



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

DETECCIÓN Y EXTRACCIÓN DE PLACAS DE VEHÍCULOS EN SEÑALES DE VIDEO

INFORME DE

MATERIA DE GRADUACIÓN

Previo a la obtención de los títulos de:

INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

INGENIERA EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS INFORMACIÓN

PRESENTADA POR:

Robert Armando Vera Santos

Helen Patricia Rodríguez Sánchez

GUAYAQUIL – ECUADOR

2011

AGRADECIMIENTO

*A Dios por estar siempre a mi lado y, brindarme
esos dones espirituales que me han llevado al
camino del éxito.*

A mi madre y a mi familia por brindarme su apoyo.

*A mi novia Rocita, con quien aprendí a distribuir
nuestro tiempo y juntos hemos alcanzado este
triunfo que Dios ha querido que sea posible en su
amor.*

*A mi compañera Helen por su convicción y entrega
a este Proyecto que juntos hemos podido realizar.*

*Al Dr. Boris Vintimilla, Director de este proyecto, por
su entrega incondicional a la docencia e
investigaciones y su valiosa guía en el transcurso
de nuestra formación académica.*

Robert Vera Santos

*Agradezco a Dios por la paciencia y salud, que nos
ha dado durante todo este tiempo.*

*A mis padres quienes han sido y son
un pilar fundamental en mi vida.*

*A nuestros profesores por todos los consejos,
y conocimientos transmitidos.*

Helen Rodríguez Sánchez

DEDICATORIA

*A Dios, a mi madre, mi familia
mi novia Rocita y a todos los
que me han ayudado en la
realización de este trabajo y
también en especial a mi hijo
y mi hermanito Miguelito que
vean en mi un ejemplo a
seguir.*

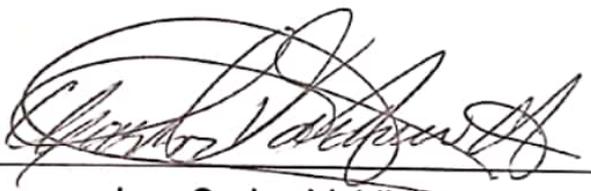
Robert Vera Santos.

*A Dios
A mi madre, quien me
apoyo durante todos mis
años de estudio, y le debo
todo lo que soy
A mi hija, quien quiero que
llegue más lejos que yo.
Helen Rodríguez Sánchez.*

TRIBUNAL DE SUSTENTACIÓN



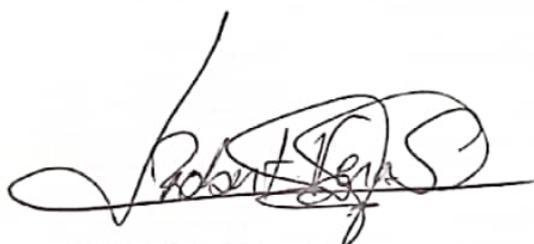
Dr. Boris X. Vintimilla B.
PROFESOR DE LA MATERIA DE GRADUACIÓN



Ing. Carlos Valdivieso
PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este informe de materia de graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".



Robert Armando Vera Santos



Helen Patricia Rodríguez Sánchez

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN EXPRESA	V
ÍNDICE GENERAL	VI
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	XI
1. INTRODUCCIÓN	1
1.1. Resumen	1
1.2. Análisis del problema y diseño de la solución.....	2
1.2.1. Análisis del problema.....	2
1.2.2. Diseño de la solución.....	3
1.3. Objetivos del proyecto	5
1.3.1. General	5
1.3.2. Específicos	5
1.4. Justificación.....	6
1.5. Organización de la tesis.....	7
2. REVISIÓN BIBLIOGRÁFICA.....	10
2.1. Resumen	10
2.2. Reconocimiento automático de matrículas	11
2.3. Historia de los ANPR.....	11
2.4. Tecnología	13
2.4.1. Componentes de un sistema ANPR	13
2.5. Componentes de hardware:.....	15
2.5.1. Sistema de iluminación.....	15
2.5.2. Cámara	15
2.5.3. Digitalizador de imagen	16
2.5.4. Computador	17

2.6.	Componentes de software	17
2.7.	Librerías de desarrollo	18
2.7.1.	OpenCV	19
2.7.2.	VideoInput.....	21
2.7.3.	CvBlobs.....	21
2.8.	Ejemplo de sistema ANPR comercial.....	22
3.	INSTALACIÓN Y CALIBRACIÓN DEL SISTEMA DE ADQUISICIÓN DE IMÁGENES	23
3.1.	Resumen	23
3.2	Introducción.....	24
3.3.	Instalación del software	24
3.3.1.	Librería OpenCV	24
3.3.2.	Instrucciones para instalar OpenCV	27
3.4.	Instalación del hardware	33
3.4.1.	Instalación y montaje de la cámara REG-L1	33
3.4.2.	Accesorios requeridos para la instalación	34
3.4.3.	Definición de un lugar físico para realizar las pruebas	36
3.4.4.	Preparación de equipos para las pruebas	37
3.4.5.	Conexión Cámara-Computador	39
3.5.	Calibración	40
3.5.1.	Tomas de medidas de prueba	40
3.6.	Adquisición de la señal de video.....	47
3.6.1.	Librerías necesarias	47
4.	DETECCIÓN Y EXTRACCIÓN DE PLACAS	51
4.1.	Resumen	51
4.2.	Introducción.....	52
4.3.	Descripción del algoritmo.....	53
4.3.1.	Extracción de parámetros constantes del algoritmo de detección.....	53
4.3.2.	Implementación del algoritmo	54
5.	SEGUIMIENTO DE LA PLACA	59
5.1.	Resumen	59

5.2.	Introducción.....	59
5.3.	Algoritmo de seguimiento (SURF).....	60
5.3.1.	Detección de máximos y mínimos de la escala.....	61
5.3.2.	Localización de puntos de interés	62
5.3.3.	Asignación de la orientación	62
5.3.4.	Descriptor del punto de interés	62
6.	DISEÑO DEL SISTEMA (GUI)	64
6.1.	Resumen	64
6.2.	Implementación del sistema	65
6.3.	Módulo de calibración de la cámara	67
6.3.1.	Datos para pruebas	68
6.3.2.	Datos resultantes	69
6.4.	Módulo de Adquisición y Configuración	71
6.5.	ANPR	74
6.6.	Ayuda	76
7.	RESULTADOS EXPERIMENTALES	77
7.1.	Resumen	77
7.2.	Metodología de las pruebas.....	78
7.3.	Resultados	81
	CONCLUSIONES	
	RECOMENDACIONES	
	BIBLIOGRAFÍA	

ÍNDICE DE FIGURAS

Figura 1-1: Módulos implementados para el proyecto propuesto.....	4
Figura 2-1: Componentes Hardware y Software para ANPR.....	13
Figura 2-2: Interfaz de Carmen	22
Figura 3-1: Diagrama de instalación y calibración de la cámara para la adquisición de imágenes	24
Figura 3-2: Herramientas de OpenCV	25
Figura 3-3: Opciones avanzadas - variables de entorno	27
Figura 3-4: Añadir el path de las dlls	28
Figura 3-5: Archivo/Nuevo/Proyecto/Aplicación de consola Win32	29
Figura 3-6: Selección de las librerías del proyecto.....	31
Figura 3-7: C:\WINDOWS\system32	32
Figura 3-8: Cámara REG-L1	34
Figura 3-9: Fuente de la cámara	34
Figura 3-10: Trípode	34
Figura 3-11: Extensión de corriente	35
Figura 3-12: Ordenador	35
Figura 3-13: Digitalizador de imágenes	35
Figura 3-14: Regulador de voltaje	35
Figura 3-15: Platina.....	36
Figura 3-16: Mesa.....	36
Figura 3-17: Lugar para pruebas	37
Figura 3-18: Calibrar cámara.....	38
Figura 3-19: Detalle de medidas para prueba.....	39
Figura 3-20: Detalles de cables de la cámara	40
Figura 3-21: Especificar los puntos de prueba	41
Figura 3-22: Perspectiva horizontal – Calibración.....	42
Figura 3-23: Perspectiva vertical – Calibración.....	43
Figura 3-24: Detalles de los 24 puntos en el carril.....	44

Figura 3-25: Calibrar la cámara con ángulo de prueba	46
Figura 3-26: Detalle de placa	46
Figura 3-27: Inclusión de librería a visual	48
Figura 3-28: Incluir librería VideoInput al proyecto	49
Figura 3-29: Ejemplo de CvBlobs	50
Figura 3-30: Inclusión de librería CvBlobs a visual studio	50
Figura 4-1: Algoritmo de detección y extracción de placas	52
Figura 4-2: Muestra de imagen original y binaria	58
Figura 4-3: Muestra de la placa detectada	58
Figura 5-1: Algoritmo de seguimiento de la placa	60
Figura 5-2: Matriz Hessiana	61
Figura 5-3: Determinante de la Hessiana	61
Figura 5-4: Ejemplo del reconocimiento mediante SURF	63
Figura 6-1: Códigos por módulo	65
Figura 6-2: Menú del sistema	67
Figura 6-3: Interfaz módulo de calibración	68
Figura 6-4: Introducir nombre del documento	69
Figura 6-5: Archivo Excel generado para la calibración	69
Figura 6-6: Calibración – Datos Resultantes	70
Figura 6-7: Datos Resultado imagen	71
Figura 6-8: Adquisición – Elegir dispositivo	71
Figura 6-9: Código para obtener lista de dispositivos	72
Figura 6-10: Código para obtener imagen de la señal de video	72
Figura 6-11: Configuración de señal de video	73
Figura 6-12: Código para abrir ventana de configuración	73
Figura 6-13: Ejemplo de ROI seleccionado	73
Figura 6-14: Código iniciar ANPR	74
Figura 6-15: Código llamada de la función detección	75
Figura 6-16: Opción de Ayuda	76

Figura 7-1: Ejemplo de imágenes de pruebas por punto	78
Figura 7-2. Video de prueba 1	79
Figura 7-3: Video de prueba 3.....	80
Figura 7-4: Video de prueba 5.....	80
Figura 7-5: Video de prueba 9.....	81
Figura 7-6: Región donde se detecta la placa	81
Figura 7-7: Seguimiento de la placa en el proceso ANPR.....	82

ÍNDICE DE TABLAS

Tabla 3-1: Datos de resultados teóricos	45
Tabla 3-2: Ejemplo de resultados	47
Tabla 7-1: Análisis de características de los blobs	79

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Resumen

El propósito de este proyecto es implementar un sistema que permita detectar y extraer placas de vehículos desde una señal de video, utilizando un sistema de visión artificial.

Analizaremos los distintos problemas que hay en nuestra sociedad en el control de acceso vehicular a los parqueaderos de centros comerciales, estadios, hospitales, etc., se diseñará una solución a partir del análisis del problema, se creará una aplicación que nos permitirá realizar un control vehicular automático para la entrada y salida de vehículos de un parqueadero.

Los objetivos de este proyecto serán la instalación y configuración de un sistema de video vigilancia que permitirá la captura de una señal de video para su posterior análisis y procesamiento para la detección de placas de vehículos.

Adicionalmente este objetivo será complementado con el desarrollo de una aplicación de detección y extracción de la placa de vehículos a partir de la señal de video capturada.

1.2. Análisis del problema y diseño de la solución

1.2.1. Análisis del problema

Al requerir un sistema para la detección y extracción de placas de vehículos en una señal de video, se descubrió un proceso de análisis de las características que tiene una placa, identificar los factores que influyen en el escenario de detección; y los pasos para extraer los resultados; esto es parte de un proceso que necesita un sistema ANPR.

En general, para desarrollar un sistema ANPR se siguen los siguientes pasos:

1. Instalación y calibración de una cámara para la adquisición de una señal de video.
2. Detección y extracción de placas de vehículos desde una señal de video
3. Normalización de imágenes de placa de vehículos a través de correcciones geométricas.
4. Segmentación de placas de vehículos usando cuatro técnicas tales como: umbralización, detección de bordes, técnicas de crecimiento de regiones, y otros afines.

5. Reconocimiento del número de placas de vehículos usando modelos de aprendizaje (OCR convencionales) y otras técnicas similares.
6. Herramienta de análisis estadístico del comportamiento vehicular en un sistema ANPR.

1.2.2. Diseño de la solución

Cuando se analizó el problema en general, se descubrió una aplicación que podía ser ampliamente aprovechada por los distintos parqueaderos de la ESPOL para facilitar el registro automático de los autos que ingresan o salen de un determinado parqueadero.

Actualmente, al ingresar un vehículo a un determinado parqueadero de la ESPOL, para control de seguridad, se anota la fecha, hora y placa de su ingreso; pero esto es una molestia para el conductor esperar a que el guardia escriba los datos respectivos para el control, sin mencionar que pueda existir una demanda de vehículos en el ingreso al parqueadero y tendrán que esperar a que se realice este proceso para cada uno; adicionalmente deben también tener en cuenta los vehículos que van a salir.

Sumando a estos problemas, no se eleva un control estadístico de los diferentes parámetros que se generan al ingresar o salir un vehículo al parqueadero, esto es, vehículos ingresados, disponibilidad de

parqueos, demandas de parqueos en el tiempo, o para prevención de futuros problemas.

Nuestro proyecto se enfocará en los procesos de instalación y calibración de una cámara para la adquisición de una señal de video; y de detección y extracción de placas de vehículos en señales de video.

En estos procesos se obtendrán los parámetros necesarios para la instalación de software y hardware; la calibración para obtener una óptima adquisición de imágenes mediante una señal de video, y el desarrollo de un algoritmo para la detección y extracción de la placa de un vehículo.

Para esto, se dividió la solución en módulos (Figura 1-1):

1. Instalación y calibración del sistema de adquisición de imágenes.
2. Adquisición de la señal de video.
3. Detección y extracción de la placa.



Figura 1-1: Módulos implementados para el proyecto propuesto.

1.3. Objetivos del proyecto

1.3.1. General

El objetivo general de este proyecto es la instalación y configuración de un sistema comercial de video vigilancia para realizar la detección de placa de vehículos, de preferencia a la entrada de algún parqueadero, la cámara REG-L1-816XC-01 será usada para este propósito [6].

El sistema será complementado con el desarrollo de una aplicación para la detección y extracción de la placa de vehículos.

1.3.2. Específicos

En nuestro proyecto tendremos 3 objetivos específicos:

1. Instalación y calibración del sistema de adquisición de imágenes.

Para la instalación se conocerá qué componentes necesitamos para nuestro sistema, ya sea de software y/o hardware. Una vez definidos los componentes, debemos seguir un proceso de pruebas para la calibración de nuestros dispositivos. Este proceso nos permite definir una serie de parámetros útiles para la correcta adquisición de la señal de video, parámetros tales como: las medidas de altura y ángulo que deberá tener la cámara para obtener un correcto enfoque de placas de vehículos, desde

camiones hasta automóviles, distancia de enfoque, resolución de la imagen, son definidas en este objeto.

2. Adquisición de la señal de video.

En la adquisición de la señal de video se describe la librería que se utilizará para obtener las imágenes de la señal de video, también se podrá configurar los parámetros de cámara para mejorar la imagen si es necesario, se define una región de interés sobre la señal de video adquirida, la cual incluye la zona donde estará contenida la placa del vehículo, entre otros parámetros.

3. Detección y extracción de la placa.

Aquí detallaremos los procesos que se van a realizar para el desarrollo del algoritmo para la detección y extracción de la placa de un vehículo; describiendo las librerías que se utilizará y los procesos para identificar las características de una placa.

1.4. Justificación

En parqueaderos o centro comerciales el sistema se puede integrar junto con un control de acceso, esto ayuda a sistematizar el ingreso y salida de vehículos, de igual forma genera mayor seguridad ya que a la salida la cámara y el control de acceso operan juntos, y el ticket que se presenta a la salida, debe coincidir con la placa del vehículo.

Utilizar todos los parámetros de control que se obtienen para generar

modelos de comportamientos al ingreso o salida de un parqueadero. Esto permite detectar o predecir problemas o signos de interés para el control.

Se puede usar en otros ámbitos (no solo en parqueaderos) por ejemplo:

- Como método de recaudación electrónica de peaje en las autopistas de pago.
- Para vigilar la actividad del tránsito en una luz roja en una intersección.
- Para vigilar el exceso de velocidad en las carreteras.

1.5. Organización de la tesis

Esta tesis ha sido organizada tal como se detalla a continuación:

En el primer capítulo se dará una introducción acerca del proyecto analizando el problema y diseñando la solución con sus respectivos objetivos y justificaciones.

En el segundo capítulo daremos una revisión bibliográfica al reconocimiento automático de placas de vehículos (ANPR) su historia, las diversas nuevas tecnologías que surgen día a día y las librerías de desarrollo que usaremos en la aplicación.

En el tercer capítulo se describe el proceso de instalación de los equipos de software y hardware que se usaron en el desarrollo de nuestro proyecto.

Adicionalmente, se describe el módulo de calibración de estos equipos para la correcta adquisición de las imágenes a partir de su correspondiente señal de video.

En el cuarto capítulo tendremos la descripción del algoritmo desarrollado para detectar y extraer la placa de un vehículo a partir de la señal de video previamente capturada.

En el quinto capítulo describiremos el algoritmo de seguimiento de la placa en la señal del video y la librería SURF.

En el sexto capítulo se describe la implementación de una interfaz gráfica (GUI) para el sistema. Esta GUI me permite la integración de los módulos de calibración, detección y extracción de la placa en un único sistema que facilita la comunicación entre el usuario y la aplicación.

El séptimo capítulo describe las pruebas realizadas en tiempo real de un vehículo en movimiento, con diferentes escenarios. Como resultado de las pruebas se incluye el alcance de la detección en una región determinada.

Finalmente, en el octavo capítulo, se presentan las conclusiones obtenidas, y se plantean recomendaciones de futuros trabajos relacionados.

CAPÍTULO 2

2. REVISIÓN BIBLIOGRÁFICA

2.1. Resumen

En este capítulo se realizó una breve revisión bibliográfica todo acerca de la historia de reconocimiento automático de matrículas (ANPR), librerías de desarrollo como OpenCV, conceptos, componentes de hardware y software que vamos a utilizar en nuestro proyecto.

Usaremos las diferentes tecnologías en:

Hardware:

Sistema de iluminación, cámara, digitalizador de imágenes y computador.

Software:

Un algoritmo de siete etapas principales de un software ANPR:

1. Adquisición de la señal de video.
2. Localización de la matrícula.
3. Orientación y tamaño de la matrícula.
4. Normalización.
5. Segmentación de los caracteres.
6. Reconocimiento óptico de caracteres.
7. Análisis sintáctico y geométrico.

2.2. Reconocimiento automático de matrículas

El reconocimiento automático de matrículas (*Automatic number plate recognition* o *ANPR* en inglés) es un método de vigilancia que utiliza reconocimiento óptico de caracteres en imágenes para leer las matrículas de los vehículos.

El ANPR se le conoce a veces con otros términos tales como:

- **Identificación automática de vehículos** (*Automatic vehicle identification, AVI*)
- **Reconocimiento de matrículas de vehículos** (*Car plate recognition, CPR*)
- **Reconocimiento de matrículas** (*License plate recognition, LPR*)
- **Reconocimiento automático de licencia de matrículas de vehículos** (*Automatic license plate recognition ALPR*)

2.3. Historia de los ANPR

El reconocimiento automático de número de placa (ANPR), es la tecnología de vigilancia que se desarrolló en 1976 en el Reino Unido y la utilización en la policía se inició en 1979.

Ésta tecnología utiliza un software de reconocimiento óptico de caracteres para escanear imágenes y leer las matrículas de los vehículos. Se está utilizando en países como Reino Unido, EE.UU., Italia, Alemania y muchos países europeos. Estas cámaras de reconocimiento, también se utilizan como cámaras de velocidad y en la gestión del tráfico.

Muchas aplicaciones pueden tomar ventaja del uso de estos sistemas. Así por ejemplo, el reconocimiento automático de placa de matrícula aumenta las posibilidades de recuperación de vehículos robados, ya que puede comprobar y verificar con una base de datos mucho más rápido que el proceso manual. En algunos casos, las compañías de seguros están ayudando en la compra de estos sistemas ya que es en última instancia, beneficioso para ellos.

Los ANPR fueron evolucionando y desde el año 2005 estos sistemas han podido escanear las matrículas de vehículos con velocidades de hasta 160 km/h.

Los actuales sistemas de ANPR pueden escanear matrículas con un rango de distancias de captura de entre 5 y 50 metros según el modelo de cámara, como las REG, que pueden registrar imágenes de matrículas de vehículos que circulen a una velocidad máxima de 193 km/h (120 mph).

2.4. Tecnología

2.4.1. Componentes de un sistema ANPR

Los componentes de hardware de los sistemas ANPR básicamente están compuestos por una PC donde almacenará las imágenes en una base de datos, una tarjeta digitalizadora que estará integrada en el case o se la puede adquirir, para que la señal analógica sea transformada a señal digital y lo más importante la cámara con la que tomará las imágenes de las placas de los vehículos como lo muestra la Figura 2-1.

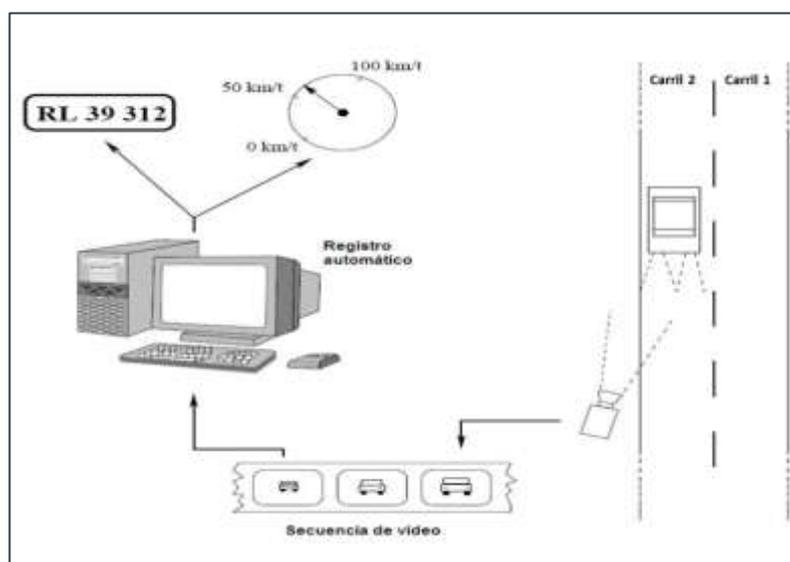


Figura 2-1: Componentes Hardware y Software para ANPR

Por otro lado, el componente de software de un sistema ANPR se ejecuta sobre un hardware de PC estándar y puede ser enlazado con otras aplicaciones o bases de datos. Este componente primero utiliza una serie de técnicas de manipulación de la imagen original capturada, esto es:

- Detecta la placa del vehículo.
- Normaliza, realza y segmenta la imagen de la placa.
- Aplicar OCR para extraer la información alfanumérica de la matrícula.

Los sistemas ANPR/ALPR se pueden utilizar de dos modos:

1. Permite que el proceso sea realizado en su totalidad en el lugar de la adquisición de las imágenes en tiempo real.
2. Transmite las imágenes capturadas desde la cámara a un computador remoto que realiza el proceso de OCR más tarde.

Cuando se realiza la toma de imágenes en el sitio donde está la cámara, la información capturada de la matrícula: alfanumérica, fecha y hora, identificación del lugar y cualquier otra información que se requiera, es completada en unos pocos milisegundos; después ésta información es convertida en pequeños paquetes de datos, se puede transmitir fácilmente a algún ordenador remoto para un posterior procesamiento en caso de que sea necesario, o ser almacenado para ser recuperada posteriormente.

Por otro lado si el proceso de OCR es realizado en un computador remoto, típicamente hay una gran cantidad de PC usados en una estación de servidores para manejar altas cargas de trabajo.

A menudo en dichos sistemas existe la necesidad de remitir imágenes al servidor remoto y éste puede requerir medios de transmisión con un gran ancho de banda.

2.5. Componentes de hardware:

2.5.1. Sistema de iluminación

Estos sistemas a menudo utilizan iluminación infrarroja para hacer posible que la cámara pueda tomar imágenes en cualquier momento del día.

Puede variar el sistema de iluminación según el tipo de cámara, como por ejemplo la cámara de tipo REG usan: Infrarrojos y tecnología de rechazo de ambiente mejorados para una captura de matrículas consistente y fiable las 24 horas del día, 7 días a la semana.

2.5.2. Cámara

Las cámaras ANPR's usan una tecnología de video vigilancia diseñadas para supervisar una diversidad de ambientes y actividades. Están equipadas con ingeniería de precisión para no comprometer la

seguridad, la vigilancia de vehículos ni las aplicaciones de sistemas inteligentes de transporte, también usan una óptica avanzada con iluminación de infrarrojos y tecnología de rechazo de ambiente.

La cámara BOSCH REG-L1 que usamos en nuestro proyecto tiene las siguientes características:

- Tecnología de imagen DHC para obtener imágenes con gran contraste.
- LED metafásico para una potente iluminación de infrarrojos
- Lente de 16 mm
- Captura de 5 a 8 m (16 a 26 pies).
- Video Standard CCIR (PAL)
- Captura permanente tanto de día como de noche.
- Tecnología de rechazo de ambiente avanzada.
- Funcionamiento con uso eficiente de la energía de 12 a 24 VCA/VCC.
- Carcasa compacta; diseño sencillo para el instalador.

2.5.3. Digitalizador de imagen

El digitalizador de imágenes (*Frame Grabber* en inglés) es una tarjeta de hardware cuyo propósito es recibir una señal analógica estándar de vídeo y transformar la señal en una imagen digital tal que el computador la pueda entender (*proceso de digitalización*).

2.5.4. Computador

Esta es una herramienta que nos ayudará a procesar o almacenar los datos de las imágenes que obtenemos mediante la cámara REGL1, está a su vez será procesada en una aplicación utilizando la librería OpenCV.

2.6. Componentes de software

En general existen siete etapas principales que un software de ANPR necesita para identificar una matrícula:

1. Adquisición de la señal de video: captura la señal de video analógica de la matrícula y la digitaliza en una imagen.
2. Localización y extracción de la placa de matrícula: responsable de encontrar y aislar la región que representa la placa de matrícula en la imagen.
3. Orientación y tamaño de la matrícula: compensa los ángulos que hacen que la matrícula parezca rotada y ajusta las dimensiones al tamaño requerido.
4. Normalización: ajusta el brillo y el contraste de la imagen.

5. Segmentación de los caracteres: encuentra los distintos caracteres presentes en la matrícula.
6. Reconocimiento óptico de caracteres.
7. Análisis sintáctico y geométrico: comprueba los caracteres encontrados y sus posiciones con las reglas específicas del país al que pertenece la matrícula.

Durante la etapa de normalización algunos sistemas utilizan técnicas de detección de borde para aumentar la diferencia en la imagen entre las letras y el fondo de la placa, también se puede utilizar un filtro digital de punto medio para reducir el "ruido" visual de la imagen.

2.7. Librerías de desarrollo

A continuación se detallan las librerías usadas en nuestro proyecto:

1. OpenCV
2. VideoInput
3. CvBlobs

Estas librerías son usadas para el proceso de detección y extracción de placa de vehículo en una señal de video.

OpenCV, es usado durante todo los procesos que realiza nuestro sistema, es principalmente una librería que implementa algoritmos para las técnicas de calibración (calibración de la cámara), detección

de rasgos, para rastrear (flujo óptico), análisis de la forma (geometría, contorno que procesa), análisis del movimiento (plantillas del movimiento), reconstrucción 3D (transformación de vistas), segmentación de objetos y reconocimiento (histograma, etc.).

VideInput, es usado para la adquisición de la señal de video, mediante esta se podrá obtener las imágenes que produce, y para después analizarlas. Esta librería es usada principalmente para identificar los dispositivos que se encuentra conectados en el computador, y escoger la configuración que utilizará la cámara.

CvBlobs, es usado en el algoritmo de detección de la placa, para identificar objetos blancos, en una imagen binaria, dándonos el área, la posición, y la media de este objeto, ayudándonos en el proceso de filtrado para identificar a la placa.

2.7.1. OpenCV

2.7.1.1. Información de la historia de OpenCV.

OpenCV (*Open Source Computer Vision Library* en inglés) es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos.

Como metas el proyecto pretende proveer un "*Tool-Kit*" o Marco de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código c y c++ optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo.

OpenCV puede además utilizar el sistema de las primitivas de rendimiento integradas de Intel. Que es un conjunto de rutinas de bajo nivel específicas para procesadores Intel (IPP).

2.7.1.2. Usos más frecuentes.

Según Wikipedia, algunos usos de ANPR son:

- Sección de control, para medir la velocidad promedio de un vehículo sobre ciertas distancias.
- Embargos de vehículos
- En las estaciones de gasolina, cuando un motociclista se va sin pagar el combustible.
- Para gestión de tráfico, para determinar el flujo de tráfico, cuánto tarda un vehículo en pasar.
- Reconocimiento de clientes o huéspedes, para la mejora del servicio al cliente.

2.7.2. VideoInput

La librería OpenCV – HIGHGUI, nos origina funciones para obtener las imágenes de una señal de video; pero es muy limitada; no reconoce algunas cámaras, por causa de su tipo de conector. Otro problema es que no es fácil la configuración de los parámetros de la cámara, y algunas veces no tiene la función para la configuración de algunos parámetros.

VideoInput se especializa en la adquisición de imágenes de una señal de video por cualquier tipo de cámara, es sólo para Windows.

Esta librería es usada internamente en la librería OpenCV, en la carpeta de 3rd party, es fácil de usar y configurar los parámetros de la cámara, usando el sistema Windows Driver Kit (WDK) el cual contiene el Windows Driver Device Kit (DDK), necesario para el uso de configuraciones de dispositivos en Windows.

2.7.3. CvBlobs

Esta librería la usamos en nuestro proyecto en el proceso de detección de placas, es una librería para visión de computadora para detectar regiones en imágenes digitales binaria, y extracción de características.

2.8. Ejemplo de sistema ANPR comercial.

CARMEN, sistema de identificación para vigilancia de tráfico, recaudación de peaje, gestión del tráfico. Creado por A.R. Hungary Inc. de Hungría.



Figura 2-2: Interfaz de Carmen

Este producto el cual se basa en reconocimiento de caracteres por medio de redes neuronales, nos permite identificar las placas de los automóviles convirtiéndolas a texto para posteriormente ser utilizadas como mejor sea conveniente, y está altamente probada en más de 60 países con miles de sistemas instalados a la fecha, no incluye Ecuador. (Figura 2-2).

CARMEN sólo reconoce imágenes tipo BMP, JPEG, JPEG2000, con una resolución máxima de 768 x 288 pixeles.

CAPÍTULO 3

3. INSTALACIÓN Y CALIBRACIÓN DEL SISTEMA DE ADQUISICIÓN DE IMÁGENES

3.1. Resumen

En este capítulo se detallan todas las instalaciones que se deben realizar tanto a nivel de software y hardware del sistema. Además se describe el proceso de calibración de los equipos para la toma de medidas de pruebas de las imágenes.

En la instalación del software se explicará que programas y librerías se requieren para realizar la aplicación de captura de la señal de video y la aplicación que permitirá detectar y extraer la placa del vehículo.

En instalación del hardware se explicará que equipos se requieren para el montaje de la cámara de video vigilancia.

En la calibración se definirá las medidas de distancia entre la cámara y la región de interés, entre la cámara y el carril, el ángulo que deberá tener la cámara para enfocar correctamente a la placa de cualquier tipo de vehículo, así como la altura de la cámara. Estas medidas serán dadas según los datos del catálogo de la cámara, para una detección óptima.

3.2 Introducción

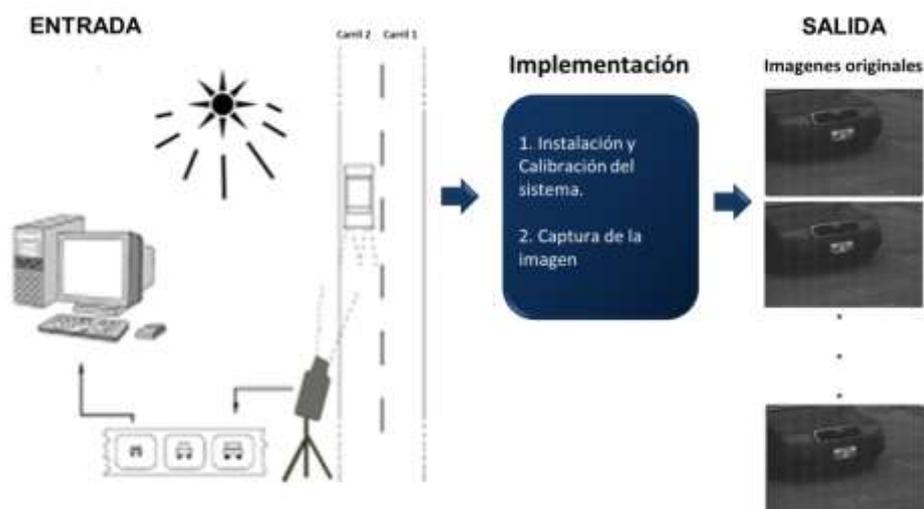


Figura 3-1: Diagrama de instalación y calibración de la cámara para la adquisición de imágenes

En este capítulo describiremos el proceso de la instalación del software y hardware del sistema, y calibraremos la cámara de video vigilancia para un carril, capturando la imagen usando una plantilla de 24 puntos. Proceso como muestra la Figura 3-1.

3.3. Instalación del software

3.3.1. Librería OpenCV

OpenCV es una librería de código abierto desarrollada por Intel. Esta librería proporciona funciones de alto nivel para procesamiento de imágenes (Figura 3-2). Permite a los programadores crear aplicaciones poderosas en el dominio de la visión artificial. OpenCV

ofrece muchos tipos de datos de alto-nivel como árboles, gráficos, matrices, etc.



Figura 3-2: Herramientas de OpenCV

Las funciones de OpenCV que nos permiten manipular las imágenes obtenidas de una señal de video.

3.3.1.1. Librería HIGHGUI

HIGHGUI nos permite abrir ventanas para mostrar las imágenes, leer, escribir archivos relacionados con los gráficos (tanto de imágenes como de vídeo), manejar el puntero del ratón, y los eventos de teclado. También podemos utilizarlo para crear otros reguladores y luego añadirlos a nuestras ventanas.

Desde nuestra perspectiva inicial, la biblioteca HIGHGUI en OpenCV se puede dividir en tres partes: la parte de hardware, la parte del sistema de archivos, y la parte de interfaz gráfica de usuario.

La parte de hardware se refiere principalmente a la operación de las cámaras. En la mayoría de sistemas operativos. HIGHGUI permite una forma fácil de consultar una cámara y recuperar la última imagen de la cámara.

La parte del sistema de archivos se ocupa principalmente de cargar y guardar imágenes. Una característica interesante de la biblioteca es que nos permite leer vídeo usando los mismos métodos que se utilizan para leer una cámara. Podemos por lo tanto abstraer del dispositivo en particular que estamos usando y seguir adelante con la escritura de código interesante.

La parte de interfaz gráfica de usuario, permite crear y manipular ventanas donde pueden aparecer imágenes o videos, y en dicha ventana se puede introducir formas geométricas y controles de usuario.

HIGHGUI nos proporciona un par relativamente universal de funciones para cargar y guardar imágenes fijas.

3.3.2. Instrucciones para instalar OpenCV

Primero para instalar OpenCV en nuestro ordenador debemos descargarnos la versión OpenCV 2.1 en la página oficial de la librería OpenCV:

1. <http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.1/>, OpenCV-2.1.0-win32-vs2008.exe [2011].
2. Tener el Visual Studio 2008 Express Edition instalado.
3. Añadir la librería al sistema operativo.

Para añadir al sistema, debemos ingresar al PATH la librería de la siguiente manera:

- Panel de control/Sistema → cambiar configuración
- Opciones avanzadas → variables de entorno, (Figura 3-3)

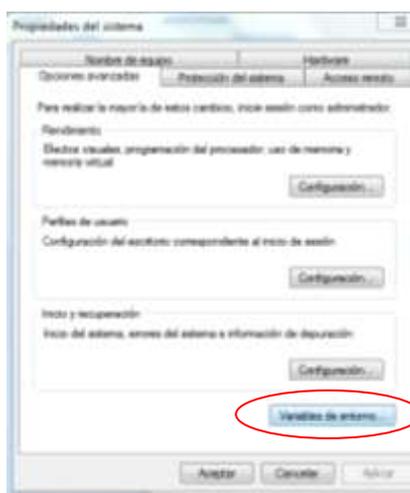


Figura 3-3: Opciones avanzadas - variables de entorno

- Editar la variable PATH y añadir el *path* de las *dlls*:
“C:\OpenCV2.1\bin\”, (Figura 3-4)

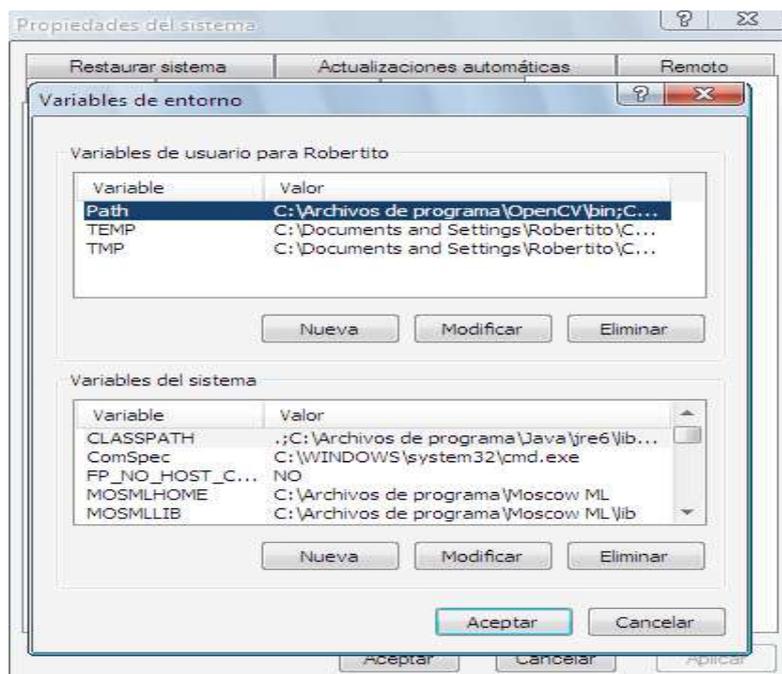


Figura 3-4: Añadir el path de las dlls

Ahora empezamos abrir el Visual C++ y seleccionamos:

Archivo/Nuevo/Proyecto

En la ventana que aparece seleccionar, (Figura 3-5)

- *Win32 y Aplicación de consola Win32*
- El nombre del proyecto “OpenCV”

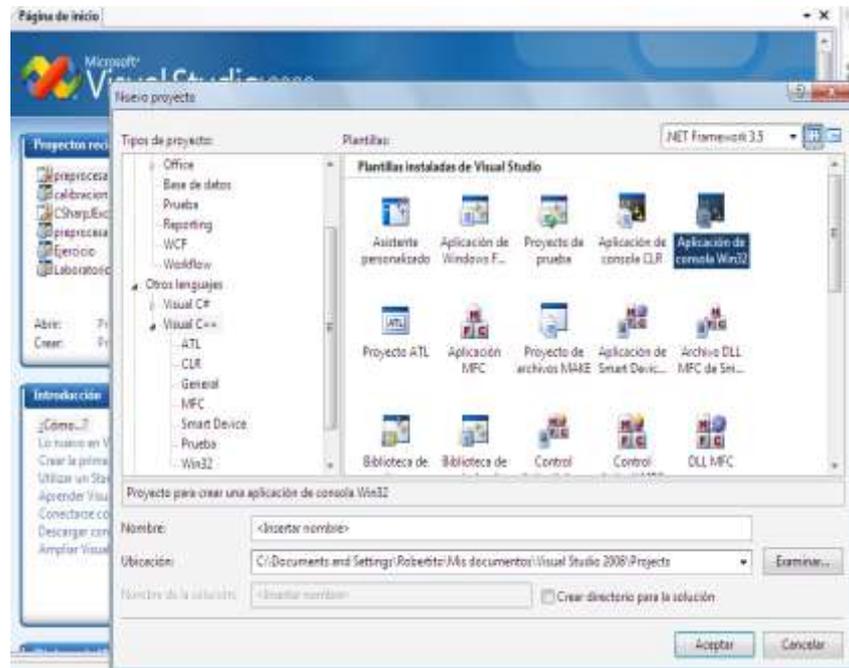


Figura 3-5: Archivo/Nuevo/Proyecto/Aplicación de consola Win32

debemos agregar las librerías, inclusiones y de biblioteca al visual studio, la cual lo realizamos de la siguiente forma:

Añadimos en el Vc++ los subdirectorios:

- Herramientas/opciones/Proyectos y soluciones/Directorio de Vc++.

Archivos de inclusión:

- C:\OpenCV2.1\include
- C:\OpenCV2.1\include\opencv
- C:\OpenCV2.1\3drparty\include

- C:\Escritorio\Proyecto6.0\interfaz\interfaz\Blob

Archivos de biblioteca:

- C:\OpenCV2.0\lib

Adicionalmente, hay que agregar las librerías en el proyecto.

En la barra Menú → Proyecto → Propiedades de Proyecto

En la categoría de propiedades de configuración, ir a:

vinculador/entrada, seleccionamos las librerías con las que se va a trabajar, (Figura 3-6). Para modo *debug*:

- cxcore210d.lib
- cv210d.lib
- highgui210d.lib
- cvaux210d.lib
- ml210d.lib

Para modo *release*:

- cxcore210.lib
- cv210.lib
- highgui210.lib
- cvaux210.lib
- ml210.lib

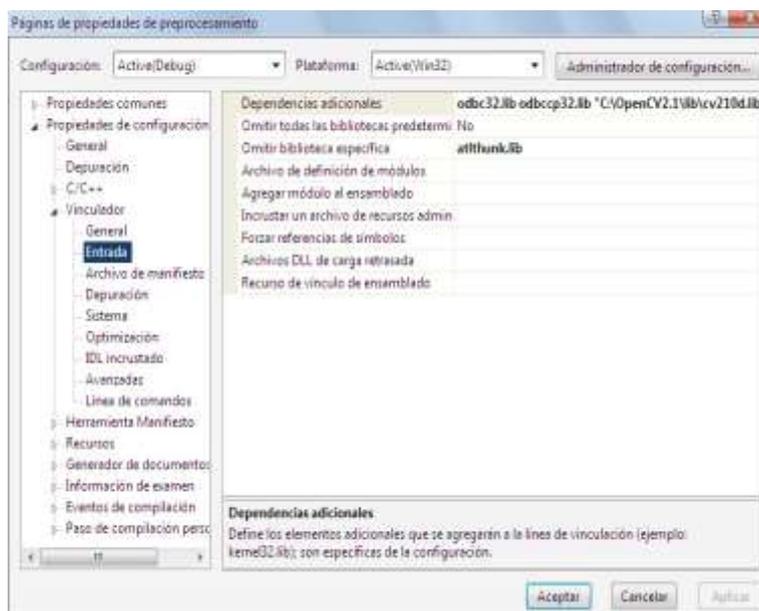


Figura 3-6: Selección de las librerías del proyecto

Una vez ya instalado las librerías se deben colocar librerías adicionales en la carpeta C:\WINDOWS\system32 estas son: cv210d.dll, cv210.dll, cvaux210.dll, cvaux210d.dll, cvcam100.dll, cxcore210.dll, cxcore210d.dll, highgui210.dll, highgui210d.dll, ml210.dll, ml210d.dll (Figura 3-7).

Esto permitirá el correcto funcionamiento de nuestro sistema y sin ningún tipo de inconvenientes a lo que queremos desarrollar.

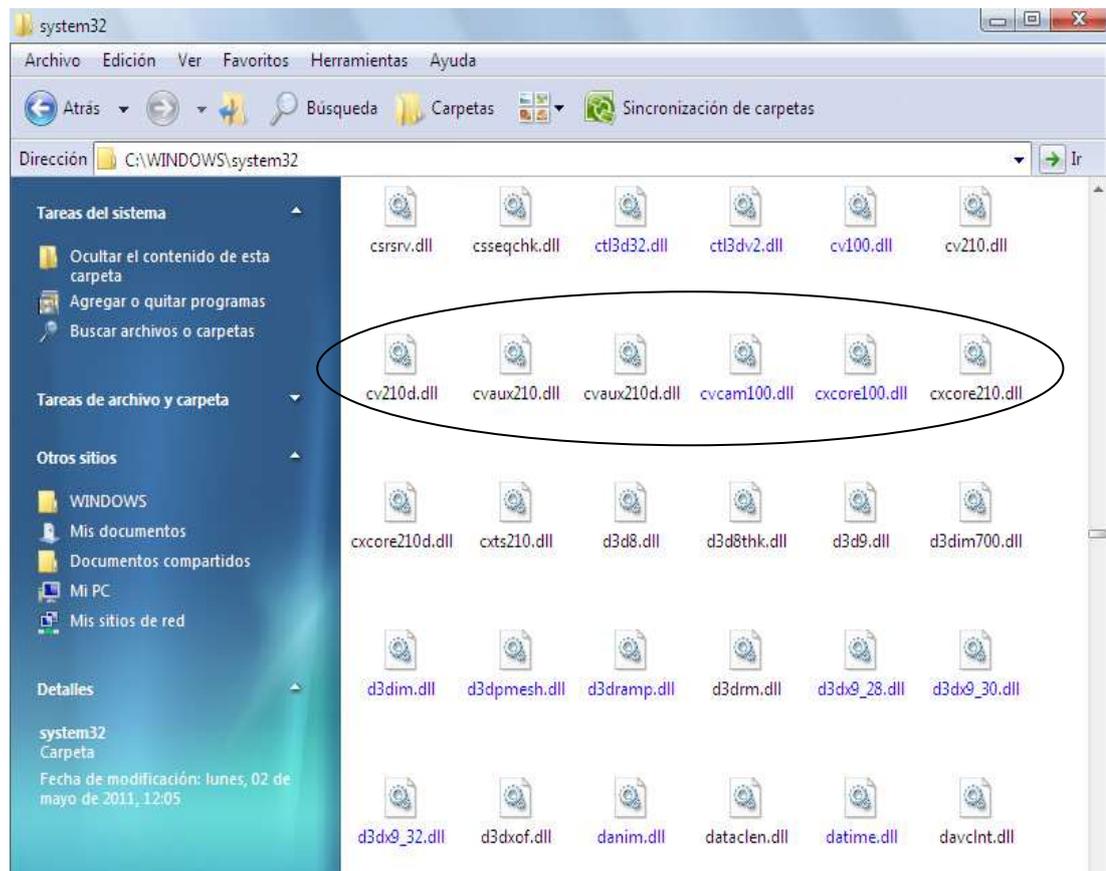


Figura 3-7: C:\WINDOWS\system32

3.4. Instalación del hardware

3.4.1. Instalación y montaje de la cámara REG-L1

El sistema REG-L1, con la innovadora tecnología de imagen DHC, proporciona un rendimiento avanzado y representa la próxima generación en soluciones de captura de matrículas.

La cámara REG-L1 integra avances en óptica, electrónica, iluminación de infrarrojos para proporcionar una captura constante y fiable de las matrículas en las condiciones más complejas.

Nosotros usaremos la REG-L1-816XC-01 (Figura 3-8), el cual trabaja con un rango de visión de 5 a 8m (16 a 26 pies), el dispositivo REG-L1 proporciona imágenes de gran contraste de las matrículas en todos los tipos de condiciones de luz ambiental, desde la oscuridad total hasta la luz directa del sol y los faros de los vehículos.

La carcasa de la unidad es hermética y preparada para funcionar con todo éxito en entornos extremos. La matriz de LED metafásico de alta eficacia de REG-L1 elimina la necesidad habitual de cambiar los focos. Un diseño electrónico perfeccionado permite el funcionamiento con un uso eficiente de la energía con 12 a 24 VCA/VCC.



Figura 3-8: Cámara REG-L1

3.4.2. Accesorios requeridos para la instalación

- Fuente de voltaje de 12VAC - 24VDC para la cámara.



Figura 3-9: Fuente de la cámara

1. Trípode para soporte de la cámara



Figura 3-10: Trípode

2. Extensión de corriente.



Figura 3-11: Extensión de corriente

3. Ordenador.



Figura 3-12: Ordenador

4. Digitalizador de imágenes (*Frame Grabber* en inglés)



Figura 3-13: Digitalizador de imágenes

5. Regulador de Voltaje



Figura 3-14: Regulador de voltaje

6. Platina (ayuda como soporte de fijación de la cámara en el trípode).



Figura 3-15: Platina

7. Mesa para apoyar los equipos



Figura 3-16: Mesa

3.4.3. Definición de un lugar físico para realizar las pruebas

Ahora teniendo los equipos listos realizamos el montaje adecuado de los mismos para la adquisición de las imágenes. Para esto debemos colocar la cámara en un sitio que esté libre de accidentes ya sea este con objetos o la interferencia no intencional del personal. (Figura 3-17)



Figura 3-17: Lugar para pruebas

3.4.4. Preparación de equipos para las pruebas

Instalamos los implementos eléctricos que darán energía al computador y a la cámara a través de un regulador de voltaje para que no sufran daños los equipos.

Se fija la cámara en el trípode y a su vez el trípode en una superficie de montaje capaz de soportar el peso de la cámara.

La cámara REG-L1 tiene un campo de visión horizontal de aproximadamente 16 pies a la distancia óptima de captura.

El ángulo máximo de la cámara para un carro es de 40° , esto es tanto horizontal como verticalmente, esta información la obtuvimos mediante el catálogo de la cámara REG-L1. Para medir los ángulos tanto vertical

como horizontalmente lo medimos con un inclinómetro como lo vemos en la Figura 3-18. Esto limita la cantidad de inclinación de las letras del número de placa. Si las letras están demasiadas inclinadas comenzará a ser irreconocible y reducirá las tasas de reconocimiento automático de software.



Figura 3-18: Calibrar cámara

Si el máximo rango es excedido las letras se hacen más pequeñas y más difíciles de leer. En el rango máximo el ancho de la placa cubre aproximadamente el 12% del ancho de la pantalla (según el catalogo de la cámara).

En nuestro proyecto dividimos el carril en 24 puntos para probar las diversas distancias en donde la cámara puede capturar la placa del vehículo.

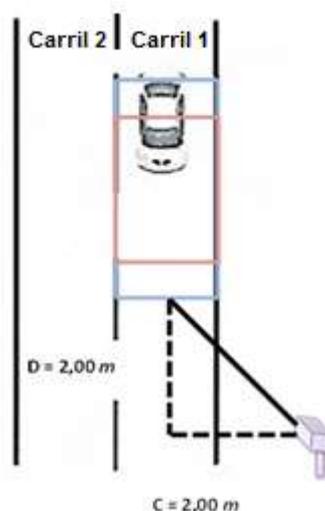


Figura 3-19: Detalle de medidas para prueba

3.4.5. Conexión Cámara-Computador

Una vez que tenemos todo medido con la cámara conectamos los cables BNC (*Bayonet Neill-Concelman* en inglés) de nuestra cámara a nuestra tarjeta de video conectada al computador y también los cables de alimentación de energía a la fuente de voltaje. (Figura 3-20)

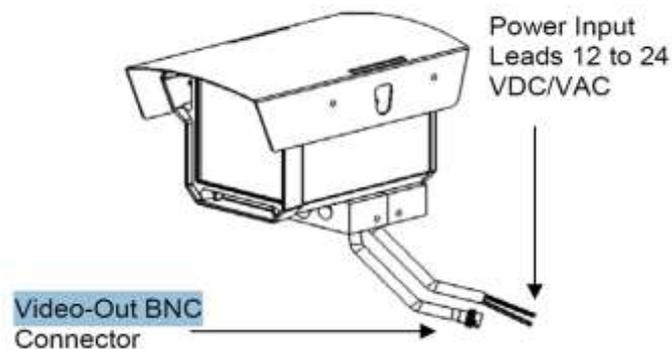


Figura 3-20: Detalles de cables de la cámara

3.5. Calibración

3.5.1. Tomas de medidas de prueba

Para la toma de datos diseñamos un sistema en base a las especificaciones de la cámara tanto vista frontal, lateral y superior para las diferentes distancias que debería enfocar la cámara y reconozcan con exactitud la placa de nuestro vehículo.

Las pruebas que se realizaron fueron en la ciudad de Guayaquil en las instalaciones de la Escuela Superior Politécnica del Litoral.



Figura 3-21: Especificar los puntos de prueba

Las especificaciones según el manual de la cámara tomaría la distancia del ancho del carril desde 3,7 m hasta 6,7 m dando un margen de error del 0,5 dependiendo de qué distancia este tomando esa prueba.

Usamos varias variables para identificar los parámetros a medir en las distancias respectivas de la cámara descritas anteriormente.

En la perspectiva horizontal tenemos las medidas (Figura 3-22):

- Distancia entre la cámara y el carril (D_{cc})
- Distancia entre la cámara y la región de interés. (D)
- Ancho del carril (W_c)

- El ángulo horizontal de la cámara (μ)

Adicionalmente se tienen otros datos:

- La distancia desde la mitad del carril a la cámara. (C)

Dato con el que trabajaremos para tener la tabla de pruebas en este módulo, y realizar el cálculo de los ángulos que tendrá la cámara.

- El margen de distancia (D) entre cada prueba. (Ra Delta)

Este dato nos sirve para saber con cuanta distancia se trabajará en cada prueba.

- La distancia entre la placa y donde está situada la cámara. (Rb)

Este dato es el resultado usando el teorema de Pitágoras, donde un lado es C y otro es D, anteriormente descrito.

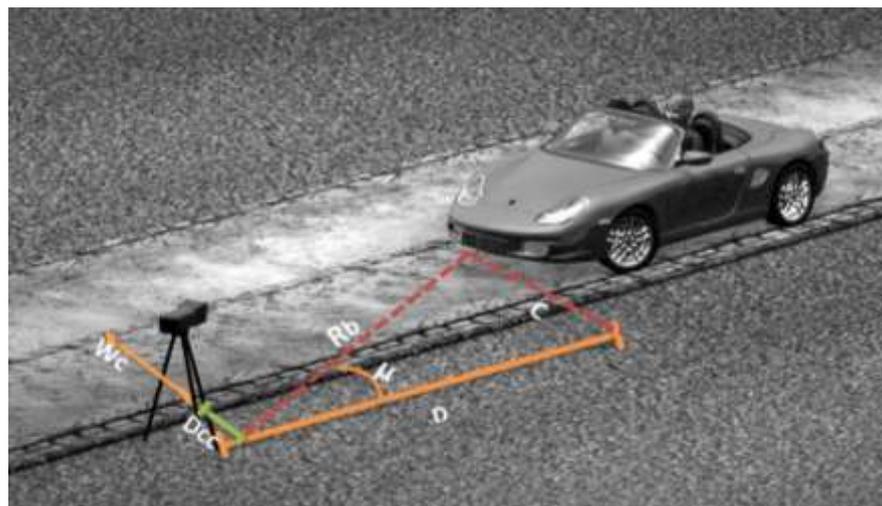


Figura 3-22: Perspectiva horizontal – Calibración

En la perspectiva vertical tenemos (Figura 3-23):

- La altura de la cámara. (H)
- La distancia entre la placa y el lente de la cámara. (Ra)
- El ángulo vertical de la cámara. (β)
- La distancia mínima entre la placa y el lente de la cámara. (RaMin)

Este dato es pedido en el sistema, refiriéndose al valor mínimo que alcanza la visión de la cámara, que es dado en el catálogo de la cámara, en nuestro caso es 3,7m.

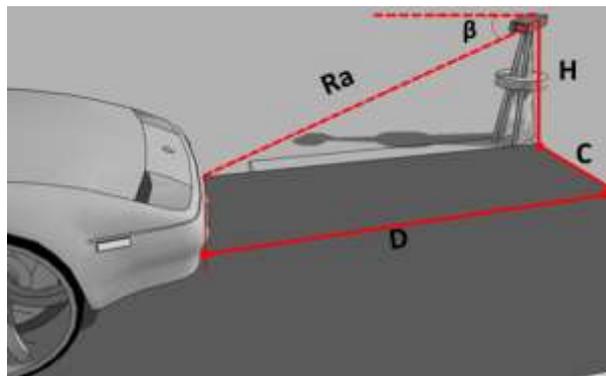


Figura 3-23: Perspectiva vertical – Calibración

Para las pruebas diseñamos 24 puntos de distancia con respecto a la cámara para analizar qué tanto es el enfoque y la vista de la placa del vehículo.

Estas pruebas se realizaron en 24 puntos diferentes del carril, como se detallan en la Figura 3-24.

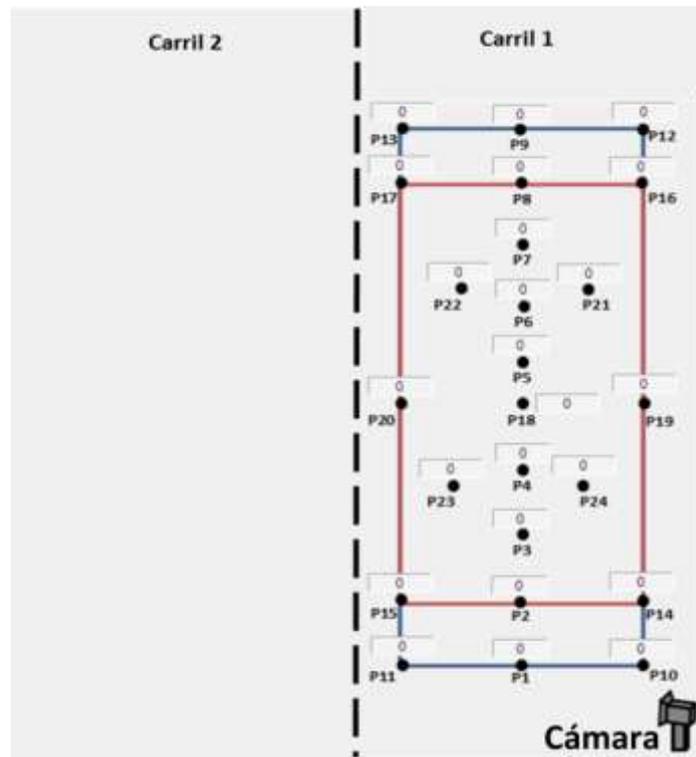


Figura 3-24: Detalles de los 24 puntos en el carril

A continuación se muestran los datos que arrojaron según las pruebas teóricas en base a las distancias y ángulos que se realizaron:

	P1	P2	P3	P4	P5	P6	P7	P8
Ra	3,20	3,70	4,20	4,70	5,20	5,70	6,20	6,70
H	1,50	1,50	1,50	1,50	1,50	1,50	1,50	1,50
Rb	2,83	3,38	3,92	4,45	4,98	5,50	6,02	6,53
C	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
D	2,00	2,73	3,37	3,98	4,56	5,12	5,67	6,22
μ	45,04	36,25	30,65	26,68	23,68	21,33	19,42	17,84
β	27,95	23,92	20,92	18,61	16,77	15,26	14,00	12,94

	P9	P10	P11	P12	P13	P14	P15	P16
Ra	7,20	4,51	2,51	7,87	6,92	4,87	3,12	7,42
H	1,50	1,50	1,50	1,50	1,50	1,50	1,50	1,50
Rb	7,04	4,25	2,02	7,72	6,75	4,64	2,74	7,26
C	2,00	3,75	0,25	3,75	0,25	3,75	0,25	3,75
D	6,75	2,00	2,00	6,75	6,75	2,73	2,73	6,22
μ	16,50	61,93	7,13	29,05	2,12	53,95	5,23	31,09
β	12,02	19,44	36,66	10,99	12,52	17,92	28,69	11,67

	P17	P18	P19	P20	P21	P22	P23	P24
Ra	6,40	5,13	6,03	4,73	6,25	5,67	4,06	4,85
H	1,50	1,50	1,50	1,50	1,50	1,50	1,50	1,50
Rb	6,23	4,91	5,84	4,49	6,07	5,47	3,77	4,61
C	0,25	2,00	3,75	0,25	2,88	1,13	1,13	2,88
D	6,22	4,48	4,48	4,48	5,35	5,35	3,60	3,60
μ	2,30	24,06	39,93	3,19	28,26	11,88	17,34	38,59
β	13,55	17,00	14,40	18,48	13,88	15,35	21,67	18,03

Tabla 3-1: Datos de resultados teóricos

Una vez generados los datos teóricos (Tabla 3-1), procedemos a ubicar el vehículo en cada punto para tomar las respectivas distancias y ángulos correspondientes, para lo cual mantenemos constante la altura de la cámara y la distancia de enfoque.

Ahora en cada punto tenemos definido teóricamente los ángulos de visión de la cámara tanto horizontal como verticalmente manipulamos la cámara con el inclinómetro. (Figura 3-25)



Figura 3-25: Calibrar la cámara con ángulo de prueba

Realizamos captura de 10 imágenes por cada punto manteniendo un margen de error de +0.1 hasta -0.1m.

Basados en estos resultados de los 24 puntos tomados, definimos un rango para la distancia de trabajo y el ángulo de visión. Estos dos permitirán definir los tamaños máximo y mínimo de la placa, ancho (W) y alto (H) Figura 3-26, y a su vez la perspectiva así mismo máximo y mínimos de los caracteres de la placa.

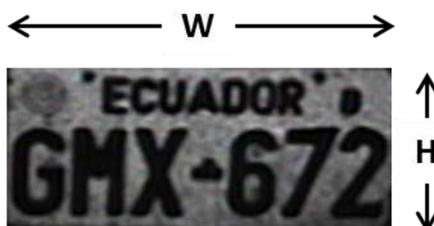


Figura 3-26: Detalle de placa

Finalmente obtenidos los resultados hacemos una comparación de los resultados teóricos y determinamos el porcentaje de error entre estos.

	P1	P2	P3	P4	P5	P6	P7	P8
Placa (px) (WxH)	155,1 x 102,9	142,2 x 87,3	132,1 x 71,1	120 x 61,8	105,8 x 59,5	98,2 x 53,5	91,2 x 48,7	85,3 x 43,1
Alto carácter (px)	50	45	38	33	30	28	26	23

	P9	P10	P11	P12	P13	P14	P15	P16
Placa (px) (WxH)	79,7 x 41	299 x 140	88 x 65	87 x 43	70 x 43	196 x 49	82 x 66	94 x 45
Alto carácter (px)	21	65	32	22	22	47	29	25

	P17	P18	P19	P20	P21	P22	P23	P24
Placa (px) (WxH)	72 x 43	106 x 63	127 x 64	85 x 46	106 x 53	84 x 48	100 x 58	144 x 67
Alto carácter (px)	21	30	31	25	27	25	30	36

Tabla 3-2: Ejemplo de resultados

3.6. Adquisición de la señal de video

En este módulo se describe la parte de la adquisición de la señal de video, y su configuración; y la parte de detección y extracción de la placa.

3.6.1. Librerías necesarias

Para la adquisición se utilizó la librería VideoInput, que es parte de OpenCV. Para la utilización de esta librería se debe instalar el Windows Driver Kit (WDK), el cual nos permite el manejo de cualquier tipo de dispositivo de video que queramos utilizar para la detección. El cual se lo puede descargar en la dirección <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=11800> [2011].

Para el funcionamiento de la librería de WDK, se debe agregar el `dllmsvcrt.dll` al `system32`, el cual lo puede conseguir en la dirección: http://www.nodevice.es/dll/Msvcrt_dll/item10840.html[2011].

La librería se lo especifica en visual, en Herramientas – Opciones – Proyectos y Soluciones – Directorios de VC++ - Archivos de biblioteca, como se detalla en la Figura 3-27.

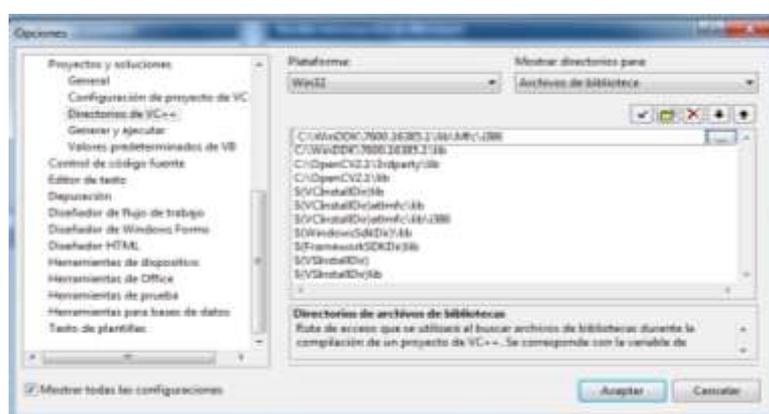


Figura 3-27: Inclusión de librería a visual

Para el funcionamiento de la librería `VideoInput`, se deben especificar los archivos `lib` en las propiedades del proyecto, como se especifica en la Figura 3-28.

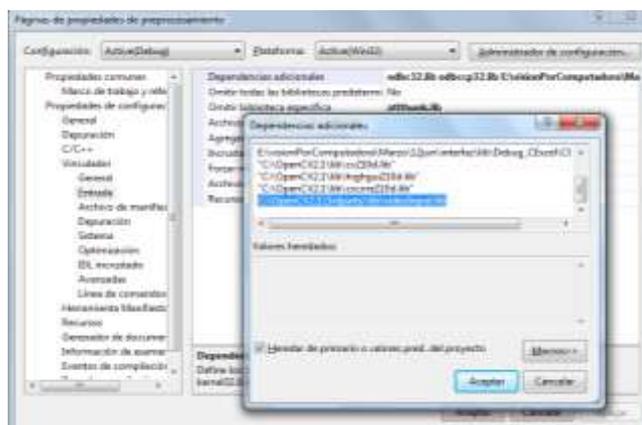


Figura 3-28: Incluir librería VideoInput al proyecto

Una vez incluida las librerías para la parte de la configuración de señal de video, se agregará la librería CvBlobsLib que ayudará en la detección de la imagen de la placa.

La librería cvBlobsLib, se encarga de tomar una imagen binarizada, y realizar un análisis de componentes conectados, convirtiéndolos en blobs, los cuales se guardan en memoria las partes blancas o negras de la imagen, según sea el caso, para que éstas pasen por un proceso de filtrado según sus características, e identificar el objeto de interés.

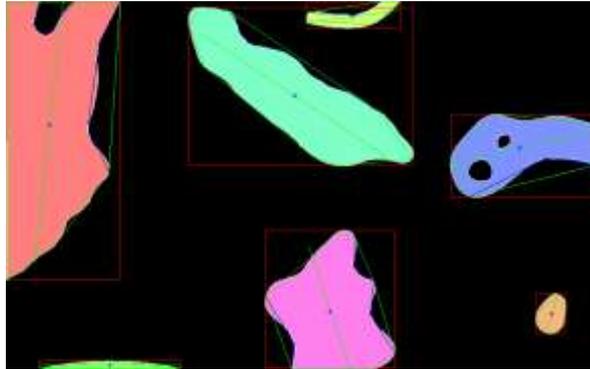


Figura 3-29: Ejemplo de CvBlobs

Es necesario incluir la librería de la solución de CvBlobs hacia nuestro proyecto, especificándolo en las Propiedades del proyecto – Propiedades de configuración – Vinculador – Entrada – Dependencias adicionales, como se especifica en la Figura 3-30.

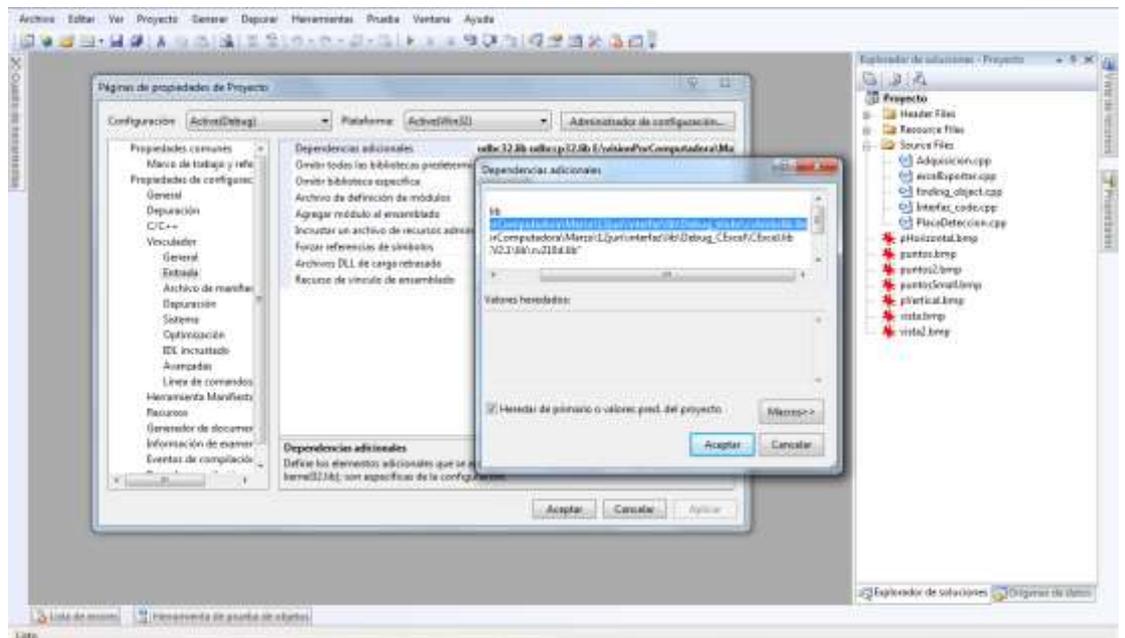


Figura 3-30: Inclusión de librería CvBlobs a visual studio

CAPÍTULO 4

4. DETECCIÓN Y EXTRACCIÓN DE PLACAS

4.1. Resumen

Aquí en este capítulo se detallará los procesos que se realizaron para crear el algoritmo de detección de la placa; se especificarán los 7 pasos para la detección y extracción de la placa del vehículo:

1. Pregunta si hay un dispositivo de video conectado
2. Obtiene *frames* para mostrar video.
3. Obtiene la imagen de la señal de video.
4. La convertimos en una imagen binaria
5. Obtenemos sus blobs con referencia a los espacios en blanco
6. Le pasamos un filtro con las características de una placa.
7. Guardamos la placa si la encontró

Estos pasos estarán detallados para entender cómo se llega a obtener la imagen desde una entrada "video", detectándola y extrayéndola hasta tener como salida la placa del vehículo pasando por varios procesos.

4.2. Introducción

Una vez obtenido el conjunto de imágenes en el capítulo de instalación y calibración, las analizamos obteniendo información para tener los parámetros constantes del algoritmo de detección.

Cuando tengamos los valores para los parámetros del algoritmo de detección; las introducimos al algoritmo para el proceso de filtrado, y obtener la placa de la imagen de entrada. (Figura 4-1)

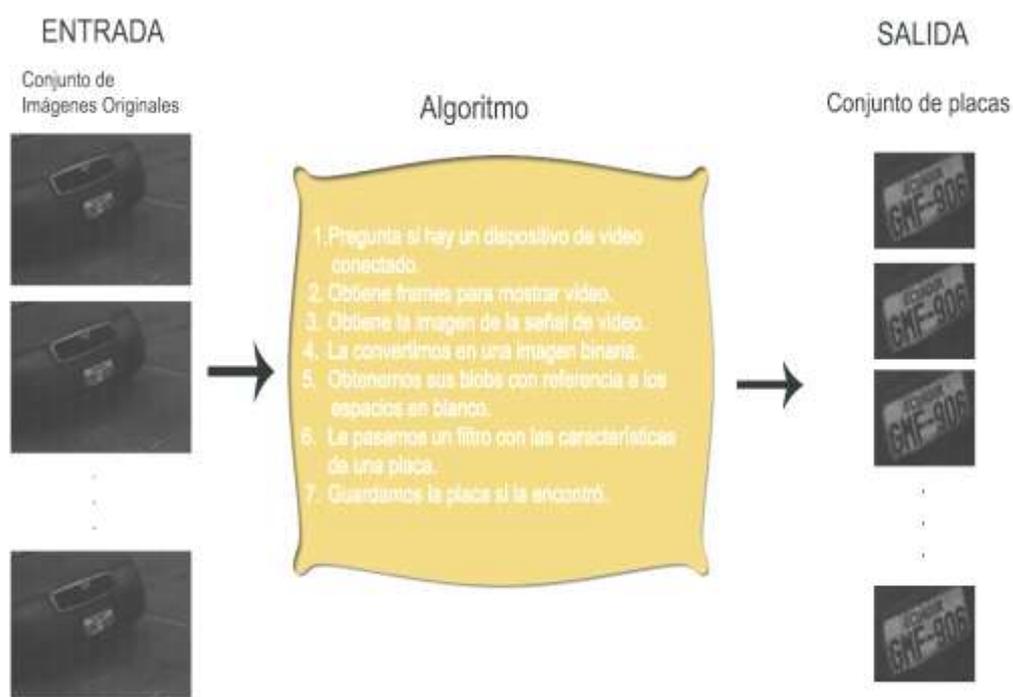


Figura 4-1: Algoritmo de detección y extracción de placas

4.3. Descripción del algoritmo

4.3.1. Extracción de parámetros constantes del algoritmo de detección

4.3.1.1. Parámetro de binarización

En este proceso, queremos obtener el umbral óptimo para que al binarizar la imagen de entrada, podamos enfocarnos en la placa y no en el ruido que pueda tener la imagen.

Para ello, se probó en las 240 imágenes, se la binarizó, y se obtuvo todos los blobs de cada imagen. Tomamos el blob que representa a la placa, y le calculamos la media en su estado original.

Una vez realizado este proceso a todas las imágenes de prueba, sacamos un promedio de estos valores, al cual llamaremos T (que proviene de la palabra *Threshold*, es umbral en inglés).

En nuestras pruebas obtuvimos que $T = 22.541$, éste es el valor que usaremos para el proceso de binarización de la imagen original.

4.3.1.2. Parámetro de área, media y firma

En este proceso, queremos obtener tres características que una placa pueda tener en una imagen de entrada: área, media y firma.

El área, es el valor del producto del alto y ancho del objeto.

Media, es el valor promedio del brillo del objeto.

Firma, es el número de veces de valores 1 que pueda tener una imagen binaria en la mitad (posición).

Para ello, se tomó las imágenes de pruebas, y se analizó una por una cual es el área de la placa de cada imagen. Una vez obtenido todos los resultados del área de las placas, se sacó un rango para introducirlo en el paso del filtro en el algoritmo.

Lo mismo se realizó para tener el rango de la media, para el filtro en el algoritmo. El procedimiento que se utilizó para obtener la media, fue obtener el ROI de la placa de las imágenes de prueba, y le calculamos la media a ese ROI.

Y para obtener el rango de firma, utilizando el ROI de la placa de cada imagen de prueba, se calculó los valores 1, que tiene el ROI en la fila del medio de la imagen, con ello se incluyó el resultado en el filtro del algoritmo.

A continuación revisaremos como quedó el algoritmo una vez obtenido sus parámetros.

4.3.2. Implementación del algoritmo

Este algoritmo se desarrolló en los siguientes pasos:

1. Pregunta si hay un dispositivo de video conectado

Nuestro sistema tendrá una entrada en la cual preguntará los diversos dispositivos que tendremos conectados en nuestro computador, una vez que haya reconocido todos los dispositivos

escogeremos el de nuestra cámara de video REG L1, la configuramos a cuantos *frames* por segundo queremos que nuestra cámara obtenga las imágenes, en nuestro caso la cámara de video tendrá 30 fps (*frames* por segundo).

2. Obtiene *frames* para mostrar video.

Una vez que tenemos configurados los *frames* en el video, estamos listos para capturar la placa del vehículo una vez que este ingrese a nuestra región de interés (ROI) inicializamos el ANPR y en cada *frame* verificará si la placa es detectada.

3. Obtiene la imagen de la señal de video.

Luego de que la placa de nuestro vehículo es detectada obtenemos la imagen de la señal de video y comenzamos a procesarla con los distintos parámetros de extracción de esa imagen.

4. La convertimos en una imagen binaria.

Uno de los parámetros de extracción es el proceso de binarización de una imagen en la cual nos ayudará a reducir la información para separar las regiones u objetos de interés en las cuales se utilizaran como máscaras de comparación o referencia.

En nuestro caso usaremos como un umbral el valor de T , que calculamos en el punto 4.3.1.1.

5. Obtenemos sus blobs con referencia a los espacios en blanco.

Después que realizamos el paso 4 de binarizar la imagen original de la señal de video obtenemos blobs de referencia por medio de la librería `cvblobresult`, como funciona esta librería, nosotros le mandamos un parámetro indicando si queremos tomar en cuenta las partes negras o blancas de la imagen binarizada; en nuestro caso tomaremos las partes blancas ya que la placa del vehículo en su mayoría es blanca.

La librería `cvbloblib` toma todas las partes blancas de la imagen binarizada entonces le pasamos un primer filtro en el área del blob.

6. Le pasamos un filtro con las características de una placa.

Para ello debemos saber:

Firma: El número de valores 0 en una imagen binaria.

Media: La media de la imagen original.

`mediaBlobOrig`: La media que tiene el blob sin binarizar.

`firmaBlobOrig`: La firma de la imagen binaria generada por el blob.

Original. (*blob en escala de grises*).

1. El ancho sea: `ancho>=80 && ancho<=210`
2. El alto sea: `alto>=30 && alto<=130`
3. La firma sea: `firma>9 && firma<145`

4. La media sea: $media > 104.400 \ \&\& \ media < 127.500$

5. La firmaBlobOrig sea: $firmaBlobOrig > 5$

6. Si $mediaBlobOrig \leq 14$

8. Si $firmaBlobOrig \leq 10$

9. Se encontró placa

Caso contrario: Si $firmaBlobOrig \geq 90 \ \&\& \ ancho > 200 \ \&\&$

$alto > 100 \ \&\& \ alto < 130 \ \&\& \ ancho < 180$

ó: $firmaBlobOrig < 90 \ \&\& \ firmaBlobOrig \geq 70 \ \&\& \ alto > 85 \ \&\&$

$alto < 100 \ \&\& \ ancho > 186 \ \&\& \ ancho < 205$

ó: $firmaBlobOrig < 70 \ \&\& \ firmaBlobOrig \geq 50 \ \&\& \ alto > 60 \ \&\&$

$alto < 80 \ \&\& \ ancho > 124 \ \&\& \ ancho < 145$

ó: $firmaBlobOrig < 50 \ \&\& \ firmaBlobOrig \geq 40 \ \&\& \ alto \geq 60$

$\ \&\& \ alto \leq 75 \ \&\& \ ancho > 120 \ \&\& \ ancho < 143$

ó: $firmaBlobOrig < 40 \ \&\& \ firmaBlobOrig \geq 35 \ \&\& \ alto \geq 40$

$\ \&\& \ alto \leq 52 \ \&\& \ ancho > 89 \ \&\& \ ancho < 102$

ó: $firmaBlobOrig < 35 \ \&\& \ firmaBlobOrig > 25 \ \&\& \ alto > 40 \ \&\&$

$alto < 100 \ \&\& \ ancho < 120 \ \&\& \ ancho > 85$

ó: $firmaBlobOrig < 25 \ \&\& \ alto > 40 \ \&\& \ alto < 70 \ \&\& \ ancho <$

$105 \ \&\& \ ancho > 80$

10. Se encontró placa.

7. Guardamos la placa si la encontramos



Figura 4-2: Muestra de imagen original y binaria



Figura 4-3: Muestra de la placa detectada

CAPÍTULO 5

5. SEGUIMIENTO DE LA PLACA

5.1. Resumen

En este capítulo, se detalla el uso de la librería SURF, para el seguimiento de la placa en la señal de video.

La librería sigue los siguientes pasos:

- 1) Detección de máximos y mínimos de la escala.
- 2) Localización de puntos de interés.
- 3) Asignación de la orientación.
- 4) Descriptor del punto de interés.

5.2. Introducción

Para un seguimiento de la placa, necesitamos un algoritmo que pueda identificar características de un objeto en una escena, que en este caso es nuestra imagen que tomamos de la señal de video. El objeto que vamos a utilizar para identificarlo en la escena, es la placa anteriormente detectada, que fue guardada en disco, para tomarlo y usarlo para identificar la ubicación de la placa en la imagen escena (Figura 5-1).

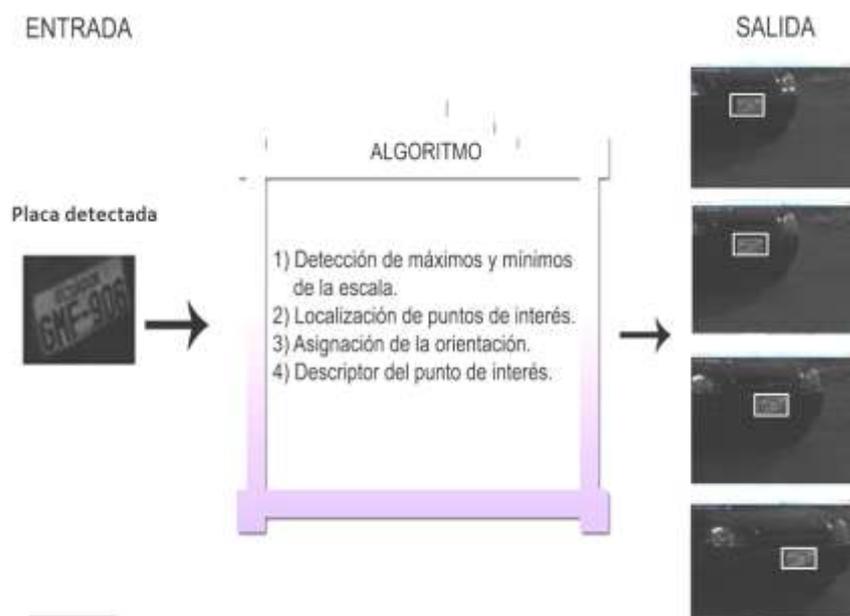


Figura 5-1: Algoritmo de seguimiento de la placa

5.3. Algoritmo de seguimiento (SURF)

SURF es un algoritmo utilizado en la literatura para la extracción de puntos de interés invariantes. La extracción de los puntos la realiza detectando en primer lugar los posibles puntos de interés y su localización dentro de la imagen. Posteriormente se representa la vecindad del punto de interés como un vector de características (por defecto con un tamaño de 64, aunque es posible aumentarlo).

El algoritmo SURF utiliza una aproximación básica de la matriz Hessiana para reducir el tiempo de computación.

La matriz Hessiana se utiliza debido a su buena relación entre la precisión y el coste temporal. El determinante de la Hessiana se utiliza para la localización de los puntos y para la determinación de la escala.

Dado un punto $x = (x, y)$ en una imagen I , la matriz Hessiana $H(x, \sigma)$ en x con escala σ se define como muestra la Figura 5-2:

$$\mathcal{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

Figura 5-2: Matriz Hessiana

donde $L_{xx}(x, \sigma)$ es la convolución de la derivada de segundo orden de la Gaussiana.

Para el cálculo del determinante, se realizan aproximaciones a las derivadas de segundo orden de la gaussiana de modo que se obtienen tres aproximaciones: D_{xx} , D_{xy} y D_{yy} . De esta forma, el determinante de la Hessiana que nos indica la escala del punto se calcula con la siguiente fórmula:

$$\det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2.$$

Figura 5-3: Determinante de la Hessiana

5.3.1. Detección de máximos y mínimos de la escala

El primer paso para obtener el descriptor una vez calculada la escala es el cálculo de la orientación del punto de interés para, posteriormente, calcular el descriptor SURF.

Para obtener un punto invariante a la orientación se calcula el “Haar-wavelet” para las direcciones x e y en una región circular de radio $6s$, donde s es la escala del punto de interés.

5.3.2. Localización de puntos de interés

Una vez calculados para todos los vecinos, se estima la orientación dominante calculando la suma de todos los resultados dentro de una ventana deslizante que cubre un ángulo de $\frac{\pi}{3}$. El cálculo del descriptor se realiza construyendo, primeramente, una región cuadrada centrada en el punto de interés y con un tamaño de $20s$.

5.3.3. Asignación de la orientación

La región se divide regularmente en 4 sub-regiones, y para cada sub-región se calculan unas pocas características simples. Seguidamente se calculan la “Haarwavelet” para x e y y se suavizan los resultados mediante una Gaussiana, obteniendo d_x y d_y . Para cada sub-región se suman los resultados dx y dy , además de calcularse su valor absoluto $|d_x|$ y $|d_y|$. De este modo, cada sub-región proporciona un vector v compuesto por: $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. El descriptor SURF se obtiene mediante la unión de los vectores de las sub-regiones.

5.3.4. Descriptor del punto de interés

La característica principal de los puntos de interés SURF es la repetibilidad, si el punto es considerado fiable el detector encontrará el mismo punto, bajo distintos puntos de vista (diferente escala,

orientación, etc.). El vector obtenido por el algoritmo deberá ser distintivo y al mismo tiempo robusto al ruido, errores, y deformaciones geométricas y fotométricas.



Figura 5-4: Ejemplo del reconocimiento mediante SURF

CAPÍTULO 6

6. DISEÑO DEL SISTEMA (GUI)

6.1. Resumen

Este capítulo es la integración del capítulo tres y cuatro. Este capítulo especifica como estará desarrollado nuestro GUI (Interfaz Gráfica de Usuario). El software tendrá varios módulos:

- Archivo.

1. Salir

- Calibración

1. Datos para pruebas
2. Datos resultantes

- Adquisición

1. Video
 - 1.1 Iniciar cámara
 - 1.2 Detener cámara

- Proceso

1. Configuración
 - 1.1 Parámetros de cámara
 - 1.2 Seleccionar ROI

- ANPR

1. Iniciar
2. Detener

- Ayuda

1. Ayuda de ANPR
2. Acerca de ANPR

6.2. Implementación del sistema

Desarrollamos nuestro sistema que tendrá varias propiedades para la extracción y detección de placas de vehículos. Los módulos del sistema están divididos en diferentes archivos .cpp en la solución en visual studio.

La Figura 6-1 muestra como está dividido el código del proyecto.

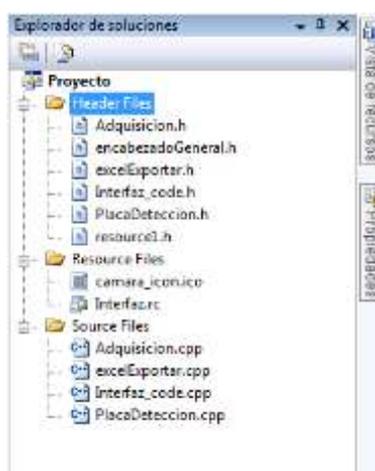


Figura 6-1: Códigos por módulo

Este software contará con los siguientes módulos:

- Archivo.

1. Salir

- Calibración

1. Datos para pruebas
2. Datos resultantes

- Adquisición

1. Video
 - 1.1 Iniciar cámara
 - 1.2 Detener cámara

- Proceso

1. Configuración
 - 1.1 Parámetros de cámara
 - 1.2 Seleccionar ROI

- ANPR

1. Iniciar
2. Detener

- Ayuda

1. Ayuda de ANPR
2. Acerca de ANPR

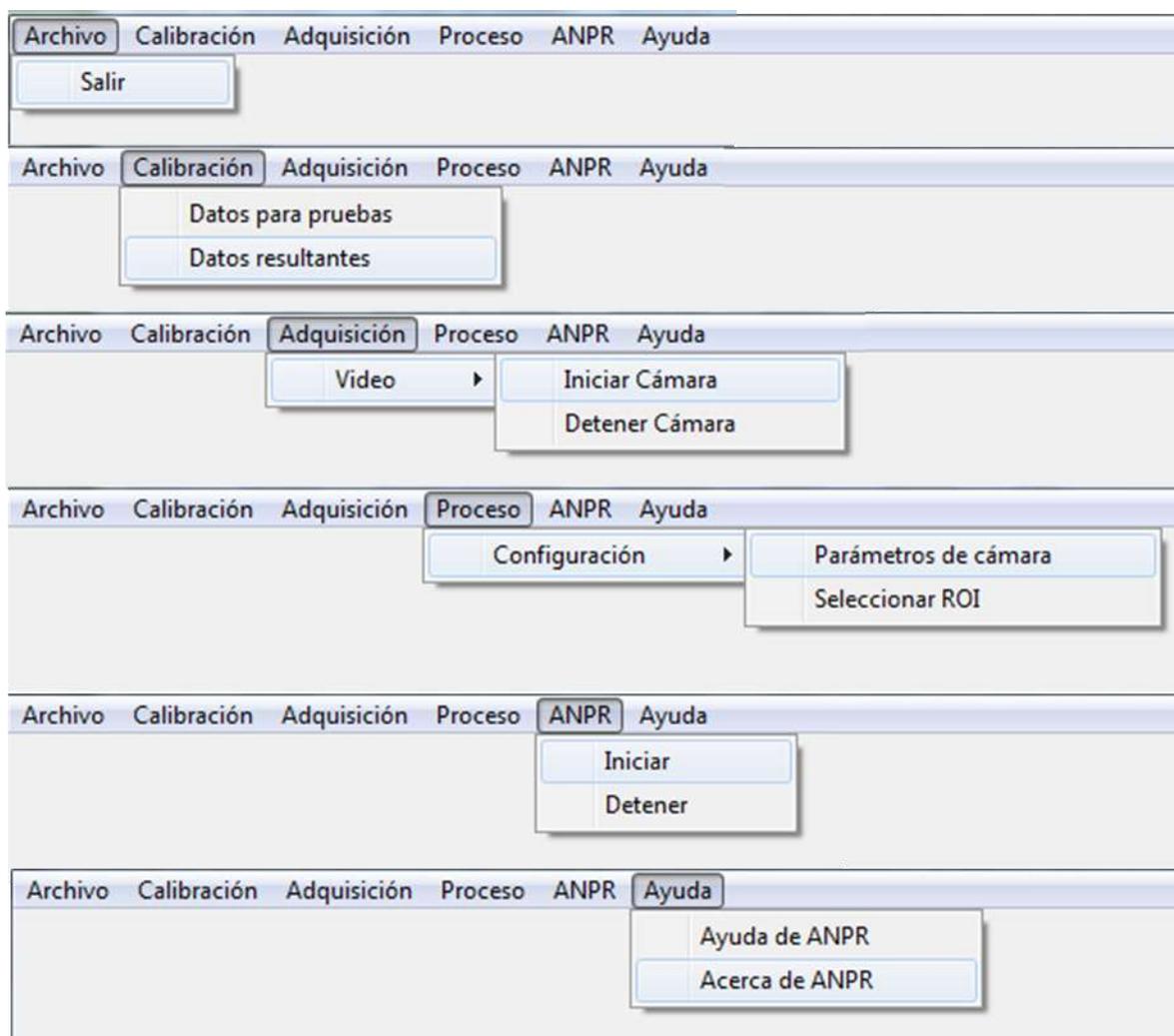


Figura 6-2: Menú del sistema

6.3. Módulo de calibración de la cámara

En nuestro sistema desarrollado en OpenCV el ANPR calculará todo estos datos de una manera sencilla y rápida dándonos a conocer los resultados teóricos que debemos realizar en una calibración de cámara, estos datos nos ayudaran en la vida real para tomar las

mediciones respectivas de distancias, ángulos, distancia de enfoque, etc.

6.3.1. Datos para pruebas

En esta opción se ingresarán los datos del lugar donde se quiera hacer la prueba de campo, y el sistema le devolverá los datos necesarios para calibrar la cámara y saber en qué región se hará la detección. Son 24 puntos de referencias para las pruebas que se deseen realizar.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
H	0.70	0.30	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20	0.70	0.20
W	0.24	0.03	0.26	0.03	0.41	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D	1.07	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
A	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figura 6-3: Interfaz módulo de calibración

Una vez que se han generado los datos tendremos la opción de exportar todos los datos a una hoja de excel, aparecerá una ventana en la cual se podrá guardar con el nombre que uno lo desee, como se muestra en la Figura 6-4. Se mostrará un mensaje que el archivo se ha generado con éxito, una vez teniendo el archivo en excel podremos

imprimirlo para hacer las mediciones físicas de distancias y ángulos respectivos.



Figura 6-4: Introducir nombre del documento

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
Ra	3.2	3.7	4.2	4.7	5.2	5.7	6.2	6.7	7.2	2.51	4.51	6.01	7.87	3.12	4.87	6.4	7.47	5.12	4.72	5.02	5.66	6.25	4.84	4.05
D	1	3.71	3.97	3.98	4.54	5.12	5.88	6.77	6.75	1	1	6.75	6.75	2.72	2.72	6.22	6.22	4.47	4.47	4.47	5.34	5.34	3.59	3.59
S	14.87	34.28	38.68	24.71	23.68	21.32	19.4	17.54	16.5	36.66	19.44	12.52	10.98	28.77	17.54	13.55	11.67	17.03	18.52	14.42	15.36	13.88	18.05	21.71
h	27.85	23.02	26.92	18.83	16.77	15.36	14	12.94	12.02	7.13	61.93	2.12	29.05	5.25	54.85	2.30	31.09	24.11	3.20	39.99	11.69	26.28	38.65	17.38

Figura 6-5: Archivo Excel generado para la calibración

6.3.2. Datos resultantes

Este módulo de calibración también tendrá una opción llamada datos resultantes aquí en esta ventana digitamos los resultados de los altos, anchos de la placa y las dimensiones de los caracteres de la placa, así mismo en cada punto observado nos daremos cuenta si fue detectada la placa y reconocido por el (*optical character recognition*,

OCR en inglés), entonces mediante un checkbox registramos en qué punto fue detectado y reconocido por el OCR la placa del vehículo.

RESULTADOS DE PRUEBAS												
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Ancho Placa	<input type="text"/>											
Alura Placa	<input type="text"/>											
Alura Caracter	<input type="text"/>											
Placa Detectada	<input type="checkbox"/>											
OCR Detectado	<input type="checkbox"/>											
	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
Ancho Placa	<input type="text"/>											
Alura Placa	<input type="text"/>											
Alura Caracter	<input type="text"/>											
Placa Detectada	<input type="checkbox"/>											
OCR Detectado	<input type="checkbox"/>											

Figura 6-6: Calibración – Datos Resultantes

Finalmente agregando los resultados de las pruebas según las dimensiones de la placa aparecerá otra ventana con 4 gráficos en la cual indicará los puntos detectados tanto de la placa como el OCR.

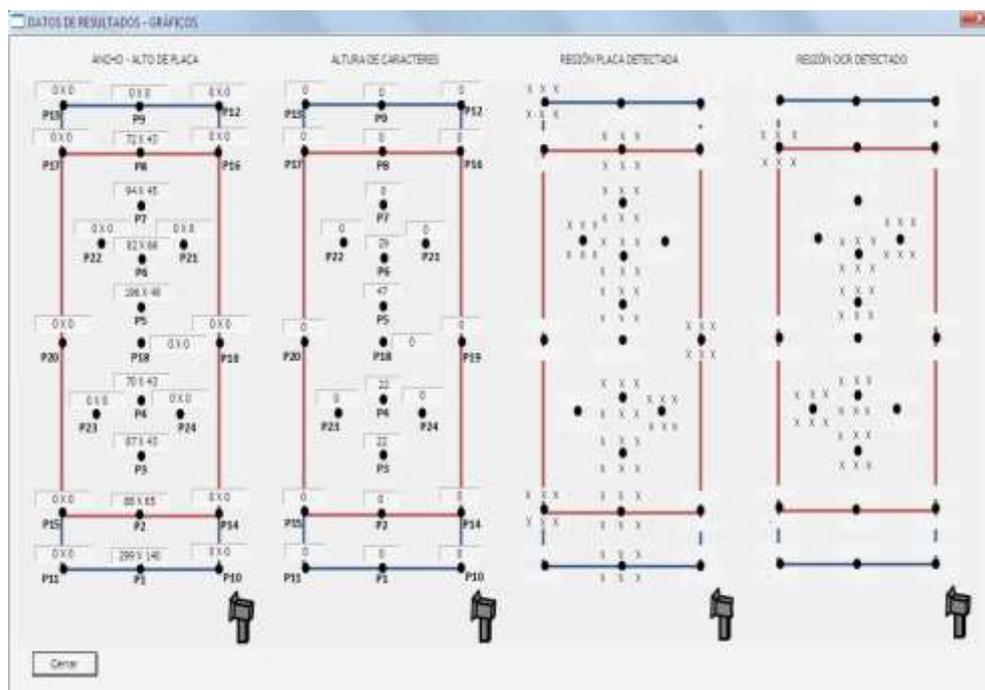


Figura 6-7: Datos Resultado imagen

6.4. Módulo de Adquisición y Configuración

El usuario deberá elegir el dispositivo de video que desea utilizar en la adquisición de video, como se muestra en la Figura 6-8.

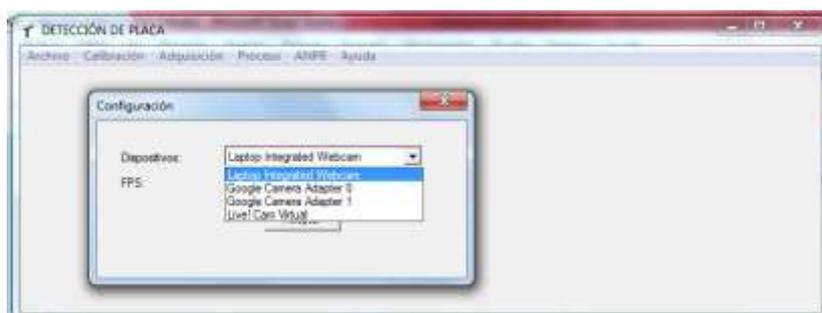


Figura 6-8: Adquisición – Elegir dispositivo

La lista que se muestra, son los diferentes drivers instalados en el ordenador, que pueden utilizarse para la adquisición de señal de video, y a continuación se obtendrá los frames para el video.

Para este paso se utilizó la librería VideoInput.

```

Adquisición.cpp  lista_cab.cpp  PicoCetecon.cpp
[Entorno global]
bool llenarCabcos()
{
    videoInput::listDevices();
    vi = new videoInput();
    int i, tam=0, device=0;
    bool resp = false;
    resp = vi->setupDevice(device);
    if(!resp)
    {
        Dispositivos = (char**)malloc(sizeof(char));
        indexDisp = (int*)malloc(sizeof(int));
        if(vi->devicesFound==0) return false;
        for(i=0; i<vi->devicesFound; i++)
        {
            Dispositivos[tam] = vi->getDeviceName(i);
            indexDisp[tam] = i;
            tam++;
        }
        tamDispositivos = tam;
        vi->stopDevice(device);
    }
    return resp;
}

```

Figura 6-9: Código para obtener lista de dispositivos

```

Adquisición.cpp  lista_cab.cpp  PicoCetecon.cpp
[Entorno global]
int main()
{
    setenv("path", ".");
    isOkSelected = false;
    img = img2;

    img = img1;
    img = imgMPP;

    if(vi->setupDevice(device))
    {
        cvNamedWindow(windowName, CV_WINDOW_NORMAL);
        cvMoveWindow(windowName, 10, 70);
        // para evento de seleccion ROI
        cvSetMouseCallback(windowName, on_mouse, 0);
        CVFont f = cvFont(1, 1);
        imgStarting = cvCvMatImage(cvSize(vi->getWidth(device), vi->getHeight(device)), 0, 3);
        while(1)
        {
            if(vi->isFrameAvailable())
            {
                cvGetImage(device, (unsigned char*)imgStarting+imgData, false, true);
                if(isOkSelected)
                {
                    dibujarROI(imgStarting, selection);
                }
                if(trackedObject == -1)
                {
                    dibujarROI(imgStarting, selection);
                    if(!isOkSelected)
                    {
                        seleccionarROI_MPP(imgStarting, cvPoint(selection.x, selection.height/2));
                    }
                }
            }
        }
    }
}

```

Figura 6-10: Código para obtener imagen de la señal de video

También usamos VideoInput para la configuración de la señal de video.



Figura 6-11: Configuración de señal de video

```

Adquisicion.cpp | main.cpp | PlacaDetector.cpp
-----
class global {
public:
    void putTextWithShadow(IplImage *img, const char *str,
        cvPoint point, int font, cvScalar color);
    void dibujarROI(IplImage *imagen, CvRect roi);
    void abrirVentanaConfiguracion();
    IplImage *obtenerImgFotografia();
    int obtenerTrackeadoObject();
};
  
```

Figura 6-12: Código para abrir ventana de configuración

Durante este proceso se puede hacer la selección de ROI, en el menú Proceso → Seleccionar ROI. Esta región indicará donde será el área en el cual se usará el algoritmo de detección de la placa.



Figura 6-13: Ejemplo de ROI seleccionado

6.5. ANPR

Aquí indicaremos cuando queremos empezar el proceso de ANPR.

Utilizando la librería cvBlobsLib, y un proceso de filtrado de blobs.

En el módulo de Interfaz_code, indicamos el inicio del ANPR, cuando haya escogido en Menú → ANPR → Iniciar ANPR, según sea el caso.

```

//Arbitrio gitea)
// WinProc:HANDLE hInst, LPDI message, WPARAM wParam, LPARAM lParam)
//
//case MM ANPR START:
if(!is_running){
    mensajeAlerta="Debe Iniciar cámara \n (Adquisición - Video - Iniciar cámara)";
    DialogBox(hInst, (LPCSTR)ID_MSGBOX, hMod, (DLGPROC)frmMsg);
    setearInicioANPR(false);
}
else{
    if(!GetLastErrorObject() != -1){//no ha dibujado ROI
        mensajeAlerta2="No se ha seleccionado ROI \n ¿Deese utilizar toda la im...";
        DialogBox(hInst, (LPCSTR)ID_MSGBOX_2, hMod, (DLGPROC)frmMsg2);
        switch(btnRespuesta){
            case -1://no
                setearInicioANPR(false);
                break;
            case 0://cancelar
                setearInicioANPR(false);
                break;
            case 1//si
                setearInicioANPR(true);
                break;
        }
    }
    else{
        if(!GetLastErrorObject() != -1){//dibujó ROI pero no se ha seleccionado
            mensajeAlerta2 = "¿Deese utilizar ROI actual para ANPR?";
            DialogBox(hInst, (LPCSTR)ID_MSGBOX_2, hMod, (DLGPROC)frmMsg2);
        }
    }
}

```

Figura 6-14: Código iniciar ANPR

Para que en el módulo de adquisición, pregunte si se ha iniciado ANPR, para que haga la detección, como vemos en la Figura 6-15.

```

#include <opencv2/opencv.hpp>
using namespace cv;

int main() {
    // ...
    Mat img;
    // ...
    if (!hayPlaca) {
        // ...
        if (!hayPlaca) {
            // ...
            if (!hayPlaca) {
                // ...
                imgPlaca = deteccion(img);
                // ...
            }
        }
    }
}

```

Figura 6-15: Código llamada de la función detección

Al preguntar si ha iniciado ANPR, realiza el proceso de detección, y luego hace el seguimiento (*tracking*) de la placa en el video. Y si al realizar el seguimiento de la placa, no lo encuentra, entonces se iguala “*hayPlaca*” a false, para que al volver a realizar la detección y ya no haga seguimiento.

En el código de detección, calculamos la media de la imagen, y le sacamos los blobs “blancos” de la imagen. Luego, estos blobs pasarán por el proceso de filtrado, con las características de la placa.

Estas características fueron calculadas, mediante el proceso de pruebas de las 240 imágenes, más los 10 videos de prueba.

6.6. Ayuda

En esta opción se muestra un detalle de lo que contiene el sistema, y sus descripciones.



Figura 6-16: Opción de Ayuda

CAPÍTULO 7

7. RESULTADOS EXPERIMENTALES

7.1. Resumen

En este capítulo tiene como objetivo presentar los valores numéricos de los distintos parámetros medidos durante las toma de pruebas. El criterio de presentación de la información obtenida sigue el criterio de la presentación gráfica de resultados en cada uno de los 24 puntos obtenidos a partir de las mediciones que se realizaron en un carril de una carretera, es decir, imagen, ancho, alto, media firma, etc.

Se obtuvieron 240 imágenes a partir de cada punto, o sea, en un punto 10 imágenes esto nos ayudaría con una distancia de +0.1 y -0.1m y poder controlar los márgenes de errores.

También se obtuvieron 10 videos para realizar pruebas con el auto en movimiento.

7.2. Metodología de las pruebas

Con el objetivo de conocer la región en donde la placa es detectada en un carril, se dibujó en la carretera los 24 puntos disponibles en el módulo de calibración. Como nos muestra la Figura 3-20.

Para cada punto se tomó 10 imágenes con un margen de distancia de $\pm 0.1 m$; por ejemplo, en el punto 3, la distancia D es de $3.37 m$, en ese punto se tomó desde 3.47 a $3.27 m$. Como muestra la Figura 7-1.

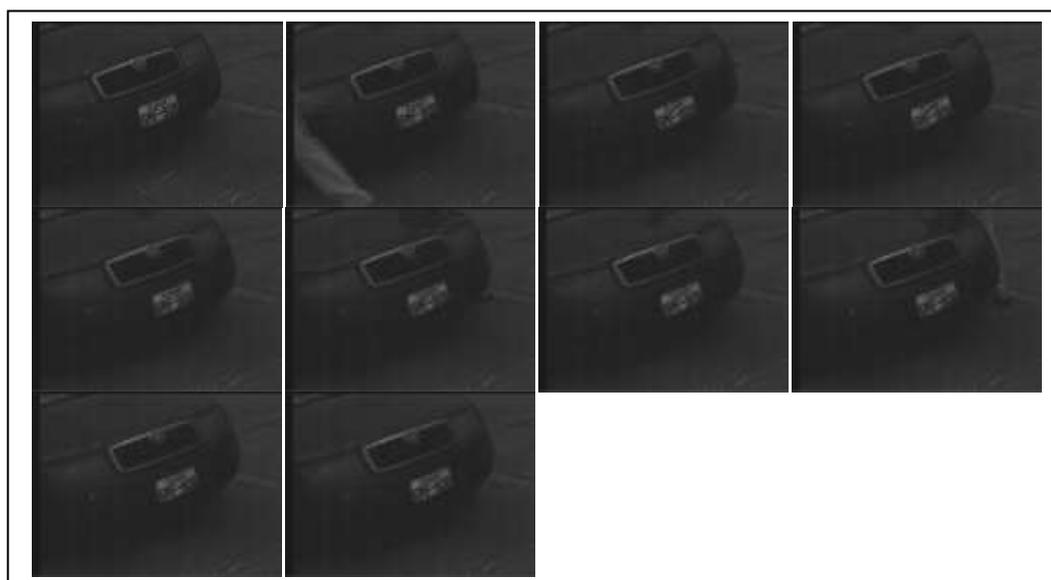


Figura 7-1: Ejemplo de imágenes de pruebas por punto

Estas imágenes fueron tomadas en la carretera de la Escuela Superior Politécnica del Litoral. Se utilizó el algoritmo de detección de placas para estas imágenes. Sus resultados fueron analizados en una tabla como muestra la Tabla 7-1.

Prueba	Imagen	ancho	alto	firma	media	media inicial	multiplico	n	i	esPlaca	media blob orig	firma blob orig
1	1	131	149	112	93.387	19.005	2	3	1	no	17.224	105
1	1	102	103	15	123.855	19.005	2	3	2	no	9.692	0
1	1	130	134	53	118.977	19.005	2	3	3	si	17.143	21
1	2	148	158	128	90.172	19.921	2	3	1	no	17.130	123
1	2	117	112	13	116.276	19.921	2	3	2	no	11.129	0
1	2	130	135	52	119.540	19.921	2	3	3	si	17.339	24
1	3	149	160	130	87.363	20.788	2	3	1	no	17.638	121

Tabla 7-1: Análisis de características de los blobs

También se realizó pruebas con el auto en movimiento, para esto se hizo 10 videos.

En el video 1 y 2, se tuvo al auto en movimiento sin ninguna luz encendida. Como se muestra en la Figura 7-2.



Figura 7-2. Video de prueba 1

En el video 3 y 4, se tuvo al auto en movimiento con las luces intermitentes encendidas. Como se muestra en la Figura 7-3.



Figura 7-3: Video de prueba 3

En el video 5, 6 y 7, se tuvo el auto en movimiento con las luces intensas encendidas. Como se muestra en la Figura 7-4.



Figura 7-4: Video de prueba 5

En el video 8, 9 y 10, se tuvo el auto en movimiento con las luces intermitentes e intensas encendidas.



Figura 7-5: Video de prueba 9

7.3. Resultados

Una vez realizado el análisis de las 240 imágenes con el algoritmo, se obtuvo una región de detección, como lo muestra la Figura 7-6.

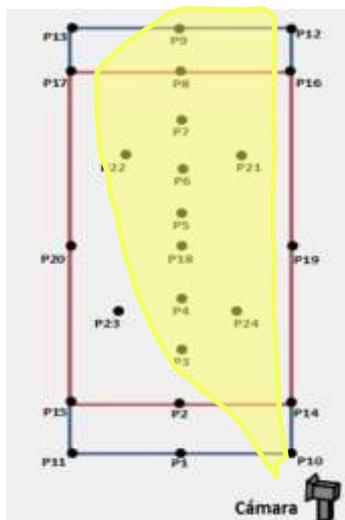


Figura 7-6: Región donde se detecta la placa

Con los videos se pudo realizar un análisis de la consistencia de la detección de la placa. Por lo que se adicionó al algoritmo, el seguimiento de la placa durante el proceso de ANPR.



Figura 7-7: Seguimiento de la placa en el proceso ANPR

CONCLUSIONES

1. Se puede realizar un análisis estadístico, bajo los parámetros de entrada que nos da el sistema, ya sea para saber las horas picos de un estacionamiento, la demanda de autos por un espacio en el estacionamiento, y sobre todo un control de acceso de los vehículos; los cuales son problemas comunes en el control de acceso de un estacionamiento.
2. Se puede detectar o predecir problemas para el control de acceso vehicular; esto quiere decir, que podemos prevenir desconciertos que pueden suceder en la entrada de un parqueadero, ya sea para un registro rápido de las placas de ingreso, como el registro de salida, o un problema por horas picos.
3. En las pruebas realizadas en el proceso de instalación y calibración se puede encontrar con diferentes puntos en el cual la cámara pueda trabajar, lo que nos indica cual es el punto óptimo, es realizando un análisis de los resultados de la detección de la placa vehicular junto con los resultados del reconocimiento de caracteres con OCR.

4. Se puede incluir al sistema, una plantilla de los 24 puntos, que se refleje en la señal de video, de tal forma, tener una idea en qué posición estén cada uno de los 24 puntos en el carril del vehículo, para que la calibración sea precisa, y se pueda conocer el punto óptimo en donde se calibre la cámara, de una manera más rápida.

RECOMENDACIONES

1. Se recomienda mantener los equipos, para el uso del sistema, que se encuentre en un lugar de temperatura menor de 22° para evitar daños futuros, y protegido para cualquier clima.
2. Al implementar tener en cuenta las características del computador, ya que los procesos de la detección consumen recurso tanto del CPU como de la memoria. Para mejor desempeño se recomienda una memoria mayor a 2GB, y más de 2 procesadores.
3. Se puede incluir al sistema, una plantilla de los 24 puntos, que se refleje en la señal de video, de tal forma, tener una idea en qué posición estén cada uno de los 24 puntos en el carril del vehículo, para que la calibración sea precisa, y se pueda conocer el punto óptimo en donde se calibre la cámara, de una manera más rápida.

BIBLIOGRAFÍA

[1] Wikipedia. “*Automatic number plate recognition*”.

[Online] http://en.wikipedia.org/wiki/Automatic_number_plate_recognition. [2011]

[2] State Of The Art Solution. “*Cómo funciona Carmen ANPR*”.

[Online] <http://es.arhungary.com/>. [2011]

[3] Adaptive Recognition Hungary. “*Container Code Recognition*”

[Online] http://www.ocrtech.com/container_code_sw.html [2011]

[4] Wikipedia. “*Police-enforced ANPR in the UK*”

[Online] http://en.wikipedia.org/wiki/Police-enforced_ANPR_in_the_UK. [2011]

[5] Visión Computacional. “*Reducción de ruido en una imagen (Filtros)*”.

[Online] <http://atmsp.wordpress.com/category/vision-computacional/>[2011]

[6] BOSCH Security. “*Cámaras REG para captura de matrículas*”.

[Online] http://www.boschsecurity.com.ar/acerca/noticias_y_eventos_microsites/microsite_reg.asp [2011]

[7] VideoInput. “*A free windows video capture library*”.

[Online] <http://muonics.net/school/spring05/videoInput/> [2011]

[8] OpenCV-cocoa. “*OpenCV 2.0 library with updated Cocoa and Quicktime support*” HIGHGUI “*Reading AVIs & Camera data*” [2011]

[Online] <http://opencv-cocoa.googlecode.com/svn-lhistory/r7/trunk/src/highgui/cvcap.cpp>[2011]

[9]Koders.com. Black Duck Software Company “*highgui.h*”

[Online] <http://www.koders.com/cpp/fid0AFD20D23F2CCA0DDB83CC9B4F9142A1D9767C94.aspx?s=opencl#L103> [2011]

[10] Lapapa.com. "*Placa capturadora de video 4 cámaras PCI Chipset: Conexant Fusion 878A*"

[Online] http://chile.lapapa.cl/cva/30228044_Capturadora-De-Video-4-Camaras-Pci-Conexant-Fusion-878a.html [2011]

[11] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[12] CvBlob. "Blob library for OpenCV"

[Online] <http://code.google.com/p/cvblob/> [2011]