diagrama de bloques donde a partir de la salida *Circles Data* del bloque IMAQ Find Circles, se realiza una medición de las posiciones en X y Y de los centros de todos los círculos encontrados en la secuencia anterior (si el número de círculos es mayor a dos) con el fin de discriminar los círculos que no se encuentran al mismo nivel y también para medir la distancia horizontal de dos círculos y si esta es la adecuada, se puede concluir que la detección de las ruedas (aros claros) es satisfactoria: por lo tanto se consigue la activación de *ruedas proceso1*.

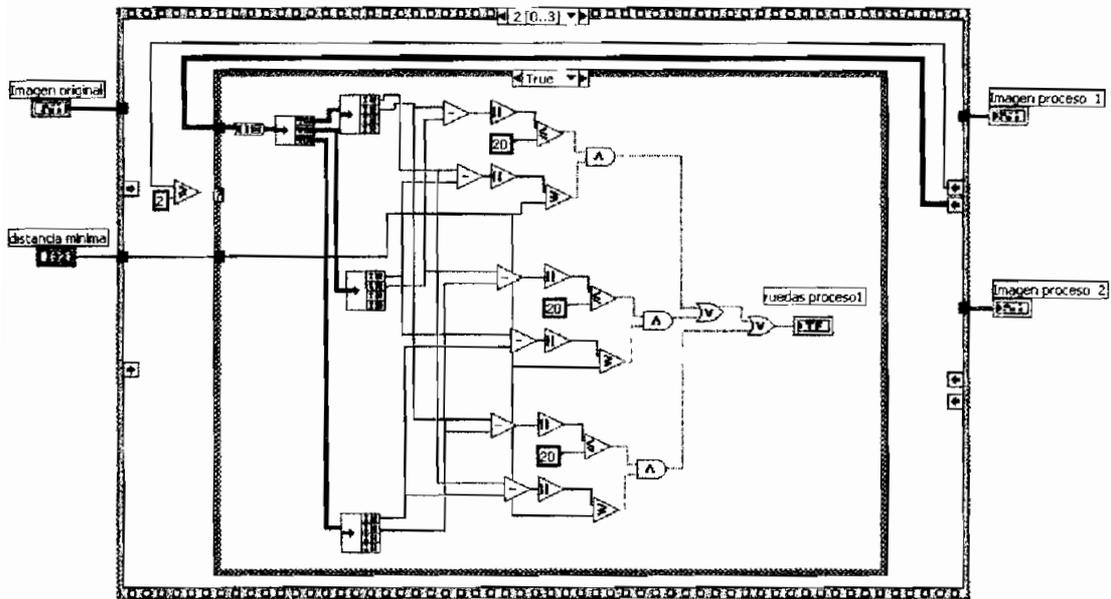


Figura 3.18 Medición de la distancia entre ruedas para el proceso de aros claros

En el proceso de vehículos cuyos aros son oscuros, dicho procedimiento de medición se la realiza de igual manera que el proceso anterior, la única diferencia consiste que para éste se logra la activación de la salida *ruedas proceso2*.

La diferenciación de cada proceso consiste en el orden de ejecución de los bloques, así como también de los parámetros que entran al análisis de partículas tal como se ilustra en la siguiente Tabla:

Bloques generales de ejecución				Config	
IMAQ Create					
Image Type	8 bits			No	
IMAQ Extract					
Optional Rectangle	(0, 135, 352, 208)			No	
IMAQ GetKernel					
Kernel Family	Smoothing			No	
Kernel Size	3			No	
Kernel Number	0			No	
IMAQ Convolute					
BLOQUES	PROCESO	Proceso1 (aros claros)	Proceso 2 (aros oscuros)	Orden de ejecución	Config
				Proceso1 Proceso2	
IMAQ Threshold				1 1	
Range		(50, 100)	(10, 58)		No
IMAQ RejectBorder				2 5	
IMAQ FillHole				3 3	
IMAQ Morphology				4 4	
Operation		AutoM	Hit – miss		No
Number of iterations		15	8		No
IMAQ Remove Particle				5 2	
Number of erosion		2	2		No
IMAQ Find Circles				6 6	
Min Radius (píxeles)		6	10		No
Max Radius (píxeles)		15	15		No

Tabla 3.5 Parámetros para el proceso detección de ruedas

De tal forma los diagramas explicados anteriormente permiten crear el subVI (ruedas.vi), cuyo icono se ilustra en la siguiente Figura.

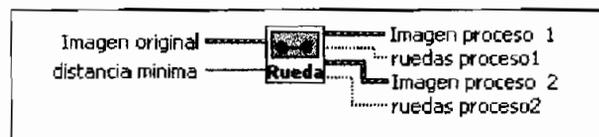


Figura 3.19 ruedas.vi

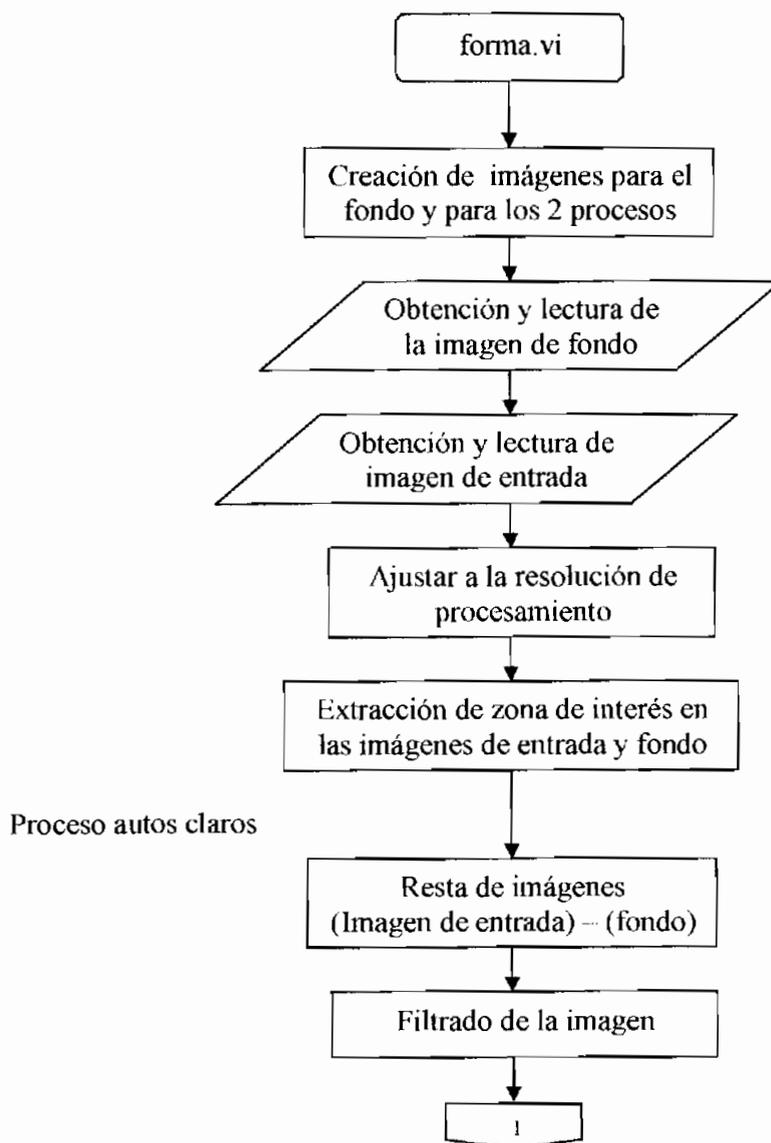
En la siguiente Tabla se señala la descripción y el tipo de las entradas y de las salidas que forman parte del subVI diseñado para éste algoritmo de reconocimiento vehicular.

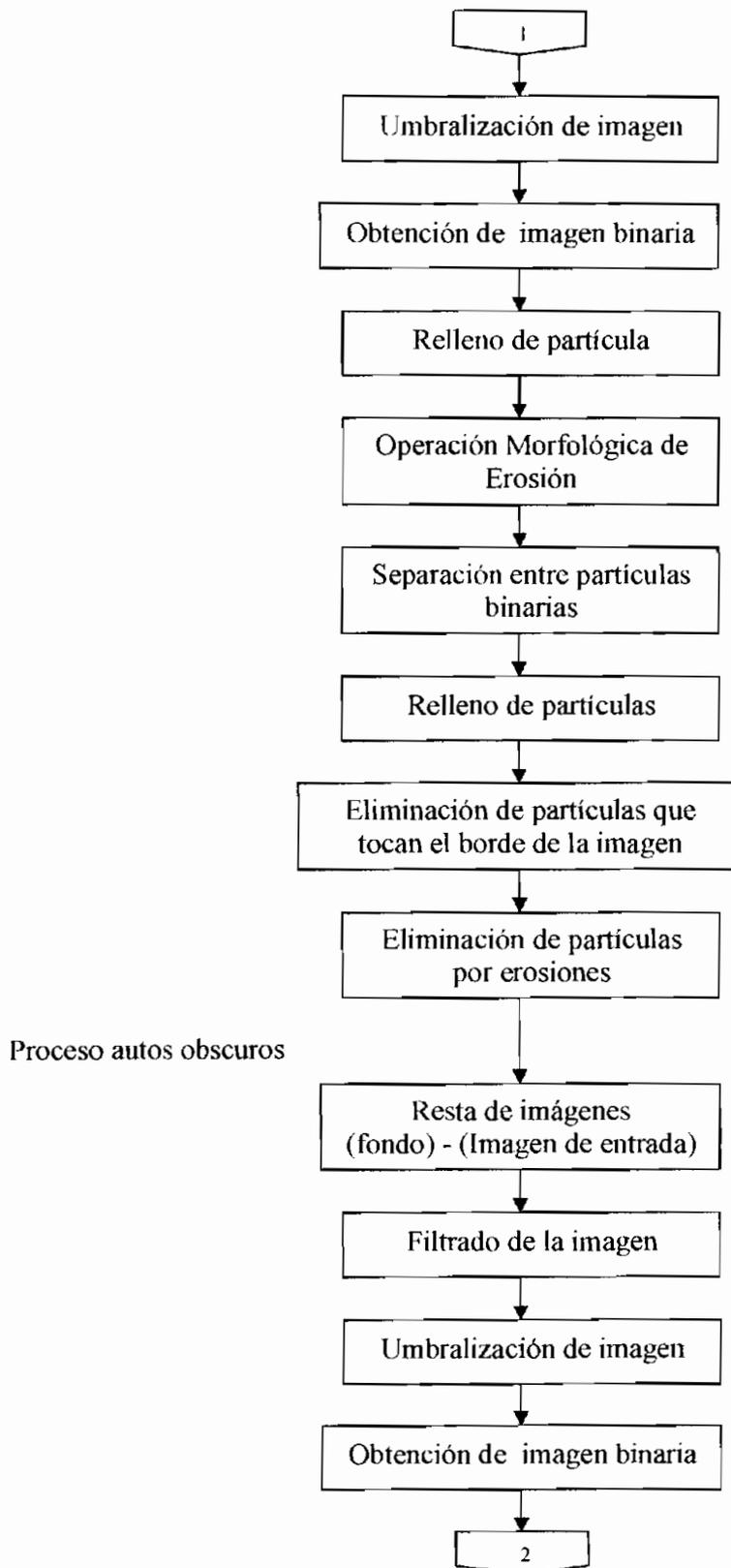
ruedas.vi				
		Descripción	Tipo	Config
Entradas	Imagen Original	Imagen en la cual va a realizarse el proceso de detección de ruedas	Cluster (2elementos): -String -Unsigned long 32 bit integer	No
	<i>Distancia mínima</i>	Es la distancia en píxeles que tienen que estar separados los centros de los círculos detectados para poder ser considerados como vehículos.	Constante (long 32 bit integer)	Si
Salidas	<i>Imagen proceso 1</i>	Es la imagen extraída en la cual se puede observar los círculos que detecta el proceso para aros claros	Cluster (2 elementos)	
	<i>ruedas proceso1</i>	Es la confirmación de detección de vehículo usando el proceso para aros claros	Booleana	
	<i>Imagen proceso 2</i>	Es la imagen extraída en la cual se puede observar los círculos que detecta el proceso para aros oscuros	Cluster (2 elementos)	
	<i>ruedas proceso2</i>	Es la confirmación de detección de vehículo usando el proceso para aros oscuros	Booleana	

Tabla 3.6 Descripción de entradas y salidas del subVI search.vi

3.4 IMPLEMENTACION DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE FORMA

La utilización de este algoritmo para el reconocimiento de vehículos se fundamenta en aislar el chasis, por consiguiente eliminando el resto de detalles presentes en la imagen. Se realizan dos procesos en serie uno para autos de color claro y otro para autos oscuros con el propósito de hacer más eficiente este método, además facilitando la etapa de umbralización presente en el mismo. Para explicar su implementación se describe el siguiente diagrama de flujo:





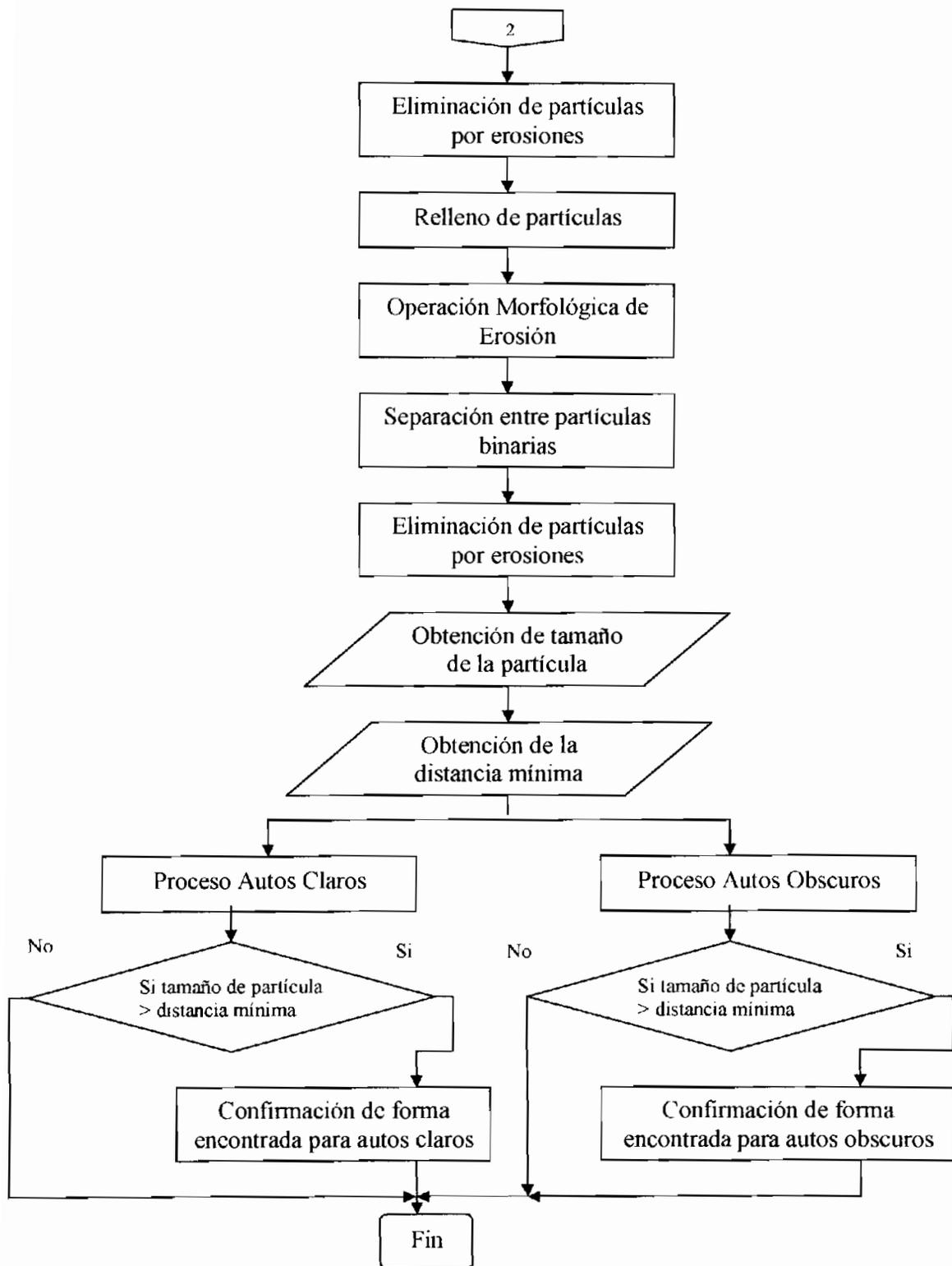


Figura 3.20 Diagrama de flujo para el proceso de detección de forma

El inicio de este subVI "forma.vi" como primer paso en la ejecución del programa, se crean las imágenes de fondo y para los procesos con una definición a 8 bits como se muestra en la Figura 3.21, las que servirán para que la imagen de entrada no sufra cambios en los procesos involucrados en este algoritmo.

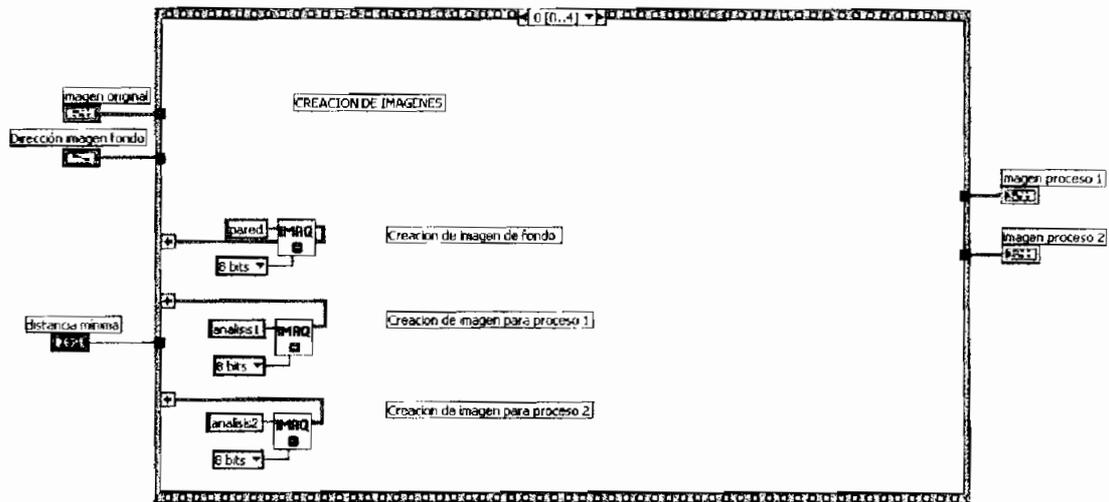


Figura 3.21 Creación de imágenes de fondo y para los procesos

La siguiente secuencia de programación se indica en la Figura 3.22 donde se procede con la obtención de la dirección, lectura de la imagen que servirá de fondo y con la focalización de búsqueda del chasis en la parte inferior de las imágenes. Cabe señalar que la resolución de las imágenes que entran al procesamiento es única (352 x240).

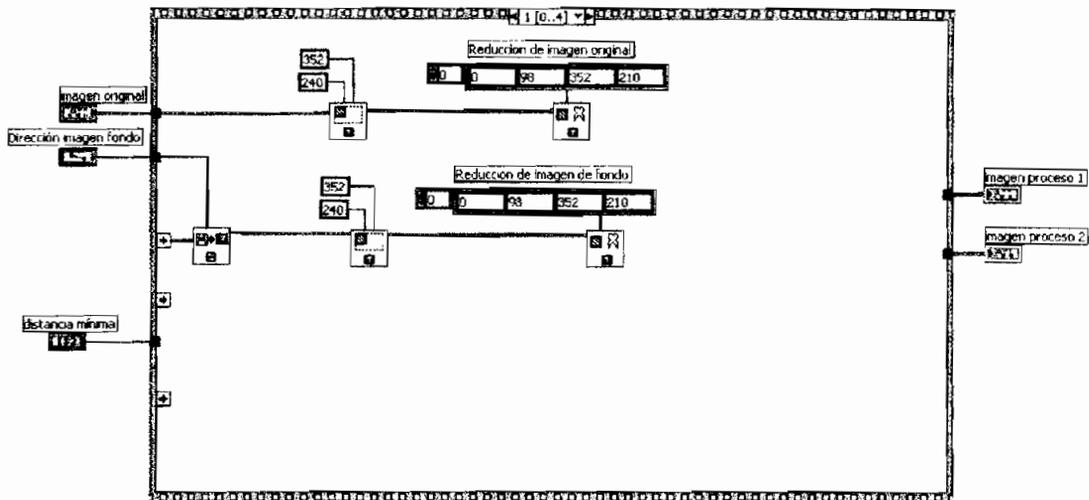


Figura 3.22 Obtención y lectura de imágenes para el algoritmo forma.vi

En cuanto a la separación de procesos según el tono del vehículo, la secuencia dos (ver Figura 3.23) del *case structure* ilustra el proceso para vehículos de color claro, de acuerdo al siguiente procedimiento:

- Pre-procesamiento: IMAQ Subtract, IMAQ GetKernel e IMAQ Convolute.
- Umbralización: IMAQ Threshold
- Morfología Binaria: IMAQ FillHole, IMAQ Morphology, IMAQ Separation, IMAQ RejectBorder e IMAQ RemoveParticle (el orden depende del tipo de proceso)

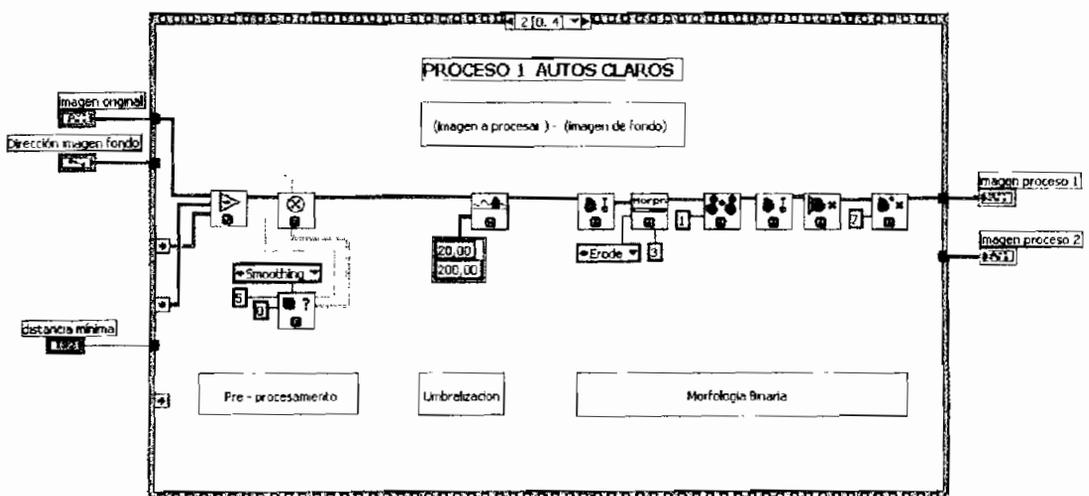


Figura 3.23 Proceso para aislar chasis de Vehículos de color claro

El segundo proceso para los vehículos de color oscuro es indicado en la secuencia tres como se muestra en la Figura 3.24, en el que se observa también que el orden de ejecución de los distintos bloques es diferente al proceso anterior.

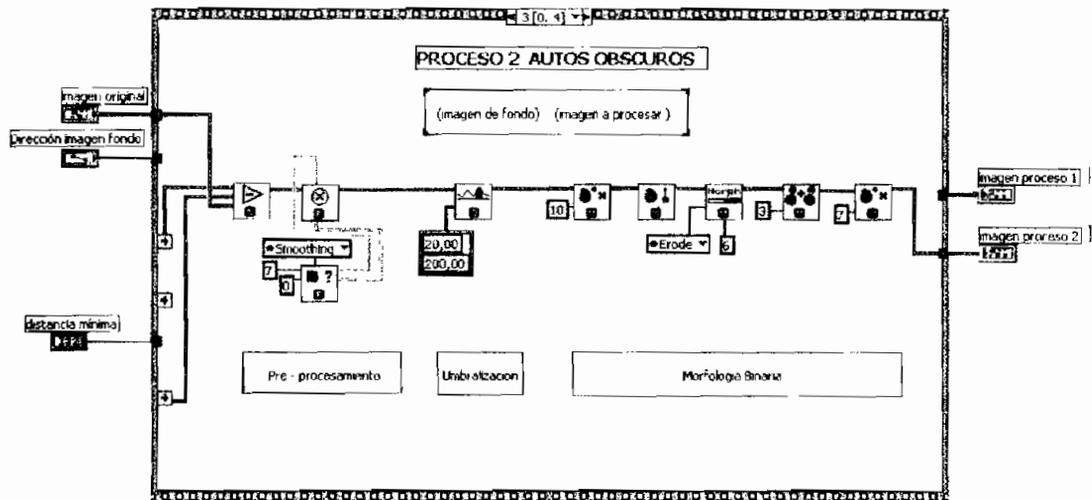


Figura 3.24 Proceso para aislar chasis de Vehículos de color oscuro

En la última secuencia Figura 3.25 se realiza la medición de la partícula con IMAQ BasicParticle para ambos procesos, determinando el tamaño en píxeles de la misma y además verificando si corresponde a la de un chasis.

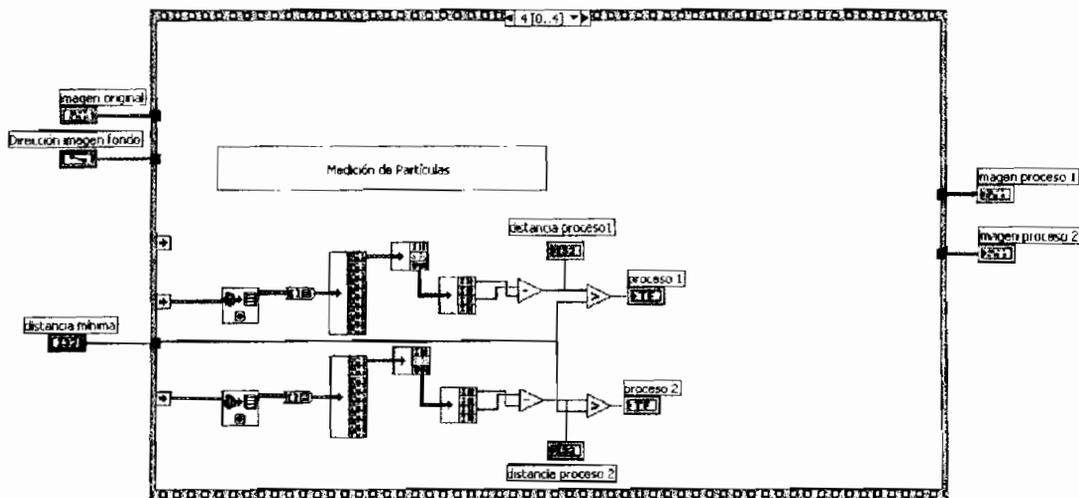


Figura 3.25 Determinación de Tamaño de la partícula para los dos procesos

En la siguiente Tabla se muestra los parámetros de configuración de los bloques de IMAQ Vision utilizados para la implementación de este algoritmo.

BLOQUES \ PROCESO	PROCESO1		PROCESO2		Orden de ejecución		Config	
	AUTOS DE COLOR CLARO		AUTOS DE COLOR OSCURO		Proces o1	Proceso 2		
IMAQ Create								
Image Type	8 bits		8 bits				No	
IMAQ Read File								
File Path	Ubicación de la secuencia de imágenes							No
IMAQ Subtract								
Image Scr A	Imagen de entrada		Imagen de fondo				No	
Image Scr B	Imagen de fondo		Imagen de entrada				No	
IMAQ GetKernel								
Kernel Family	Smoothing		Smoothing				No	
Kernel Size	5		7				No	
Kernel Number	0		0				No	
IMAQ Threshold					1	1		
Range	(20,200)		(20,200)				No	
IMAQ FillHole					2 - 5	3		
IMAQ Morphology					3	4		
Operation	Erode		Erode				No	
Number of iterations	3		6				No	
IMAQ Separation					4	5		
Nb of Erosion	1		3				No	
IMAQ RejectBorder					6	-		
IMAQ Remove Particle					7	2 - 6		
Number Of Erosion	7		10 - 7				No	

Tabla 3.7 Parámetros para el proceso de detección de forma

La simplificación de todas estas secuencias de programación se da con la creación del subVI forma.vi (ver Figura 3.26). La descripción de sus entradas y salidas se indican en la Tabla 3.8.

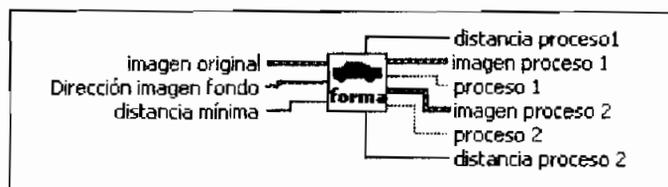


Figura 3.26 forma.vi

forma.vi				
		Descripción	Tipo	Config
Entradas	<i>Imagen Original</i>	Imagen en la cual va a realizarse el proceso de aislar el chasis	Cluster (2elementos): -String -Unsigned long 32 bit integer	No
	<i>Dirección imagen fondo</i>	Ubicación dentro de la secuencia de imágenes para el fondo	Dirección de archivo	Si
	<i>Distancia mínima</i>	Es la distancia en pixeles de la partícula para poder ser considerada como chasis del vehículo.	Constante (long 32 bit integer)	Si
Salidas	<i>Distancia proceso1</i>	Tamaño en pixeles de la partícula en el proceso de autos de color claro	Constante (long 32 bit integer))	
	<i>Imagen proceso 1</i>	Es la imagen en la cual se observa aislado la partícula (chasis) para autos de color claro	Cluster (2 elementos)	
	<i>proceso1</i>	Es la confirmación de detección de vehículo de color claro	Booleana	
	<i>Distancia proceso2</i>	Tamaño en pixeles de la partícula en el proceso de autos de color oscuro	Constante (long 32 bit integer))	
	<i>Imagen proceso 2</i>	Es la imagen en la cual se observa aislado la partícula (chasis) para autos de color oscuro	Cluster (2 elementos)	
	<i>proceso2</i>	Es la confirmación de detección de vehículo de color oscuro	Booleana	

Tabla 3.8 Descripción de entradas y salidas del subVI forma.vi

3.5 IMPLEMENTACION DEL PROGRAMA PRINCIPAL

Ahora se empezará con la descripción misma de la ejecución del programa principal, el cual será el encargado de monitorear el video (secuencia de imágenes), así como también de procesar las distintas imágenes que componen dicha secuencia. El sistema del presente proyecto, de manera general presenta la siguiente concatenación de eventos:

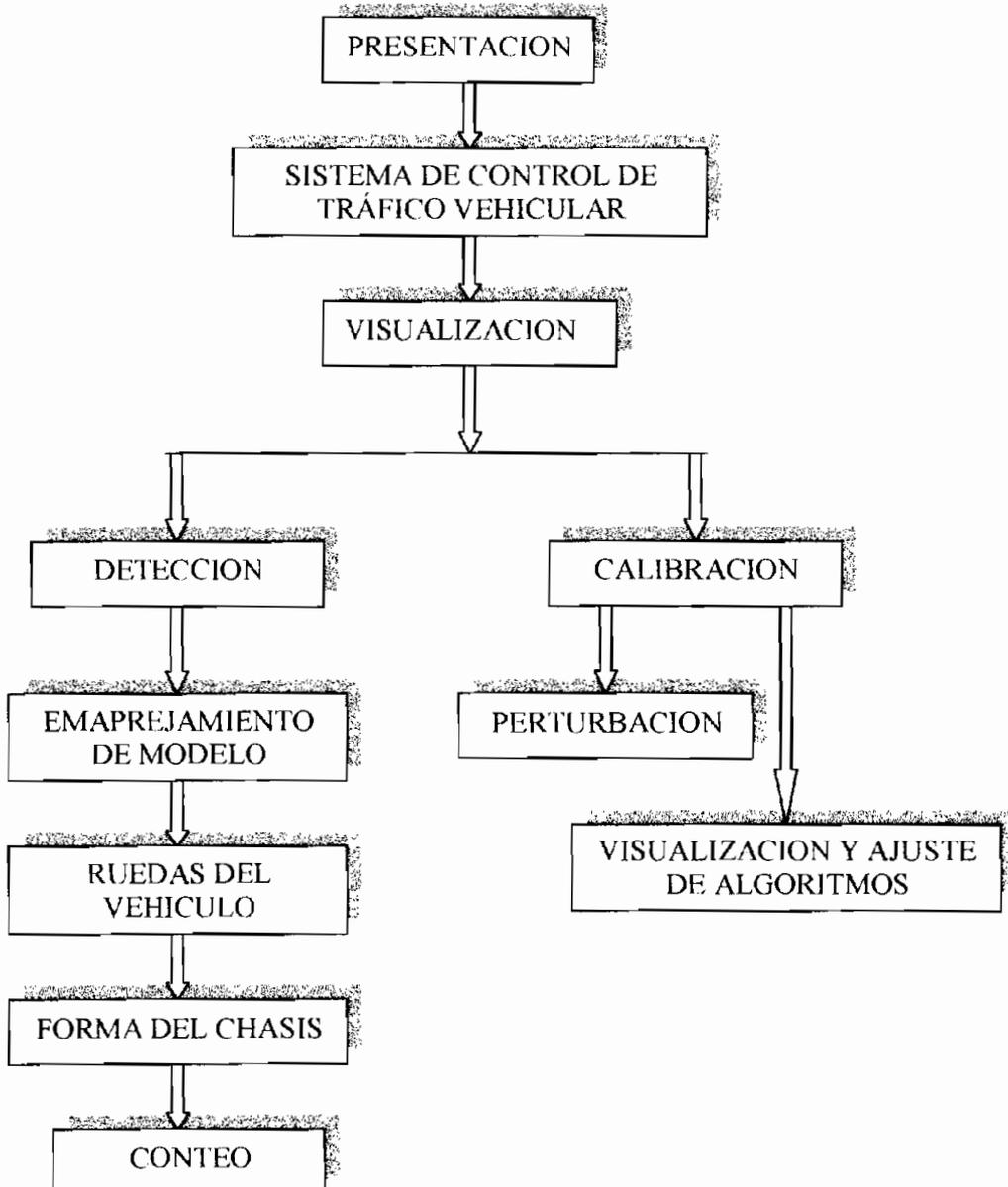


Figura 3.27 Esquema general del sistema de control de tráfico vehicular

La pantalla de presentación (ver Figura 3.28) se la realiza de igual forma en el programa LabVIEW, y se encarga de dar paso al proceso mismo de Monitoreo.

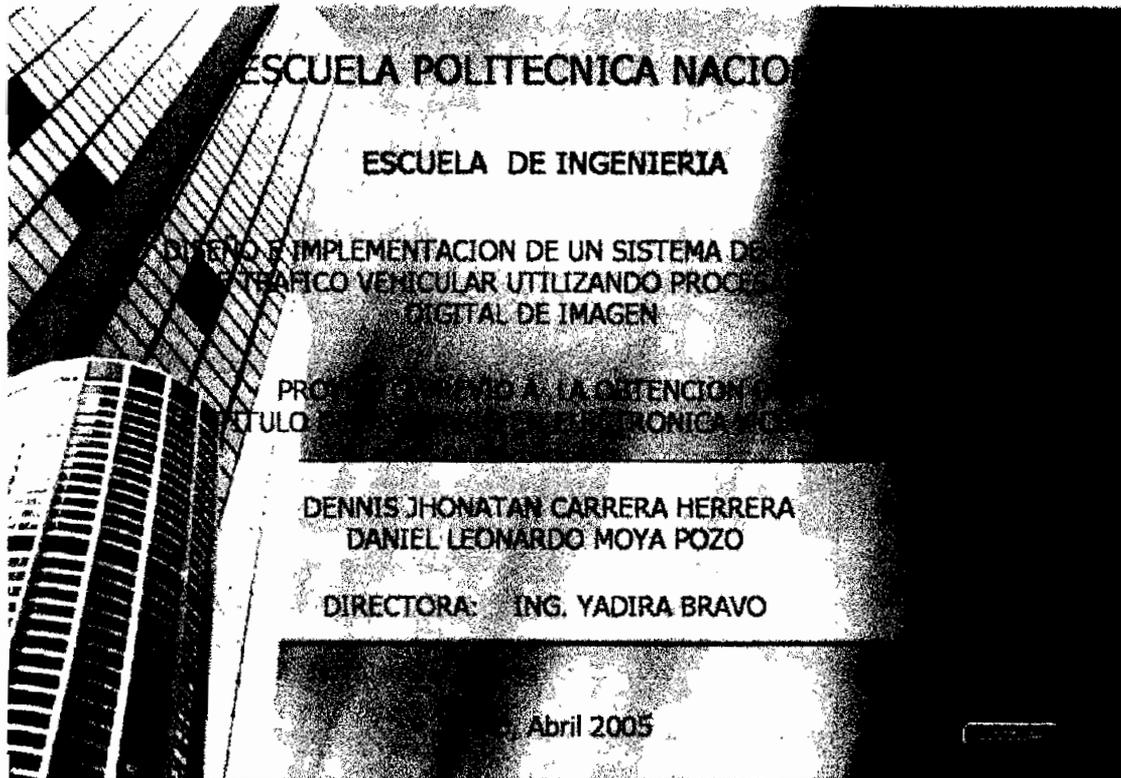


Figura 3.28 Pantalla de presentación e introducción al sistema de control

En el momento de presionar el botón *CONTINUAR* en la interfaz gráfica anterior, se despliega la pantalla de reproducción y calibración del sistema (panel frontal del programa Detección y Conteo de Vehículos.vi), como se ilustra en la Figura 3.29, la cual consta de las siguientes partes:

- Control principal
- Ventana de visualización normal (MONITOREO VEHICULAR)
- Controladores para detección de perturbación
- Ventana de CALIBRACION
- Controladores de configuración general



Figura 3.29 Pantalla de reproducción y calibración del sistema de control de tráfico vehicular

3.5.1 CONTROL PRINCIPAL

El control principal es la única interfaz que aparece en el momento de entrar al sistema de control, en la cual el usuario podrá realizar acciones de control como:

- Seleccionar y abrir la carpeta que contiene la secuencia de imágenes
- Reproducir el video seleccionado
- Detener la reproducción y el conteo de vehículos
- Dar paso a la configuración para detección de perturbación (C1)
- Dar paso a la configuración de parámetros generales (C2)
- Salir del sistema

A la vez en ésta interfaz se pueden visualizar parámetros como los siguientes:

- Indicador tipo LED, al detectar un vehículo es encendido
- Contador del número de vehiculos

- Mensajes de Estado como: ABRIR VIDEO, REPRODUCIENDO, DETENIDO (2 seg.)
- Nombre de la carpeta con la secuencia de imágenes. Ej. Clip: seg_clip3-1_7fps.
- Tamaño (en frames) que contiene la carpeta seleccionada.
- Ubicación de la carpeta
- Tipo de video procesado: MPEG – 1.

Por lo tanto dichos controladores e indicadores se visualizan de mejor forma en la Figura 3.30.

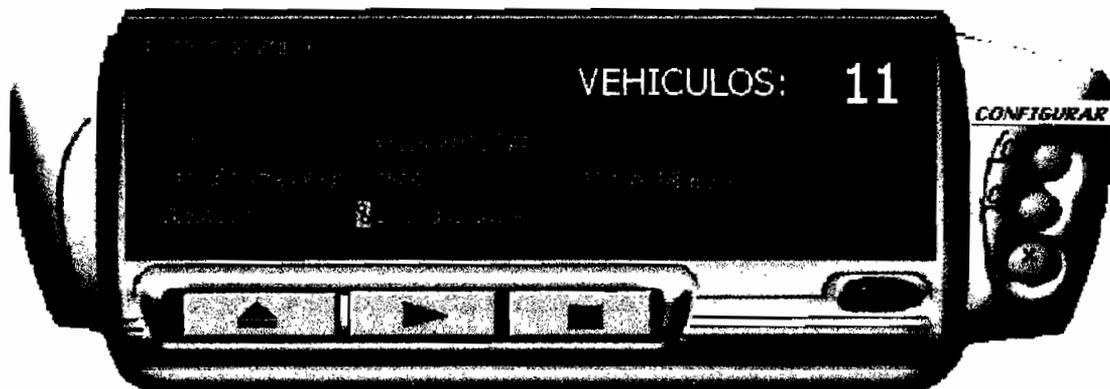


Figura 3.30 Interfaz gráfica para el control principal

3.5.2 VENTANA DE VISUALIZACION NORMAL (MONITOREO VEHICULAR)

Se trata de la ventana de imagen en la cual se podrá visualizar el video (secuencia de imágenes) que está siendo procesado por los distintos algoritmos de reconocimiento vehicular; por lo tanto, cuando esta ventana se encuentre activa, el contador de vehículos lo estará también. La Figura 3.31 ilustra dicha ventana, la cual es flotante; es decir, se la puede mover a cualquier lugar de la pantalla del monitor, y además ésta ventana puede variar en base a los parámetros que se pueden ajustar en los controles de configuración general.

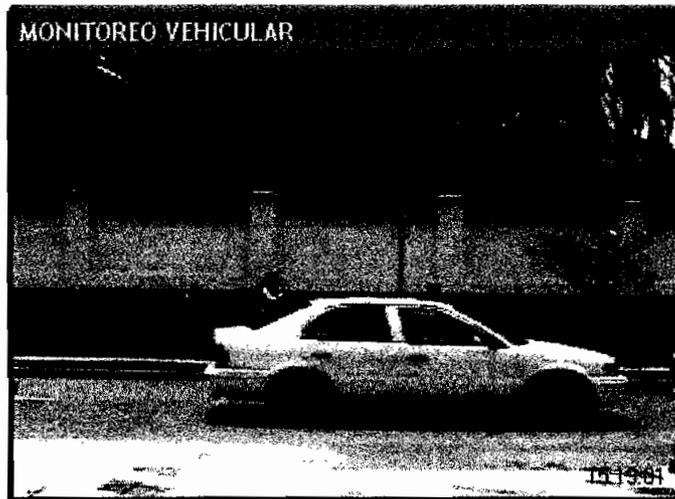


Figura 3.31 Ventana MONITOREO VEHICULAR

3.5.3 CONTROLADORES PARA DETECCION DE PERTURBACION

Antes de comenzar con la descripción de estos controladores, se explicarán el bloque usado para la medición del nivel de gris medio de las dos líneas colocadas en la parte inferior de las imágenes con el fin de comprobar si existe perturbación o no, así como también los bloques que servirán para la visualización de la hora en la parte inferior derecha (Monitoreo) y para mostrar las dos líneas (Calibración).

IMAQ LineProfile

Calcula el perfil de una línea de píxeles.

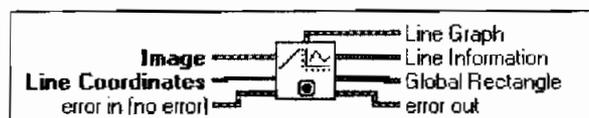


Figura 3.32 IMAQ LineProfile

IMAQ Overlay Text

Sobrescribe texto en la imagen.

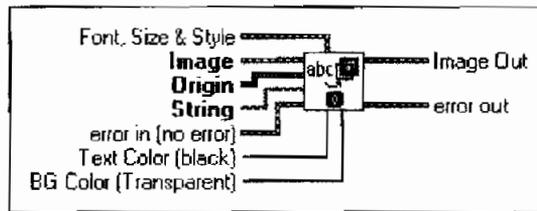


Figura 3.33 IMAQ Overlay Text

IMAQ Overlay Line

Sobrescribe una línea sobre una imagen.

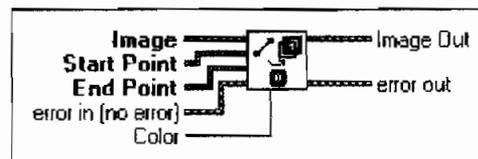


Figura 3.34 IMAQ Overlay Line

IMAQ Clear Overlay

Limpia cualquier sobre escritura que exista sobre la imagen.

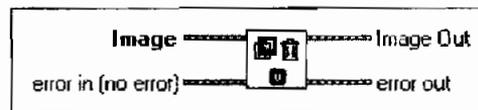


Figura 3.35 IMAQ Clear Overlay

Ahora bien, dichos controladores (ver Figura 3.36) son visibles únicamente al presionar el botón de configurar C1 (CONFIG1) y permite la calibración de parámetros que servirán para la detección de perturbación, y por ende hacerle a este sistema de control eficiente para cualquier tipo de filmación con los requerimientos señalados en el capítulo 2 (Captura de Video).

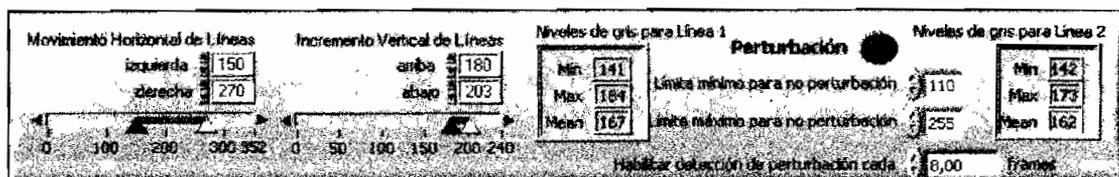


Figura 3.36 Botones para Configuración 1

La clasificación de todos estos parámetros se ilustran en la Tabla 3.9.

Parámetro	Descripción	Rango
Movimiento Horizontal de Líneas (Controlador, slide horizontal con doble puntero)	Mueve cada una de las dos líneas en el sentido horizontal y de manera independiente. Sirve para la medición de ng medio en un determinado sector de las imágenes.	(0 – 352) píxeles Tanto para el puntero izquierdo como para el derecho
Incremento Vertical de Líneas (Controlador, slide horizontal con doble puntero)	Incrementa el tamaño en el sentido vertical de las dos líneas a la vez. Sirve para la medición de ng medio en un determinado sector de las imágenes.	(0 – 240) píxeles Tanto para el puntero izquierdo como para el derecho
Niveles de gris Línea1 (Indicador)	Permite visualizar los valores mínimo, máximo y medio de la línea izquierda. Sirve para ajustar los valores min y max para no perturbación.	(0 – 255) ng
Niveles de gris Línea 2 (Indicador)	Permite visualizar los valores mínimo, máximo y medio de la línea derecha.	(0 – 255) ng
Límite mínimo para no perturbación (Controlador)	Nivel de gris medio de las líneas, debajo del cual, existirá perturbación	(0 – 255) ng
Límite máximo para no perturbación (Controlador)	Nivel de gris medio de las líneas, por encima del cual, existirá perturbación	(0 – 255) ng
Perturbación (Indicador, tipo LED)	Señal que indica que se ha producido perturbación.	ON - OFF
Habilitar detección de perturbación cada (Controlador)	Ocurrido una perturbación, este parámetro se refiere al número de frames en que el sistema puede volver a detectar una perturbación. Depende de la velocidad de los vehículos.	0 - ∞ frames

Tabla 3.9 Descripción de parámetros de CONFIG1

Los valores por defecto de cada uno de los controladores ajustados de acuerdo a las primeras filmaciones son los siguientes:

- Movimiento Horizontal de Líneas: (izquierda 150, derecha 270)
- Incremento Vertical de Líneas: (arriba 180, abajo 203)
- Límite mínimo para no perturbación: 110
- Límite máximo para no perturbación: 255
- Habilitar detección de perturbación cada: 8

3.5.4 VENTANA DE CALIBRACION

Se trata de una ventana de visualización del video seleccionado, aparece en el momento de presionar el botón de configurar C1 (CONFIG1); por lo tanto, al variar los parámetros de calibración anteriores, se podrá ir observando dichos cambios en esta ventana con el fin de ajustar la señal de perturbación, y así dar paso a los distintos algoritmos de reconocimiento (modo monitoreo y conteo).

Esta ventana al estar activa, el contador de vehículos se bloquea ya que el sistema se encuentra en modo de calibración; para su visualización se utilizó un tipo de imagen a 8 bits, paleta de color gris, resolución mínima y la sobre escritura de las dos líneas, tal como se indica en la siguiente Figura:

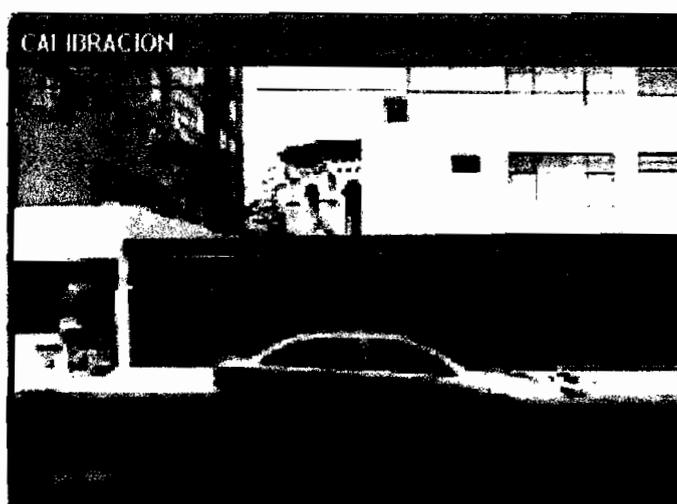


Figura 3.37 Pantalla CALIBRACION

3.5.5 CONTROLADORES DE CONFIGURACION GENERAL

Antes de comenzar con la descripción de estos controladores, se explicarán bloques de la librería IMAQ Vision usados exclusivamente en lo que se refiere a la visualización de imágenes en las distintas ventanas, y configuración de las mismas como la posición, tamaño, color de paleta, título de la barra, etc.

IMAQ WindDraw

Despliega una imagen en una ventana de imagen. La ventana de imagen aparece automáticamente cuando éste VI es ejecutado.

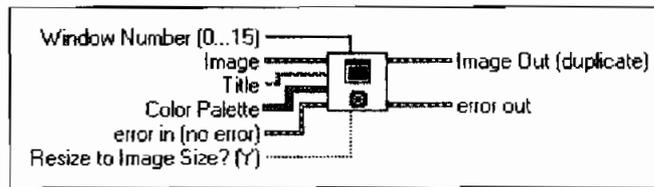


Figura 3.38 IMAQ WindDraw

IMAQ GetPalette

Selecciona una paleta de visualización, existen a disposición cinco paletas predefinidas (gray, binary, gradient, rainbow, temperature). La salida Color Palette será unido a la entrada correspondiente en el bloque indicado anteriormente.

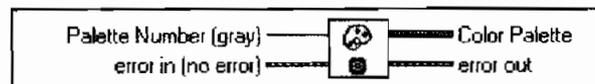


Figura 3.39 IMAQ GetPalette

Los bloques usados para la configuración de las ventanas de imagen en el programa principal, son los siguientes:

IMAQ Resample

Ajusta la resolución de una imagen a un tamaño definido por el usuario. Se puede usar este VI para desplegar imágenes reducidas o alargadas.

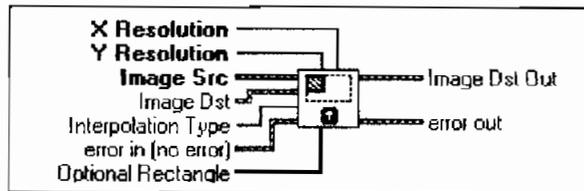


Figura 3.40 IMAQ Resample

IMAQ WindMove

Indica y configura la posición de una ventana de imagen.

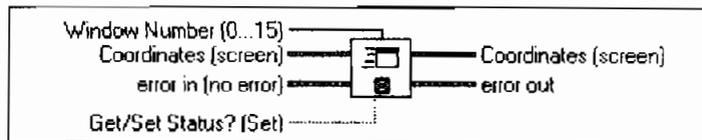


Figura 3.41 IMAQ WindMove

IMAQ WindSetup

Configura la apariencia y atributos de una ventana.

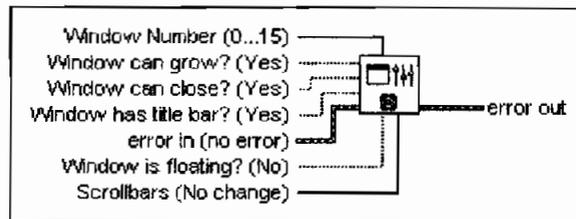


Figura 3.42 IMAQ WindSetup

Estos controladores permiten la calibración de parámetros de visualización, de reproducción y de ajuste para los algoritmos de reconocimiento vehicular. Cabe indicar que estos controladores de visualización son ajustables únicamente para la ventana (MONITOREO VEHICULAR). La Figura 3.43 muestra éstos controladores que son visibles en el momento de presionar el botón de configurar C2 (CONFIG2).

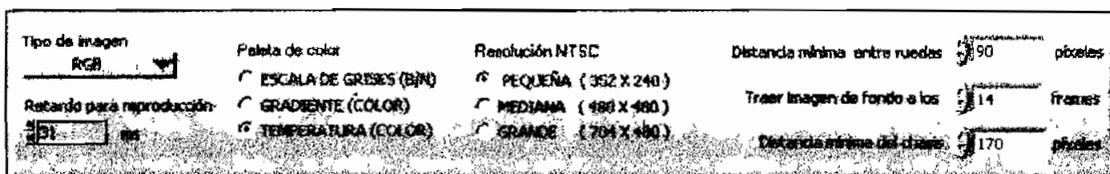


Figura 3.43 Botones para Configuración 2

La clasificación de todos estos parámetros se ilustran en la Tabla 3.10.

Parámetro	Descripción	Opciones / Rango
Tipo de imagen (monitoreo)	Es el tipo de imagen usado para la visualización del video	RGB, HSL, 8 bits, 16 bits, complex.
Paleta de Color (monitoreo)	Se refiere a la matriz de colores que serán visualizados en la ventana de imagen	Gris, temperatura, gradiente, arco iris, binaria
Resolución NTSC (monitoreo)	Se refiere al tamaño de la ventana para la visualización del video	352 x 240 480 x 480 708 x 480
Retardo para reproducción	Se trata del tiempo de visualización entre imágenes en las cuales no se ha detectado perturbación. Sirve para empatar el tiempo de reproducción real (video) con los tiempos de procesamiento y reproducción (secuencia de imágenes)	0 - ∞ milisegundos
Distancia mínima entre ruedas	Es la distancia que tienen que estar separados los centros de los círculos detectados para poder ser considerados como vehículos.	(0 – 352) píxeles
Traer imagen de fondo a los	Se trata del número de frames anteriores al actual, para poder recuperar como imagen de fondo. Usado en el proceso de detección de forma	0 - ∞ frames
Distancia mínima de chasis	Es la distancia de la partícula para poder ser considerada como chasis del vehículo.	(0 – 352) píxeles

Tabla 3.10 Descripción de parámetros de CONFIG2

Los valores por defecto y bloques IMAQ Vision asociados a cada uno de los controladores son los siguientes:

- Tipo de imagen: RGB / IMAQ Create
- Paleta de Color: Temperatura / IMAQ GetPalette
- Resolución NTSC: 352 x 240 / IMAQ Resample

- Retardo para reproducción: 31 ms
- Distancia mínima entre ruedas: 90 / (ruedas.vi)
- Traer imagen de fondo a los: 14
- Distancia mínima de chasis:170 / (forma.vi)

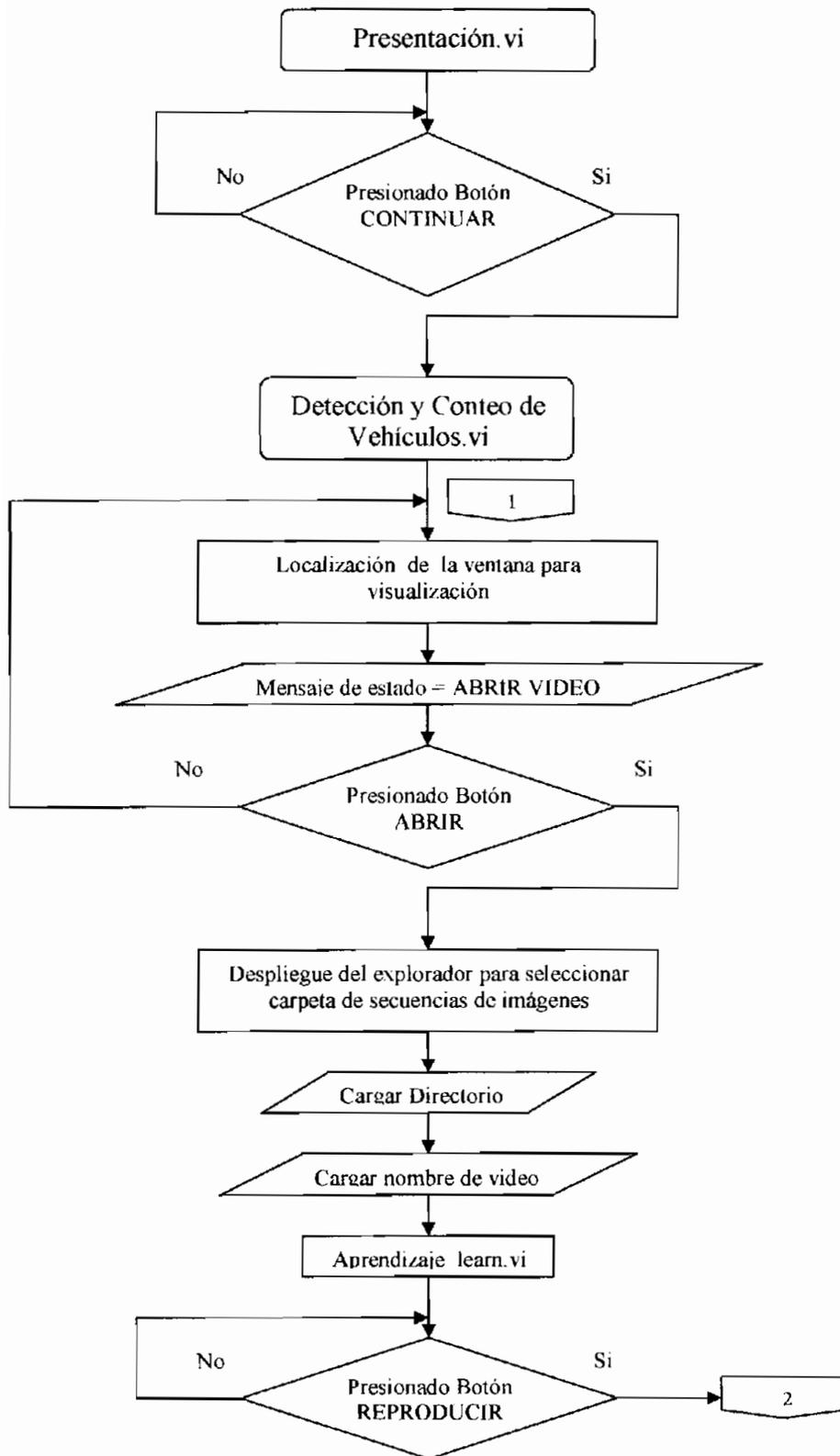
3.6 DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL

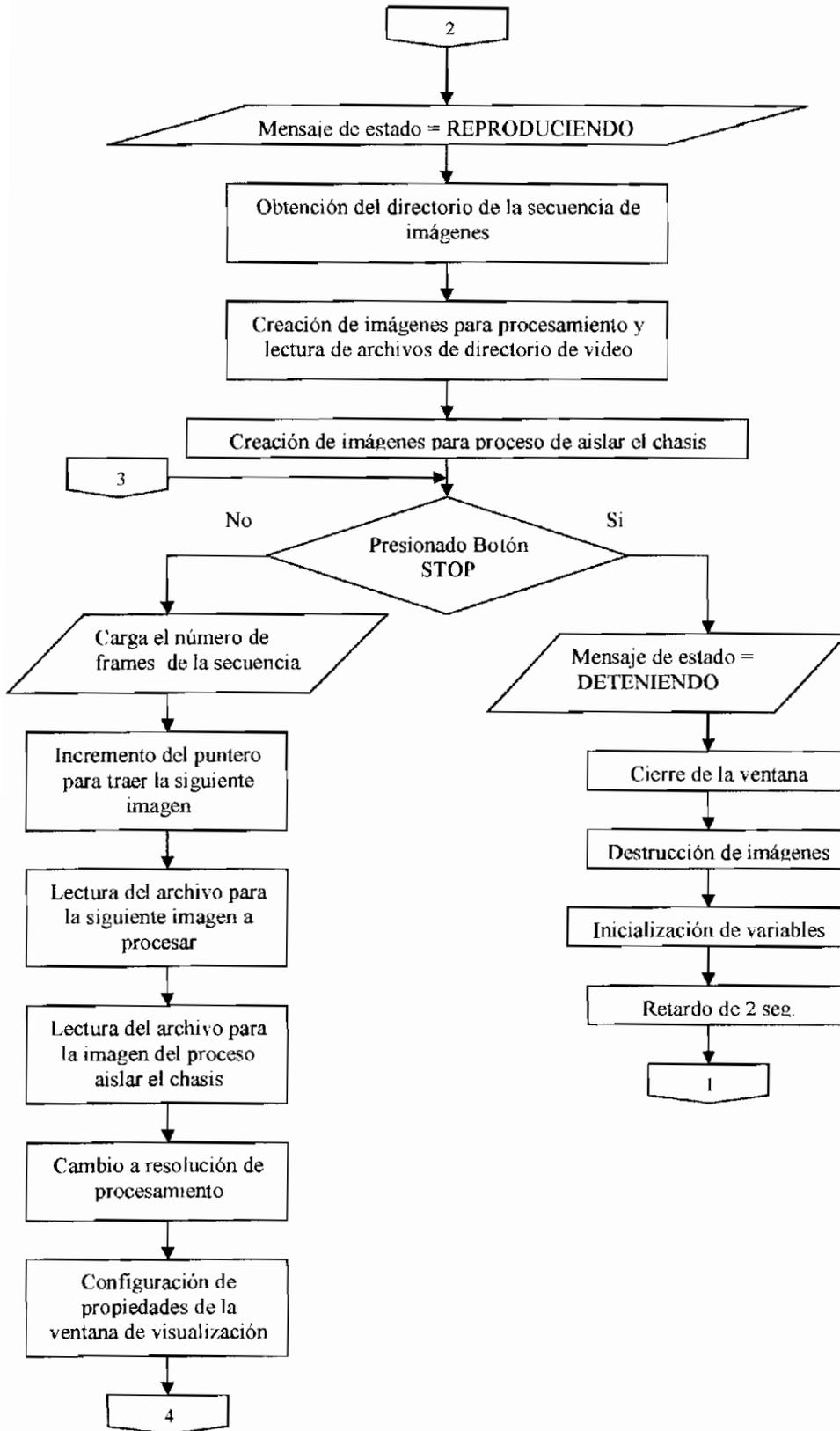
Antes de ver el flujo total de información del sistema de control implementado, cabe recalcar que la resolución de la imagen de procesamiento siempre se la realiza con los valores de defecto para el formato MPEG – 1 (352 x240 para NTSC); para que de esta forma las focalizaciones de los distintos algoritmos de reconocimiento sean únicas, y no son ajustables por el usuario en el modo de Configuración. Además, el tipo de imagen involucrado en el procesamiento a lo largo de todo el programa, es de 8 bits; con el fin trabajar con imágenes en niveles de gris dada su adaptabilidad a cualquier tipo de procesamiento al cual sea sometido y también para incrementar la velocidad del proceso.

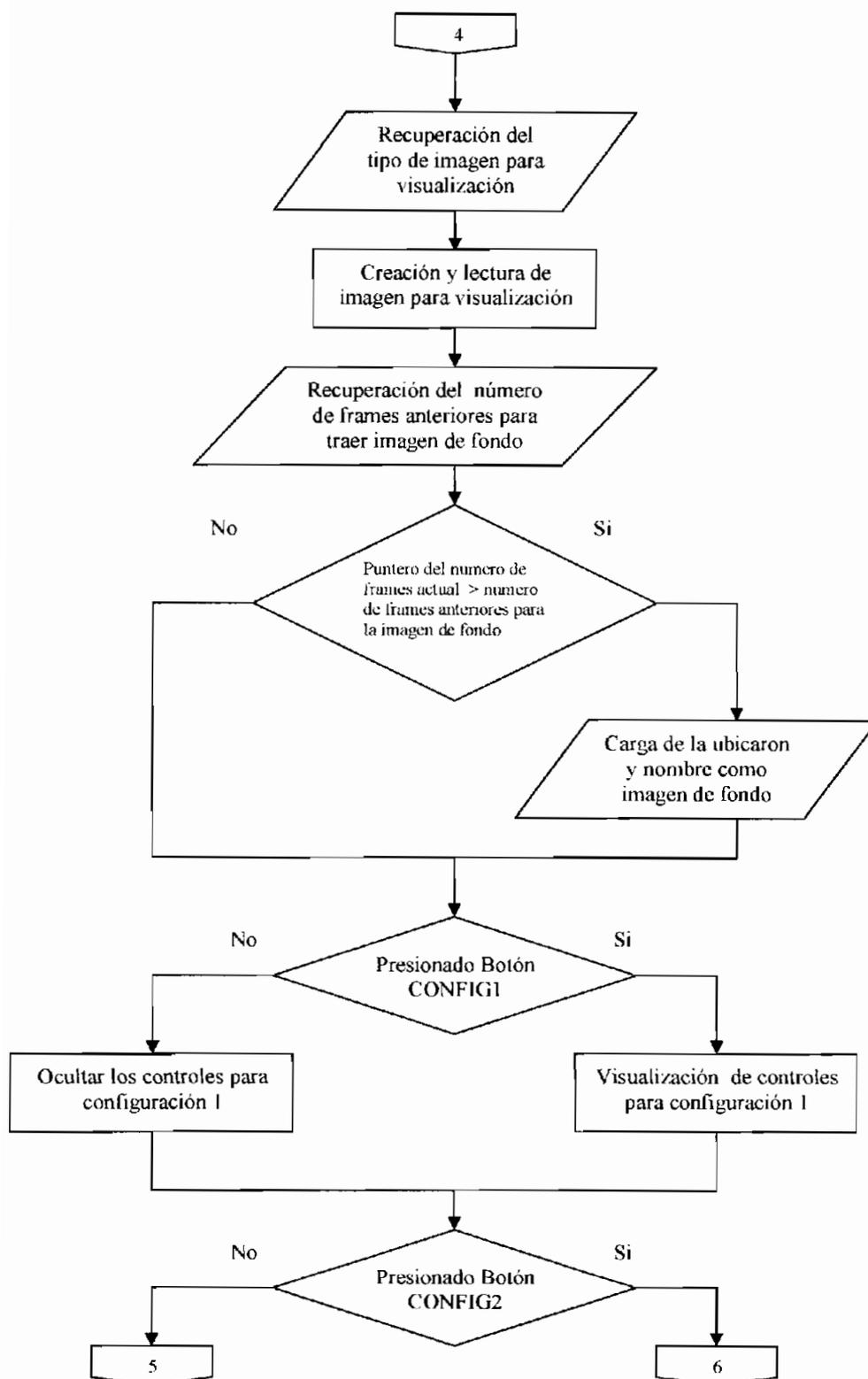
A continuación se describen dos variables que forma parte del programa principal en el proceso de detección de perturbación:

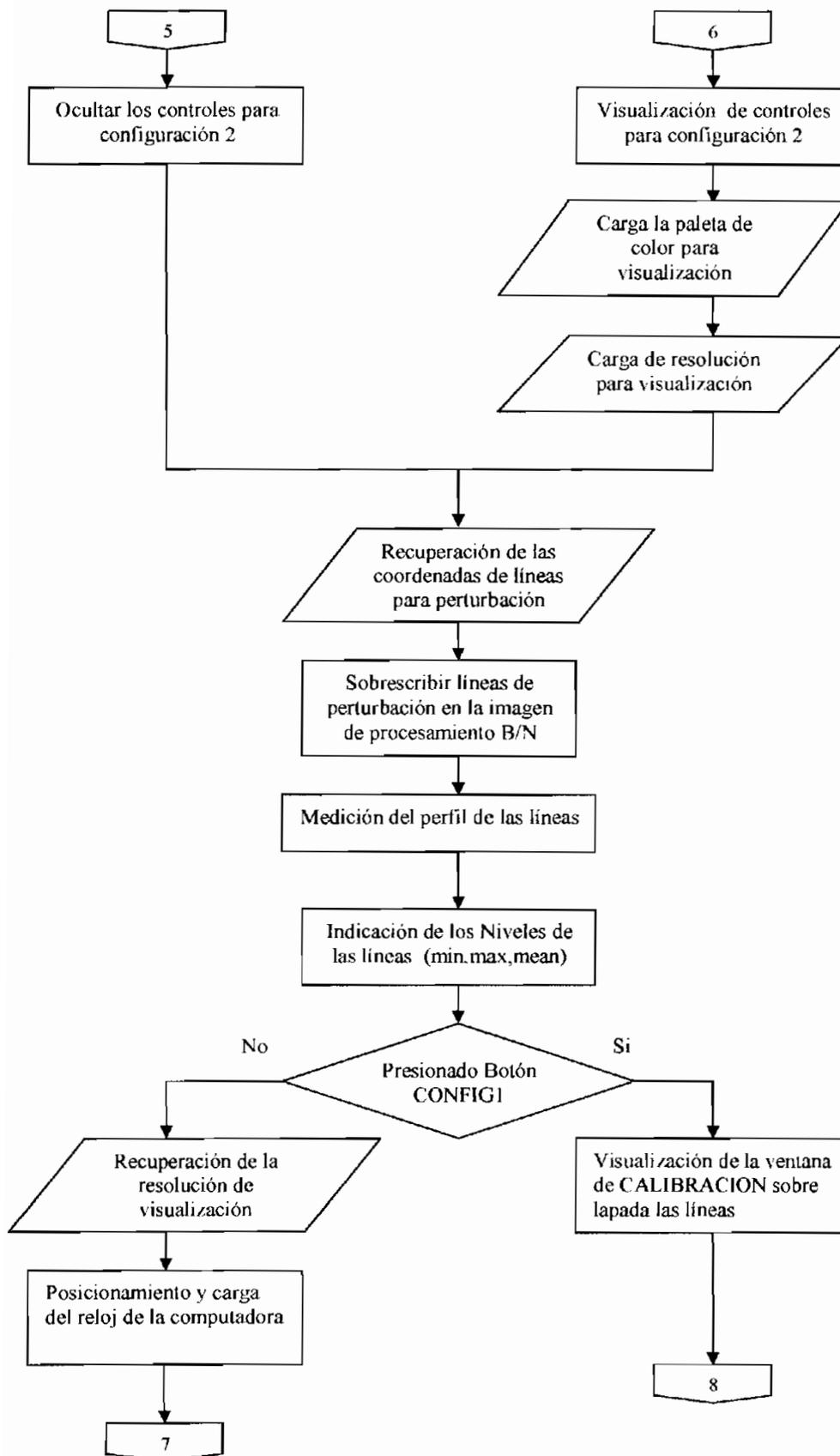
- **ALARMA**: es un contador de cuadros, durante los cuales no se ha producido ninguna perturbación; por lo tanto, éste es limpiado cada vez que existe una perturbación efectiva.
- **VECES**: es una variable booleana (ON / OFF) que cuando está activa, permite que se detecte una perturbación efectiva. Es puesta en ON solamente cuando el valor de Alarma es mayor a la variable "*Habilitar detección de perturbación cada*" (configurable por el usuario).

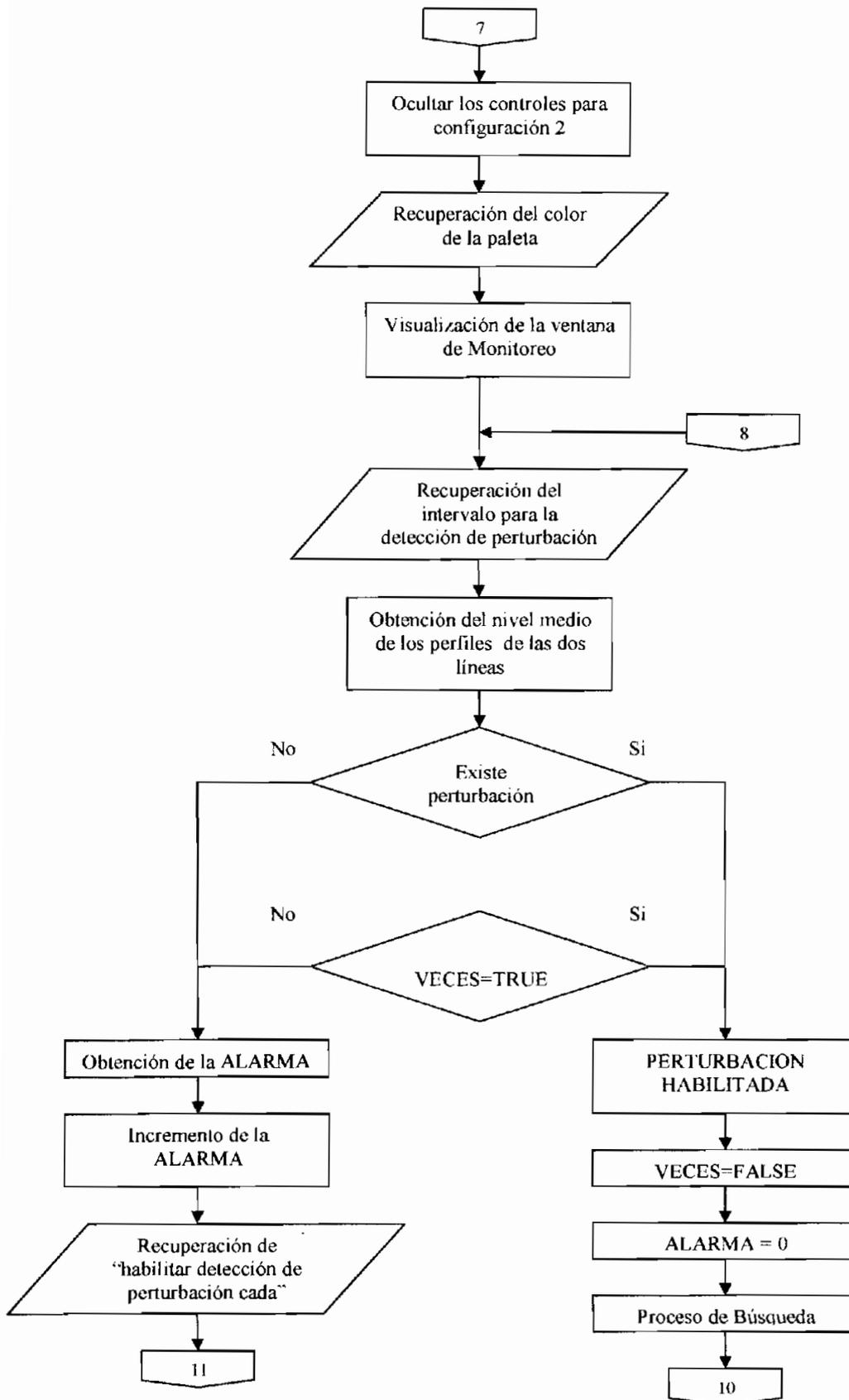
La siguiente Figura muestra el diagrama de flujo del programa principal, en donde se observa claramente los dos modos de operación: Monitoreo (conteo automático de vehículos) y Calibración (ajuste de parámetros para visualización, reconocimiento y perturbación). Esta estructura, hace que el sistema sea muy óptimo para su implementación en tiempo real.

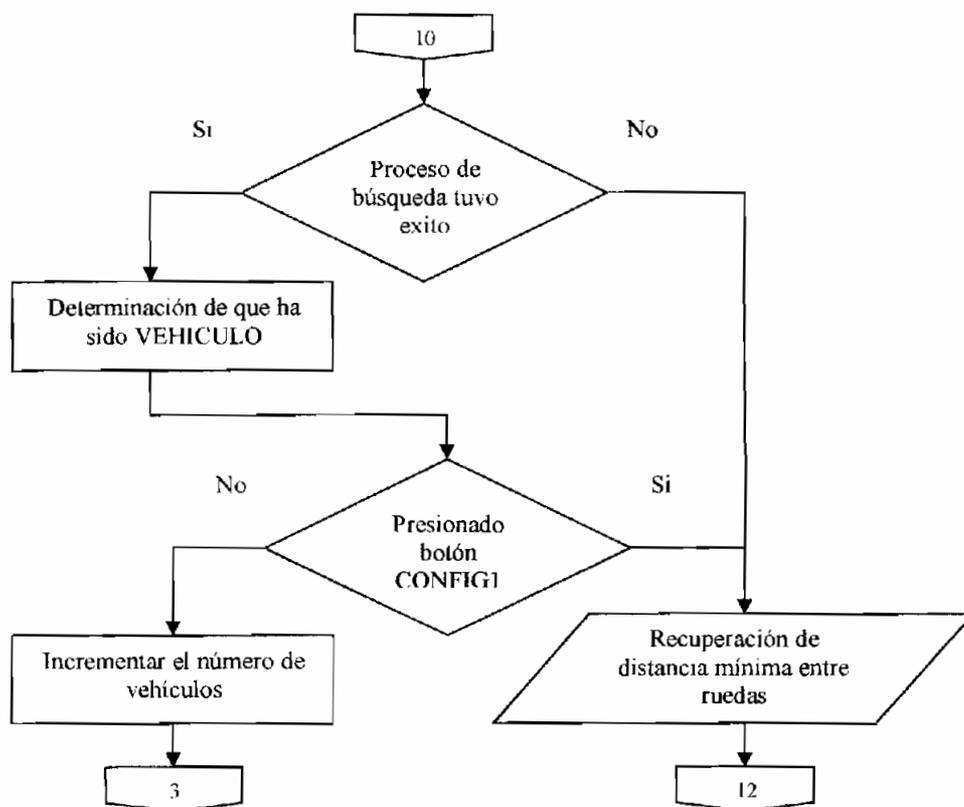
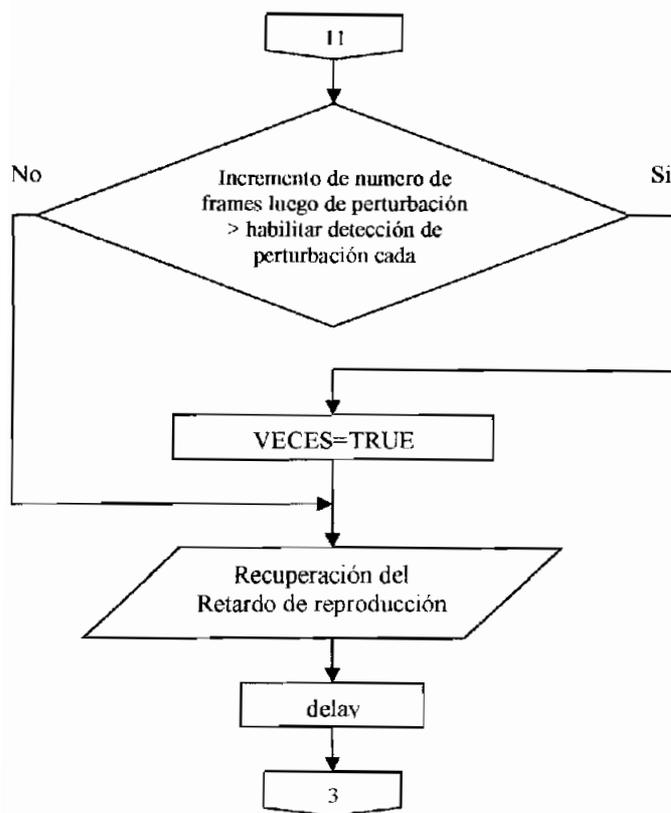


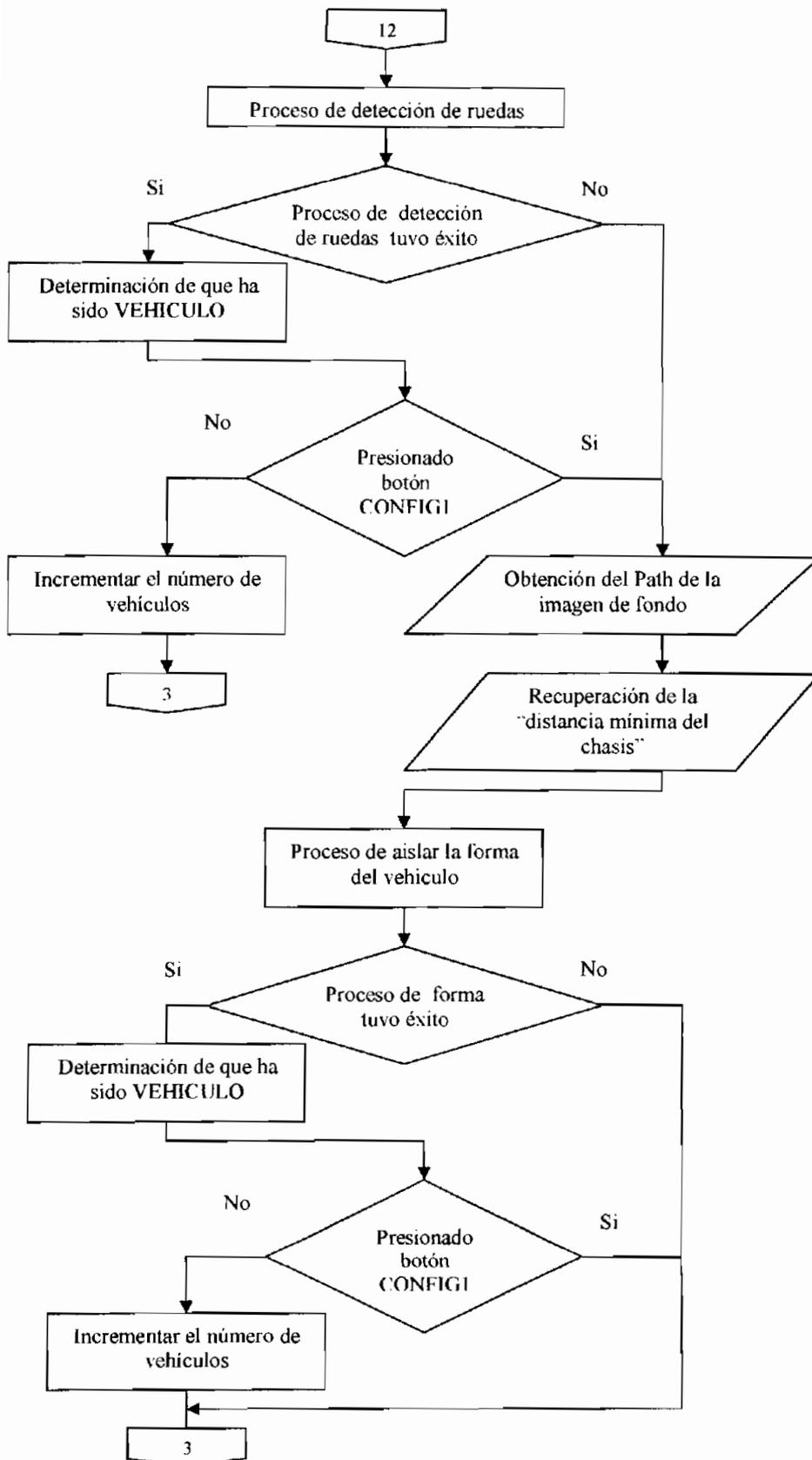












En el siguiente capítulo se describen y analizan los resultados que arroje el sistema de control implementado ante distintas clases de filmaciones en las cuales se quiere llevar a cabo el conteo de vehículos, así como también el comportamiento de los distintos algoritmos de reconocimiento de manera independiente.

CAPITULO 4

PRUEBAS Y RESULTADOS

4.1 INTRODUCCIÓN

En el presente capítulo se describen y analizan los resultados obtenidos en los distintos algoritmos de reconocimiento de manera independiente, así como también el comportamiento del sistema de control general implementado, ante las distintas clases de filmaciones en las cuales se quiere llevar a cabo el conteo de vehículos.

El sistema de control de tráfico tiene dos modos de operación, en este capítulo, se realizó un ejemplo para uno de estos procesos; el de calibración donde se ajusta los parámetros de un video seleccionado que no cumpla totalmente con la efectividad que el sistema requiere.

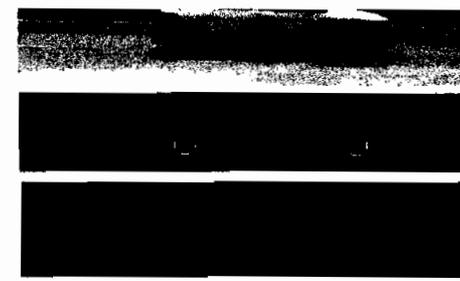
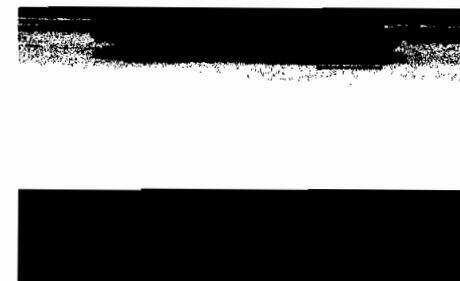
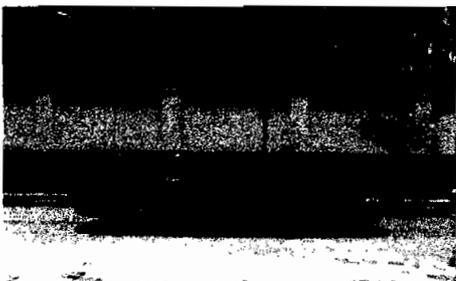
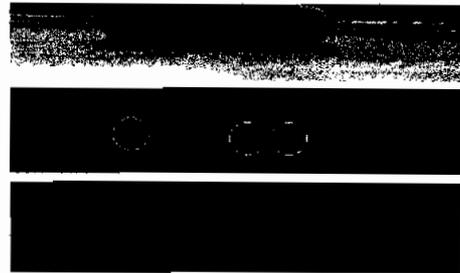
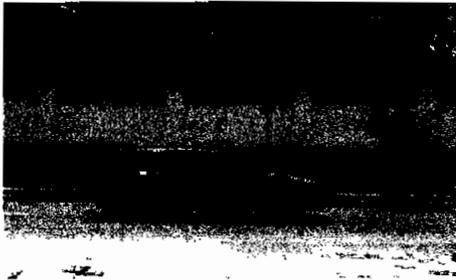
4.2 RESULTADOS DE LOS DISTINTOS ALGORITMOS DE CONTROL

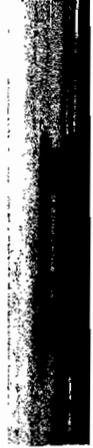
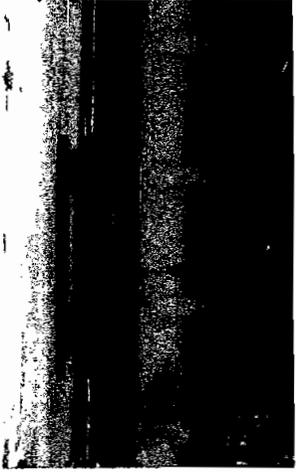
Los resultados se analizan y comparan con la utilización de una secuencia de imágenes y presentando el resultado del procesamiento de cada imagen al momento que el vehículo esta en la posición indicada y activa la señal de perturbación; es decir, en el centro del cuadro para cada uno de los algoritmos de detección y conteo de vehículos como se indica en la Figura 4.1.

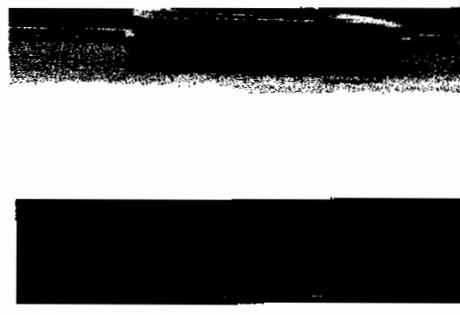
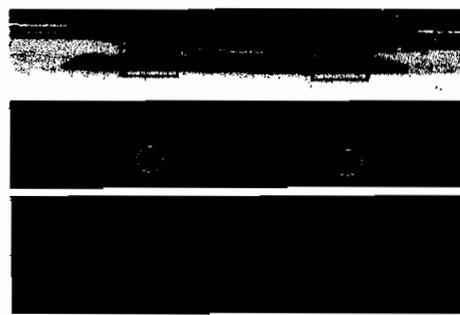
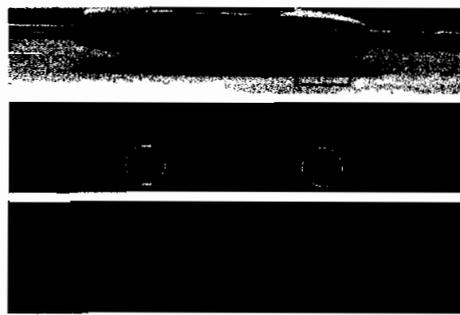
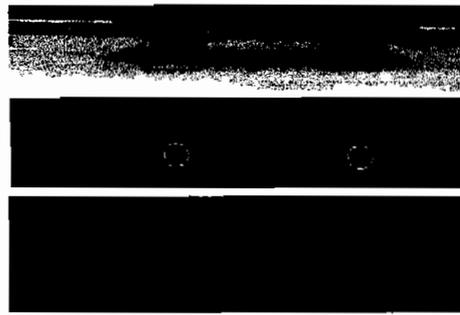
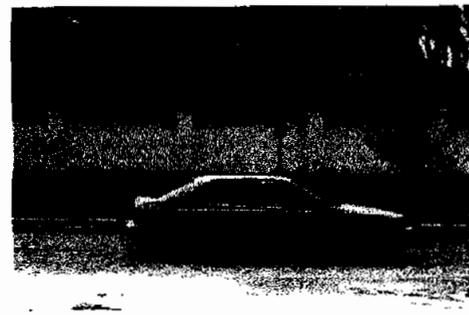
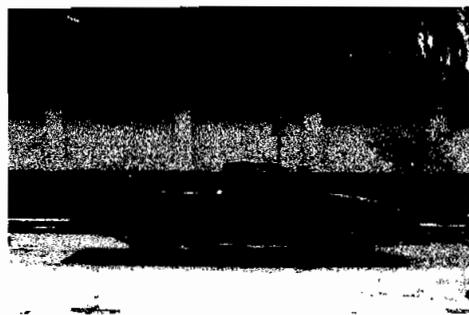
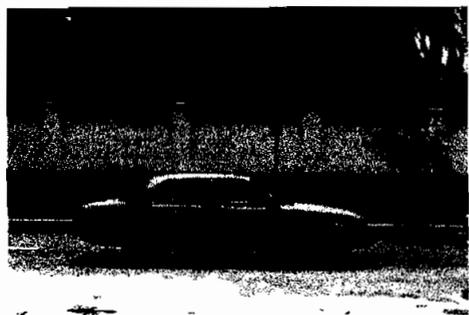
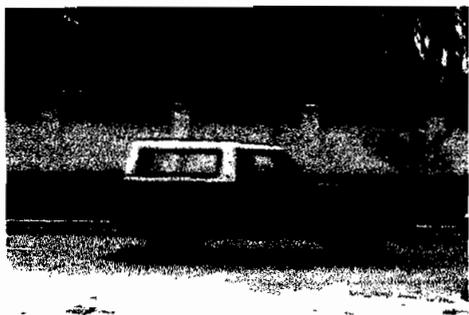


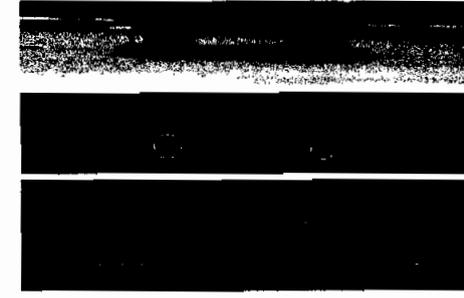
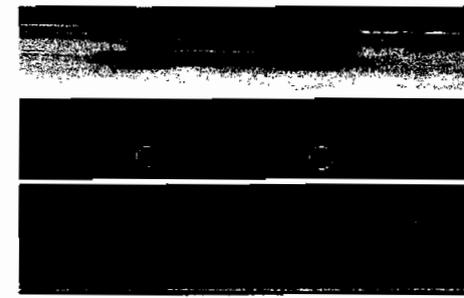
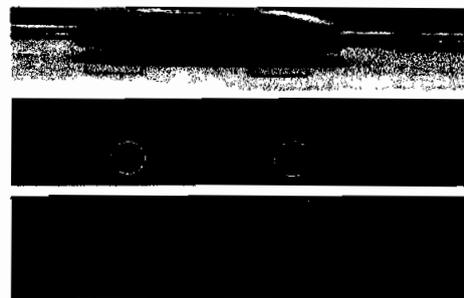
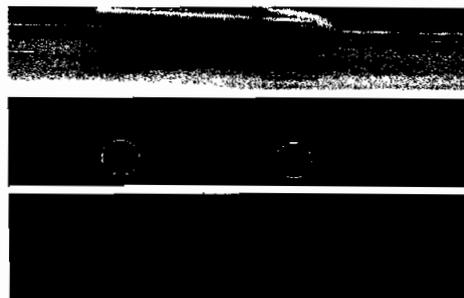
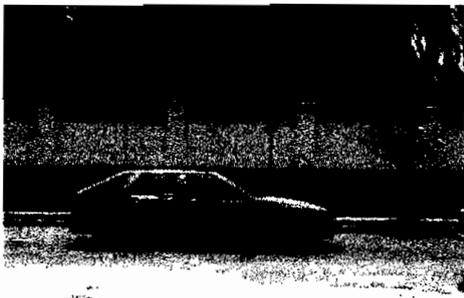
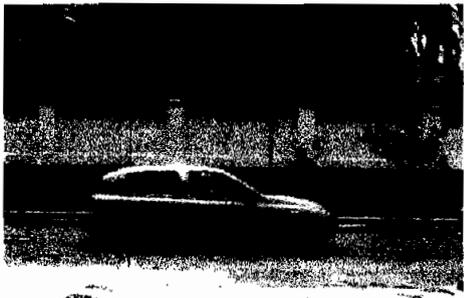
Figura 4.1: Imágenes resultantes de cada algoritmo

Los objetivos son probar la versatilidad y la eficiencia de los algoritmos diseñados. El orden de ejecución en el programa principal de los métodos de reconocimiento, se fundamentaron en las pruebas realizadas para cada uno de los subVI's por separado, la mayor confiabilidad está demostrada por el método de emparejamiento de modelo, seguido del algoritmo de detección de ruedas; y por último, detección de forma. Cabe señalar que en las siguientes Figuras el espacio en blanco significa el no reconocimiento de un determinado algoritmo.









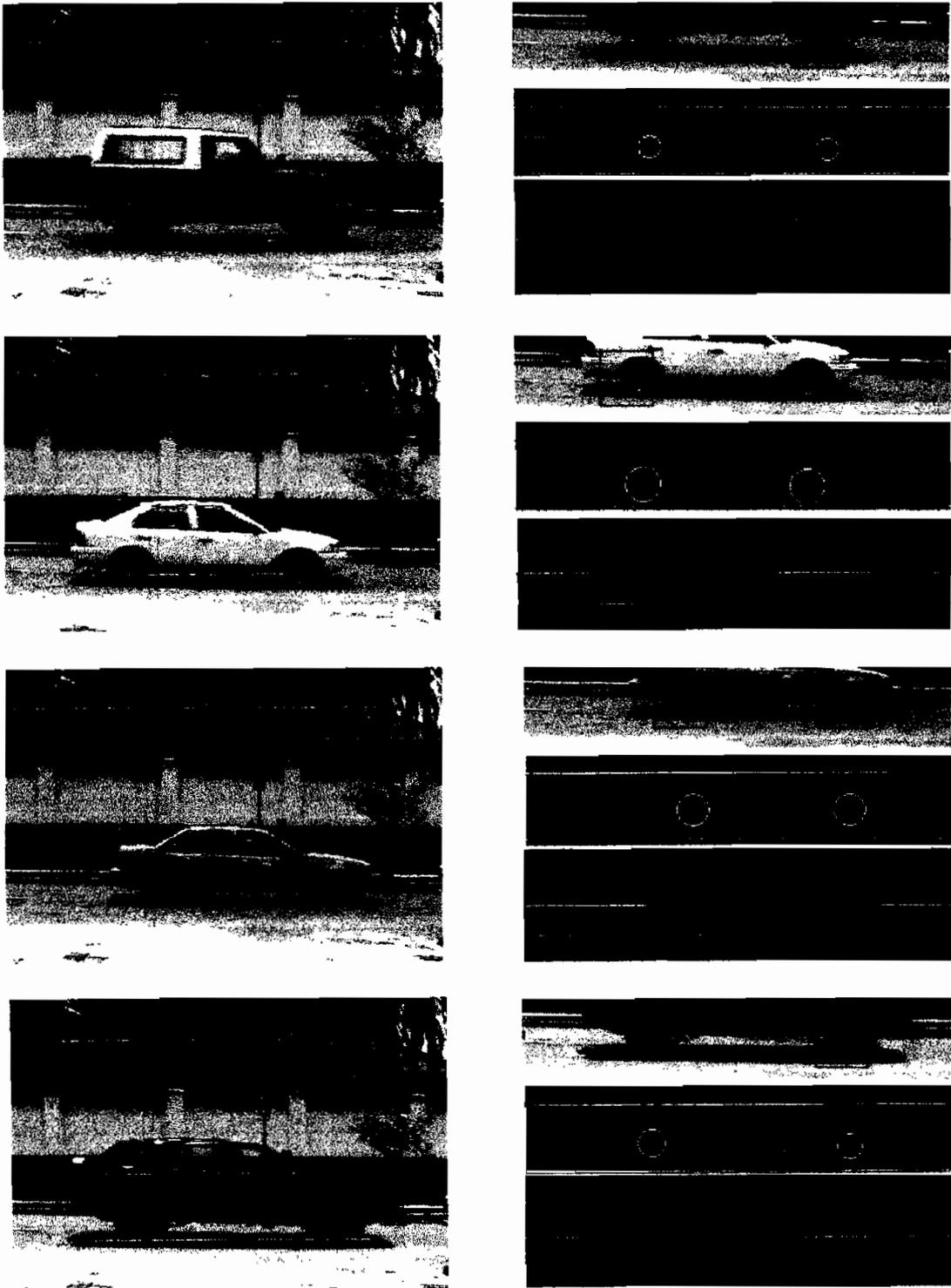


Figura 4.2 Aplicación de los algoritmos por separado

En este capítulo se muestran solamente los resultados arrojados de los videos que inicialmente se obtuvieron para el desarrollo de este proyecto, cabe indicar que a partir de ellos se lograron ajustar todos los parámetros de los bloques de la librería IMAQ Vision utilizados y por ende tener parámetros que se utilizan por defecto en modo de calibración del sistemas de control de tráfico vehicular implementado.

4.2.1 RESULTADOS DEL ALGORITMO DE RECONOCIMIENTO EMPAREJAMIENTO DE MODELO

A continuación se ilustra la Tabla 4.1 que indica la efectividad de cada uno de los modelos para este algoritmo. Se indica además el número de vehículos involucrados en cada uno de los dos videos.

Posición del vehículo en el video	NUMERO DEL MODELO ENCONTRADO (1-2-3-4-5)	
	VIDEO 1 (29 vehículos)	VIDEO 2 (25 vehículos)
1	1-3	2
2	3-4-5	2
3	1-2-3-4	4-5
4	3-4	1-3
5	3-4	5
6	1-3	4
7	1-3	4
8	1-3-4	5
9	1-4	4
10	1-3	5
11	2-3-4-5	4
12	2-3-4	-
13	1-2-3-4	2
14	1-2-3	-
15	2-4	-
16	1-3	5
17	1-2-3-4	-

18	1-3	5
19	1-3	5
20	2-4-5	5
21	1-3	4-5
22	2-3-4	3-4
23	2-4-5	5
24	1-3	3
25	1-3	4-5
26	1-3	
27	2-4	
28	2-4	
29	1-3	

Tabla 4.1 Resultados del algoritmo de emparejamiento de modelo para dos videos

Ejemplo de Calculo para la efectividad del modelo uno:

$$\% = \frac{\text{número de vehículos emparejados}}{\text{número total de vehículos}} \times 100 \quad (\text{Ec. 4.1})$$

$$\% = \frac{18}{29} \times 100 = 62\%$$

En la siguiente Tabla se analiza el porcentaje de efectividad de cada uno de los modelos y del algoritmo total para cada video.

MODELOS	EFECTIVIDAD (%)	
	VIDEO 1	VIDEO 2
1	62	4
2	41.4	12
3	79.3	12
4	55.2	32
5	13.8	44
ALGORITMO TOTAL	100	84

Tabla 4.2 Efectividad total y parcial del algoritmo de emparejamiento de modelo para dos videos

El Video 2 (*seg_clip1-2_7fps*) como se observará en la ejecución del sistema de control de tráfico vehicular, es de baja calidad con respecto a los demás debido a la mala configuración de los parámetros en la descarga del video a la computadora. Por consiguiente la efectividad indicada para éste, se debe a la poca nitidez de las imágenes que componen el video; además, se observa que los modelos 4 y 5 emparejan a la mayoría de dichas imágenes.

4.2.2 RESULTADOS DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE RUEDAS

En la Tabla 4.3 se indica la distancia entre las ruedas que logró reconocer cada uno de los procesos, para los dos videos bajo análisis y en base a ello analizar el margen de confiabilidad para este algoritmo de reconocimiento vehicular. Si la mencionada distancia es mayor o igual que 90 píxeles (*distancia mínima*), la imagen procesada corresponde a la de un vehículo y por ende la confirmación de este algoritmo.

Posición del vehículo en el video	DISTANCIA ENTRE RUEDAS (píxeles)			
	VIDEO 1 (29 vehículos)		VIDEO 2 (25 vehículos)	
	Proceso1	Proceso2	Proceso1	Proceso2
1	137	-	112	-
2	-	90	Una rueda	Una rueda
3	135	134	Una rueda	-
4	-	-	Una rueda	-
5	132	127	Una rueda	172
6	-	-	189	Una rueda
7	Una rueda	-	-	-
8	-	125	Una rueda	-
9	122	-	Una rueda	-
10	-	-	Una rueda	109
11	Una rueda	-	91	-
12	-	140	Una rueda	-

13	136	-	178	Una rueda
14	224	-	125	124
15	-	97	101	Una rueda
16	140	-	100	166
17	141	135	Una rueda	-
18	151	-	Una rueda	110
19	Una rueda	-	Una rueda	-
20	-	134	Una rueda	Una rueda
21	138	-	174	-
22	-	132	-	-
23	-	125	115	109
24	133	-	140	-
25	118	-	134	-
26	143	-		
27	135	132		
28	-	137		
29	159	-		

Tabla 4.3 Resultados del algoritmo de detección de ruedas para los dos videos

Ejemplo de Calculo para la efectividad del proceso1:

$$\% = \frac{\text{número de vehículos que se han encontrado sus ruedas}}{\text{número total de vehículos}} \times 100 \quad (\text{Ec. 4.2})$$

$$\% = \frac{15}{29} \times 100 = 51.7\%$$

En la siguiente Tabla se analiza el porcentaje de efectividad de cada uno de los procesos y del algoritmo total para cada video.

PROCESOS	EFECTIVIDAD (%)	
	VIDEO 1	VIDEO 2
1	51.7	44
2	41.4	24
ALGORITMO TOTAL	79.3	56

Tabla 4.4 Efectividad total y parcial del algoritmo de detección de ruedas para dos videos

Este algoritmo tiene la necesidad de procesar videos de buena nitidez y calidad, así como también del nivel de iluminación en las distintas locaciones de filmación debe ser lo más regular posible, para de esta manera lograr distinguir mejor los contornos circulares de las imágenes bajo procesamiento.

La distancia entre las ruedas juega un papel importante ya que se puede diferenciar vehículos livianos como: bicicletas y motocicletas, que no entran al reconocimiento del presente proyecto.

En este método de reconocimiento también es importante el número de ruedas encontradas, ya que el programa logra discriminar solamente los círculos que tengan un nivel similar con respecto al suelo, para luego proceder con la medición entre las mismas.

4.2.3 RESULTADOS DEL ALGORITMO DE RECONOCIMIENTO DETECCIÓN DE FORMA

Los resultados obtenidos para este algoritmo se ilustran en la Tabla 4.5, en la cual se indica la distancia horizontal de la partícula que representa a la forma del chasis en cada uno de los procesos, para los dos videos bajo análisis. Por consiguiente analizar la efectividad para este algoritmo de reconocimiento vehicular.

Si la distancia medida es mayor o igual que 170 píxeles (*distancia mínima*), la imagen procesada corresponde a la de un vehículo y por ende la confirmación de este algoritmo. Las cantidades que se encuentran en paréntesis se refieren a que el vehículo es pequeño, por ende el parámetro *distancia mínima* deberá ser ajustado a un valor de 140 para el video en cuestión y así lograr tener mayor número de aciertos.

Posición del vehículo en el video	DISTANCIA DEL CHASIS (píxeles)			
	VIDEO 1 (29 vehículos)		VIDEO 2 (25 vehículos)	
	Proceso1	Proceso2	Proceso1	Proceso2
1	-	207	-	197
2	-	171	-	194
3	199	200	-	169
4	-	228	-	186
5	181	180	-	159
6	-	264	-	165
7	-	258	-	-
8	181	193	-	163
9	175	177	-	224
10	-	280	-	184
11	224	222	-	176
12	203	285	(103)	206
13	-	287	-	185
14	315	320	-	166
15	163	160	-	(142)
16	-	221	160	-
17	214	219	-	245
18	-	287	-	-
19	-	273	-	167
20	224	-	-	(142)
21	-	190	-	167
22	181	249	-	(140)
23	186	191	-	171
24	-	200	-	210
25	-	212	-	(142)

26	-	246
27	214	-
28	194	219
29	-	243

Tabla 4.5 Resultados del algoritmo de detección de forma del chasis para los dos videos

Ejemplo de Calculo para la efectividad del proceso1:

$$\% = \frac{\text{número de vehículos aislados el chasis}}{\text{número total de vehículos}} \times 100 \quad (\text{Ec. 4.3})$$

$$\% = \frac{14}{29} \times 100 = 48.3\%$$

En la siguiente Tabla se analiza el porcentaje de efectividad de cada uno de los procesos y del algoritmo total para cada video.

PROCESOS	EFECTIVIDAD (%)	
	VIDEO 1	VIDEO 2
1	48.3	8
2	93.1	88
ALGORITMO TOTAL	100	92

Tabla 4.6 Efectividad total y parcial del algoritmo de detección de forma para dos videos

Como se observa en la Tabla el porcentaje de efectividad del proceso2 es mayor debido a la presencia de un número superior de vehículos con tonalidad oscura. De igual forma en este caso se requiere que el video bajo procesamiento sea de buena calidad y nitidez para aislar mejor la forma del chasis.

El parámetro “*Traer imagen de fondo a los*” es muy importante ya que ésta al ser lo mas pequeña posible, el proceso de resta de imágenes será más exitoso, debido a que en el transcurso del tiempo las condiciones ambientales y de fondo pueden cambiar drásticamente.

La distancia horizontal de la partícula puede ser calibrada con el parámetro *distancia mínima* para lograr tener mayor número de aciertos, y este valor depende del campo de visión que se tenga en la captura del video.

4.3 RESULTADO TOTAL DEL SISTEMA DE CONTROL VEHICULAR

De los resultados arrojados anteriormente, se observa rápidamente que el sistema de control de tráfico vehicular al estar en modo de Monitoreo y Conteo, puede reconocer y llevar una cuenta de todos los vehículos que compone los mencionados videos.

Por tanto en la Tabla 4.7 se ilustra el porcentaje de efectividad del sistema total obtenidos para cinco diferentes videos (ejecutados con los parámetros por defecto) que se cuenta para la comprobación de su confiabilidad en el conteo de vehículos.

VIDEO	Nombre de la carpeta con secuencia de imágenes	Número de vehículos	Número de vehículos contados	EFFECTIVIDAD (%)
1	<i>seg_clip1-1_7fps</i>	17	15	88.2
2	<i>seg_clip1-2_7fps</i>	25	25	100
3	<i>seg_clip2-1_7fps</i>	8	7	87.5
4	<i>seg_clip3-1_7fps</i>	29	29	100
5	<i>seg_clip3-2_7fps</i>	29	29	100

Tabla 4.7 Efectividad del sistema de control para cinco diferentes videos

Los videos que no detectaron todos los vehículos, fueron capturados en condiciones diferentes a los videos mencionados para la efectividad de cada algoritmo; por lo tanto, será necesario entrar al modo de calibración de parámetros para tener éxito en la ejecución de dichos videos, como se indica en el siguiente ejemplo.

4.4 EJEMPLO DEL PROCESO DE CALIBRACION

Los pasos a seguir para la calibración de cualquier video serán:

- Abrir y seleccionar el video
- Reproducirlo (modo monitoreo y conteo)
- Pulsar el botón C1, para empezar calibrando la señal de Perturbación con los controladores indicados en la Figura 3.36. En este paso el indicador de Perturbación se deberá encender el instante que pase un vehículo. Esta calibración depende del sentido de movimiento de los vehículos y del campo de visión.
- Pulsar nuevamente C1 para retornar al modo de Monitoreo.
- Pulsar el botón C2, para calibrar los parámetros para los algoritmos de reconocimiento con la ayuda de los controladores indicados en la Figura 3.43. Esta calibración depende del campo de visión.
- Pulsar nuevamente C2 para retornar al modo de Monitoreo.
- Pulsar el botón C2, para configurar parámetros de visualización con los controladores indicados en la Figura 3.43.
- Pulsar nuevamente C2 para retornar al modo de Monitoreo.
- Finalmente, en modo normal verificar que el sistema reconoce todos los vehículos que componen el video. Caso contrario repetir la configuración de los parámetros para los algoritmos de reconocimiento.

El video a calibrar en este ejemplo es *seg_clip_2-1_7fps*, el cual corresponde a una filmación de baja calidad por el hecho de que no existe un apoyo fijo de la cámara durante la filmación.

Para los valores de defecto, se tiene la siguiente ventana de calibración que se ilustra en la Figura 4.3, e indica el único vehículo que no detecta el sistema para este video.

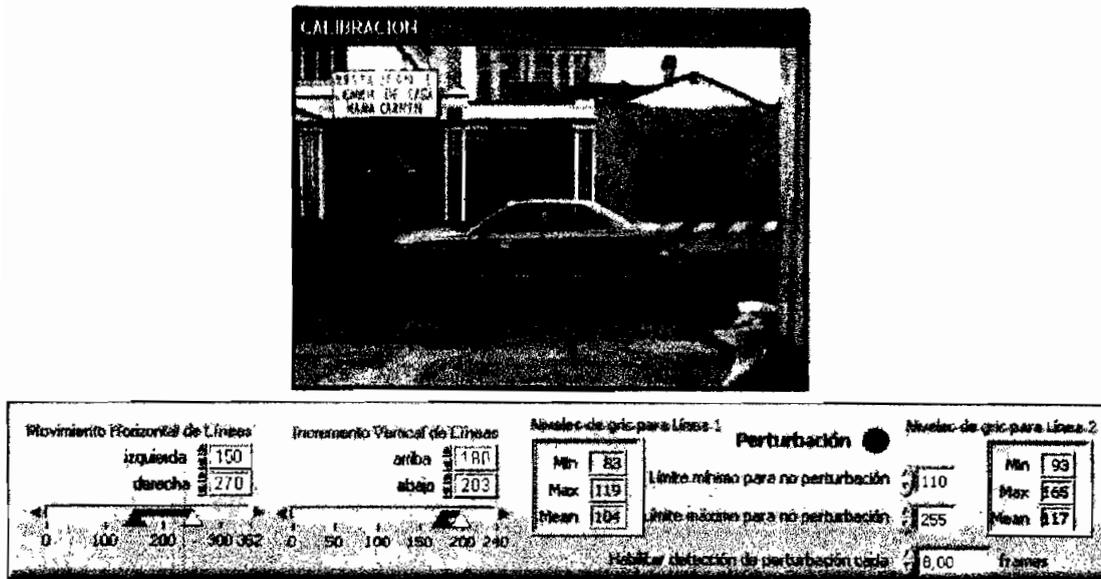


Figura 4.3 Vehículo que no produce Perturbación

En la Figura anterior se observa también que el indicador de perturbación no se enciende en el momento del paso del vehículo, ya que la misma no atraviesa las líneas de medición de ng medio, y ésta no produce una variación significativa que se ajuste al rango de análisis para perturbación.

Por lo tanto, se desplazarán las líneas hacia la izquierda, ya que el sentido del vehículo es de derecha hacia izquierda, también deberán ser subidas para lograr tener "contacto" con el vehículo. Es también importante fijarse en el nivel medio (*Median*) de las dos líneas para, en base a esto ajustar los *límites mínimo y máximo para no perturbación*.

Con los criterios mencionados, pulsando el botón C1 se ajustarán dichos controles que permiten ahora sí la detección de perturbación en el instante del paso del vehículo, como se observa en la Figura 4.4.

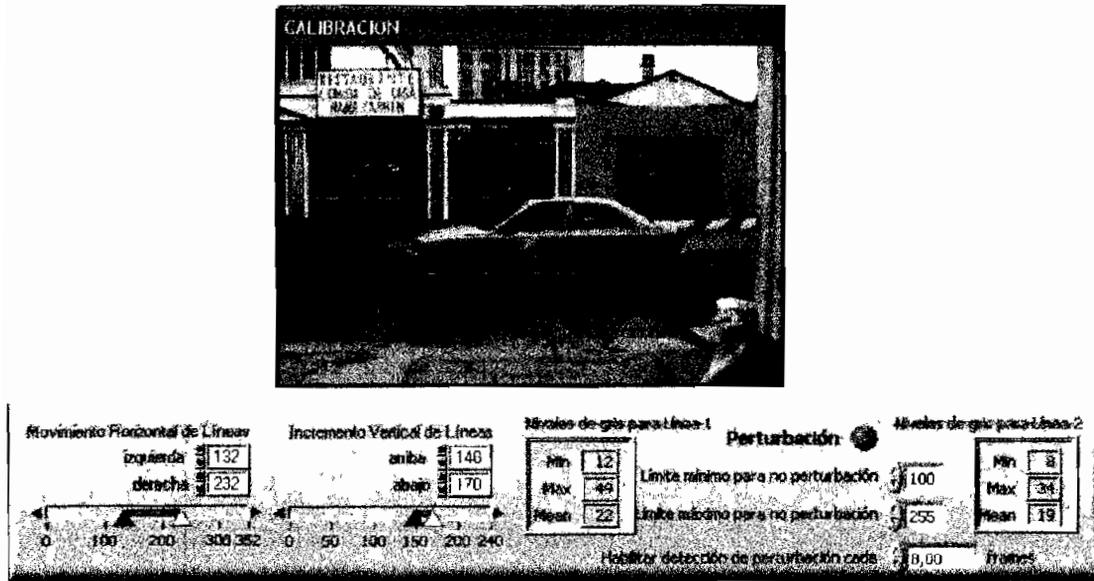


Figura 4.4 Ajuste de controladores para que exista Perturbación

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La extracción de información de secuencias de imágenes es muy importante, ya que puede ser útil en aplicaciones como la de control de tráfico de automóviles, también en aplicaciones de seguridad, peajes y en lugares donde un operario humano puede distraerse con facilidad. El sistema implementado en este proyecto además de ser versátil, eficaz y fácil de utilizar, no tiene los inconvenientes de otros sistemas con respecto a la instalación y generación de ruido debido por ejemplo a campos electromagnéticos.
- La información en el dominio del espacio y el tiempo del video es provista en base a una secuencia de imágenes y el efecto del movimiento es llevado a cabo a través de cambios pequeños y continuos dichas imágenes.
- La herramienta computacional de LabVIEW es muy importante por su entorno de programación gráfica, ya que sus funciones y librerías tienen bloques predefinidos para el procesamiento digital de imágenes. Esto es una ventaja comparado con otros lenguajes de programación que realizan las mismas tareas.
- El presente proyecto al ser implementado en tiempo no real, fue necesaria la segmentación de los videos capturados a secuencias de imágenes para su posterior procesamiento en LabVIEW.
- Para lograr reconocer los vehículos, la posición y estabilidad de la cámara es fundamental porque se requiere un campo de visión apropiado para

aplicar los distintos algoritmos desarrollados para el reconocimiento vehicular.

- Por facilidades en el procesamiento y tiempo de ejecución, todo el sistema fue desarrollado con imágenes en escala de grises con una definición de 8 bits. Además la resolución de las imágenes en cuestión es la misma a lo largo de todo el proceso, y corresponde al tamaño para un tipo de video NTSC con formato MPEG-1.
- La resolución de las ventanas de imágenes están cuantizadas en unidades de píxeles, mas no en unidades del mundo real debido a que la distancia entre la cámara al vehículo no será la misma para las distintas condiciones de filmación y aplicaciones.
- Con la finalidad de no aplicar los algoritmos de detección a todas las imágenes, sino solamente en las cuales se sospeche que puedan contener un vehículo, se diseñó un artificio para sensar en absolutamente todas las imágenes que componen el video, el nivel de gris promedio de dos líneas colocadas a una distancia prudencial. Por consiguiente el cambio más drástico de contraste se da en la presencia de un vehículo y su sombra, produciéndose así la llamada "perturbación".
- Para lograr ser más eficaz al sistema implementado, se diseñaron tres algoritmos diferentes de reconocimiento vehicular, y son aplicados en uno tras de otro. El orden de ejecución de éstos se determinó según las pruebas realizadas a varios videos, y dando prioridad al que dio mayor número de aciertos.
- Para aumentar la velocidad de procesamiento y disminuir los errores, fue conveniente para cada uno de los algoritmos, focalizar una región de interés sobre cada una de las imágenes.

- El algoritmo de emparejamiento de modelo localiza regiones de una imagen que empata con una plantilla determinada representando el objeto de búsqueda. Este método encuentra plantillas a pesar de que exista poca iluminación, ruido, emborronamiento, cambio o rotaciones de plantilla; razón por la cual, se eligieron cinco plantillas de búsqueda que corresponde a una parte esencial e indiscutible de un vehículo.
- En cuanto el proceso de detección de ruedas se utiliza técnicas de morfología binaria, este método requiere inicialmente un mejoramiento de la imagen para lograr el aislamiento de los dos aros del vehículo, para posteriormente medir la distancia entre los círculos y reconocer si la imagen bajo procesamiento corresponde o no a la de un vehículo.
- El algoritmo de detección de forma también hace un buen uso de las técnicas mencionadas por el proceso anterior, al final del cual se tendrá una sola partícula que representa la forma del chasis del vehículo, siempre y cuando dicha distancia horizontal sea mayor al parámetro correspondiente escogido. Un elemento importante para este proceso es la imagen de fondo, la cual es continuamente refrescada con el fin de tener pocos cambios en el entorno del vehículo para lograr mejor aislamiento.
- Por cuanto el sistema de control vehicular puede ser implementado en diferentes sitios, se necesita de ciertos ajustes para su correcto funcionamiento; por tanto, se creó dos modos de operación uno de monitoreo y otro de calibración, el mismo que contiene parámetros indispensable para su satisfactorio desenvolvimiento.

5.2 RECOMENDACIONES

- Se recomienda para futuras filmaciones utilizar cámaras digitales de iguales o mejores características para lograr conseguir una buena calidad de los videos y así aumentar la efectividad de los algoritmos principalmente para las detecciones de ruedas y forma.

- La posición de la cámara de captura debe tener un campo de visión considerable, será colocada siempre en forma perpendicular al sentido de movimiento de los vehículos a una distancia aproximada de 5 metros y a una altura de 1.7 metros para conseguir tener al vehículo completo dentro de la pantalla.
- Para buen desempeño del programa de monitoreo y conteo, en especial el algoritmo de emparejamiento de modelo, los plantillas anteriormente seleccionados deben tener una ubicación específica en la unidad de disco (D:\modelos) para poderlos encontrar sin pérdida de tiempo y agilizar el proceso.
- Con respecto al modo de calibración, funciona mientras esté en ejecución el programa; es decir, cuando se reproducen la secuencia de imágenes. Este modo consta de dos partes, una para los parámetros referentes a la perturbación en la cual se debe tomar muy en cuenta la información de las líneas especialmente en su valor medio cuando cruza un vehículo (para hacer adecuaciones en los límites tanto mínimos como máximos para no perturbación), y la otra usada para configuración de la reproducción a gusto del usuario, así como también para la calibración de las distancias de los algoritmos de detección de ruedas y de forma.
- EL sistema de monitoreo de tráfico vehicular utilizando procesamiento digital de imagen fue hecho de manera modular y puede funcionar en tiempo real, su limitación es la adquisición de la secuencia de imágenes a través de las tarjetas y cámaras industriales. Para un mejor funcionamiento y transferencia de información, ésta debe realizarse mediante la interfaz IEEE 1394 cuyas características fueron mencionadas en el Capítulo 2.
- El sistema actualmente implementado en tiempo no real, se recomienda cambiar el proceso de adquisición de la secuencia de imágenes con la ayuda de los bloques Imaq acquisition, los cuales dependerán del modo de captura. A parte de esto es necesario poseer los drivers respectivos (NI-

IMAQ for IEEE 1394) para configurar el hardware, para este se recomienda la cámara Sony DFW-SX900 y la tarjeta de transferencia PCI FireWire800 1394b, para así lograr el control en tiempo real.

- La cámara Sony DFW-SX900 es la elegida ya que da buenas prestaciones para la transferencia y es compatible con varias tarjetas de adquisición (ver Anexo B).
- La tarjeta para la transferencia de datos de la cámara mencionada hacia el computador es la PCI FireWire800 1394b (ver Anexo C). Compatible con los estándares 400 y 800 (IEEE 1394a/b, o iLink). Dispone además de slot tipo PCI de 64-bit que puede ser instalado en el compartimiento PCI de 32/64-bit (33Mhz). Dispone de conector de alimentación en la propia tarjeta que permite suministrar hasta 1.5A de potencia por puerto. También dispone de 3 puertos FireWire en la tarjeta, dos de ellos del tipo FireWire 800 (conector Bilingual 9-pin IEEE 1394b, compatible con FireWire400), y un conector del tipo FireWire 400 (conector de 6-pin IEEE 1394a). Compatible con entornos Windows (98SE, ME, XP, 2000 o superior). Ideal para aplicaciones de vídeo y conexión de dispositivos que requieran un gran ancho de banda.
- Los drivers con la ayuda de NI Measurement & Automation Explorer, permite configurar fácilmente el hardware respectivo, y así poner a punto el sistema de visión e inmediatamente empezar la adquisición de imágenes.

REFERENCIAS BIBLIOGRAFICAS

[1] Digital Signal Processing / Steven W. Smith / www.DSPguide.com

[2] Teoría de video digital.

- Wanadoo. Zona Técnica. MPEG y Video Digital
<http://perso.wanadoo.es/titirs/vcd.html>
- Zonadvd. Enciclopedia. NTSC / PAL
<http://www.zonadvd.com/modules.php?name=Encyclopedia&op=content&tid=58>
- *Linuca. Rubén Ramos. Formatos de Video*
<http://linuca.org/body.phtml?nIdNoticia=148>

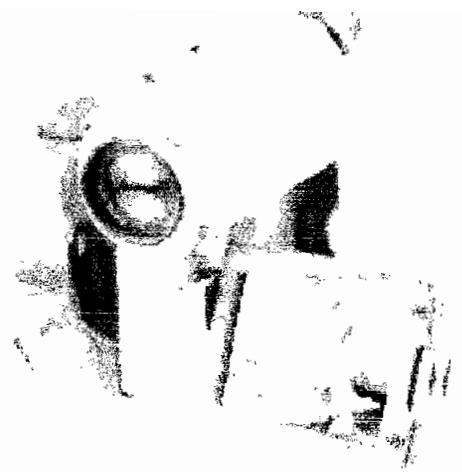
[3] IMAQ Vision 6.0 Concepts Manual

[4] IMAQ Vision 6.0 for LabVIEW Users Manual

[5] Visión Artificial Dinámica. Determinación de movimiento a partir de secuencias de imágenes. / Universidad del País Vasco / Aitzol Zuloaga Izaguirre y José Luis Martín González.

[6] Bus IEEE 1394

- [duiops.net](http://www.duiops.net). IEEE 1394 Firewire
<http://www.duiops.net/hardware/articulo/ie31394f.htm>
- MiniDV. Javier E. Firewire IEEE 1394
<http://www.imagendv.com/firewire.htm>



> 1.0 Megapixel (Gross) Advanced HAD™ CCD Imager

MEGA

(520 lines of horizontal resolution)

MiniDV Recording

Ultra Compact Design

Professional Quality Carl Zeiss® Vario-Sonnar® Lens

10X Optical/20X Precision Digital Zoom

2.5" SwivelScreen™ Touch Panel LCD Display (123K Pixels)

Color Viewfinder (113K)

> Digital Still Memory Made with Memory Stick™ Media

MPEG Movie EX Mode

USB Streaming

i.LINK® DV Interface (IEEE1394)

Progressive Shutter System

Super SteadyShot Picture Stabilization System

Super NightShot Infrared System and Color Slow Shutter

Playback Zoom

Analog-to-Digital Conversion with Pass-Through

Imaging Device:
Video Actual:
Still Actual:
Local Distance:
5mm Conversion:
Filter Diameter:
Optical Zoom:
Digital Zoom:
Focusing:
Minimum Illumination:
LightShot Infrared System:
Shutter Speed:
Memory Mode:
MPEG Movie EX Mode:
Viewfinder:
LCD:
Accessory Shoe:
Video Input/Output:
Audio Input/Output:
LINK DV Interface (IEEE 1394):
USB Terminal:
Memory Stick PRO™ Media Compatibility:
USB Streaming:
Headphone Jack:
Mic Input:
White Balance:
Exposure:
Software OS Compatibility:
Power Consumption (VF/LCD/VF+LCD):
Dimensions (WHD):
Weight:
Supplied Accessories:

worry FREE™ BATTERY LIFE

Battery (fully charged)	Rec. Disp. VF/LCD/VF+LCD	Playback Time (LDC On)*
NP-FM50	1.5h/2.5h/4.0h	1.5h/2.5h/4.0h

MEMORY MODE

Memory Stick Media	Final Size 1152 x 864	Final Size 640 x 480	Memory Stick PRO™ Media	Final Size 1152 x 864	Final Size 640 x 480
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h

MPEG MOVIE EX MODE

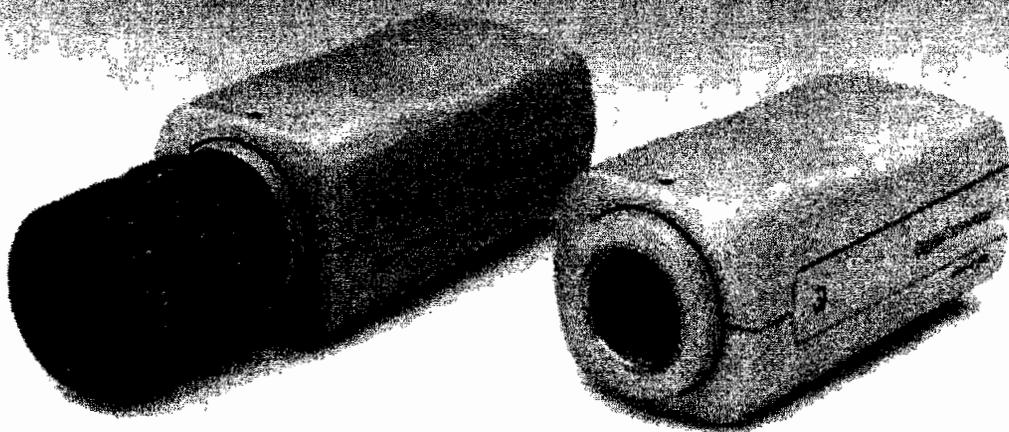
Memory Stick™ Error Mode	Final Size 1152 x 864	Final Size 640 x 480	Memory Stick PRO™ Error Mode	Final Size 1152 x 864	Final Size 640 x 480
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h
MS	1.5h	1.5h	MS	1.5h	1.5h

SONY

The camera is designed to be used with the supplied accessories. For more information, please refer to the user manual. The camera is not intended for use in hazardous environments. The camera is not waterproof. The camera is not for use in areas with high humidity or high salt content. The camera is not for use in areas with high magnetic fields. The camera is not for use in areas with high electromagnetic interference. The camera is not for use in areas with high temperature. The camera is not for use in areas with high pressure. The camera is not for use in areas with high vibration. The camera is not for use in areas with high shock. The camera is not for use in areas with high dust. The camera is not for use in areas with high smoke. The camera is not for use in areas with high fumes. The camera is not for use in areas with high noise. The camera is not for use in areas with high light. The camera is not for use in areas with high radiation. The camera is not for use in areas with high magnetic fields. The camera is not for use in areas with high electromagnetic interference. The camera is not for use in areas with high temperature. The camera is not for use in areas with high pressure. The camera is not for use in areas with high vibration. The camera is not for use in areas with high shock. The camera is not for use in areas with high dust. The camera is not for use in areas with high smoke. The camera is not for use in areas with high fumes. The camera is not for use in areas with high noise. The camera is not for use in areas with high light. The camera is not for use in areas with high radiation.

DFW-SX900 DFW-X700

Component / OEM



OUTLINE

The DFW-SX900 is a high resolution SXGA 1280 x 960 pixel color C-mount camera featuring a 1/2 type progressive scan 1,450,000 pixel CCD. Incorporating an IEEE 1394 digital interface, this advanced digital camera produces new levels of detailed, uncompressed images. Its compact and lightweight design makes the DFW-SX900/X700 easy to install and ideal for a variety of applications such as machine vision, image processing, microscopy and factory automation.

The DFW-X700 incorporates a 1/2 type progressive scan 800,000 pixel CCD and possesses the same features as the DFW-SX900. Both models have square pixels as well as a primary filter which enables high color quality. Moreover, both models incorporate an external trigger with exposure range from 1/20,000 to 2 seconds, allowing the cameras to capture fast moving objects clearly or image still objects in low light.

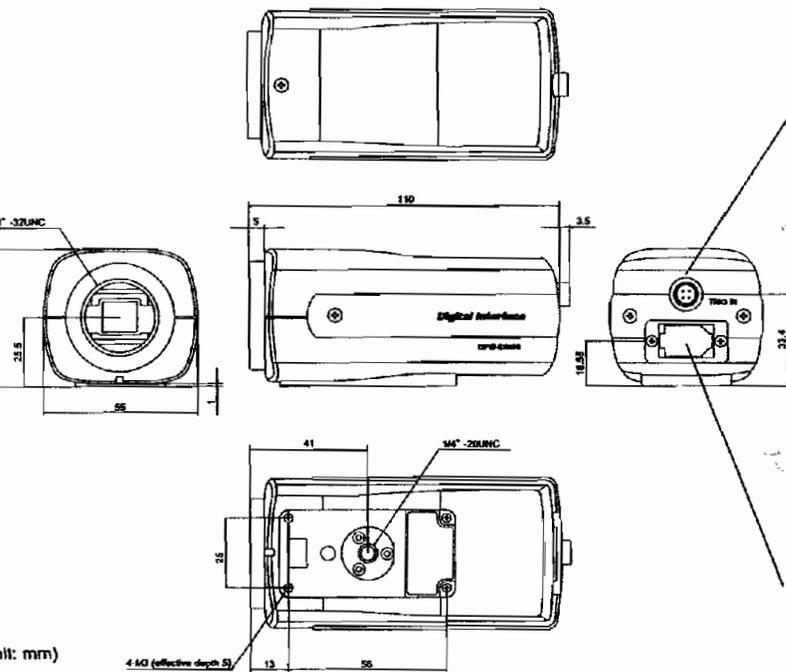
FEATURES

- High resolution video format
 - SXGA 1280 x 960 (DFW- SX900)
 - XGA 1024 x 768 (DFW- X700)
- IEEE 1394 Digital camera specification (Ver.1.30)
- 400 Mbps IEEE 1394 digital interface
- C mount
- External trigger function
- Scalable function
(16 zone selectable – electronic pan/tilt)
Scalable mode allows selection of an area within a full image for output.
- Flange back adjustment

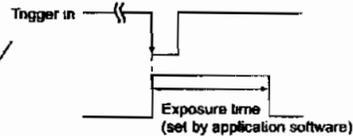
SPECIFICATIONS

	DFW-SX900	DFW-X700
Image device	1/2 type IT progressive scan CCD	
Effective picture elements	Approx. 1,450,000; 1392 (H) x 1040 (V)	Approx. 800,000; 1034 (H) x 779 (V)
Picture size	SXGA (1280 x 960)	XGA (1024 x 768)
Pixel size	4.65 µm (H) x 4.65 µm (V)	6.25 µm (H) x 6.25 µm (V)
Lens mount	C mount	
Minimum illumination	20 lx (F1.4, Gain + 18 dB)	
Digital interface	IEEE 1394-1995	
Transfer rate	400 Mbps/200 Mbps	
Frame rate	7.5 or 3.75 frames/s	15 or 7.5 frames/s
Gain control	Manual (0 - 18 dB)	
Shutter speed	1/20,000 - 2 s	
External trigger shutter	Yes (Trigger mode 0)	
Zone function	16 zones (4 x 4)	
Power requirements	DC 8 - 30 V from IEEE 1394 camera cable	
Power consumption	3.3 W (DC 12 V)	
Dimensions	55 (W) x 50 (H) x 110 (D) mm (2 1/4 x 2 x 4 3/8 inches)	
Mass	250 g (9 oz)	
Operation temp.	- 5 to 45 °C (23 to 113 °F)	
Storage temp.	- 20 to 60 °C (-4 to 140 °F)	
Regulations	UL1950, FCC Class B Digital Device, CE (EN6060-1)	
Supplied accessories	Lens mount cap (1), Clamp filter (2), External trigger connector (male)(1), Operating instructions (1), Cable (1)	

DIMENSIONS



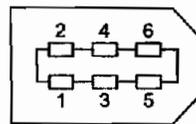
TRIGGER SHUTTER



4 PIN ASSIGNMENT

1	NC
2	GND
3	TRIGIN
4	NC

6 PIN ASSIGNMENT



- 1: Power
- 2: Power (GND)
- 3: TPB-
- 4: TPB+
- 5: TPA-
- 6: TPA+

Sony Electronics Inc. (USA) HQ

1 Sony Drive Park Ridge, NJ 07656

(TEL:+1-800-686-7669)

Sony of Canada Ltd. (CANADA)

115 Gordon Baker Rd, Toronto, Ontario M2H 3R6

<http://www.sony.com/videocameras>

Sony Broadcast & Professional Europe HQ

Schipholweg 275, 1171 PK Badhoevedrop The Netherlands

(TEL:+1-416-499-1414) (FAX:+1-416-497-1774)

(TEL:+31-20-44-99-351) (FAX:+31-20-44-99-333)

<http://www.pro.sony-europe.com>

Germany

Hugo-Eckener-Strasse 20 D-50829 Köln

(TEL:+49-221-5966-322) (FAX:+49-221-5966-491)

France

15, rue Floreal F-75031 Paris Cedex 17

(TEL:+33-1-55-90-41-58) (FAX:+33-1-55-90-42-20)

UK

The Heights, Brooklands, Weybridge, Surrey KT13 0XW

(TEL:+44-990-331122) (FAX:+44-1932-817011)

Nordic

Per Albin Hanssons väg 20 S-214 32 Malmö Sweden

(TEL:+46-40-190-800) (FAX:+46-40-190-450)

Italy

Via Galileo Galilei 40 I-20092 Cinisello Balsamo, Milano

(TEL:+39-2-618-38-431) (FAX:+39-2-618-38-402)

K7603V11W00JUL

Sony online <http://www.sony.co.jp/ISP/>

© 2000 Sony Corporation. All rights reserved.

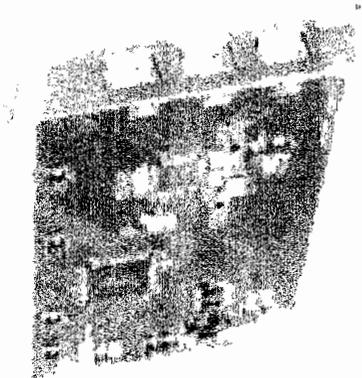
Reproduction in whole or in part without written permission is prohibited.

Design and specifications are subject to change without notice.

Printed in Japan on recycled paper.

Firewire 800 PCI Card

- HP-117526-1000 (1.800.954.0311)



- Installation Guide
- System Requirements
- Package Contents
- Features

CE FC

Installation Guide

For Windows & Mac OS

1. Turn off your computer and unplug the power cable
2. Open the computer case
3. Insert the Firewire 800 Host Card (See Figure 1)
4. Fasten the screw
5. Close the computer case and plug in the power cable
6. Turn on your computer
7. The Firewire 800 PCI Card will be recognized by the system
8. The software driver will be installed automatically
9. Your Host Card is now ready to use (See Figure 2)

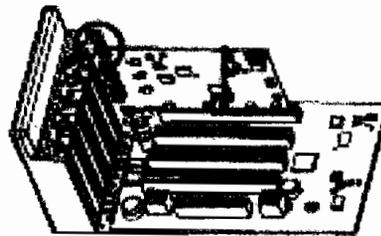


Figure 1

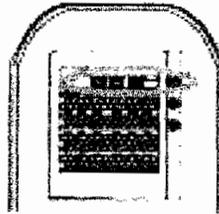


Figure 2

System Requirements

- **PC**

Windows 98SE / ML / 2000 / XP

Minimum of 200MHz processor, 128 MB RAM

33-MHz/64-bit or 33-MHz/32-bit PCI slot*

- **Macintosh**

Mac OS 9.2.2, OSX 10.1.5, 10.2.3 or above

Minimum of 128 MB RAM

33-MHz/64-bit or 33-MHz/32-bit PCI slot*

Package Contents

- FireWire 800 PCI Card x 1
- Instruction Sheet x 1

Features

- Rewireable 800-PCI Card with three external 1394b 9-pin bilingual ports
- Best solutions to connect with 1394b external hard disk and other external 1394b devices
- Serial bus data rates of 100 / 200 / 400 / 800 Mbps
- 33-MHz/64-bit and 33-MHz/32-bit selectable PCI interface
- Backward compatible with 1394a
- Easy to install and use

* If you insert the PCI Card to a 33-MHz/32-bit slot, the transferring data rate will be reduced to 1394a

P/N: AP-BX0388

Version: 1.0