

# CONTROL BORROSO Y VISIÓN ARTIFICIAL EN EL GUIADO DE VEHÍCULOS

Carlos Ortega, Luis A. L. Nozal, Joseba L. Arroyabe y José Luis Martín

Departamento de Electrónica y Telecomunicaciones. Universidad del País Vasco  
Escuela de Ingenieros, Alameda de Urkijo, s/n 48013 BILBAO  
{carlos, luis, josebal, joseluis}@bise08.bi.ehu.es

## RESUMEN

Este trabajo describe una aplicación de guiado autónomo de vehículos basada en visión artificial, donde el uso de un controlador borroso simplifica notablemente el análisis de las imágenes requerido. Dicho análisis se reduce a la división de las imágenes captadas en 9 zonas, y a la posterior cuenta del número de pixels "oscuros" (discriminados mediante un umbral fijo) en cada una de ellas. Con estos datos se generan los valores de las seis variables de entrada del controlador borroso, que proporciona como variables de salida la velocidad y el ángulo de giro del vehículo. El conjunto de reglas borrosas utilizado es muy reducido, y el procesamiento involucrado permite que el sistema trabaje en tiempo real.

## 1. INTRODUCCIÓN

En el presente trabajo se presenta un sistema de guiado de vehículos que hace uso de una combinación de visión artificial y un control borroso. La idea de fondo de este sistema es que las tareas propias de la visión artificial se simplifican en gran medida si su resultado se utiliza de forma borrosa.

El controlador borroso utilizado ha sido diseñado en el Grupo de Diseño Electrónico del Departamento de Electrónica y Telecomunicaciones de la E.T.S.I.I. e I.T. de Bilbao. Su arquitectura Harvard es similar a la de un microcontrolador RISC, y su juego de instrucciones está optimizado para llevar a cabo las tareas propias del procesamiento borroso: cálculo del grado de pertenencia de las variables de entrada a subconjuntos borrosos, cálculo de las variables de salida en términos borrosos, y desborrosificación de dichas variables de salida. Adicionalmente, su estructura interna de 5 etapas en pipeline reduce el tiempo de procesamiento, lo que permite trabajar hasta con 32 variables en tiempo real, lo que suele bastar para la mayoría de las aplicaciones. Los detalles de este controlador se exponen en [1].

Este controlador borroso ya ha sido utilizado con éxito en otra aplicación de guiado de vehículos, empleando sensores de ultrasonidos para medir distancias a los obstáculos del recorrido, que son fácilmente esquivados por el vehículo [2].

## 2. DESCRIPCIÓN DE LA APLICACIÓN

La aplicación que describiremos a continuación consiste en dotar a un vehículo de la capacidad de recorrer un circuito, señalado mediante unas balizas situadas estratégicamente. En cada una de estas balizas hay un símbolo de color negro representado sobre un fondo blanco.

La forma de proceder del vehículo es la siguiente: en primer lugar "ve" una de estas balizas y se aproxima hacia ella; una vez que se ha acercado lo suficiente reconoce el símbolo representado en la baliza, y por último realiza la acción que indica dicho símbolo. Las posibles acciones son continuar el recorrido girando hacia la izquierda, continuar el recorrido girando hacia la derecha, o finalizar el recorrido. En cualquiera de los dos primeros casos el vehículo gira hacia la dirección indicada y busca una nueva baliza que le indique el siguiente tramo del circuito.

### 2.1. Procesamiento de las imágenes

Las imágenes captadas por la cámara de vídeo incorporada al vehículo son digitalizadas a un tamaño de 256 x 256 pixels. Cada pixel se codifica con 8 bits, correspondientes a 256 niveles de gris.

En cada imagen se intenta detectar alguno de los símbolos que aparecen en la figura 1. Para ello se divide cada imagen en nueve zonas, y se cuenta el número de pixels "oscuros" (discriminados mediante un umbral fijo) en cada una de ellas. Con estos datos el controlador borroso es capaz de generar las órdenes adecuadas para controlar la velocidad y la dirección del vehículo.



Figura 1. a) Zonas de la imagen. b) Símbolos utilizados.

El procesamiento de las imágenes puede ser tan elemental en esta aplicación gracias a que el controlador borroso es capaz de actuar correctamente con unas variables de entrada aproximadas. Como ventaja adicional, este procesamiento es tremendamente rápido, lo que permite que el sistema trabaje en tiempo real.

### 2.2. Análisis y control borroso

El controlador borroso sigue dos estrategias fundamentales: el acercamiento a un símbolo y la identificación de dicho símbolo. Para ello únicamente necesita 6 variables de entrada, correspondientes a combinaciones lineales del número de puntos "oscuros" de las siguientes zonas de la imagen (ver figura 1): a) 1, 2 y 3; b) 7, 8 y 9; c) 1, 4 y 7; d) 2, 5 y 8; e) 3, 6 y 9;

f) imagen completa. Los actuadores de salida son dos variables, correspondientes a la velocidad y el ángulo de giro del vehículo.

La figura 2 muestra ejemplos de las reglas que sigue el controlador borroso para calcular el valor de los actuadores de salida. Estas reglas se han representado en forma de tablas, para facilitar su comprensión. Las entradas a las tablas se corresponden con los términos lingüísticos de dos de las citadas variables de entrada, mientras que se obtiene como resultado los términos lingüísticos de una de las variables de salida. No hay que olvidar que el valor final de los actuadores de salida se determinará tras el proceso de desborrosificación de dichos actuadores.

En el controlador borroso se utilizan ciertas simplificaciones en estas reglas, que aceleran el procesamiento borroso y reducen notablemente los recursos requeridos en la implementación del propio controlador.

|                          |            | f) Imagen completa |         |           |
|--------------------------|------------|--------------------|---------|-----------|
|                          |            | Vacío              | Pequeña | Grande    |
| d) Zona vertical central | Vacío      |                    | Lenta   | Muy_lenta |
|                          | Poco_lleno |                    | Normal  | Muy_lenta |
|                          | Muy_lleno  |                    | Rápida  | Nula      |

|                            |            | c) Zona vertical derecha |            |           |
|----------------------------|------------|--------------------------|------------|-----------|
|                            |            | Vacío                    | Poco_lleno | Muy_lleno |
| e) Zona vertical izquierda | Vacío      | Centro                   | Izq_poco   | Izq_poco  |
|                            | Poco_lleno | Dcha_poco                | Centro     | Izq_poco  |
|                            | Muy_lleno  | Dcha_poco                | Dcha_poco  | Centro    |

|                          |            | a) Velocidad |           |
|--------------------------|------------|--------------|-----------|
|                          |            | Pequeña      | Grande    |
| d) Zona vertical central | Vacío      | Lenta        | Muy_lenta |
|                          | Poco_lleno | Normal       | Muy_lenta |
|                          | Muy_lleno  | Rápida       | Nula      |

|                            |            | b) Giro (al acercarse a la baliza) |            |           |
|----------------------------|------------|------------------------------------|------------|-----------|
|                            |            | Vacío                              | Poco_lleno | Muy_lleno |
| e) Zona vertical izquierda | Vacío      | Centro                             | Izq_poco   | Izq_poco  |
|                            | Poco_lleno | Dcha_poco                          | Centro     | Izq_poco  |
|                            | Muy_lleno  | Dcha_poco                          | Dcha_poco  | Centro    |

Figura 2. Reglas borrosas para obtener: a) La velocidad. b) El ángulo de giro al acercarse a una baliza.

### 2.3. Efecto sobre los actuadores

En los prototipos de vehículos de tamaño reducido se suelen utilizar motores eléctricos, tales como los motores de continua, motores “paso a paso” y servomotores. En la aplicación que nos ocupa se ha empleado tanto un motor de continua como un servomotor. Ambos motores son controlados por señales PWM (*pulse width modulation* – modulación por anchura de pulsos) generadas por un microcontrolador. De esta forma, este microcontrolador se encarga de recoger los valores de salida del controlador borroso, implementado en una FPGA, y adecuar las señales PWM a estos actuadores.

### 2.4. Arquitectura del sistema

El sistema completo consta de tres partes fundamentales: un conversor A/D de vídeo, una FPGA y un microcontrolador.

La primera parte simplemente muestrea la señal de vídeo procedente de una cámara de vídeo, extrae los sincronismos vertical y horizontal de cada imagen y genera una señal que identifica el campo de las imágenes. En la FPGA se ha implementado tanto el controlador borroso como la parte de procesamiento de las imágenes, utilizando principalmente el lenguaje VHDL para el diseño del circuito. Por último, el

microcontrolador se encarga de actuar sobre los motores de la manera indicada anteriormente.

### 2.5. Simulación software

Como paso previo a la implementación de un prototipo del sistema se ha desarrollado un programa en C++ que simula el comportamiento del controlador borroso. La figura 3 muestra el resultado de dos simulaciones sobre imágenes reales, consistentes en la imagen original, la imagen binarizada, los valores de las variables de entrada y los actuadores de salida.

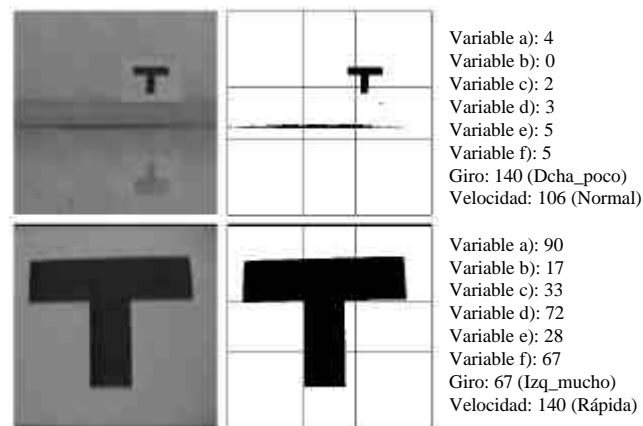


Figura 3. Simulación software. Parte superior: fase de acercamiento a la baliza. Parte inferior: fase de identificación de un símbolo.

### 3. CONCLUSIONES

El análisis que exige la visión artificial se simplifica cuando sus resultados van a ser utilizados de forma borrosa para el control de un vehículo.

El sistema presentado en este documento consigue unos resultados excelentes utilizando una arquitectura sencilla, capaz de actuar en tiempo real.

### 4. AGRADECIMIENTOS

Este trabajo ha sido financiado por el Departamento de Educación, Universidades e Investigación del Gobierno Vasco en el marco del proyecto de investigación PI-1998-41.

### 5. REFERENCIAS

- [1] Aranguren, G., Barrón, M., Arroyabe, J.L., and Carreira, G.G., "A pipe-line fuzzy controller in FPGA", Proc. of 6<sup>th</sup> IEEE International Conference on Fuzzy Systems, 1997, pp. 635-640.
- [2] Arroyabe, J.L., Aranguren, G., Nozal, L.A. and Martín, J.L., "Autonomous vehicle guidance with fuzzy algorithm", Proc. 2000 International Conference on Industrial Electronics, Control and Instrumentation (IECON'2000), pp. 1503-1508.