

# Normalización de imágenes de placas vehiculares a través de corrección geométrica

Carlos Leonardo Choez Álvarez<sup>(1)</sup>, Steve Fernando Salas Guerrero<sup>(2)</sup>, Boris X. Vintimilla<sup>(3)</sup>  
Facultad de Ingeniería Eléctrica y Computación  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 vía Perimetral  
Apartado 09-01-5863. Guayaquil-Ecuador  
cchoez@fiec.espol.edu.ec<sup>(1)</sup>, ssalas@fiec.espol.edu.ec<sup>(2)</sup>, boris.vintimilla@espol.edu.ec<sup>(3)</sup>

## Resumen

*El proyecto en general consiste en implementar un sistema de control de acceso vehicular mediante el reconocimiento del número de placa de manera automática usando una cámara y algoritmos de procesamiento de imágenes incluyendo Reconocimiento Óptico de Caracteres (OCR). Para el reconocimiento de las placas se tomara una imagen al vehículo al momento de entrar a un parqueadero específico.*

*Este proyecto consta de cinco partes: 1. Detección y extracción de placas. 2. Normalización de imágenes de placa. 3. Segmentación de placas. 4. Reconocimiento de número de placas. 5. Interfaz gráfica. Mas el alcance de nuestro proyecto es la Normalización de imágenes de placa. El cual consiste en que una vez que la placa ha sido detectada, la imagen de la placa debe ser normalizada. En este sentido el objetivo principal de este proyecto es implementar una aplicación para alinear la imagen de la placa, así como ajustar la imagen resultante para los posteriores pasos de reconocimiento del número de placa vehicular. Para lo cual se implementaron tres algoritmos que son 1. Detección de silueta mediante esquinas de Harris, 2. Detección de bordes de la placa mediante detección de Blobs y 3. Detección de bordes mediante aproximación de polígonos. Con las pruebas realizadas se llevo a determinar que el tercer algoritmo nos dio mejores resultados, mostrándonos una imagen corregida con la mayoría de las características de la placa bien definidas de tal manera que las posteriores etapas del proyecto global tengan la facilidad de trabajar con estos resultados.*

**Palabras Claves:** OCR, Reconocimiento de números de placa.

## Abstract

*The project overall is to implement an access control system through the automatically recognition of vehicle license plate number using a camera with image processing algorithms that includes Optical Character Recognition (OCR). For the license plate recognition a picture will be taken when the vehicle enters in a specific parking lot.*

*This project has five parts: 1. Detection and extraction of license plates. 2. Images Standardization for license plates. 3. License plates recognition. 4. Number license plate recognition. 5. Graphical interface. But the scope of this project is plate image standardization. This means that once the license plate is detected, the image should be standardized. In this sense, the main objective of this project is to deploy an application to align the image on the license plate, and adjust the resulting image for subsequent steps in recognizing of the license plate number. To which were implemented three algorithms that are 1. Harris corners silhouette detection, 2. license plate edge detection through blobs detection and 3. edge detection through polygon approximation.*

*With the tests it was determined that the third algorithm gave better result, showing a corrected image with most of the license plate characteristics well defined in a way that the later stages of the global project will have working facilities with these results .*

**Keywords:** OCR, License plate numbers recognition.

## 1. Introducción

La altos niveles de inseguridad han generado una gran necesidad de controlar el acceso a los

parqueaderos, tanto en el ingreso de los vehículos como el tiempo de permanecía en los mismos. Existen varias soluciones actualmente en el mercado que van desde simples barreras mecánicas manuales hasta sofisticados sistemas de control con cámaras de

circuito cerrados, cuyo costo varía según lo sofisticado que sea cada sistema.

El mayor problema que existe con estos sistemas de control de acceso vehiculares es que en la actualidad en la mayoría de garitas se lleva un control manual que está sujeto a errores humanos, ya que muchas veces esta información no es correctamente tomada, organizada ni almacenada, por lo tanto se carece de datos detallados que podrían servir para algún problema de seguridad que llegue a suscitarse. Además de mediante el uso de estos datos permitir la predicción de futuros inconvenientes que se puedan presentar en los parqueaderos.

Por lo tanto es de mucha importancia guardar la información detallada de la fecha y hora de ingreso y salida de los vehículos, adicionalmente de reconocer los números de su placa para así tener control completo, de esta manera se podría mantener una organización del estacionamiento y poder obtener información tales como tiempo de permanencia en el estacionamiento, cantidad de parqueos disponibles.

El proyecto está dividido en varios módulos:

- Modulo de adquisición
- Modulo de normalización de placas
- Modulo de segmentación de placas
- Modulo de reconocimiento de placas

Esta tesis se enfoca específicamente en el modulo “Normalización de placas” Para lo cual se recibe una imagen ROI que define la región de interés (placa del vehículo) sobre la imagen original.

El propósito es evitar la presencia de información indeseable por lo cual la imagen será procesada mediante algoritmos de restauración y realzado de imágenes de tal forma de eliminar la presencia de ruido y de emborronamiento, mientras que se usarán transformaciones geométricas para corregir los problemas de orientación de la placa.

El resultado de nuestro tema debe ser una placa alineada tanto horizontal como verticalmente, de esta manera quedaría ajustada para los siguientes pasos de segmentación y reconocimiento de la placa vehicular.

## 2. Materiales y Funcionamiento

### 2.1. Materiales.

Los principales componentes que utilizamos en este proyecto fueron.

Software:

- Visual Studio 2008 – C++
- OpenCV

### 2.2. Funcionamiento.

Para encontrar la mejor solución para la normalización y corrección geométrica se implementaron dos algoritmos previos, los cuales se describen a continuación.

**Algoritmo 1: Detección de la silueta de la placa mediante esquinas de Harris.** En esta primera etapa se buscó corregir geoméricamente la silueta de la placa buscando las líneas de los bordes superiores e inferiores de la placa mediante la transformada de Hough, para luego detectar las esquinas de estas líneas, mediante el detector de esquinas de Harris y así con las coordenadas de estas esquinas poder proyectar la silueta detectada a una imagen base mediante la función WarpPerspective.

Para su implementación se realizan los siguientes pasos.

- Adquisición de la imagen.
- Preprocesamiento
- Binarización de imagen
- Detección de bordes mediante Canny
- Detección de líneas mediante las transformadas de Hough
- Detección de esquinas mediante CornerHarris
- Proyectar a una imagen base mediante WarpPerspective.

**Algoritmo 2: Detección de bordes mediante aproximación de polígonos.** En esta tercera etapa se busca corregir geoméricamente la silueta de la placa, reutilizando funciones implementadas en etapas anteriores tales como Canny, cvWarpPerspective.

En esta etapa se presenta una solución que consiste en aplicar un filtro para el ruido gaussiano, para luego aplicar Canny, niveles de binarización, y con esto encontrar contornos para luego por medio de varias heurísticas seleccionar un polígono que se aproxime de mejor manera a la silueta de la placa, una vez encontrado este polígono se procede a encontrar los puntos de las esquinas para proyectarlo a una imagen con una posición correcta, de esta manera se realiza una corrección geométrica de la placa.

La solución planteada en la presente etapa consta de los siguientes pasos:

- Adquisición de la imagen
- Filtro de ruido Gaussiano
- Binarización
- Detección de bordes
- Encontrar contornos

- Aproximar de polígonos
- Calcular área de polígonos
- Aplicar heurísticas para seleccionar polígono
- Seleccionar contorno
- Proyectar imágenes

Para una mejor explicación se muestra las imágenes de entrada y resultantes por cada paso del algoritmo.

#### Filtro de ruido Gaussiano

Imagen de Entrada



Imagen Filtro Gaussiano



Figura 1. Filtro de ruido Gaussiano

#### Binarización

Imagen de Entrada



Imagen Binarizada



Figura 2. Filtro de Canny

#### Detección de bordes

Imagen de Entrada



Imagen Filtro de Canny



Figura 3. Filtro de Canny

#### Encontrar contornos

Imagen de Entrada



Imagen de contornos



Figura 4. Detección de contornos

#### Aproximar de polígonos

Imagen de Entrada



Imagen Aprox polígono



Figura 5. Imagen de placa en la cual se detectaron contornos

#### Aplicar heurísticas para seleccionar polígono



Figura 6. Imagen de placa en la cual se muestran los contornos seleccionados

#### Proyectar imágenes

Imagen de Entrada



Imagen Resultante



Figura 7. Imagen de placa en la cual se muestran la imagen resultante rotada.

#### Algoritmo Final: Detección de silueta de la placa mediante detección de Blobs.

El algoritmo final está basado en el algoritmo dos “Detección de bordes mediante aproximación de polígonos”, realizando un análisis más profundo.

- Encontrar valores de parámetros constantes
  - Mayor ángulo de rotación, en los planos XY, XZ, YZ
  - Rango de binarización
  - Porcentaje de relación entre el área de la placa y la caja englobante.
- Resolver problemas detectados:
  - Problema de orientación.
  - Problemas de bordes cortados.
  - Problema de proyección de los puntos detectados

### 3. Resultados Experimentales

A continuación se muestran los resultados obtenidos de las pruebas de campo realizadas con las imágenes proporcionadas por el grupo de detección y extracción de placas, el cual obtuvo 10 imágenes para cada uno de los 24 puntos, dando un total de 240 imágenes analizadas.

#### 3.1.- Pruebas de campo

Sobre los 24 puntos señalados por el grupo de adquisición en base a parámetros establecidos por el mencionado grupo, se procedió a realizar el

reconocimiento sobre las placas tomadas en cada uno de los puntos.

Los puntos mostrados de color verde en la figura encierran los puntos en los cuales se obtuvo un reconocimiento óptimo. Los puntos amarillos son aquellos cuyo porcentaje de reconocimiento es aceptable, es decir la placa fue corregida sin presentar mayores complicaciones; mientras que en los puntos rojos las correcciones geométricas no fueron buenas esto es debido a la calidad de las imágenes obtenidas en este punto.

- Puntos analizados: 24
- Tomas: 240
- Corrección aplicada: Alineamiento

Color	Escala	%
Verde	Excelente	75%
Rojo	Buena	16.6%
Amarillo	Malo	8.33%

Tabla 1: Resultados del análisis

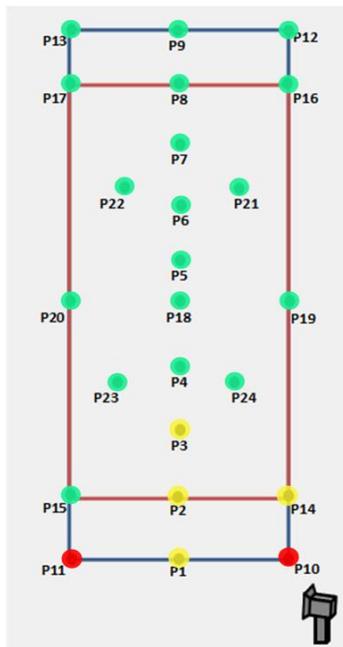


Figura 8. Resultados del análisis por puntos

Imagen de Entrada



Imagen Resultante



Figura 9. Imágenes del análisis punto 1

Imagen de Entrada



Imagen Resultante



Figura 10. Imágenes del análisis punto 5

Imagen de Entrada



Imagen Resultante



Figura 11. Imágenes del análisis punto 8

Imagen de Entrada



Imagen Resultante



Figura 12. Imágenes del análisis punto 19

Imagen de Entrada



Imagen Resultante



Figura 12. Imágenes del análisis punto 22

#### 4. Conclusiones

El algoritmo usado en la primera etapa es muy inestable y es muy susceptible a sombras lo cual provoca que las líneas Hough no siempre cubran el borde superior e inferior por completo. La debilidad del algoritmo está en las líneas de Hough que es la parte clave, por lo tanto se tornaba muy inestable.

El haber utilizado blobs le dio una mayor posibilidad de detectar la silueta de la placa, pero al momento de buscar las esquinas con Harris el algoritmo mostró problemas, tal como no detectar correctamente las esquinas.

La corrección geométrica aplicada en este proyecto muestra una imagen corregida con la mayoría de las características de la placa bien definida de tal manera que las posteriores etapas del proyecto global tengan la facilidad de trabajar con estos resultados.

Para el algoritmo final se utilizó como base el algoritmo anterior es decir la corrección geométrica,

adicionándole como clave el tener un patrón de la placa a analizar, para mediante estos patrones lograr detectar la forma geométrica de la placa y su distorsión y de esta manera al haberla detectado saber a qué tamaño y forma patrón debe llegar a ser corregida (Proyectada).

Este algoritmo puede ser implementado para distintos tipos de placas sin importar el tipo y la cantidad de caracteres, ya que únicamente se debe realizar ajustes en los patrones que recibe el algoritmo.

Cabe indicar que las conclusiones, sintetizan la contribución del conocimiento del autor al mundo.

## 5. Agradecimientos

Agradecemos a Dios, a nuestros padres por brindarnos siempre su apoyo y confianza incondicional. A nuestros amigos y a todas aquellas personas que de una u otra manera nos ayudaron a forjarnos profesionalmente.

## 6. Referencias

- [1] Computer Vision Algorithms in Image Algebra, *Gerhard X. Ritter and Joseph N. Wilson*. Florida: CRC, 1996
- [2] Computer Vision, *Gérard Medioni, Sing Bing Kang*. July 21, 2004
- [3] Wikipedia. “Automatic number plate recognition”. Disponible en [http://en.wikipedia.org/wiki/Automatic\\_number\\_plate\\_recognition](http://en.wikipedia.org/wiki/Automatic_number_plate_recognition)
- [4] Structural Analysis Reference. Disponible en [http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/OpenCVRef\\_StructAnalysis.htm](http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/OpenCVRef_StructAnalysis.htm)
- [5] CV Reference Manual. Disponible en [http://www.seas.upenn.edu/~bensapp/opencvdocs/ref/opencvref\\_cv.htm](http://www.seas.upenn.edu/~bensapp/opencvdocs/ref/opencvref_cv.htm)
- [6] cvCanny Method. Disponible en <http://www.emgu.com/wiki/files/1.3.0.0/html/08eaf5db-eeb2-62c9-578e-ab1de01709d0.htm>
- [7] Operaciones Morfológicas. Disponible en [http://www.tsc.uc3m.es/imagen/Curso\\_ProcesadoMorfologico/Contenido/Operaciones/OperacionesMorfologicas.html](http://www.tsc.uc3m.es/imagen/Curso_ProcesadoMorfologico/Contenido/Operaciones/OperacionesMorfologicas.html)
- [8] Elementos del procesamiento morfológico. Disponible en [http://www.tsc.uc3m.es/imagen/Curso\\_ProcesadoMorfologico/Contenido/Elementos/Elementos.html](http://www.tsc.uc3m.es/imagen/Curso_ProcesadoMorfologico/Contenido/Elementos/Elementos.html)
- [9] Transformaciones Geométricas. Disponible en <http://dis.um.es/~ginesgm/files/doc/pav/tema4.ppt>
- [10] Filtros y Transformaciones Locales. Disponible en <http://dis.um.es/~ginesgm/files/doc/pav/tema3.ppt>
- [11] Transformaciones Geométricas. Disponible en <http://alereimondo.no-ip.org/OpenCV/uploads/41/tema4.pdf>
- [12] Aproximación de Polígonos. Disponible en [www3.espe.edu.ec:8700/bitstream/21000/248/.../T-ESPE-027516.pdf](http://www3.espe.edu.ec:8700/bitstream/21000/248/.../T-ESPE-027516.pdf)
- [13] CvBlob. “Blob library for OpenCV” Disponible en <http://code.google.com/p/cvblob/>
- [14] Geometric Image Transformations. Disponible en [http://opencv.willowgarage.com/documentation/cpp/geometric\\_image\\_transformations.html](http://opencv.willowgarage.com/documentation/cpp/geometric_image_transformations.html)