

Vehículo autónomo a escala para el seguimiento de marcas viales de carretera

Jorge de Castro Casares

Asociación de Microbótica de la Universidad de Valladolid, Pº del Cauce 59, 47011 Valladolid, España.
jorge.decastro.es@ieee.org

Eduardo Zalama Casanova

Instituto de Tecnologías Avanzadas de la Producción, Pº del Cauce 59, 47011 Valladolid, España.
ezalama@eii.uva.es

Eusebio de la Fuente López

Instituto de Tecnologías Avanzadas de la Producción, Pº del Cauce 59, 47011 Valladolid, España.
efuente@eii.uva.es

Jaime Gómez García-Bermejo

Instituto de Tecnologías Avanzadas de la Producción, Pº del Cauce 59, 47011 Valladolid, España.
jaigom@eii.uva.es

Resumen

En este artículo se presenta el desarrollo de un vehículo a escala que permite su conducción autónoma mediante el auto guiado a través de la detección de las marcas viales de la carretera. El vehículo se basa en un modelo de radiocontrol modificado al que se le ha añadido una cámara para la detección de marcas viales y semáforos, y un láser lidar para la detección de obstáculos. Además, incluye un procesador Odroid con sistema base Linux Ubuntu y ROS (Robot Operation System) [8] para el control de alto nivel, y un microcontrolador Arduino para el control de los motores. El sistema de control está basado en el modelo de arquitectura Subsumption en el que se incluyen tres módulos con diferentes niveles de prioridad. El seguimiento de marcas viales utiliza un algoritmo RANSAC para la detección de rectas y un sistema de control PID. El sistema de control incluye la detección de obstáculos y maniobras de adelantamiento, y la detección de semáforos. La arquitectura propuesta ha sido verificada en la edición 2017 de la Seat Autonomous Driving Challenge alcanzando la primera posición.

Palabras clave: Conducción Autónoma, Visión Artificial, Reconocimiento de Carretera.

1 INTRODUCCIÓN

La conducción autónoma es uno de los retos de investigación al que más esfuerzos se está dedicado en los últimos años. Todos los fabricantes de vehículos se han lanzado a una carrera de desarrollo de vehículos autónomos que comportarán grandes

beneficios en términos de ahorro de costes y seguridad. La conducción autónoma ha sido objeto de estudio desde hace varios años utilizando cámaras para la detección de marcas viales [4],[5] mientras que otros sistemas utilizan la detección mediante láser [13],[14],[15].

En este contexto general, las competiciones de conducción autónoma son una manera eficaz de comprobar los desarrollos y verificar su grado de madurez en un entorno de *benchmarking*. Así una de las primeras competiciones ha sido el Darpa Grand Challenge que inicialmente, en 2004, se planteó como una competición para ir de un lugar a otro en el desierto de Mojave, y en 2007 se introdujo en un circuito urbano. Sin embargo el desarrollo de una competición con coches reales supone un gran coste por lo que, en los últimos años, diferentes fabricantes y universidades han promovido el desarrollo de competiciones utilizando coches a escala. El aumento de la capacidad de procesamiento de los microcontroladores y la reducción de tamaños de los sensores ha permitido mantener las condiciones de operación en condiciones de escala. Así, cabe destacar por ejemplo las competiciones organizadas por Audi en el Audi driving Challenge [1], Seat en el Seat Autonomous Driving Challenge [12] o General Motors en el Sae Autodrive Challenge [11].

El resto del artículo se organiza de la siguiente forma. En la sección 2 se describe la estructura del vehículo y los elementos que lo integran. La sección 3 describe el sistema de control así como una de los módulos que lo integran. La sección 4 se dedica a la descripción del módulo de detección de líneas y

seguimiento de la carretera. Finalmente, la sección 5 resume las conclusiones obtenidas.

2 ESTRUCTURA DEL VEHÍCULO

Para la realización del vehículo se ha utilizado un prototipo previo a escala 1:10 de un coche con geometría Ackermann basada en dos ruedas traseras motrices paralelas y dos ruedas delanteras paralelas de direccionamiento (Figura 1). Como equipo de computación principal a bordo encargado del control del vehículo se ha utilizado un Odroid XU4.



Figura 1.- Prototipo a escala de coche autónomo.

El prototipo cuenta para la detección de entorno una cámara Intel Realsense ZR300 [9]. Esta cámara se utiliza principalmente como elemento para la detección de las marcas viales de la carretera. Para la detección de obstáculos el prototipo dispone de un sistema láser RPLIDAR A2 [10]. El uso del LIDAR se reserva exclusivamente a las tareas de aparcamiento, adelantamiento de vehículos y detección de obstáculos. También se puede utilizar para la construcción de mapas de entorno y localización del vehículo.

Además, el prototipo cuenta con un microcontrolador para la señalización de maniobras, el control de iluminación, el control de las ruedas motrices y la dirección del vehículo. Para este fin se ha elegido un Arduino Nano por su facilidad de programación y bajo costo.

En cuanto al sistema motriz, se ha utilizado un motor con escobillas y codificador incremental, lo que permite realizar un control de velocidad, así como realizar una estimación de posición del robot mediante odometría.

3 SISTEMA DE CONTROL

El vehículo debe circular por carreteras de doble carril de 80 centímetros de ancho, en el que se pueden incluir cruces, semáforos, zonas de aparcamiento (su presencia se indica mediante códigos QR) u obstáculos en la carretera. En la Figura se muestra una imagen de entorno de circulación correspondiente a la competición Seat Autonomous Driving Challenge de 2017 [12].



Figura 2.- Imagen del entorno en el Seat Autonomous Driving Challenge 2017 en Barcelona.

3.1 ESTRUCTURA SOFTWARE

La programación del sistema de control del vehículo se ha realizado mediante programación en Robot Operating System (ROS) [8]. Se trata de un sistema operativo de código abierto (Open source) que cuenta con una comunidad desarrolladora muy extensa y facilita la programación modular basada en componentes, y abstracción hardware.

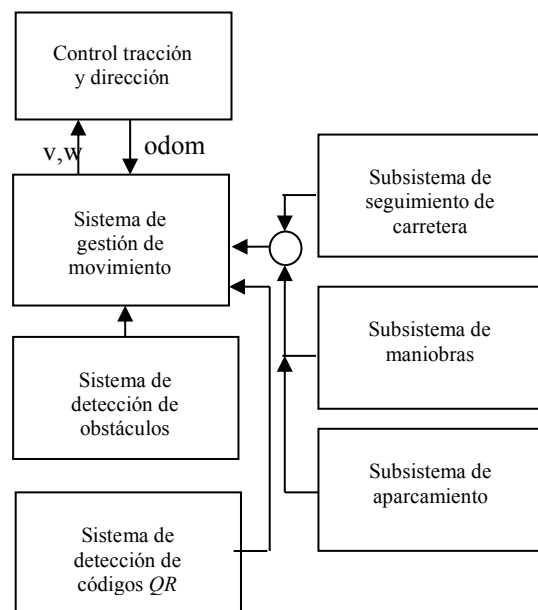


Figura 3.- Diagrama de bloques del sistema de control.

En la figura 3 se muestra un detalle de los módulos que componen el sistema.

El sistema de tracción y dirección se ejecuta en la placa Arduino, a partir de las referencias de velocidad lineal y angular que debe desarrollar el vehículo en función de la actividad a realizar. A su vez, Arduino publica una estimación de la posición del vehículo en base a la odometría de este (integración de movimientos incrementales del vehículo). El sistema de gestión del vehículo utiliza un modelo de arquitectura tipo Subsumption [2] en el que se incluyen tres módulos que se pueden activar dependiendo del nivel de prioridad, en función de la información sensorial. El módulo de menor prioridad es el seguimiento de carretera. Este es el módulo activo en ausencia de obstáculos y otras marcas viales diferentes de la propia carretera. Cuando se detecta un código *QR* correspondiente a un semáforo o una bifurcación se activa el módulo de maniobras y se inhibe el de seguimiento de carretera, con lo cual se ejecuta la acción correspondiente. Por ejemplo si el código corresponde al de un semáforo, se analiza la luz y si esta es roja, el vehículo se detiene a una distancia adecuada. Si lo que se detecta es una bifurcación a la izquierda se activa la maniobra pre-programada de giro a la izquierda, etc. Si el láser detecta un obstáculo en medio de la carretera se activa la maniobra de adelantamiento. Una vez se ha realizado la maniobra correspondiente, ante la ausencia de otros estímulos externos se activa nuevamente el módulo de seguimiento de carretera. El subsistema de aparcamiento se activa de igual forma cuando se detecta el símbolo *QR* pertinente.

4 SEGUIMIENTO DE CARRETERA

El sistema de seguimiento de carretera se compone de dos módulos Ros independientes. El módulo de detección de líneas de la carretera y módulo de control de movimiento que determina la velocidad lineal y angular que debe realizar el vehículo.

4.1 MÓDULO DE DETECCIÓN DE LÍNEAS

Para el seguimiento de las marcas viales nuestro vehículo emplea un algoritmo de detección de rectas en imágenes.

Existen numerosas técnicas para la detección de rectas en imágenes en la literatura de visión por computador.

Una primera alternativa consiste en la detección directa de la recta mediante métodos de optimización basados en mínimos cuadrados. Sin embargo, esta técnica sólo proporciona resultados robustos cuando opera sobre los puntos de imagen correspondientes a una misma línea en las marcas viales de la calzada, los denominados *inliers*, ya que estos puntos son los únicos

datos cuya distribución puede ser explicada por el modelo de una recta de parámetros desconocidos. El resto de los puntos de contorno serán denominados datos atípicos o outliers, debido a su inconsistencia con el modelo de recta buscado, y pueden provenir de otras marcas viales presentes en la escena o de otras características visuales como brillos e imperfecciones de la calzada, elementos en el horizonte, semáforos, etc .

Una alternativa más robusta es el empleo de la conocida Transformada de Hough (TH)[6][3]. Se trata de una técnica propuesta hace ya más de 50 años y que desde entonces ha sido ampliamente aplicada para la detección de muchas primitivas geométricas en procesamiento de imagen. La idea básica de la TH es la construcción de una tabla en el espacio de parámetros. Se transforma cada punto de contorno al espacio de parámetros y se incrementa el acumulador de la celda correspondiente en la tabla. Los parámetros asociados a la celda más votada son los escogidos como modelo. La TH tiene básicamente un problema, que es la dificultad para encontrar los máximos locales en la tabla de acumulación lo que origina cierta imprecisión en la detección de las rectas.

En el presente trabajo proponemos la utilización de una tercera alternativa, el algoritmo para estimación robusta RANSAC, abreviatura para "RANDOM SAMple Consensus", introducido en el trabajo [7], y profusamente utilizado desde entonces por su robustez y eficiencia computacional. RANSAC selecciona de forma aleatoria muestras de n puntos para obtener los parámetros del modelo, pero en lugar de usar una tabla de acumulación, RANSAC calcula, de acuerdo a un umbral de proximidad prefijado, el número de puntos de contorno de la imagen que están *en consenso* con el modelo.

La eficiencia del algoritmo RANSAC estándar se ve mejorada si la elección de las muestras se guía de alguna manera, como alternativa a hacer un muestreo puramente aleatorio entre todos los puntos de contorno. Esto se hace en el caso del sistema de visión del vehículo en dos etapas:

- Dividiendo el área de la imagen en tres zonas que se procesarán por separado. En cada una de estas zonas aparecerá previsiblemente una marca vial correspondiente a cada una de las líneas que aparecen en la calzada: en el área izquierda la línea continua del arcén izquierdo de la vía, en el área central la línea discontinua de la calzada y en el área derecha la línea continua del arcén derecho. Esta división permitirá extraer por separado cada una de las líneas rectas presentes en la imagen, sin que se mezclen los puntos de cada una de ellas en el proceso de detección haciendo de esta forma más eficiente y precisa la detección.
- Imposición de una restricción adicional de distancia entre las muestras aleatorias cada región con objeto de obtener modelos de recta significativos. Si el hipotético modelo de recta se genera a partir de una muestra con dos puntos muy próximos, este va a ser muy impreciso y va a tener pocas posibilidades de prosperar.

Tabla 1. Algoritmo RANSAC

Hasta realizar K iteraciones

I. Hipótesis

- (i) Obtener una muestra al azar de 2 puntos de contorno de entre todos los puntos de contorno.
- (ii) Calcular los parámetros de la recta que pasa por esos 2 puntos.

II. Verificación

- (iii) Determinar el conjunto de consenso, es decir, los puntos en la imagen de contornos que se ajustan a la recta con una tolerancia.
- (iv) Si el número de *inliers* en el conjunto de consenso es el mayor encontrado, guardar los parámetros de este modelo.

End

El principio del algoritmo consiste en buscar el mejor modelo de recta entre todos los puntos de contorno seleccionando aleatoriamente muestras de dos puntos. Se calculan entonces los parámetros de la recta que pasa por los puntos de la muestra y se establece el conjunto de consenso, es decir, los puntos en la imagen de contornos original que pertenecen a la recta, calculados de acuerdo a un umbral de proximidad dado. Este procedimiento se repite K veces, como se muestra en la descripción del algoritmo básico que aparece en la tabla 1.

Después de K iteraciones, si el número de puntos del mejor conjunto de consenso encontrado es superior a un umbral preestablecido el modelo se asume como válido. Como los parámetros del modelo estimado por RANSAC no serán muy precisos, al ser obtenidos a partir de únicamente dos puntos, se suelen recalcular estos parámetros empleando técnicas de mínimos cuadrados sobre todo el conjunto de puntos que soportan ese modelo.

El número de iteraciones K puede determinarse a partir de la probabilidad P de encontrar al menos un conjunto bueno libre de outliers. El RANSAC es un algoritmo no determinista que proporciona un buen modelo sólo con una cierta probabilidad y esta probabilidad se incrementa cuantas más iteraciones se lleven a cabo. En general, al algoritmo se le añade algún criterio de parada, típicamente asociado al momento en el que el número de puntos en consenso sobrepasa un umbral preestablecido.

Si consideramos un número de puntos N con una fracción de *inliers* ε , la probabilidad de seleccionar una muestra con todos *inliers* es ε^s , siendo s el tamaño de la muestra y considerando que $s \ll N$. La probabilidad de no seleccionar una muestra formada totalmente por *inliers* será $(1-\varepsilon^s)$ y $(1-\varepsilon^s)^K$ la probabilidad de no seleccionar una muestra buena en K iteraciones. Entonces, la probabilidad de seleccionar una muestra no contaminada en K intentos al menos una vez será

$P = 1 - (1 - \varepsilon^s)^K$. Por tanto, el número de iteraciones que son necesarios para obtener el modelo con una probabilidad predeterminada P es:

$$K = \frac{\log(1-P)}{\log(1-\varepsilon^s)} \quad (1)$$

Para determinar el número K de iteraciones, se eligen normalmente valores de probabilidad P bastante conservadores, entorno al 95%. En la aplicación que nos ocupa, considerando el porcentaje típico de *inliers* que aparecen en las imágenes, hemos obtenido que con diez iteraciones bastan para encontrar la recta correcta, lo cual hace que el algoritmo proporcione el resultado realmente rápido como se muestra en la Figura 4.

Otro parámetro que es preciso establecer en el algoritmo RANSAC es la tolerancia, que hace referencia a la distancia máxima que puede tener un punto al modelo para que sea considerado como *inlier*. El valor de la tolerancia no debe ser excesivamente bajo para poder asumir los errores del ruido y la digitalización en los puntos de contorno, pero tampoco muy elevado de forma que puntos espurios sean tomados como *inliers*.

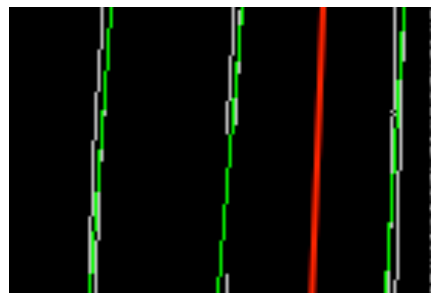


Figura 4.- Resultado de interacción del algoritmo para varias rectas para el cálculo de trayectoria.

4.2 MÓDULO DE CONTROL DE MOVIMIENTO

El módulo de control de movimiento determina las referencias de velocidad lineal y angular que debe realizar el vehículo para seguir la carretera, para lo que utiliza un controlador PID en función de la distancia y ángulo a la línea referencia que debe seguir.

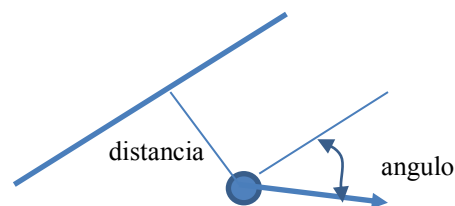


Figura 5.- Detalle de línea de referencia que debe seguir el vehículo. Este se representa mediante círculo con vector de orientación.

La acción de control debe ser tal que se minimice el error, en este caso la distancia y el ángulo a la recta de referencia. La velocidad lineal puede establecerse como una magnitud constante o puede darse una acción de control por la que el vehículo se desplace más rápido en rectas y disminuya al llegar a zonas de curvas. No obstante el control más importante es el determinar la velocidad angular que debe tomar el vehículo para seguir la carretera.

$$w = k_d d + k_\alpha \alpha \quad (1)$$

donde d corresponde a la distancia del robot a la recta tangente de la curva a seguir y α es el ángulo existente entre la orientación del robot y la recta tangente determinada. Los valores de k_d y k_α se determinan experimentalmente hasta conseguir un buen seguimiento de la carretera. La ecuación (1) corresponde a un controlador proporcional, no obstante se puede incluir términos integral y derivativo si se precisa reducir error estacionario u obtener mayor velocidad de respuesta.

5 CONCLUSIONES

En este artículo se ha presentado el desarrollo de un coche a escala para la conducción autónoma y auto guiado mediante la detección de marcas viales. La organización de modular mediante ROS ha facilitado la integración y verificación de los diferentes elementos de forma independiente y distribuida. Aunque inicialmente, los módulos de procesamiento se han ejecutado en un computador externo, finalmente el procesador Odroid se ha mostrado capaz de ejecutar todos los procesos sin apenas pérdidas de rendimiento. La utilización de la transformada de Hough y Ransac se ha mostrado como metodología eficaz para la detección de marcas viales. Por otra parte la utilización de un modelo de arquitectura Subsumption ha facilitado la integración del sistema pues relaja el requerimiento de necesitar una localización precisa del robot y ha permitido desarrollar comportamientos reactivos como el responder a obstáculos imprevistos. Entre las futuras líneas de desarrollo que se proponen está la mejora del sistema de detección de marcas viales mediante un cálculo preciso de la curvatura de la vía pues aunque el sistema propuesto se ha mostrado que realiza un buen seguimiento de la vía el sistema se ha mostrado muy sensible a la elección de parámetros del controlador. También se pretende integrar un subsistema de localización y planificación de rutas. Respecto a otros sistemas de procesamiento utilizados con anterioridad para la detección de marcas viales, se ha obtenido un porcentaje de éxito con el sistema actual de un 80% de los casos frente a

40%, considerando éxito la correcta navegación del vehículo por el carril, permitiendo así la navegación a una mayor velocidad con la misma estabilidad.

Agradecimientos

Agradecemos a Seat y a CARNET por habernos dado la posibilidad de participar en el Seat Autonomous Driving Challenge cuyos resultados se han presentado en este trabajo.

English summary

Autonomous scale vehicle for road following.

Abstract

This article presents the development of a vehicle at scale that allows autonomous driving by self-guided through the detection of road signs. The vehicle is based on a modified radio control model with a camera for the detection of road signs and traffic lights, and a lidar laser for obstacle detection. It also includes an Odroid processor with Ubuntu Linux base system and ROS (Robot Operation System) [5] for high level control, and an Arduino microcontroller for motor control. The control system is based on the Subsumption architecture model, which includes three modules with different priority levels. The tracking of road markings uses a RANSAC algorithm for the detection of straight lines and a PID control system. The control system includes the detection of obstacles and maneuvers, and the detection of traffic lights. The proposed architecture has been verified in the 2017 edition of the Seat Autonomous Driving Challenge reaching the first position.

Keywords: Autonomous Driving, Artificial Vision, Road Recognition.

Referencias

- [1] Audi Driving Challenge (2018). <https://www.audi-autonomous-driving-cup.com/>
- [2] R. A. Brooks (1986), "A Robust Layer Control System for a Mobile Robot", IEEE Journal of Robotics and Automation RA-2, 14-23.
- [3] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, p.111-122, 1981.

- [4] Dellaert, F., & Thorpe, C. (1998). Robust car tracking using Kalman filtering and Bayesian templates. In Proceedings of SPIE (Vol. 3207, p. 72).
- [5] Dickmanns, E. (1998). Vehicles capable of dynamic vision: a new breed of technical beings? *Artificial Intelligence*, 103(1–2), 49–76.
- [6] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM, Vol. 15*, pp. 11–15 (January, 1972)
- [7] Martin A. Fischler and Robert C. Bolles (junio de 1981). «Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography». *Comm. of the ACM* 24 (6): 381-395. doi:10.1145/358669.358692
- [8] «Página oficial de ROS,» [En línea]. Available: <http://wiki.ros.org/>.
- [9] «Página de cámara Intel Real Sense ZR300,» [En línea]. Available: <https://software.intel.com/es-es/realsense/zr300/>.
- [10] «Página Ros RPLidar,» [En línea]. Available: <http://wiki.ros.org/rplidar/>
- [11] Sae Autodrive Challenge (2018). <http://autodrivechallenge.com/> Seat Autonomous Driving Challenge (2018). <http://www.autonomousdrivingchallenge.com/>
- [12] Seat Autonomous Driving Challenge. <http://www.autonomousdrivingchallenge.com/>
- [13] Steller, D., Furstenberg, K., & Dietmayer, K. (2002). Vehicle and object models for robust tracking in traffic scenes using laser range images. In The IEEE 5th international conference on intelligent transportation systems, 2002. Proceedings (pp. 118–123).
- [14] Wang, C., Thorpe, C., Thrun, S., Hebert, M., & Durrant-Whyte, H. (2007). Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26, 889–916.
- [15] Zhao, L., & Thorpe, C. (1998). Qualitative and quantitative car tracking from a range image sequence. In 1998 IEEE computer society conference on computer vision and pattern recognition. Proceedings (pp. 496–501).
- [16] Zielke, T., Brauckmann, M., & von Seelen, W. (1993). Intensity and edge-based symmetry detection with an application to car-following. *CVGIP: Image Understanding*, 58(2), 177–190.

Deben estar ordenadas por orden alfabético y justificadas con la sangría correspondiente.



© 2018 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC 3.0 license (<https://creativecommons.org/licenses/by-nc/3.0>).