

# DETECCIÓN Y EXTRACCIÓN DE PLACAS DE VEHÍCULOS EN SEÑALES DE VIDEO

Helen Rodríguez<sup>(1)</sup>, Robert Vera<sup>(2)</sup>, Boris Vintimilla<sup>(3)</sup>

Facultad de Ingeniería en Electricidad y Computación  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 Vía Perimetral  
Apartado 09-01-5863. Guayaquil, Ecuador

hhelparod@espol.edu.ec<sup>(1)</sup>, robertitojr@gmail.com<sup>(2)</sup>, bvintim@fiec.espol.edu.ec<sup>(3)</sup>

## Resumen

*En este proyecto se implementó un sistema para detectar y extraer placas de vehículos desde una señal de video utilizando un sistema de visión artificial. El proyecto se divide en dos etapas: en la primera etapa se realiza la instalación y calibración de un sistema de video vigilancia que permite capturar una señal de video para su posterior análisis y procesamiento. Luego, esta etapa es complementada con el desarrollo de una aplicación de detección y extracción de placas vehiculares a partir de la señal de video capturada. En la etapa de instalación y calibración se definen los recursos que necesita el sistema, la instalación de software y hardware, así como también la calibración de la cámara mediante cálculos utilizando los datos del catálogo de la cámara y cómo influye el medio en donde va a trabajar el sistema. Asimismo se indica cómo realizar la calibración utilizando una plantilla predefinida. En la etapa de extracción de la placa del vehículo, se realiza un algoritmo de detección de placas, utilizando las librerías de OpenCV en lenguaje C con la herramienta VisualStudio 2008. Como resultado final del proyecto se obtiene un sistema que captura una señal de video, la analiza y procesa y finalmente encuentra una placa en la región que haya sido calibrada, mostrando además un seguimiento de la placa en la señal de video.*

**Palabras Claves:** ANPR, señal de video, reconocimiento, detección, placa vehicular.

## Abstract

*In this project had implemented a system to detect and extract vehicle plate number from a video signal, using an artificial vision system. The project is divided by two phases: in the first one we make the installation and calibration of a surveillance video system that allows capturing a signal video for its subsequent analyzing and processing. Then, this phase is supplemented developing of an application of detection and extraction of the vehicles plates from signal video captured. In the Installation and calibration, is defined the resources that need the system, the installation of software and hardware, likewise the camera calibration, by calculation using data from the camera's catalog and how ambience affects in the place where it's going to work. It also shows how to perform the calibration using a predefined template. In the extraction of the vehicle plate phase, is made a plate detection algorithm using the OpenCV library in C language with VisualStudio 2008. As a final result, it got a system which capture a video signal, analyze and process it, and finally find a plate in the region that has been calibrated, showing up the plate track on the video signal.*

**Palabras Claves:** ANPR, signal video, recognition, detection, vehicle plate.

## 1. Introducción

Actualmente existen diferentes tipos de problemas con respecto al control de acceso vehicular en parqueaderos o áreas de acceso restringido. Esto es típico en parqueaderos tales como: centros comerciales, aeropuertos o parqueaderos públicos o privados. Sumado a estos problemas no se lleva un control estadístico de los diferentes parámetros que se generan al ingresar o salir un vehículo del parqueadero, esto es, vehículos ingresados, disponibilidad de parqueos, demandas de parqueos en el tiempo, o para prevención de futuros problemas.

Este proyecto se enfoca en el desarrollo de un sistema de detección y extracción de placas vehiculares en la zona de acceso a un parqueadero basado en visión artificial. El sistema inicialmente realiza un proceso de instalación y calibración de una cámara para adquirir eficientemente una señal de video; luego a partir de esta señal realiza la detección y extracción de placas. En estos procesos se obtendrán los parámetros necesarios tanto para la instalación del software y hardware, así como para la calibración a modo de obtener una óptima adquisición de imágenes mediante una señal de video. La detección y extracción de la placa vehicular se realizará a través de un algoritmo basado en el procesamiento de la imagen adquirida.

## 2. Sistemas ANPR

El sistema de reconocimiento automático de matrículas también llamado sistema ANPR (*Automatic number plate recognition* en inglés) es una tecnología que permite la vigilancia en masa a través del uso de un software de reconocimiento óptico de caracteres para escanear imágenes y leer las matrículas de los vehículos.

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, que pueden registrar imágenes de matrículas de vehículos que circulan a velocidades considerables.

Estos sistemas se basan principalmente en el uso de componentes de hardware y software. :

### *Componentes de Software*

- Herramientas de desarrollo
- Librerías de desarrollo:
  - o Captura de video
  - o Procesamiento de imágenes
  - o OCR

### *Componentes de Hardware*

- Sistema de iluminación
- Cámara
- Digitalizador de imagen
- Computador

### 3. Instalación y calibración del sistema de adquisición de imágenes

En esta etapa 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. El esquema general de este proceso se muestra en la Figura 1.



**Figura 1.** Esquema general del proceso de instalación y calibración del sistema ANPR.

#### 3.1 Componentes de software

En esta sección se describen las herramientas de software que se usaron para el desarrollo de este proyecto. Debemos conocer que el sistema operativo usado fue Windows 7.

Los componentes usados fueron:

**Herramientas de desarrollo:**

- *VisualStudio* 2008: Entorno de desarrollo integrado para la programación del sistema.

**Librerías de desarrollo:**

- *Librería OpenCv 2.1* para el procesamiento de imágenes. Ésta librería debe ser incluida al proyecto de visualStudio para su correcto funcionamiento. Previo a esto debe ser incluida a las variables de entorno del sistema operativo.
- *Librería CvBlobs*, usado 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.
- *Librería VideoInput*, usado para la adquisición de la señal de video, mediante esta se podrá obtener las imágenes que produce, para después analizarlas.

#### 3.2 Componentes de hardware

En esta sección se describen los componentes de hardware que se usaron para el desarrollo de este proyecto.

Los componentes utilizados fueron los siguientes:

- Cámara
- Sistema de iluminación
- Digitalizador de imagen
- Computador

##### 3.2.1. Cámara REG- L1

Para la adquisición de las imágenes se usó una cámara Bosch modelo REG-L1-816XC-01, Figura 2, la cual trabaja con un rango de visión de 5 a 8 m. Esta cámara realza el brillo de las placas, permitiendo una mejor detección de la misma. Para una óptima

detección de la placa, la cámara deberá tener un ángulo máximo de inclinación de 40° tanto horizontal como vertical.

Una de las características principales de la cámara, es que cuenta con un sistema de iluminación infrarrojo incorporado, que le permite visualizar tanto de día como en la noche. De esta manera esta cámara proporciona imágenes de gran contraste de las placas en diferentes tipos de condiciones de luz ambiental, esto es, desde la oscuridad de la noche hasta la luz directa del sol o los faros de los vehículos. Para mayores detalles de las características de esta cámara en [6].



**Figura 2.** Cámara Bosch-REG-L1-816XC-01

Adicionalmente utilizamos una fuente de voltaje de 12VAC – 24VDC para conectar la cámara a la fuente de poder, Figura 3.



**Figura 3.** Fuente de la cámara

### 3.2.2. Digitalizador de imagen

El digitalizador de imágenes, o también llamado *frame grabber* en inglés, es una tarjeta de hardware cuyo propósito es recibir la señal analógica de vídeo enviada por la cámara y transformar la señal en una

imagen digital tal que el computador la pueda entender.

El Conexant Fusion 878A, como lo muestra la Figura 4, fue el digitalizador usado para este proyecto, el cual cuenta con las siguientes características:

- Interface: PCI
- Entradas: 4 puertos de video conectores BNC + 1 audio.
- Conversión: NTSC/PAL
- Tecnologías: CMOS Señal Mixta 3.3
- Calidad de Registro:
  - NTSC 320x240, 640x480
  - PAL 384x288, 768x576
- Frecuencia de imágenes:
  - NTSC 30fps, PAL 25fps
- Expansión: 4 tarjetas / 16 cámaras



**Figura 4.** Digitalizador de imágenes

### 3.2.3. Computador

El computador nos permite recibir la señal digital de video, procesar la señal y almacenar los datos extraídos de las imágenes que obtenemos mediante la cámara.

### 3.3 Calibración del sistema de adquisición de imágenes

Dada la información del catálogo de la cámara especificado en la sección 3.3.1., en donde la cámara tomará 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 5:

- Distancia entre la cámara y el carril (Dcc)
- Distancia entre la cámara y la región de interés. (D)
- Ancho del carril (Wc)
- El ángulo horizontal de la cámara ( $\mu$ )

Adicionalmente se tienen otros datos:

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

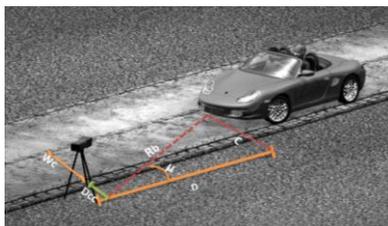
Dato con el que trabajaremos para tener la tabla de pruebas en este módulo, y realizar el cálculo de los ángulos que tendrá la cámara.

- 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.

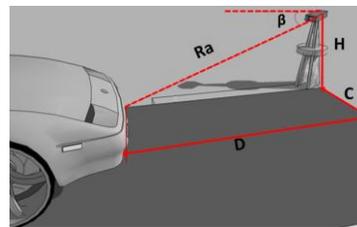


**Figura 5:** Perspectiva horizontal

En la perspectiva vertical tenemos Figura 6:

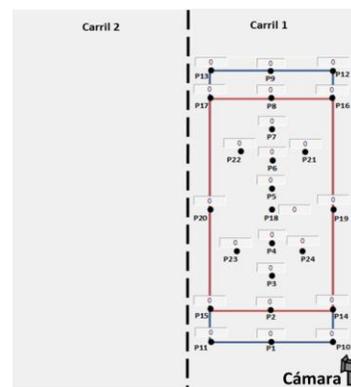
- La altura de la cámara. (H)
- La distancia entre la placa y el lente de la cámara. (Ra)
- El ángulo vertical de la cámara. ( $\beta$ )
- La distancia mínima entre la placa y el lente de la cámara. (RaMin)

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 6:** Perspectiva vertical.

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, Figura 7.



**Figura 7.** 24 distancias de enfoque-carril 1

Una vez instalado todos los requerimientos del sistema,

procedemos a realizar el proceso de adquisición de imágenes.

### 3.4 Adquisición de la señal de video

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. Para el funcionamiento de la librería de WDK, se debe agregar el dllsvcrtd.dll al system32.

## 4. Detección y extracción de placas

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, como muestra la Figura 8.



**Figura 8.** Detección y extracción de placas

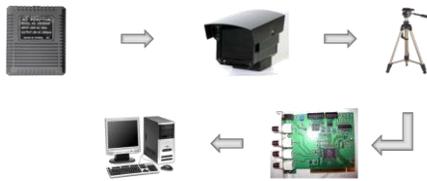
Aquí en este capítulo se detallará los procesos que se realizaron para crear el algoritmo de detección de la placa; se especificarán los 7 pasos para la detección y extracción de la placa del vehículo:

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.

### 4.1 Obtener imagen de la señal de video

Para llegar al resultado final de la imagen capturada por la señal de video, pasa por un proceso de conversión, ya que la cámara que usamos es analógica. Por lo que realizamos una secuencia como muestra la Figura 9.



**Figura 9.** Adquisición de imagen

## 4.2. Descripción del algoritmo

Para realizar la detección de la placa, utilizamos la librería cvBlobs, el cual obtiene varios blobs dada una imagen binaria. Por ejemplo, en nuestro escenario podemos observar cuales son los valores blancos (0) y negros (1) encerrándolos en un rectángulo, lo que llamamos segmentación de regiones, presentándolo como muestra la Figura 10.



**Figura 10.** Segmentación de regiones

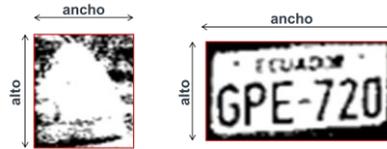
El algoritmo de detección contiene varios procesos de filtrado que a continuación se describirán:

- Alto – Ancho
- Firma de blob
- Media de blob en binario
- Media de blob en escala de grises

### 4.2.1. Filtro: Alto-Ancho

En este filtro, se basa en no dejar pasar blobs que tengan las

características del área de una placa, estas características se las obtuvieron en el proceso de pruebas.



**Figura 11.** Alto – Ancho de la placa

### 4.2.2. Filtro: Firma de blob

En este filtro, comparará los valores altos (1) de la imagen binaria, que se encuentra en la mitad de la imagen, con los valores altos que una placa en binario pueda tener.



**Figura 12.** Firma de la placa

### 4.2.3. Filtro: Media de Blob en binario

El valor de la media de la imagen del blob en binario, se comparará con el valor promedio que tiene una imagen de la placa en binario, el cual se obtiene en las pruebas.



**Figura 13.** Media de placa en binario

### 4.2.4. Filtro: Media de Blob en escala de grises

Al igual que el filtro anterior, se mide la media del blob, en escalas de grises, comparándolos con los valores resultantes que se obtuvieron en las pruebas.



**Figura 14.** Media de placa en escala de grises

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.

### **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, 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 es blanca en binario.

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).

## 7. Se encontró placa



Figura 15. Muestra de la placa detectada

## 5. Seguimiento de la placa

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

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 a identificar en la escena, es la placa anteriormente detectada, que fue guardada en disco, para tomarlo e identificar la ubicación de la placa en la imagen escena.

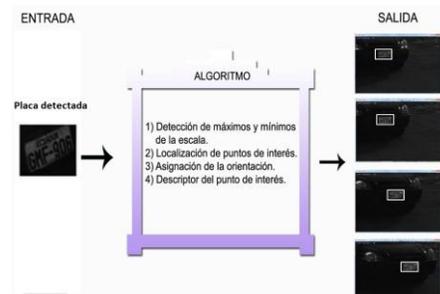


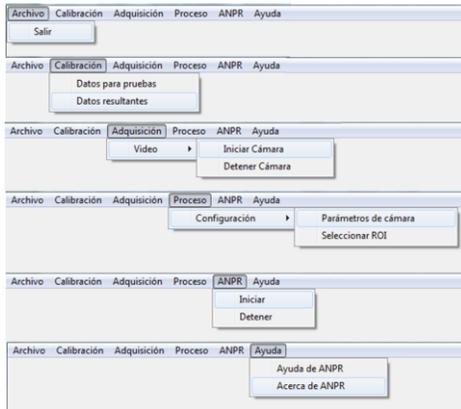
Figura 16. Algoritmo de seguimiento

## 6. Diseño del sistema GUI

Este capítulo especifica cómo está desarrollado nuestro GUI (Interfaz Gráfica de Usuario). El software tiene 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



**Figura 17.** Menú del sistema

En la opción Calibración están especificados los procesos que se describieron en el capítulo de Instalación y calibración del sistema de adquisición de imágenes.

Para el proceso de Adquisición de imágenes por señal de video, se inicia en la opción Adquisición del menú de nuestra interfaz.

Para los procesos de detección y extracción se encuentra en la opción ANPR → Iniciar.

El sistema también cuenta con una opción de ayuda para alegar a cualquier duda sobre las opciones del menú de la interfaz.

## 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.

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.

## BIBLIOGRAFIA

- [1] Wikipedia. “Automatic number plate recognition”. [Online] [http://en.wikipedia.org/wiki/Automatic\\_number\\_plate\\_recognition](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](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](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](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/ffd0AFD20D23F2CCA0DDB83CC9B4F9142A1D9767C94.aspx?s=opencv#L103> [2011]